Adversarial Robustness in Two-Stage Learning-to-Defer: Algorithms and Guarantees

Anonymous Authors[1]

Abstract

Learning-to-Defer (L2D) facilitates optimal task allocation between AI systems and decision-makers. Despite its potential, we show that current two-stage L2D frameworks are highly vulnerable to adversarial attacks, which can misdirect queries or overwhelm decision agents, significantly degrading system performance. This paper conducts the first comprehensive analysis of adversarial robustness in *two-stage* L2D frameworks. We introduce two novel attack strategies—*untargeted* and *targeted*—that exploit inherent structural vulnerabilities in these systems. To mitigate these threats, we propose SARD, a robust, convex, deferral algorithm rooted in Bayes and $(\mathcal{R}, \mathcal{G})$-consistency. Our approach guarantees optimal task allocation under adversarial perturbations for all surrogates in the cross-entropy family. Extensive experiments on classification, regression, and multi-task benchmarks validate the robustness of SARD.

1. Introduction

Learning-to-Defer (L2D) is a powerful framework that enables decision-making systems to optimally allocate queries among multiple agents, such as AI models, human experts, or other decision-makers (Madras et al., 2018). In the *two-stage* framework, agents are trained offline to harness their domain-specific expertise, enabling the allocation of each task to the decision-maker with the highest confidence (Mao et al., 2023a; 2024d; Montreuil et al., 2024). L2D is particularly valuable in high-stakes applications where reliability and performance are critical (Mozannar & Sontag, 2020; Verma et al., 2022). In healthcare, L2D systems integrate an AI diagnostic model with human specialists, delegating routine tasks to AI while deferring edge cases, such as anomalous imaging data, to specialists for nuanced evaluation (Mozannar et al., 2023). By dynamically assigning tasks to the most suitable agent, L2D ensures both accuracy and reliability, making it ideal for safety-critical domains.

Robustness in handling critical decisions is, therefore, essential for such systems. However, existing L2D frameworks are typically designed under the assumption of clean, non-adversarial input data, leaving them highly susceptible to adversarial perturbations—subtle input manipulations that disrupt task allocation and alter decision boundaries. Unlike traditional machine learning systems, where adversarial attacks primarily affect prediction outputs (Goodfellow et al., 2014; Szegedy et al., 2014; Awasthi et al., 2023), L2D systems are susceptible to more sophisticated adversarial threats, including query redirection to less reliable agents or intentional agent overloading. These vulnerabilities severely impact performance, drive up operational costs, and compromise trust.

This paper addresses the critical yet unexplored challenge of adversarial robustness in L2D systems. Inspired by adversarial attacks on classification (Goodfellow et al., 2014; Madry et al., 2017; Gowal et al., 2020), we introduce two novel attack strategies tailored to *two-stage* L2D: *untargeted attacks*, which disrupt agent allocation, and *targeted attacks*, which redirect queries to specific agents. To counter these attacks, we propose a robust family of surrogate losses based on cross-entropy (Mao et al., 2023a; 2024d; Montreuil et al., 2024), designed to ensure robustness in classification, regression, and multi-task settings. Building upon advances in consistency theory for adversarial robustness (Bao et al., 2021; Awasthi et al., 2022; 2023; Mao et al., 2023b), we establish both Bayes-consistency and $(\mathcal{R}, \mathcal{G})$ consistency for our surrogate losses, enabling reliable task allocation even under adversarial scenarios. Our algorithm, **SARD**, leverages these guarantees while preserving convexity.

Our key contributions are:

1. We introduce two novel adversarial attack strategies for *two-stage* L2D: *targeted* attacks that redirect queries and *untargeted* attacks that disrupt task allocation.

2. We propose a robust family of surrogate losses with guarantees of Bayes-consistency and $(\mathcal{R}, \mathcal{G})$ consis-

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

tency across classification, regression, and multi-task settings. Our convex algorithm, SARD, leverages these guarantees to achieve robust task allocation.

3. We empirically demonstrate that our novel attacks effectively exploit vulnerabilities in state-of-the-art *two-stage* L2D approaches, while our algorithm, SARD, exhibits strong robustness against the attack across diverse tasks.

This work lays the first theoretical foundation for adversarial robustness in L2D systems.

2. Related Works

Learning-to-Defer: The first *single-stage* L2D approach was introduced by Madras et al. (2018), training both the predictor and the rejector, which were built upon the framework established in Cortes et al. (2016). In a seminal work, Mozannar & Sontag (2020) proposed the first approach proven to be Bayes-consistent, ensuring optimal allocation. Verma et al. (2022) presented an alternative formulation based on one-versus-all surrogates, also proven to be Bayes-consistent, and later extended to a broader family of losses by Charusaie et al. (2022). More recently, Cao et al. (2024) proposed an asymmetric softmax surrogate to improve probability estimation between agents, addressing limitations in both Mozannar & Sontag (2020) and Verma et al. (2022). Furthermore, Mozannar et al. (2023) demonstrated that the approaches from Mozannar & Sontag (2020); Verma et al. (2022) are not realizable-$\mathcal{H}$-consistent, leading to suboptimal performance for some distributions. Mao et al. (2024b) generalized the work of Mozannar & Sontag (2020) proving both Bayes and $\mathcal{H}$-consistency, while Mao et al. (2024c) extended it to realizable-$\mathcal{H}$-consistency.

In the *two-stage* setting, where agents are already trained offline, Mao et al. (2023a) introduced the first classification approach that guarantees both Bayes-consistency and $\mathcal{H}$-consistency. This work was further extended by Mao et al. (2024d), who adapted the two-stage framework to regression tasks while maintaining these consistency guarantees. Additionally, Montreuil et al. (2024) generalized the approach to multi-task learning.

Adversarial Robustness: The robustness of neural networks against adversarial perturbations has been extensively studied, with foundational work highlighting their vulnerabilities (Biggio et al., 2013; Szegedy et al., 2014; Goodfellow et al., 2014; Madry et al., 2017). A key focus in recent research has been on developing consistency frameworks for formulating robust defenses. Bao et al. (2021) proposed a Bayes-consistent surrogate loss tailored for adversarial training, which was further analyzed and extended in subsequent works (Meunier et al., 2022; Awasthi et al., 2021). Beyond Bayes-consistency, $\mathcal{H}$-consistency has been explored to address robustness in diverse settings. Notably, Awasthi et al. (2022) derived $\mathcal{H}$-consistency bounds for several surrogate families, and Mao et al. (2023b) conducted an in-depth analysis of the cross-entropy family. Building on these theoretical advancements, Awasthi et al. (2023) introduced a smooth algorithm that leverages consistency guarantees to enhance robustness in adversarial settings.

Our work builds upon recent advancements in consistency theory to further improve adversarial robustness in two-stage L2D.

3. Preliminaries

Multi-task scenario. We consider a multi-task setting that addresses both classification and regression problems simultaneously. Let $\mathcal{X}$ denote the input space, $\mathcal{Y} = \{1, \dots, n\}$ represent the set of n distinct classes for classification, and $\mathcal{T} \subseteq \mathbb{R}$ denote the target space for regression. Each data point is represented as a triplet $z = (x, y, t) \in \mathcal{Z}$, where $\mathcal{Z} = \mathcal{X} \times \mathcal{Y} \times \mathcal{T}$. We assume the data is drawn independently and identically distributed (i.i.d.) from an underlying distribution $\mathcal{D}$ over $\mathcal{Z}$. To model this multi-task problem, we introduce a *backbone* $w \in \mathcal{W}$, which acts as a shared feature extractor. The backbone maps inputs $x \in \mathcal{X}$ to a latent feature representation $q \in \mathcal{Q}$, via the function $w : \mathcal{X} \to \mathcal{Q}$. Building upon this backbone, we define a *classifier* $h \in \mathcal{H}$, representing all possible classification heads. Formally, $h : \mathcal{Q} \times \mathcal{Y} \to \mathbb{R}$ is a scoring function, with predictions computed as $h(q) = \arg\max_{y \in \mathcal{Y}} h(q, y)$. Similarly, we define a *regressor* $f \in \mathcal{F}$, which maps latent features to real-valued targets, $f : \mathcal{Q} \to \mathcal{T}$. These components are integrated into a single multi-head network $g \in \mathcal{G}$, defined as $\mathcal{G} = \{g : g(x) = (h \circ w(x), f \circ w(x)) \mid w \in \mathcal{W}, h \in \mathcal{H}, f \in \mathcal{F}\}$.

Consistency in classification: In classification, the primary objective is to identify a classifier $h \in \mathcal{H}$ that minimizes the true error $\mathcal{E}_{\ell_{01}}(h)$, defined as $\mathcal{E}_{\ell_{01}}(h) = \mathbb{E}_{(x,y)}[\ell_{01}(h, x, y)]$. The Bayes-optimal error is expressed as $\mathcal{E}^B_{\ell_{01}}(\mathcal{H}) = \inf_{h \in \mathcal{H}} \mathcal{E}_{\ell_{01}}(h)$. However, minimizing $\mathcal{E}_{\ell_{01}}(h)$ directly is challenging due to the non-differentiability of the *true multiclass* 0-1 loss (Zhang, 2002; Steinwart, 2007; Awasthi et al., 2022). To address this challenge, surrogate losses are employed as convex, non-negative upper bounds on ℓ_{01}. A notable family of multiclass surrogate losses is the comp-sum (Mohri et al., 2012; Mao et al., 2023b), which we refer to as a family of *multiclass surrogate* losses:

$$\Phi^u_{01}(h, x, y) = \Psi^u\Big(\sum_{y' \neq y} \Psi_{\mathrm{e}}(h(x, y) - h(x, y'))\Big), \quad (1)$$

where $\Psi_e(v) = \exp(-v)$, which defines the cross-entropy family. For $u > 0$, the transformation is given by:

$$\Psi^u(v) = \begin{cases} \log(1+v) & \text{if } u = 1, \\ \frac{1}{1-u}\left[(1-v)^{1-u} - 1\right] & \text{if } u > 0 \wedge u \neq 1. \end{cases} \tag{2}$$

This formulation generalizes several well-known loss functions, including the sum-exponential loss (Weston & Watkins, 1998), logistic loss (Ohn Aldrich, 1997), generalized cross-entropy (Zhang & Sabuncu, 2018), and mean absolute error loss (Ghosh et al., 2017). The corresponding true error for Φ_{01}^u is defined as $\mathcal{E}_{\Phi_{01}^u}(h) = \mathbb{E}_{(x,y)}[\Phi_{01}^u(h,x,y)]$, with its optimal value expressed as $\mathcal{E}^*_{\Phi_{01}^u}(\mathcal{H}) = \inf_{h\in\mathcal{H}} \mathcal{E}_{\Phi_{01}^u}(h)$.

A key property of a surrogate loss is *Bayes-consistency*, which ensures that minimizing the surrogate excess risk leads to minimizing the true excess risk (Zhang, 2002; Bartlett et al., 2006; Steinwart, 2007; Tewari & Bartlett, 2007). Formally, Φ_{01}^u is Bayes-consistent with respect to ℓ_{01} if, for any sequence $\{h_k\}_{k\in\mathbb{N}} \subset \mathcal{H}$, the following implication holds:

$$\begin{aligned} &\mathcal{E}_{\Phi_{01}^u}(h_k) - \mathcal{E}^*_{\Phi_{01}^u}(\mathcal{H}) \xrightarrow{k\to\infty} 0 \\ &\implies \mathcal{E}_{\ell_{01}}(h_k) - \mathcal{E}^B_{\ell_{01}}(\mathcal{H}) \xrightarrow{k\to\infty} 0. \end{aligned} \tag{3}$$

This property typically assumes $\mathcal{H} = \mathcal{H}_{\text{all}}$, which may not hold for restricted hypothesis classes such as $\mathcal{H}_{\text{lin}}$ or $\mathcal{H}_{\text{ReLU}}$ (Long & Servedio, 2013; Awasthi et al., 2022; Mao et al., 2024a). To characterize consistency with a particular hypothesis set, Awasthi et al. (2022) introduced $\mathcal{H}$-consistency bounds, which rely on a non-decreasing function $\Gamma : \mathbb{R}^+ \to \mathbb{R}^+$ and take the following form:

$$\begin{aligned} &\mathcal{E}_{\Phi_{01}^u}(h) - \mathcal{E}^*_{\Phi_{01}^u}(\mathcal{H}) + \mathcal{U}_{\Phi_{01}^u}(\mathcal{H}) \geq \\ &\Gamma\Big(\mathcal{E}_{\ell_{01}}(h) - \mathcal{E}^B_{\ell_{01}}(\mathcal{H}) + \mathcal{U}_{\ell_{01}}(\mathcal{H})\Big), \end{aligned} \tag{4}$$

where the minimizability gap $\mathcal{U}_{\ell_{01}}(\mathcal{H})$ quantifies the difference between the best-in-class excess risk and the expected pointwise minimum error: $\mathcal{U}_{\ell_{01}}(\mathcal{H}) = \mathcal{E}^B_{\ell_{01}}(\mathcal{H}) - \mathbb{E}_x\left[\inf_{h\in\mathcal{H}} \mathbb{E}_{y|x}\left[\ell_{01}(h,x,y)\right]\right]$. The gap vanishes when $\mathcal{H} = \mathcal{H}_{\text{all}}$ (Steinwart, 2007; Awasthi et al., 2022). In the asymptotic limit, inequality (4) ensures recovery of Bayes-consistency (3).

Adversarial robustness: Adversarial robust classification aims to train classifiers that are robust to small, imperceptible perturbations of the input (Goodfellow et al., 2014; Madry et al., 2017). The objective is to minimize the *true multiclass loss* ℓ_{01} evaluated on an adversarial input $x' = x + \delta$ (Gowal et al., 2020; Awasthi et al., 2022). A perturbation δ is constrained by its magnitude, and we define the adversarial region around x as $B_p(x,\gamma) = \{x' \mid \|x'-x\|_p \leq \gamma\}$, where $\|\cdot\|_p$ is the p-norm and $\gamma \in (0,1)$ specifies the maximum allowed perturbation. The *adversarial true multiclass loss* $\widetilde{\ell}_{01} : \mathcal{H}\times\mathcal{X}\times\mathcal{Y} \to \{0,1\}$ is given by:

$$\widetilde{\ell}_{01}(h,x,y) = \sup_{x'\in B_p(x,\gamma)} \ell_{01}(h(x'),y). \tag{5}$$

Similarly to classification, minimizing $\widetilde{\ell}_{01}$ is computationally infeasible (Zhang & Agarwal, 2020; Bartlett et al., 2006; Awasthi et al., 2022). To address this, we introduce the family of *adversarial margin surrogate* losses $\widetilde{\Phi}_{01}^{\rho,u}$ from the comp-sum ρ-margin family, which approximate the *adversarial true multiclass loss* $\widetilde{\ell}_{01}$. This family is defined as:

$$\widetilde{\Phi}_{01}^{\rho,u}(h,x,y) = \sup_{x'\in B_p(x,\gamma)} \Psi^u\Big(\sum_{y'\neq y} \Psi_\rho(h(x',y') - h(x',y))\Big). \tag{6}$$

Here, Ψ^u and Ψ_ρ represent transformations that characterize the behavior of the family, where the non-convex transformation is defined as $\Psi_\rho(v) = \min\left\{\max\left(0, 1-\frac{v}{\rho}\right), 1\right\}$. Recent studies have demonstrated that algorithms employing smooth regularized variants of the comp-sum ρ-margin losses achieve $\mathcal{H}$-consistency, thereby offering strong theoretical guarantees (Awasthi et al., 2022; 2023; Mao et al., 2023b).

