Risk-sensitive Inverse Reinforcement Learning
via Semi- and Non-Parametric Methods

Sumeet Singh\textsuperscript{1}, Jonathan Lacotte\textsuperscript{2}, Anirudha Majumdar\textsuperscript{3}, and Marco Pavone\textsuperscript{1}

\textsuperscript{1}Department of Aeronautics and Astronautics, Stanford University \({}^\ast\)
\textsuperscript{2}Department of Electrical Engineering, Stanford University \({}^\dagger\)
\textsuperscript{3}Department of Mechanical and Aerospace Engineering, Princeton University \({}^\ddagger\)

\begin{abstract}

The literature on Inverse Reinforcement Learning (IRL) typically assumes that humans take actions in order to minimize the expected value of a cost function, i.e., that humans are \textit{risk neutral}. Yet, in practice, humans are often far from being risk neutral. To fill this gap, the objective of this paper is to devise a framework for \textit{risk-sensitive} IRL in order to explicitly account for a human’s risk sensitivity. To this end, we propose a flexible class of models based on \textit{coherent risk measures}, which allow us to capture an entire spectrum of risk preferences from risk-neutral to worst-case. We propose efficient non-parametric algorithms based on linear programming and semi-parametric algorithms based on maximum likelihood for inferring a human’s underlying risk measure and cost function for a rich class of static and dynamic decision-making settings. The resulting approach is demonstrated on a simulated driving game with ten human participants. Our method is able to infer and mimic a wide range of qualitatively different driving styles from highly risk-averse to risk-neutral in a data-efficient manner. Moreover, comparisons of the Risk-Sensitive (RS) IRL approach with a risk-neutral model show that the RS-IRL framework more accurately captures observed participant behavior both qualitatively and quantitatively, especially in scenarios where catastrophic outcomes such as collisions can occur.

\end{abstract}

\section{Introduction}

Imagine a world where robots and humans coexist and work seamlessly together. In order to realize this vision, robots must be able to (1) accurately predict the actions of

\begin{thebibliography}{99}

\item\textsuperscript{\ast} \{ssingh19, pavone\}@stanford.edu
\item\textsuperscript{\dagger} lacotte@stanford.edu
\item\textsuperscript{\ddagger} ani.majumdar@princeton.edu

\end{thebibliography}
humans in their environment, (2) quickly learn the preferences of human agents in their proximity and act accordingly, and (3) learn how to accomplish new tasks from human demonstrations. Inverse Reinforcement Learning (IRL) (Russell, 1998; Ng and Russell, 2000; Abbeel and Ng, 2005; Levine and Koltun, 2012; Ramachandran and Amir, 2007; Ziebart et al., 2008; Englert and Toussaint, 2015) is a powerful and flexible framework for tackling these challenges and has been previously used for a wide range of tasks, including modeling and mimicking human driver behavior (Abbeel and Ng, 2004; Kuderer et al., 2015; Sadigh et al., 2016a), pedestrian trajectory prediction (Ziebart et al., 2009; Mombaur et al., 2010), and legged robot locomotion (Zucker et al., 2010; Kolter et al., 2007; Park and Levine, 2013). More recently, the popular technique of Max-Entropy (MaxEnt) IRL, an inspiration for some of the techniques leveraged in this work, has been adopted in a deep learning framework (Wulfmeier et al., 2015), and embedded within the guided policy optimization algorithm (Finn et al., 2016). The underlying assumption behind IRL is that humans act optimally with respect to an (unknown) cost function. The goal of IRL is then to infer this cost function from observed actions of the human. By learning the human’s underlying preferences (in contrast to, e.g., directly learning a policy for a given task), IRL allows one to generalize one’s predictions to novel scenarios and environments.

The prevalent modeling assumption made by existing IRL techniques is that humans take actions in order to minimize the expected value of a random cost. Such a model, referred to as the expected value (EV) model, implies that humans are risk neutral with respect to the random cost; yet, humans are often far from being risk neutral. A generalization of the EV model is represented by the expected utility (EU) theory in economics (von Neumann and Morgenstern, 1944), whereby one assumes that a human is an optimizer of the expected value of a disutility function of a random cost. Despite the historical prominence of EU theory in modeling human behavior, a large body of literature from the theory of human decision making strongly suggests that humans behave in a manner that is inconsistent with the EU model. At a high level, the EU model has two main limitations: (1) experimental evidence consistently confirms that this model is lacking in its ability to describe human behavior in risky scenarios (Allais, 1953; Ellsberg, 1961; Kahneman and Tversky, 1979), and (2) the EU model assumes that humans make no distinction between scenarios in which the probabilities of outcomes are known and ones in which they are unknown, which is often not the case. Consequently, a robot interacting with a human in a safety-critical setting (e.g., autonomous driving or navigation using shared autonomy), while leveraging such an inference model, could make incorrect assumptions about the human agent’s behavior, potentially leading to catastrophic outcomes.

The known and unknown probability scenarios are referred to as risky and ambiguous respectively in the decision theory literature. An elegant illustration of the role of ambiguity is provided by the Ellsberg paradox (Ellsberg, 1961). Imagine an urn (Urn 1) containing 50 red and 50 black balls. Urn 2 also contains 100 red and black balls, but the relative composition of colors is unknown. Suppose that there is a payoff of $10 if a red ball is drawn (and no payoff for black). In human experiments, subjects display an overwhelming
preference towards having a ball drawn from Urn 1. However, now suppose the subject is told that a black ball has $10 payoff (and no payoff for red). Humans still prefer to draw from Urn 1. This is a paradox, since choosing to draw from Urn 1 in the first case (payoff for red) indicates that the human assesses the proportion of red in Urn 1 to be higher than in Urn 2, while choosing Urn 1 in the second case (payoff for black) indicates that the human assesses a lower proportion of red in Urn 1 than in Urn 2. Indeed, there is no utility function for the two outcomes that can resolve such a contradictory assessment of underlying probabilities since it stems from a subjective distortion of outcome probabilities rather than rewards.

The limitations of EU theory in modeling human behavior has prompted substantial work on various alternative theories such as rank-dependent expected utility (Quigggin, 1982), expected uncertain utility (Gul and Pesendorfer, 2014), dual theory of choice (distortion risk measures) (Yaari, 1987), prospect theory (Kahneman and Tversky, 1979; Barberis, 2013), and many more (see Majumdar and Pavone, 2017) for a recent review of the various axiomatic underpinnings of these risk measures). Further, one way to interpret the Ellsberg paradox is that humans are not only risk averse, but are also ambiguity averse – an observation that has sparked an alternative set of literature in decision theory on “ambiguity-averse” modeling; see, e.g., the recent review (Gilboa and Marinacci, 2016). It is clear that the assumptions made by EU theory thus represent significant restrictions from a modeling perspective in an IRL context since a human expert is likely to be both risk and ambiguity averse, especially in safety critical applications such as driving where outcomes are inherently ambiguous and can possibly incur very high cost.

The key insight of this paper is to address these challenges by modeling humans as evaluating costs according to an (unknown) risk measure. A risk measure is a function that maps an uncertain cost to a real number (the expected value is thus a particular risk measure and corresponds to risk neutrality). In particular, we will consider the class of coherent risk measures (CRMs) (Artzner et al., 1999; Shapiro, 2009; Ruszczyński, 2010). CRMs were proposed within the operations research community and have played an influential role within the modern theory of risk in finance (Rockafellar and Uryasev, 2000; Acerbi and Tasche, 2002; Acerbi, 2002; Rockafellar, 2007). This theory has also recently been adopted for risk-sensitive (RS) Model Predictive Control and decision making (Chow and Pavone, 2014; Chow et al., 2015), and guiding autonomous robot exploration for maximizing information gain in time-varying environments (Axelrod et al., 2016).

Coherent risk measures enjoy a number of advantages over EV and EU theories in the context of IRL. First, they capture an entire spectrum of risk assessments from risk-neutral to worst-case and thus offer a significant degree of modeling flexibility. Second, they capture risk sensitivity in an axiomatically justified manner; specifically, they formally capture a number of intuitive properties that one would expect any risk measure to satisfy (see Section 2.2). Third, a representation theorem for CRMs (Section 2.2) implies that they can be interpreted as computing the expected value of a cost function in a worst-case sense over a set of probability distributions (referred to as the risk envelope). Thus, CRMs
capture both risk and ambiguity aversion within the same modeling framework since the risk envelope can be interpreted as capturing uncertainty about the underlying probability distribution that generates outcomes in the world. Finally, they are tractable from a computational perspective; the representation theorem allows us to solve both the inverse and forward problems in a computationally tractable manner for a rich class of static and dynamic decision-making settings.

Statement of contributions: This paper presents an IRL algorithm that explicitly takes into account risk sensitivity under general axiomatically-justified risk models that jointly capture risk and ambiguity within the same modeling framework. To this end, this paper makes four primary contributions. First, we propose a flexible modeling framework for capturing risk sensitivity in humans by assuming that the human demonstrator (hereby referred to as the “expert”) acts according to a CRM. This framework allows us to capture an entire spectrum of risk assessments from risk-neutral to worst-case. Second, we develop efficient algorithms based on Linear Programming (LP) for inferring an expert’s underlying risk measure for a broad range of static (Section 3) decision-making settings, including a proof of convergence of the predictive capability of the algorithm in the case where we only attempt to learn the risk measure. We additionally consider cases where both the cost and risk measure of the expert are unknown. Third, we develop a maximum likelihood based model for inferring the expert’s risk measure and cost function for a rich class of dynamic decision-making settings (Section 4), generalizing our work in (Majumdar et al., 2017). Fourth, we demonstrate our approach on a simulated driving game (visualized in Figure 1) using a state-of-the-art commercial driving simulator and present results on ten human participants (Section 5). We show that our approach is able to infer and mimic qualitatively different driving styles ranging from highly risk-averse to risk-neutral using only a minute of training data from each participant. We also compare the predictions made by our risk-sensitive IRL (RS-IRL) approach with one that models the expert using expected value theory and demonstrate that the RS-IRL framework more accurately captures observed participant behavior both qualitatively and quantitatively, especially in scenarios involving significant risk to the participant-driven car.

Related Work: Safety-critical control and decision making applications demand increased resilience to events of low probability and detrimental consequences (e.g., a UAV crashing due to unexpectedly large wind gusts or an autonomous car failing to accommodate for an erratic neighboring vehicle). Such problems have inspired the recent advancement of various restricted versions of the problems considered here. In particular, there is a large body of work on RS decision making. For instance, in (Howard and Matheson, 1972) the authors leverage the exponential (or entropic) risk. This has historically been a very popular technique for parameterizing risk-attitudes in decision theory but suffers from the usual drawbacks of the EU framework such as the calibration theorem (Rabin, 2000). The latter states that very little risk aversion over moderate costs leads to unrealistically high degrees of risk aversion over large costs, which is undesirable from a modeling perspective. Other RS Markov Decision Process (MDP) formulations include Markowitz-inspired
mean-variance (Filar et al., 1989; Tamar et al., 2012), percentile criteria on objectives (Wu and Yuanlie, 1999) and constraints (Geibel and Wysotzki, 2005), and cumulative prospect theory (Prashanth et al., 2016). This has driven research in the design of learning-based solution algorithms, i.e., RS reinforcement learning (Mihatsch and Neuneier, 2002; Bäuerle and Ott, 2011; Tamar et al., 2012; Petrik and Subramanian, 2012; Shen et al., 2014; Tamar et al., 2016). Ambiguity in MDPs is also well studied via the robust MDP framework, see e.g., (Nilim and El Ghaoui, 2005; Xu and Mannor, 2010), as well as (Osogami, 2012; Chow et al., 2015) where the duality between risk and ambiguity as a result of CRMs is exploited. The key difference between this literature and the present work is that we consider the inverse reinforcement learning problem.

Results in the RS-IRL setting are more limited and have largely been pursued in the neuroeconomics literature (Glimcher and Fehr, 2014). For example, (Hsu et al., 2005) performed Functional Magnetic Resonance Imaging (FMRI) studies of humans making decisions in risky and ambiguous settings and modeled risk and ambiguity aversion using parametric utility and weighted probability models. In a similar vein, (Shen et al., 2014) models risk aversion using utility based shortfalls (with utility functions fixed a priori) and presents FMRI studies on humans performing a sequential investment task. While this literature may be interpreted in the context of IRL, the models used to predict risk and ambiguity aversion are quite limited. Risk in (Sadigh et al., 2016b) is captured via a single parameter to represent the aggressiveness of the expert driver – a fairly limited

(a) Visualization of simulator during the interactive game experiment as seen by participant. (b) Logitech G29 game input hardware consists of a force-feedback steering wheel and accelerator and brake pedals.

Figure 1: The simulated driving game considered in this paper. The human controls the follower car using a force-feedback steering wheel and two pedals and must follow the leader (an “erratic driver”) as closely as possible without colliding. We observed a wide range of behaviors from participants reflecting varying attitudes towards risk.
model that additionally does not account for probabilistic uncertainty. More recently, the authors in (Ratliff and Mazumdar, 2017) leverage the shortfall-risk model and associated $Q$–value decomposition introduced in (Shen et al., 2014) to devise a gradient based RS-IRL algorithm. The model again assumes an a priori known risk measure and parameterized utility function and the learning loss function is taken to be the likelihood of the observed actions assuming the Boltzmann distribution fit to the optimal $Q$–values. There are two key limitations of this approach. First, learning is performed assuming a known utility functional and risk measure – both of which, in general, are difficult to fix a priori for a given application. Second, computing gradients involves taking expectations with respect to the optimal policy as determined by the current value of the parameters and thus must be determined by solving the fixed-point equations defining the “forward” RL problem – a computationally demanding task for large or infinite domains. This limitation is not an artifact of RS-IRL but in fact a standard complexity issue in any MaxEnt IRL-based algorithm. In contrast, this work (1) harnesses the elegant dual representation results for CRMs to avoid having to assume a known risk functional, and (2) solves a significantly less complex forward problem by leveraging a receding-horizon planning model for the expert – a technique used to great effect also in (Sadigh et al., 2016a).

A first version of this work was presented in (Majumdar et al., 2017). In this revised and extended edition, we include the following additional contributions: (1) a significant improvement in the multi-step RS-IRL model which now accounts for an expert planning over sequential disturbance modes (as opposed to the single-branch model in (Majumdar et al., 2017)); (2) a formal proof of convergence guaranteeing that in the limit, the single-step RS-IRL model will exactly replicate the expert’s behavior; (3) introduction of a new maximum likelihood based approach for inferring both the risk measure and cost function for the multi-step model without assuming any a priori functional form; (4) extensive experimental validation on a realistic driving simulator where we demonstrate a significant improvement in predictive performance enabled by the RS-IRL algorithm over the standard risk-neutral model.

