LEARNING UNKNOWN CONSTRAINTS WITHOUT UNSAFE DATA VIA OPTIMALITY AND COUNTERFACTUAL REGULARIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Learning from demonstrations (LfD) provides a framework for inferring unknown constraints from locally optimal, constraint-satisfying expert behavior. Existing approaches largely fall into two paradigms, constrained inverse optimal control (CIOC) and inverse constrained reinforcement learning (ICRL). CIOC exploits optimality conditions such as the Karush–Kuhn–Tucker (KKT) conditions but typically assumes known dynamics and structured constraint representations. Meanwhile, ICRL accommodates complex unknown constraints and unknown transition dynamics but often requires extensive online exploration, during which unsafe constraint violations may occur. In this work, we introduce Counterfactual KKT (CF-KKT), a constraint learning framework that leverages learned dynamics and locally optimal demonstrations to recover unknown constraints without requiring known dynamics or additional risky exploration, thereby combining the data efficiency and safety advantages of CIOC with the flexibility of ICRL. First, we use a locally learned differentiable dynamics model to impose KKT-inspired optimality conditions directly on the demonstrations. Second, we use the learned dynamics to generate reward-improving counterfactual behaviors near the demonstrations, revealing behaviors that would be preferable in the absence of the unknown constraint and thus providing synthetic infeasible data. When the constraint parameterization is known, the same learned-dynamics framework enables direct CIOC-based parameter recovery, and we characterize its sensitivity to dynamics misspecification. Across high-dimensional robotic control tasks, our approach learns neural constraint representations with improved safety and data efficiency relative to state-of-the-art offline ICRL baselines.

1 INTRODUCTION

Autonomous systems deployed in safety-critical applications, such as autonomous driving and warehouse operations, must adhere to constraints that encode safety and task specifications (Altman, 1999; Achiam et al., 2017; Borkar, 2005; Mayne et al., 2000; Berberich et al., 2021; Magni & Scattolini, 2004). Given the difficulty of manually engineering such constraints and the advent of data-driven methods, learning from demonstration (LfD) has attracted increasing attention as a promising framework for recovering constraints from locally optimal and constraint-abiding expert trajectories (Chou et al., 2020b; 2022; Quan et al., 2024; Liu et al., 2023). LfD-based constraint learning paradigms are often either based on inverse optimal control (IOC) (Englert et al., 2017; Keshavarz et al., 2011; Johnson et al., 2013; Rickenbach et al., 2024; Wulfmeier et al., 2017; Chou et al., 2020b;a; 2018) or inverse constrained reinforcement learning (ICRL) (Scobee & Sastry, 2019; McPherson et al., 2021; Malik et al., 2021; Papadimitriou et al., 2022; Liu & Zhu, 2022; 2024; Park et al., 2020; Jang et al., 2023). IOC-based methods recover constraints by enforcing consistency with the Karush–Kuhn–Tucker (KKT) conditions that characterize the local optimality of a given set of demonstrations (Rickenbach et al., 2024; Menner et al., 2019; Nguyen et al., 2026; Zhang et al., 2026; Chiu et al., 2026) under the dynamics models (Rickenbach et al., 2024; Menner et al., 2019) and control methodology (Nguyen et al., 2026; Zhang et al., 2026; Chiu et al., 2026) from which the demonstrations were obtained. However, such methods assume access to known, smooth system dynamics, which precludes their applicability to high-dimensional, black-box systems. In contrast, ICRL-based constraint recovery (Scobee & Sastry, 2019; McPherson et al., 2021; Malik et al., 2021; Papadimitriou et al., 2022; Liu & Zhu, 2022; 2024; Park et al., 2020; Jang et al., 2023) accommodates unknown dynamics and flexible neural constraint representations. However, such

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.

methods usually require computation- (McPherson et al., 2021; Liu et al., 2023) and data-heavy (Scobee & Sastry, 2019; Liu & Zhu, 2022; Quan et al., 2024) online exploration, may induce unsafe constraint violations during training (Malik et al., 2021; Liu et al., 2023; Quan et al., 2024), and have few theoretical guarantees of constraint recovery accuracy (Malik et al., 2021; Park et al., 2020; Quan et al., 2024). While recent offline variants Quan et al. (2024) avoid online exploration during training, they still rely on offline data generated by potentially constraint-violating policies to identify unsafe regions, which can be difficult or unsafe to collect in real-world systems.

In this work, we bridge CIOC and ICRL to enable constraint learning under unknown dynamics and constraint representations without requiring constraint-violating data. Our approach, *Counterfactual KKT (CF-KKT)*, learns local dynamics from constraint-satisfying expert demonstrations and uses them to combine KKT-inspired optimality conditions with synthetic reward-improving counterfactual trajectories. These counterfactuals provide information about constraint-violating states without requiring constraint-violating trajectories to be executed in the environment. We further show that locally learned dynamics can support CIOC-based parameter recovery when the constraint parameterization is known. Concretely, our contributions are:

1. For settings with closed-form *a priori* known constraint parameterizations, we present a two-stage framework which first learns system dynamics without additional demonstration data, then leverages the learned system dynamics to infer the unknown constraint via CIOC;
2. For settings with neural network (NN)-represented constraints, we introduce Counterfactual KKT (CF-KKT), which combines KKT-inspired optimality conditions with model-generated counterfactual trajectories to learn unknown constraints without observed constraint-violating data;
3. We present a theoretical analysis of the sensitivity of the recovered constraints to system dynamics misspecification;
4. We evaluate CF-KKT in simulation across low- and high-dimensional systems with black-box dynamics, demonstrating improved constraint recovery and downstream safety over existing offline ICRL methods while structurally using less data, all of which are safe.

Notation Given non-negative integers n_1 and n_2 with $n_1 < n_2$, we define $[n_2] := \{1, \cdots, n_2\}$ and $[n_1, n_2] := \{n_1, \cdots, n_2\}$. We use C^k to denote k-times continuous differentiability. Given a differentiable function $f : \mathbb{R}^n \to \mathbb{R}^m$, we define the Jacobian matrix $\nabla_{\mathbf{x}} f \in \mathbb{R}^{m \times n}$ such that $[\nabla_{\mathbf{x}} f]_{ij} := \frac{\partial f_i}{\partial x_j}$ for each $i \in [m]$, $j \in [n]$.

2 Related Work

Our work relates most closely to constrained inverse optimal control (CIOC), which infers constraints from expert demonstrations using Karush–Kuhn–Tucker (KKT) optimality conditions (Englert et al., 2017; Keshavarz et al., 2011; Johnson et al., 2013; Rickenbach et al., 2024; Chou et al., 2020b; 2018; Menner et al., 2019; Nguyen et al., 2026; Zhang et al., 2026; Chou et al., 2019; Pérez-D'Arpino & Shah, 2017; Chiu et al., 2026). Existing CIOC-based constraint learning methods are typically highly data-efficient because they exploit the local optimality of demonstrations directly (Chou et al., 2020b). However, many such approaches assume *known differentiable dynamics* and structured constraint parameterizations (Chou et al., 2020b;a), which hampers their applicability to high-dimensional and neural network-represented systems. While GP-based approaches relax parameterization assumptions (Chou et al., 2022), they scale poorly with dataset size and still require known dynamics. In contrast, our work considers high-dimensional *unknown dynamics* and *unknown constraint representations* while preserving data efficiency, and enables further data-efficiency gains when the constraint parameterization is provided.

Partly in parallel with CIOC, ICRL-based methods (Liu et al., 2025; 2023) were developed for inferring constraints from expert trajectories generated via constrained MDPs (CMDPs) (Malik et al., 2021). ICRL usually assumes access to a simulator rather than closed-form dynamics (in contrast to CIOC), and can be adapted to settings with unknown system dynamics or rewards (Liu & Zhu, 2022; 2024; Park et al., 2020; Jang et al., 2023). However, ICRL-based constraint recovery typically relies on iterative online environment exploration and constrained RL subroutines, leading to high computational cost, high sample complexity, and potentially unsafe exploration during training, which are unacceptable in time-sensitive and safety-critical settings. Uncertainty-aware ICRL methods aim to reduce constraint violation risk (Xu & Liu, 2024; Subramanian et al., 2024) but nonetheless still require online data. Offline ICRL (Liu et al., 2025) avoids online exploration. The primary prior offline

method, ICSDICE (Quan et al., 2024), constructs higher-reward constraint-violating distributions to identify unsafe regions (echoing earlier CIOC work by Chou et al. (2018)); however, doing so still requires a fixed dataset containing behaviors that are not guaranteed to satisfy the unknown constraint.

In contrast to the literature surveyed above, our work combines the safety and data efficiency advantage of CIOC with the representational flexibility of ICRL. Our approach leverages learned local dynamics to enable KKT-inspired inference under unknown dynamics, while synthetic reward-improving counterfactual data provide off-demonstration evidence of infeasibility without requiring observed constraint violations. In contrast, Chou et al. (2019) uses hit-and-run sampling to generate counterfactuals but still relies heavily on known parametric constraints and explicit dynamics.

3 Preliminaries and Problem Formulation

We first review ICRL methods, with an emphasis on ICSDICE (Quan et al., 2024), which serves as our primary offline ICRL baseline (Sec. 3.1). We then introduce the constrained inverse optimal control formulation and its associated KKT conditions (Sec. 3.2), before formally stating the constraint-recovery problem considered in this work (Sec. 3.3).

3.1 Inverse Constrained Reinforcement Learning (ICRL)

Consider a discrete-time, continuous-space constrained Markov decision process (CMDP) $(\mathcal{S}, \mathcal{A}, f, r, c, \bar{\rho}, \gamma)$ specified by a state space $\mathcal{S} \subseteq \mathbb{R}^n$, control (action) space $\mathcal{A} \subseteq \mathbb{R}^m$, system (transition) dynamics map $f : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$, reward map $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, constraint map $c : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, initial state distribution $\bar{\rho}$, and discount factor $\gamma \in [0, 1)$, with $f(\cdot, \cdot)$, $r(\cdot, \cdot)$, and $c(\cdot, \cdot)$ all in C^1. Given a CMDP with *known* constraint function $c(\cdot, \cdot)$, we consider the following horizon-T *forward constrained reinforcement learning (FCRL) problem*, which computes the optimal policy $\pi : \mathcal{S} \to \mathcal{A}$ that maximizes, in expectation, the discounted cumulative reward $\sum_{t=0}^{T} \gamma^t r(x_t, u_t)$ subject to the hard constraint $c(x_t, u_t) \leq 0$ over all times $t \in [0, T]$:

$$\pi^\star = \arg\max_{\pi : \mathcal{S} \to \mathcal{A}} \quad \mathbb{E}_{\bar{\rho}, \pi} \left[\sum_{t=0}^{T} \gamma^t r\big(x_t, u_t\big) \right] \tag{1a}$$

$$\text{s.t.} \quad \mathbb{P}_{\bar{\rho}, \pi}(c(x_t, u_t) \leq 0) = 1, \quad \forall t \in [0, T]. \tag{1b}$$

Above, the notations $\mathbb{E}_{\bar{\rho}, \pi}$ in (1a) and $\mathbb{P}_{\bar{\rho}, \pi}$ in (1b) respectively denote expectations and probabilities taken over $x_0 \sim \bar{\rho}$ and $u_t \sim \pi$ for each $t \in [0, T]$, and, given $\mathbf{u} := (u_0, \cdots, u_T)$ drawn componentwise from π, the states $\mathbf{x} := (x_0, \cdots, x_T)$ are defined to satisfy $x_{t+1} = f(x_t, u_t)$ for each $t \in [0, T]$.

In contrast, the ICRL problem aims to recover *a priori* unknown constraints $c(\cdot)$ from demonstrations of reward-maximizing and constraint-satisfying state-control trajectories. Although current ICRL methods often assume that the demonstrations available for constraint learning are generated *online* via continuous environment exploration ("interaction"), such exploratory actions are often costly or unsafe in safety-critical settings (Malik et al., 2021; Liu et al., 2023; Kim et al., 2023; Hugessen et al., 2024). Offline ICRL methods include ICSDICE Quan et al. (2024), which leverages two fixed demonstration datasets $\mathcal{D}_E$ and $\mathcal{D}_\pi$ for constraint inference, but does not allow environmental exploration during training. Concretely, the *expert dataset* $\mathcal{D}_E := \{(x_t^{(d)}, u_t^{(d)}, x_{t+1}^{(d)}, r_t^{(d)}) : t \in [0, T-1], d \in [D_E]\}$ contains D_E state-control trajectories generated using a constraint-satisfying optimal expert policy π_E, and a dataset $\mathcal{D}_\pi := \{(x_t^{(d)}, u_t^{(d)}, x_{t+1}^{(d)}, r_t^{(d)}) : t \in [0, T-1], d \in [D_\pi]\}$ contains D_π state-control trajectories generated using possibly constraint-violating policies.

