Keeping up with dynamic attackers: Certifying robustness to adaptive online data poisoning

Avinandan Bose
University of Washington
avibose@cs.washington.edu

Laurent Lessard
Northeastern University
l.lessard@northeastern.edu

Maryam Fazel
University of Washington
mfazel@uw.edu

Krishnamurthy Dj Dvijotham
ServiceNow Research
dvij@cs.washington.edu

Abstract

The rise of foundation models fine-tuned on human feedback from potentially untrusted users has increased the risk of adversarial data poisoning, necessitating the study of robustness of learning algorithms against such attacks. Existing research on provable certified robustness against data poisoning attacks primarily focuses on certifying robustness for static adversaries who modify a fraction of the dataset used to train the model *before the training algorithm is applied*. In practice, particularly when learning from human feedback in an online sense, adversaries can observe and react to the learning process and inject poisoned samples that optimize adversarial objectives better than when they are restricted to poisoning a static dataset once, before the learning algorithm is applied. Indeed, it has been shown in prior work that online dynamic adversaries can be significantly more powerful than static ones. We present a novel framework for computing certified bounds on the impact of dynamic poisoning, and use these certificates to design robust learning algorithms. We give an illustration of the framework for the mean estimation and binary classification problems and outline directions for extending this in further work. The code to implement our certificates and replicate our results is available at `https://github.com/Avinandan22/Certified-Robustness`.

Proceedings of the 28[th] International Conference on Artificial Intelligence and Statistics (AISTATS) 2025, Mai Khao, Thailand. PMLR: Volume 258.

1 INTRODUCTION

With the advent of foundation models fine tuned using human feedback gathered from potentially untrusted users (for example, users of a publicly available language model) [Christiano et al., 2017, Ouyang et al., 2022, Bose et al., 2024c], the potential for adversarial or malicious data entering the training data of a model increases substantially. This motivates the study of robustness of learning algorithms to poisoning attacks [Biggio et al., 2012]. More recently, there have been works that attempt to achieve "certified robustness" to data poisoning, i.e., proving that the worst case impact of poisoning is below a certain bound that depends on parameters of the learning algorithm. All the work in this space, to the best of our knowledge, focuses on the *static* poisoning adversary [Steinhardt et al., 2017, Zhang et al., 2022]. Even in [Wang and Feizi, 2024] which is the closest setting to our work, the poisoning adversary acts over offline datasets in a temporally extended fashion which are poisoned in one shot, and thus is not dynamic. There has been work on *dynamic* attack algorithms [Zhang et al., 2020, Wang and Chaudhuri, 2018] showing that these attacks can indeed be more powerful than static attacks. This motivates the question we study: can we obtain certificates of robustness for a broad class of learning algorithms against *dynamic* poisoning adversaries?

In this paper, we study learning algorithms corrupted by a dynamic poisoning adversary who can observe the behavior of the algorithm and adapt the poisoning in response. This is relevant in scenarios where models are continuously/periodically updated in the face of new feedback, as is common in RLHF/fine tuning applications (see Figure 1). We provide (to the best of our knowledge) the first general framework for computing certified bounds on the worst case impact of a dynamic data poisoning attacker, and further, use this certificate to design robust learning algorithms (see Section 2). We give an illustration of the framework for the mean estimation problem (see Section 3) and binary classification problem (see Section 4) and suggest

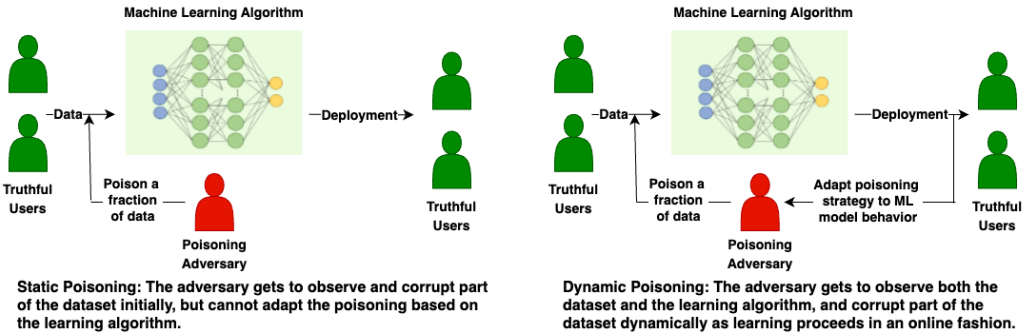

Figure 1: A schematic diagram to highlight the differences between static and dynamic poisoning.

directions for future work to apply the framework to other practical learning scenarios. Our contributions are as follows:

1. We develop a framework for computing certified bounds on the worst case impact of a dynamic online poisoning adversary on a learning algorithm as a finite dimensional optimization problem. The framework applies to an arbitrary learning algorithm and a general adversarial formulation described in Section 2. However, instantiating the framework in a computationally tractable way requires additional work, and we show that this instantiation can be done for certain cases.

2. We demonstrate that for learning algorithms designed for mean estimation (Section 3) and binary classification problems with linear classifiers (Section 4), we can tractably compute bounds (via dual certificates) for learning algorithms that use either regularization or noise addition as a defense against data poisoning. We leave extensions to broader learning algorithms to future work.

3. We use these certificates to choose parameters of a learning algorithm so as to trade off performance and robustness, and thereby derive robust learning algorithms (Section 2.2).

4. We conduct experiments on real and synthetic datasets to empirically validate our certificates of robustness, as well as using the meta-learning setup to design defenses (Section 5).

1.1 Related Work

Data Poisoning. Modern machine learning pipelines involve training on massive, uncurated datasets that are potentially untrustworthy and of such scale that conducting rigorous quality checks becomes impractical. Poisoning attacks [Biggio et al., 2012, Newsome et al., 2006, Biggio and Roli, 2018] pose big security concerns upon deployment of ML models. Depending on which stage (training / deployment) the poisoning takes place, they can be characterised as follows: 1. Static attacks: The model is trained on an offline dataset with poisoned data. Attacks could be untargeted, which aim to prevent training convergence rendering an unusable model and thus denial of service [Tian et al., 2022], or targeted, which are more task-specific and instead of simply increasing loss, attacks of this kind seeks to make the model output wrong predictions on specific tasks. 2. Backdoor attacks: In this setting, the test / deployment time data can be altered [Chen et al., 2017, Gu et al., 2017, Han et al., 2022, Zhu et al., 2019]. Attackers manipulate a small proportion of the data such that, when a specific pattern / trigger is seen at test-time, the model returns a specific, erroneous prediction. 3. Dynamic (and adaptive) attacks: In scenarios where models are continuously/periodically updated in the face of new feedback, as is common in RLHF/fine tuning applications [Bose et al., 2024a], a dynamic poisoning adversary [Wang and Chaudhuri, 2018], [Zhang et al., 2020] can observe the behavior of the learning algorithm and adapt the poisoning in response.

Certified Poisoning Defense. Recently, there have been works that attempt to achieve "certified robustness" to data poisoning, i.e., proving that the worst case impact of *any* poisoning strategy is below a certain bound that depends on parameters of the learning algorithm. All the work in this space, to the best of our knowledge, focuses on the *static* or *backdoor* attack adversary. [Steinhardt et al., 2017] provide certificates for linear models trained with gradient descent, [Rosenfeld et al., 2020] present a statistical upper-bound on the effectiveness of ℓ_2 perturbations on training labels for linear models using randomized smoothing, [Zhang et al., 2022, Sosnin et al., 2024] present a model-agnostic certified approach that can effectively defend against both trigger-less and backdoor attacks, [Xie et al., 2022] observe that differential privacy, which usually covers addition or removal of data points, can also provide statistical guarantees in

Attack Type	Adversary adapts poisoning strategy upon observing model behavior	Adversary can poison data for deployed model	Certified robustness
Static / One-shot	✗	✗	✓
Backdoor	✗	✓	✓
Dynamic attack only	✓	✓	✗
Dynamic attack & defense (Ours)	✓	✓	✓

Table 1: A comparison with lines of work closest to ours. Static/One-shot ([Tian et al., 2022](), [Steinhardt et al., 2017](), [Rosenfeld et al., 2020]()), Backdoor([Chen et al., 2017](), [Gu et al., 2017](), [Han et al., 2022](), [Zhu et al., 2019](), [Zhang et al., 2022](), [Sosnin et al., 2024]()), Dynamic attack only ([Wang and Chaudhuri, 2018](), [Zhang et al., 2020]().A detailed description is provided in Section 1.1.

some limited poisoning settings. Even in [Wang and Feizi, 2024]() which is the closest setting to our work, the poisoning adversary acts over offline datasets in a temporally extended fashion which are poisoned in one shot, and thus is not dynamic.

2 PROBLEM SETUP

Notation	Interpretation	Belongs to
$\boldsymbol{\theta}$	Model Parameters	$\boldsymbol{\Theta}$
$\boldsymbol{\phi}$	Hyper-parameters	$\boldsymbol{\Phi}$
$\boldsymbol{z}$	Data point	$\boldsymbol{Z}$
F_ϕ	Update rule	$\boldsymbol{\Theta} \times \boldsymbol{Z} \mapsto \boldsymbol{\Theta}$
$\boldsymbol{z}^{\text{adv}}$	Adversarial data point	$\mathcal{A}$
ℓ_{adv}	Adversarial objective fn.	$\boldsymbol{\Theta} \mapsto \mathbb{R}$
$\mathbb{P}^{\text{data}}$	Benign Data Dist.	$\mathcal{P}[\boldsymbol{Z}]$
$\Pi(\cdot\|\boldsymbol{\theta}, \boldsymbol{z}^{\text{adv}})$	State Transition Kernel	$\boldsymbol{\Theta} \times \boldsymbol{Z} \mapsto \mathcal{P}[\boldsymbol{\Theta}]$

Table 2: Notation.