2 Problem Formulation

2.1 Dynamics

Consider the following discrete-time dynamical system:

$$x_{k+1} = f(x_k, u_k, w_k),$$

(1)

where $k$ is the time index, $x_k \in \mathbb{R}^n$ is the state, $u_k \in \mathbb{R}^m$ is the control input, and $w_k \in \mathcal{W}$ is the disturbance. The control input is assumed to be bounded component-wise: $u_k \in \mathcal{U} := \{u : u^- \leq u \leq u^+\}$. We take $\mathcal{W}$ to be a finite set $\{w[1], \ldots, w[L]\}$ with probability mass function (pmf) $p := [p(1), p(2), \ldots, p(L)]$, where $\sum_{i=1}^{L} p(i) = 1$ and $p(i) > 0, \forall i \in \{1, \ldots, L\}$. The time-sampling of the disturbance $w_k$ will be discussed
in Section 4. We assume that we are given demonstrations from an expert in the form of sequences of state-control pairs \( \{(x_k^*, u_k^*)\}_k \) and that the expert has knowledge of the underlying dynamics (1) but does not necessarily have access to the disturbance pmf \( p \).

2.2 Model of the Expert

We model the expert as a risk-sensitive decision-making agent acting according to a coherent risk measure (defined formally below). We refer to such a model as a coherent risk model.

We assume that the expert has a cost function \( C(x_k, u_k) \) that captures his/her preferences about outcomes. Let \( Z \) denote the cumulative cost accrued by the agent over a horizon \( N \):

\[
Z := \sum_{k=0}^{N} C(x_k, u_k).
\]

Note that since the process \( \{x_k\} \) is stochastic, \( Z \) is a random variable adapted to the sequence \( \{x_k\} \). A risk measure is a function \( \rho(Z) \) that maps this uncertain cost to a real number. We will assume that the expert is assessing risks according to a coherent risk measure, defined as follows.

**Definition 1 (Coherent Risk Measures).** Let \( (\Omega, \mathcal{F}, \mathbb{P}) \) be a probability space and let \( Z \) be the space of random variables on \( \Omega \). A coherent risk measure (CRM) is a mapping \( \rho: Z \to \mathbb{R} \) that obeys the following four axioms. For all \( Z, Z' \in Z \):

A1. **Monotonicity:** \( Z \leq Z' \Rightarrow \rho(Z) \leq \rho(Z') \).

A2. **Translation invariance:** \( \forall a \in \mathbb{R}, \rho(Z + a) = \rho(Z) + a \).

A3. **Positive homogeneity:** \( \forall \lambda \geq 0, \rho(\lambda Z) = \lambda \rho(Z) \).

A4. **Subadditivity:** \( \rho(Z + Z') \leq \rho(Z) + \rho(Z') \).

These axioms were originally proposed in (Artzner et al., 1999) to ensure the “rationality” of risk assessments. For example, A1 states that if a random cost \( Z \) is less than or equal to a random cost \( Z' \) regardless of the disturbance realizations, then \( Z \) must be considered less risky (one may think of the different random costs arising from different control policies). A4 reflects the intuition that a risk-averse agent should prefer to diversify. We refer the reader to (Artzner et al., 1999; Majumdar and Pavone, 2017) for a thorough justification of these axioms. An important characterization of CRMs is provided by the following representation theorem.

**Theorem 1 (Representation Theorem for Coherent Risk Measures (Artzner et al., 1999)).** Let \( (\Omega, \mathcal{F}, \mathbb{P}) \) be a probability space, where \( \Omega \) is a finite set with cardinality \( |\Omega| \), \( \mathcal{F} \) is the \( \sigma \)-algebra over subsets (i.e., \( \mathcal{F} = 2^\Omega \)), probabilities are assigned according to \( \mathbb{P} = (p(1), p(2), \ldots, p(|\Omega|)) \), and \( Z \) is the space of random variables on \( \Omega \). Denote by \( \mathcal{C} \) the set
of valid probability densities:
\[
C := \left\{ \zeta \in \mathbb{R}^{[\Omega]} \mid \sum_{i=1}^{[\Omega]} p(i)\zeta(i) = 1, \zeta \geq 0 \right\}.
\] (3)

Define \( q_\zeta \in \mathbb{R}^{[\Omega]} \) where \( q_\zeta(i) = p(i)\zeta(i), \ i = 1, \ldots, [\Omega] \). A risk measure \( \rho : Z \to \mathbb{R} \) with respect to the space \((\Omega, \mathcal{F}, \mathbb{P})\) is a CRM if and only if there exists a compact convex set \( B \subset C \) such that for any \( Z \in Z \):
\[
\rho(Z) = \max_{\zeta \in B} \mathbb{E}_{q_\zeta}[Z] = \max_{\zeta \in B} \sum_{i=1}^{[\Omega]} p(i)\zeta(i)Z(i).
\] (4)

This theorem is important for two reasons. Conceptually, it gives us an interpretation of CRMs as computing the worst-case expectation of the cost function over a set of densities \( B \) (referred to as the risk envelope). Coherent risk measures thus allow us to consider risk and ambiguity (ref. Section 1) in a unified framework since one may interpret an agent acting according to a coherent risk model as being uncertain about the underlying probability density. Second, it provides us with an algorithmic handle over CRMs and will form the basis of our approach to measuring experts’ risk preferences.

In this work, we will take the risk envelope \( B \) to be a polytope. We refer to such risk measures as polytopic risk measures, which were also considered in (Eichhorn and Römisch, 2005). By absorbing the density \( \zeta \) into the pmf \( p \), we can represent (without loss of generality) a polytopic risk measure as:
\[
\rho(Z) = \max_{p \in \mathcal{P}} \mathbb{E}_p[Z],
\] (5)
where \( \mathcal{P} \) is a polytopic subset of the probability simplex:
\[
\mathcal{P} = \left\{ p \in \Delta^{[\Omega]} \mid A_{ineq}p \leq b_{ineq} \right\},
\] (6)
where \( \Delta^{[\Omega]} := \left\{ p \in \mathbb{R}^{[\Omega]} \mid \sum_{i=1}^{[\Omega]} p(i) = 1, \ p \geq 0 \right\} \). Polytopic risk measures constitute a rich class of risk measures, encompassing a spectrum ranging from risk neutrality (\( \mathcal{P} = \{p\} \)) to worst-case assessments (\( \mathcal{P} = \Delta^{[\Omega]} \)). Examples include CVaR, mean absolute semi-deviation, spectral risk measures, optimized certainty equivalent, and the distributionally robust risk (Chow and Pavone, 2014). We further note that the ambiguity interpretation of CRMs is reminiscent of Gilboa & Schmeidler’s Minmax EU model for ambiguity-aversion (Gilboa and Schmeidler, 1989) which was shown to outperform various competing models in (Hey et al., 2010) for single-stage decision problems, albeit with more restrictions on the set \( \mathcal{B} \).

Goal: Given demonstrations from the expert in the form of state-control trajectories, our goal is to approximate the expert’s risk preferences by finding an approximation of the risk envelope \( \mathcal{P} \).
3 Risk-sensitive IRL: Single Decision Period

In this section we consider the single step decision problem, i.e., \( N = 0 \) in equation (2). Thus, the probability space \((\Omega, \mathcal{F}, \mathbb{P})\) is simply \((\mathcal{W}, 2^\mathcal{W}, \mathbb{P})\).

3.1 Known Cost Function

We first consider the static decision-making setting where the expert’s cost function is known but the risk measure is unknown. A coherent risk model then implies that the expert is solving the following optimization problem at state \( x \) in order to compute an optimal action:

\[
\tau^* := \min_{u \in \mathcal{U}} \rho(C(x, u)) = \min_{u \in \mathcal{U}} \max_{p \in \mathcal{P}} \mathbb{E}_p[C(x, u)] = \min_{u \in \mathcal{U}} \max_{p \in \mathcal{P}} g(x, u)^T p,
\]

where \(\rho(\cdot)\) is a CRM with respect to the space \((\mathcal{W}, 2^\mathcal{W}, \mathbb{P})\) (i.e., \(\mathcal{P} \subseteq \Delta^\mathcal{W}\)). In the last equation, \(g(x, u)(j)\) is the cost when the disturbance \(w[j] \in \mathcal{W}\) is realized. Since the inner maximization problem is linear in \(p\), the optimal value is achieved at a vertex of the polytope \(\mathcal{P}\). Denoting the set of vertices of \(\mathcal{P}\) as \(\text{vert}(\mathcal{P}) = \{v_i\}_{i \in \{1, \ldots, N_V\}}\), we can thus rewrite problem (7) above as follows:

\[
\min_{u \in \mathcal{U}, \tau} \tau \quad \text{s.t.} \quad \tau \geq g(x, u)^T v_i, \quad i \in \{1, \ldots, N_V\}
\]

If the cost function \(C(\cdot, \cdot)\) is convex in the control input \(u\), the resulting optimization problem is convex. Given a dataset \(D = \{(x^{*,d}, u^{*,d})\}_{d=1}^D\) of state-control pairs of the expert taking action \(u^{*,d}\) at state \(x^{*,d}\), our goal is to deduce an approximation \(\mathcal{P}_o\) of \(\mathcal{P}\) from the given data. The key idea of our technical approach is to examine the Karush-Kuhn-Tucker (KKT) conditions for Problem (9). The use of KKT conditions for Inverse Optimal Control is a technique also adopted in (Englert and Toussaint, 2015). The KKT conditions are necessary for optimality in general and are also sufficient in the case of convex problems. We can thus use the KKT conditions along with the dataset \(D\) to constrain the constraints of Problem (9). In other words, the KKT conditions will allow us to constrain where the vertices of \(\mathcal{P}\) must lie in order to be consistent with the fact that the state-control pairs represent optimal solutions to Problem (9). Importantly, we will not assume access to the number of vertices \(N_V\) of \(\mathcal{P}\).

Let \((x^*, u^*)\) be an optimal state-control pair and let \(\mathcal{J}^+\) and \(\mathcal{J}^-\) denote the sets of components of the control input \(u^*\) that are saturated above and below respectively (i.e., \(u(j) = u^+(j), \forall j \in \mathcal{J}^+\) and \(u(j) = u^-(j), \forall j \in \mathcal{J}^-\)).

**Theorem 2** (KKT-Based Inference). Consider the following optimization problem:
\[
\max_{v \in \Delta^L, \sigma_+, \sigma_- \geq 0} g(x^*, u^*)^T v
\]
\[
s.t. \quad 0 = \nabla_{u(j)} g(x, u)^T v|_{x^*, u^*} + \sigma_+(j), \forall j \in J^+
\]
\[
0 = \nabla_{u(j)} g(x, u)^T v|_{x^*, u^*} - \sigma_-(j), \forall j \in J^-
\]
\[
0 = \nabla_{u(j)} g(x, u)^T v|_{x^*, u^*}, \forall j \notin J^+, j \notin J^-
\]

Denote the optimal value of this problem by \(\tau'\) and define the halfspace:
\[
\mathcal{H}(x^*, u^*) := \{v \in \mathbb{R}^L \mid \tau' \geq g(x^*, u^*)^T v\}.
\]

Then, the risk envelope \(\mathcal{P}\) satisfies \(\mathcal{P} \subset (\mathcal{H}(x^*, u^*) \cap \Delta^L)\).

**Proof.** The KKT conditions for Problem (9) are:
\[
1 = \sum_{i=1}^{N_V} \lambda_i,
\]
\[
0 = \lambda_i[g(x^*, u^*)^T v_i - \tau], i = 1, \ldots, N_V,
\]
and for \(j = 1, \ldots, m\):
\[
0 = \sigma_+(j) - \sigma_-(j) + \sum_{i=1}^{N_V} \lambda_i \nabla_{u(j)} g(x, u)^T v_i|_{x^*, u^*},
\]
\[
0 = \sigma_+(j)[u^+(j) - u^-(j)], 0 = \sigma_-(j)[u^-(j) - u^*(j)],
\]
where \(\lambda_i, \sigma_+(j), \sigma_-(j) \geq 0\) are multipliers. Now, suppose there are multiple optimal vertices \(\{v_i\}_{i \in I}\) for Problem (9) in the sense that \(\tau^* = g(x^*, u^*)^T v_i, \forall i \in I\). Defining \(\bar{v} := \sum_{i \in I} \lambda_i v_i\), we see that \(\bar{v}\) satisfies:
\[
0 = \nabla_{u(j)} g(x^*, u^*(j))^T \bar{v} + \sigma_+(j) - \sigma_-(j), \quad j = 1, \ldots, m,
\]
and \(\tau^* = g(x^*, u^*)^T \bar{v}\) since \(\sum_{i \in I} \lambda_i = 1\). Now, since \(\bar{v}\) satisfies the constraints of Problem (10) (which are implied by the KKT conditions), it follows that \(\tau' \geq \tau^*\). From problem (9), we see that \(\tau' \geq \tau^* \geq g(x^*, u^*)^T v_i, \forall v_i \in \text{vert}(\mathcal{P})\) and thus \(\mathcal{P} \subset (\mathcal{H}(x^*, u^*) \cap \Delta^L)\).

Problem (10) is a Linear Program (LP) and can thus be solved efficiently. For each demonstration \((x^*, u^*, d) \in \mathcal{D}\), Theorem 2 provides a halfspace constraint on the risk envelope \(\mathcal{P}\). By aggregating these constraints, we obtain a polytopic outer approximation \(\mathcal{P}_o\) of \(\mathcal{P}\). This is summarized in Algorithm 1. Note that Algorithm 1 operates sequentially through the data \(\mathcal{D}\) and is thus directly applicable in online settings.
Algorithm 1 Outer Approximate Risk Envelope

1: Initialize $P_0 = \Delta_L$
2: for $d = 1, \ldots, D$ do
3: Solve Linear Program (10) with $(x^*, d, u^*, d)$ to obtain a hyperplane $H_{(x^*, d, u^*, d)}$
4: Update $P_o \leftarrow P_o \cap H_{(x^*, d, u^*, d)}$
5: end for
6: Return $P_o$

Remark 1. Algorithm 1 is a non-parametric algorithm for inferring the expert’s risk measure; i.e., we are not fitting parameters for an a priori chosen risk measure. Instead, by operating directly within the dual space, Algorithm 1 can recover any risk measure within the class of all CRMs, that best explains the expert’s demonstrations.

As we collect more half-space constraints in Algorithm 1, the constraint $v \in \Delta_L$ in Problem (10) above can be replaced by $v \in P_o$, where $P_o$ is the current outer approximation of the risk envelope. It is easily verified that the results of Theorem 2 still hold. This allows us to obtain a tighter (i.e., lower) upper bound $\tau'$ for $\tau^*$, thus resulting in tighter halfspace constraints for each new demonstration processed by the algorithm.