Although ICSDICE avoids online exploration during training, constructing $\mathcal{D}_\pi$ still requires behavior from policies that are not guaranteed to satisfy the unknown constraint.

3.2 Constrained Inverse Optimal Control (CIOC)

Let $\mathcal{S} \subseteq \mathbb{R}^n$, $\mathcal{A} \subseteq \mathbb{R}^m$, $f : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$, $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, and $c : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ denote a state space, control (action) space, system dynamics map, reward map, and constraint map, respectively. The horizon-T *constrained forward optimal control* (CFOC) problem, presented below, aims to compute optimal state-control trajectories $(\mathbf{x}, \mathbf{u}) := (x_0, \cdots, x_T, u_0, \cdots, u_T) \in \mathbb{R}^{(n+m)(T+1)}$ which minimize the cumulative cost $\sum_{t=0}^{T} -r(x_t, u_t)$ while satisfying the system dynamics $x_{t+1} = f(x_t, u_t)$ and constraints $c(x_t, u_t) \leq 0$ throughout $[0, T]$:

$$\min_{\mathbf{x}, \mathbf{u}} \quad \sum_{t=0}^{T} -r(x_t, u_t) \tag{2a}$$

$$\text{s.t.} \quad x_{t+1} = f(x_t, u_t),\ \forall\, t \in [0, T-1], \quad c(x_t, u_t) \leq 0,\ \forall\, t \in [0, T]. \tag{2b}$$

Meanwhile, in the *constrained inverse optimal control* (CIOC) problem, a constraint learner aims to recover *a priori* unknown constraints $c(\cdot, \cdot)$ from a set of expert trajectories $\mathcal{D}_E := \{(x_t^{(d)}, u_t^{(d)}) : t \in [0, T], d \in [D]\}$ which optimize (2), assuming the reward map $r(\cdot, \cdot)$ is known. (The dynamics map $f(\cdot, \cdot)$ may either be *a priori* known or unknown to the constraint learner.) The standard approach in CIOC for recovering the unknown constraint map $c(\cdot, \cdot)$ entails formulating the KKT equations corresponding to (2), as presented below in (4). Let $\boldsymbol{\lambda} := (\lambda_0, \cdots, \lambda_T)$ and $\boldsymbol{\nu} := (\nu_0, \cdots, \nu_{T-1})$ denote dual variables corresponding to the constraint and dynamics constraints in (2b), respectively, across times t, and define the Lagrangian $L : \mathcal{S}^{T+1} \times \mathcal{A}^{T+1} \times \mathbb{R}^{T+1} \times \mathbb{R}^{nT} \to \mathbb{R}$ by:

$$L(\mathbf{x}, \mathbf{u}, \boldsymbol{\lambda}, \boldsymbol{\nu}) := \textstyle\sum_{t=0}^{T} -r(x_t, u_t) + \sum_{t=0}^{T-1} \nu_t^\top (x_{t+1} - f(x_t, u_t)) + \sum_{t=0}^{T} \lambda_t^\top c(x_t, u_t). \tag{3}$$

The KKT equations for the CFOC problem (2) are thus:

$$c(x_t, u_t) \leq 0 \quad \perp \quad \lambda_t \geq 0, \qquad \forall t \in [0, T], \tag{4a}$$

$$\nabla_{(\mathbf{x},\mathbf{u})} L(\mathbf{x}, \mathbf{u}, \boldsymbol{\lambda}, \boldsymbol{\nu}) = 0. \tag{4b}$$

Above, $\perp$ indicates complementarity of the left- and right-hand-side conditions. In (4a), $c(x_t, u_t) < 0$ implies $\lambda_t = 0$, while $\lambda_t > 0$ implies $c(x_t, u_t) = 0$; equivalently, $\lambda_t c(x_t, u_t) = 0$. In settings in which the constraint map $c(\cdot, \cdot)$ is *a priori* known to be parameterized in a certain form, the CIOC problem reduces to the recovery of parameter values for which the resulting constraint map $c(\cdot, \cdot)$ satisfies the KKT system (4), when evaluated at expert state-control trajectories $(\mathbf{x}, \mathbf{u})$ in $\mathcal{D}^E$ (see Sec. 4.1). Such parameter recovery problems can often be recast as mixed-integer programs (MIP) that, for certain constraint maps characterized by low-dimensional parameters, are efficient to solve exactly Chou et al. (2020b).

Comparing CIOC and ICRL Methods for Constraint Recovery CIOC exploits demonstration optimality directly, but typically requires known closed-form dynamics and structured constraint parameterizations. ICRL, in contrast, accommodates unknown dynamics and flexible constraint representations, but obtains off-expert coverage from additional behavior data beyond the expert demonstrations. This trade-off motivates the hybrid setting considered in this work.

3.3 Problem Statement

We are given an expert dataset $\mathcal{D}_E$ of constraint-abiding optimal trajectories generated by solving a CFOC problem of the form (2). Moreover, suppose the reward map $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is known (we consider an unknown reward case in Appendix B.2) but the dynamics map $f : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$ and constraint map $c : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ are unknown. We aim to recover a neural network (NN) representation $\hat{c}(\cdot, \cdot; \theta) : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ for an *a priori* unknown constraint map $c : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, where $\theta \in \mathbb{R}^{M_\theta}$ describes NN weights to be trained using the dataset $\mathcal{D}_E$.

4 KKT-Informed Constraint Learning

We consider two constraint-recovery settings that share a learned local dynamics model. We first treat known parametric constraints with a CIOC-based mixed-integer program (MIP) in Sec. 4.1, then remove the known parameterization and introduce *Counterfactual KKT (CF-KKT)* for neural constraints in Sec. 4.2.

Learned local dynamics. Using expert transitions $(x_t, u_t, x_{t+1}) \in \mathcal{D}_E$, we fit $\hat{f}(\cdot, \cdot; \psi) : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$ by $\mathcal{L}_{\text{dyn}}(\psi) = \mathbb{E}_{\mathcal{D}_E}[\|\hat{f}(x_t, u_t; \psi) - x_{t+1}\|_2^2]$, and use $\hat{f}(\cdot, \cdot; \psi)$ for both inference procedures below.

4.1 Learning Parametric Constraints via MIP

When the unknown constraint has a known closed-form parameterization $\hat{c}(\cdot, \cdot; \theta)$, we recover the unknown parameter θ by applying CIOC to the learned dynamics model. We define the full Lagrangian and its stationarity residual by:

$$\hat{L}(\mathbf{x}, \mathbf{u}, \boldsymbol{\lambda}, \boldsymbol{\nu}; \theta, \psi) := \textstyle\sum_{t=0}^{T} -r(x_t, u_t) + \sum_{t=0}^{T} \lambda_t \hat{c}(x_t, u_t; \theta) + \sum_{t=0}^{T-1} \nu_t^\top (x_{t+1} - \hat{f}(x_t, u_t; \psi)), \tag{5}$$

$$\hat{\mathbf{s}}(\mathbf{x}, \mathbf{u}, \boldsymbol{\lambda}, \boldsymbol{\nu}; \theta, \psi) := \nabla_{(\mathbf{x},\mathbf{u})} \hat{L}(\mathbf{x}, \mathbf{u}, \boldsymbol{\lambda}, \boldsymbol{\nu}; \theta, \psi). \tag{6}$$

We seek constraint parameter values θ which are consistent with the KKT-optimality of the state-control trajectories $\{(\mathbf{x}^{(d)}, \mathbf{u}^{(d)}) : d \in [D_E]\}$ in the expert demonstration dataset $\mathcal{D}^E$. To this end, in (7), we pose a relaxed inverse KKT problem analogous to (Chou et al., 2020b, Prob. 2). Concretely, we aim to recover parameters θ and dual variables $\boldsymbol{\lambda}_{1:D} := (\boldsymbol{\lambda}^{(d)} : d \in [D_E])$ and $\boldsymbol{\nu}_{1:D} := (\boldsymbol{\nu}^{(d)} : d \in [D_E])$ that minimize the sum, over all $d \in [D_E]$, of the magnitudes of stationarity residuals $\hat{\mathbf{s}}(\mathbf{x}^{(d)}, \mathbf{u}^{(d)}, \boldsymbol{\lambda}^{(d)}, \boldsymbol{\nu}^{(d)}; \theta, \psi)$, while adhering to the primal/dual feasibility and complementary slackness constraints in (4a):

$$\min_{\theta, \boldsymbol{\lambda}_{1:D}, \boldsymbol{\nu}_{1:D}} \quad \sum_{d=1}^{D} \left\| \hat{\mathbf{s}}(\mathbf{x}^{(d)}, \mathbf{u}^{(d)}, \boldsymbol{\lambda}^{(d)}, \boldsymbol{\nu}^{(d)}; \theta, \psi) \right\|_2^2 \tag{7a}$$

$$\text{s.t.} \qquad \hat{c}(x_t^{(d)}, u_t^{(d)}; \theta) \leq 0 \quad \perp \quad \lambda_t^{(d)} \geq 0, \qquad \forall t \in [0, T], d \in [D_E]. \tag{7b}$$

In the CIOC-based constraint recovery literature (Chou et al., 2020b; Zhang et al., 2026), (7) is often solved using off-the-shelf solvers such as Gurobi Gurobi Optimization, LLC (2024), with the constraints (7b) encoded via big-M reformulations Bertsimas & Tsitsiklis (1998). We provide an analysis of MIP sensitivity to dynamics misspecification in App. E.

4.2 Counterfactual KKT (CF-KKT) Constraint Learning

When the constraint form is unknown, we represent it by a neural network $\hat{c}(\cdot, \cdot; \theta)$ and combine expert-side KKT supervision with synthetic counterfactual supervision.

Transition-wise auxiliary multipliers. The exact KKT stationarity condition in Sec. 3.2 is defined over an entire trajectory and couples neighboring time steps through the dynamics multipliers $\boldsymbol{\nu} = (\nu_0, \ldots, \nu_{T-1})$. Directly imposing this trajectory-level system is inconvenient for stochastic mini-batch training, where transition tuples $(x_t^{(d)}, u_t^{(d)}, x_{t+1}^{(d)}, r_t^{(d)})$ are sampled across time indices and demonstrations. We therefore construct a transition-wise KKT-informed objective using bounded auxiliary multipliers for the local inequality and dynamics terms, as defined below:

$$\tilde{\lambda}_t(x_t, u_t; \theta) = \lambda_{\max} \sigma\big(\hat{c}(x_t, u_t; \theta)\big), \qquad \tilde{\nu}_t = \nu_{\max} \tanh(\eta_t), \tag{8}$$

where $\lambda_{\max}, \nu_{\max} > 0$ are fixed bounds, $\eta_t \in \mathbb{R}^n$ is per-timestep, and $\tanh(\cdot)$ is applied elementwise. Above, (8) ensures $\tilde{\lambda}_t \geq 0$, consistent with dual feasibility, and calibrates $\|\tilde{\lambda}_t\|_2$ such that $\tilde{\lambda}_t$ becomes small for states predicted well inside the feasible region and larger as $\hat{c}(x_t, u_t; \theta)$ approaches or exceeds zero. The equality multiplier $\tilde{\nu}_t \in [-\nu_{\max}, \nu_{\max}]^n$ can take either sign within a bounded range. Bounding both multipliers improves numerical stability during training.

KKT-informed loss. For a single transition, define the local Lagrangian:

$$\hat{L}_t(x_t, u_t, x_{t+1}, \lambda, \nu; \theta, \psi) := -r(x_t, u_t) + \lambda_t \hat{c}(x_t, u_t; \theta) + \nu_t^\top \big[\hat{f}(x_t, u_t; \psi) - x_{t+1}\big]. \tag{9}$$

Given weights $w_p, w_c, w_a > 0$, we define

$$\mathcal{L}_{\text{KKT}}(\theta) = \mathbb{E}_{\mathcal{D}_E}\Big[\Big\|\nabla_{x_t, u_t} \hat{L}_t(x_t, u_t, x_{t+1}, \lambda, \nu; \theta, \psi)\big|_{\lambda=\tilde{\lambda}_t, \nu=\tilde{\nu}_t}\Big\|_2^2 + w_p \max\big\{\hat{c}(x_t, u_t; \theta), 0\big\}^2 + w_c\big(\tilde{\lambda}_t \hat{c}(x_t, u_t; \theta)\big)^2 + w_a\big|\hat{c}(x_t, u_t; \theta)\big|\Big]. \tag{10}$$

The first three terms respectively penalize a transition-wise stationarity residual, primal constraint violation, and violation of complementary slackness. The final activity regularizer discourages the learned constraint from assigning large negative values to all expert transitions.