Two-stage Learning-to-Defer: The Learning-to-Defer framework assigns queries $x \in \mathcal{X}$ to the most confident *agent*, aiming to enhance performance by leveraging the strengths of multiple agents. The agents consist of a primary model and J experts, denoted by the set $\mathcal{A} = \{0\} \cup [J]$, where 0 corresponds to the primary model g defined in Section 3. Each expert M_j provides a prediction pair $m_j(x) = (m_j^h(x), m_j^f(x))$, where $m_j^h(x) \in \mathcal{Y}$ is a categorical prediction and $m_j^f(x) \in \mathcal{T}$ is a regression estimate. The combined predictions of all J experts are represented as $m(x) = \big(m_1(x), \ldots, m_J(x)\big)$, which lies in the joint prediction space $\mathcal{M}$. In the two-stage setting, all agents are trained offline, and the framework focuses on *query allocation*, keeping agent parameters fixed.

A rejector function $r \in \mathcal{R}$, defined as $r : \mathcal{X}\times\mathcal{A} \to \mathbb{R}$, is learned to assign a query x to the agent $j \in \mathcal{A}$ with the highest rejection score $r(x) = \arg\max_{j\in\mathcal{A}} r(x,j)$, as described by Mao et al. (2024d; 2023a); Montreuil et al. (2024).

Definition 3.1 (Two-Stage L2D losses)**.** Let an input $x \in \mathcal{X}$, for any $r \in \mathcal{R}$, we have the *true deferral loss*:

$$\ell_{\text{def}}(r,g,m,z) = \sum_{j=0}^{J} c_j(g(x), m_j(x), z) 1_{r(x)=j},$$

and its family of convex, non-negative, upper-bound *surro-*

gate deferral losses:

$$\Phi_{\text{def}}^{u}(r,g,m,z)=\sum_{j=0}^{J}\tau_j(g(x),m(x),z)\Phi_{01}^{u}(r,x,j),$$

where c_j denotes the non-negative bounded cost of assigning a decision to agent $j \in \mathcal{A}$ (Madras et al., 2018). If the rejector $r \in \mathcal{R}$ assigns $r(x)=0$, the query is handled by the primary model g, which predicts $g(x)=(h(w(x)), f(w(x)))$ and incurs a general cost $c_0(g(x),z)=\psi(g(x),z)$. Here, $\psi:\mathcal{Y}\times\mathcal{T}\times\mathcal{Z}\to\mathbb{R}^+$ is a general measure used to quantify the prediction quality of g with respect to z. If $r(x)=j$ for some $j>0$, the query is deferred to expert j, incurring a cost $c_j(m_j(x),z)=\psi(m_j(x),z)+\beta_j$, where β_j represents the consultation cost associated with expert j. The aggregated cost across all agents is then defined as:

$$\tau_j(g(x),m(x),z)=\sum_{i=0}^{J}c_i(g(x),m_i(x),z)1_{i\neq j} \quad (7)$$

recovering the formulation from Mao et al. (2024d) and Montreuil et al. (2024). Note that in the case of classification, the function ψ corresponds to the ℓ_{01} loss.

4. Adversarial Attacks on Two-Stage L2D

Motivation and Setting The two-stage L2D framework is designed to route queries to the most accurate agents, ensuring optimal decision-making (Mao et al., 2023a; 2024d; Montreuil et al., 2024). Despite its effectiveness, we demonstrate that this framework is inherently vulnerable to adversarial attacks that exploit its reliance on the rejector function, a key component responsible for query allocation. Given this critical role, our analysis focuses on adversarial attacks and corresponding defenses targeting the rejector $r \in \mathcal{R}$, rather than individual agents. This focus is justified because adversarial defenses for specific agents can typically be deployed offline within the two-stage L2D setup. Moreover, evaluating the robustness of individual agents under adversarial conditions simplifies to selecting the most robust agent, whereas ensuring robustness at the system level constitutes a fundamentally different challenge.

Untargeted Attack: In classification, the goal of an untargeted attack is to find a perturbed input $x' \in B_p(x,\gamma)$ that causes the classifier $h \in \mathcal{H}$ to misclassify the input (Goodfellow et al., 2014; Akhtar & Mian, 2018). Specifically, for a clean input $x \in \mathcal{X}$ where the classifier correctly predicts $h(x)=y$, the attacker aims to identify a perturbed input x' such that $h(x') \neq y$.

In the context of Learning-to-Defer, the attack extends beyond misclassification to compromising the decision allocation mechanism. Given an optimal agent $j^* \in \mathcal{A}$, the attacker aims to find an adversarial input $x' \in B_p(x,\gamma)$ such that $r(x') \neq j^*$, thereby forcing deferral to a suboptimal agent $j \in \mathcal{A}\setminus\{j^*\}$—incurring a higher loss. This leads to the following untargeted attack formulation:

Definition 4.1 (Untargeted Attack in L2D)**.** Let $x' \in B_p(x,\gamma)$ be an adversarial input, where $B_p(x,\gamma)$ denotes the p-norm ball of radius γ centered at x. The untargeted attack that maximizes misallocation in L2D is formulated as follows:

$$x'=\operatorname*{arg\,sup}_{x'\in B_p(x,\gamma)}\sum_{j=0}^{J}\tau_j(g(x),m(x),z)\Phi_{01}^{u}(r,x',j).$$

Definition (4.1) characterizes an attack in which the adversary maximizes the aggregate discrepancy between the rejector's allocation for the adversarial input x' and its original allocation, thereby forcing the system to defer to an unintended agent. The adversarial input x' is designed to maximize these surrogate losses. As a result, the adversarial attack increases the system's overall loss, significantly degrading its performance. An illustration of this attack is provided in Appendix Figure 1.

Targeted Attack: Targeted attacks are often more impactful than untargeted ones, as they exploit specific system vulnerabilities to achieve precise adversarial goals (Akhtar & Mian, 2018; Chakraborty et al., 2021). For example, in autonomous driving classification, a targeted attack could deliberately misclassify a stop sign as a speed limit sign, potentially leading to hazardous consequences. A targeted attack exploits this asymmetry by forcing the classifier $h \in \mathcal{H}$ to predict a specific target class $y_t \in \mathcal{Y}$, potentially leading to harmful consequences.

In the context of L2D, an attacker aims to manipulate the system into assigning a query $x' \in B_p(x,\gamma)$ to a predetermined expert $j_t \in \mathcal{A}$ rather than the optimal expert $j^* \in \mathcal{A}$. This targeted attack objective can be formally expressed as follows:

Definition 4.2 (Targeted Attack in L2D)**.** Let $x' \in B_p(x,\gamma)$, where $B_p(x,\gamma)$ denotes the p-norm ball of radius γ centered at x. The attack targeting the allocation of a query x to the expert $j_t \in \mathcal{A}$ is defined as:

$$x'=\operatorname*{arg\,inf}_{x'\in B_p(x,\gamma)}\tau_{j_t}(g(x),m(x),z)\Phi_{01}^{u}(r,x',j_t).$$

The adversarial input x' minimizes the loss associated with the targeted agent j_t, thereby biasing the allocation process towards agent j_t as described in Definition (4.2). For instance, an attacker may have an affiliated partner j_t among the system's agents. Suppose the system operates under a pay-per-query model—for example, a specialist doctor in a medical decision-making system or a third-party service provider in an AI-powered platform. By manipulating the allocation mechanism to systematically route more queries toward j_t, the attacker artificially inflates its workload, leading to unjustified financial gains. These gains may benefit both the attacker and the affiliated expert through direct financial compensation, revenue-sharing agreements, or other collusive incentives. An illustration of this attack is provided in Appendix Figure 2.

5. Adversarially Consistent Formulation for Two-Stage Learning-to-Defer

In this section, we introduce an adversarially consistent formulation of the two-stage L2D framework, ensuring robustness against attacks while preserving optimal query allocation.

5.1. Novel Two-stage Learning-to-Defer formulation

Adversarial True Deferral Loss: To defend against the novel attacks introduced in Section 4, we define the worst-case *adversarial true deferral loss*, $\widetilde{\ell}_{\text{def}} : \mathcal{R}\times\mathcal{G}\times\mathcal{M}\times\mathcal{Z} \to \mathbb{R}^+$, which quantifies the maximum incurred loss under adversarial perturbations. Specifically, for each $j \in \mathcal{A}$, an adversarial perturbation δ_j is applied, yielding the perturbed input $x'_j = x + \delta_j$, which lies within an ℓ_p-norm ball of radius γ, i.e., $x'_j \in B_p(x,\gamma)$. We define the *j-th adversarial true multiclass loss* as $\widetilde{\ell}^{j}_{01}(r,x,j) = \sup_{x'_j\in B_p(x,\gamma)} \ell_{01}(r(x'_j),j)$, which captures the worst-case misclassification loss when deferring to agent j under adversarial conditions. The formal definition of the adversarial true deferral loss is provided in Lemma 5.1.

Lemma 5.1 (Adversarial True Deferral Loss)**.** *Let $x \in \mathcal{X}$ denote the clean input, c_j the cost associated with agent $j \in \mathcal{A}$, and τ_j the aggregated cost. The adversarial true deferral loss $\widetilde{\ell}_{def}$ is defined as:*

$$\widetilde{\ell}_{def}(r,g,m,z) = \sum_{j=0}^{J} \tau_j(g(x),m(x),z)\widetilde{\ell}^{j}_{01}(r,x,j) + (1-J)\sum_{j=0}^{J} c_j(g(x),m_j(x),z).$$

See Appendix D.1 for the proof of Lemma 5.1. The attacker's objective is to compromise the allocation process by identifying perturbations δ_j that maximize the loss for each agent $j \in \mathcal{A}$. Importantly, the costs c_j and τ_j are evaluated based on the clean input x, as the agents' predictions remain unaffected by the perturbations (see Section 4).

Minimizing the *adversarial true deferral loss* in Lemma 5.1 is NP-hard (Zhang, 2002; Bartlett et al., 2006; Steinwart, 2007; Awasthi et al., 2022). Therefore, as in classification problems, we approximate this discontinuous loss using surrogates.

Adversarial Margin Deferral Surrogate Losses: In the formulation of the *adversarial true deferral loss* (Lemma 5.1), discontinuities arise due to the indicator function in the loss definition. To approximate this discontinuity, we build on recent advancements in consistency theory for adversarially robust classification (Bao et al., 2021; Awasthi et al., 2022; 2023; Mao et al., 2023b) and propose a continuous, upper-bound surrogate family for the *adversarial true deferral loss*. Specifically, we define the *j-th adversarial margin surrogate* family $\widetilde{\Phi}^{\rho,u,j}_{01}(r,x,j) = \sup_{x'_j\in B_p(x,\gamma)} \Psi^u\Big(\sum_{j'\neq j}\Psi_\rho\big(r(x'_j,j') - r(x'_j,j)\big)\Big)$ where Ψ^u and Ψ_ρ are defined in Equation (6). Building on this, we derive the *adversarial margin deferral surrogate* losses as:

Lemma 5.2 (Adversarial Margin Deferral Surrogate Losses)**.** *Let $x \in \mathcal{X}$ denote the clean input and τ_j the aggregated cost. The adversarial margin deferral surrogate losses $\widetilde{\Phi}^{\rho,u}_{def}$ are then defined as:*

$$\widetilde{\Phi}^{\rho,u}_{def}(r,g,m,z) = \sum_{j=0}^{J}\tau_j(g(x),m(x),z)\widetilde{\Phi}^{\rho,u,j}_{01}(r,x,j).$$

The proof is provided in Appendix D.2. One notable limitation of the *adversarial margin deferral surrogate* family is the non-convexity of the *j-th adversarial margin surrogate loss* family $\widetilde{\Phi}^{\rho,u,j}_{01}$, which poses significant challenges for efficient optimization.

Adversarial Smooth Deferral Surrogate Losses: As detailed by Awasthi et al. (2023); Mao et al. (2023b), the non-convex *adversarial margin surrogate* family can be replaced with a smooth and convex approximation. To this end, we adapt their results and introduce the *smooth adversarial surrogate* family, denoted as $\widetilde{\Phi}^{\text{smth},u}_{01} : \mathcal{R}\times\mathcal{X}\times\mathcal{A}\to\mathbb{R}^+$, which approximates the supremum term in Lemma 5.1. Crucially, $\widetilde{\Phi}^{\text{smth},u}_{01}$ acts as a convex, non-negative upper bound for the *j-th adversarial margin surrogate* family, such that $\widetilde{\Phi}^{\rho,u,j}_{01} \leq \widetilde{\Phi}^{\text{smth},u}_{01}$. We derive the *smooth adversarial surrogate* losses in Lemma 5.3.

Lemma 5.3 (Smooth Adversarial Surrogate Losses)**.** *Let $x \in \mathcal{X}$ denote the clean input and hyperparameters $\rho > 0$ and $\nu > 0$. We define the smooth adversarial surrogate losses as:*

$$\widetilde{\Phi}^{smth,u}_{01}(r,x,j) = \Phi^u_{01}(\frac{r}{\rho},x,j) + \nu \sup_{x'_j\in B_p(x,\gamma)} \|\overline{\Delta}_r(x'_j,j) - \overline{\Delta}_r(x,j)\|_2.$$

For completeness, the proof is provided in Appendix D.3. For $x \in \mathcal{X}$, define $\Delta_r(x,j,j') = r(x,j) - r(x,j')$, and let $\overline{\Delta}_r(x,j)$ denote the J-dimensional vector $\big(\Delta_r(x,j,0),\ldots,\Delta_r(x,j,j-1),\Delta_r(x,j,j+1),\ldots,\Delta_r(x,j,J)\big)$. The first term, $\Phi_{01}^u(r/\rho,x,j)$, corresponds to the *multiclass surrogate* losses modulated by the coefficient $\rho>0$. The second term incorporates adversarial evaluations $x'_j\in B_p(x,\gamma)$ for each agent $j\in\mathcal{A}$, with a smooth adversarial component scaled by the coefficient $\nu>0$ (Awasthi et al., 2023; Mao et al., 2023b). The coefficients (ρ,ν) are typically selected through cross-validation to balance allocation performance and robustness against adversarial perturbations.

Using the *smooth adversarial surrogate* family from Lemma 5.3, we define the *smooth adversarial deferral surrogate* (SAD) family $\widetilde{\Phi}_{\text{def}}^{\text{smth},u}:\mathcal{R}\times\mathcal{G}\times\mathcal{M}\times\mathcal{Z}\to\mathbb{R}^+$, which is convex, non-negative, and serves as an upper bound for $\widetilde{\ell}_{\text{def}}$ by construction. The formal definition of our novel surrogates is given as:

Lemma 5.4 (SAD: **S**mooth **A**dversarial **D**eferral Surrogate Losses)**.** *Let $x\in\mathcal{X}$ denote the clean input and τ_j the aggregated cost. Then, the smooth adversarial surrogate family (or SAD) $\widetilde{\Phi}_{def}^{smth,u}$ is defined as:*

$$\widetilde{\Phi}_{def}^{smth,u}(r,g,m,z)=\sum_{j=0}^{J}\tau_j(g(x),m(x),z)\widetilde{\Phi}_{01}^{smth,u}(r,x,j).$$

The proof is provided in Appendix D.4. While the *smooth adversarial deferral surrogate* family (SAD) provides a smooth and computationally efficient approximation of the *adversarial true deferral loss*, the question of its consistency remains a critical consideration (Zhang, 2002; Bartlett et al., 2006).

5.2. Theoretical Guarantees

To establish the theoretical foundations of SAD, we prove the Bayes-consistency and $(\mathcal{R},\mathcal{G})$-consistency of the *adversarial margin deferral surrogate* family $\widetilde{\Phi}_{\text{def}}^{\rho,u}$ (Lemma 5.2). Furthermore, we demonstrate that these guarantees naturally extend to a regularized empirical formulation of SAD, referred to as SARD.