Denote $P_D$ the output of Algorithm 1 after processing sequentially the first $D$ demonstrations $\{(x^*, d, u^*)\}_{d=1}^D$. Observe that for all $D \geq 1$, $P_{D+1} \subseteq P_D$. We can then define the limiting set as $P_\infty := \bigcap_{d=1}^\infty P_d$. An important consideration for this algorithm is whether it is possible to recover, at least from an imitation perspective, the risk envelope $P$ from sufficiently many optimal demonstrations. In other words, we are specifically interested in the question of whether the limiting set $P_\infty$ (whenever such a limit exists) allows one to exactly predict the actions of a decision maker that operates under a risk model characterized by the set $P$. In the following theorem we establish, under mild technical conditions, that this is indeed possible. The proof is provided in Appendix A.

Theorem 3 (Convergence of Algorithm 1). Let $S \subseteq \mathbb{R}^n$ be a convex, compact subset of the state space. Let $\{(x^*, d, u^*)\}_{d=1}^\infty$ be a set of infinitely many optimal demonstrations such that the sequence $\{x^*\}$ is dense in $S$. Assume that the following technical conditions hold:

A.1 The expert’s cost vector $g(x, u)$ is strictly convex with respect to $(x, u)$.

A.2 For all $j \in \{1, \ldots, L\}$ and any state $x \in S$, the cost function associated with the $j$-th disturbance $u \mapsto g(x, u)(j)$ has bounded level sets.

Finally, for any risk envelope $P' \subseteq \Delta_L$ and any state $x \in S$, define

$$u(P', x) := \arg\min_{u \in U} \max_{v \in P'} v^T g(x, u),$$

as the optimal control action of an expert with risk envelope $P'$ at state $x$. Then, for any state $x \in S$,

$$u(P_\infty, x) = u(P, x).$$  \hspace{1cm} (17)
That is, for any state \( x \in S \), the optimal action predicted using the limiting envelope \( P_\infty \) matches that computed using the true expert polytope \( \mathcal{P} \).

Once we have recovered an approximation \( \mathcal{P}_o \) of \( \mathcal{P} \), we can solve the “forward” problem (i.e., compute actions at a given state \( x \)) by solving the optimization problem (7) with \( \mathcal{P}_o \) as the risk envelope.

### 3.1.1 Example: Linear-Quadratic System

As a simple illustrative example to gain intuition for the convergence properties of Algorithm 1, consider a linear dynamical system with multiplicative uncertainty of the form 
\[
f(x_k, u_k, w_k) = A(w_k)x_k + B(w_k)u_k.
\]
We consider the one-step decision-making process with a quadratic cost on state and action: 
\[
Z := u_0^T R u_0 + x_1^T Q x_1,
\]
where \( x_1 = A(w_0)x_0 + B(w_0)u_0 \). Here, \( R > 0 \) and \( Q \succeq 0 \). We consider a 10-dimensional state space with a 5-dimensional control input space. The number of realizations is taken to be \( L = 3 \) for ease of visualization. The \( L \) different \( A(w_k) \) and \( B(w_k) \) matrices corresponding to each realization are generated randomly by independently sampling elements of the matrices from the standard normal distribution \( \mathcal{N}(0, 1) \). The cost matrix \( Q \) is a randomly generated positive semi-definite matrix and \( R \) is the identity. States \( x^* \) are drawn \( \mathcal{N}(0, 1) \). The true envelope was generated by taking the convex hull of a set of random samples in the simplex \( \Delta^L \).

Figure 2 shows the outer approximations of the risk envelope obtained using Algorithm 1. We observe rapid convergence (approximately 20 sampled states \( x^* \)) of the outer approximations \( \mathcal{P}_o \) (red) to the true risk envelope \( \mathcal{P} \) (green).

Figure 3 shows the mean squared error (on an independent test set with 30 demonstrations) between actions predicted using the sequentially refined polytope approximations generated by Algorithm 1 and the expert’s true actions, as a function of the number of training demonstrations.

![Figure 3: Rapid decrease of the mean squared error between predicted and expert’s actions as a function of the number of demonstrations.](image-url)
3.2 Unknown Cost Function

Now we consider the more general case where both the expert’s cost function and risk measure are unknown. We parameterize the cost function as a linear combination of features in \((x, u)\). Then, the expected value of the cost function w.r.t. \(p \in \Delta^L\) can be written as

\[
g(x, u)(j) = \sum_{h=1}^{H} c(h) \phi_{j,h}(x, u), \quad j = 1, \ldots, L, \tag{18}
\]

with nonnegative weights \(c \in \mathbb{R}_H^+\). Since the solution of problem (7) solved by the expert is invariant to positive scalings of the cost function due to the positive homogeneity property of coherent risk measures (see Definition 1), one can assume without loss of generality that the feature weights sum to 1.

With this cost structure, we see that the KKT conditions derived in Section 3.1 involve products of the feature weights \(c\) and the vertices \(v_i\) of \(P\). Similarly, an analogous version of optimization problem (10) can be used to bound the optimal value. This problem again contains products of the unknown feature weights \(c\) and the probability vertex \(v\). The key idea here is to introduce new decision variables \(z\) that replace each product \(v(j)c(h)\) by a new variable \(z_{jh}\) which allows us to re-write problem (10) as an LP in \((z, \sigma_+, \sigma_-)\), with
the addition of the following two simple constraints: $0 \leq z_{jh} \leq 1, \forall j, h$, and $\sum_{j,h} z_{jh} = 1$. In a manner analogous to Theorem 2, this optimization problem allows us to obtain bounding hyperplanes in the space of product variables $z$ which can then be aggregated as in Algorithm 1. Denoting this polytope as $\mathcal{P}_z$, we can then proceed to solve the “forward” problem (i.e., computing actions at a given state $x$) by solving the following optimization problem:

$$\min_{u \in U} \max_{z \in \mathcal{P}_z} \sum_{j,h} z_{jh} \phi_{j,h}(x,u). \quad (19)$$

This problem can be solved by enumerating the vertices of the polytope $\mathcal{P}_z$ in a manner similar to problem (9). Similar to the case where the cost function is known, this provides us with a way to conservatively approximate the expert’s decision-making process (in the sense that we are considering a larger risk envelope).

### 3.2.1 Approximate Recovery of Cost and Risk Measure

While the procedure described above operates in the space of product variables $z$ and does not require explicitly recovering the cost function and risk envelope separately, it may nevertheless be useful to do so for two reasons. First, the number of vertices of $\mathcal{P}_z$ may be quite large (since the space of product variables may be high dimensional) and thus solving the forward problem (19) may be computationally expensive. Recovering the cost and risk envelope separately allows us to solve a smaller optimization problem (since the risk envelope is lower dimensional in this case). Second, recovering the cost and risk measure separately may provide additional intuition and insights into the expert’s decision-making process and may also allow us to make useful predictions in novel settings (e.g., where we expect the expert’s risk measure to be the same but not the cost function or vice versa).

Here we describe a procedure for approximately recovering the feature weights and the risk envelope from the polytope $\mathcal{P}_z$. The key observation that makes this possible is to note that the matrix $z$ containing the variables $z_{jh}$ is equal to the outer product $ve^T$ by definition. Hence, for $h = 1, \ldots, H$, we have:

$$\sum_{j=1}^{L} z_{jh} = \sum_{j=1}^{L} v(j)c(h) = c(h) \sum_{j=1}^{L} v(j) = c(h). \quad (20)$$

The last equality follows from the fact $v$ is a probability vector and sums to 1. Similarly, for $j = 1, \ldots, L$, we have:

$$\sum_{h=1}^{H} z_{jh} = \sum_{h=1}^{H} v(j)c(h) = v(j) \sum_{h=1}^{H} c(h) = v(j). \quad (21)$$

The last equality follows from the fact that we assumed without loss of generality that the feature weights sum to 1.
Let \( \{\hat{z}_i\} \) be the set of vertices of the polytope \( P_z \). We can then apply equations (20) and (21) to each vertex. If we have exactly recovered the polytope \( P_z \) in the space of product variables, the estimates of the feature weights obtained from the different vertices will be equal. Since in general this will not be the case, the different estimates of the weights will be different (but ideally similar). We can take the mean of the different estimates for \( c \) (from (20)) as our estimate for the weight vector \( c \) and the convex hull of the estimates of the probability vectors from equation (21) applied to each vertex of \( P_z \) to be our estimate of the risk envelope.

It is important to be able to gauge the quality of the estimates we obtain from the procedure above. We can do this in two ways. First, if the estimates of the weight vector are tightly clustered, this is a good indication that we have an accurate recovery. Second, if each vertex \( \hat{z}_i \) of the polytope is close to a rank one matrix, then this is again a good indication (since the true product variables \( z \) equal \( v c^T \)).

### 3.2.2 Example: Linear-Quadratic System

Consider the same system as 3.1.1, but now we assume that the cost function is unknown. We take the cost function as the weighted sum of three quadratic features (i.e., \( H = 3 \)). The quadratic features are generated randomly by taking them to be equal to \( MM^T \), where the elements of \( M \) are sampled from the standard normal distribution. The corresponding weights are drawn uniformly between 0 and 1 and are normalized to sum to 1.

Figure 4 a) illustrates the tightness of the approximate envelope as compared with the true polytope while Figure 4 b) is a scatter plot of the first two feature weights (the third is uniquely determined given the first two) as recovered from applying eq. (20) to each vertex of the compound polytope \( P_z \). Notice that the cost weight estimates are tightly clustered, as desired.
4 Risk-sensitive IRL: Multi-step case

We now consider the dynamical system given by (1) and generalize the one-step decision problem to the multi-step setting. We consider a model where the disturbance $w_k$ is sampled every $N$ time-steps and held constant in the interim. Such a model generalizes settings where disturbances are sampled i.i.d. at every time-step (corresponding to $N = 1$ in our model) as it also allows us to model delays in the expert’s reaction to changing disturbances. We model the expert as planning in a receding horizon manner by looking ahead for a finite horizon. Owing to the need to account for future disturbances, the multi-step finite-horizon problem is a search over control policies (i.e., the executed control inputs depend on which disturbance is realized).

4.1 Prepare-React Model

In this section we reprise the “prepare” – “react” model introduced in (Majumdar et al., 2017), and depicted below in Figure 5. The expert’s policy is decomposed into two phases (shown in Figure 5), referred to as “prepare” and “react.” The “prepare” phase precedes each disturbance realization by $N - n_d$ steps while the “react” phase follows it for $n_d$ steps. Intuitively, this model captures the idea that in the period preceding a disturbance (i.e., the “prepare” phase) the expert controls the system to a state from which he/she can recover well (in the “react” phase) once a disturbance is realized. Studies showing that humans have a relatively short look-ahead horizon in uncertain decision-making settings lend credence to such a model (Carton et al., 2016). As in (Majumdar et al., 2017), the delay parameter $n_d$ would be learned directly from the demonstrations. In this work, we wish to overcome the primary limitation of the dynamic model in (Majumdar et al., 2017), namely that of single branching events in each prediction/planning horizon. To do this, we first must define the notion of dynamic risk measures, used to assess risk over sequential realizations of uncertainty.

Figure 5: Scenario tree as visualized at time $k = 0$. The disturbance is sampled every $N$ steps. The control look-ahead has two phases: “prepare” and “react”.

\[ k = 0 \]
\[ w_0 = w^{(1)} \]
\[ x_{k+1} = f(x_k, u_k, w_0) \]
\[ k = n_d \]
\[ k = N \]
\[ k = N + n_d \]

“Prepare”

Control look-ahead

“React”
4.2 Dynamic Risk Measures

Consider a discrete-time stochastic cost sequence \( \{Z_t\} \), where \( Z_t \in \mathcal{Z}_t \) the space of real-valued non-negative random variables at stage \( t \). Let \( \mathcal{Z}_{t,t'} := \mathcal{Z}_t \times \cdots \times \mathcal{Z}_{t'} \). A dynamic risk measure is a sequence of risk measures \( \rho_{t,t'} : \mathcal{Z}_{t,t'} \to \mathbb{R} \), each mapping a future stream of random costs into a risk assessment at stage \( t \) and satisfying the monotonicity property \( \rho_{t,t'}(Z_{t,t'}) \leq \rho_{t,t'}(Y_{t,t'}) \) for all \( Z_{t,t'}, Y_{t,t'} \in \mathcal{Z}_{t,t'} \) such that \( Z_{t,t'} \leq Y_{t,t'} \). The monotonicity property is an intuitive extension of the monotonicity property for single-step risk assessments, and an arguably defensible axiom for all risk assessments.

To give dynamic risk measures a concrete functional form, we need to generalize the CRM axioms presented in Definition 1 to the dynamic case.

**Definition 2** (Coherent One-Step Conditional Risk Measures). A coherent one-step conditional risk measure is a mapping \( \rho_t : Z_{t+1} \to \mathcal{Z}_t \), for all \( t \in \mathbb{N} \), that obeys the following four axioms. For all \( Z_{t+1}, Y_{t+1} \in \mathcal{Z}_{t+1} \) and \( Z_t \in \mathcal{Z}_t \):

**A1. Monotonicity:** \( Z_{t+1} \leq Y_{t+1} \Rightarrow \rho_t(Z_{t+1}) \leq \rho_t(Y_{t+1}) \).

**A2. Translation invariance:** \( \rho_t(Z_{t+1} + Z_t) = \rho_t(Z_{t+1}) + \rho_t(Z_t) \).

**A3. Positive homogeneity:** \( \forall \lambda \geq 0, \rho_t(\lambda Z_{t+1}) = \lambda \rho_t(Z_{t+1}) \).

**A4. Subadditivity:** \( \rho_t(Z_{t+1} + Y_{t+1}) \leq \rho_t(Z_{t+1}) + \rho_t(Y_{t+1}) \).

Note that each \( \rho_t \) is a random variable on the space \( \mathcal{Z}_t \) and given the discrete underlying probability space, each component of \( \rho_t \) is uniquely identified by the sequence of disturbances preceding stage \( t \) (hence the term conditional). Furthermore, it is readily observed that a mapping \( \rho_t : Z_{t+1} \to Z_t \) is a coherent one-step conditional risk measure if and only if each component of \( \rho_t \) is a CRM.

As investigated in (Ruszczyński, 2010), in order for dynamic risk assessments to satisfy the intuitive monotonicity condition and to ensure rationality of evaluations over time, a dynamic risk measure must have the following compositional form:

\[
\rho_{t,t'}(Z_{t,t'}) := Z_t + \rho_t(Z_{t+1} + \rho_{t+1}(Z_{t+2} + \cdots + \rho_{t'+1}(Z_{t'}) \cdots)),
\]

(22)

where each \( \rho_t \) is a coherent one-step conditional risk measure. Figure 6 provides a helpful visualization of this compounded functional form.