Remark 1. *Transition tuples of the form $(x_t^{(d)}, u_t^{(d)}, x_{t+1}^{(d)})$ can be shuffled across time indices and trajectories to support stochastic mini-batch training. The full trajectory KKT conditions, however, include cross-time coupling through the dynamics multipliers ν_t. In our implementation, training with the full trajectory KKT residual produces Inf/NaN losses on* `BlockedAnt` *and* `HalfCheetah`*, while collapsing the cumulative reward on the other environments, which we attribute to the numerical instability of cross-time multiplier ν_t recursion. Motivated by this observation and the computational convenience of transition-wise training, we instead incorporate the per-time Lagrangian* (9) *into $\mathcal{L}_{KKT}(\theta)$.*

Synthetic Counterfactual Trajectories The KKT-informed loss in (10) is evaluated only on expert demonstrations, which are constraint-satisfying by construction. Consequently, it provides limited direct supervision information about infeasible behaviors away from the expert trajectories. To address this issue without collecting unsafe data, CF-KKT uses the learned dynamics model to generate synthetic counterfactual trajectories near the demonstrations.

We sample multiple windows of H consecutive expert transitions from demonstrations. We denote the window from trajectory d, starting at time index i as $\{(x_{i+h}^{(d)}, u_{i+h}^{(d)})\}_{h=0}^{H-1}$. For each window, we generate K perturbed control sequences, indexed by $k \in [K]$, by independently perturbing the corresponding expert control at each rollout step, while holding d fixed:

$$u_h^{(k)} = \mathrm{clip}\big(u_{i+h}^{(d)} + \sigma\, \epsilon_h^{(d,k)}, -1, 1\big), \qquad \epsilon_h^{(d,k)} \sim \mathcal{N}(0, I_m). \tag{11}$$

The perturbed and baseline rollouts start from the same demonstrated state, $x_0^{(k)} = \tilde{x}_0 = x_i^{(d)}$, and use the same learned dynamics. The counterfactual rollout follows $x_{h+1}^{(k)} = \hat{f}(x_h^{(k)}, u_h^{(k)}; \psi)$ for $h \in [0, H-1]$, while the baseline follows the corresponding expert action sequence, with states rolled out through the learned dynamics: $\tilde{u}_h = u_{i+h}^{(d)}$ and $\tilde{x}_{h+1} = \hat{f}(\tilde{x}_h, \tilde{u}_h; \psi)$.

A truncated H-step reward alone does not account for the cost-to-go of the terminal state x_H, since a perturbation may obtain larger short-horizon reward while steering the system towards a state with poor future cumulative reward. We therefore augment the rollout reward with a value function estimate $\hat{V}(\cdot)$, whose estimation from expert return-to-go is described in App. B:

$$\hat{J}_V^{(k)} = \textstyle\sum_{h=0}^{H-1} \gamma^h r\big(x_h^{(k)}, u_h^{(k)}\big) + \gamma^H \hat{V}\big(x_H^{(k)}\big), \tag{12}$$

$$\hat{J}_{V,\mathrm{base}} = \textstyle\sum_{h=0}^{H-1} \gamma^h r\left(\tilde{x}_h, \tilde{u}_h\right) + \gamma^H \hat{V}\left(\tilde{x}_H\right). \tag{13}$$

We then define the value-augmented counterfactual advantage as $\hat{A}_V^{(k)} := \hat{J}_V^{(k)} - \hat{J}_{V,\mathrm{base}}$. The terminal-value term $\hat{V}(\cdot)$ accounts for consequences that extend beyond the finite H-step rollout. Comparing the perturbed rollout against the expert-action trajectory from the same initial state then isolates the effect of the action perturbation, rather than differences due to the starting state.

We retain counterfactuals whose value-augmented return exceeds that of the expert-action baseline, i.e., $w_k = \mathbf{1}\big[\hat{A}_V^{(k)} > 0\big]$. Because both $\hat{f}$ and $\hat{V}$ are learned, w_k is a practical approximation to an ideal selection rule. To state the approximate rule, let $V_{i+H}^\star(x)$ denote the optimal feasible discounted cost-to-go from state x over the remainder of the expert horizon under the true dynamics, with $V_{i+H}^\star(x) = -\infty$ if no feasible continuation exists. Let $J_{V^\star}^{(k)}$ and $J_{V^\star,\mathrm{base}}$ denote the two rewards in (12) using true-dynamics rollouts and replacing $\hat{V}$ by $V_{i+H}^\star$.

Lemma 1 (Positive Counterfactual Advantage Implies Infeasibility)**.** *Let $(x_{0:T}^E, u_{0:T}^E)$ be a globally optimal feasible expert trajectory, and consider an H-step window beginning at expert time i. Denote the corresponding expert controls by $u_{i:i+H-1}^E := (u_i^E, \ldots, u_{i+H-1}^E)$. For any alternative control sequence $u_{0:H-1} := (u_0, \ldots, u_{H-1})$ initialized at $x_0 = x_i^E$, define*

$$J_{i,H}^V(u_{0:H-1}) = \textstyle\sum_{h=0}^{H-1} \gamma^h r(x_h, u_h) + \gamma^H V_{i+H}^\star(x_H), \tag{14}$$

where $x_{h+1} = f(x_h, u_h)$ follows the true dynamics and $V_{i+H}^\star(x_H)$ is the optimal constrained value achievable from x_H over the remaining time steps $i+H, \ldots, T$. If no feasible trajectory exists from x_H over these remaining steps, we define $V_{i+H}^\star(x_H) = -\infty$. Then

$$J_{i,H}^V(u_{0:H-1}) > J_{i,H}^V(u_{i:i+H-1}^E) \quad \Longrightarrow \quad \exists\, h \in \{0, \ldots, H-1\} :\ c(x_h, u_h) > 0. \tag{15}$$

Proof. Let $J_{E,i}$ denote the expert return from time i to T. Because the expert trajectory is globally optimal, the expert controls from time i onward are optimal when initialized at x_i^E. Hence,

$$J_{i,H}^V(u_{i:i+H-1}^E) = J_{E,i}.$$

Now suppose that $c(x_h, u_h) \le 0$ for every $h \in \{0, \ldots, H-1\}$. If no feasible trajectory exists from x_H over the remaining time steps, then $J_{i,H}^V(u_{0:H-1}) = -\infty$, so the strict inequality cannot hold. Otherwise, following $u_{0:H-1}$ for H steps and then taking the optimal feasible controls from x_H defines a feasible trajectory starting from x_i^E. By global optimality of the expert,

$$J_{i,H}^V(u_{0:H-1}) \le J_{E,i} = J_{i,H}^V(u_{i:i+H-1}^E),$$

which contradicts the assumed strict inequality. □

The assumptions of Lemma 1 are sufficient conditions rather than requirements we expect to hold exactly in practice. Accordingly, CF-KKT uses positive value-augmented advantage as a soft selection criterion, whose empirical validity is supported by the results in Appendix C.

Counterfactual Constraint Loss Let $\mathcal{B}^{(k)} = \{(x_h^{(k)}, u_h^{(k)})\}_{h=0}^{H-1}$ denote the k-th counterfactual set. For the selected sets, we train the average learned constraint value along the rollout to exceed a positive margin m:

$$\mathcal{L}_{\mathrm{CF}}(\theta) = \frac{\sum_{k=1}^{K} w_k \cdot \max\left\{m - \frac{1}{H}\sum_{h=0}^{H-1} \hat{c}\big(x_h^{(k)}, u_h^{(k)}; \theta\big), 0\right\}^2}{\sum_{k=1}^{K} w_k}. \tag{16}$$

The counterfactual trajectories are treated as fixed synthetic samples during the constraint update, so gradients of (16) propagate only through $\hat{c}(\cdot, \cdot; \theta)$. By Lemma 1, a counterfactual trajectory which generates a higher reward than a constrained optimal trajectory must violate constraint feasibility at one or more timesteps. We nevertheless use the mean constraint value along the rollout to provide denser information. This is motivated by the empirical observation that the rollout may violate feasibility at multiple timesteps. In particular, App. C shows high state-level precision among selected counterfactuals, indicating that a substantial fraction of states along a positive-advantage rollout are indeed infeasible. Averaging therefore provides denser and smoother loss gradient than a max-based or single-step objective that uses fewer steps to train. Finally, CF-KKT combines the original KKT-informed loss with the synthetic counterfactual loss:

$$\mathcal{L}_{\text{CF-KKT}}(\theta) = \alpha_{\text{KKT}}\mathcal{L}_{\text{KKT}}(\theta) + \alpha_{\text{CF}}\mathcal{L}_{\text{CF}}(\theta). \tag{17}$$

5 Experiments

We evaluate three increasingly unstructured settings: known parametric constraints under unknown dynamics, neural constraint recovery in continuous mazes, and high-dimensional MuJoCo tasks with unknown dynamics and constraint geometry. Our methods infer constraints from expert demonstrations, using learned dynamics to provide synthetic off-demonstration supervision when needed.

Baselines. We compare against **ICSDICE** (Quan et al., 2024), **SMODICE** (Ma et al., 2022), and **IQ-Learn** (Garg et al., 2021). ICSDICE is our primary offline ICRL baseline and explicitly learns a constraint from expert and suboptimal offline data. SMODICE is an offline imitation method based on distribution matching and does not natively recover a constraint mask $\mathbf{1}[c(x) \leq 0]$; for constraint-recovery and downstream-planning comparisons, we modify SMODICE to convert its learned state score into a feasibility mask. IQ-Learn likewise does not recover an explicit constraint; we therefore keep its standard imitation-learning formulation and evaluate the learned policy directly. In the known-parametric setting, we additionally compare against a parametric variant of ICSDICE using the same constraint family as the MIP method.

5.1 Known Parametric Constraints under Unknown Dynamics

We first consider the structured setting in which the constraint family is known up to a scalar parameter θ but the dynamics are unknown. We evaluate `HalfCheetah`, `BlockedAnt`, `Swimmer`, and `Quadcopter`, using one expert trajectory to fit $\hat{f}$ and recover θ via the MIP in Sec. 4.1. Environment, constraint, and demonstration details are given in App. A.1.

Comparison against Parametric ICSDICE. For `HalfCheetah` and `BlockedAnt`, we also train ICSDICE with the same scalar constraint family $c_\theta(x) = \theta - x_{\text{pos}}$, with pos being position on the x-axis. MIP method uses the expert trajectory, whereas ICSDICE additionally uses suboptimal trajectories from (Liu et al., 2023).

Table 1 shows that the MIP closely recovers the true value of θ on `Swimmer`, `Quadcopter`, and `HalfCheetah`. The larger `BlockedAnt` error (-2.05 versus -3.0) is consistent with larger KKT residuals in its demonstrations. Parametric ICSDICE yields -1.23 on `HalfCheetah` and -3.99 on `BlockedAnt`, showing that the same parametric family can produce substantially different estimates under an occupancy-based objective.

5.2 Unknown Constraint Geometry in Continuous Mazes

We next remove the assumption of known constraint parameterization while keeping the state dimension small enough to limit constraint complexity. These continuous maze environments let us test whether CF-KKT can recover qualitatively different constraint geometries, rather than only a scalar parameter within a known family.