$(\mathcal{R},\mathcal{G})$-consistency bounds of $\widetilde{\Phi}_{\text{def}}^{\rho,u}$: A key foundational step involves demonstrating the $\mathcal{R}$-consistency of the *j-th adversarial margin surrogate* losses $\widetilde{\Phi}_{01}^{\rho,u,j}$, under the assumption that $\mathcal{R}$ is symmetric and that there exists a rejector $r\in\mathcal{R}$ that is *locally ρ-consistent*.

Definition 5.5 (Locally ρ-consistent)**.** A hypothesis set $\mathcal{R}$ is *locally ρ-consistent* if, for any $x\in\mathcal{X}$, there exists a hypothesis $r\in\mathcal{R}$ such that:

$$\inf_{x'\in B_p(x,\gamma)}|r(x',i)-r(x',j)|\geq\rho,$$

where $\rho>0$, $i\neq j\in\mathcal{A}$, and $x'\in B_p(x,\gamma)$. Additionally, for any $x'\in B_p(x,\gamma)$, the set $\{r(x',j):j\in\mathcal{A}\}$ preserves the same ordering as for x.

As shown in Awasthi et al. (2022); Mao et al. (2023b); Awasthi et al. (2023), commonly used hypothesis sets, such as linear models, neural networks, and the set of all measurable functions, are locally ρ-consistent for some $\rho>0$. Consequently, the guarantees established in Lemma 5.6 are general and broadly applicable across diverse practical settings. The proof of Lemma 5.6 is deferred to Appendix D.5.

Lemma 5.6 ($\mathcal{R}$-consistency bounds for $\widetilde{\Phi}_{01}^{\rho,u,j}$)**.** *Assume $\mathcal{R}$ is symmetric and locally ρ-consistent. Then, for the agent set $\mathcal{A}$, any hypothesis $r\in\mathcal{R}$, and any distribution $\mathcal{P}$ with probabilities $p=(p_0,\cdots,p_J)\in\Delta^{|\mathcal{A}|}$, the following inequality holds:*

$$\sum_{j\in\mathcal{A}}p_j\widetilde{\ell}_{01}^j(r,x,j)-\inf_{r\in\mathcal{R}}\sum_{j\in\mathcal{A}}p_j\widetilde{\ell}_{01}^j(r,x,j)\leq$$
$$\Psi^u(1)\Big(\sum_{j\in\mathcal{A}}p_j\widetilde{\Phi}_{01}^{\rho,u,j}(r,x,j)-\inf_{r\in\mathcal{R}}\sum_{j\in\mathcal{A}}p_j\widetilde{\Phi}_{01}^{\rho,u,j}(r,x,j)\Big).$$

Lemma 5.6 establishes the consistency of the *j-th adversarial margin surrogate* family $\widetilde{\Phi}_{01}^{\rho,u,j}$ for probabilities $p_j\in\Delta^{|\mathcal{A}|}$, explicitly incorporating adversarial inputs defined for each $j\in\mathcal{A}$. This result distinguishes our contribution from prior works (Mao et al., 2023b; Awasthi et al., 2023; 2022), which do not address adversarial inputs at the level of the distribution (dependent on j). By addressing this limitation, Lemma 5.6 provides a critical theoretical guarantee, demonstrating that the *j-th adversarial margin surrogate* family $\widetilde{\Phi}_{01}^{\rho,u,j}$ aligns with the adversarial loss $\widetilde{\ell}_{01}^j$ under the specified assumptions.

Building on the foundational result of Lemma 5.6, we prove the Bayes and $(\mathcal{R},\mathcal{G})$-consistency of the *adversarial margin deferral surrogate* losses. The proof of Theorem 5.7 is provided in Appendix D.6.

Theorem 5.7 ($(\mathcal{R},\mathcal{G})$-consistency bounds of $\widetilde{\Phi}_{\text{def}}^{\rho,u}$)**.** *Let $\mathcal{R}$ be symmetric and locally ρ-consistent. Then, for the agent set $\mathcal{A}$, any hypothesis $r\in\mathcal{R}$, and any distribution $\mathcal{D}$, the following holds for a multi-task model $g\in\mathcal{G}$:*

$$\begin{aligned}&\mathcal{E}_{\widetilde{\ell}_{def}}(r,g)-\mathcal{E}^B_{\widetilde{\ell}_{def}}(\mathcal{R},\mathcal{G})+\mathcal{U}_{\widetilde{\ell}_{def}}(\mathcal{R},\mathcal{G})\leq\\&\quad\Psi^u(1)\Big(\mathcal{E}_{\widetilde{\Phi}^{\rho,u}_{def}}(r)-\mathcal{E}^*_{\widetilde{\Phi}^{\rho,u}_{def}}(\mathcal{R})+\mathcal{U}_{\widetilde{\Phi}^{\rho,u}_{def}}(\mathcal{R})\Big)\\&\quad+\mathcal{E}_{c_0}(g)-\mathcal{E}^B_{c_0}(\mathcal{G})+\mathcal{U}_{c_0}(\mathcal{G}).\end{aligned}$$

Theorem 5.7 establishes the consistency of the *adversarial margin deferral surrogate* family $\widetilde{\Phi}_{\text{def}}^{\rho,u}$, ensuring its alignment with the *true adversarial deferral loss* $\widetilde{\ell}_{\text{def}}$. The minimizability gaps derived in Theorem 5.7 vanish when $\mathcal{R}=\mathcal{R}_{\text{all}}$ and $\mathcal{G}=\mathcal{G}_{\text{all}}$ (Steinwart, 2007; Awasthi et al., 2022). Under the assumption that these gaps vanish, the following holds:

$$\mathcal{E}_{\widetilde{\ell}_{\text{def}}}(r, g) - \mathcal{E}^B_{\widetilde{\ell}_{\text{def}}}(\mathcal{R}, \mathcal{G}) \leq \mathcal{E}_{c_0}(g) - \mathcal{E}^B_{c_0}(\mathcal{G}) + \Psi^u(1)\Big(\mathcal{E}_{\widetilde{\Phi}^{\rho,u}_{\text{def}}}(r) - \mathcal{E}^*_{\widetilde{\Phi}^{\rho,u}_{\text{def}}}(\mathcal{R})\Big). \tag{8}$$

After offline training of the multi-task model, we assume that the excess c_0-risk is bounded as $\mathcal{E}_{c_0}(g) - \mathcal{E}^B_{c_0}(\mathcal{G}) \leq \epsilon_0$. Similarly, after training the rejector, the excess risk of the surrogate satisfies $\mathcal{E}_{\widetilde{\Phi}^{\rho,u}_{\text{def}}}(r) - \mathcal{E}^*_{\widetilde{\Phi}^{\rho,u}_{\text{def}}}(\mathcal{R}) \leq \epsilon_1$. Under these conditions, the ℓ_{def}-excess risk is bounded as $\mathcal{E}_{\widetilde{\ell}_{\text{def}}}(r, g) - \mathcal{E}^B_{\widetilde{\ell}_{\text{def}}}(\mathcal{R}, \mathcal{G}) \leq \epsilon_0 + \Psi^u(1)\epsilon_1$, establishing both the Bayes-consistency and $(\mathcal{R}, \mathcal{G})$-consistency of the surrogate losses $\widetilde{\Phi}^{\rho,u}_{\text{def}}$.

Building on the theoretical guarantees of the non-convex family $\widetilde{\Phi}^{\rho,u}_{\text{def}}$, we introduce a *smooth adversarial regularized deferral* (SARD) algorithm. SARD extends the standard SAD framework by incorporating a regularization term that enhances stability and robustness. Despite this modification, SARD preserves the key theoretical guarantees of $\widetilde{\Phi}^{\rho,u}_{\text{def}}$, ensuring consistency and minimizability under the same conditions (Mao et al., 2023b; Awasthi et al., 2023).

Guarantees for SARD: Using the fact that $\widetilde{\Phi}^{\text{smth},u}_{01} \geq \widetilde{\Phi}^{\rho,u,j}_{01}$, we establish guarantees for our *smooth adversarial deferral surrogate* family $\widetilde{\Phi}^{\text{smth},u}_{\text{def}}$ under the same conditions.

Corollary 5.8 (Guarantees for SAD)**.** *Assume $\mathcal{R}$ is symmetric and locally ρ-consistent. Then, for the agent set $\mathcal{A}$, any hypothesis $r \in \mathcal{R}$, and any distribution $\mathcal{D}$, the following holds for a multi-task model $g \in \mathcal{G}$:*

$$\begin{aligned}
&\mathcal{E}_{\widetilde{\ell}_{def}}(r, g) - \mathcal{E}^B_{\widetilde{\ell}_{def}}(\mathcal{R}, \mathcal{G}) + \mathcal{U}_{\widetilde{\ell}_{def}}(\mathcal{R}, \mathcal{G}) \leq \\
&\quad + \Psi^u(1)\Big(\mathcal{E}_{\widetilde{\Phi}^{smth,u}_{def}}(r) - \mathcal{E}^*_{\widetilde{\Phi}^{\rho,u}_{def}}(\mathcal{R}) + \mathcal{U}_{\widetilde{\Phi}^{\rho,u}_{def}}(\mathcal{R})\Big) \\
&\quad + \mathcal{E}_{c_0}(g) - \mathcal{E}^B_{c_0}(\mathcal{G}) + \mathcal{U}_{c_0}(\mathcal{G}).
\end{aligned}$$

Corollary 5.8 establishes that SAD shows similar consistency properties under the given conditions, with the minimizability gap vanishing for $\mathcal{R} = \mathcal{R}_{\text{all}}$ and $\mathcal{G} = \mathcal{G}_{\text{all}}$. This motivates the development of an adversarial robustness algorithm based on minimizing a regularized empirical formulation of SAD, referred to as SARD.

Proposition 5.9 (SARD: **S**mooth **A**dversarial **R**egularized **D**eferral Algorithm)**.** *Assume $\mathcal{R}$ is symmetric and locally ρ-consistent. For a regularizer Ω and hyperparameter $\eta > 0$, the regularized empirical risk minimization problem for SARD is:*

$$\min_{r \in \mathcal{R}} \left[\frac{1}{K} \sum_{k=1}^{K} \widetilde{\Phi}^{smth,u}_{def}(r, g, m, z_k) + \eta \Omega(r) \right].$$

The pseudo-code for SARD is provided in Appendix C.

6. Experiments

We evaluate the robustness of SARD against state-of-the-art two-stage L2D frameworks across three tasks: classification, regression, and multi-task learning. Our experiments reveal that while existing baselines achieve slightly higher performance under clean conditions, they suffer from severe performance degradation under adversarial attacks. In contrast, SARD consistently maintains high performance, demonstrating superior robustness to both untargeted and targeted attacks. To the best of our knowledge, this is the first study to address adversarial robustness within the context of Learning-to-Defer.

6.1. Multiclass Classification Task

We compare our robust SARD formulation against the method introduced by Mao et al. (2023a) on the CIFAR-100 dataset (Krizhevsky, 2009).

Setting: Categories were assigned to three experts with a correctness probability $p = 0.94$, while the remaining probability was uniformly distributed across the other categories, following the approach in (Mozannar & Sontag, 2020; Verma et al., 2022; Cao et al., 2024). To further evaluate robustness, we introduced a weak expert M_3, with only a few assigned categories, and assumed that the attacker is aware of this weakness. Agent costs are defined as $c_0(h(x), y) = \ell_{01}(h(x), y)$ for the model and $c_{j>0}(m^h_j(x), y) = \ell_{01}(m^h_j(x), y)$, aligned with (Mozannar & Sontag, 2020; Mozannar et al., 2023; Verma et al., 2022; Cao et al., 2024; Mao et al., 2023a). Both the model and the rejector were implemented using ResNet-4 (He et al., 2015). The agents' performance, additional training details, and experimental results are provided in Appendix E.1.

Baseline	Clean	Untarg.	Targ. M_1	Targ. M_2	Targ. M_3
Mao et al. (2023a)	72.8 ± 0.4	17.2 ± 0.2	54.4 ± 0.1	45.4 ± 0.1	13.4 ± 0.1
Our	67.0 ± 0.4	49.8 ± 0.3	62.4 ± 0.3	62.1 ± 0.2	64.8 ± 0.3

Table 1. Comparison of accuracy results between the proposed SARD and the baseline (Mao et al., 2023a) on the CIFAR-100 validation set, including clean and adversarial scenarios.

Results: The results in Table 6.1 underscore the robustness of our proposed SARD algorithm. While the baseline achieves a higher clean accuracy (72.8% vs. 67.0%), this comes at the cost of extreme vulnerability to adversarial attacks. In contrast, SARD prioritizes robustness, significantly outperforming the baseline under adversarial conditions. Specifically, in the presence of untargeted attacks, SARD retains an accuracy of 49.8%, a 2.9 times improvement over the baseline's sharp decline to 17.2%. Similarly, under targeted attacks aimed at the weak expert M_3, our method achieves 64.8% accuracy, a stark contrast to the baseline's 13.4%, highlighting SARD's ability to counteract adversarial exploitation of weak experts. These findings validate the efficacy of SARD in preserving performance across diverse attack strategies.

6.2. Regression Task

We evaluate the performance of SARD against the method proposed by Mao et al. (2024d) using the California Housing dataset involving median house price prediction (Kelley Pace & Barry, 1997).

Setting: We train three experts, each implemented as an MLP, specializing in a specific subset of the dataset based on a predefined localization criterion. Among these, expert M_3 is designed to specialize in a smaller region, resulting in comparatively weaker overall performance. Agent costs for regression are defined as $c_0(f(x), t) = \text{RMSE}(f(x), t)$ for the model and $c_{j>0}(m_j^f(x), t) = \text{RMSE}(m_j^f(x), t)$, aligned with (Mao et al., 2024d). Both the model and the rejector are trained on the full dataset using MLPs. We provide detailed agent performance results, training procedures, and additional experimental details in Appendix E.2.

Baseline	Clean	Untarg.	Targ. M_1	Targ. M_2	Targ. M_3
Mao et al. (2024d)	0.17 ± 0.01	0.29 ± 0.3	0.40 ± 0.02	0.21 ± 0.01	0.41 ± 0.05
Our	0.17 ± 0.01	0.17 ± 0.01	0.18 ± 0.01	0.18 ± 0.01	0.18 ± 0.01

Table 2. Performance comparison of SARD with the baseline (Mao et al., 2024d) on the California Housing dataset. The table reports Root Mean Square Error (RMSE).

Results: Table 6.2 presents the comparative performance of the baseline and SARD under clean and adversarial conditions. Under clean settings, both approaches achieve similar performance with an RMSE of 0.17. However, under adversarial attacks—both untargeted and targeted at specific experts (e.g., M_3)—SARD demonstrates significant robustness, maintaining an RMSE of 0.18 across all conditions. In contrast, the baseline's performance degrades substantially, with RMSE values increasing to 0.29 and 0.41 under untargeted and M_3-targeted attacks, respectively.

6.3. Multi-Task

We evaluate the performance of our robust SARD algorithm against the baseline introduced by Montreuil et al. (2024) on the Pascal VOC dataset (Everingham et al., 2010), a benchmark for object detection tasks combining both interdependent classification and regression objectives.

Setting: We train two Faster R-CNN models (Ren et al., 2016) as experts, each specializing in a distinct subset of the dataset. Expert M_1 is trained exclusively on images containing animals, while expert M_2 focuses on images with vehicles. Agent costs are defined as $c_0(g(x), z) = \text{mAP}(g(x), z)$ for the model and $c_{j>0}(m_j(x), z) = \text{mAP}(m_j(x), z)$, aligned with (Montreuil et al., 2024). The primary model and the rejector are implemented as lightweight versions of Faster R-CNN using MobileNet (Howard et al., 2017). We provide detailed performance results, training procedures, and additional experimental details in Appendix E.3.