4.3 Multiple-Branch Prepare-React Model

We are now ready to define the multiple-branch formulation of the expert’s multi-stage problem, as envisioned at time-step \( k \), with look-ahead horizon \( TN \) steps where \( T \in \mathbb{N}_{\geq 1} \) denotes the number of branching events within the prediction horizon. As in the prepare-react model introduced earlier, we assume that the disturbance mode for the first \( N - n_d \) steps starting at time-step \( k \) corresponds to \( w_{k-n_d} \), following which the disturbance is resampled every \( N \) steps. A schematic of the multiple-branch generalization of the prepare-react model is depicted below in Figure 7.
Let $x_{k|k}$ denote the predicted state for time-step $k + k'$, where $k' \in [0, TN - 1]$, as predicted at time-step $k$ within the expert’s multi-stage optimization problem. Similarly, let $w'_t$ for $t \in [0, T - 1]$ represent the predicted (stage-wise) disturbance sequence. As the optimization is over “prepare” – “react” control policies, let $\hat{\pi}_t(\omega_{t-1}, w'_t)$, $t \in [0, T - 1]$ denote the expert’s “prepare” – “react” control policy for stage $t$ (i.e., control input sequence for time-steps $k' \in [TN, (t + 1)N - 1]$) as a function\(^1\) of the partial predicted disturbance history $\omega_{t-1} := \{w'_0, w'_1, \ldots, w'_{t-1}\}$, and the next predicted disturbance mode $w'_t$. We take $w'_{-1} =: w^{*}_{-1|k}$ to represent the actual disturbance mode in progress at the time of solving the multi-stage optimization problem at time-step $k$. Note that, by causality, only the “react” portion of $\hat{\pi}_t$ may be a function of $w'_t$ but not the “prepare” portion. Finally, denote $C_{t N:(t+1)N-1}(x_{1N}|k, \hat{\pi}_t(\cdot))$ to be the accumulated cost over time-steps $[t N, (t+1)N - 1]$ given the stage $t$ “prepare” – “react” control policy $\hat{\pi}_t$. The expert’s multi-period optimization problem is then given as:

$$\min_{\hat{\pi}_t} \rho_t\left(C_{0:N-1}(\cdot, \hat{\pi}_0) + \rho_1(C_{N:2N-1}(\cdot, \hat{\pi}_1) + \cdots + \rho_{T-1}(C_{(T-1)N:TN-1}(\cdot, \hat{\pi}_{T-1})) \cdots)\right),$$

(23)

where each $\rho_t$, $t = 0, \ldots, T - 1$ is a coherent one-step conditional risk measure such that each component of $\rho_t$ is a CRM $\rho(\cdot)$ with respect to the probability space $(\mathcal{W}, 2^\mathcal{W}, P)$ and characterized by the fixed risk-envelope $\mathcal{P} \subset \Delta^N$. Leveraging the translational invariance property, the objective may be equivalently re-written as

$$C_{0:N-n_d-1} + \rho_0\left(C_{N-n_d:N-1} + C_{N:2N-n_d-1} + \rho_1(C_{2N-n_d:2N-1} + \cdots + \rho_{T-1}(C_{TN-n_d:TN-1}) \cdots)\right).$$

One should notice that (1) for each $t \in \{0, \ldots, T-1\}$, the cost sequence $C_{t N:(t+1)N-1}$ is split across the risk operator $\rho_t$ due to the “prepare”–“react” structure and the translational invariance property, and (2) the risk mapping is over a sum of costs since disturbances are

\(^1\)For notational clarity, we suppress the obvious dependence on $x_k$ and partial policy history $\{\hat{\pi}_0, \ldots, \hat{\pi}_{t-1}\}$. 

---

Figure 6: A scenario tree with three uncertain outcomes at each stage. The one-step risk mapping $\rho_1(Z_2) \in \mathcal{Z}_1$ maps the random cost $Z_2 \in \mathcal{Z}_2$ to a risk assessment at stage 1, i.e., is a random variable on $\mathcal{Z}_1$ and is thus isomorphic to the space $\mathbb{R}^3_{\geq 0}$. Here, component $j$ of $\rho_1(Z_2)$, associated with node $j$ at stage 1 (green node), is a coherent one-step conditional risk mapping over the children of node $j$ at stage 2. The mapping $\rho_0(\rho_1(Z_2))$ subsequently maps the risk-assessments at stage 1 (i.e., $\rho_1(\cdot)$) back to stage 0.
Figure 7: Multiple-branch scenario tree schematic for the prepare-react model, as “visualized” by the expert at time-step $k$ (and indexed internally using $k'$ for simplicity). The disturbance is sampled every $N$ steps. The control look-ahead consists of multiple nested branches of “prepare” and “react”; displayed in the figure above (shaded green) is one such nested branch corresponding to $w_0 = w_1^{[1]}$. To evaluate costs over $t = 1$, it is assumed that the expert is leveraging the conditional CRM $\rho_1(\cdot)$ where for each realization of $x_N | k$ (identified uniquely by the observed disturbance branch at stage 0), the expert uses the static CRM $\rho(\cdot)$ over the nested outcomes (shown in green for one possible realization of $x_N | k$). The observed control sequence is the beginning “prepare” – “react” sequence corresponding to the actual realized disturbance $w_0^* | k$.

sampled every $N$ steps. These two observations elucidate the stage-wise decomposition of the dynamic risk measure as introduced in (22), where each stage corresponds to the cost accrued over the $N$ time-steps in between consecutive disturbance branching events. The observed input from the expert is the “prepare” – “react” control policy $\hat{\pi}_0^*(w_{-1 | k}, w_0^* | k)$ where $w_0^* | k$ represents the actual disturbance mode sampled after time-step $k$, following which the expert re-solves the problem.

Notice that by setting $T = 1$, we recover the single-branch prepare/react model presented in (Majumdar et al., 2017). The success of the dynamic model in (Majumdar et al., 2017) followed from reducing the multi-step inference problem to be mathematically equivalent to the single-step case by first inferring the (un-observed) control policies of the human agent, corresponding to the un-realized disturbance branches. Consider the scenario tree decomposition in Figure 5. If disturbance $w_{[3]}$ is realized at $k = N$, then we only observe the “react” control sequence corresponding to the third branch. The algorithm in (Majumdar et al., 2017) proceeded by first inferring the “react” control sequences for the un-observed branches and then constructing a bounding hyperplane using a similar version of problem (10). In a multiple-branch setting however, it is exceedingly difficult to exactly infer (or approximate) the unobserved control policies as each of these policies involves an unobserved nested optimization over future branching events. Consequently, the optimality conditions of an observed control policy are defined by equalities that are non-linear in the unobserved variables. Therefore, extending the use of KKT conditions
to infer an outer approximation of the risk envelope in the style of Theorem 2 leads to an intractable non-convex optimization problem. To address this fundamental observability issue, we introduce a semi-parametric representation of the risk-envelope, discussed next.

### 4.3.1 Semi-Parametric CRM

Fix a set of $M$ normal vectors $a_j \in \mathbb{R}^L, j = 1, \ldots, M$. Define the polytope $\mathcal{P}_r$, parameterized by the vector $r \in \mathbb{R}^M$ as:

$$
\mathcal{P}_r := \{ v \in \Delta^L : a_j^T v \leq b(j) - r(j), \quad j = 1, \ldots, M \},
$$

where for each $j$, $b(j) := \max_{v \in \Delta^L} a_j^T v$. The CRM induced by this dual representation is denoted as $\rho^r(\cdot)$ and assumes the functional form

$$
\rho^r(Z) = \max_{v \in \Delta^L \cap \mathcal{P}_r} E[v][Z],
$$

where $Z \in \mathbb{R}^L$ is a discrete random variable with $L$ possible realizations. This induced CRM is termed semi-parametric since unlike methods where one seeks to find the parameters defining a fixed disutility function (e.g., Shen et al. (2014); Ratliff and Mazumdar (2017)), here we do not assume a fixed chosen risk measure. Instead, by parameterizing the risk functional in the dual space (via its risk envelope characterization), we retain the generality to recover any polytopic CRM, given sufficient number of normal vectors $a_j$. A potential method to choose the normal vectors $a_j$ is to take the halfplane normals from the multi-step KKT method described in (Majumdar et al., 2017).

In order to ensure that the polytope described by (24) intersected with the probability simplex $\Delta^L$ is non-empty, define the extended polytope

$$
\tilde{\mathcal{P}}_r := \{ (v, r) : v \in \Delta^L, a_j^T v + r(j) \leq b(j), j = 1, \ldots, M \},
$$

and define $\mathcal{R} := \text{proj}_r \tilde{\mathcal{P}}_r$ to be the projection of this extended polytope along the $r$ variables. Then, $r \in \mathcal{R}$ ensures that the polytope $\mathcal{P}_r$ is non-empty. It is readily observed that the set $\mathcal{R}$ is also a polytope.

**Remark 2.** While we lose the outer-approximation of the risk envelope and convergence guarantees associated with the KKT method, in its place we obtain a tractable algorithm that enables us to accommodate a substantially larger class of dynamic decision-making inference problems. Experimental results, as discussed in Section 5, confirm that the method works well in approximating a wide range of risk profiles.

### 4.3.2 Constrained Maximum Likelihood

Given the semi-parametric representation of the risk envelope $\mathcal{P}_r$ given in (24), the RS-IRL problem reduces to inference over the offset vector $r$, and cost-weight vector $c$. We will
perform this inference using a constrained maximum likelihood model. Consider, first, the following likelihood model:

\[
l(r, c) = \frac{\Pr(\hat{\pi}_0^*(w^-_{-1|k}, w^*_0|k))}{\int \Pr(\hat{\pi}_0^*(w^-_{-1|k}, w^*_0|k)) d\hat{\pi}_0(w^-_{-1|k}, w^*_0|k)}, \tag{25}
\]

where, similar to the motivation for the MaxEnt IRL model (Ziebart et al., 2008), we take \(\Pr(\hat{\pi}_0^*(w^-_{-1|k}, w^*_0|k))\) to be proportional to the exponential of the negative optimal value of (23), computed using the envelope (24) and conditioned on \(w'_1 = w^*_0|k\) and \(\hat{\pi}_0 = \hat{\pi}_0(w^-_{-1|k}, w^*_0|k)\) (see Appendix B for a detailed derivation of this distribution). While the original MaxEnt IRL model is motivated by finding the maximum entropy distribution subject to an expected feature matching constraint, the robust performance of this model even in the absence of such a statistical motivation has been extensively observed and leveraged in the IRL literature.

A key limitation of this formulation however is the complexity of the partition function and the resulting gradients. The form of (25) is a distribution over all possible \(N\)-length policies. This makes sampling-based approximations intractable, as similarly observed in (Kretzschmar et al., 2016), and Laplace integral-based approximations as used in (Levine and Koltun, 2012) too imprecise.

In order to construct a tractable algorithm, we employ the simplification whereby at the beginning of any “prepare” stage in Figure 7, the expert may only choose from a finite set of open-loop control trajectories \(\Pi := \{\hat{u}^{[1]}, \ldots, \hat{u}^{[K]}\}\), each of length \(N\), thereby eliminating the notion of a “react” policy and replacing it with an open-loop sequence spanning the entire “prepare” – “react” stage. These trajectories can be chosen for instance by running the K-means clustering algorithm on the raw input trajectories. This simplification allows us to interpret problem (23) as a game between the expert with action set \(\Pi\) and nature with action set \(\mathcal{W}\), and uniquely identify any predicted state \(x_{TN|k} \in [1, T - 1]\) using the predicted game history, i.e., disturbance history \(\omega_{t-1}\) and control history \(u_{t-1} := \{\hat{u}_0, \ldots, \hat{u}_{t-1}\}\). Leveraging this discrete representation and dynamic programming, we can construct the optimal solution to the expert’s multi-stage optimization problem using a “risk-sensitive” Bellman recursion, defined below.

**Terminal Stage:** For all possible game histories at stage \(T - 1\), define

\[
\tau[u_{T-2}, \omega_{T-2}](\hat{u}) := \rho(C(T-1)N,TN-1(x(T-1)N|k, \hat{u}))
\]

\[
\hat{\pi}_{T-1}^*[u_{T-2}, \omega_{T-2}] := \arg\min_{\hat{u} \in \Pi} \tau[u_{T-2}, \omega_{T-2}](\hat{u}).
\]
**Recursion:** For all possible game histories at stage \( t \), for \( t = T - 2, \ldots, 1 \):

\[
\tau[\mathbf{u}_{t-1}, \mathbf{w}_{t-1}](\hat{u}) := \rho \left( C_{t:N}(t+1)N_{-1}(x_{tN|k}, \hat{u}) + \min_{\hat{u}' \in \Pi} \tau[\{\mathbf{u}_{t-1}, \hat{u}'\}, \{\mathbf{w}_{t-1}, \mathbf{w}'_{t}\}](\hat{u}') \right)
\]

\[
\hat{\pi}_t[\mathbf{u}_{t-1}, \mathbf{w}_{t-1}] := \arg\min_{\hat{u} \in \Pi} \tau[\mathbf{u}_{t-1}, \mathbf{w}_{t-1}](\hat{u}).
\]

**First Stage:**

\[
\tau[w_{T-1}'](\hat{u}) := \rho \left( C_{0,N-1}(x_k, \hat{u}) + \min_{\hat{u}' \in \Pi} \tau[\{\hat{u}'\}, \{\omega_0'\}](\hat{u}') \right)
\]

\[
\hat{\pi}_{T-1}^*[w_{T-1}'] := \arg\min_{\hat{u} \in \Pi} \tau[w_{T-1}'](\hat{u}).
\]

The value \( \min_{\hat{u} \in \Pi} \tau[w_{t-1}^*|k](\hat{u}) \) is the optimal value of problem (23). In the equations above, it is understood that for each \( t \in [0, T-1] \), the cost sequence \( C_{t:N}(t+1)N_{-1} \) is evaluated based on the previous disturbance mode \( w_{t-1}' \) for the first \( N - n_d \) steps, followed by \( w_t' \) for the remaining \( n_d \) steps.