Env.	θ^*	MIP $\hat{\theta}$	ICSDICE $\hat{\theta}$	MIP error
Swimmer	0.04	0.0484	–	$+0.0084$
Quadcopter	64	64	–	0
HalfCheetah	-3.0	-2.98	-1.23	$+0.02$
BlockedAnt	-3.0	-2.05	-3.99	$+0.95$

Table 1: Parametric recovery performance

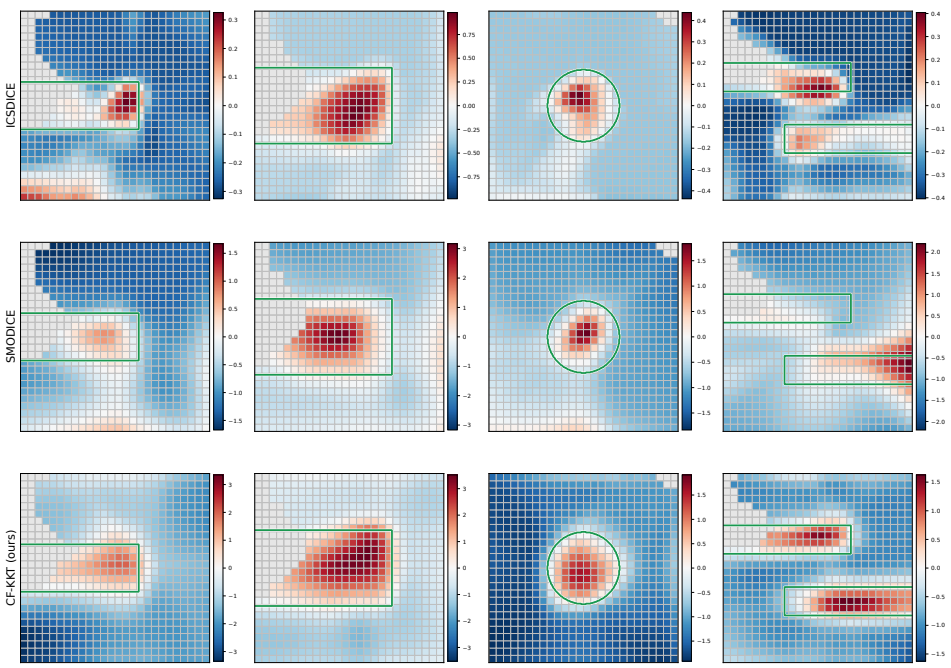


Figure 1: Ground-truth and learned constraint maps for 4 maze environments across ICSDICE (top), SMODICE (middle), and CF-KKT (bottom). Red, blue, and green denote predicted unsafe states, predicted safe states, and the ground-truth constraint boundary.

Environment and Demonstrations. We use four 2D navigation tasks (`Obstacle`, `Wide`, `Disk`, `Zigzag`) with double-integrator dynamics and unknown obstacle geometries. CF-KKT receives feasible expert demonstrations; baselines additionally receive a suboptimal dataset that may contain violations. For more dynamics, reward, geometry, and dataset-generation details, see App. A.2.

Constraint IoU. Let $\mathcal{U}^\star$ be the ground-truth unsafe region and $\hat{\mathcal{U}} = \{x : \hat{c}(x;\theta) > \tau\}$ the predicted unsafe set. We report $\mathrm{IoU} = |\hat{\mathcal{U}} \cap \mathcal{U}^\star| / |\hat{\mathcal{U}} \cup \mathcal{U}^\star|$ on a uniform 161×161 position grid with zero velocity. To account for method-dependent score scales, we use the common percentile threshold $\tau = Q_{0.85}(\{\hat{c}(x;\theta) : x \in \mathrm{grid}\})$. Table 2 and Fig. 1 show that CF-KKT recovers the constraint geometry more accurately on every environment. Its mean IoU is 0.707, compared with 0.425 for ICSDICE and 0.379 for SMODICE. The largest gains over ICSDICE occur on `Disk` (0.815 versus 0.428) and `Obstacle` (0.590 versus 0.276).

Downstream Planning. We convert each learned constraint to a 61×61 occupancy grid, plan to the goal with Dijkstra over predicted feasible cells, and track the waypoints with the expert PD controller. Across 20 feasible starts per seed, we report goal reaching, return (cumulative reward), and ground-truth violation rate $\mathrm{Violation} = T^{-1} \sum_{t=0}^{T-1} \mathbf{1}[c(x_t, u_t) > 0]$. IQ-Learn does not recover a mask, so we evaluate its imitation policy directly. Representative planning examples are shown in App. B.1.

Env.	SMODICE	ICSDICE	CF-KKT	Δ
Obstacle	0.275 ± 0.405	0.276 ± 0.193	$\mathbf{0.590 \pm 0.153}$	$+0.314$
Wide	0.547 ± 0.113	0.600 ± 0.140	$\mathbf{0.724 \pm 0.083}$	$+0.124$
Disk	0.363 ± 0.133	0.428 ± 0.218	$\mathbf{0.815 \pm 0.057}$	$+0.387$
Zigzag	0.330 ± 0.054	0.394 ± 0.172	$\mathbf{0.698 \pm 0.079}$	$+0.304$
Mean	0.379	0.425	$\mathbf{0.707}$	$+0.282$

Table 2: Maze constraint IoU (mean $\pm$ std over seeds); Δ is relative to ICSDICE.

Table 3 shows the return-violation table. IQ-Learn achieves a violation rate of 0.001 on `Wide` but reaches the goal from only $5/20$ starts. The oracle planner itself has nonzero violation on the rectangular environments because PD tracking can clip corners, which provides a planner-dependent floor. Relative to ICSDICE, CF-KKT lowers violation on all four environments while retaining comparable goal-reaching performance: $0.174 \to 0.042$ on `Obstacle`, $0.200 \to 0.085$ on `Wide`, $0.140 \to 0.080$ on `Disk`, and $0.209 \to 0.093$ on `Zigzag`.

5.3 High-Dimensional Neural Constraint Learning

Finally, we evaluate our method over the fully unstructured high-dimensional setting, where the transition dynamics are unknown and the constraint is represented by a neural network rather than a known parametric family.

Env.	Metric	Oracle	IQ-Learn	SMODICE	ICSDICE	CF-KKT
Obstacle	Goal	$20.0/20$	$10.0/20$	$18.3/20$	$18.7/20$	$20.0/20$
	Viol.	0.018	0.124 ± 0.203	0.175 ± 0.053	0.174 ± 0.038	$\mathbf{0.042 \pm 0.068}$
	Return	-58.2	-89.7 ± 5.8	-47.1 ± 7.4	-45.8 ± 1.6	-55.6 ± 6.8
Wide	Goal	$19.0/20$	$5.0/20$	$17.7/20$	$18.3/20$	$17.7/20$
	Viol.	0.032	0.001 ± 0.002	0.226 ± 0.111	0.200 ± 0.028	$\mathbf{0.085 \pm 0.121}$
	Return	-70.2	-106.5 ± 11.3	-58.2 ± 7.3	-63.4 ± 7.0	-68.5 ± 11.4
Disk	Goal	$20.0/20$	$14.3/20$	$19.3/20$	$19.3/20$	$20.0/20$
	Viol.	0.000	0.041 ± 0.039	0.181 ± 0.016	0.140 ± 0.060	$\mathbf{0.080 \pm 0.019}$
	Return	-48.5	-84.0 ± 2.4	-46.4 ± 0.6	-47.1 ± 0.8	-48.3 ± 0.5
Zigzag	Goal	$20.0/20$	$7.0/20$	$18.3/20$	$17.3/20$	$18.7/20$
	Viol.	0.060	0.086 ± 0.084	0.230 ± 0.033	0.209 ± 0.052	$\mathbf{0.093 \pm 0.044}$
	Return	-68.3	-92.3 ± 5.8	-45.6 ± 5.4	-56.4 ± 7.2	-62.4 ± 1.6

Table 3: Maze downstream planning over 20 starts."Viol." is the true violating-timestep fraction.

	ICSDICE		IQ-Learn		SMODICE		CF-KKT	
Env.	Ret.	Viol.	Ret.	Viol.	Ret.	Viol.	Ret.	Viol.
Ant_ls	1257 ± 123	87.5 ± 117.0	$65 \pm 155^{\dagger}$	6.3 ± 1.8	970 ± 186	86.8 ± 113.5	1130 ± 91	$\mathbf{2.8 \pm 0.9}$
BlockedAnt	2265 ± 400	82.3 ± 56.1	$258 \pm 316^{\dagger}$	145.3 ± 135.2	$1323 \pm 537^{\dagger}$	124.2 ± 84.7	2449 ± 237	$\mathbf{15.3 \pm 4.2}$
BlockedHopper	420 ± 112	110.6 ± 26.6	$269 \pm 274^{\dagger}$	134.1 ± 260.6	325 ± 275	82.9 ± 63.4	454 ± 252	$\mathbf{17.6 \pm 21.2}$
HalfCheetah	2657 ± 318	83.0 ± 58.9	$-270 \pm 447^{\dagger}$	0.0 ± 0.0	1915 ± 977	313.1 ± 349.0	2338 ± 788	$\mathbf{54.3 \pm 48.9}$
Walker_ls	682 ± 392	170.1 ± 106.9	361 ± 233	39.0 ± 16.8	140 ± 224	$\mathbf{37.6 \pm 56.5}$	410 ± 283	74.3 ± 56.3

Table 4: MuJoCo downstream return and violations ($N = 10$). Bold marks the lowest mean violation among explicit-mask methods; † denotes severe return collapse.

Datasets. We use the five continuous-control environments and offline datasets from ICSDICE Quan et al. (2024): `Ant_ls`, `BlockedAnt`, `BlockedHopper`, `HalfCheetahWithObstacle`, and `Walker_ls`. Each environment has a hidden ground-truth constraint used only for evaluation. The benchmark contains $50{,}000$ expert transitions in $\mathcal{D}_E$ and $200{,}000$ suboptimal transitions in $\mathcal{D}_\pi$. Under the data protocol above, the CF-KKT constraint learner receives $\mathcal{D}_E$, while SMODICE, IQ-Learn, and ICSDICE use more data, i.e., $\mathcal{D}_E \cup \mathcal{D}_\pi$.

Downstream Offline Policy Learning. Following the offline policy extraction procedure of Quan et al. (2024), we train the downstream value function and advantage-weighted policy, while filtering transitions according to the learned constraint. For the state-only constraints in these environments, we use the feasibility mask $M(x_{t+1}) = \mathbf{1}[\hat{c}(x_{t+1}; \theta) \leq 0]$. Transitions predicted to be unsafe are excluded from the value and advantage-weighted policy updates.

Evaluation Metrics. For downstream control, we report **constraint violations**, defined as the number of violating timesteps in a 1000-step evaluation episode, and **return**, the cumulative task reward. Results are averaged over 10 random seeds.

Downstream Safety and Return. Table 4 shows that CF-KKT reduces mean violations relative to ICSDICE on all five tasks, by 34.6%–96.8%. On `BlockedAnt` and `BlockedHopper`, it also improves mean return. The remaining tasks exhibit a safety–return trade-off; on `Walker_ls`, SMODICE obtains fewer violations but substantially lower return. Thus, the learned CF-KKT mask consistently improves safety relative to ICSDICE while preserving useful task performance.

6 Conclusion

We present a framework for learning unknown constraints from expert demonstrations under unknown dynamics. For known parametric constraints, we combine learned local dynamics with CIOC-based recovery and analyze sensitivity to model error. For neural constraints, CF-KKT combines KKT-inspired expert supervision with synthetic reward-improving counterfactuals to provide off-demonstration constraint information without violations. Experiments on robotic control tasks show improved constraint recovery and downstream safety over offline ICRL baselines. Limitations include environment-dependent loss weights, the assumption of a known differentiable reward, omission of cross-time coupling in the transition-wise KKT objective, and dependence of counterfactual quality on learned dynamics away from the demonstrations. Automating loss weighting and incorporating reward and dynamics uncertainty are promising directions for future work.

GENERATIVE AI USE DISCLOSURE

Generative AI tools were used in the preparation of this work to assist with refining mathematical statements and annotations, providing feedback on human-designed experimental methodology and ablation, and improving the organization, clarity, and presentation of the manuscript. They were also used for language editing, restructuring sections, and formatting equations and tables.

All AI-assisted mathematical arguments were reviewed and revised by the authors, and all reported experimental results were produced and verified by the authors. Generative AI was not treated as a source of scientific evidence or authority; technical claims, derivations, and conclusions were checked against the underlying analysis and experimental results. The authors take full responsibility for the final content of the paper, including all text, claims, and artifacts produced with the assistance of generative AI.

REFERENCES

Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. Constrained Policy Optimization. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 22–31, 06–11 Aug 2017.

E. Altman. *Constrained Markov Decision Processes*. Chapman and Hall, 1999.

Julian Berberich, Johannes Köhler, Matthias A. Müller, and Frank Allgöwer. Data-Driven Model Predictive Control With Stability and Robustness Guarantees. *IEEE Transactions on Automatic Control (TAC)*, 66(4):1702–1717, 2021.

Dimitris Bertsimas and John Tsitsiklis. *Introduction to Linear Optimization*. Athena Scientific, 1998.

V.S. Borkar. An Actor-critic Algorithm for Constrained Markov Decision Processes. *Systems & Control Letters*, 54(3):207–213, 2005.

Chih-Yuan Chiu, Zhouyu Zhang, and Glen Chou. Learning Constraints From Stochastic Partially-Observed Closed-Loop Demonstrations. *IEEE Control Systems Letters*, 10:43–48, 2026.