We now develop the exact problem setup that we study in the paper. We consider a learning algorithm aimed at estimating parameters $\boldsymbol{\theta} \in \Theta$, and each step of the learning algorithm updates the estimates of these parameters based on potentially poisoned data. The following components fully define the problem setup.

Online learning algorithm. We consider online learning algorithms that operate by receiving a new datapoint at each step and making an update to model parameters being estimated. In particular, we consider learning algorithms that can be written as

$$\boldsymbol{\theta}_{t+1} \leftarrow F_\phi\Big(\underbrace{\boldsymbol{\theta}_t}_{\text{Parameter}}, \underbrace{\boldsymbol{z}_t}_{\text{Datapoint}} \Big), \tag{1}$$

where $F_\phi : \boldsymbol{\Theta} \times \boldsymbol{Z} \to \boldsymbol{\Theta}$ is a parameterized function that maps the current model parameters $\boldsymbol{\theta}_t$ to new model parameters $\boldsymbol{\theta}_{t+1}$, based on the received datapoint $\boldsymbol{z}_t$, where $\phi \in \boldsymbol{\Phi}$ is a hyperparameter, for example, learning rate in a gradient based learning algorithm, or strength of regularization used in the objective function.

Example To illustrate the setup we consider a simple toy example where we try to estimate the mean of the datapoints via gradient descent on the ℓ_2 regularized squared Euclidean loss. Given a current estimate $\boldsymbol{\theta}_t$, upon receiving a datapoint $\boldsymbol{z}_t$, the update step can be written as:

$$\begin{aligned}\boldsymbol{\theta}_{t+1} &= \boldsymbol{\theta}_t - \eta \nabla\big(\tfrac{1}{2}\|\boldsymbol{z}_t - \boldsymbol{\theta}_t\|_2^2 + \tfrac{\sigma}{2}\|\boldsymbol{\theta}_t\|_2^2\big) \\ &= (1 - \eta - \eta\sigma)\boldsymbol{\theta}_t + \eta \boldsymbol{z}_t \\ &= F_\phi(\boldsymbol{\theta}_t, \boldsymbol{z}_t).\end{aligned}$$

Here $\phi = \{\eta, \sigma\}$ denotes the learning rate and regularization parameter, and are the hyperparameters of the learning algorithm. Note that F_ϕ is a general update rule and we do not make any assumptions about F_ϕ.

Poisoned learning algorithm. We consider a setting where some of the data points received by the learning algorithm are corrupted by an adversary, who is allowed to choose corruptions as a function of the entire trajectory of the learning algorithm up to that point. We refer to such an adversary, who can observe the full trajectory and decide on the next corruption accordingly, as a *dynamic adaptive adversary*. While this may seem unrealistic, since our goal here is to compute certified bounds on the worst case adversary, we refrain from placing informational constraints on the adversary, as an adversary with sufficient side knowledge can still infer hidden parameters of the model from even from just a prediction API [Tramèr et al., 2016]().

The adversary is restricted to select a corrupted data point $\boldsymbol{z}_t^{\text{adv}} \in \mathcal{A}$, which reflects constraints such as input feature normalization or the adversary trying to avoid outlier detection mechanisms used by the learner. We make no additional assumptions about the specific poisoning strategy employed by the adversary. Thus, our certificates of robustness to poisoning apply to *any dynamic adaptive adversary who chooses poisoned data points from the set* $\mathcal{A}$.

We assume that with a fixed probability, the data point the algorithm receives at each time step is poisoned. In practice, this could reflect the situation

that out of a large population of human users providing feedback to a learning system, a small fraction are adversarial and will provide poisoned feedback. Let $\mathbb{P}^{\text{data}}$ denote the benign distribution of data points. Mathematically, the data point $\boldsymbol{z}_t$ received by the learning algorithm at time t is sampled according to $\boldsymbol{z}_t \sim \epsilon\delta(\boldsymbol{z}_t^{\text{adv}}) + (1-\epsilon)\mathbb{P}^{\text{data}}$, where $\delta(\cdot)$ denotes the Dirac delta function, and ϵ is a parameter that controls the "level" of poisoning (analogous to the fraction of poisoned samples in static poisoning settings [Steinhardt et al., 2017]). This is a special case of Huber's contamination model, which is used in the robust statistics literature [Diakonikolas and Kane, 2023] with the contamination model being a Dirac distribution. For compactness of the data generation process we define the following:

$$\mathbb{P}_\epsilon(\boldsymbol{z}^{\text{adv}}) := \epsilon\delta(\boldsymbol{z}^{\text{adv}}) + (1-\epsilon)\mathbb{P}^{\text{data}}. \tag{2}$$

2.1 Adversarial Objective

Transition Kernel. Starting with a parameter estimate $\boldsymbol{\theta}$, the adversary chooses $\boldsymbol{z}^{\text{adv}}$, then the learning algorithm updates the parameter estimate (via F_ϕ). The transition kernel gives the probability (or probability density) that the parameter estimate assumes a value $\boldsymbol{\theta}'$ after the above steps, and is defined as (recall that $\delta(\cdot)$ denotes the Dirac delta function):

$$\Pi(\boldsymbol{\theta}'|\boldsymbol{\theta}, \boldsymbol{z}^{\text{adv}}) = \mathop{\mathbb{E}}_{\boldsymbol{z}\sim\mathbb{P}_\epsilon(\boldsymbol{z}^{\text{adv}})}\left[\delta(F_\phi(\boldsymbol{\theta}, \boldsymbol{z}) - \boldsymbol{\theta}')\right]. \tag{3}$$

Dynamics as a Markov Process. The dynamics in Eq. (1) gives rise to a Markov process over the parameters $\boldsymbol{\theta}$. If $\mathbb{P}_t$ denotes the distribution over parameters at time t, we have

$$\mathbb{P}_{t+1}(\boldsymbol{\theta}') = \int \Pi(\boldsymbol{\theta}'|\boldsymbol{\theta}, \boldsymbol{z}^{\text{adv}})\mathbb{P}_t(\boldsymbol{\theta})\, d\boldsymbol{\theta}. \tag{4}$$

Since the learning algorithm (dynamics of the parameters) is a Markov process, the sequence of actions for the adversary (i.e., choices of $\boldsymbol{z}^{\text{adv}}$) constitute a Markov Decision Process with

$$\underbrace{\boldsymbol{\theta}}_{\text{States}}, \underbrace{\boldsymbol{z}^{\text{adv}}}_{\text{Actions}}, \underbrace{\Pi(\cdot|\boldsymbol{\theta}, \boldsymbol{z}^{\text{adv}})}_{\text{Transition Kernel}}.$$

Adversarial objective function. We assume that the poisoning adversary is interested in maximizing some adversarial objective $\ell_{\text{adv}} : \boldsymbol{\Theta} \mapsto \mathbb{R}$, for example, the expected prediction error on some target distribution of interest to the adversary. The adversary wants to choose actions such that it can maximize its average reward over time. Utilizing the fact that the optimal policy for an MDP is stationary (i.e., the policy is time invariant), we define the adversary's objective for an arbitrary stationary policy $\boldsymbol{z}^{\text{adv}} \sim \pi(\cdot|\boldsymbol{\theta})$ as follows:

$$\rho(\pi) = \lim_{T\to\infty} \frac{1}{T}\mathop{\mathbb{E}}_{\pi}\left[\sum_{t=1}^{T} \ell_{\text{adv}}(\boldsymbol{\theta}_t)\right], \tag{5}$$

where the expectation is with respect to the noisy state transition dynamics induced by the adversary's poisoning policy π.

We utilize the fact that $\rho(\pi)$ is equal to the expected reward under the *stationary state distribution* (assuming the MDP is ergodic, see details in Appendix A):

$$\rho(\pi) = \mathop{\mathbb{E}}_{\boldsymbol{\theta}\sim d_\pi(\cdot)}\left[\ell_{\text{adv}}(\boldsymbol{\theta})\right].$$

The stationary state is defined as a condition where the distribution of parameters remains unchanged over time. In other words, the distribution of parameters at any given point in the stationary state is identical to the distribution of parameters at the next state.

The stationarity condition can be expressed mathematically in terms of the transition kernel as:

$$\mathop{\mathbb{E}}_{\substack{\boldsymbol{\theta}\sim d_\pi(\cdot)\\ \boldsymbol{z}^{\text{adv}}\sim\pi(\cdot|\boldsymbol{\theta})}}\left[\Pi(\boldsymbol{\theta}'|\boldsymbol{\theta}, \boldsymbol{z}^{\text{adv}})\right] = d_\pi(\boldsymbol{\theta}') \quad \forall\boldsymbol{\theta}' \in \Theta.$$

Given a family of learning algorithms F_ϕ with tunable parameters $\phi \in \boldsymbol{\Phi}$, our goal is to estimate ϕ so that our learning algorithm is robust to the poisoning as described above. However, since we assume that we are working in the online setting, it is seldom the case that we know the data distribution $\mathbb{P}^{\text{data}}$ in advance, making the adversary's objective intractable. In Section 2.2, we use a meta learning formulation to overcome the lack of knowledge about $\mathbb{P}^{\text{data}}$ in advance.