Baseline	Clean	Untarg.	Targ. M_1	Targ. M_2
Montreuil et al. (2024)	44.4 ± 0.4	9.7 ± 0.1	17.4 ± 0.2	20.4 ± 0.2
Our	43.9 ± 0.4	39.0 ± 0.3	39.7 ± 0.3	39.5 ± 0.3

Table 3. Performance comparison of SARD with the baseline (Montreuil et al., 2024) on the Pascal VOC dataset. The table reports mean Average Precision (mAP) under clean and adversarial scenarios.

Results: Table 6.3 presents the performance comparison between SARD and the baseline under clean and adversarial scenarios. Both methods perform comparably in clean conditions, with the baseline achieving a slightly higher mAP of 44.4 compared to 43.9 for SARD. However, under adversarial scenarios, the baseline experiences a significant performance drop, with mAP decreasing to 9.7 in untargeted attacks and 17.4 in targeted attacks on M_1. In contrast, SARD demonstrates strong robustness, maintaining mAP scores close to the clean setting across all attack types. Specifically, SARD achieves an mAP of 39.0 under untargeted attacks and 39.7 when targeted at M_1, highlighting its resilience to adversarial perturbations.

7. Conclusion

In this paper, we address the critical and previously underexplored problem of adversarial robustness in two-stage Learning-to-Defer systems. We introduce two novel adversarial attack strategies—untargeted and targeted—that exploit inherent vulnerabilities in existing L2D frameworks. To mitigate these threats, we propose R-ADVs-L2D, a robust deferral algorithm that provides theoretical guarantees based on Bayes consistency and $(\mathcal{R}, \mathcal{G})$-consistency. We evaluate our approach across classification, regression, and multi-task scenarios. Our experiments demonstrate the effectiveness of the proposed adversarial attacks in significantly degrading the performance of existing two-stage L2D baselines. In contrast, R-ADVs-L2D exhibits strong robustness against these attacks, consistently maintaining high performance.

Impact Statement

This paper introduces methods to improve the adversarial robustness of two-stage Learning-to-Defer frameworks, which allocate decision-making tasks between AI systems and human experts. The work has the potential to advance the field of Machine Learning, particularly in high-stakes domains such as healthcare, finance, and safety-critical systems, where robustness and reliability are essential. By mitigating vulnerabilities to adversarial attacks, this research ensures more secure and trustworthy decision-making processes.

The societal implications of this work are largely positive, as it contributes to enhancing the reliability and fairness of AI systems. However, as with any advancement in adversarial robustness, there is a potential for misuse if adversarial strategies are exploited for harmful purposes. While this paper does not directly address these ethical concerns, we encourage further exploration of safeguards and responsible deployment practices in future research.

No immediate or significant ethical risks have been identified in this work, and its societal impacts align with the well-established benefits of improving robustness in Machine Learning systems.

References

Akhtar, N. and Mian, A. Threat of adversarial attacks on deep learning in computer vision: A survey. *Ieee Access*, 6:14410–14430, 2018.

Awasthi, P., Frank, N., Mao, A., Mohri, M., and Zhong, Y. Calibration and consistency of adversarial surrogate losses, 2021. URL https://arxiv.org/abs/2104.09658.

Awasthi, P., Mao, A., Mohri, M., and Zhong, Y. Multi-class h-consistency bounds. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2022. Curran Associates Inc. ISBN 9781713871088.

Awasthi, P., Mao, A., Mohri, M., and Zhong, Y. Theoretically grounded loss functions and algorithms for adversarial robustness. In Ruiz, F., Dy, J., and van de Meent, J.-W. (eds.), *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pp. 10077–10094. PMLR, 25–27 Apr 2023. URL https://proceedings.mlr.press/v206/awasthi23c.html.

Bao, H., Scott, C., and Sugiyama, M. Calibrated surrogate losses for adversarially robust classification, 2021. URL https://arxiv.org/abs/2005.13748.

Bartlett, P., Jordan, M., and McAuliffe, J. Convexity, classification, and risk bounds. *Journal of the American Statistical Association*, 101:138–156, 02 2006. doi: 10.1198/016214505000000907.

Biggio, B., Corona, I., Maiorca, D., Nelson, B., Šrndić, N., Laskov, P., Giacinto, G., and Roli, F. *Evasion Attacks against Machine Learning at Test Time*, pp. 387–402. Springer Berlin Heidelberg, 2013. ISBN 9783642387098. doi: 10.1007/978-3-642-40994-3_25. URL http://dx.doi.org/10.1007/978-3-642-40994-3_25.

Cao, Y., Mozannar, H., Feng, L., Wei, H., and An, B. In defense of softmax parametrization for calibrated and consistent learning to defer. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2024. Curran Associates Inc.

Chakraborty, A., Alam, M., Dey, V., Chattopadhyay, A., and Mukhopadhyay, D. A survey on adversarial attacks and defences. *CAAI Transactions on Intelligence Technology*, 6(1):25–45, 2021.

Charusaie, M.-A., Mozannar, H., Sontag, D., and Samadi, S. Sample efficient learning of predictors that complement humans, 2022.

Cortes, C., DeSalvo, G., and Mohri, M. Learning with rejection. In Ortner, R., Simon, H. U., and Zilles, S. (eds.), *Algorithmic Learning Theory*, pp. 67–82, Cham, 2016. Springer International Publishing. ISBN 978-3-319-46379-7.

Everingham, M., Van Gool, L., Williams, C., Winn, J., and Zisserman, A. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88: 303–338, 06 2010. doi: 10.1007/s11263-009-0275-4.

Ghosh, A., Kumar, H., and Sastry, P. S. Robust loss functions under label noise for deep neural networks. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, pp. 1919–1925. AAAI Press, 2017.

Goodfellow, I. J., Shlens, J., and Szegedy, C. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

Gowal, S., Qin, C., Uesato, J., Mann, T. A., and Kohli, P. Uncovering the limits of adversarial training against norm-bounded adversarial examples. *ArXiv*, abs/2010.03593, 2020. URL https://api.semanticscholar.org/CorpusID:222208628.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition, 2015. URL https://arxiv.org/abs/1512.03385.

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., and Adam, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications, 2017. URL https://arxiv.org/abs/1704.04861.

Kelley Pace, R. and Barry, R. Sparse spatial autoregressions. *Statistics and Probability Letters*, 33(3):291–297, 1997. ISSN 0167-7152. doi: https://doi.org/10.1016/S0167-7152(96)00140-X. URL https://www.sciencedirect.com/science/article/pii/S016771529600140X.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization, 2017. URL https://arxiv.org/abs/1412.6980.

Krizhevsky, A. Learning multiple layers of features from tiny images. 2009. URL https://api.semanticscholar.org/CorpusID:18268744.

Long, P. and Servedio, R. Consistency versus realizable h-consistency for multiclass classification. In Dasgupta, S. and McAllester, D. (eds.), *Proceedings of the 30th International Conference on Machine Learning*, number 3 in Proceedings of Machine Learning Research, pp. 801–809, Atlanta, Georgia, USA, 17–19 Jun 2013. PMLR. URL https://proceedings.mlr.press/v28/long13.html.

Madras, D., Pitassi, T., and Zemel, R. Predict responsibly: Improving fairness and accuracy by learning to defer, 2018.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. Towards deep learning models resistant to adversarial attacks. *ArXiv*, abs/1706.06083, 2017. URL https://api.semanticscholar.org/CorpusID:3488815.

Mao, A., Mohri, C., Mohri, M., and Zhong, Y. Two-stage learning to defer with multiple experts. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023a. URL https://openreview.net/forum?id=GIlsH0T4b2.

Mao, A., Mohri, M., and Zhong, Y. Cross-entropy loss functions: Theoretical analysis and applications, 2023b. URL https://arxiv.org/abs/2304.07288.

Mao, A., Mohri, M., and Zhong, Y. h-consistency bounds: Characterization and extensions. *Advances in Neural Information Processing Systems*, 36, 2024a.

Mao, A., Mohri, M., and Zhong, Y. Principled approaches for learning to defer with multiple experts, 2024b. URL https://arxiv.org/abs/2310.14774.

Mao, A., Mohri, M., and Zhong, Y. Realizable h-consistent and bayes-consistent loss functions for learning to defer, 2024c. URL https://arxiv.org/abs/2407.13732.

Mao, A., Mohri, M., and Zhong, Y. Regression with multi-expert deferral, 2024d. URL https://arxiv.org/abs/2403.19494.

Meunier, L., Ettedgui, R., Pinot, R., Chevaleyre, Y., and Atif, J. Towards consistency in adversarial classification, 2022. URL https://arxiv.org/abs/2205.10022.

Mohri, M., Rostamizadeh, A., and Talwalkar, A. *Foundations of machine learning*. MIT Press, 2012.

Montreuil, Y., Yeo, S. H., Carlier, A., Ng, L. X., and Ooi, W. T. Two-stage learning-to-defer for multi-task learning, 2024. URL https://arxiv.org/abs/2410.15729.

Mozannar, H. and Sontag, D. Consistent estimators for learning to defer to an expert. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org, 2020.

Mozannar, H., Lang, H., Wei, D., Sattigeri, P., Das, S., and Sontag, D. A. Who should predict? exact algorithms for learning to defer to humans. In *International Conference on Artificial Intelligence and Statistics*, 2023. URL https://api.semanticscholar.org/CorpusID:255941521.

Ohn Aldrich, R. A. Fisher and the making of maximum likelihood 1912-1922. *Statistical Science*, 12(3):162–179, 1997.

Ren, S., He, K., Girshick, R., and Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks, 2016. URL https://arxiv.org/abs/1506.01497.

Steinwart, I. How to compare different loss functions and their risks. *Constructive Approximation*, 26:225–287, 2007. URL https://api.semanticscholar.org/CorpusID:16660598.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. Intriguing properties of neural networks, 2014. URL https://arxiv.org/abs/1312.6199.

Tewari, A. and Bartlett, P. L. On the consistency of multiclass classification methods. *Journal of Machine Learning Research*, 8(36):1007–1025, 2007. URL http://jmlr.org/papers/v8/tewari07a.html.

Verma, R., Barrejon, D., and Nalisnick, E. Learning to defer to multiple experts: Consistent surrogate losses, confidence calibration, and conformal ensembles. In *International Conference on Artificial Intelligence and Statistics*, 2022. URL https://api.semanticscholar.org/CorpusID:253237048.

Weston, J. and Watkins, C. Multi-class support vector machines. Technical report, Citeseer, 1998.

Zhang, M. and Agarwal, S. Bayes consistency vs. h-consistency: The interplay between surrogate loss functions and the scoring function class. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 16927–16936. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/c4c28b367e14df88993ad475dedf6b77-Paper.pdf.

Zhang, T. Statistical behavior and consistency of classification methods based on convex risk minimization. *Annals of Statistics*, 32, 12 2002. doi: 10.1214/aos/1079120130.

Zhang, Z. and Sabuncu, M. R. Generalized cross entropy loss for training deep neural networks with noisy labels, 2018. URL https://arxiv.org/abs/1805.07836.

A. Notation and Preliminaries for the Appendices

We summarize the key notations and concepts introduced in the main text:

Input Space and Outputs:

- $\mathcal{X}$: Input space for $x \in \mathcal{X}$.
- $\mathcal{Q}$: Latent representation $q \in \mathcal{Q}$.
- $\mathcal{Y} = \{1, \dots, n\}$: Categorical output space for classification tasks.
- $\mathcal{T} \subseteq \mathbb{R}$: Continuous output space for regression tasks.
- $\mathcal{Z} = \mathcal{X} \times \mathcal{Y} \times \mathcal{T}$: Combined space of inputs and labels.

Learning-to-Defer Setting:

- $\mathcal{A} = \{0\} \cup [J]$: Set of agents, where 0 refers to the primary model $g = (h, f)$, and J denotes the number of experts.
- $m_j(x) = (m_j^h(x), m_j^f(x))$: Predictions by expert j, where $m_j^h(x) \in \mathcal{Y}$ is a categorical prediction and $m_j^f(x) \in \mathcal{T}$ is a regression estimate.
- $c_0(g(x), z) = \psi(g(x), z)$: The cost associated to the multi-task model.
- $c_{j>0}(m(x), z) = \psi(m(x), z) + \beta_j$: The cost associated to the expert j with query cost $\beta_j \geq 0$.
- $\psi : \mathcal{Y} \times \mathcal{T} \times \mathcal{Z} \to \mathbb{R}^+$: Quantify the prediction's quality.

Hypothesis Sets:

- $\mathcal{W}$: Set of backbones $w : \mathcal{X} \to \mathcal{Q}$.
- $\mathcal{H}$: Set of classifiers $h : \mathcal{Q} \times \mathcal{Y} \to \mathbb{R}$.
- $\mathcal{F}$: Set of regressors $f : \mathcal{Q} \to \mathcal{T}$.
- $\mathcal{G}$: Single multi-head network $\mathcal{G} = \{g : g(x) = (h \circ w(x), f \circ w(x)) \mid w \in \mathcal{W}, h \in \mathcal{H}, f \in \mathcal{F}\}$.
- $\mathcal{R}$: Set of rejectors $r : \mathcal{X} \to \mathcal{A}$.

Adversarial Definitions:

- $x'_j \in B_p(x, \gamma)$: the adversarial input for the agent $j \in \mathcal{A}$ in the p-norm ball $B_p(x, \gamma) = \{x'_j \in \mathcal{X} \mid \|x'_j - x\|_p \leq \gamma\}$
- $\widetilde{\ell}_{01}^j(r, x, j) = \sup_{x'_j \in B_p(x,\gamma)} \ell_{01}(r, x'_j, j)$: j-th Adversarial multiclass loss.
- $\widetilde{\Phi}_{01}^{\rho,u,j}(r, x, j) = \sup_{x'_j \in B_p(x,\gamma)} \Psi^u \Big(\sum_{j' \neq j} \Psi_\rho \big(r(x'_j, j') - r(x'_j, j) \big) \Big)$: j-th Adversarial margin surrogate losses, providing a differentiable proxy for $\widetilde{\ell}_{01}^j$.

This notation will be consistently used throughout the appendices to ensure clarity and coherence in theoretical and empirical discussions.

B. Attacks Illustration

B.1. Untargeted Attack

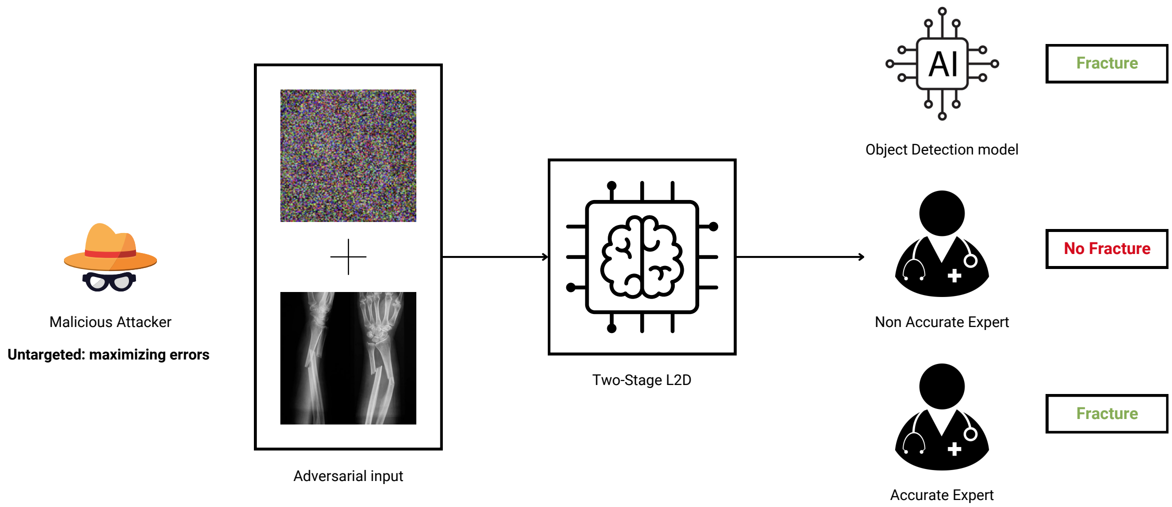

Figure 1. Untargeted Attack: The malicious attacker perturbs the input to increase the probability that the query is assigned to a less accurate expert, thereby maximizing classification errors. Rather than targeting a specific expert, the attack injects adversarial noise to disrupt the expert allocation process, leading to erroneous routing and degraded decision-making.