Glen Chou, Dmitry Berenson, and Necmiye Ozay. Learning Constraints from Demonstrations. *Workshop on the Algorithmic Foundations of Robotics (WAFR)*, 2018.

Glen Chou, Necmiye Ozay, and Dmitry Berenson. Learning Parametric Constraints in High Dimensions from Demonstrations. In *CoRL*, 2019.

Glen Chou, Necmiye Ozay, and Dmitry Berenson. Uncertainty-Aware Constraint Learning for Adaptive Safe Motion Planning from Demonstrations. In *CoRL 2020, Cambridge, MA, USA*, 2020a.

Glen Chou, Necmiye Ozay, and Dmitry Berenson. Learning Constraints From Locally-Optimal Demonstrations Under Cost Function Uncertainty. *IEEE Robotics and Automation Letters (RA-L)*, 5(2):3682–3690, 2020b.

Glen Chou, Hao Wang, and Dmitry Berenson. Gaussian Process Constraint Learning for Scalable Chance-Constrained Motion Planning From Demonstrations. *IEEE Robotics and Automation Letters*, 7(2):3827–3834, 2022. doi: 10.1109/LRA.2022.3148436.

Peter Englert, Ngo Anh Vien, and Marc Toussaint. Inverse KKT: Learning Cost Functions of Manipulation Tasks from Demonstrations. *The International Journal of Robotics Research (IJRR)*, 36(13-14):1474–1488, 2017.

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

Gurobi Optimization, LLC. Gurobi Optimizer Reference Manual, 2024. URL `https://www.gurobi.com`.

Taylor Howell, Nimrod Gileadi, Saran Tunyasuvunakool, Kevin Zakka, Tom Erez, and Yuval Tassa. Predictive Sampling: Real-time Behaviour Synthesis with MuJoCo. *arXiv preprint arXiv:2212.00541*, 2022.

Adriana Hugessen, Harley Wiltzer, and Glen Berseth. Simplifying constraint inference with inverse reinforcement learning. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 44501–44525. Curran Associates, Inc., 2024. doi: 10.52202/079017-1413. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/4eb5cc598a6271528ed9b84bb9879e9b-Paper-Conference.pdf.

Jaehwi Jang, Minjae Song, and Daehyung Park. Inverse Constraint Learning and Generalization by Transferable Reward Decomposition. *IEEE Robotics and Automation Letters*, 9(1):279–286, 2023.

Miles Johnson, Navid Aghasadeghi, and Timothy Bretl. Inverse Optimal Control for Deterministic Continuous-time Nonlinear Systems. In *CDC*, 2013.

Arezou Keshavarz, Yang Wang, and Stephen P. Boyd. Imputing a Convex Objective Function. In *ISIC*, pp. 613–619. IEEE, 2011.

Konwoo Kim, Gokul Swamy, Zuxin Liu, Ding Zhao, Sanjiban Choudhury, and Steven Z Wu. Learning Shared Safety Constraints from Multi-task Demonstrations. *Advances in Neural Information Processing Systems*, 36:5808–5826, 2023.

Guiliang Liu, Yudong Luo, Ashish Gaurav, Kasra Rezaee, and Pascal Poupart. Benchmarking Constraint Inference in Inverse Reinforcement Learning. In *The Eleventh International Conference on Learning Representations*, 2023.

Guiliang Liu, Sheng Xu, Shicheng Liu, Ashish Gaurav, Sriram Ganapathi Subramanian, and Pascal Poupart. A Comprehensive Survey on Inverse Constrained Reinforcement Learning: Definitions, Progress and Challenges. *Transactions on Machine Learning Research (TMLR)*, 2025. ISSN 2835-8856. Survey Certification.

Shicheng Liu and Minghui Zhu. Distributed Inverse Constrained Reinforcement Learning for Multi-agent Systems. *Advances in Neural Information Processing Systems*, 35:33444–33456, 2022.

Shicheng Liu and Minghui Zhu. Meta inverse constrained reinforcement learning: Convergence guarantee and generalization analysis. In *International Conference on Learning Representations*, volume 2024, pp. 12314–12351, 2024.

Yecheng Ma, Andrew Shen, Dinesh Jayaraman, and Osbert Bastani. Versatile offline imitation from observations and examples via regularized state-occupancy matching. In *International Conference on Machine Learning*, pp. 14639–14663. PMLR, 2022.

L. Magni and R. Scattolini. Model Predictive Control of Continuous-time Nonlinear Systems with Piecewise Constant Control. *IEEE Transactions on Automatic Control*, 49(6):900–906, 2004.

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

D.Q. Mayne, J.B. Rawlings, C.V. Rao, and P.O.M. Scokaert. Constrained Model Predictive Control: Stability and Optimality. *Automatica*, 36(6):789–814, 2000.

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.

Marcel Menner, Peter Worsnop, and Melanie N. Zeilinger. Constrained Inverse Optimal Control with Application to a Human Manipulation Task. *IEEE TCST*, 2019.

Minh Nguyen, Jingqi Li, Gechen Qu, and Claire J. Tomlin. Inverse safety filtering: Inferring constraints from safety filters for decentralized coordination, 2026. URL https://arxiv.org/abs/2604.02687.

Dimitris Papadimitriou, Usman Anwar, and Daniel S Brown. Bayesian Methods for Constraint Inference in Reinforcement Learning. *Transactions on Machine Learning Research (TMLR)*, 2022.

Daehyung Park, Michael Noseworthy, Rohan Paul, Subhro Roy, and Nicholas Roy. Inferring Task Goals and Constraints using Bayesian Nonparametric Inverse Reinforcement Learning. In *Conference on robot learning*, pp. 1005–1014. PMLR, 2020.

Claudia Pérez-D'Arpino and Julie A. Shah. C-LEARN: Learning Geometric Constraints from Demonstrations for Multi-Step Manipulation in Shared Autonomy. In *ICRA*, 2017.

Guorui Quan, Guiliang Liu, et al. Learning constraints from offline demonstrations via superior distribution correction estimation. In *Forty-first International Conference on Machine Learning*, 2024.

Rahel Rickenbach, Anna Scampicchio, and Melanie N. Zeilinger. Inverse Optimal Control as an Errors-in-Variables Problem. In Alessandro Abate, Mark Cannon, Kostas Margellos, and Antonis Papachristodoulou (eds.), *Proceedings of the 6th Annual Learning for Dynamics and Control Conference*, volume 242 of *Proceedings of Machine Learning Research*, pp. 375–386. PMLR, 15–17 Jul 2024.

Walter Rudin. *Principles of Mathematical Analysis, 3rd Edition*. McGraw-Hill, Inc., 1976.

Dexter RR Scobee and S Shankar Sastry. Maximum Likelihood Constraint inference for Inverse Reinforcement Learning. *arXiv preprint arXiv:1909.05477*, 2019.

Sriram Ganapathi Subramanian, Guiliang Liu, Mohammed Elmahgiubi, Kasra Rezaee, and Pascal Poupart. Confidence Aware Inverse Constrained Reinforcement Learning. *arXiv preprint arXiv:2406.16782*, 2024.

Markus Wulfmeier, Dushyant Rao, Dominic Zeng Wang, Peter Ondruska, and Ingmar Posner. Large-scale Cost Function Learning for Path Planning using Deep Inverse Reinforcement Learning. *The International Journal of Robotics Research (IJRR)*, 36:1073–1087, 2017.

Sheng Xu and Guiliang Liu. Uncertainty-aware constraint inference in inverse constrained reinforcement learning. In *International Conference on Learning Representations*, volume 2024, pp. 17792–17816, 2024.

Zhouyu Zhang, Chih-Yuan Chiu, and Glen Chou. Constraint Learning in Multi-Agent Dynamic Games From Demonstrations of Local Nash Interactions. *IEEE Robotics and Automation Letters*, 11(6):6696–6703, 2026.

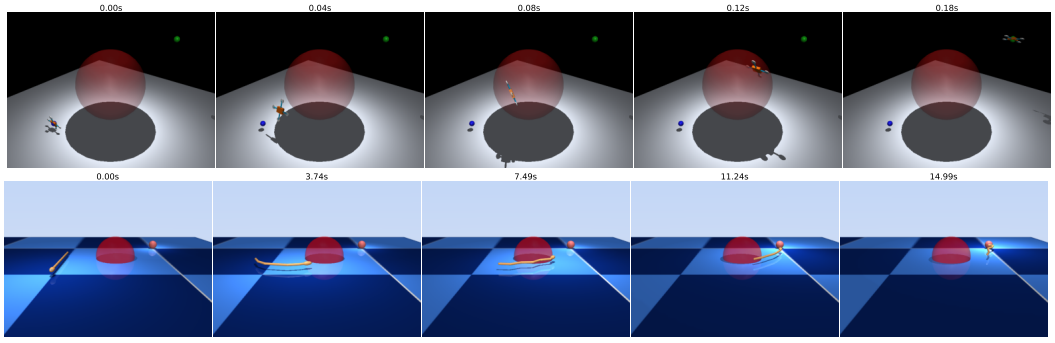


Figure 2: `Quadcopter` (top) and `Swimmer` (bottom) goal-reaching trajectories with the learned collision-avoidance constraint; red denotes the unsafe set.

A Benchmark Environments and Datasets

This appendix collects environment and dataset details omitted from the main experimental section.

A.1 Known-Parametric Experiments

For `HalfCheetah` and `BlockedAnt`, the unknown constraint is the position lower bound $c(x;\theta) = \theta - x_{\text{pos}} \leq 0$, with true threshold $\theta^* = -3.0$. Their trajectory data comes from the ICRL benchmark of Liu et al. (2023). For `Swimmer` and `Quadcopter`, the constraint encodes spherical obstacle avoidance, $c(x;\theta) = \theta - \|p(x) - p_{\text{obs}}\|_2^2 \leq 0$, where $p(x)$ is the position of the designated body (the swimmer nose or quadcopter center of mass) and p_{obs} is the known obstacle center. The true squared radii are $\theta^* = 0.04$ and 64, respectively. Their expert trajectories are generated offline with MuJoCo-MPC Howell et al. (2022). The parametric experiment uses one expert trajectory per environment to fit the local dynamics and recover θ.

A.2 Continuous-Maze Experiments

We use a double-integrator system on $\mathcal{S} = [-1,1]^2 \times \mathbb{R}^2$, with $x_t = (p_t, v_t) \in \mathbb{R}^4$ and $u_t \in [-1,1]^2$:

$$v_{t+1} = (1-\kappa)v_t + \Delta t\, \alpha u_t, \qquad \|v_{t+1}\|_2 \leq v_{\max}, \tag{18}$$

$$p_{t+1} = \Pi_{\mathcal{X}}(p_t + \Delta t\, v_{t+1}), \tag{19}$$

with $\Delta t = 0.1$, $\alpha = 1$, $\kappa = 0.15$, and $v_{\max} = 2$. The reward is $r(x_t, u_t) = -\|p_t - g\|_2 - 0.02\|u_t\|_2^2$ with $\gamma = 0.99$. We consider four unknown state-only unsafe regions—`Obstacle`, `Wide`, `Disk`, and `Zigzag`—covering wall-attached, compact, and non-convex geometries. Although the ground-truth constraint depends only on position, the learner is not given this structural knowledge.

Each environment and seed contains approximately 30,000 feasible expert transitions. Expert paths are generated by Dijkstra on a 61×61 occupancy grid and tracked by a PD controller. IQ-Learn, SMODICE, and ICSDICE additionally receive a 7000-transition suboptimal dataset that may contain constraint violations.

A.3 MuJoCo Neural-Constraint Experiments

Task specifications. We evaluate CF-KKT on five continuous-control tasks built from MuJoCo locomotion environments in Gymnasium (`-v4`), following the benchmark used by ICSDICE (Quan et al., 2024). Each task augments the nominal locomotion objective with a state constraint that is not explicitly penalized by the task reward.

Relationship between reward and constraint. The five tasks differ in how directly reward-improving behavior drives the constrained physical quantity. This distinction helps explain the counterfactual diagnostics in Appendix C and the supervision ablation in Appendix D.

Task	Base environment	$\dim(\mathcal{S})$	$\dim(\mathcal{A})$	Constraint: infeasible when
Ant_ls	Ant-v4	27	8	$v_x > 0.5$
BlockedAnt	Ant-v4	27	8	$\sum_{j\in\mathcal{J}} \lvert\omega_j\rvert > 1$
BlockedHopper	Hopper-v4	11	3	$\lvert\phi_{\text{foot}}\rvert > 0.3$
HalfCheetah	HalfCheetah-v4	18	6	$x_{\text{pos}} < -3$
Walker_ls	Walker2d-v4	17	6	$v_x > 1$

Table 5: MuJoCo benchmark specifications. Here v_x denotes forward velocity, ω_j the angular velocity of joint j with $\mathcal{J}$ denoting the four constrained joints, ϕ_{foot} the foot-joint angle, and x_{pos} the absolute horizontal torso position.