Given the structure of the optimal solution of problem (23), presented in Bellman form above using the true CRM \( \rho(\cdot) \), we now construct a *computationally tractable* likelihood model for the parameters \( r \) and \( c \) by defining the soft risk-sensitive Bellman recursion using the semi-parametric CRM \( \rho^s(\cdot) \). For the terminal stage, define

\[
\hat{\tau}[\mathbf{u}_{T-2}, \mathbf{w}_{T-2}](\hat{u}) := \rho^s \left( C_{(T-1)N:T}N_{-1}(x_{(T-1)N|k}, \hat{u}) \right),
\]

and for all \( t = T - 2, \ldots, 1 \):

\[
\hat{\tau}[\mathbf{u}_{t-1}, \mathbf{w}_{t-1}](\hat{u}) := \rho^s \left( C_{t:N}(t+1)N_{-1}(x_{tN|k}, \hat{u}) + \text{softmin}_{\hat{u}' \in \Pi} \hat{\tau}[\{\mathbf{u}_{t-1}, \hat{u}'\}, \{\mathbf{w}_{t-1}, \mathbf{w}'_{t}\}](\hat{u}') \right)
\]

\[
\hat{\tau}[w_{t-1}'](\hat{u}) := \rho^s \left( C_{0:N-1}(x_k, \hat{u}) + \text{softmin}_{\hat{u}' \in \Pi} \hat{\tau}[\{\hat{u}'\}, \{\omega_0'\}](\hat{u}') \right),
\]

where \( \text{softmin}_{x} f(x) := -\log \sum_x \exp(-f(x)) \). Let \( \hat{u}_t^* \) be the closest (in \( L_2 \) norm) trajectory in \( \Pi \) to the observed control sequence over time-steps \( [tN, (t+1)N - 1] \). Similar to the MaxEnt IRL approach, we allow for imperfect human demonstrations by postulating that lower risk-sensitive cost actions (i.e., \( \hat{\tau}[w_{t-1}^*|tN](\hat{u}) \)) are exponentially preferred, i.e.,

\[
\Pr(\hat{u}) \propto \exp \left( -\beta \hat{\tau}[w_{t-1}^*|tN](\hat{u}) \right),
\]

\(^2\text{We use } t \text{ here for notational consistency between the stagewise decomposition of the multi-step problem and demonstrated action trajectories.}\)
where $\beta > 0$ is an inverse temperature parameter. Thus, the likelihood of parameters $r, c$ is given by:

\[
l(r, c|\hat{u}_t^*) := \frac{\exp \left( -\beta \tilde{\tau} [w^*_{t-1|tN}](\hat{u}_t^*) \right)}{\sum_{\tilde{u}} \exp \left( -\beta \tilde{\tau} [w^*_{t-1|tN}](\tilde{u}) \right)}.
\]

As the expert is assumed to solve the problem in receding horizon fashion, we may treat each $(w^*_{t-1|tN}, \hat{u}_t^*)$ tuple in the demonstrated trajectory $T^*$ independently. Consequently, the log likelihood given the entire trajectory is simply

\[
l(r, c|T^*) = \frac{1}{|T^*|} \sum_{\hat{u}_t^* \in T^*} -\beta \tilde{\tau} [w^*_{t-1|tN}](\hat{u}_t^*) + \text{softmin} \beta \tilde{\tau} [w^*_{t-1|tN}](\hat{u}),
\]

where $|T^*|$ is the number of $N$-step demonstrations in the trajectory $T^*$. The inference problem is now defined as

\[
\{r^*, c^*\} := \arg\max_{c \in \Delta^H, r \in \mathbb{R}} l(r, c|T^*),
\]

where as before, we assume that the cost weights are non-negative and sum to one (and thus lie in the simplex $\Delta^H$). We solve the problem using projected gradient descent on $r$ and entropic mirror descent on $c$. The gradient formulas are derived by propagating gradients of the $\tilde{\tau}$ variables in recursive fashion from the terminal to the first stage (similar to computation of $\tilde{\tau}$ itself) and leveraging LP sensitivity results. For ease of exposition, we provide these recursive formulas in Appendix C.

5 Example: Driving Game Scenario

We now apply our RS-IRL framework on a simulated driving game (Figure 1) with ten human participants to demonstrate that our approach is able to infer individuals’ varying attitudes toward risk and mimic the resulting driving styles. In particular, we note that the experimental setting here constitutes a significantly more challenging and dynamic testbed than typical benchmark examples such as grid-world or sequential investment tasks.

5.1 Experimental Setting

The setting consists of a leader car and a follower car, simulated in the commercial driving simulator Vires VTD (VIRES Simulationstechnologie GmbH). Participants controlled the follower car with the Logitech G29 control suite, consisting of a steering wheel and pedals (Figure 1). The follower car is modeled using the simple car model with states: $x_f$ (along-track position), $y_f$ (lateral position), $v_f$ (speed), $\theta_f$ (yaw angle) and $\delta_f$ (steering angle). The dynamics are given by:

\[
\dot{x}_f = v_f \cos(\theta_f), \quad \dot{y}_f = v_f \sin(\theta_f), \quad \dot{v}_f = u_a, \quad \dot{\theta}_f = -\frac{v_f}{l} \tan(\delta_f), \quad \dot{\delta}_f = u_s.
\]
where $u_a$ and $u_s$ are respectively the longitudinal acceleration and the steering rate inputs, and $l = 3.476$ m. The leader car plays the role of an “erratic driver” and is modeled with double integrator dynamics along-track and triple integrator dynamics in the lateral direction to mimic continuous steering inputs. The state of the leader’s car is described by $x_l$ (along-track position), $y_l$ (lateral position), $v_{x,l}$ (forward speed), $v_{y,l}$ (lateral speed) and $a_y$ (lateral acceleration). The dynamics are given by:

$$
\begin{align*}
\dot{x} &= v_x, & \dot{v}_x &= w_x, \\
\dot{y} &= v_y, & \dot{v}_y &= a_y, & \dot{a}_y &= w_y,
\end{align*}
$$

where $[w_x, w_y]^T$ is the leader’s control input. We simulate this system in discrete time at $60$ Hz and analyze the data with a time step $\Delta t := 0.1$ s.

In this setting, a disturbance $w[i]$ corresponds to a sequence of control inputs executed by the leader car $w[i] := \{(w_x, w_y)[k] \}_{k=1}^N$ over $N$ time steps. Each disturbance is sampled from a finite set $\mathcal{W} = \{w[1], \ldots, w[L]\}$ with $L = 4$. These “disturbance” realizations correspond to different maneuvers for the leader (doing nothing, accelerating, decelerating, and swapping lanes) and are generated randomly according to the pmf $p = [0.3, 0.3, 0.3, 0.1]$. The disturbance is sampled every $N = 15$ time steps. Thus, the leader car can be viewed as executing a random maneuver every 1.5 seconds. The whole system is described by the state:

$$
\xi := [x_f, y_f, \theta_f, v_f, \delta_f, x_l, v_{x,l}, y_l, v_{y,l}, a_{y,l}]^T.
$$

Participants in the study were informed that their goal was to follow the leader car (described as an “erratic driver”), as closely as possible in the $x$ and $y$ directions, while staying behind the leader and avoiding any collision. The leader car’s four maneuvers were described to participants, along with the fact that these sequences of actions are generated every 1.5 s, independent of (as opposed to an interactive game) the participant’s actions and position.

The experimental protocol for each participant consisted of three phases. The first phase ($\sim 1$ minute) was meant for the participant to familiarize themselves with the simulator. The second and third phases (one minute each) involved the leader car acting according to the model described above (with actions being sampled according to the pmf $p$). The data collected during the second phase was used to train the model and data collected during the third phase was used to test it (the second and third phase disturbance sequences were kept same for all participants).

Note that the pmf $p$ is not shared with the participants. This experimental setting may thus be considered ambiguous. However, since participants are exposed to a training phase where they may build a mental model of disturbances, the setting may also be interpreted as one involving risk.

While the “game” setting is identical to the one introduced in our earlier work in (Majumdar et al., 2017), the use of non-linear dynamics and a realistic driving simulator as opposed to the first-order integrator MATLAB game in (Majumdar et al., 2017) lends the experiment more realism. All data, algorithm, and plotting code is made available at https://github.com/StanfordASL/RSIRL.
5.2 Modeling and Implementation

We model participants’ behavior using the “multiple-branch prepare-react” framework presented in Section 4 with the “prepare” phase starting 0.7 seconds before the leader’s action is sampled. The “react” phase thus extends to 0.8 seconds after the disturbance. This parameter was chosen as being roughly reflective of observed participant behavior during the training phase. We use \( T = 2 \) decision stages to model the participants. Hence, the planning horizon is \( NT = 30 \), which involves planning over two sequences of disturbances in a receding fashion.

We represent our cost function as a linear combination of the following features (with unknown weights):

- \( \phi_1 = 1_{x_{rel} < 2.5} [\log(1 + e^{-r_1(x_{rel} - 2.5)}) - \log(2)] \),
- \( \phi_2 = 1_{x_{rel} > 2.5} [\log(1 + e^{r_2(x_{rel} - 2.5)}) - \log(2)] \),
- \( \phi_3 = \log(1 + e^{r_3 |v_{x, rel}|}) - \log(2) \),
- \( \phi_4 = r_4 \sum_{k=2}^{N} (u_{a,k} - u_{a,k-1})^2 \),
- \( \phi_5 = \log(1 + e^{r_5 |y_{rel}|}) - \log(2) \),
- \( \phi_6 = 1_{y_f > 2} [\log(1 + e^{r_6 (y_f - 2)}) - \log(2)] + 1_{y_f < -2} [\log(1 + e^{-r_6 (y_f + 2)}) - \log(2)] \),

where \( x_{rel}, y_{rel}, \) and \( v_{x, rel} \) are respectively the relative along-track position, lateral position, and along-track velocity between the leader and the follower. Hence, the first feature translates the instruction of staying behind the leader; the second, third, and fifth features penalize the relative distance and velocity between the leader and follower; the fourth feature penalizes change in longitudinal acceleration (effectively jerk of the trajectory); the sixth feature penalizes crossing the road boundaries. We use \( r_1 = 1, r_2 = 0.05, r_3 = 0.1, r_4 = 1.0, r_5 = 0.1, \) and \( r_6 = 0.5 \). These values were chosen to ensure that the costs were well conditioned over the usual range of relative states observed during the experiments.

In order to optimize the model by maximum likelihood estimation as described in Section 4, we discretized the control space \([u_a, u_s]\) to generate the participant action space. For each participant, we ran the K-Means clustering algorithm on the training control inputs, and chose \( K_1 = 15 \) control trajectories for the first decision stage and \( K_2 = 5 \) trajectories for the second stage. Since we model each participant as planning over a receding horizon with two decision stages, it is reasonable to assume that the plan for the second stage is not as fine-grained as over the first. In addition to reducing the computational burden, this concept of mixing coarse-fine planning is a feature also described in (Carton et al., 2016) to model human locomotion. We observe that using 15 control trajectories for the first stage and 5 trajectories for the second stage was sufficient to generate a diverse expert action set in terms of accelerations (Figure 8a, 8c) and steering rates (Figure 8b, 8d); the
span of all $x/y$ traces resulting from the combination of these first and second stage control trajectories is shown in Figure 9.

![Figure 8](image1)

Figure 8: Example of control trajectories computed by K-Means algorithm using 15 centroids ((a), (b)) and 5 centroids ((c), (d)).

![Figure 9](image2)

Figure 9: Span of all possible along-track/lateral ($x/y$) 3 second trajectories encapsulated within a single 2-stage optimization problem with the 15/5 discrete control trajectory space. All lengths in meters.
The polytope $P_r$ (alternatively, the semi-parameterized CRM $\rho^r(\cdot)$) was parametrized with 8 normal vectors $\{a_j\}_{j=1}^8$ corresponding to the positive and negative standard basis vectors in $\mathbb{R}^4$ (i.e., $\{(e_i, -e_i)\}_{i=1}^4$). Our MATLAB implementation uses the parser YALMIP (Löfberg, 2004) and the solver Mosek (ApS).

5.3 Results
Interestingly, our simulated driving scenario was rich enough to elicit a wide variety of qualitative behaviors from the ten participants. In particular, we observed two extreme policies. One extreme involved the driver following the leader very closely with a small separation (Figure 17). Another extreme was to follow the leader with a distance large enough to avoid any collision, often decelerating preemptively to avoid such an event (Figure 11). These two extremes can be interpreted as reflecting varying attitudes towards risk. The first policy corresponds to risk-neutral behavior, where the perceived (as captured by the inferred risk measure) probability of collision is lower than for highly risk-averse participants who were more sensitive to the worst-case eventuality (leader slowing down). We also observed a range of behaviors that lie between these two extremes.

We compare the RS-IRL approach with one where the expert is modeled as minimizing the expected value of his/her cost function computed with respect to the pmf $p$ and with a receding finite planning horizon of two decision stages. Similar to eq. (29), we assume that the risk-neutral stochastic policy is given by the Boltzmann distribution induced by the risk-neutral costs, thereby coinciding with the standard, MaxEntropy IRL model and representing an important benchmark for comparison. We refer to this approach as risk-neutral IRL (RN-IRL). The analog of the recursion equations (26)–(28) for the risk-neutral model are obtained by simply replacing the conditional risk measures with the expected value with respect to the pmf $p$.

Since the expert is assumed to plan his/her decisions every 1.5 s in a receding fashion, we evaluate RS-IRL and RN-IRL predictions based on their errors with respect to each 1.5 s observed demonstration in the test trajectory. In particular, we define the following error metric for predictions in $x_{rel} = x_l - x_f$:

$$\Delta x_{rel,t} := \mathbb{E} \left[ \sqrt{\sum_k (x_{rel,k|t}^{predicted} - x_{rel,k|t}^{human})^2} \right],$$

(36)

where $x_{rel,k|t}^{predicted}$ and $x_{rel,k|t}^{human}$ are, respectively, the predicted and actual $x_{rel}$ trajectories at time $k \in [tN, (t+1)N]$ corresponding to the $t$th demonstrated trajectory. The expectation is taken with respect to the stochastic policy (i.e., Boltzmann distribution) induced by the RS or RN costs (see eq. (29) and (47)). The errors $\Delta y_{rel}$, $\Delta v_x,rel$ and $\Delta v_y,rel$ are computed similarly. As RN-IRL consistently performed better with $T = 2$ decision stages, we only present comparison results between RS-IRL and RN-IRL for $T = 2$. To get a scale for the
values reported in this section, the figure below illustrates the two cars almost colliding ($x_{rel} \approx 2.5$ m) and when they are 5 m apart.

Figure 10: Left: Simulator visual when cars are almost at collision distance ($x_{rel} \approx 2.5$ m); Right: Simulator visual of a participant driving 5 meters behind the leader. Lane-width: 3 m.