B.2. Targeted Attack

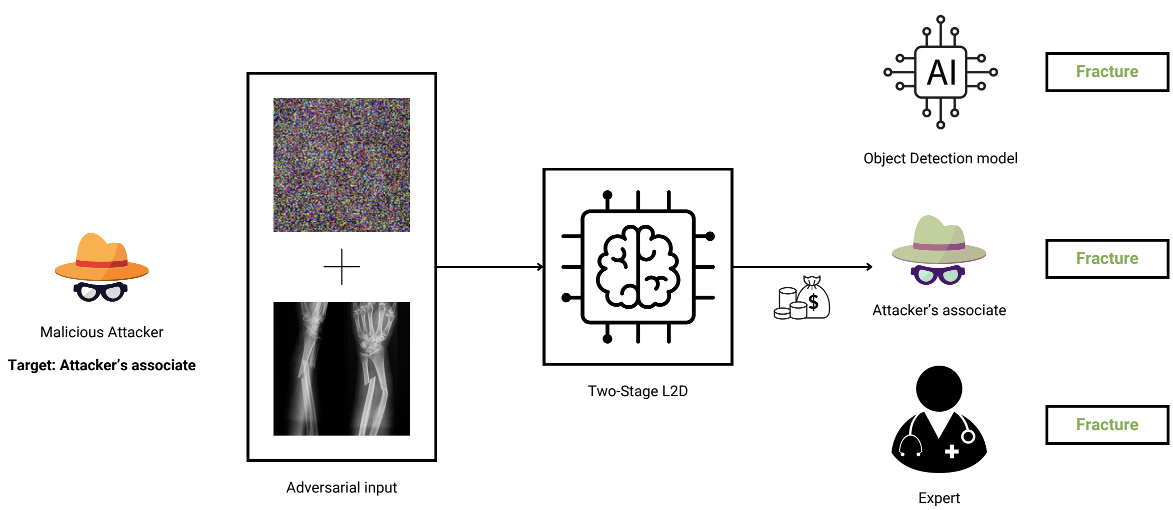

Figure 2. Targeted Attack: The malicious attacker perturbs the input to increase the probability that the query is assigned to its associated agent. By manipulating the L2D system to systematically route queries to this associate, the adversary ensures that the associate receives a higher volume of queries, thereby increasing its earnings.

C. Algorithm

Algorithm 1 SARD Algorithm

Input: Dataset $\{(x_k, y_k, t_k)\}_{k=1}^K$, multi-task model $g \in \mathcal{G}$, experts $m \in \mathcal{M}$, rejector $r \in \mathcal{R}$, number of epochs EPOCH, batch size BATCH, adversarial parameters (ρ, ν), regularizer parameter η, learning rate λ.
Initialization: Initialize rejector parameters θ.
for $i = 1$ to EPOCH **do**
  Shuffle dataset $\{(x_k, y_k, t_k)\}_{k=1}^K$.
  for each mini-batch $\mathcal{B} \subset \{(x_k, y_k, t_k)\}_{k=1}^K$ of size BATCH **do**
    Extract input-output pairs $z = (x, y, t) \in \mathcal{B}$.
    Query model $g(x)$ and experts $m(x)$. {Agents have been trained offline and fixed}
    Evaluate costs $c_0(g(x), z)$ and $c_{j>0}(m(x), z)$. {Compute costs}
    for $j = 0$ to J **do**
      Evaluate rejector score $r(x, j)$. {Rejection score of agent j}
      Generate adversarial input $x'_j = x + \delta_j$ with $\delta_j \in B_p(x, \gamma)$. {$\ell_p$-ball perturbation for agent j}
      Run PGD attack on x'_j:
        $\sup_{x'_j \in B_p(x,\gamma)} \|\overline{\Delta}_r(x'_j, j) - \overline{\Delta}_r(x, j)\|_2$. {Smooth robustness evaluation}
      Compute Adversarial Smooth surrogate losses $\widetilde{\Phi}_{01}^{\text{smth},u}(r, x, j)$.
    end for
    Compute the regularized empirical risk minimization:
      $\widehat{\mathcal{E}}_{\Phi_{\text{def}}}^{\Omega}(r;\theta) = \frac{1}{\text{BATCH}} \sum_{z\in\mathcal{B}} \left[\widetilde{\Phi}_{\text{def}}^{\text{smth},u}(r, g, m, z)\right] + \eta\Omega(r)$.
    Update parameters θ:
      $\theta \leftarrow \theta - \lambda \nabla_\theta \widehat{\mathcal{E}}_{\Phi_{\text{def}}}^{\Omega}(r;\theta)$. {Gradient update}
  end for
end for
Return: trained rejector model r^*.

D. Proof Adversarial Robustness in Two-Stage Learning-to-Defer

D.1. Proof Lemma 5.1

Lemma 5.1 (Adversarial True Deferral Loss). *Let $x \in \mathcal{X}$ denote the clean input, c_j the cost associated with agent $j \in \mathcal{A}$, and τ_j the aggregated cost. The adversarial true deferral loss $\widetilde{\ell}_{def}$ is defined as:*

$$\begin{aligned}\widetilde{\ell}_{def}(r, g, m, z) &= \sum_{j=0}^{J} \tau_j(g(x), m(x), z)\widetilde{\ell}_{01}^{j}(r, x, j) \\ &\quad + (1 - J)\sum_{j=0}^{J} c_j(g(x), m_j(x), z).\end{aligned}$$

Proof. In adversarial training, the objective is to optimize the worst-case scenario of the objective function under adversarial inputs $x' \in B_p(x, \gamma)$. For our case, we start with the standard L2D loss for the two-stage setting (Mao et al., 2024d; Montreuil et al., 2024):

$$\begin{aligned}\ell_{\text{def}}(r, g, m, z) &= \sum_{j=0}^{J} c_j(g(x), m_j(x), z) 1_{r(x)=j} \\ &= \sum_{j=0}^{J} \tau_j(g(x), m(x), z) 1_{r(x)\neq j} + (1 - J)\sum_{j=0}^{J} c_j(g(x), m_j(x), z)\end{aligned} \tag{9}$$

using

$$\tau_j(g(x), m(x), z) = \begin{cases} \sum_{i=1}^{J} c_i(m_i(x), z) & \text{if } j = 0 \\ c_0(g(x), z) + \sum_{i=1}^{J} c_i(m_i(x), z) 1_{i \neq j} & \text{otherwise} \end{cases} \tag{10}$$

Next, we derive an upper bound for Equation (9) by considering the supremum over all adversarial perturbations $x' \in B_p(x, \gamma)$, under the fact that the attack is solely on the rejector $r \in \mathcal{R}$:

$$\ell_{\text{def}}(r, g, m, z) \leq \sup_{x' \in B_p(x,\gamma)} \Big(\sum_{j=0}^{J} \tau_j(g(x), m(x), z) 1_{r(x') \neq j} \Big) + (1 - J) \sum_{j=0}^{J} c_j(g(x), m_j(x), z) \tag{11}$$

However, the formulation in Equation (11) does not fully capture the worst-case scenario in L2D. Specifically, this formulation might not result in a robust approach, as it does not account for the adversarial input $x'_j \in B_p(x, \gamma)$ that maximizes the loss for every agent $j \in \mathcal{A}$. Incorporating this worst-case scenario, we obtain:

$$\ell_{\text{def}}(r, g, m, z) \leq \sum_{j=0}^{J} \tau_j(g(x), m(x), z) \sup_{x'_j \in B_p(x,\gamma)} 1_{r(x'_j) \neq j} + (1 - J) \sum_{j=0}^{J} c_j(g(x), m_j(x), z) \tag{12}$$

Thus, formulating with the margin loss $\rho_r(x, j) = r(x, j) - \max_{j' \neq j} r(x, j')$, leads to the desired result:

$$\begin{aligned} \widetilde{\ell}_{\text{def}}(r, g, m, z) &= \sum_{j=0}^{J} \tau_j(g(x), m(x), z) \sup_{x'_j \in B_p(x,\gamma)} 1_{\rho_r(x'_j, j) \leq 0} + (1 - J) \sum_{j=0}^{J} c_j(g(x), m_j(x), z) \\ &= \sum_{j=0}^{J} \tau_j(g(x), m(x), z) \widetilde{\ell}^{j}_{01}(r, x, j) + (1 - J) \sum_{j=0}^{J} c_j(g(x), m_j(x), z) \end{aligned} \tag{13}$$

with $\widetilde{\ell}^{j}_{01}(r, x, j) = \sup_{x'_j \in B_p(x,\gamma)} 1_{\rho_r(x'_j, j) \leq 0}$ □

D.2. Proof Lemma 5.2

Lemma 5.2 (Adversarial Margin Deferral Surrogate Losses)**.** *Let $x \in \mathcal{X}$ denote the clean input and τ_j the aggregated cost. The adversarial margin deferral surrogate losses $\widetilde{\Phi}^{\rho,u}_{def}$ are then defined as:*

$$\widetilde{\Phi}^{\rho,u}_{def}(r, g, m, z) = \sum_{j=0}^{J} \tau_j(g(x), m(x), z) \widetilde{\Phi}^{\rho,u,j}_{01}(r, x, j).$$

Proof. Referring to adversarial true deferral loss defined in Lemma 5.1, we have:

$$\begin{aligned} \widetilde{\ell}_{\text{def}}(r, g, m, z) &= \sum_{j=0}^{J} \tau_j(g(x), m(x), z) \sup_{x'_j \in B_p(x,\gamma)} 1_{\rho_r(x'_j, j) \leq 0} + (1 - J) \sum_{j=0}^{J} c_j(g(x), m_j(x), z) \\ &= \sum_{j=0}^{J} \tau_j(g(x), m(x), z) \widetilde{\ell}^{j}_{01}(r, x, j) + (1 - J) \sum_{j=0}^{J} c_j(g(x), m_j(x), z) \end{aligned}$$

By definition, $\widetilde{\Phi}^{\rho,u,j}_{01}$ upper bounds the j-th adversarial classification loss $\widetilde{\ell}^{j}_{01}$, leading to:

$$\widetilde{\ell}_{\text{def}}(r, g, m, z) \leq \sum_{j=0}^{J} \tau_j(g(x), m(x), z) \widetilde{\Phi}^{\rho,u,j}_{01}(r, x, j) + (1 - J) \sum_{j=0}^{J} c_j(g(x), m_j(x), z) \tag{14}$$

Then, dropping the term that does not depend on $r \in \mathcal{R}$, leads to the desired formulation:

$$\widetilde{\Phi}^{\rho,u}_{\text{def}}(r, g, m, z) = \sum_{j=0}^{J} \tau_j(g(x), m(x), z) \widetilde{\Phi}^{\rho,u,j}_{01}(r, x, j) \tag{15}$$

□

D.3. Proof Lemma 5.3

Lemma 5.3 (Smooth Adversarial Surrogate Losses)**.** *Let $x \in \mathcal{X}$ denote the clean input and hyperparameters $\rho > 0$ and $\nu > 0$. We define the smooth adversarial surrogate losses as:*

$$\widetilde{\Phi}_{01}^{smth,u}(r,x,j) = \Phi_{01}^{u}(\frac{r}{\rho},x,j) + \nu \sup_{x'_j \in B_p(x,\gamma)} \|\overline{\Delta}_r(x'_j,j) - \overline{\Delta}_r(x,j)\|_2.$$

Proof. Let $x \in \mathcal{X}$ denote an input and $x'_j \in B_p(x,\gamma)$ an adversarially perturbed input within an ℓ_p-norm ball of radius γ for each agent. Let $r \in \mathcal{R}$ be a rejector. We now define the composite-sum ρ-margin losses for both clean and adversarial scenarios:

$$\begin{aligned}\Phi_{01}^{\rho,u}(r,x,j) &= \Psi^u\left(\sum_{j'\neq j}\Psi_\rho\big(r(x,j') - r(x,j)\big)\right)\\ \widetilde{\Phi}_{01}^{\rho,u,j}(r,x,j) &= \sup_{x'_j \in B_p(x,\gamma)} \Psi^u\left(\sum_{j'\neq j}\Psi_\rho\big(r(x'_j,j') - r(x'_j,j)\big)\right)\end{aligned} \tag{16}$$

where $\Psi_{\text{e}}(v) = \exp(-v)$. For $u > 0$, the transformation Ψ^u is defined as:

$$\Psi^{u=1}(v) = \log(1+v), \quad \Psi^{u\neq 1}(v) = \frac{1}{1-u}\left[(1-v)^{1-u} - 1\right]$$

It follows that for all $u > 0$ and $v \geq 0$, we have $\left|\frac{\partial \Psi^u}{\partial v}(v)\right| = \frac{1}{(1+v)^u} \leq 1$ ensuring that Ψ^u is 1-Lipschitz over $\mathbb{R}^+$ (Mao et al., 2023b).

Define $\Delta_r(x,j,j') = r(x,j) - r(x,j')$ and let $\overline{\Delta}_r(x,j)$ denote the J-dimensional vector:

$$\overline{\Delta}_r(x,j) = \big(\Delta_r(x,j,0), \ldots, \Delta_r(x,j,j-1), \Delta_r(x,j,j+1), \ldots, \Delta_r(x,j,J)\big)$$

For any $u > 0$, with Ψ^u non-decreasing and 1-Lipschitz:

$$\widetilde{\Phi}_{01}^{\rho,j}(r,x,j) \leq \Phi_{01}^{\rho,u}(r,x,j) + \sup_{x'_j \in B_p(x,\gamma)} \sum_{j'\neq j}\Big(\Psi_\rho\big(-\Delta_r(x'_j,j,j')\big) - \Psi_\rho\big(-\Delta_r(x,j,j')\big)\Big) \tag{17}$$

Since $\Psi_\rho(z)$ is $\frac{1}{\rho}$-Lipschitz, by the Cauchy-Schwarz inequality and for $\nu \geq \frac{\sqrt{n-1}}{\rho} \geq \frac{1}{\rho}$:

$$\widetilde{\Phi}_{01}^{\rho,u,j}(r,x,j) \leq \Phi_{01}^{\rho,u}(r,x,j) + \nu \sup_{x'_j \in B_p(x,\gamma)} \|\overline{\Delta}_r(x'_j,j) - \overline{\Delta}_r(x,j)\|_2 \tag{18}$$

Using $\Phi_{01}^{u}(r,x,y) = \Psi^u\big(\sum_{y'\neq y}\Psi_{\text{e}}(r(x,y) - r(x,y'))\big)$ with $\Psi_{\text{e}}(v) = \exp(-v)$ and the fact that $\Psi_{\text{e}}(v/\rho) \geq \Psi_\rho(v)$, we obtain:

$$\widetilde{\Phi}_{01}^{\rho,u,j}(r,x,j) \leq \Phi_{01}^{u}\left(\frac{r}{\rho},x,j\right) + \nu \sup_{x'_j \in B_p(x,\gamma)} \|\overline{\Delta}_r(x'_j,j) - \overline{\Delta}_r(x,j)\|_2 \tag{19}$$