Task	Reward	Constrained quantity
Ant_ls	$v_x + 1 - 0.5\|u\|^2$	forward velocity (rewarded)
BlockedAnt	$v_x + 1 - 0.5\|u\|^2$	joint angular velocities
BlockedHopper	$v_x + 1 - 10^{-3}\|u\|^2$	foot-joint angle
HalfCheetah	$\lvert v_x\rvert - 0.1\|u\|^2$	horizontal position
Walker_ls	$v_x + 1 - 10^{-3}\|u\|^2$	forward velocity (rewarded)

Table 6: Task rewards and constrained physical quantities. The constant $+1$ is the survival bonus in the corresponding locomotion task. `HalfCheetah` rewards $|v_x|$, so motion in either horizontal direction contributes positively to the task reward.

Direct coupling. In `Ant_ls` and `Walker_ls`, the constraint directly bounds forward velocity, which also appears positively in the task reward. Reward improvement therefore tends to move the system toward the constraint boundary, making positive-value-advantage counterfactuals informative for constraint inference.

Indirect coupling. `BlockedAnt` constrains joint angular velocities while rewarding forward body velocity. These quantities are correlated through the locomotion dynamics, but the constrained quantity does not appear directly in the reward.

Delayed angular coupling. `BlockedHopper` constrains a joint angle, which evolves over multiple dynamics steps rather than changing instantaneously with the control. As a result, violations can emerge progressively rather than at the first counterfactual step.

Delayed positional coupling. `HalfCheetah` constrains absolute horizontal position, whereas action perturbations affect position indirectly through the evolution of velocity. Multi-step rollouts are therefore important for allowing perturbations to propagate to the constrained quantity. Appendix C quantifies these coupling effects using counterfactual violation diagnostics.

Datasets. Each task provides approximately 5×10^4 expert transitions in $\mathcal{D}_E$, organized into 50 trajectories, together with 2×10^5 transitions in the benchmark suboptimal dataset $\mathcal{D}_\pi$. `Hopper` and `Walker2d` can terminate early when the agent falls, so their expert trajectories need not have equal length (e.g., `BlockedHopper` contains $49{,}945$ transitions across 50 episodes); the remaining tasks use fixed 1000-step episodes. Ground-truth constraint labels are reserved for evaluation and oracle diagnostics.

B EXPERIMENT DETAILS

Experiment Setup. All networks are two-layer MLPs with 256 hidden units and ReLU activations, implemented in JAX/Flax. The constraint network c_θ, dynamics model f_ψ, continuation-value model $\hat{V}$, and downstream policy π_ϕ use this architecture. All networks are trained with Adam at a learning rate of 3×10^{-4}. We use a mini-batch size of 256 and discount factor $\gamma = 0.99$. For the KKT terms, the default weights are $w_p = 10.0$, $w_c = 0.01$, and $w_a = 0.1$, with outer scale $\alpha_{\mathrm{KKT}} = 1$; proxy dual bounds are $\lambda_{\max} = \nu_{\max} = 100$.

The continuation-value model is trained from the expert trajectories by regressing to the discounted return-to-go. For an expert state $x_i^{(d)}$, we use the target

$$G_i^{(d)} = \sum_{j=i}^{T} \gamma^{j-i} r\left(x_j^{(d)}, u_j^{(d)}\right), \tag{20}$$

and minimize

$$\mathcal{L}_V = \mathbb{E}_{\mathcal{D}_E}\left[\left(\hat{V}(x_i^{(d)}) - G_i^{(d)}\right)^2\right]. \tag{21}$$

The resulting $\hat{V}$ is used only to score the terminal states of counterfactual and expert-action rollouts in the value-augmented advantage.

All results are averaged over the same set of $N = 10$ random seeds. Experiments are conducted on a cloud-based Linux compute node with 32 CPU cores (Intel Xeon Gold 6426Y, 2.5 GHz), 64 GB system memory, and one NVIDIA L40S GPU with 46 GB VRAM. One seed with 20,000 total training steps takes approximately 5–10 minutes. Counterfactual generation uses $B = 256$ sampled expert windows, $K = 64$ perturbed action sequences, perturbation scale $\sigma = 2$ (for MuJoCo), $\sigma = 0.8$ (for 2D maze), and rollout horizon $H = 24$. The counterfactual margin is $m = 2$ in the 2D maze experiments and $m = 1$ in the MuJoCo experiments, with $\alpha_{\mathrm{CF}} = 1$.

In MuJoCo environments, the downstream policy extraction involves using suboptimal data for Advantage Weighted Regression. We hereby retain this evaluation protocol consistent with ICSDICE and SMODICE because they explicitly *require* a suboptimal dataset to function as a constraint learner. We clarify that the additional suboptimal data never enters the constraint learning phase for CF-KKT, and is used solely for evaluation.

B.1 MAZE PLANNING VISUALIZATIONS

Figure 3 shows representative downstream plans produced from the learned constraint masks.

B.2 ROBUSTNESS TO LEARNED REWARD MODELS

Our main method assumes access to the analytical task reward, as is standard in inverse optimal control, because KKT stationarity depends explicitly on $\nabla_{x,u} r(x, u)$. We evaluate how sensitive CF-KKT is to relaxing this assumption by replacing r with a neural approximation $\hat{r}(\cdot, \cdot; \omega)$.

A reward model trained only to match scalar reward values need not recover the correct local reward geometry: small prediction error in $\hat{r}$ does not imply that $\nabla_{x,u}\hat{r}$ matches $\nabla_{x,u} r$, even though the latter directly enters the KKT stationarity residual. We therefore compare two reward models. **MLP** minimizes only reward-value prediction error, while **MLP + reg.** uses the same architecture and additionally penalizes reward-gradient mismatch:

$$\mathcal{L}_r(\omega) = \mathbb{E}\Big[\big(\hat{r}(x_t, u_t; \omega) - r(x_t, u_t)\big)^2 + w_{\nabla r} \left\|\nabla_{x_t,u_t}\hat{r}(x_t, u_t; \omega) - \nabla_{x_t,u_t} r(x_t, u_t)\right\|_2^2\Big]. \tag{22}$$

Here $w_{\nabla r} = 0$ recovers the plain MLP, whereas $w_{\nabla r} > 0$ explicitly regularizes the learned reward gradient toward the analytical one. In this ablation, $\hat{r}$ replaces r in the KKT stationarity term and in the counterfactual return calculation.

Table 7 shows that gradient regularization reduces constraint violations on all five environments, with especially large reductions on `HalfCheetahWithObstacle` ($133.5 \rightarrow 51.1$) and

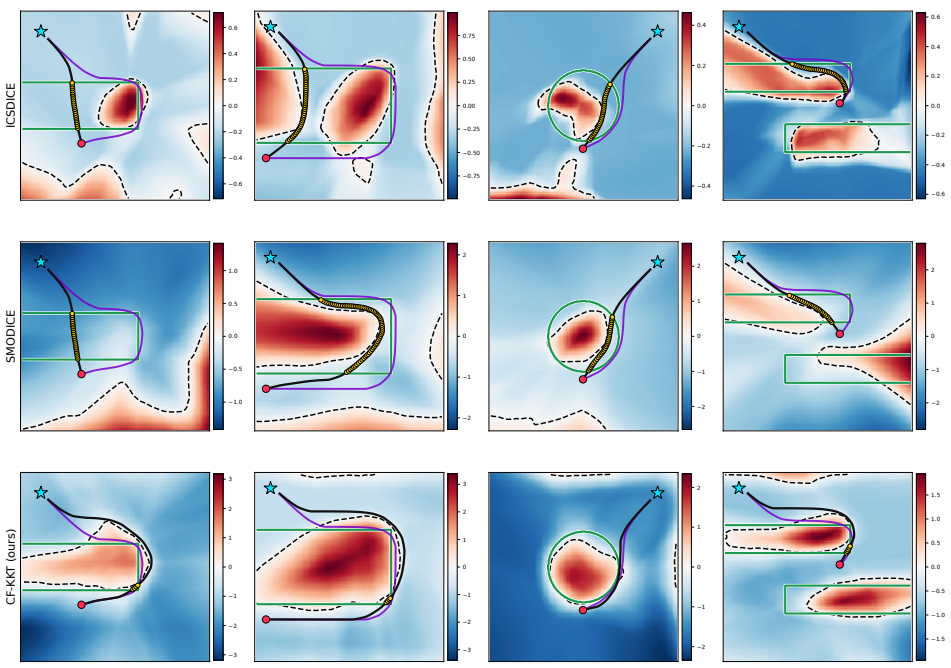


Figure 3: Representative maze plans. The dashed contour is the predicted unsafe boundary, green the true unsafe region, purple the oracle route, and yellow the executed timesteps inside the true unsafe region. The goal is indicated by blue star.

Table 7: Learned-reward robustness (mean $\pm$ std, $N = 10$); MLP + reg. also matches analytical reward gradients.

	Violations ↓		Return ↑	
Environment	MLP	MLP + reg.	MLP	MLP + reg.
Ant_ls	7.1 ± 13.8	2.3 ± 0.5	1177 ± 98	1177 ± 68
BlockedAnt	89.4 ± 40.2	67.0 ± 19.4	2142 ± 621	2165 ± 552
BlockedHopper	65.1 ± 26.9	37.3 ± 30.3	342 ± 239	277 ± 102
HalfCheetah (obst.)	133.5 ± 32.7	51.1 ± 33.9	3126 ± 407	2938 ± 408
Walker_ls	138.0 ± 76.4	100.2 ± 76.5	676 ± 420	447 ± 309

`BlockedHopper` ($65.1 \rightarrow 37.3$). These results indicate that CF-KKT can remain effective when the analytical reward is replaced by a learned approximation, provided that the local reward-gradient structure used by the KKT conditions is also preserved.

B.3 Additional Benchmarking Against Online ICRL Methods.

Our main experiments focus on offline constraint inference. As an auxiliary comparison, we also evaluate the *KKT-informed component* of our framework within the online ICRL benchmark of Liu et al. (2023). This experiment does not use the counterfactual component of CF-KKT and should therefore be interpreted as a diagnostic of the KKT-based constraint signal rather than an evaluation of the full method.

We compare this KKT-only constraint learner with BC2L, MECL, and GACL in the `BlockedHalfCheetah` environment. The online baselines receive additional interaction data from the environment, whereas the KKT-only constraint learner relies on the initial expert demonstrations. Figure 4 reports downstream policy performance when each method is used as the upstream constraint-learning component. The KKT-only variant achieves lower constraint violation and higher feasible reward than BC2L and MECL, while remaining comparable to GACL, and exhibits lower variance than BC2L and MECL.

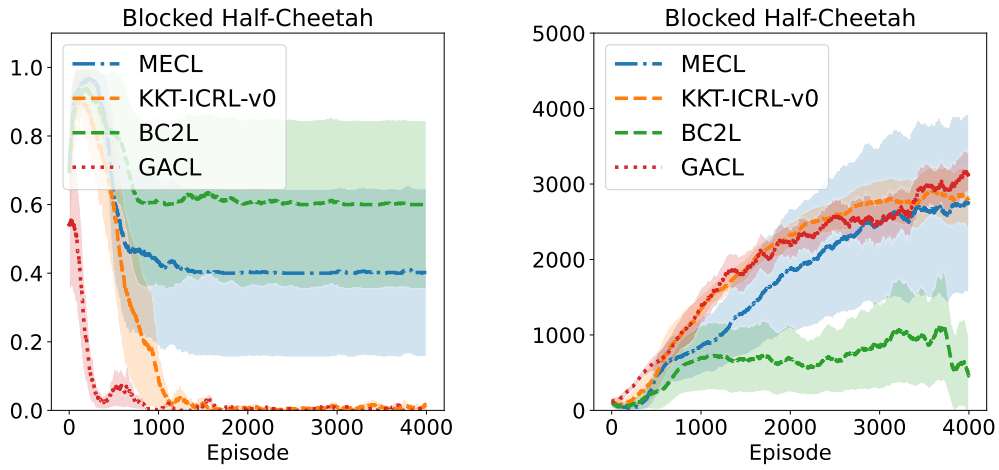


Figure 4: Constraint violation rate (left) and feasible cumulative reward (right) for the auxiliary online ICRL comparison.