2.2 Meta-learning a robust learning algorithm

In a meta learning setup [Hochreiter et al., 2001, Andrychowicz et al., 2016, Bose et al., 2024b], we suppose that we have access to a meta-distribution from which data distributions can be sampled. In such a setup, we can "simulate" various data distributions and consider the following approach: We take a family of learning algorithms F_ϕ with tunable parameters $\phi \in \boldsymbol{\Phi}$, and attempt to design the parameters ϕ of the learning algorithm to trade-off performance and robustness in expectation over the data distributions sampled from the meta-distribution.

In particular, in the absence of poisoned data, the updates (1) on data sampled from benign data distribution $\mathbb{P}^{\text{data}}$ result in a stationary distribution over

model parameters $\boldsymbol{\theta}$ denoted by $\mathbb{P}(\phi, \mathbb{P}^{\text{data}})$. The expected benign target loss can be written as :

$$b(\phi, \mathbb{P}^{\text{data}}) = \mathop{\mathbb{E}}_{\boldsymbol{\theta} \sim \mathbb{P}(\phi, \mathbb{P}^{\text{data}})} \left[\ell \left(\boldsymbol{\theta} \right) \right], \tag{6}$$

where $\ell : \boldsymbol{\Theta} \mapsto \mathbb{R}$ is the loss the learning algorithm wants to minimize.

In Section 2.3, we propose a general formulation to derive an upper bound on the worst case impact of an adversary on the target loss (a certificate), which we denote by $c(\phi, \mathbb{P}^{\text{data}})$ for a given data distribution $\mathbb{P}^{\text{data}}$ and parameter of the learning algorithm $\phi \in \boldsymbol{\Phi}$.

Given a meta distribution $\mathcal{P}$, we can propose the following criterion to design a robust learning algorithm:

$$\inf_{\phi \in \boldsymbol{\Phi}} \mathbb{E}_{\mathbb{P}^{\text{data}} \sim \mathcal{P}} \Big[b(\phi, \mathbb{P}^{\text{data}}) + \kappa \cdot c(\phi, \mathbb{P}^{\text{data}}) \Big]. \tag{7}$$

where $\kappa > 0$ is a trade-off parameter. The expectation over $\mathcal{P}$ is a meta-learning inspired formulation, where we are designing a learning algorithm that is good "in expectation" under a meta-distribution over data distributions. The first term constitutes "doing well" in the absence of the adversary by converging to a stationary distribution over parameters that incurs low expected loss. The second term is an upper bound on the worst case loss incurred by the learning algorithm in the presence of the adversary.

2.3 Technical Approach: Certificate of Robustness

For a given $\mathbb{P}^{\text{data}} \sim \mathcal{P}$, we attempt to find an upper bound on the worst case impact of the adversary. Recalling that the sequence of actions for the adversary constitutes a Markov Decision Process, the value of the adversarial objective for the adversary's optimal action sequence is therefore the average reward in the infinite horizon Markov Decision Process setting [Malek et al., 2014] and can be written as the solution of an *infinite dimensional* linear program (LP) [Puterman, 2014]. The LP can be written as:

$$\begin{aligned} &\sup_{\substack{d_\pi \in \mathcal{P}[\boldsymbol{\Theta}] \\ \pi \in \mathcal{P}[\boldsymbol{\Theta} \times \boldsymbol{Z}]}} \mathop{\mathbb{E}}_{\boldsymbol{\theta} \sim d_\pi(\cdot)} \left[\ell_{\text{adv}} \left(\boldsymbol{\theta} \right) \right], \quad \text{subject to} \\ &\mathop{\mathbb{E}}_{\substack{\boldsymbol{\theta} \sim d_\pi(\cdot) \\ \boldsymbol{z}^{\text{adv}} \sim \pi(\cdot | \boldsymbol{\theta})}} \left[\Pi(\boldsymbol{\theta}' | \boldsymbol{\theta}, \boldsymbol{z}^{\text{adv}}) \right] = d_\pi(\boldsymbol{\theta}') \quad \forall \boldsymbol{\theta}' \in \Theta, \end{aligned} \tag{8}$$

where $\mathcal{P}[\boldsymbol{\Theta}]$, $\mathcal{P}[\boldsymbol{\Theta} \times \boldsymbol{Z}]$ denote the space of probability measures on $\boldsymbol{\Theta}$ and $\boldsymbol{\Theta} \times \boldsymbol{Z}$ respectively.

We are now ready to present our certificate of robustness against dynamic data poisoning adversaries, which is the largest objective value any dynamic adversary can attain in the stationary state.

Theorem 1. *For any function $\lambda : \Theta \mapsto \mathbb{R}$, for any dynamic adaptive adversary, the average loss (5) is bounded above by*

$$\sup_{\substack{\boldsymbol{\theta} \in \boldsymbol{\Theta} \\ \boldsymbol{z}^{\text{adv}} \in \mathcal{A}}} \mathop{\mathbb{E}}_{\boldsymbol{\theta}' \sim \Pi(\cdot | \boldsymbol{\theta}, \boldsymbol{z}^{\text{adv}})} \left[\lambda \left(\boldsymbol{\theta}' \right) \right] + \ell_{\text{adv}} \left(\boldsymbol{\theta} \right) - \lambda \left(\boldsymbol{\theta} \right). \tag{9}$$

Proof. Follows by weak duality for the LP (8). Detailed proof in Appendix A. □

If strong duality holds Nash and Anderson [1987], Clark [2003], we further have that the optimal value of (8) is exactly equal to

$$\inf_{\lambda : \boldsymbol{\Theta} \mapsto \mathbb{R}} \sup_{\substack{\boldsymbol{\theta} \in \boldsymbol{\Theta} \\ \boldsymbol{z}^{\text{adv}} \in \mathcal{A}}} \mathop{\mathbb{E}}_{\boldsymbol{\theta}' \sim \Pi(\cdot | \boldsymbol{\theta}, \boldsymbol{z}^{\text{adv}})} \left[\lambda \left(\boldsymbol{\theta}' \right) \right] + \ell_{\text{adv}} \left(\boldsymbol{\theta} \right) - \lambda \left(\boldsymbol{\theta} \right). \tag{10}$$

3 MEAN ESTIMATION

Consider the mean estimation problem, where we aim to learn the parameter $\boldsymbol{\theta} \in \mathbb{R}^d$ to estimate the mean $\mu = \mathbb{E}_{\boldsymbol{z} \sim \mathbb{P}^{\text{data}}} [\boldsymbol{z}]$ of a distribution $\mathbb{P}^{\text{data}}$. Given a data point $\boldsymbol{z}_t$, the learning rule is given by:

$$\boldsymbol{\theta}_{t+1} \leftarrow (1 - \eta) \boldsymbol{\theta}_t + \eta \boldsymbol{z}_t + \eta \boldsymbol{B} \boldsymbol{w}_t, \tag{11}$$

where η is the learning rate, $\boldsymbol{S} = \boldsymbol{B}\boldsymbol{B}^\top \in \mathbb{S}_+^d$ is the tunable defense hyperparameter and $\boldsymbol{w}_t \sim \mathcal{N}(0, \boldsymbol{I})$ is Gaussian noise. The adversary wants to maximize its average reward according to the following objective function:

$$\ell_{\text{adv}} \left(\boldsymbol{\theta} \right) = \| \mu - \boldsymbol{\theta} \|^2. \tag{12}$$

Certificate on adversarial loss (analysis).

Theorem 2. *Choosing $\lambda : \mathbb{R}^d \to \mathbb{R}$ in Theorem 1 to be quadratic, i.e. $\lambda \left(\boldsymbol{\theta} \right) = \boldsymbol{\theta}^\top \boldsymbol{A} \boldsymbol{\theta} + \boldsymbol{\theta}^\top \boldsymbol{b}$, the adversarial constraint set of the form $\| \boldsymbol{z}^{\text{adv}} - \mu \|_2^2 \leq r$, the certificate for the mean estimation problem for $\mathbb{P}^{\text{data}}(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{z} | \mu, \boldsymbol{\Sigma})$ for a fixed learning algorithm (i.e. $\boldsymbol{S}$ is fixed) is given by:*

$$\inf_{\boldsymbol{A} \in \mathbb{S}^d, \boldsymbol{b} \in \mathbb{R}^d, \nu \geq 0} g(\boldsymbol{A}, \boldsymbol{b}, \nu, \boldsymbol{S}, \mu, \boldsymbol{\Sigma}), \tag{13}$$

where $g(\boldsymbol{A}, \boldsymbol{b}, \nu, \boldsymbol{S}, \mu, \boldsymbol{\Sigma})$ is a convex objective in $\boldsymbol{A}, \boldsymbol{b}, \nu$ as defined below:

$$\begin{cases} \frac{1}{4} \left\| \begin{bmatrix} 2(1-\epsilon)\eta(1-\eta)\boldsymbol{A}\mu - 2\mu - \eta \boldsymbol{b} \\ \epsilon \eta \boldsymbol{b} + 2\nu\mu \end{bmatrix} \right\|_{\boldsymbol{D}}^2 \\ \quad + (1-\epsilon)(\eta^2 \operatorname{Trace}(\boldsymbol{\Sigma}\boldsymbol{A}) + \eta^2 \mu^\top \boldsymbol{A} \mu + \eta \boldsymbol{b}^\top \mu) \\ \quad\quad + \mu^\top \mu + \eta^2 \operatorname{Trace}(\boldsymbol{A}\boldsymbol{S}) + \nu(r - \mu^\top \mu), \\ \quad\quad\quad \text{if } \nu \geq 0 \text{ and } \boldsymbol{D} \succeq 0 \\ -\infty \qquad \text{otherwise,} \end{cases} \tag{14}$$

where $\boldsymbol{D} = \begin{bmatrix} (1-(1-\eta)^2)\boldsymbol{A} - \boldsymbol{I} & -\eta\epsilon(1-\eta)\boldsymbol{A} \\ -\eta\epsilon(1-\eta)\boldsymbol{A} & -\epsilon\eta^2\boldsymbol{A} - \nu\boldsymbol{I} \end{bmatrix}$ *and* $\|\boldsymbol{x}\|_{\boldsymbol{D}}^2 = \boldsymbol{x}^\top \boldsymbol{D}^{-1}\boldsymbol{x}$.