Finally, we have the desired smooth surrogate losses upper-bounding $\widetilde{\Phi}_{01}^{\text{smth},u} \geq \widetilde{\Phi}_{01}^{\rho,u,j}$:

$$\widetilde{\Phi}_{01}^{\text{smth},u}(r,x,j) = \Phi_{01}^{u}\left(\frac{r}{\rho},x,j\right) + \nu \sup_{x'_j \in B_p(x,\gamma)} \|\overline{\Delta}_r(x'_j,j) - \overline{\Delta}_r(x,j)\|_2 \tag{20}$$

□

D.4. Proof Lemma 5.4

Lemma 5.4 (SAD: **S**mooth **A**dversarial **D**eferral Surrogate Losses)**.** *Let $x \in \mathcal{X}$ denote the clean input and τ_j the aggregated cost. Then, the smooth adversarial surrogate family (or SAD) $\widetilde{\Phi}_{def}^{smth,u}$ is defined as:*

$$\widetilde{\Phi}_{def}^{smth,u}(r, g, m, z) = \sum_{j=0}^{J} \tau_j(g(x), m(x), z)\widetilde{\Phi}_{01}^{smth,u}(r, x, j).$$

Proof. Using Lemma 5.2, we have:

$$\widetilde{\Phi}_{\text{def}}^{\rho,u}(r, g, m, z) = \sum_{j=0}^{J} \tau_j(g(x), m(x), z)\widetilde{\Phi}_{01}^{\rho,j}(r, x, j) \tag{21}$$

Since $\widetilde{\Phi}_{01}^{\rho,u,j} \leq \widetilde{\Phi}_{01}^{\text{smth},u}$ by Lemma 5.3, we obtain:

$$\widetilde{\Phi}_{\text{def}}^{\text{smth},u}(r, g, m, z) = \sum_{j=0}^{J} \tau_j(g(x), m(x), z)\widetilde{\Phi}_{01}^{\text{smth},u}(r, x, j) \tag{22}$$

□

D.5. Proof Lemma 5.6

Lemma 5.6 ($\mathcal{R}$-consistency bounds for $\widetilde{\Phi}_{01}^{\rho,u,j}$)**.** *Assume $\mathcal{R}$ is symmetric and locally ρ-consistent. Then, for the agent set $\mathcal{A}$, any hypothesis $r \in \mathcal{R}$, and any distribution $\mathcal{P}$ with probabilities $p = (p_0, \cdots, p_J) \in \Delta^{|\mathcal{A}|}$, the following inequality holds:*

$$\begin{aligned}&\sum_{j\in\mathcal{A}} p_j \widetilde{\ell}_{01}^{j}(r, x, j) - \inf_{r\in\mathcal{R}} \sum_{j\in\mathcal{A}} p_j \widetilde{\ell}_{01}^{j}(r, x, j) \leq \\ &\Psi^u(1)\Big(\sum_{j\in\mathcal{A}} p_j \widetilde{\Phi}_{01}^{\rho,u,j}(r, x, j) - \inf_{r\in\mathcal{R}} \sum_{j\in\mathcal{A}} p_j \widetilde{\Phi}_{01}^{\rho,u,j}(r, x, j)\Big).\end{aligned}$$

Proof. We define the margin as $\rho_r(x, j) = r(x, j) - \max_{j'\neq j} r(x, j')$, which quantifies the difference between the score of the j-th dimension and the highest score among all other dimensions. Starting from this, we can define a space $\overline{\mathcal{R}}_\gamma(x) = \{r \in \mathcal{R} : \inf_{x'\in B_p(x,\gamma)} \rho_r(x', r(x)) > 0\}$ for $B_p(x, \gamma) = \{x' \mid \|x' - x\|_p \leq \gamma\}$ representing hypothesis that correctly classifies the adversarial input. By construction, we have that $x'_j \in B_p(x, \gamma)$.

In the following, we will make use of several notations. Let $p(x) = (p(x, 0), \dots, p(x, J))$ denote the probability distribution over $\mathcal{A}$ at point $x \in \mathcal{X}$. We sort these probabilities $\{p(x, j) : j \in \mathcal{A}\}$ in increasing order $p_{[0]}(x) \leq p_{[1]}(x) \leq \cdots \leq p_{[J]}(x)$. Let $\mathcal{R}$ be a hypothesis class for the rejector $r \in \mathcal{R}$ with $r : \mathcal{X} \times \mathcal{A} \to \mathbb{R}$. We assume this hypothesis class to be *symmetric* implying that for any permutation π of $\mathcal{A}$ and any $r \in \mathcal{R}$, the function r^π defined by $r^\pi(x, j) = r(x, \pi(j))$ is also in $\mathcal{R}$ for $j \in \mathcal{A}$. We similarly have $r \in \mathcal{R}$, such that $r(x, \{0\}_x^r), r(x, \{1\}_x^r), \cdots, r(x, \{J\}_x^r)$ sorting the scores $\{r(x, j) : j \in \mathcal{A}\}$ in increasing order.

For $\mathcal{R}$ symmetric and locally ρ-consistent, there exists $r^* \in \mathcal{R}$ with the same ordering of the $j \in \mathcal{A}$, regardless of any $x'_j \in B_p(x, \gamma)$. This implies $\inf_{x'_j\in B_p(x,\gamma)} |r^*(x'_j, q) - r^*(x'_j, q')| \geq \rho$ for $\forall q' \neq q \in \mathcal{A}$. Using the symmetry of $\mathcal{R}$, we can find a r^* with the same ordering of $j \in \mathcal{A}$, i.e. $p(x, \{k\}_x^{r^*}) = p_{[k]}(x)$ for any $k \in \mathcal{A}$:

$$\forall j \in \mathcal{A}, \quad r^*(x'_j, \{0\}_{x'_j}^{r^*}) \leq r^*(x'_j, \{1\}_{x'_j}^{r^*}) \leq \cdots \leq r^*(x'_j, \{J\}_{x'_j}^{r^*}) \tag{23}$$

We introduce a new notation $\xi'_k = x'_{\{k\}}$ corresponding to the k-th ordered adversarial input. For instance, if we have an ordered list $\{r^*(x'_2, 2), r^*(x'_0, 0), r^*(x'_1, 1)\}$, using the notation we have $\{r^*(\xi'_0, \{0\}_{\xi'_0}^{r^*}), r^*(\xi'_1, \{1\}_{\xi'_1}^{r^*}), r^*(\xi'_2, \{2\}_{\xi'_2}^{r^*})\}$.

We define a conditional risk $\mathcal{C}_{\widetilde{\Phi}_{01}^{\rho,u,j}}$ parameterized by the probability $p_j \in \Delta^{|\mathcal{A}|}$ along with its optimum:

$$\begin{aligned} \mathcal{C}_{\widetilde{\Phi}_{01}^{\rho,u,j}}(r,x) &= \sum_{j\in\mathcal{A}} p_j \widetilde{\Phi}_{01}^{\rho,u,j}(r,x,j) \\ \mathcal{C}^*_{\widetilde{\Phi}_{01}^{\rho,u,j}}(\mathcal{R},x) &= \inf_{r\in\mathcal{R}} \sum_{j\in\mathcal{A}} p_j \widetilde{\Phi}_{01}^{\rho,u,j}(r,x,j) \end{aligned} \tag{24}$$

The optimum $\mathcal{C}^*_{\widetilde{\Phi}_{01}^{\rho,u,j}}$ is challenging to characterize directly. To address this, we instead derive an upper bound by analyzing $\mathcal{C}_{\widetilde{\Phi}_{01}^{\rho,u,j}}(r^*,x)$. In what follows, the mapping from j to i is defined based on the rank of $p(x,j)$ within the sorted list $\{p_{[i]}(x)\}$.

$$\begin{aligned} \mathcal{C}^*_{\widetilde{\Phi}_{01}^{\rho,u,j}}(\mathcal{R},x) &\le \mathcal{C}_{\widetilde{\Phi}_{01}^{\rho,u,j}}(r^*,x) \\ &= \sum_{j\in\mathcal{A}} p(x,j) \sup_{x'_j\in B_p(x,\gamma)} \Psi^u\Bigg(\sum_{\substack{j'\in\mathcal{A}\\ j'\neq j}} \Psi_\rho\Big(r^*(x'_j,j)-r^*(x'_j,j')\Big)\Bigg) \\ &= \sum_{i=0}^{J} \sup_{\xi'_i\in B_p(x,\gamma)} p(x,\{i\}^{r^*}_{\xi'_i})\Psi^u\Bigg(\sum_{j=0}^{i-1}\Psi_\rho\Big(r^*(\xi'_i,\{i\}^{r^*}_{\xi'_i})-r^*(\xi'_i,\{j\}^{r^*}_{\xi'_i})\Big) \\ &\qquad + \sum_{j=i+1}^{J}\Psi_\rho\Big(r^*(\xi'_i,\{i\}^{r^*}_{\xi'_i})-r^*(\xi'_i,\{j\}^{r^*}_{\xi'_i})\Big)\Bigg) \\ &= \sum_{i=0}^{J} \sup_{\xi'_i\in B_p(x,\gamma)} p(x,\{i\}^{r^*}_{\xi'_i})\Psi^u\Bigg(\sum_{j=0}^{i-1}\Psi_\rho\Big(r^*(\xi'_i,\{i\}^{r^*}_{\xi'_i})-r^*(\xi'_i,\{j\}^{r^*}_{\xi'_i})\Big)+J-i\Bigg) \quad (\Psi_\rho(t)=1,\forall t\le 0) \\ &= \sum_{i=0}^{J} \sup_{\xi'_i\in B_p(x,\gamma)} p(x,\{i\}^{r^*}_{\xi'_i})\Psi^u(J-i) \quad (\Psi_\rho(v)=0,\forall v\ge\rho \text{ and } \inf_{x'_j\in B_p(x,\gamma)}|r^*(x'_j,q)-r^*(x'_j,q')|\ge\rho) \\ &= \sum_{i=0}^{J} p_{[i]}(x)\Psi^u(J-i) \quad (r^* \text{ and } p(x) \text{ same ordering of } j\in\mathcal{A}) \end{aligned} \tag{25}$$

Then, assuming $\overline{\mathcal{R}}_\gamma(x)\neq\varnothing$ and $\mathcal{R}$ symmetric, we have:

$$\begin{aligned} \Delta\mathcal{C}_{\widetilde{\Phi}_{01}^{\rho,u,j}}(r,x) &= \mathcal{C}_{\widetilde{\Phi}_{01}^{\rho,u,j}}(r,x)-\mathcal{C}^*_{\widetilde{\Phi}_{01}^{\rho,u,j}}(\mathcal{R},x) \ge \mathcal{C}_{\widetilde{\Phi}_{01}^{\rho,u,j}}(r,x)-\mathcal{C}_{\widetilde{\Phi}_{01}^{\rho,u,j}}(r^*,x) \\ &\ge \sum_{i=0}^{J} \sup_{\xi'_i\in B_p(x,\gamma)} p(x,\{i\}^{r}_{\xi'_i})\Psi^u\Bigg(\sum_{j=0}^{i-1}\Psi_\rho\Big(r(\xi'_i,\{i\}_{\xi'_i})-r(\xi'_i,\{j\}_{\xi'_i})\Big)+J-i\Bigg)-\Big(\sum_{i=0}^{J}p_{[i]}(x)\Psi^u(J-i)\Big) \end{aligned} \tag{26}$$

Then, for Ψ_ρ non negative, $\Psi_\rho(v) = 1$ for $v \leq 0$, and Ψ^u non-decreasing, we have that:

$$
\begin{aligned}
\Delta\mathcal{C}_{\widetilde{\Phi}_{01}^{\rho,u,j}}(r,x) &\geq \Psi^u(1)p(x,r(x))1_{r\notin\overline{\mathcal{R}}_\gamma(x)} + \sum_{i=0}^{J} \sup_{\xi_i'\in B_p(x,\gamma)} p(x,\{i\}^r_{\xi_i'})\Psi^u(J-i) - \Big(\sum_{i=0}^{J} p_{[i]}(x)\Psi^u(J-i)\Big) \\
&\geq \Psi^u(1)p(x,r(x))1_{r\notin\overline{\mathcal{R}}_\gamma(x)} - \sum_{i=0}^{J} p_{[i]}(x)\Psi^u(J-i) + \sum_{i=0}^{J} p(x,\{i\}^r_x)\Psi^u(J-i) \quad (\sup_{\xi_i'\in B_p(x,\gamma)} p(x,\{i\}^r_{\xi_i'}) \geq p(x,\{i\}^r_x)) \\
&= \Psi^u(1)p(x,r(x))1_{r\notin\overline{\mathcal{R}}_\gamma(x)} + \Psi^u(1)\left(\max_{j\in\mathcal{A}} p(x,j) - p(x,r(x))\right) + \begin{bmatrix}\Psi^u(1)\\ \Psi^u(1)\\ \Psi^u(2)\\ \vdots \\ \Psi^u(J)\end{bmatrix} \cdot \begin{bmatrix} p(x,\{J\}^r_x)\\ p(x,\{J-1\}^r_x)\\ p(x,\{J-2\}^r_x)\\ \vdots\\ p(x,\{0\}^r_x)\end{bmatrix} \\
&\quad - \begin{bmatrix}\Psi^u(1)\\ \Psi^u(1)\\ \Psi^u(2)\\ \vdots \\ \Psi^u(J)\end{bmatrix} \cdot \begin{bmatrix} p_{[J]}(x)\\ p_{[J-1]}(x)\\ p_{[J-2]}(x)\\ \vdots\\ p_{[0]}(x)\end{bmatrix}
\end{aligned}
$$

Rearranging terms for $\Psi^u(1) \leq \Psi^u(1) \leq \Psi^u(2) \leq \cdots \leq \Psi^u(J)$ and similarly for probabilities $p_{[J]}(x) \geq \cdots \geq p_{[0]}(x)$, leads to:

$$
\begin{aligned}
\Delta\mathcal{C}_{\widetilde{\Phi}_{01}^{\rho,u,j}}(r,x) &\geq \Psi^u(1)p(x,r(x))1_{r\notin\overline{\mathcal{R}}_\gamma(x)} + \Psi^u(1)\left(\max_{j\in\mathcal{A}} p(x,j) - p(x,r(x))\right) \\
&= \Psi^u(1)\left(\max_{j\in\mathcal{A}} p(x,j) - p(x,r(x))1_{r\in\overline{\mathcal{R}}_\gamma(x)}\right)
\end{aligned} \tag{27}
$$

for any $r \in \mathcal{R}$, we have:

$$
\begin{aligned}
\Delta\mathcal{C}_{\widetilde{\ell}_{01}^j}(r,x) &= \mathcal{C}_{\widetilde{\ell}_{01}^j}(r,x) - \mathcal{C}^B_{\widetilde{\ell}_{01}^j}(\mathcal{R},x) \\
&= \sum_{j\in\mathcal{A}} p(x,j) \sup_{x_j'\in B_p(x,\gamma)} 1_{\rho_r(x_j',j)\leq 0} - \inf_{r\in\mathcal{R}} \sum_{j\in\mathcal{A}} p(x,j) \sup_{x_j'\in B_p(x,\gamma)} 1_{\rho_r(x_j',j)\leq 0} \\
&= (1-p(x,r(x)))1_{r\in\overline{\mathcal{R}}_\gamma(x)} + 1_{r\notin\overline{\mathcal{R}}_\gamma(x)} - \inf_{r\in\mathcal{R}} \left[(1-p(x,r(x)))1_{r\in\overline{\mathcal{R}}_\gamma(x)} + 1_{r\notin\overline{\mathcal{R}}_\gamma(x)}\right] \\
&= (1-p(x,r(x)))1_{r\in\overline{\mathcal{R}}_\gamma(x)} + 1_{r\notin\overline{\mathcal{R}}_\gamma(x)} - \left(1 - \max_{j\in\mathcal{A}} p(x,j)\right) \quad (\mathcal{R} \text{ is symmetric and } \overline{\mathcal{R}}_\gamma(x) \neq \emptyset) \\
&= \max_{j\in\mathcal{A}} p(x,j) - p(x,r(x))1_{r\in\overline{\mathcal{R}}_\gamma(x)}
\end{aligned} \tag{28}
$$