	$H = 12$		$H = 48$		CF-KKT ($H = 24$)	
Env.	Viol. ↓	Ret. ↑	Viol. ↓	Ret. ↑	Viol. ↓	Ret. ↑
Ant_ls	3.4±1.5	1157±57	3.3±1.6	**1167±84**	**2.8±0.9**	1130±91
BlockedAnt	16.5±3.3	2136±635	18.5±4.9	1968±733	**15.3±4.2**	**2449±237**
BlockedHopper	64.9±83.5	317±139	28.6±32.7	**510±433**	**17.6±21.2**	454±252
HalfCheetah	93.9±67.5	2196±583	80.2±78.7	**2605±621**	**54.3±48.9**	2338±788
Walker_ls	85.4±43.1	**413±213**	89.6±65.6	406±279	**74.3±56.3**	410±283

Table 8: Effect of counterfactual rollout horizon with the positive-advantage gate fixed. CF-KKT uses the default $H = 24$.

B.4 Method Sweep on Counterfactual Generation

We separately study the two design choices used in counterfactual generation: the rollout horizon H and the counterfactual acceptance rule. First, we vary $H \in \{12, 24, 48\}$ while keeping the positive-advantage gate $w_k = \mathbf{1}[\hat{A}_V^{(k)} > 0]$ fixed, where $H = 24$ corresponds to the default CF-KKT setting. Second, fixing $H = 24$, we compare this gate against a quantile-based rule that retains the top 25% of counterfactuals ranked by $\hat{A}_V^{(k)}$.

The horizon sweep in Table 8 shows that CF-KKT is not highly sensitive to the precise rollout length: $H = 12$, 24, and 48 all yield comparable downstream performance, while the default $H = 24$ achieves the lowest mean violation count on all five environments. The acceptance-rule comparison in Table 9 shows CF-KKT's positive advantage based gating achieves lower violation than the quantile-based rule.

C Empirical Validation of the Counterfactual Premise

CF-KKT retains a generated counterfactual when its learned-model value-augmented advantage is positive, i.e., $\hat{A}_V^{(k)} > 0$. Lemma 1 establishes that, under global expert optimality, exact dynamics, and the exact constrained continuation value, a positive value-augmented advantage implies that the counterfactual cannot remain feasible throughout the rollout. In practice, CF-KKT evaluates this criterion using the learned dynamics $\hat{f}$ and continuation-value estimate $\hat{V}$. Here we empirically assess how reliably the resulting practical selection rule identifies true constraint violations. Ground-truth constraint labels are used exclusively for this diagnostic and are never used to train CF-KKT.

	$q = 0.25$		CF-KKT	
Env.	Viol. ↓	Ret. ↑	Viol. ↓	Ret. ↑
Ant_ls	3.3 ± 1.7	1117 ± 102	$\mathbf{2.8 \pm 0.9}$	$\mathbf{1130 \pm 91}$
BlockedAnt	21.8 ± 14.4	2096 ± 604	$\mathbf{15.3 \pm 4.2}$	$\mathbf{2449 \pm 237}$
BlockedHopper	27.3 ± 32.3	$\mathbf{624 \pm 261}$	$\mathbf{17.6 \pm 21.2}$	454 ± 252
HalfCheetah	84.6 ± 58.2	$\mathbf{2735 \pm 494}$	$\mathbf{54.3 \pm 48.9}$	2338 ± 788
Walker_ls	97.3 ± 92.3	$\mathbf{422 \pm 393}$	$\mathbf{74.3 \pm 56.3}$	410 ± 283

Table 9: Effect of the counterfactual acceptance rule at $H = 24$. The quantile rule retains the top 25% of candidates by $\hat{A}_V^{(k)}$, whereas CF-KKT retains all candidates with $\hat{A}_V^{(k)} > 0$.

Metrics. For each selected counterfactual rollout $\{x_h^{(k)}\}_{h=1}^H$, we evaluate the benchmark ground-truth state constraint $c^\star$ and report:

- **Bag precision:** $\Pr[\exists h : c^\star(x_h^{(k)}) > 0 \mid \hat{A}_V^{(k)} > 0]$, the fraction of selected rollouts containing at least one truly infeasible state. This is the metric most directly aligned with the trajectory-level implication in Lemma 1.
- **State precision:** $\Pr[c^\star(x_h^{(k)}) > 0 \mid \hat{A}_V^{(k)} > 0]$, evaluated over all states in selected rollouts. Unlike bag precision, this metric measures the density of infeasible states within a selected rollout; the lemma does not require every state to be infeasible.
- **Enrichment:** the state-level violation rate among selected counterfactual rollouts divided by the violation rate in the benchmark suboptimal dataset $\mathcal{D}_\pi$. This metric measures whether the generator preferentially samples infeasible behavior relative to the benchmark background distribution. Here $\mathcal{D}_\pi$ serves only as the reference distribution for this metric.
- **Per-step violation rate:** the violation frequency at the first and final rollout states, $h = 1$ and $h = H$, indicating how infeasibility develops over the counterfactual horizon.

Environment	**Bag prec.**	**State prec.**	**Enrich.**	h=1	h=H
Ant_ls	1.000	0.848	3.75	0.87	0.81
BlockedAnt	1.000	1.000	2.72	1.00	1.00
BlockedHopper	0.922	0.749	1.37	0.08	0.72
Walker_ls	1.000	0.969	1.31	1.00	0.95
HalfCheetah	0.953	0.867	1.78	0.00	0.83

Table 10: Ground-truth diagnostics of selected CF-KKT counterfactuals over $N = 10$ seeds. Metrics are defined in the text.

Positive value-augmented advantage consistently identifies infeasible rollouts. Bag precision is at least 0.922 and enrichment exceeds 1 on all five environments. Thus, the practical learned-model selection rule reliably identifies trajectory-level infeasibility and concentrates selected rollouts in regions with more violations than the benchmark suboptimal distribution.

Constraint violations may emerge only after several model steps. This effect is most apparent on `BlockedHopper` and `HalfCheetah`. In `BlockedHopper`, the constraint bounds a joint angle, so action perturbations influence the constrained quantity progressively through the dynamics; the violation rate increases from 0.08 at $h = 1$ to 0.72 at $h = H$. In `HalfCheetah`, the constraint depends on absolute horizontal position, which responds indirectly through the evolution of velocity. Consequently, no selected counterfactual violates the constraint at $h = 1$, whereas the violation rate reaches 0.83 by $h = H$. These results support the use of multi-step rollouts for constraints whose effects are dynamically delayed.

State-level density supports the mean-based surrogate. State precision ranges from 0.749 to 1.000 across the five tasks, showing that violations are typically not isolated to a single state within a selected rollout. This empirical density supports the mean constraint value in the counterfactual loss as a dense, smooth surrogate, even though the theoretical guarantee requires only the existence of a violation somewhere along the trajectory.

D Ablation of Supervision Signals

D.1 Experimental Setting

CF-KKT combines expert-side KKT supervision with synthetic counterfactual supervision. Because these MuJoCo benchmarks use state-only constraints, we write $\hat{c}_\theta(x)$ for brevity. To isolate the contribution of each term, we decompose the expert-side objective into

$$\mathcal{L}_{\text{stat}} = \mathbb{E}_{\mathcal{D}_E}\left[\left\|\nabla_\xi \hat{L}(\xi)\right\|_2^2\right], \tag{23}$$

$$\mathcal{L}_{\text{prim}} = \mathbb{E}_{\mathcal{D}_E}\left[\max\{\hat{c}_\theta, 0\}^2\right], \tag{24}$$

$$\mathcal{L}_{\text{comp}} = \mathbb{E}_{\mathcal{D}_E}\left[(\lambda \hat{c}_\theta)^2\right], \tag{25}$$

$$\mathcal{L}_{\text{act}} = \mathbb{E}_{\mathcal{D}_E}[|\hat{c}_\theta|]. \tag{26}$$

The complete KKT-informed loss is

$$\mathcal{L}_{\text{KKT}} = \mathcal{L}_{\text{stat}} + w_p \mathcal{L}_{\text{prim}} + w_c \mathcal{L}_{\text{comp}} + w_a \mathcal{L}_{\text{act}}, \tag{27}$$

with $w_p = 10$, $w_c = 0.01$, and $w_a = 0.1$. We further define a generic classifier that does not use KKT optimality:

$$\mathcal{L}_{\text{safe}} = \mathbb{E}_{\mathcal{D}_E}\left[w_p \max\{\hat{c}_\theta, 0\}^2 + w_a|\hat{c}_\theta|\right], \tag{28}$$

and denote the counterfactual loss by

$$\mathcal{L}_{\text{CF}} = \mathbb{E}_b\left[\max\left\{m - \frac{1}{H}\sum_h \hat{c}_\theta(x_{b,h}),\, 0\right\}^2\right], \qquad m = 1. \tag{29}$$

We compare the following seven controlled conditions:

- A**:** $\mathcal{L}_{\text{safe}} + \mathcal{L}_{\text{CF}}$. This retains expert-side feasibility/activity supervision and counterfactual supervision, but removes the optimality residual from Lagrangians.
- B**:** $A + w_c\mathcal{L}_{\text{comp}}$. This isolates the additional contribution of complementary slackness.
- $A^{-\text{act}}$**:** A with $w_a = 0$. This isolates the effect of the activity term.
- $A^{-\text{cf}}$**:** $\mathcal{L}_{\text{safe}}$ only. This removes counterfactual supervision.
- **CF-only:** $\mathcal{L}_{\text{CF}}$ only. This removes all supervision acting directly on expert demonstrations.
- **KKT-only:** $\mathcal{L}_{\text{KKT}}$ only, obtained by setting $\alpha_{\text{CF}} = 0$. This condition therefore isolates what the optimality conditions alone identify, and serves as the counterpart of CF-only.
- **CF-KKT:** the complete objective $\alpha_{\text{KKT}}\mathcal{L}_{\text{KKT}} + \alpha_{\text{CF}}\mathcal{L}_{\text{CF}}$.

All experimental settings other than the indicated objective terms are held fixed across conditions.

D.2 Metrics

We evaluate the learned constraint directly using two complementary metrics. First, the mask admission rate

$$\rho_\theta = \mathbb{E}_{\mathcal{D}_\pi}[\mathbf{1}[\hat{c}_\theta(x') \le 0]] \tag{30}$$

is the fraction of transitions in the benchmark suboptimal dataset admitted by the learned feasibility mask. We compare this quantity with $\rho^\star$, the fraction admitted by the ground-truth constraint. Values substantially below $\rho^\star$ indicate an overly restrictive mask, while values substantially above $\rho^\star$ indicate an overly permissive one.

Second, we report ROC AUC of $\hat{c}_\theta$ against ground-truth constraint labels on held-out suboptimal transitions. AUC is threshold-free and therefore measures whether the learned score ranks feasible and infeasible states correctly independently of the operating threshold. Ground-truth labels are used only for evaluation.

Environment	$\rho^\star$	A	B	$A^{-\mathrm{act}}$	$A^{-\mathrm{cf}}$	CF-only	KKT-only	CF-KKT
Ant_ls	0.774	0.023	0.014	0.095	0.426	0.000	0.992	0.681
BlockedAnt	0.632	0.151	0.149	0.281	0.480	0.000	0.890	0.528
BlockedHopper	0.453	0.088	0.080	0.445	0.471	0.000	0.586	0.241
HalfCheetah	0.512	0.009	0.003	0.064	0.431	0.000	0.394	0.360
Walker_ls	0.259	0.043	0.051	0.208	0.460	0.000	0.692	0.203

Table 11: Mask admission rates across the seven ablation conditions.

Environment	A	B	$A^{-\mathrm{act}}$	$A^{-\mathrm{cf}}$	CF-only	KKT-only	CF-KKT
Ant_ls	0.768	0.746	0.804	0.570	**0.907**	0.458	0.831
BlockedAnt	0.444	0.438	0.446	0.449	**0.904**	0.522	0.616
BlockedHopper	0.614	0.617	0.574	0.510	**0.802**	0.481	0.602
HalfCheetah	0.635	0.619	0.636	0.494	0.655	0.976	**0.993**
Walker_ls	0.779	**0.787**	0.783	0.524	0.624	0.382	0.705

Table 12: ROC AUC of the learned constraint score on held-out transitions.

D.3 Results

D.4 Analysis

Across the five environments, CF-only and full CF-KKT attain the two highest mean AUC values, 0.778 and 0.749, respectively, whereas KKT-only attains 0.564. We therefore compare CF-only and CF-KKT downstream in Table 13, while using the KKT-only condition to isolate what can be identified from expert-side optimality supervision alone.