Proof. The detailed proof can be found in Appendix B. The proof sketch follows by noting that for a fixed λ the dual of the inner constrained maximization problem is a quadratic expression in $\boldsymbol{z}^{\text{adv}}, \boldsymbol{\theta}$ and a finite supremum exists if the Hessian is negative semidefinite (which leads to the $\mathbf{D} \succeq 0$ constraint). Plugging in the maximizer we get a minimization problem in the dual variables ν. Finally we note that the overall minimization problem is jointly convex in $\boldsymbol{A}, \boldsymbol{b}, \nu$. □

Remark 3.1. *The problem above is a convex problem since it has a matrix fractional objective function Boyd and Vandenberghe [2004] with a Linear Matrix Inequality (LMI) constraint.*

Meta-Learning Algorithm. Following the formulation in Eq. (7), we wish to learn a defense parameter $\boldsymbol{S}$ that minimizes the expected loss (expectation over different $\mathbb{P}^{\text{data}}$ from the meta distribution $\mathcal{P}$). For the mean estimation problem this boils down to solving

$$\inf_{\boldsymbol{S}\in\mathbb{S}_+^d} \eta^2 \operatorname{Trace}(\boldsymbol{S}) + \kappa \mathbb{E}_{\mu,\boldsymbol{\Sigma}\sim\mathcal{P}}[\inf_{\substack{\nu\geq 0 \\ \boldsymbol{A}\in\mathbb{S}^d, \boldsymbol{b}\in\mathbb{R}^d}} g(\boldsymbol{A}, \boldsymbol{b}, \nu, \boldsymbol{S}, \mu, \boldsymbol{\Sigma})]. \tag{15}$$

Remark 3.2. *Note that problem (15) is not jointly convex in* $\boldsymbol{S}, \boldsymbol{A}, \boldsymbol{b}, \nu$ *because of the* $\operatorname{Trace}(\boldsymbol{AS})$ *term in* g*; see (14). However, it is convex individually in* $\boldsymbol{S}$ *and* $\{\boldsymbol{A}, \boldsymbol{b}, \nu\}$*. We use an alternating minimization approach to seek a local minimum of problem (15), as detailed in Algorithm 1.*

In practice, one observes a finite number of distributions from $\mathcal{P}$, and sample average approximation is leveraged, with the aim of learning a defense parameter which generalizes well to unseen distributions from $\mathcal{P}$. This process is stated in Algorithm 1.

4 BINARY CLASSIFICATION

We consider the binary classification problem. Given an input feature $\boldsymbol{x} \in \mathbb{R}^d$ such that $\|\boldsymbol{x}\|_2 \leq 1$, a linear predictor $\boldsymbol{\theta} \in \mathbb{R}^d$ tries to predict the label $y \in \{-1, 1\}$ via the sign of $\boldsymbol{\theta}^\top \boldsymbol{x}$. To define the losses, we introduce $\boldsymbol{z} = y\boldsymbol{x} \in \mathbb{R}^d$ which is the label multiplied by the feature and note that $\|\boldsymbol{z}\|_2 \leq 1$.

A dynamic adversary tries to corrupt samples so that the learning algorithm learns a $\boldsymbol{\theta}$ that maximizes the hinge loss on a target distribution $\mathbb{P}^{\text{target}}$, captured by the following objective:

$$\ell_{\text{adv}}(\boldsymbol{\theta}) = \mathop{\mathbb{E}}_{z\sim\mathbb{P}^{\text{target}}} \left[\max\{0, 1 - \boldsymbol{\theta}^\top \boldsymbol{z}\}\right]. \tag{16}$$

Algorithm 1 Meta learning a robust learning algorithm for mean estimation

Input: Set of K distributions $\{\mathcal{N}(\mu_i, \boldsymbol{\Sigma}_i)\}_{i\in[K]}$ sampled from $\mathcal{P}[\mathbb{P}^{\text{data}}]$, tradeoff parameter κ, and max iterations T.
Initialize: $\boldsymbol{S}^{(1)} \in \mathbb{S}_+^d$ randomly.
Alternating Minimization over Lagrange multipliers $\{\boldsymbol{A}_i, \boldsymbol{b}_i, \nu_i\}_{i\in[K]}$ *and defense parameter* $\boldsymbol{S}$.
for $t = 1, \ldots, T$ **do**
  for $i = 1, \ldots, K$ **do**
    $\boldsymbol{A}_i, \boldsymbol{b}_i, \nu_i = \operatorname{argmin}_{\boldsymbol{A}\in\mathbb{S}^d, \boldsymbol{b}\in\mathbb{R}^d, \nu\geq 0} g(\boldsymbol{A}, \boldsymbol{b}, \nu, \boldsymbol{S}^{(t)}, \mu_i, \boldsymbol{\Sigma}_i)$
  end for
  $\boldsymbol{S}^{(t+1)} = \operatorname{argmin}_{\boldsymbol{S}\in\mathbb{S}_+^d} \big(\frac{\kappa}{K} \sum_{i\in[K]} g(\boldsymbol{A}_i, \boldsymbol{b}_i, \nu_i, \boldsymbol{S}, \mu_i, \boldsymbol{\Sigma}_i) + \eta^2 \operatorname{Trace}(\boldsymbol{S})\big)$
end for
return $\boldsymbol{S}^{(T+1)}$

The learning algorithm tires to minimize the regularized hinge loss on the observed datapoints:

$$l(\boldsymbol{\theta}, \boldsymbol{z}) = \max\{0, 1 - \boldsymbol{\theta}^\top \boldsymbol{z}\} + \frac{\sigma}{2}\|\boldsymbol{\theta}\|_2^2. \tag{17}$$

Upon observing a sample $\boldsymbol{z}_t$, the parameter is updated via a gradient descent: $\boldsymbol{\theta}_{t+1} = F(\boldsymbol{\theta}_t, \boldsymbol{z}_t)$, where

$$\begin{aligned} F(\boldsymbol{\theta}, \boldsymbol{z}) &= \boldsymbol{\theta}_t - \eta \nabla_{\boldsymbol{\theta}} l(\boldsymbol{\theta}_t, \boldsymbol{z}_t) \\ &= (1 - \sigma\eta)\boldsymbol{\theta} + \eta \mathbb{I}[\boldsymbol{\theta}^\top \boldsymbol{z} \leq 1]\boldsymbol{z}. \end{aligned} \tag{18}$$

Below, we provide a certificate for the adversarial objective at stationarity of this learning algorithm.

Theorem 3. *Choosing* $\lambda : \mathbb{R}^d \to \mathbb{R}$ *in Theorem 1 to be quadratic, i.e.,* $\lambda(\boldsymbol{\theta}) = \boldsymbol{\theta}^\top \boldsymbol{A}\boldsymbol{\theta} + \boldsymbol{b}^\top \boldsymbol{\theta}$*, parameter space* $\boldsymbol{\Theta} = \{\boldsymbol{\theta} \mid \|\boldsymbol{\theta}\|_2 \leq \frac{1}{\sigma}\}$ *and the adversarial constraint set of the form* $\mathcal{A} = \{\boldsymbol{z}^{\text{adv}} \mid \|\boldsymbol{z}^{\text{adv}}\|_2 \leq 1\}$*, the certificate for the binary classification problem for* $\mathbb{P}^{\text{data}}(\boldsymbol{z}) = \{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_N\}$ *for a learning algorithm with regularization parameter* σ*, and learning rate* η *is upper bounded by*

$$\max\left(\mathsf{OPT}_1, \mathsf{OPT}_2\right)$$

$$\begin{aligned} \mathsf{OPT}_1 = \inf_{\nu, \boldsymbol{A}, \boldsymbol{b}} \quad & \|\boldsymbol{p}(\boldsymbol{b}, \nu)\|^2_{\boldsymbol{D}(\boldsymbol{A},\nu)^{-1}} + q(\nu) \\ \text{s.t.} \quad & \boldsymbol{D}(\boldsymbol{A}, \nu) \succeq 0, \\ & \boldsymbol{r}_i(\nu) + \boldsymbol{s}(z_i, \boldsymbol{A}, \boldsymbol{b}) = 0, \quad \forall i \in [N]. \end{aligned}$$

$$\begin{aligned} \mathsf{OPT}_2 = \inf_{\nu, \boldsymbol{A}, \boldsymbol{b}} \quad & \|\boldsymbol{p}'(\boldsymbol{b}, \nu)\|^2_{\boldsymbol{D}'(\boldsymbol{A},\nu)^{-1}} + q'(\nu) \\ \text{s.t.} \quad & \boldsymbol{D}'(\boldsymbol{A}, \nu) \succeq 0, \\ & \boldsymbol{r}_i(\nu) + \boldsymbol{s}(z_i, \boldsymbol{A}, \boldsymbol{b}) = 0, \quad \forall i \in [N]. \end{aligned}$$

where $\boldsymbol{p}()$*,* $\boldsymbol{p}'()$*,* $q()$*,* $q'()$*,* $\boldsymbol{D}()$*,* $\boldsymbol{D}'()$*,* $\boldsymbol{r}_1(), \ldots, \boldsymbol{r}_N()$*,* $\boldsymbol{s}()$ *are affine functions of the optimization variables*