We therefore have proven that:

$$
\begin{aligned}
\Delta\mathcal{C}_{\widetilde{\ell}_{01}^j}(r,x) &\leq \Psi^u(1)\Big(\Delta\mathcal{C}_{\widetilde{\Phi}_{01}^{\rho,u,j}}(r,x)\Big) \\
\sum_{j\in\mathcal{A}} p_j \widetilde{\ell}_{01}^j(r,x,j) - \inf_{r\in\mathcal{R}} \sum_{j\in\mathcal{A}} p_j \widetilde{\ell}_{01}^j(r,x,j) &\leq \Psi^u(1)\Big(\sum_{j\in\mathcal{A}} p_j \widetilde{\Phi}_{01}^{\rho,u,j}(r,x,j) - \inf_{r\in\mathcal{R}} \sum_{j\in\mathcal{A}} p_j \widetilde{\Phi}_{01}^{\rho,u,j}(r,x,j)\Big)
\end{aligned} \tag{29}
$$

□

D.6. Proof Theorem 5.7

Theorem 5.7 (($\mathcal{R}, \mathcal{G}$)-consistency bounds of $\widetilde{\Phi}_{\text{def}}^{\rho,u}$). *Let $\mathcal{R}$ be symmetric and locally ρ-consistent. Then, for the agent set $\mathcal{A}$, any hypothesis $r \in \mathcal{R}$, and any distribution $\mathcal{D}$, the following holds for a multi-task model $g \in \mathcal{G}$:*

$$\begin{aligned}&\mathcal{E}_{\widetilde{\ell}_{def}}(r,g) - \mathcal{E}^B_{\widetilde{\ell}_{def}}(\mathcal{R},\mathcal{G}) + \mathcal{U}_{\widetilde{\ell}_{def}}(\mathcal{R},\mathcal{G}) \leq \\ &\quad \Psi^u(1)\Big(\mathcal{E}_{\widetilde{\Phi}^{\rho,u}_{def}}(r) - \mathcal{E}^*_{\widetilde{\Phi}^{\rho,u}_{def}}(\mathcal{R}) + \mathcal{U}_{\widetilde{\Phi}^{\rho,u}_{def}}(\mathcal{R})\Big) \\ &\quad + \mathcal{E}_{c_0}(g) - \mathcal{E}^B_{c_0}(\mathcal{G}) + \mathcal{U}_{c_0}(\mathcal{G}).\end{aligned}$$

Proof. Using Lemma 5.2, we have:

$$\widetilde{\Phi}_{\text{def}}^{\rho,u}(r,g,m,z) = \sum_{j=0}^{J} \tau_j(g(x),m(x),z)\widetilde{\Phi}_{01}^{\rho,j}(r,x,j) \tag{30}$$

We define several important notations. For a quantity $\omega \in \mathbb{R}$, we note $\overline{\omega}(g(x),x) = \mathbb{E}_{y,t|x}[\omega(g,z=(x,y,t))]$, an optimum $\omega^*(z) = \inf_{g\in\mathcal{G}}[\omega(g,z)]$, and the combination $\overline{w}^*(x) = \inf_{g\in\mathcal{G}} \mathbb{E}_{y,t|x}[w(g,z)]$:

$$c_j^*(m_j(x),z) = \begin{cases} c_0^*(z) = \inf_{g\in\mathcal{G}}[c_0(g(x),z)] & \text{if } j=0 \\ c_j(m_j(x),z) & \text{otherwise}\end{cases} \tag{31}$$

Referring to (7), we have:

$$\tau_j^*(m(x),z) = \begin{cases} \tau_0(m(x),z) = \sum_{k=1}^J c_k(m_k,z) & \text{if } j=0 \\ \inf_{g\in\mathcal{G}}[\tau_j(g(x),m(x),z)] = c_0^*(z) + \sum_{k=1}^J c_k(m_k(x),z)1_{k\neq j} & \text{otherwise}\end{cases} \tag{32}$$

Next, we define the conditional risk $\mathcal{C}_{\widetilde{\ell}_{\text{def}}}$ associated to the adversarial true deferral loss.

$$\begin{aligned}\mathcal{C}_{\widetilde{\ell}_{\text{def}}}(r,g,x) &= \mathbb{E}_{y,t|x}\left[\sum_{j=0}^J \tau_j(g(x),m(x),z)\widetilde{\ell}_{01}^j(r,x,j) + (1-J)\sum_{j=0}^J c_j(g(x),m_j(x),z)\right] \\ &= \sum_{j=0}^J \overline{\tau}_j(g(x),m(x),x)\widetilde{\ell}_{01}^j(r,x,j) + (1-J)\sum_{j=0}^J \overline{c}_j(g(x),m_j(x),x)\end{aligned} \tag{33}$$

Now, we assume $r \in \mathcal{R}$ symmetric and define the space $\overline{\mathcal{R}}_\gamma(x) = \{r \in \mathcal{R} : \inf_{x'\in B_p(x,\gamma)} \rho_r(x', r(x)) > 0\}$. Assuming $\overline{\mathcal{R}}_\gamma(x) \neq \varnothing$, it follows:

$$\mathcal{C}_{\widetilde{\ell}_{\text{def}}}(r,g,x) = \sum_{j=0}^J \Big(\overline{\tau}_j(g(x),m(x),x)[1_{r(x)\neq j}1_{r\in\overline{\mathcal{R}}_\gamma(x)} + 1_{r\notin\overline{\mathcal{R}}_\gamma(x)}]\Big) + (1-J)\sum_{j=0}^J \overline{c}_j(g(x),m_j(x),x) \tag{34}$$

Intuitively, if $r \notin \mathcal{R}_\gamma(x)$, this means that there is no r that correctly classifies $x' \in B_p(x,\gamma)$ inducing an error of 1. It follows at the optimum:

$$\begin{aligned}
\mathcal{C}^B_{\widetilde{\ell}_{\text{def}}}(\mathcal{R},\mathcal{G},x) &= \inf_{g\in\mathcal{G},r\in\mathcal{R}}\Big[\sum_{j=0}^J\Big(\overline{\tau}_j(g(x),m(x),x)[1_{r(x)\neq j}1_{r\in\overline{\mathcal{R}}_\gamma(x)}+1_{r\notin\overline{\mathcal{R}}_\gamma(x)}]\Big)+(1-J)\sum_{j=0}^J\overline{c}_j(g(x),m_j(x),x)\Big]\\
&= \inf_{r\in\mathcal{R}}\Big[\sum_{j=0}^J\Big(\overline{\tau}^*_j(m(x),x)[1_{r(x)\neq j}1_{r\in\overline{\mathcal{R}}_\gamma(x)}+1_{r\notin\overline{\mathcal{R}}_\gamma(x)}]\Big)\Big]+(1-J)\sum_{j=0}^J\overline{c}^*_j(m_j(x),x)\\
&= \inf_{r\in\mathcal{R}}\sum_{j=0}^J\Big(\overline{\tau}^*_j(m(x),x)1_{r(x)\neq j}\Big)+(1-J)\sum_{j=0}^J\overline{c}^*_j(m_j(x),x)\quad (\overline{\mathcal{R}}_\gamma(x)\neq\varnothing,\text{ then }\exists r\in\overline{\mathcal{R}}_\gamma(x))\\
&= \sum_{j=0}^J\overline{\tau}^*_j(m(x),x)(1-\sup_{r\in\mathcal{R}}1_{r(x)=j})+(1-J)\sum_{j=0}^J\overline{c}^*_j(m_j(x),x)\\
&= \sum_{j=0}^J\overline{\tau}^*_j(m(x),x)-\max_{j\in\mathcal{A}}\overline{\tau}^*_j(m(x),x)+(1-J)\sum_{j=0}^J\overline{c}^*_j(m_j(x),x)
\end{aligned}\tag{35}$$

We can still work on making the last expression simpler:

$$\begin{aligned}
\sum_{j=0}^J\overline{\tau}^*_j(m(x),x) &= \sum_{j=1}^J\overline{c}_j(m_j(x),x)+\sum_{j=1}^J\Big(\overline{c}^*_0(x)+\sum_{k=1}^J\overline{c}_k(m_k(x),x)1_{k\neq j}\Big)\\
&= J\overline{c}^*_0(x)+\sum_{j=1}^J\Big(\overline{c}_j(m_j(x),x)+\sum_{k=1}^J\overline{c}_k(m_k(x),x)1_{k\neq j}\Big)\\
&= J\overline{c}^*_0(x)+\sum_{j=1}^J\Big(\overline{c}_j(m_j(x),x)+\sum_{k=1}^J\overline{c}_k(m_k(x),x)(1-1_{k=j})\Big)\\
&= J\overline{c}^*_0(x)+\sum_{j=1}^J\sum_{k=1}^J\overline{c}_k(m_k(x),x)\\
&= J\Big(\overline{c}^*_0(x)+\sum_{j=1}^J\overline{c}_j(m_j(x),x)\Big)
\end{aligned}\tag{36}$$

Then, reinjecting (36) in (35) gives:

$$\mathcal{C}^B_{\widetilde{\ell}_{\text{def}}}(\mathcal{R},\mathcal{G},x)=\sum_{j=0}^J\overline{c}^*_j(m_j(x),x)-\max_{j\in\mathcal{A}}\overline{\tau}^*_j(m(x),x)\tag{37}$$

if $j=0$:

$$\begin{aligned}
\mathcal{C}^B_{\widetilde{\ell}_{\text{def}}}(\mathcal{R},\mathcal{G},x) &= \sum_{j=0}^J\overline{c}^*_j(m_j(x),x)-\overline{\tau}_0(m(x),x)\\
&= \sum_{j=0}^J\overline{c}^*_j(m_j(x),x)-\sum_{j=1}^J\overline{c}_j(m_j(x),x)\\
&= \overline{c}^*_0(x)
\end{aligned}\tag{38}$$

if $j \neq 0$:

$$\begin{aligned}
\mathcal{C}^B_{\widetilde{\ell}_{\text{def}}}(\mathcal{R},\mathcal{G},x) &= \sum_{j=0}^J \overline{c}_j^*(m_j(x),x) - \overline{\tau}^*_{j>0}(m(x),x) \\
&= \sum_{j=0}^J \overline{c}_j^*(m_j(x),x) - \Big(\overline{c}_0^*(x) + \sum_{k=1}^J \overline{c}_k(m_k(x),x)1_{k\neq 1}\Big) \\
&= \overline{c}_{j>0}(m_j(x),x)
\end{aligned} \tag{39}$$

Therefore, it can be reduced to:

$$\mathcal{C}^B_{\widetilde{\ell}_{\text{def}}}(\mathcal{R},\mathcal{G},x) = \min_{j\in\mathcal{A}} \overline{c}_j^*(m_j(x),x) = \min_{j\in\mathcal{A}} \Big\{\overline{c}_0^*(x), \overline{c}_{j>0}(m_j(x),x)\Big\} \tag{40}$$

We can write the calibration gap as $\Delta\mathcal{C}_{\widetilde{\ell}_{\text{def}}}(r,g,x) := \mathcal{C}_{\widetilde{\ell}_{\text{def}}}(r,g,x) - \mathcal{C}^B_{\widetilde{\ell}_{\text{def}}}(\mathcal{R},\mathcal{G},x) \geq 0$, it follows:

$$\begin{aligned}
\Delta\mathcal{C}_{\widetilde{\ell}_{\text{def}}}(r,g,x) &= \mathcal{C}_{\widetilde{\ell}_{\text{def}}}(r,g,x) - \min_{j\in\mathcal{A}} \overline{c}_j^*(m_j(x),x) \\
&= \underbrace{\mathcal{C}_{\widetilde{\ell}_{\text{def}}}(r,g,x) - \min_{j\in\mathcal{A}} \overline{c}_j(g(x),m_j(x),x)}_{A} + \underbrace{\Big(\min_{j\in\mathcal{A}} \overline{c}_j(g(x),m_j(x),x) - \min_{j\in\mathcal{A}} \overline{c}_j^*(m_j(x),x)\Big)}_{B}
\end{aligned} \tag{41}$$

Term B: Let's first focus on B. We can write the following inequality:

$$B = \min_{j\in\mathcal{A}} \overline{c}_j(g(x),m_j(x),x) - \min_{j\in\mathcal{A}} \overline{c}_j^*(m_j(x),x) \leq \overline{c}_0(g(x),x) - \overline{c}_0^*(x) \tag{42}$$

Indeed, we have the following relationship:

1. if $\overline{c}_0(g(x),x) < \min_{j\in[J]} \overline{c}_j(m_j(x),x) \implies B = \overline{c}_0(g(x),x) - \overline{c}_0^*(x)$
2. if $\overline{c}_0(g(x),x) > \min_{j\in[J]} \overline{c}_j(m_j(x),x)$ and $\overline{c}_0^*(x) \leq \min_{j\in[J]} \overline{c}_j(m_j(x),x)$
$\implies B = \min_{j\in[J]} \overline{c}_j(m_j(x),x) - \overline{c}_0^*(x) \leq \overline{c}_0(g(x),x) - \overline{c}_0^*(x)$

Term A: Then using the term A:

$$\begin{aligned}
A &= \mathcal{C}_{\widetilde{\ell}_{\text{def}}}(r,g,x) - \min_{j\in\mathcal{A}} \overline{c}_j(m_j(x),x) \\
&= \mathcal{C}_{\widetilde{\ell}_{\text{def}}}(r,g,x) - \inf_{r\in\mathcal{R}} \mathcal{C}_{\widetilde{\ell}_{\text{def}}}(r,g,x) \\
&= \sum_{j=0}^J \Big(\overline{\tau}_j(g(x),m(x),x)\widetilde{\ell}^j_{01}(r,x,j)\Big) - \inf_{r\in\mathcal{R}} \sum_{j=0}^J \Big(\overline{\tau}_j(g(x),m(x),x)\widetilde{\ell}^j_{01}(r,x,j)\Big)
\end{aligned} \tag{43}$$

Now, we introduce a change of variables to define a probability distribution $p = (p_0, \cdots, p_j) \in \Delta^{|\mathcal{A}|}$, accounting for the fact that τ_j does not inherently represent probabilities. Consequently, for each $j \in \mathcal{A}$, we obtain the following expression:

$$p_j = \frac{\overline{\tau}_j(g(x),m(x),x)}{\sum_{j=0}^J \overline{\tau}_j(g(x),m(x),x)} = \frac{\overline{\tau}_j}{\|\boldsymbol{\tau}\|_1} \quad (\text{for } \boldsymbol{\tau} = \{\tau_j \geq 0\}_{j\in\mathcal{A}}) \tag{44}$$

We then, have:

$$A = \|\boldsymbol{\tau}\|_1 \Bigg(\sum_{j=0}^J \Big(p_j\widetilde{\ell}^j_{01}(r,x,j)\Big) - \inf_{r\in\mathcal{R}} \sum_{j=0}^J \Big(p_j\widetilde{\ell}^j_{01}(r,x,j)\Big)\Bigg) \tag{45}$$