5.3.1 Case Study # 1: Risk-Averse Participant

Figure 11 plots the $x_{rel}$ trajectory (normalized by car length $\approx 4.2$ m) for a highly risk-averse participant during the third (test) phase. On average, the along-track relative distance is quite large ($\approx 6$ car-lengths). The expected prediction errors $\Delta x_{rel,t}$ (normalized by car-length) from RS-IRL and RN-IRL for each of the 51 1.5 s demonstrations comprising the test phase are plotted in Figure 12a as absolute errors, and in Figure 12b as percentage differences with positive values indicating an improvement of RS-IRL over RN-IRL. A similar error plot is shown in Figure 12c and 12d for along-track velocity error $\Delta v_{x,rel,t}$.

Figure 11: Full $x_{rel}$ (longitudinal distance) trajectory (normalized by car length) for a highly risk-averse participant. On average, the relative distance is quite large ($\approx 6$ car-lengths). The boxed sections are discussed in further detail below.
(a) Expected (w.r.t. stochastic policy) prediction errors $\Delta x_{rel,t}$ from RS-IRL and RN-IRL for each 1.5s trajectory segment.

(b) Percentage improvement in $\Delta x_{rel,t}$ for the RS-IRL model over RN-IRL for each 1.5s trajectory segment.

(c) Expected (w.r.t. stochastic policy) prediction errors $\Delta v_{x,rel,t}$ from RS-IRL and RN-IRL for each 1.5s trajectory segment.

(d) Percentage improvement in $\Delta v_{x,rel,t}$ for the RS-IRL model over RN-IRL for each 1.5s trajectory segment.

Figure 12: Comparison of the $\Delta x_{rel,t}$ and $\Delta v_{x,rel,t}$ prediction errors (normalized by car length) for the RS-IRL and RN-IRL models for a highly risk-averse participant. The RS-IRL model almost always outperforms RN-IRL, on average providing about 22% improvement.

Figure 13: Inferred risk envelope for risk-averse participant. Notice the overweighting (and ambiguity therein) of probabilities associated with the leader’s deceleration maneuver and the underweighting of the leader’s acceleration. The wire-frame pyramid is the projection of the probability simplex $\Delta^4$ onto the first three dimensions.
It can be observed that the RS-IRL model almost always outperforms RN-IRL, on average providing an improvement of about 22%. Notice however, the four prominent negative peaks in Figure 12b corresponding to the RN-IRL model outperforming RS-IRL on these instances. All four peaks can be explained by considering the inferred risk envelope $P_r$, whose projections are plotted in Figures 13a and 13b. Notice how the participant’s polytope is significantly biased towards high probabilities in the third dimension (corresponding to the decelerate maneuver for the leader), and furthermore, spans a wide range in probabilities ($\approx [0.5, 0.85]$) for this event. Thus, not only does the participant overweight the true probability of this outcome (0.3), he/she is also ambiguous about the true probability. On the other hand, the envelope suggests a very low and narrow range of probabilities for the leader’s accelerate maneuver. These two properties are what result in the four negative peaks in Figure 12b.

Consider the first negative peak (corresponding to Sequence 1 in Figure 11). At this decision stage, the true distance decreases due to the participant accelerating. As the learned RS-IRL model assumes significant ambiguity in the leader’s deceleration yet small probability in acceleration, the most probable trajectories from the RS-IRL model prefer weaker acceleration. Thus, this particular artifact may be attributed to a slight overfitting of the parameterized polytope offsets $r$. This overfitting manifests itself again at the very end (Sequence 4) where the cars are far apart (6 car-lengths) and the RS-IRL model predicts a weaker deceleration than that observed. Essentially the high bias associated with the leader’s deceleration as encoded in the inferred polytope along with the already large separation between the cars (recall that the second feature penalizes large separations) leads to a less aggressive deceleration prediction. The remaining peaks (Sequence 2) are shown in detail in Figure 14b. At both these decision stages, the RS-IRL model preempt, by about one decision stage, a deceleration maneuver for the participant. Again this is a result of the deceleration ambiguity implied by the RS-IRL model and suggests that additional training data is needed to further refine (decrease) this ambiguity. This would naturally also alleviate the prediction errors in Sequences 1 and 4.

Notice however that all of the spurious predictions by the RS-IRL model as discussed above occur when the cars are quite far apart, on the order of 5–6 car-lengths. In contrast, when the two cars are quite close (e.g., less than 4 car-lengths during Sequence 3), there is now a significant collision risk in the event that the leader decelerates. Figure 14a illustrates the RS-IRL model correctly predicting the participant braking to increase the relative separation, and with high probability ($\approx 0.89$). In contrast, the RN-IRL induced stochastic policy is not only of high entropy but also suggests quite absurd trajectories that lead to a further decrease in the relative separation. Thus, these results suggest that when RS-IRL underperforms RN-IRL, it does so at non-critical stages characterized by low risk.
Figure 14: Comparisons of the most-probable (under the stochastic Boltzmann policy) RS-IRL and RN-IRL trajectory predictions in $x_{rel}$ as compared with the true data.

5.3.2 Case Study # 2: Risk/Ambiguity-Averse Participant

We next discuss a participant for whom RS-IRL again significantly outperformed RN-IRL, however, due to a different manifestation of risk. Consider the inferred risk envelopes in Figure 15.

![Projection of polytope along the {nothing, accelerate, decelerate} “dimensions”](image1.png)

![Projection of polytope along the {accelerate, decelerate} “dimensions”](image2.png)

Figure 15: Inferred risk envelope for ambiguity-averse participant. Notice the drastic underweighting of probabilities associated with leader’s deceleration and significant ambiguity regarding leader’s acceleration.

Notice that the participant places very low probability on deceleration yet is ambiguous regarding the leader’s acceleration (i.e., a polar opposite of the first participant). This ambiguity led to several preemptive braking maneuvers in an attempt to maintain a “safe” distance to the leader car. Figures 16a–16d illustrate the consistency of RS-IRL’s improvement over RN-IRL in predicting these maneuvers over the entire test trajectory.
(a) Expected (w.r.t. stochastic policy) prediction errors $\Delta x_{rel,t}$ from RS-IRL and RN-IRL for each 1.5s trajectory segment.

(b) Percentage improvement in $\Delta x_{rel,t}$ for the RS-IRL model over RN-IRL for each 1.5s trajectory segment.

(c) Expected (w.r.t. stochastic policy) prediction errors $\Delta v_{x,rel,t}$ from RS-IRL and RN-IRL for each 1.5s trajectory segment.

(d) Percentage improvement in $\Delta v_{x,rel,t}$ for the RS-IRL model over RN-IRL for each 1.5s trajectory segment.

Figure 16: Comparison of the $\Delta x_{rel,t}$ and $\Delta v_{x,rel,t}$ prediction errors (normalized by car length) for the RS-IRL and RN-IRL models for an ambiguity-averse participant. The RS-IRL model almost always outperforms RN-IRL, on average providing about 13–14% improvement.

5.3.3 Case Study # 3: Risk-Neutral Participant

Finally, we consider a participant for whom both RS-IRL and RN-IRL performed on par with each other. Figure 17 plots the $x_{rel}$ trajectory while Figure 18 illustrates the inferred risk envelope.

Figure 17: Full $x_{rel}$ trajectory (normalized by car length) for a risk-neutral participant. On average, the relative distance is noticeably smaller, on the order of 3 car-lengths.
The inferred risk envelope features no bias along any dimension nor any appreciable level of ambiguity; thereby suggesting a risk-neutral profile for this participant. Furthermore, notice in Figure 17 that the participant stays quite a bit closer to the leader car than the first participant, a clear indicator of his/her risk-neutral stance. Figures 19 and 20 confirm the two models performing on par with each other. This, however, is to be expected for a strongly risk-neutral participant since the RS-IRL model subsumes the RN-IRL model.

5.3.4 Limitations of Cost Shaping

In this section we argue that both the cost weights $c$ and the risk measure $\rho(\cdot)$ are necessary to reasonably approximate diverse risk-sensitive behaviors. In Figure 21, we plot the cost stemming from the weighted combination of features $\phi_1, \phi_2$ using the optimal learned weights from RS-IRL and RN-IRL for the first two participants considered above, and
(a) Expected (w.r.t. stochastic policy) prediction errors \( \Delta v_{x,rel,t} \) from RS-IRL and RN-IRL for each 1.5s trajectory segment.

(b) Percentage improvement in \( \Delta v_{x,rel,t} \) for the RS-IRL model over RN-IRL for each 1.5s trajectory segment.

Figure 20: Comparison of the \( \Delta v_{x,rel,t} \) prediction errors (normalized by car length) for the RS-IRL and RN-IRL models for a risk-neutral participant. The two models perform on par with each other.

Additionally include the cost weights for features \( \phi_3, \phi_4 \). These are the four features that dominate the along-track behavior of the participants.

(a) Case-Study 1 participant optimal cost weights: \( (c_3, c_4) = (0.0993, 0.2352) \) (RS-IRL); \( (c_3, c_4) = (0.1052, 0.2330) \) (RN-IRL).

(b) Case-Study 2 participant optimal cost weights: \( (c_3, c_4) = (0.1864, 0.0562) \) (RS-IRL); \( (c_3, c_4) = (0.3170, 0.0624) \) (RN-IRL).

Figure 21: Comparison of the cost weights from RS-IRL and RN-IRL for case study participants 1 and 2.

Notice that in spite of the extremely similar cost weights for the first participant, the improvement in performance using RS-IRL is substantial. For the second participant, a difference in the cost weights and the use of a risk measure were needed to obtain the observed performance boost with RS-IRL. This clearly establishes the benefits of risk sensitive inference, particularly highlighting the deficiency of only using cost shaping (in general, risk-neutral algorithms) due to its inability to cope with more nuanced manifestations of uncertainty in decision-making.

5.3.5 Summary

Table 1 presents the average (over the 51 1.5 s demonstrations in the test trajectory) percentage improvement (RN-IRL to RS-IRL) in the prediction errors, i.e., \( \Delta x_{rel}, \Delta y_{rel} \).
As expected, the RS-IRL predictions are almost always better than those by RN-IRL, with as much as 22.0% improvement in $x_{rel}$ and 23.1% improvement in $v_{x,rel}$. Regarding errors in $y_{rel}$ and $v_{y,rel}$, RS-IRL and RN-IRL perform comparably (absolute errors were in the range 0.19 – 0.42 m) since the primary source of risk-aversion stems from the leader’s acceleration and deceleration rather than its lateral motion. In the cases with noticeable improvement, either in position or velocity (i.e., participants # 1, 4, 6, 10), the better predictions were a result of the RS-IRL model more accurately representing participants with higher levels of risk- and/or ambiguity-aversion. Indeed the first two detailed case-studies presented earlier correspond to participants # 1 and 6. In contrast, for more risk-neutral participants (e.g., the last presented case-study corresponding to participant # 7), the performance improvements were either less pronounced or the two models performed comparably.

| Participant # | 1    | 2    | 3    | 4    | 5    | 6    | 7    | 8    | 9    | 10   |
|---------------|------|------|------|------|------|------|------|------|------|------|
| $\Delta x_{rel}$ (T=2) | 22.0 | 10.1 | 3.9  | 14.9 | 7.5  | 13.3 | 2.6  | 9.7  | 3.3  | 18.6 |
| $\Delta y_{rel}$ (T=2) | 22.2 | 21.4 | 8.4  | 9.5  | 10.7 | 4.8  | 12.0 | 7.3  | -2.7 | 16.3 |
| $\Delta v_{x,rel}$ (T=2) | 23.1 | 10.4 | 8.1  | 13.0 | 12.7 | 14.3 | -0.3 | 10.5 | 5.0  | 17.1 |
| $\Delta v_{y,rel}$ (T=2) | 22.9 | 22.5 | 9.1  | 6.0  | 8.7  | 0.2  | 13.7 | 3.0  | -3.0 | 14.9 |

Table 1: Average percentage improvement (over 51 demonstrations in the test trajectory) in prediction errors for RS-IRL over RN-IRL. The RS-IRL predictions with $T = 2$ for $x_{rel}$ and $v_{x,rel}$ are more accurate than those for RN-IRL for all but one participant, with as much as 23.1% average improvement. The most pronounced improvements (highlighted in red) corresponded to participants with higher levels of risk- and/or ambiguity-aversion, some of whom are studied in detail in the case-studies. The improvements in the lateral direction are less significant since the primary source of risk and ambiguity was in the longitudinal dynamics.

### 6 Discussion and Conclusions

We have presented an approach for IRL that explicitly accounts for risk and ambiguity sensitivity in experts. We proposed a flexible modeling framework based on coherent risk measures that allows us to capture an entire spectrum of risk assessments from risk-neutral to worst-case for a rich class of static and dynamic decision-making settings. We developed efficient LP based non-parametric algorithms for static, and likelihood based semi-parametric algorithms for dynamic decision making settings. Notably, we significantly improved the modeling framework in (Majumdar et al., 2017) for dynamic decision making, losing little as we transitioned from the exact LP iteration to semi-parametric likelihood based algorithms. The proposed inference framework was rigorously evaluated on a realistic simulated driving game with ten participants and shown to be able to infer and mimic qualitatively different driving styles ranging from risk-neutral to highly risk-averse in a data-efficient manner, while more accurately capturing participant behavior than a risk-neutral model. Most importantly, by performing inference using the dual representation of
coherent risk measures, we retain the generality to be able to recover any risk functional within this rich class of risk measures, without assuming any a priori knowledge (e.g., fixed dis-utility and/or risk measure).

**Challenges and future work:** We propose three key areas for further research.  
*Modeling:* As in the majority of IRL literature, we hand-picked features for the driving game. While performance on the test trajectory given $\sim 1$ minute of training data supported these choice of features, incorporating risk-sensitivity in large-scale IRL algorithms requires automatic feature extraction. There has been some recent work on using deep neural nets in the MaxEnt IRL framework (Wulfmeier et al., 2015) as well as using general non-linear cost representations (Finn et al., 2016). A promising area of future research then is to embed the semi-parametric approach within deep cost networks to yield RS-IRL algorithms for high-dimensional systems.  
*Interaction:* The inference framework described assumes that the human expert is subject to an independent (non-interactive) source of disturbance. The natural extension therefore is to modularize the entire risk-sensitive IRL algorithm within a game-theoretic interactive setting involving multiple human agents and robotic systems. The key challenge here is to efficiently balance offline and online learning (see e.g., (Sadigh et al., 2016b; Waugh et al., 2011)), to enable the autonomous robot to actively infer intent and risk-sensitive preferences for the human agents, and use the resulting information to consequently influence the human agents.  
*Experimental Validation:* As a direct extension of the driving game and the game-theoretic adaptation of this work, we plan on testing our algorithms on an autonomous car testbed.  

We believe that the approach described here along with the indicated future directions represent an important step towards endowing our robotic systems with the ability to predict, infer, and mimic risk-sensitive behavior, which is crucial for safety-critical applications where humans and robots interact.