$\nu, \boldsymbol{A}, \boldsymbol{b}$ *as defined below and* $\nu = \{\nu_1, \ldots, \nu_{10}\}$*:*

$$\boldsymbol{p}(\boldsymbol{b}, \nu) = \frac{1}{2}\begin{bmatrix} -\sigma\eta\boldsymbol{b} + \sum_{i\in[N]}\left((\nu_{1i} - \nu_{2i})\boldsymbol{z}_i - \nu_{4i} + \nu_{6i}\right) \\ \epsilon\eta\boldsymbol{b} \end{bmatrix},$$

$$\boldsymbol{D}(\boldsymbol{A}, \nu) = \begin{bmatrix} [1-(1-\sigma\eta)^2]\boldsymbol{A} + \nu_8\boldsymbol{I} & -\epsilon(1-\sigma\eta)\eta\boldsymbol{A} + \nu_9\boldsymbol{I} \\ -\epsilon(1-\sigma\eta)\eta\boldsymbol{A} + \nu_9\boldsymbol{I} & -\epsilon\eta^2\boldsymbol{A} + \nu_{10}\boldsymbol{I} \end{bmatrix},$$

$$q(\nu) = q'(\nu) + 2\nu_9 + \nu_{10},$$

$$\boldsymbol{p}'(\boldsymbol{b}, \nu) = \frac{1}{2}\left[-\sigma\eta\boldsymbol{b} + \sum_{i\in[N]}\left((\nu_{1i} - \nu_{2i})\boldsymbol{z}_i - \nu_{4i} + \nu_{6i}\right)\right],$$

$$\boldsymbol{D}'(\boldsymbol{A}, \nu) = [1-(1-\sigma\eta)^2]\boldsymbol{A} + \nu_8\boldsymbol{I},$$

$$q'(\nu) = -\nu_1^\top\mathbf{1} + (2 + \tfrac{1}{\sigma})\nu_2^\top\mathbf{1} + \tfrac{(\nu_4+\nu_6)^\top\mathbf{1}}{\sigma} + \mathbf{1}^\top\nu_7 + \frac{\nu_8}{\sigma^2},$$

$$\boldsymbol{r}_i(\nu) = \begin{bmatrix} (1+\frac{1}{\sigma})(\nu_{1i} - \nu_{2i}) - \frac{\mathbf{1}^\top(\nu_{4i}+\nu_{6i})}{\sigma} - \nu_{7i} \\ \nu_{3i} + \nu_{4i} - \nu_{5i} - \nu_{6i} \end{bmatrix},$$

$$\boldsymbol{s}(\boldsymbol{z}_i, \boldsymbol{A}, \boldsymbol{b}) = \frac{1}{N}\begin{bmatrix} (1-\epsilon)\eta^2\boldsymbol{z}_i^\top\boldsymbol{A}\boldsymbol{z}_i + (1-\epsilon)\eta\boldsymbol{b}^\top\boldsymbol{z}_i + 1 \\ 2(1-\epsilon)\eta(1-\sigma\eta)\boldsymbol{A}\boldsymbol{z}_i - \boldsymbol{z}_i \end{bmatrix}.$$

Proof. A detailed derivation can be found in the Appendix C. The proof steps are: (i) Regularization implicitly bounds the decision variable $\boldsymbol{\theta} \in \boldsymbol{\Theta}$ without the need for projections, (ii) Considering 2 cases for the indicator term of $\boldsymbol{z}^{\text{adv}}$ and taking the maximum of these 2 cases, (iii) Relaxing the indicator terms for benign samples into continuous variables between $[0, 1]$, and (iv) Using McCormick relaxations [Mitsos et al., 2009] to convexify the bilinear terms in the objective. □

Extension to Multi-Class Setting: Our analysis can be extended to the multi-class classification setting. Let us consider score based classifiers, where $\Theta = \{\theta_1, \ldots, \theta_K\} \in \mathbb{R}^d$ are the learnable parameters and the class prediction for a feature $x \in \mathbb{R}^d$ is given by $\arg\max_{i\in[K]} \theta_i^\top x$.

The SVM loss for any arbitrary feature x with label y is defined as: $\ell(\Theta, (x, y)) = \sum_{j\neq y} \max\{0, 1+(\theta_j - \theta_y)^\top x\}$

The gradient update takes the form: $F(\theta_y, (x, y)) = \theta_y + \eta\sum_{j\neq y}\mathbb{I}[1 + (\theta_j - \theta_y)^\top x > 0]x$ and for all $j \neq y$, we have $F(\theta_j, (x, y)) = \theta_j - \eta\mathbb{I}[1 + (\theta_j - \theta_y)^\top x > 0]x$.

Note that the non-linearity in both the loss function and the update is composed with a linear combination of the parameters (i.e. $\theta_j - \theta_y$), and thus the analysis in the proof of Theorem 3 still holds, and our analysis for the binary classification generalizes to the multi-class classification.

5 EXPERIMENTS

We conduct experiments on both synthetic and real data to empirically validate our theoretical tools.

5.1 Mean Estimation

We wish to evaluate the robustness of our meta learning algorithm in Eq. (7) to design a defense against a dynamic best responding adversary on a ($d = 20$) mean estimation task. We consider 3 different learning algorithms: 1. **No Defense:** Eq. (11) with $\boldsymbol{B} = \mathbf{0}$, i.e. no additive Gaussian noise, 2. **Baseline Defense:** $\boldsymbol{B}$ in Eq. (11) is restricted to be Isotropic Gaussian, 3. **Defense:** $\boldsymbol{B}$ in Eq. (11) can be arbitrarily shaped. We use Algorithm 1 to compute the defense parameter $\boldsymbol{S} = \boldsymbol{B}\boldsymbol{B}^\top$ for the latter 2 learning algorithms by training on 10 randomly chosen Gaussians drawn from standard Gaussian prior for the mean and standard Inverse Wishart prior for the covariance. We report the average test performance on 50 Gaussian distributions drawn from the same prior (see Figure 2).

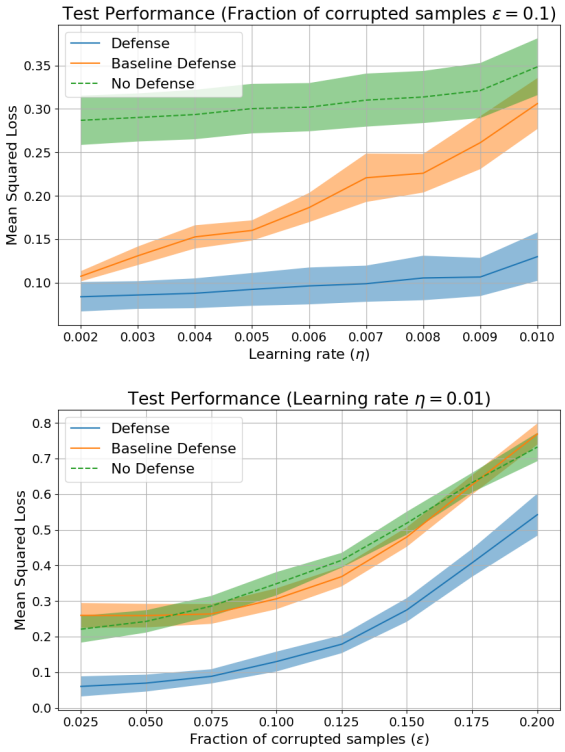

Figure 2: Test performance (mean squared error between true and estimated means) upon varying the learning rates (above) and the the fraction of samples corrupted by the dynamic adversary (below) and observed that our defense significantly outperforms training without defense.

5.2 Image Classification

We consider binary classification tasks on multiple datasets : (i) MNIST [Deng, 2012], (ii) FashionMNIST [Xiao et al., 2017], (iii) CIFAR-10 [Krizhevsky et al., 2009]. Detailed dataset descriptions and preprocessing steps can be found in Appendix D.

In our experiments, we choose $\mathbb{P}^{\text{target}}$ in Eq. (16) to be the same as $\mathbb{P}^{\text{data}}$, i.e., the adversary's objective is to make the model perform poorly on the benign data distribution.