Counterfactual supervision provides off-demonstration discrimination, but does not calibrate the feasibility threshold by itself. CF-only achieves the highest mean AUC, indicating that counterfactuals provide a strong signal for ranking feasible against infeasible states. However, $\rho_\theta = 0$ on all five environments, so the fixed feasibility threshold rejects every transition. The comparison between A and $A^{-\mathrm{cf}}$ shows the same effect: adding counterfactual supervision improves AUC on four of five tasks, but also makes the learned mask substantially more restrictive, ρ_θ of A is close to 0 on 4/5 envs, meaning that it rejects almost every transition not from the expert. Thus, counterfactual supervision provides useful off-demonstration separation, but by itself does not determine where the feasibility boundary should lie.

KKT supervision provides complementary expert-side information, but alone does not reliably identify the unsafe region. KKT-only exhibits nearly the opposite failure mode from CF-only. Its admission rate exceeds $\rho^\star$ on four of five environments, indicating that expert-side optimality supervision alone generally produces an overly permissive mask. Its AUC is also near chance on `Ant_ls`, `BlockedAnt`, `BlockedHopper`, and `Walker_ls`, showing that supervision evaluated only on expert transitions provides limited information about how unseen states should be ranked. `HalfCheetah` is a notable exception, where KKT-only already achieves AUC 0.976, suggesting that the local optimality structure is particularly informative in that environment.

The two supervision sources correct complementary failure modes. Combining KKT and counterfactual supervision substantially improves calibration: the mean absolute deviation $\frac{1}{5}\sum_e |\rho_{\theta,e} - \rho^\star_e|$ is 0.526 for CF-only and 0.232 for KKT-only, but decreases to 0.123 for full CF-KKT. The finer expert-side ablations support the same interpretation. Removing the activity term, $A^{-\mathrm{act}}$, increases ρ_θ in every environment while changing AUC only modestly, indicating that this term primarily affects threshold placement. Conditions A and B are nearly identical, suggesting that complementary slackness contributes little in the absence of the stationarity residual. Full CF-KKT combines the off-demonstration discrimination supplied by counterfactuals with the expert-side structure sup-

	CF-only		CF-KKT	
Environment	**Viol. ↓**	**Return ↑**	**Viol. ↓**	**Return ↑**
Ant_ls	16.6 ± 10.4	$246 \pm 176^{\dagger}$	$\mathbf{2.8 \pm 0.9}$	$\mathbf{1130 \pm 91}$
BlockedAnt	163.4 ± 137.9	$276 \pm 182^{\dagger}$	$\mathbf{15.3 \pm 4.2}$	$\mathbf{2449 \pm 237}$
BlockedHopper	$2.1 \pm 6.1^{\dagger}$	$112 \pm 298^{\dagger}$	17.6 ± 21.2	$\mathbf{454 \pm 252}$
HalfCheetah	$0.0 \pm 0.0^{\dagger}$	$-3 \pm 3^{\dagger}$	54.3 ± 48.9	$\mathbf{2338 \pm 788}$
Walker_ls	$0.0 \pm 0.0^{\dagger}$	$-6 \pm 9^{\dagger}$	74.3 ± 56.3	$\mathbf{410 \pm 283}$

Table 13: Downstream comparison of CF-only and CF-KKT. Mean $\pm$ standard deviation over $N = 10$ seeds; † denotes severe return collapse.

plied by the KKT-informed objective, yielding both strong ranking performance and a substantially better-calibrated feasibility mask.

Downstream control requires both discrimination and a useful operating threshold. Table 13 illustrates why AUC alone is insufficient. Despite its high mean AUC, CF-only suffers severe return collapse on every task; its low violation counts on `BlockedHopper`, `HalfCheetah`, and `Walker_ls` therefore arise largely because the learned mask prevents meaningful task progress. CF-KKT instead attains substantially higher return on all five environments and improves both return and violation count on `Ant_ls` and `BlockedAnt`. Together, these results support the intended division of roles: counterfactual supervision identifies off-demonstration infeasibility, while KKT-informed expert supervision helps place that discrimination at a practically useful feasibility threshold.

E Proofs

E.1 Sensitivity Analysis for MIP-based Inference of Constraint Parameters

Here, we analyze the sensitivity of the MIP-based constraint inference method in Sec. 4.1 against dynamics misspecification under a set of mild regularity conditions. Let $\boldsymbol{\eta}_{1:D} := (\eta_t^{(d)} : t \in [T], d \in [D]) \in \mathbb{R}^{nTD_E}$, $\boldsymbol{\rho}_{1:D} := (\rho_t^{(d)} : t \in [T], d \in [D]) \in \mathbb{R}^{nTD_E}$, and $\boldsymbol{\kappa}_{1:D} := (\kappa_t^{(d)} : t \in [T], d \in [D]) \in \mathbb{R}^{nTD_E}$ denote dual variables for (7) that respectively correspond to the constraints $\hat{c}(x_t^{(d)}, u_t^{(d)}; \theta) \le 0$, $\lambda_t^{(d)} \ge 0$, and $\lambda_t^{(d)} \hat{c}(x_t^{(d)}, u_t^{(d)}; \theta) = 0$, across $d \in [D_E]$ and $t \in [T]$. Then (7) admits a Lagrangian map given by:

$$
\begin{aligned}
&L_{\text{inv,rel}}(\theta, \boldsymbol{\lambda}_{1:D}, \boldsymbol{\nu}_{1:D}, \boldsymbol{\eta}_{1:D}, \boldsymbol{\rho}_{1:D}, \boldsymbol{\kappa}_{1:D}) \\
:= &\sum_{d=1}^{D} \Big[\big\| \hat{\mathbf{s}}(\mathbf{x}^{(d)}, \mathbf{u}^{(d)}, \boldsymbol{\lambda}^{(d)}, \boldsymbol{\nu}^{(d)}; \theta, \psi) \big\|_2^2 + \sum_{t=0}^{T} \eta_t^{(d)} \hat{c}(x_t^{(d)}, u_t^{(d)}; \theta) \\
&\qquad - \sum_{t=0}^{T} \rho_t^{(d)} \lambda_t^{(d)} + \sum_{t=0}^{T} \kappa_t^{(d)} \lambda_t^{(d)} \hat{c}(x_t^{(d)}, u_t^{(d)}; \theta) \Big],
\end{aligned} \tag{31}
$$

and KKT conditions given by:

$$
\hat{c}(x_t^{(d)}, u_t^{(d)}; \theta) \le 0 \quad \perp \quad \lambda_t^{(d)} \ge 0, \quad \forall t \in [T],\ \forall d \in [D_E], \tag{32a}
$$

$$
\hat{c}(x_t^{(d)}, u_t^{(d)}; \theta) \le 0 \quad \perp \quad \eta_t^{(d)} \ge 0, \quad \forall t \in [T],\ \forall d \in [D_E], \tag{32b}
$$

$$
-\lambda_t^{(d)} \le 0 \quad \perp \quad \rho_t^{(d)} \ge 0, \quad \forall t \in [T],\ \forall d \in [D_E], \tag{32c}
$$

$$
\nabla_{(\theta, \boldsymbol{\lambda}_{1:D}, \boldsymbol{\nu}_{1:D})} L_{\text{inv,rel}}(\theta, \boldsymbol{\lambda}_{1:D}, \boldsymbol{\nu}_{1:D}, \boldsymbol{\eta}_{1:D}, \boldsymbol{\rho}_{1:D}, \boldsymbol{\kappa}_{1:D}) = \mathbf{0}. \tag{32d}
$$

We say (32a) holds with *strict complementarity* if for each $t \in [0, T]$, one of the two inequalities $c(x_t, u_t; \theta) \le 0$ and $\lambda_t \ge 0$ holds with strict inequality while the other holds with equality, and likewise for (32b)-(32c). The following proposition uses the KKT system (32) and the Implicit Function Theorem to analyze the local sensitivity of the recovered constraint parameter θ to the neural network weights ψ which characterize the learned dynamics map $\hat{f}(\cdot, \cdot; \psi)$. For any weights ψ, we will use $\boldsymbol{z}(\psi) := (\theta, \boldsymbol{\lambda}_{1:D}, \boldsymbol{\nu}_{1:D}, \boldsymbol{\eta}_{1:D}, \boldsymbol{\rho}_{1:D}, \boldsymbol{\kappa}_{1:D})(\psi)$ to denote the concatenation of feasible primal and dual variables for (7) at ψ.

Proposition 1. *Suppose $\hat{f}(\cdot;\cdot;\psi)$ is C^4 for each $\psi \in \mathbb{R}^{M_\psi}$. Suppose also that, for a given NN weight $\hat{\psi} \in \mathbb{R}^{M_\psi}$, the relaxed inverse problem* (7) *admits a primal-dual solution satisfying strict complementarity. Define the map $F(\cdot)$ such that the equality $F(\boldsymbol{z}, \psi) = 0$, at $\psi = \hat{\psi}$, represents the KKT system* (32) *when all active inequalities in* (32a)-(32c) *are treated as equalities, and all inactive inequalities are ignored. Suppose $F(\cdot)$ is C^2 in a neighborhood of $\hat{\psi}$, and $\nabla_{\boldsymbol{z}} F(\boldsymbol{z}^\star(\hat{\psi}), \hat{\psi})$ is non-singular. Then, there exists an open neighborhood U of $\hat{\psi}$ and an open neighborhood V of $\boldsymbol{z}^\star(\hat{\psi})$ such that, for each $\psi \in U$, there exists a unique $\boldsymbol{z}^\star(\psi) \in V$ satisfying $F(\boldsymbol{z}^\star(\psi), \psi) = \mathbf{0}$. Moreover, for each $\psi \in U$:*

$$\|\boldsymbol{z}^\star(\psi) - \boldsymbol{z}^\star(\hat{\psi})\|_2 \leq \|\nabla_{\boldsymbol{z}} F(\boldsymbol{z}^\star(\hat{\psi}), \hat{\psi})^{-1} \nabla_\psi F(\boldsymbol{z}^\star(\hat{\psi}), \hat{\psi})\|_2 \cdot \|\psi - \hat{\psi}\|_2 + \mathcal{O}(\|\psi - \hat{\psi}\|_2^2). \quad (33)$$

Proof. By the Implicit Function Theorem (Rudin, 1976, Thm. 9.28), there exist neighborhoods U of $\hat{\psi}$ and V of $\boldsymbol{z}^\star(\hat{\psi})$ such that, for each $\psi \in U$, $\boldsymbol{z} = \boldsymbol{z}^\star(\psi)$ is the unique solution in V to $F(\boldsymbol{z}, \psi) = 0$, and $\boldsymbol{z}^\star(\cdot)$ is C^2 over U, with $\nabla_\psi \boldsymbol{z}^\star(\psi) = -\nabla_{\boldsymbol{z}} F(\boldsymbol{z}^\star(\psi), \psi)^{-1} \nabla_\psi F(\boldsymbol{z}^\star(\psi), \psi)$ for each $\psi \in U$. By continuity and strict complementarity, U and V can be chosen sufficiently small such that the strict inequalities at $(\boldsymbol{z}^\star(\hat{\psi}), \hat{\psi})$ remain strict, so $F(\boldsymbol{z}^\star(\psi), \psi) = 0$ continues to represent the KKT system (32) for each $\psi \in U$. The bound (33) then follows by applying the Lagrange remainder theorem, followed by the triangle and Cauchy-Schwarz inequalities. □

Remark 2. *Strict complementarity does not require all inequality constraints to be active, but merely excludes the degenerate case in which the primal and dual inequalities evaluated at an optimal primal-dual pair are simultaneously tight. The role of strict complementarity in Prop. 1 is to identify an unambiguous smooth equality representation of each complementarity condition at the nominal solution $\hat{\psi}$. Under strict complementarity, by continuity, the strict inequalities in the KKT system remain strict under sufficiently small perturbations of ψ, so the same equality representation remains valid, and the classical Implicit Function Theorem applies. Thus, Prop. 1 does not require the active set to remain fixed a priori or under arbitrary dynamics perturbations. More general sensitivity results that admit active-set transitions can be obtained by representing the KKT conditions as a generalized equation (e.g., formulated via an inclusion operator) under appropriate regularity conditions, although the resulting solution map need not be differentiable at such transitions.*

Under the conditions in Prop. 1, given any weights $\psi \in U$ for the learned dynamics map $\hat{f}(\cdot,\cdot;\psi)$, the optimal constraint parameter recovered by solving (7) is the set of the first M_θ components of $\boldsymbol{z}^\star(\psi)$. Thus, the bound (33) quantifies the robustness of the recovered constraint parameter against fluctuations in the learned, dynamics-encoding weights ψ.