**Acknowledgments**

The authors were partially supported by the Office of Naval Research, Science of Autonomy Program, under Contract N00014-15-1-2673, and by the Toyota Research Institute (“TRI”). This article solely reflects the opinions and conclusions of its authors and not ONR, TRI or any other Toyota entity.  

The authors would also like to acknowledge the contributions of Ajay Mandlekar towards the conference version of this paper presented at RSS 2017.

**References**

Abbeel P and Ng AY (2004) Apprenticeship learning via inverse reinforcement learning.  
In: *Int. Conf. on Machine Learning.*

36
Abbeel P and Ng AY (2005) Exploration and apprenticeship learning in reinforcement learning. In: Int. Conf. on Machine Learning.

Acerbi C (2002) Spectral measures of risk: A coherent representation of subjective risk aversion. Journal of Banking & Finance 26(7): 1505–1518.

Acerbi C and Tasche D (2002) On the coherence of expected shortfall. Journal of Banking & Finance 26(7): 1487–1503.

Allais M (1953) Le comportement de l’homme rationnel devant le risque: critique des postulats et axiomes de l’école américaine. Econometrica 21(4): 503–546.

ApS M (?????) MOSEK optimization software. Available at https://mosek.com/.

Artzner P, Delbaen F, Eber JM and Heath D (1999) Coherent measures of risk. Mathematical Finance 9(3): 203–228.

Axelrod A, Carlone L, Chowdhary G and Karaman S (2016) Data-driven prediction of EVAR with confidence in time-varying datasets. In: IEEE Conf. on Decision and Control.

Barberis NC (2013) Thirty years of prospect theory in economics: A review and assessment. Journal of Economic Perspectives 27(1): 173–195.

Bäuerle N and Ott J (2011) Markov decision processes with average-value-at-risk criteria. Mathematics of Operations Research 74(3): 361–379.

Burke JV, Lewis AS and Overton ML (2005) A robust gradient sampling algorithm for nonsmooth nonconvex optimization. SIAM Journal on Optimization 15(3): 751–779.

Carton D, Nitsch V, Meinzer D and Wollherr D (2016) Towards assessing the human trajectory planning horizon. PLoS ONE 11(12): e0167021.

Chen X (2012) Smoothing methods for nonsmooth, nonconvex minimization. Mathematical Programming 134(1): 71–99.

Chow Y and Pavone M (2014) A framework for time-consistent, risk-averse model predictive control: Theory and algorithms. In: American Control Conference.

Chow Y, Tamar A, Mannor S and Pavone M (2015) Risk-sensitive and robust decision-making: a CVaR optimization approach. In: Advances in Neural Information Processing Systems.

Eichhorn A and Römisch W (2005) Polyhedral risk measures in stochastic programming. SIAM Journal on Optimization 16(1): 69–95.
Ellsberg D (1961) Risk, ambiguity, and the savage axioms. *The Quarterly Journal of Economics* 75(4): 643–669.

Englert P and Toussaint M (2015) Inverse KKT learning cost functions of manipulation tasks from demonstrations. In: *Int. Symp. on Robotics Research*.

Filar JA, Kallenberg LCM and Lee HM (1989) Variance-penalized Markov decision processes. *Mathematics of Operations Research* 14(1): 147–161.

Finn C, Levine S and Abbeel P (2016) Guided cost learning: Deep inverse optimal control via policy optimization. In: *Int. Conf. on Machine Learning*.

Geibel P and Wysotzki F (2005) Risk-sensitive reinforcement learning applied to control under constraints. *Journal of Artificial Intelligence Research* 24(1): 81–108.

Gilboa I and Marinacci M (2016) Ambiguity and the Bayesian paradigm. In: *Readings in Formal Epistemology*, first edition, chapter 21.

Gilboa I and Schmeidler D (1989) Maximin expected utility with non-unique prior. *Journal of Mathematical Economics* 18(2): 141–153.

Glimcher P and Fehr E (2014) *Neuroeconomics*. Second edition. Elsevier.

Gul F and Pesendorfer W (2014) Expected uncertain utility theory. *Econometrica* 82(1): 1–39.

Hey JD, Lotito G and Maffioletti A (2010) The descriptive and predictive adequacy of theories of decision making under uncertainty/ambiguity. *Journal of Risk and Uncertainty* 41(2): 81–111.

Howard R and Matheson J (1972) Risk-sensitive Markov decision processes. *Management Science* 8(7): 356–369.

Hsu M, Bhatt M, Adolphs R, Tranel D and Camerer CF (2005) Neural systems responding to degrees of uncertainty in human decision-making. *Science* 310(5754): 1680–1683.

Kahneman D and Tversky A (1979) Prospect theory: An analysis of decision under risk. *Econometrica* : 263–291.

Kolter JZ, Abbeel P and Ng AY (2007) Hierarchical apprenticeship learning with application to quadruped locomotion. In: *Advances in Neural Information Processing Systems*.

Kretzschmar H, Spies M, Sprunk C and Burgard W (2016) Socially compliant mobile robot navigation via inverse reinforcement learning. *Int. Journal of Robotics Research* 35(11): 1289–1307.
Kuderer M, Gulati S and Burgard W (2015) Learning driving styles for autonomous vehicles from demonstration. In: Proc. IEEE Conf. on Robotics and Automation.

Lanza A, Morigi S, Selesnick I and Sgallari F (2017) Nonconvex nonsmooth optimization via convex nonconvex majorization minimization. Numerische Mathematik 136(2): 343–381.

Levine S and Koltun V (2012) Continuous inverse optimal control with locally optimal examples. In: Int. Conf. on Machine Learning.

Löfberg J (2004) YALMIP : A toolbox for modeling and optimization in MATLAB. In: IEEE Int. Symp. on Computer Aided Control Systems Design.

Majumdar A and Pavone M (2017) How should a robot assess risk? Towards an axiomatic theory of risk in robotics. In: Int. Symp. on Robotics Research. In Press.

Majumdar A, Singh S, Mandlekar A and Pavone M (2017) Risk-sensitive inverse reinforcement learning via coherent risk models. In: Robotics: Science and Systems.

Mihatsch O and Neuneier R (2002) Risk-sensitive reinforcement learning. Machine Learning 49(2): 267–290.

Mombaur K, Truong A and Laumond JP (2010) From human to humanoid locomotion—an inverse optimal control approach. Autonomous Robots 28(3): 369–383.

Ng A and Russell S (2000) Algorithms for inverse reinforcement learning. In: Int. Conf. on Machine Learning.

Nilim A and El Ghaoui L (2005) Robust control of Markov decision processes with uncertain transition matrices. Operations Research 53(5): 780–798.

Osogami T (2012) Robustness and risk-sensitivity in Markov decision processes. In: Advances in Neural Information Processing Systems.

Park T and Levine S (2013) Inverse optimal control for humanoid locomotion. In: Robotics: Science and Systems Workshop on Inverse Optimal Control and Robotic Learning from Demonstration.

Petrik M and Subramanian D (2012) An approximate solution method for large risk-averse Markov decision processes. In: Proc. Conf. on Uncertainty in Artificial Intelligence.

Prashanth LA, Jie C, Fu M, Marcus S and Szepesvári C (2016) Cumulative prospect theory meets reinforcement learning: Prediction and control. In: Int. Conf. on Machine Learning.
Quiggin J (1982) A theory of anticipated utility. *Journal of Economic Behavior & Organization* 3(4): 323–343.

Rabin M (2000) Risk aversion and expected-utility theory: A calibration theorem. *Econometrica* 68(5): 1281–1292.

Ramachandran D and Amir E (2007) Bayesian inverse reinforcement learning. In: *International Joint Conference on Artificial Intelligence*.

Ratliff LJ and Mazumdar E (2017) Risk-sensitive inverse reinforcement learning via gradient methods. Available at: [https://arxiv.org/abs/1703.09842](https://arxiv.org/abs/1703.09842).

Rockafellar RT (2007) Coherent approaches to risk in optimization under uncertainty. In: *OR Tools and Applications: Glimpses of Future Technologies*, chapter 3. INFORMS.

Rockafellar RT and Uryasev S (2000) Optimization of conditional value-at-risk. *Journal of Risk* 2: 21–41.

Russell S (1998) Learning agents for uncertain environments. In: *Proc. Computational Learning Theory*.

Ruszczyński A (2010) Risk-averse dynamic programming for Markov decision process. *Mathematical Programming* 125(2): 235–261.

Sadigh D, Sastry S, Seshia SA and Dragan AD (2016a) Planning for autonomous cars that leverage effects on human actions. In: *Robotics: Science and Systems*.

Sadigh D, Sastry SS, Seshia SA and Dragan A (2016b) Information gathering actions over human internal state. In: *IEEE/RSJ Int. Conf. on Intelligent Robots & Systems*.

Shapiro A (2009) On a time consistency concept in risk averse multi-stage stochastic programming. *Operations Research Letters* 37(3): 143–147.

Shen Y, Tobia MJ, Sommer T and Obermayer K (2014) Risk-sensitive reinforcement learning. *Neural Computation* 26(7): 1298–1328.

Tamar A, Chow Y, Ghavamzadeh M and Mannor S (2016) Sequential decision making with coherent risk. *IEEE Transactions on Automatic Control* 62(7): 3323–3338.

Tamar A, Di Castro D and Mannor S (2012) Policy gradients with variance related risk criteria. In: *Int. Conf. on Machine Learning*.

VIRES Simulationstechnologie GmbH (????) VTD - Virtual Test Drive. Available at [https://vires.com/vtd-vires-virtual-test-drive/](https://vires.com/vtd-vires-virtual-test-drive/).

von Neumann J and Morgenstern O (1944) *Theory of Games and Economic Behavior*. Princeton University Press.
Waugh K, Ziebart BD and Bagnell JA (2011) Computational rationalization: The inverse equilibrium problem. In: Int. Conf. on Machine Learning.

Wu C and Yuanlie L (1999) Minimizing risk models in markov decision process with policies depending on target values. Journal of Mathematical Analysis and Applications 231(1): 47–67.

Wulfmeier M, Ondruska P and Posner I (2015) Maximum entropy deep inverse reinforcement learning. Available at: https://arxiv.org/abs/1507.04888.

Xu H and Mannor S (2010) Distributionally robust Markov decision processes. In: Advances in Neural Information Processing Systems.

Yaari ME (1987) The dual theory of choice under risk. Econometrica 55(1): 95–115.

Ziebart BD, Maas A, Bagnell JA and Dey AK (2008) Maximum entropy inverse reinforcement learning. In: Proc. AAAI Conf. on Artificial Intelligence.

Ziebart BD, Ratliff N, Gallagher G, Mertz C, Peterson K, Bagnell JA, Hebert M, Key AK and Srinivasa S (2009) Planning-based prediction for pedestrians. In: IEEE/RSJ Int. Conf. on Intelligent Robots & Systems.

Zucker M, Bagnell JA, Atkeson CG and Kuffner J (2010) An optimization approach to rough terrain locomotion. In: Proc. IEEE Conf. on Robotics and Automation.

Appendix A Theoretical Guarantees

A.1 Proof of Theorem 3

In order to prove the algorithm’s consistency, we need to establish intermediate results. Since $\mathcal{P}_\infty$ is an intersection of convex (respectively compact) sets, it is also convex (respectively compact). Moreover, since each $\mathcal{P}_D$ contains the expert’s polytope $\mathcal{P}$, then $\mathcal{P}_\infty$ is also an outer approximation of $\mathcal{P}$. In particular, $\mathcal{P}_\infty$ is not empty.

Denote with $d_H$ the Hausdorff distance between subsets of $\mathbb{R}^L$ associated with the Euclidean norm $\| \cdot \|$, i.e: for two subsets $A$ and $B$ of $\mathbb{R}^L$,

$$d_H(A, B) = \max \left\{ \sup_{b \in B} \inf_{a \in A} \|a - b\|, \sup_{a \in A} \inf_{b \in B} \|a - b\| \right\}.$$  \hspace{1cm} (37)

The Hausdorff distance defines a metric on the set of non-empty compact subsets of $\mathbb{R}^L$, that we use to measure the distance between risk envelopes. The sequence $\mathcal{P}_d$ is non-increasing (for set inclusion). Therefore, $(d_H(\mathcal{P}_d, \mathcal{P}_\infty))_{d \geq 1}$ is a non-increasing sequence of non-negative real numbers. Thus, its limit exists and:

$$\lim_{d \to \infty} d_H(\mathcal{P}_d, \mathcal{P}_\infty) = 0$$ \hspace{1cm} (38)
Lemma 4. Consider a sequence \( (\beta_n)_{n \geq 1} \) of compact convex subsets of \( \Delta^L \) such that for all \( n \geq 1 \), \( P_\infty \subseteq \beta_n \) and \( \lim_{n \to \infty} d_H(\beta_n, P_\infty) = 0 \). Consider also a sequence of states \( (x_n)_{n \geq 1} \) such that \( x_n \to x^* \) when \( n \to \infty \). Define the functions \( \varphi_n(u) := \max_{v \in \beta_n} v^T g(x_n, u) \), \( \varphi_{n, \infty}(u) := \max_{v \in P_\infty} v^T g(x_n, u) \) and \( \varphi(u) := \max_{v \in P_\infty} v^T g(x^*, u) \). Then, for any compact \( K \subseteq U \):

\[
\lim_{n \to \infty} \sup_{u \in K} |\varphi_n(u) - \varphi(u)| = 0. \tag{39}
\]

Proof. Fix \( u \in U \). For any \( n \geq 1 \), denote with \( v_n \in \beta_n \) a point such that \( \varphi_n(u) = v_n^T g(x_n, u) \) and \( v_{n, \infty} \in P_\infty \) a point such that \( \varphi_{n, \infty}(u) = v_{n, \infty}^T g(x_n, u) \). Let \( \Gamma_\infty \) be the projection operator onto the compact convex set \( P_\infty \). Then,

\[
\Gamma_\infty(v_n)^T g(x_n, u) \leq v_{n, \infty}^T g(x_n, u) \leq v_n^T g(x_n, u). \tag{40}
\]

The second inequality in (40) results from the fact that \( P_\infty \subseteq \beta_n \). By the Cauchy-Schwarz inequality,

\[
| (\Gamma_\infty(v_n) - v_n)^T g(x_n, u) | \leq \| \Gamma_\infty(v_n) - v_n \|_2 \| g(x_n, u) \|_2. \tag{41}
\]

Since \( \| \Gamma_\infty(v_n) - v_n \|_2 \leq d_H(\beta_n, P_\infty) \) and \( \lim_{n \to \infty} x_n = x^* \), we get that the left-hand side (LHS) in (41) tends to 0 when \( n \to \infty \), which implies that

\[
\lim_{n \to \infty} \varphi_n(u) - \varphi_{n, \infty}(u) = 0. \tag{42}
\]

Since \( g \) is convex with respect to \( x \), we get that \( x \mapsto \max_{v \in P_\infty} v^T g(x, u) \) is convex and, hence, continuous with respect to \( x \). Therefore, \( \lim_{n \to \infty} \varphi_{n, \infty}(u) = \varphi(u) \). Using (42), \( \lim_{n \to \infty} \varphi_n(u) = \varphi(u) \). But, by Theorem 10.8 in (Rockafellar, 2007), pointwise convergence of a sequence of convex functions over \( U \) implies uniform convergence over any compact set \( K \subseteq U \), which is the desired result.