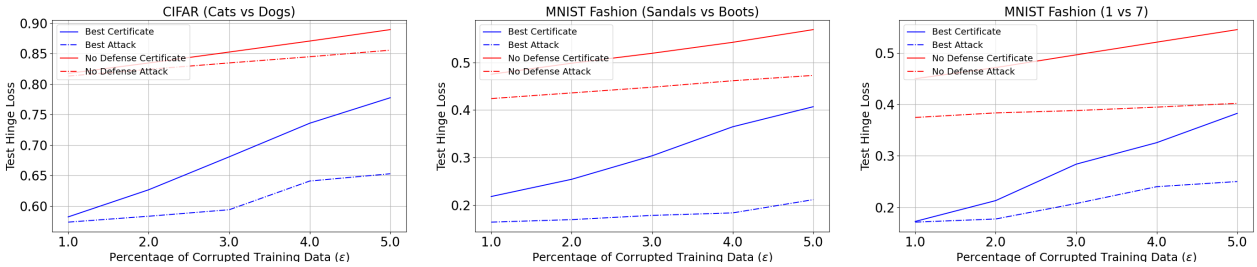

Figure 3: We plot the robustness certificates for different hyperparameter settings: one obtained by optimizing the meta-learning loss to balance benign performance and robustness (Best Certificate) and the other arbitrarily chosen (No Defense). The certificates provide upper bounds on the objective of the optimal dynamic adversary. Additionally, we plot the highest test losses under various adversarial attacks, serving as lower bounds on the optimal adversary's objective.

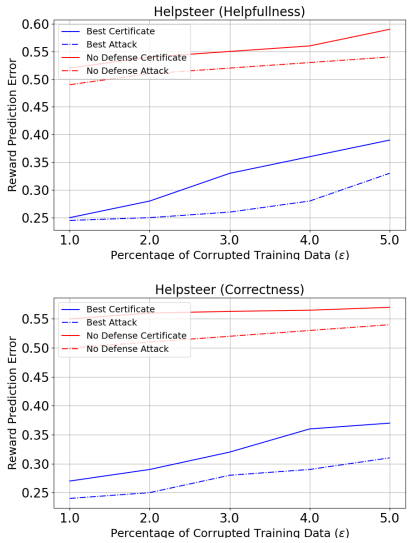

Figure 4: We plot the robustness certificates for different hyperparameter settings: one obtained by optimizing the meta-learning loss to balance benign performance and robustness (Best Certificate) and the other arbitrarily chosen (No Defense). The certificates provide upper bounds on the objective of the optimal dynamic adversary. Additionally, we plot the highest test losses under various adversarial attacks, serving as lower bounds on the optimal adversary's objective.

To learn the hyperparameters for a robust online learning algorithm (Eq. (18)), we adopt the Meta learning setup presented in Eq. (7). We choose $\mathcal{P}$ as a set of binary classification datasets on label pairs different from the label pair the online algorithm receives data from. For example, we use data from MNIST: 4 vs 9, 5 vs 8, 3 vs 8, 0 vs 6 to construct the objective in Eq. (7) and then test the performance of the onlne learning algorithm with the learnt hyperparameters on MNIST: 1 vs 7 (see Appendix D for more details). We compute certificates for different fractions of the training data to be corrupted $\epsilon \in \{1\%, 2\%, 3\%, 4\%, 5\%\}$, and vary η, σ for various values of η and σ by performing a grid search over $\eta \in \{5\times10^{-5}, 1\times10^{-4}, 5\times10^{-4}, 1\times10^{-3}, 5\times10^{-3}\}$ and $\sigma \in \{3\times10^{-3}, 6\times10^{-3}, 1\times10^{-2}, 3\times10^{-2}, 6\times10^{-2}\}$. We tune the hyperparameter κ in Eq. (7) to trade off benign loss and robustness certificate by the generalization test loss on a held-out validation set. In Figure 3 we plot the certificates of robustness corresponding to hyperparameters (η, σ) pairs having smallest objective values in Eq. (7).

Additionally, we evaluate the learning algorithm against three types of attacks: 1. **fgsm:** The attacker, having access to model parameters at each time step, initializes a random $\mathbf{z}_{\text{adv}}$, computes the gradient of $\boldsymbol{z}^{\text{adv}}$ on the adversarial loss, takes a gradient ascent step on $\boldsymbol{z}^{\text{adv}}$ and projects the result onto the unit ℓ_2 ball, 2. **pgd:** similar to fgsm, but instead of a single big gradient step, the attacker takes multiple small steps and projects onto the unit ℓ_2 ball, 3. **label flip:** the attacker picks an arbitrary data point and flips its label. In Figure 3, we plot the best attack (corresponding to largest prediction error) for each value of ϵ. We observe the following: The test adversarial loss under different heuristic attacks is consistently lower than the computed robustness certificate.

5.3 Reward Learning

We conduct reward learning experiments using Helpsteer, an open-source helpfulness dataset that is used to align large language models to become more helpful, factually correct and coherent, while being adjustable in terms of the complexity and verbosity of its responses [Wang et al., 2023, Dong et al., 2023]. Since the datasets of human feedback, both for open source and closed sourced models, are typically composed of users 'in the wild' using the model, there is potential for adversaries to introduce poisoning. This can lead to the learned reward model favoring specific demographic groups, political entities or unscientific points of view, eventually leading to bad and potentially harmful experiences for users of

language models fine tuned on the learned reward.

To apply our framework to this problem, we consider a simple reward model. Given a (prompt, response) pair, we extract a feature representation $\boldsymbol{x}$ using a pretrained BERT model, and our reward model parameterized by $\boldsymbol{\theta}$, predicts the reward as $\boldsymbol{\theta}^\top \boldsymbol{x}$. Given the score y (normalised to fall within the range $[-1, 1]$) on a particular attribute (say helpfulness), the learning algorithm proceeds to learn the reward model by performing gradient descent on the regularized hinge loss objective Eq. (17). We follow the similar hyperprameter search space as the image classification example using the meta learning setup in Eq. (7). The results for helpfulness and correctness are presented in Figure 4. More details are deferred to Appendix D.

6 CONCLUSION AND FUTURE WORK

This paper presents a novel framework for computing certified bounds on the worst-case impacts of dynamic data poisoning adversaries in machine learning. This framework lays the groundwork for developing robust algorithms, which we demonstrate for mean estimation and linear classifiers. Extending this framework to efficient algorithms for more general learning setups, particularly deep learning setups that use human feedback with potential for malicious feedback, would be a compelling direction of future work. Furthermore, considering alternative strategies for computing the bound in (9), particularly ones driven by AI advances, would be a promising approach for scaling. Recent work has demonstrated that AI systems can be used to discover Lyapunov functions for dynamical systems [Alfarano et al., 2023], indicating that AI driven approaches could hold promise in this setting.

References

Alberto Alfarano, François Charton, Amaury Hayat, and CERMICS-Ecole des Ponts Paristech. Discovering lyapunov functions with transformers. In *The 3rd Workshop on Mathematical Reasoning and AI at NeurIPS*, volume 23, 2023.

Marcin Andrychowicz, Misha Denil, Sergio Gomez, Matthew W Hoffman, David Pfau, Tom Schaul, Brendan Shillingford, and Nando De Freitas. Learning to learn by gradient descent by gradient descent. *Advances in neural information processing systems*, 29, 2016.

Battista Biggio and Fabio Roli. Wild patterns: Ten years after the rise of adversarial machine learning. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pages 2154–2156, 2018.

Battista Biggio, Blaine Nelson, and Pavel Laskov. Poisoning attacks against support vector machines. *arXiv preprint arXiv:1206.6389*, 2012.

Avinandan Bose, Mihaela Curmei, Daniel Jiang, Jamie H Morgenstern, Sarah Dean, Lillian Ratliff, and Maryam Fazel. Initializing services in interactive ml systems for diverse users. *Advances in Neural Information Processing Systems*, 37:57701–57732, 2024a.

Avinandan Bose, Simon Shaolei Du, and Maryam Fazel. Offline multi-task transfer rl with representational penalization. *arXiv preprint arXiv:2402.12570*, 2024b.

Avinandan Bose, Zhihan Xiong, Aadirupa Saha, Simon Shaolei Du, and Maryam Fazel. Hybrid preference optimization for alignment: Provably faster convergence rates by combining offline preferences with online exploration. *arXiv preprint arXiv:2412.10616*, 2024c.

Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu, and Dawn Song. Targeted backdoor attacks on deep learning systems using data poisoning. *arXiv preprint arXiv:1712.05526*, 2017.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30, 2017.

Stephen A Clark. An infinite-dimensional lp duality theorem. *Mathematics of Operations Research*, 28 (2):233–245, 2003.

Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

Ilias Diakonikolas and Daniel M Kane. *Algorithmic high-dimensional robust statistics*. Cambridge university press, 2023.

Steven Diamond and Stephen Boyd. CVXPY: A Python-embedded modeling language for convex optimization. *Journal of Machine Learning Research*, 17(83):1–5, 2016.

Yi Dong, Zhilin Wang, Makesh Narsimhan Sreedhar, Xianchao Wu, and Oleksii Kuchaiev. Steerlm: Attribute conditioned sft as an (user-steerable) alternative to rlhf, 2023.

Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Identifying vulnerabilities in the machine learning model supply chain. *arXiv preprint arXiv:1708.06733*, 2017.

Xingshuo Han, Guowen Xu, Yuan Zhou, Xuehuan Yang, Jiwei Li, and Tianwei Zhang. Physical backdoor attacks to lane detection systems in autonomous driving. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 2957–2968, 2022.

Sepp Hochreiter, A Steven Younger, and Peter R Conwell. Learning to learn using gradient descent. In *Artificial Neural Networks—ICANN 2001: International Conference Vienna, Austria, August 21–25, 2001 Proceedings 11*, pages 87–94. Springer, 2001.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Alan Malek, Yasin Abbasi-Yadkori, and Peter Bartlett. Linear programming for large-scale markov decision problems. In *International conference on machine learning*, pages 496–504. PMLR, 2014.