Then, using Lemma 5.6, it leads to:

$$
\begin{aligned}
A &\leq \|\boldsymbol{\tau}\|_1 \Psi^u(1) \left[\sum_{j=0}^{J} \Big(p_j \widetilde{\Phi}_{01}^{\rho,u,j}(r,x,j) \Big) - \inf_{r\in\mathcal{R}} \sum_{j=0}^{J} \Big(p_j \widetilde{\Phi}_{01}^{\rho,u,j}(r,x,j) \Big) \right] \\
&= \|\boldsymbol{\tau}\|_1 \Psi^u(1) \frac{1}{\|\boldsymbol{\tau}\|_1} \left[\sum_{j=0}^{J} \Big(\overline{\tau}_j(g(x), m(x), x) \widetilde{\Phi}_{01}^{\rho,u,j}(r,x,j) \Big) - \inf_{r\in\mathcal{R}} \sum_{j=0}^{J} \Big(\overline{\tau}_j(g(x), m(x), x) \widetilde{\Phi}_{01}^{\rho,u,j}(r,x,j) \Big) \right] \\
&= \Psi^u(1) \left[\sum_{j=0}^{J} \Big(\overline{\tau}_j(g(x), m(x), x) \widetilde{\Phi}_{01}^{\rho,u,j}(r,x,j) \Big) - \inf_{r\in\mathcal{R}} \sum_{j=0}^{J} \Big(\overline{\tau}_j(g(x), m(x), x) \widetilde{\Phi}_{01}^{\rho,u,j}(r,x,j) \Big) \right] \\
&= \Psi^u(1) \Big[\mathcal{C}_{\widetilde{\Phi}_{\text{def}}^{\rho,u}}(r,x) - \mathcal{C}^*_{\widetilde{\Phi}_{\text{def}}^{\rho,u}}(\mathcal{R},x) \Big]
\end{aligned}
\tag{46}
$$

Then, adding B leads to:

$$
\begin{aligned}
\Delta\mathcal{C}_{\widetilde{\ell}_{\text{def}}}(r,g,x) &= A + B \quad \text{(using Eq. 41)} \\
&\leq \Psi^u(1) \Big[\mathcal{C}_{\widetilde{\Phi}_{\text{def}}^{\rho,u}}(r,x) - \mathcal{C}^*_{\widetilde{\Phi}_{\text{def}}^{\rho,u}}(\mathcal{R},x) \Big] + \overline{c}_0(g(x),x) - \overline{c}_0^*(x)
\end{aligned}
\tag{47}
$$

By construction, we have $\overline{c}_0(g(x),x) = \mathbb{E}_{y,t|x}[c_0(g(x),z)]$ with $c_0(g(x),z) = \psi(g(x),z)$ and $\overline{c}_0^*(x) = \inf_{g\in\mathcal{G}} \mathbb{E}_{y,t|x}[c_0(g(x),z)]$. Therefore, we can write for $g \in \mathcal{G}$ and the cost c_0:

$$
\Delta\mathcal{C}_{c_0}(g,x) = \overline{c}_0(g(x),x) - \overline{c}_0^*(x) \tag{48}
$$

Then,

$$
\begin{aligned}
\Delta\mathcal{C}_{\widetilde{\ell}_{\text{def}}}(r,g,x) &\leq \Psi^u(1) \Big[\mathcal{C}_{\widetilde{\Phi}_{\text{def}}^{\rho,u}}(r,x) - \mathcal{C}^*_{\widetilde{\Phi}_{\text{def}}^{\rho,u}}(\mathcal{R},x) \Big] + \Delta\mathcal{C}_{c_0}(g(x),x) \\
&= \Psi^u(1) \Big[\Delta\mathcal{C}_{\widetilde{\Phi}_{\text{def}}^{\rho,u}}(r,x) \Big] + \Delta\mathcal{C}_{c_0}(g,x)
\end{aligned}
\tag{49}
$$

Therefore, by definition:

$$
\begin{aligned}
\mathcal{E}_{\widetilde{\ell}_{\text{def}}}(r,g) - \mathcal{E}^*_{\widetilde{\ell}_{\text{def}}}(\mathcal{R},\mathcal{G}) + \mathcal{U}_{\widetilde{\ell}_{\text{def}}}(\mathcal{R},\mathcal{G}) &= \mathbb{E}_x[\Delta\mathcal{C}_{\widetilde{\ell}_{\text{def}}}(r,g,x)] \\
&\leq \Psi^u(1) \mathbb{E}_x \Big[\Delta\mathcal{C}_{\widetilde{\Phi}_{\text{def}}^{\rho,u}}(r,x) \Big] + \mathbb{E}_x \Big[\Delta\mathcal{C}_{c_0}(g(x),x) \Big] \\
&= \Psi^u(1) \Big(\mathcal{E}_{\widetilde{\Phi}_{\text{def}}^{\rho,u}}(r) - \mathcal{E}^*_{\widetilde{\Phi}_{\text{def}}^{\rho,u}}(\mathcal{R}) + \mathcal{U}_{\widetilde{\Phi}_{\text{def}}^{\rho,u}}(\mathcal{R}) \Big) \\
&\quad + \mathbb{E}_x \Big[\Delta\mathcal{C}_{c_0}(g(x),x) \Big] \\
&= \Psi^u(1) \Big(\mathcal{E}_{\widetilde{\Phi}_{\text{def}}^{\rho,u}}(r) - \mathcal{E}^*_{\widetilde{\Phi}_{\text{def}}^{\rho,u}}(\mathcal{R}) + \mathcal{U}_{\widetilde{\Phi}_{\text{def}}^{\rho,u}}(\mathcal{R}) \Big) \\
&\quad + \mathbb{E}_x[\overline{c}_0(g(x),x)] - \mathbb{E}_x[\overline{c}_0^*(x)] \\
&= \Psi^u(1) \Big(\mathcal{E}_{\widetilde{\Phi}_{\text{def}}^{\rho,u}}(r) - \mathcal{E}^*_{\widetilde{\Phi}_{\text{def}}^{\rho,u}}(\mathcal{R}) + \mathcal{U}_{\widetilde{\Phi}_{\text{def}}^{\rho,u}}(\mathcal{R}) \Big) \\
&\quad + \Delta\mathcal{E}_{c_0}(g)
\end{aligned}
\tag{50}
$$

where $\Delta\mathcal{E}_{c_0}(g) = \mathcal{E}_{c_0}(g) - \mathcal{E}^B_{c_0}(\mathcal{G}) + \mathcal{U}_{c_0}(\mathcal{G})$.

In the special case of the log-softmax ($u = 1$), we have that $\Psi^u(1) = \log(2)$.

□

E. Experiments details

We present empirical results comparing the performance of state-of-the-art Two-Stage Learning-to-Defer frameworks (Mao et al., 2023a; 2024d; Montreuil et al., 2024) with our robust SARD algorithm. To the best of our knowledge, this is the first study to address adversarial robustness within the context of Learning-to-Defer.

All baselines use the log-softmax surrogate for Φ_{01} with $\Psi^{u=1}(v) = \log(1+v)$ and $\Psi_e(v) = \exp(-v)$. Adversarial attacks and supremum evaluations over the perturbation region $B_p(x, \gamma)$ are evaluated using Projected Gradient Descent (Madry et al., 2017). For each experiment, we report the mean and standard deviation over four independent trials to account for variability in results. Experiments are conducted on one NVIDIA H100 GPU. Additionally, we make our scripts publicly available.

E.1. Multiclass Classification Task

Experts: We assigned categories to three distinct experts: expert M_1 is more likely to be correct on 58 categories, expert M_2 on 47 categories, and expert M_3 on 5 categories. To simulate a realistic scenario, we allow for overlapping expertise, meaning that for some $x \in \mathcal{X}$, multiple experts can provide correct predictions. On assigned categories, an expert has a probability $p = 0.94$ to be correct, while following a uniform probability if the category is not assigned.

Agent costs are defined as $c_0(h(x), y) = \ell_{01}(h(x), y)$ for the model and $c_{j>0}(m_j^h(x), y) = \ell_{01}(m_j^h(x), y)$, consistent with (Mozannar & Sontag, 2020; Mozannar et al., 2023; Verma et al., 2022; Cao et al., 2024; Mao et al., 2023a). We report respective accuracies of experts in Table 4.

Model: We train the classifier offline using a ResNet-4 architecture (He et al., 2015) for 100 epochs with the Adam optimizer (Kingma & Ba, 2017), a learning rate of 0.1, and a batch size of 64. The checkpoint corresponding to the lowest empirical risk on the validation set is selected. Corresponding performance is indicated in Table 4.

	Model	Expert M_1	Expert M_2	Expert M_3
Accuracy	61.0	53.9	45.1	5.8

Table 4. Agent accuracies on the CIFAR-100 validation set. Since the training and validation sets are pre-determined in this dataset, the agents' knowledge remains fixed throughout the evaluation.

Baseline (Mao et al., 2023a): We train a rejector using a ResNet-4 (He et al., 2015) architecture for 500 epochs, a learning rate of 0.005, a cosine scheduler, Adam optimizer (Kingma & Ba, 2017), and a batch size of 2048. We report performance of the checkpoints corresponding to the lower empirical risk on the validation set.

SARD: We train a rejector using the ResNet-4 architecture (He et al., 2015) for 1500 epochs with a learning rate of 0.005, a cosine scheduler, the Adam optimizer (Kingma & Ba, 2017) with L2 weight decay of 10^{-4} acting as regularizer, and a batch size of 2048. The hyperparameters are set to $\rho = 1$ and $\nu = 0.01$. The supremum component from the adversarial inputs is estimated using PGD40 (Madry et al., 2017) with $\epsilon = 8/255$, the ℓ_∞ norm, and a step size of $\epsilon/40$, following the approach in (Mao et al., 2023b; Awasthi et al., 2023).

Baseline	Clean	Untarg.	Targ. Model	Targ. M_1	Targ. M_2	Targ. M_3
Mao et al. (2023a)	72.8 ± 0.4	17.2 ± 0.2	61.1 ± 0.1	54.4 ± 0.1	45.4 ± 0.1	13.4 ± 0.1
Our	67.0 ± 0.4	49.8 ± 0.3	64.8 ± 0.2	62.4 ± 0.3	62.1 ± 0.2	64.8 ± 0.3

Table 5. Comparison of accuracy results between the proposed SARD and the baseline (Mao et al., 2023a) on the CIFAR-100 validation set, including clean and adversarial scenarios.

E.2. Regression Task

Experts: We train three experts offline with three layers MLPs (128, 64, 32), each specializing in a specific subset of the dataset based on a predefined localization criterion. The first expert M_1 trains on Southern California (latitude lower than 36), the second expert M_2 on Central California (latitude between 36 and 38.5), and the last in Northern California (otherwise) representing a smaller area. MLPs are trained using a ReLU, an Adam optimizer (Kingma & Ba, 2017), a learning rate of 0.001, and 500 epochs. Agent costs are defined as $c_0(f(x), t) = \text{RMSE}(g(x), t)$ for the model and $c_{j>0}(m_j^f(x), t) = \text{RMSE}(m_j(x), t)$, consistent with (Mao et al., 2024d). We report respective RMSE of experts in Table E.2.

Model: We train a regressor using a two-layer MLP with hidden dimensions (64, 32) on the full training set. The model uses ReLU activations, the Adam optimizer (Kingma & Ba, 2017), a learning rate of 0.001, and is trained for 500 epochs. We report performance of the checkpoints corresponding to the lower empirical risk on the validation set. The model performance is reported in Table E.2.

	Model	Expert M_1	Expert M_2	Expert M_3
RMSE	$0.27 \pm .01$	$1.23 \pm .02$	$1.85 \pm .02$	$0.91 \pm .01$

Table 6. Agent RMSE on the California Housing validation set (20% of the dataset).

Baseline (Mao et al., 2024d): We train a rejector using a MLP (8,16) for 100 epochs, a learning rate of 0.01, a cosine scheduler, Adam optimizer (Kingma & Ba, 2017), and a batch size of 8096. We report performance of the checkpoints corresponding to the lower empirical risk on the validation set.

SARD: We train a rejector using a MLP (8,16) for 400 epochs, a learning rate of 0.01, a cosine scheduler, Adam optimizer (Kingma & Ba, 2017) with L2 weight decay of 10^{-4} acting as regularizer, and a batch size of 8096. The hyperparameters are set to $\rho = 1$ and $\nu = 0.05$. The supremum component from the adversarial inputs is estimated using PGD10 (Madry et al., 2017) with ϵ equal to 25% of the variance of dataset's features, the ℓ_∞ norm, and a step size of $\epsilon/10$, following the approach in (Mao et al., 2023b; Awasthi et al., 2023).

Baseline	Clean	Untarg.	Targ. Model	Targ. M_1	Targ. M_2	Targ. M_3
Mao et al. (2024d)	0.17 ± 0.01	0.29 ± 0.3	0.19 ± 0.01	0.40 ± 0.02	0.21 ± 0.01	0.41 ± 0.05
Our	0.17 ± 0.01	0.17 ± 0.01	0.17 ± 0.01	0.18 ± 0.01	0.18 ± 0.01	0.18 ± 0.01

Table 7. Performance comparison of SARD with the baseline (Mao et al., 2024d) on the California Housing dataset. The table reports Root Mean Square Error (RMSE) under clean and adversarial scenarios.

E.3. Multi Task

Experts: We train two specialized experts using a Faster R-CNN (Ren et al., 2016) architecture with a MobileNet (Howard et al., 2017) backbone. The first expert, M_1, is trained on images containing *animals*, while the second expert, M_2, is trained on images containing *vehicles*. Both experts are trained using the Adam optimizer (Kingma & Ba, 2017) with a learning rate of 0.005, a batch size of 128, and trained for 50 epochs. Agent costs are defined as $c_0(g(x), z) = \text{mAP}(g(x), z)$ for the model and $c_{j>0}(m_j(x), z) = \text{mAP}(m_j(x), z)$, consistent with (Montreuil et al., 2024). We report respective mAP of experts in Table E.3.

Model: We train an object detection model using a larger Faster R-CNN (Ren et al., 2016) with ResNet-50 FPN (He et al., 2015) backbone. We train this model with Adam optimizer (Kingma & Ba, 2017), a learning rate of 0.005, a batch size of 128, and trained for 50 epochs. We report performance of the checkpoints corresponding to the lower empirical risk on the validation set. The model performance is reported in Table E.3.

	Model	Expert M_1	Expert M_2
mAP	39.5	17.2	20.0

Table 8. Agents mAP Pascal VOC validation set. Since the training and validation sets are pre-determined in this dataset, the agents' knowledge remains fixed throughout the evaluation.

Baseline (Montreuil et al., 2024): We train a rejector using a Faster R-CNN (Ren et al., 2016) with a MobileNet backbone (Howard et al., 2017) and a classification head. We train this rejector for 70 epochs, a learning rate $5e^{-4}$, a cosine scheduler, Adam optimizer (Kingma & Ba, 2017), and a batch size of 256. We report performance of the checkpoints corresponding to the lower empirical risk on the validation set.

SARD: We train a rejector using a Faster R-CNN (Ren et al., 2016) with a MobileNet backbone (Howard et al., 2017) and a classification head for 70 epochs with a learning rate of 0.001, a cosine scheduler, the Adam optimizer (Kingma & Ba, 2017) with L2 weight decay of 10^{-4} acting as regularizer, and a batch size of 64. The hyperparameters are set to $\rho = 1$ and $\nu = 0.01$. The supremum component from the adversarial inputs is estimated using PGD20 (Madry et al., 2017) with $\epsilon = 8/255$, the ℓ_∞ norm, and a step size of $\epsilon/20$, following the approach in (Mao et al., 2023b; Awasthi et al., 2023).

Baseline	Clean	Untarg.	Targ. Model	Targ. M_1	Targ. M_2
Montreuil et al. (2024)	44.4 ± 0.4	9.7 ± 0.1	39.5 ± 0.1	17.4 ± 0.2	20.4 ± 0.2
Our	43.9 ± 0.4	39.0 ± 0.3	39.5 ± 0.1	39.7 ± 0.3	39.6 ± 0.1

Table 9. Performance comparison of SARD with the baseline (Montreuil et al., 2024) on the Pascal VOC dataset. The table reports mean Average Precision (mAP) under clean and adversarial scenarios.