Since we assumed the cost functions to be strictly convex with respect to \( u \), we get that for any convex compact subset \( B \subseteq \Delta^L \), \( (x, u) \mapsto \max_{v \in B} v^T g(x, u) \) is strictly convex with respect to \( u \).

Lemma 5. Define the sequence of functions \( \varphi_n \) and \( \varphi \) as in Lemma 4. Denote \( u_n := \arg\min_u \varphi_n(u) \) and \( u^* := \arg\min_u \varphi(u) \) (each of these minima are unique by strict convexity). Then:

\[
\lim_{n \to \infty} u_n = u^*. \tag{43}
\]

Proof. Observe first that the sequence \( (u_n)_{n \geq 1} \) is bounded. Indeed, for any \( v \in P_\infty \), we have:

\[
v^T g(x_n, u_n) \leq \varphi_n(u_n) \leq \varphi_n(u^*). \tag{44}
\]

Since \( (x_n)_{n \geq 1} \) is bounded, by pointwise convergence of \( (\varphi_n)_{n \geq 1} \) and by the bounded level sets’ assumption on the cost functions, \( (u_n)_{n \geq 1} \) must be bounded. Otherwise, the LHS in (44) goes to \( +\infty \) whereas the RHS goes to \( \varphi(u^*) \).
We proceed by contradiction to prove (43). Assume that \((u_n)_{n \geq 1}\) does not converge to \(u^*\). Without loss of generality, assume that there exists some \(\eta > 0\) such that for all \(n\), \(\|u_n - u^*\| \geq \eta\). Define \(u'_n\) and \(\alpha_n \in [0, 1]\) such that \(u'_n = \alpha_n u_n + (1 - \alpha_n) u^*\), and \(\|u^* - u'_n\|_2 = \frac{\eta}{2}\). By convexity of \(\varphi_n\):

\[
\varphi_n(u'_n) \leq \alpha_n \varphi_n(u_n) + (1 - \alpha_n) \varphi_n(u^*). \tag{45}
\]

Using the uniform convergence property of \((\varphi_n)_{n \geq 1}\) and the fact that \((u_n)_{n \geq 1}\) is bounded, one gets that the RHS in (45) goes to \(\varphi(u^*)\). For the LHS, we consider a converging subsequence of \((u'_n)_{n \geq 1}\). Denoting with \(u'\) its limit, then \(\varphi(u') \leq \varphi(u^*)\). But \(\|u' - u^*\| = \frac{\eta}{2}\) and by uniqueness of the minimum, this is a contradiction. 

Lemma 5 gives the convergence of the optimal action of a sequence of outer approximations that converges to some risk envelope. We take advantage of that fact to prove Theorem 3.

**Proof.** (Theorem 3). Fix any \(x^* \in S\) and choose a subsequence of \((P_d)_{d \geq 1}\) (that we still denote \((P_d)\) for simplicity) such that \(x_d \to x^*\). For any \(d \geq 1\) and the corresponding demonstration \((x_d, u(P, x_d))\), according to the update of the outer approximation \(P_d\) in Algorithm 1,

\[
u(P_d, x_d) = u(P, x_d). \tag{46}\]

From Lemma 5, we have that \(\lim_{d \to \infty} u(P_d, x_d) = u(P_\infty, x^*)\). From equation (46), we get that \(\lim_{d \to \infty} u(P_d, x_d) = u(P, x^*)\). Combining the two previous observations, we get the desired result, i.e: \(u(P, x^*) = u(P_\infty, x^*)\).

**Appendix B  Derivation of Prepare-React Policy Likelihood**

Recall the multi-stage optimization problem objective, repeated here for convenience:

\[
C_{0:N-n_d-1} + \rho_0 \left( C_{N-n_d:N-1} + C_{N:2N-n_d-1} + \rho_1 \left( C_{2N-n_d:2N-1} + \cdots + \rho_{T-1} \left( C_{TN-n_d:TN-1} \right) \cdots \right) \right).
\]

For a “prepare” – “react” policy \(\hat{\pi}_t\) at stage \(t\), let \(\hat{\pi}_{t|p}\) denote the “prepare” portion and \(\hat{\pi}_{t|r}\) denote the “react” portion. Then, we can re-write the objective above to show explicit dependence as follows:

\[
C_{0:N-n_d-1}(\cdot, \hat{\pi}_{0|p}) + \rho_0 \left( C_{N-n_d:N-1}(\cdot, \hat{\pi}_{0|r}) + C_{N:2N-n_d-1}(\cdot, \hat{\pi}_{1|p}) + \rho_1 \left( C_{2N-n_d:2N-1}(\cdot, \hat{\pi}_{1|r}) + \cdots + \rho_{T-1} \left( C_{TN-n_d:TN-1}(\cdot, \hat{\pi}_{T-1|r}) \right) \cdots \right) \right).
\]
Note that the expression within the large brackets is a random variable in $\mathbb{R}^L$, indexed by all possible realizations of $w'_0$, and a function of the first “prepare” sequence $\hat{\pi}_{0|p}$. Thus, for a given “prepare” sequence $\hat{\pi}_{0|p}$, define the tail cost

$$
\tau[w'_0, \hat{\pi}_{0|p}](\hat{\pi}_{0|p}) := C_{N-n_d:L-1}(\cdot, \hat{\pi}_{0|t}) + \min_{\hat{\pi}_t} \rho_1(C_{N:2N-1}(\cdot, \hat{\pi}_1) + \cdots + \rho_{T-1}(C_{(T-1):TN-1}(\cdot, \hat{\pi}_{T-1})).
$$

Then, we define the conditional distribution for the “react” sequence corresponding to disturbance mode $w'_0$, given the first “prepare” sequence, as

$$
\Pr(\hat{\pi}_{0|t} | \hat{\pi}_{0|p} ; w'_0) \propto \exp\left(-\tau[w'_0, \hat{\pi}_{0|p}](\hat{\pi}_{0|t})\right).
$$

The distribution for the first “prepare” sequence then is given as

$$
\Pr(\hat{\pi}_{0|p}) \propto \exp\left(-C_{0:L-1}(\cdot, \hat{\pi}_0|p) + \rho^r\left(\text{softmin}_{\hat{\pi}_{0|t}} \tau[w'_0, \hat{\pi}_{0|p}](\hat{\pi}_{0|t})\right)\right),
$$

where we use softmin in place of min to ensure differentiability. Thus, for an observed “prepare” – “react” sequence $\hat{\pi}^*_0$ associated with the observed disturbance mode $w^*_0$, we obtain

$$
\Pr(\hat{\pi}^*_0) = \Pr(\hat{\pi}^*_{0|p}) \cdot \Pr(\hat{\pi}^*_0 | \hat{\pi}^*_{0|p} ; w^*_0).
$$

### Appendix C  Likelihood Gradient Computations

Define

$$
\sigma[w_{-1|tN}](\hat{u}) := \frac{\exp\left(-\beta \tilde{\tau}[w_{-1|tN}](\hat{u})\right)}{\sum_{\hat{u}} \exp\left(-\beta \tilde{\tau}[w_{-1|tN}](\hat{u}')\right)}
$$

(47)

to be the probability of choosing action trajectory $\hat{u}$ at time-step $tN$ as assumed by the Boltzmann likelihood model in (29). Then, the gradient of the log-likelihood in (31) with respect to parameter $s \in \{r, c\}$ is given by

$$
\beta \left[ \frac{1}{T^*} \sum_{\hat{u}_t^* \in T^*} \left[ \sum_{\hat{u}_t \neq \hat{u}_t^*} \sigma[w_{-1|tN}](\hat{u}) \nabla_s \tilde{\tau}[w_{-1|tN}](\hat{u}) + \left( \sigma[w_{-1|tN}](\hat{u}_t^*) - 1 \right) \nabla_s \tilde{\tau}[w_{-1|tN}](\hat{u}_t^*) \right] \right].
$$

(48)

From equations (26), (27), and (28), we see that the derivative of $\tilde{\tau}[w_{-1|tN}](\hat{u})$ can be computed through a recursive implementation of the chain rule, starting from the terminal stage. In the event that all nested LPs are non-degenerate, we obtain the following recursive set of equations for computing the gradients.
From eq. (28), \( \tilde{\tau}[w^*_{-1|tN}](\hat{u}) \) is the optimal value of the LP:

\[
\max_{v \in \Delta_{r}\cap \mathcal{P}_r} (g(w'_0, \hat{u}; c) + \tilde{g}(w'_0, \hat{u}; c, r))^T v,
\]

where \( g \in \mathbb{R}^L \) is the immediate stage-cost (i.e., \( C_{0:-1} \)) and \( \tilde{g} \in \mathbb{R}^L \) is the softmin over \( \tilde{\tau}[\hat{u}, w'_0] \) (the tail risk-sensitive costs), indexed by the next disturbance mode \( w'_0 \), and parameterized appropriately with respect to \( r, c \). Let \( v^* \) denote the optimal primal solution and \( \lambda^* \) the optimal dual solution for the constraints in \( \mathcal{P}_r \). Then,

\[
\nabla_r \tilde{\tau}[w^*_{-1|tN}](\hat{u}) = \sum_{i=1}^{L} v^*(i) \nabla_r \tilde{g}(w[i], \hat{u}; c, r) - \lambda^* ,
\]

\[
\nabla_c \tilde{\tau}[w^*_{-1|tN}](\hat{u}) = \sum_{i=1}^{L} v^*(i) \left[ \Phi_1(\hat{u}) + \nabla_c \tilde{g}(w[i], \hat{u}; c, r) \right],
\]

where \( \Phi_1(\hat{u})^3 \) is the vector whose \( j \)-th component is the \( \phi_j \) feature sum over \( N \) time steps under action trajectory \( \hat{u} \) and disturbance \( w[i] \).

\[
\nabla_r \tilde{g}(w[i], \hat{u}; c, r) = \mathbb{E}_{\hat{u} \sim \sigma} \left[ \tilde{\tau}[\{w^*_{-1|tN}, w[i]\}, \{\hat{u}\}] \right] \left[ \nabla_r \tilde{\tau}[\{w^*_{-1|tN}, \{\hat{u}\}\}] \right],
\]

\[
\nabla_c \tilde{g}(w[i], \hat{u}; c, r) = \mathbb{E}_{\hat{u} \sim \sigma} \left[ \tilde{\tau}[\{w^*_{-1|tN}, w[i]\}, \{\hat{u}\}] \right] \left[ \nabla_c \tilde{\tau}[\{w^*_{-1|tN}, \{\hat{u}\}\}] \right],
\]

where \( \sigma \left[ \tilde{\tau}[\{w^*_{-1|tN}, w[i]\}, \{\hat{u}\}] \right] \) is the discrete (cost-based) Boltzmann distribution, i.e.,

\[
\sigma \left[ \tilde{\tau}[\{w^*_{-1|tN}, w[i]\}, \{\hat{u}\}] \right] \propto \exp \left( -\tilde{\tau}[\{w^*_{-1|tN}, w[i]\}, \{\hat{u}\}] \right).
\]

Continuing the recursion, we arrive at the final two equations for the terminal stage:

\[
\nabla_r \tilde{\tau}[\omega_{T-2}, u_{T-2}](\hat{u}) = -\lambda^* ,
\]

\[
\nabla_c \tilde{\tau}[\omega_{T-2}, u_{T-2}](\hat{u}) = \sum_{i=1}^{L} v^*(i) \left[ \Phi_1(\hat{u}) \right],
\]

where \((v^*, \lambda^*)\) are the primal-dual optimal pair for the terminal LP (eq. (26)). For our experiments, we found a different form of the softmin function to be more numerically stable. In particular, we used

\[
\text{softmin}_\beta f(x) = \mathbb{E}_{x \sim \sigma_{\beta}[\cdot] - f(x)},
\]

\footnote{For notational convenience, we omit the obvious dependence on state.}
where $\sigma_\beta[-f]$ is the Boltzmann distribution defined with inverse temperature $\beta$ as:

$$\sigma_\beta[-f](x) \propto \exp(-\beta f(x)).$$

The gradients take the exact same form as (51) and (52) with $\sigma$ replaced by $\sigma_\beta$.

Note that the derivation above assumes non-degeneracy of the LPs. It is readily observed that by the piecewise linearity of LPs with respect to objective coefficients and constraint right-hand-sides, and the Lipschitz property of the softmin function, that $\tilde{\tau}$ is locally Lipschitz. Consequently, the log-likelihood too is locally Lipschitz. Thus, by the Rademacher theorem, the log-likelihood is non-differentiable only over a Lebesgue set of measure zero. If, however, during the updates, any (nested) LP is primal degenerate (multiple dual optimal solutions) or dual degenerate (multiple primal optimal solutions), the log-likelihood is non-differentiable (despite directional-derivatives existing in all directions). Thus, in its full generality, the max likelihood problem corresponds to a non-convex, non-smooth optimization and at points of non-differentiability, one must do extra work to compute a suitable descent direction. Some proposed approaches in the literature include penalized smoothing (Chen, 2012), sampling-based estimation (Burke et al., 2005) to approximate the Clarke generalized subdifferential and compute a descent direction, and majorization-minimization (Lanza et al., 2017) to iteratively optimize an upper-bound on a minimization problem. Arguably, the field is an active area of research.

In our implementation, we implemented the following two heuristics to avoid non-degeneracy (and consequent non-differentiability of the log-likelihood):

- During the projection step of the projected gradient method for $r$, if a constraint $a_j^T v \leq b(j) - r(j)$ is found to be redundant, the parameter $r(j)$ is re-adjusted to $r(j) := r(j) + 0.01$, provided that the resulting polytope is not empty. This has the effect of eliminating the possibility of redundant constraints at primal optimal vertices (thereby eliminating primal degeneracy).

- In the event that the objective vector is parallel to one of the bounding hyperplanes of the region $\Delta^L \cap \mathcal{P}_r$ (i.e., dual degeneracy), we added a small distortion to the objective vector.