Alexander Mitsos, Benoit Chachuat, and Paul I Barton. Mccormick-based relaxations of algorithms. *SIAM Journal on Optimization*, 20(2):573–601, 2009.

Peter Nash and Edward J Anderson. Linear programming in infinite-dimensional spaces: theory and applications. *(No Title)*, 1987.

James Newsome, Brad Karp, and Dawn Song. Paragraph: Thwarting signature learning by training maliciously. In *Recent Advances in Intrusion Detection: 9th International Symposium, RAID 2006 Hamburg, Germany, September 20-22, 2006 Proceedings 9*, pages 81–105. Springer, 2006.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

Elan Rosenfeld, Ezra Winston, Pradeep Ravikumar, and Zico Kolter. Certified robustness to label-flipping attacks via randomized smoothing. In *International Conference on Machine Learning*, pages 8230–8241. PMLR, 2020.

Philip Sosnin, Mark N Müller, Maximilian Baader, Calvin Tsay, and Matthew Wicker. Certified robustness to data poisoning in gradient-based training. *arXiv preprint arXiv:2406.05670*, 2024.

Jacob Steinhardt, Pang Wei W Koh, and Percy S Liang. Certified defenses for data poisoning attacks. *Advances in neural information processing systems*, 30, 2017.

Zhiyi Tian, Lei Cui, Jie Liang, and Shui Yu. A comprehensive survey on poisoning attacks and countermeasures in machine learning. *ACM Computing Surveys*, 55(8):1–35, 2022.

Florian Tramèr, Fan Zhang, Ari Juels, Michael K Reiter, and Thomas Ristenpart. Stealing machine learning models via prediction {APIs}. In *25th USENIX security symposium (USENIX Security 16)*, pages 601–618, 2016.

Wenxiao Wang and Soheil Feizi. Temporal robustness against data poisoning. *Advances in Neural Information Processing Systems*, 36, 2024.

Yizhen Wang and Kamalika Chaudhuri. Data poisoning attacks against online learning. *arXiv preprint arXiv:1808.08994*, 2018.

Zhilin Wang, Yi Dong, Jiaqi Zeng, Virginia Adams, Makesh Narsimhan Sreedhar, Daniel Egert, Olivier Delalleau, Jane Polak Scowcroft, Neel Kant, Aidan Swope, and Oleksii Kuchaiev. Helpsteer: Multi-attribute helpfulness dataset for steerlm, 2023.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

Chulin Xie, Yunhui Long, Pin-Yu Chen, and Bo Li. Uncovering the connection between differential privacy and certified robustness of federated learning against poisoning attacks. *arXiv preprint arXiv:2209.04030*, 2022.

Xuezhou Zhang, Xiaojin Zhu, and Laurent Lessard. Online data poisoning attacks. In *Learning for Dynamics and Control*, pages 201–210. PMLR, 2020.

Yuhao Zhang, Aws Albarghouthi, and Loris D'Antoni. Bagflip: A certified defense against data poisoning. *Advances in Neural Information Processing Systems*, 35:31474–31483, 2022.

Chen Zhu, W Ronny Huang, Hengduo Li, Gavin Taylor, Christoph Studer, and Tom Goldstein. Transferable clean-label poisoning attacks on deep neural nets. In *International conference on machine learning*, pages 7614–7623. PMLR, 2019.

A Proofs

Theorem 1. *For any function $\lambda : \Theta \mapsto \mathbb{R}$, for any dynamic adaptive adversary, the average loss (5) is bounded above by*

$$\sup_{\substack{\boldsymbol{\theta}\in\boldsymbol{\Theta}\\ \boldsymbol{z}^{\text{adv}}\in\mathcal{A}}} \mathop{\mathbb{E}}_{\boldsymbol{\theta}'\sim\Pi(\cdot|\boldsymbol{\theta},\boldsymbol{z}^{\text{adv}})} \left[\lambda\left(\boldsymbol{\theta}'\right)\right] + \ell_{\text{adv}}\left(\boldsymbol{\theta}\right) - \lambda\left(\boldsymbol{\theta}\right). \tag{9}$$

Proof. The primal problem is defined below:

$$\sup_{\substack{d_\pi\in\mathcal{P}[\boldsymbol{\Theta}]\\ \pi\in\mathcal{P}[\boldsymbol{\Theta}\times\boldsymbol{Z}]}} \mathop{\mathbb{E}}_{\boldsymbol{\theta}\sim d_\pi(\cdot)} \left[\ell_{\text{adv}}\left(\boldsymbol{\theta}\right)\right], \quad \text{subject to}$$
$$\mathop{\mathbb{E}}_{\substack{\boldsymbol{\theta}\sim d_\pi(\cdot)\\ \boldsymbol{z}^{\text{adv}}\sim\pi(\cdot|\boldsymbol{\theta})}} \left[\Pi(\boldsymbol{\theta}'|\boldsymbol{\theta},\boldsymbol{z}^{\text{adv}})\right] = d_\pi(\boldsymbol{\theta}') \quad \forall\boldsymbol{\theta}'\in\Theta.$$

For a given function $\lambda : \boldsymbol{\Theta} \mapsto \mathbb{R}$, the Lagrangian can be written as:

$$\begin{aligned}
\mathcal{L}(d_\pi,\pi,\lambda) &= \mathop{\mathbb{E}}_{\boldsymbol{\theta}\sim d_\pi(\cdot)} \left[\ell_{\text{adv}}\left(\boldsymbol{\theta}\right)\right] + \int_{\boldsymbol{\theta}'\in\boldsymbol{\Theta}} \lambda(\boldsymbol{\theta}') \left[\mathop{\mathbb{E}}_{\substack{\boldsymbol{\theta}\sim d_\pi(\cdot)\\ \boldsymbol{z}^{\text{adv}}\sim\pi(\cdot|\boldsymbol{\theta})}} \left[\Pi(\boldsymbol{\theta}'|\boldsymbol{\theta},\boldsymbol{z}^{\text{adv}})\right] - d_\pi(\boldsymbol{\theta}')\right] d\boldsymbol{\theta}' \\
&= \mathop{\mathbb{E}}_{\boldsymbol{\theta}\sim d_\pi(\cdot)} \left[\ell_{\text{adv}}\left(\boldsymbol{\theta}\right)\right] + \mathop{\mathbb{E}}_{\substack{\boldsymbol{\theta}\sim d_\pi(\cdot)\\ \boldsymbol{z}^{\text{adv}}\sim\pi(\cdot|\boldsymbol{\theta})}} \left[\mathop{\mathbb{E}}_{\boldsymbol{\theta}'\sim\Pi(\cdot|\boldsymbol{\theta},\boldsymbol{z}^{\text{adv}})} \left[\lambda(\boldsymbol{\theta}')\right]\right] - \mathop{\mathbb{E}}_{\boldsymbol{\theta}'\sim d_\pi(\cdot)} \left[\lambda(\boldsymbol{\theta}')\right] \\
&= \mathop{\mathbb{E}}_{\substack{\boldsymbol{\theta}\sim d_\pi(\cdot)\\ \boldsymbol{z}^{\text{adv}}\sim\pi(\cdot|\boldsymbol{\theta})}} \left[\mathop{\mathbb{E}}_{\boldsymbol{\theta}'\sim\Pi(\cdot|\boldsymbol{\theta},\boldsymbol{z}^{\text{adv}})} \left[\lambda(\boldsymbol{\theta}')\right] + \ell_{\text{adv}}\left(\boldsymbol{\theta}\right) - \lambda(\boldsymbol{\theta})\right],
\end{aligned}$$

which serves as an upper bound on the primal objective. Note that the value of the Lagrangian for a given λ depends on the expectation of $\mathbb{E}_{\boldsymbol{\theta}'\sim\Pi(\cdot|\boldsymbol{\theta},\boldsymbol{z}^{\text{adv}})}\left[\lambda(\boldsymbol{\theta}')\right] + \ell_{\text{adv}}\left(\boldsymbol{\theta}\right) - \lambda(\boldsymbol{\theta})$ over the joint distribution of $\boldsymbol{\theta}\times\boldsymbol{z}^{\text{adv}}$. Since $\boldsymbol{\Theta}$ and $\mathcal{A}$ are compact, the supremum for a given λ occurs for the distribution placing all its mass on the point $\boldsymbol{\theta}, \boldsymbol{z}^{\text{adv}}$ maximizing: $\mathbb{E}_{\boldsymbol{\theta}'\sim\Pi(\cdot|\boldsymbol{\theta},\boldsymbol{z}^{\text{adv}})}\left[\lambda(\boldsymbol{\theta}')\right] + \ell_{\text{adv}}\left(\boldsymbol{\theta}\right) - \lambda(\boldsymbol{\theta})$.

Therefore, for any choice of λ and any feasible d_π, we have:

$$\sup_{\substack{d_\pi\in\mathcal{P}[\boldsymbol{\Theta}]\\ \pi\in\mathcal{P}[\boldsymbol{\Theta}\times\boldsymbol{Z}]}} \mathcal{L}(d_\pi,\pi,\lambda) = \sup_{\substack{\boldsymbol{\theta}\in\boldsymbol{\Theta}\\ \boldsymbol{z}^{\text{adv}}\in\mathcal{A}}} \mathop{\mathbb{E}}_{\boldsymbol{\theta}'\sim\Pi(\cdot|\boldsymbol{\theta},\boldsymbol{z}^{\text{adv}})} \left[\lambda\left(\boldsymbol{\theta}'\right)\right] + \ell_{\text{adv}}\left(\boldsymbol{\theta}\right) - \lambda\left(\boldsymbol{\theta}\right).$$

This completes the proof. □

B Mean Estimation

Theorem 2. *Choosing $\lambda : \mathbb{R}^d \to \mathbb{R}$ in Theorem 1 to be quadratic, i.e. $\lambda\left(\boldsymbol{\theta}\right) = \boldsymbol{\theta}^\top\boldsymbol{A}\boldsymbol{\theta} + \boldsymbol{\theta}^\top\boldsymbol{b}$, the adversarial constraint set of the form $\|\boldsymbol{z}^{\text{adv}} - \mu\|_2^2 \le r$, the certificate for the mean estimation problem for $\mathbb{P}^{\text{data}}(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{z}|\mu,\boldsymbol{\Sigma})$ for a fixed learning algorithm (i.e. $\boldsymbol{S}$ is fixed) is given by:*

$$\inf_{\boldsymbol{A}\in\mathbb{S}^d,\boldsymbol{b}\in\mathbb{R}^d,\nu\ge 0} g(\boldsymbol{A},\boldsymbol{b},\nu,\boldsymbol{S},\mu,\boldsymbol{\Sigma}), \tag{13}$$

where $g(\boldsymbol{A},\boldsymbol{b},\nu,\boldsymbol{S},\mu,\boldsymbol{\Sigma})$ is a convex objective in $\boldsymbol{A},\boldsymbol{b},\nu$ as defined below:

$$\begin{cases}
\frac{1}{4}\left\|\begin{bmatrix}2(1-\epsilon)\eta(1-\eta)\boldsymbol{A}\mu - 2\mu - \eta\boldsymbol{b}\\ \epsilon\eta\boldsymbol{b} + 2\nu\mu\end{bmatrix}\right\|_{\boldsymbol{D}}^2 \\
\quad +(1-\epsilon)(\eta^2\,\text{Trace}(\boldsymbol{\Sigma}\boldsymbol{A}) + \eta^2\mu^\top\boldsymbol{A}\mu + \eta\boldsymbol{b}^\top\mu) \\
\qquad +\mu^\top\mu + \eta^2\,\text{Trace}(\boldsymbol{A}\boldsymbol{S}) + \nu(r - \mu^\top\mu), \\
\qquad\quad \textit{if } \nu\ge 0 \textit{ and } \boldsymbol{D}\succeq 0 \\
-\infty \qquad \textit{otherwise,}
\end{cases} \tag{14}$$

where $\boldsymbol{D} = \begin{bmatrix}(1-(1-\eta)^2)\boldsymbol{A} - \boldsymbol{I} & -\eta\epsilon(1-\eta)\boldsymbol{A}\\ -\eta\epsilon(1-\eta)\boldsymbol{A} & -\epsilon\eta^2\boldsymbol{A} - \nu\boldsymbol{I}\end{bmatrix}$ *and* $\|\boldsymbol{x}\|_{\boldsymbol{D}}^2 = \boldsymbol{x}^\top\boldsymbol{D}^{-1}\boldsymbol{x}$.

Proof. We can write the learning algorithm in Eq. (1) for the case of mean estimation as follows:

$$\boldsymbol{\theta}_{t+1} = F\left(\boldsymbol{\theta}_t, \boldsymbol{z}_t\right) + \eta \boldsymbol{B} \boldsymbol{w}_t,$$

where $F\left(\boldsymbol{\theta}, \boldsymbol{z}\right) = \boldsymbol{\theta}\left(1-\eta\right) + \eta \boldsymbol{z}$, which is a linear transformation of $\boldsymbol{\theta}$ followed by additive Gaussian noise.

The transition distribution for the parameter is given by:

$$\mathbb{P}_{\boldsymbol{S}, \mathbb{P}^{\text{data}}, \boldsymbol{z}^{\text{adv}}}\left(\boldsymbol{\theta}' | \boldsymbol{\theta}\right) = \epsilon \mathcal{N}\left(\boldsymbol{\theta}' | F(\boldsymbol{\theta}, \boldsymbol{z}^{\text{adv}}), \eta^2 \boldsymbol{S}\right) + (1-\epsilon) \mathop{\mathbb{E}}_{\boldsymbol{z} \sim \mathbb{P}^{\text{data}}}\left[\mathcal{N}\left(\boldsymbol{\theta}' | F(\boldsymbol{\theta}, \boldsymbol{z}), \eta^2 \boldsymbol{S}\right)\right] \tag{19}$$

which is a Gaussian distribution whose mean depends linearly on $\boldsymbol{\theta}$ and $\boldsymbol{z}^{\text{adv}}$.

Then, we have from Eq. (10) that the certified bound on the adversarial objective is given by:

$$\sup_{\substack{\boldsymbol{z}^{\text{adv}} \in \mathcal{A} \\ \boldsymbol{\theta}}} \epsilon \mathop{\mathbb{E}}_{\boldsymbol{\theta}' \sim \mathcal{N}(F(\boldsymbol{\theta}, \boldsymbol{z}^{\text{adv}}), \eta^2 \boldsymbol{S})}\left[\lambda\left(\boldsymbol{\theta}'\right)\right] + (1-\epsilon) \mathop{\mathbb{E}}_{\boldsymbol{z} \sim \mathbb{P}^{\text{data}}}\left[\mathop{\mathbb{E}}_{\boldsymbol{\theta}' \sim \mathcal{N}(F(\boldsymbol{\theta}, \boldsymbol{z}), \eta^2 \boldsymbol{S})}\left[\lambda\left(\boldsymbol{\theta}'\right)\right]\right] - \lambda\left(\boldsymbol{\theta}\right) + \ell_{\text{adv}}\left(\boldsymbol{\theta}\right) \tag{20a}$$

We choose $\lambda\left(\boldsymbol{\theta}\right) = \boldsymbol{\theta}^\top \boldsymbol{A} \boldsymbol{\theta} + \boldsymbol{\theta}^\top \boldsymbol{b}$ to be a quadratic function. Then we have:

$$= \sup_{\substack{\boldsymbol{z}^{\text{adv}} \in \mathcal{A} \\ \boldsymbol{\theta}}} \epsilon\left(\lambda\left(F\left(\boldsymbol{\theta}, \boldsymbol{z}^{\text{adv}}\right)\right) + \eta^2 \left\langle \nabla^2 \lambda\left(0\right), \boldsymbol{S}\right\rangle\right) + (1-\epsilon) \mathop{\mathbb{E}}_{\boldsymbol{z} \sim \mathbb{P}^{\text{data}}}\left[\lambda\left(F\left(\boldsymbol{\theta}, \boldsymbol{z}\right)\right) + \eta^2 \left\langle \nabla^2 \lambda\left(0\right), \boldsymbol{S}\right\rangle\right] - \lambda\left(\boldsymbol{\theta}\right) + \ell_{\text{adv}}\left(\boldsymbol{\theta}\right) \tag{20b}$$

$$= \sup_{\substack{\boldsymbol{z}^{\text{adv}} : \|\boldsymbol{z}^{\text{adv}} - \mu\|_2^2 \leq r \\ \boldsymbol{\theta}}} - \left\| \begin{bmatrix} \boldsymbol{\theta} \\ \boldsymbol{z}^{\text{adv}} \end{bmatrix} \right\|_{\boldsymbol{E}^{-1}}^2 + \begin{bmatrix} \boldsymbol{\theta} \\ \boldsymbol{z}^{\text{adv}} \end{bmatrix}^\top \begin{bmatrix} 2(1-\epsilon)\eta(1-\eta)\boldsymbol{A}\mu - 2\mu - \eta \boldsymbol{b} \\ \epsilon \eta \boldsymbol{b} \end{bmatrix}$$

$$+ (1-\epsilon)(\eta^2 \operatorname{Trace}(\boldsymbol{\Sigma}\boldsymbol{A}) + \eta^2 \mu^\top \boldsymbol{A} \mu + \eta \boldsymbol{b}^\top \mu) + \mu^\top \mu, \tag{20c}$$

where $\boldsymbol{E} = \begin{bmatrix} (1-(1-\eta)^2)\boldsymbol{A} - \boldsymbol{I} & -\eta\epsilon(1-\eta)\boldsymbol{A} \\ -\eta\epsilon(1-\eta)\boldsymbol{A} & -\epsilon\eta^2 \boldsymbol{A} \end{bmatrix}$,

The dual function of this supremum (with dual variable ν) can be written as:

$$= \inf_{\nu \geq 0} \sup_{\substack{\boldsymbol{z}^{\text{adv}} \\ \boldsymbol{\theta}}} - \left\| \begin{bmatrix} \boldsymbol{\theta} \\ \boldsymbol{z}^{\text{adv}} \end{bmatrix} \right\|_{\boldsymbol{D}^{-1}}^2 + \begin{bmatrix} \boldsymbol{\theta} \\ \boldsymbol{z}^{\text{adv}} \end{bmatrix}^\top \begin{bmatrix} 2(1-\epsilon)\eta(1-\eta)\boldsymbol{A}\mu - 2\mu - \eta \boldsymbol{b} \\ \epsilon \eta \boldsymbol{b} + 2\nu\mu \end{bmatrix}$$

$$+ (1-\epsilon)(\eta^2 \operatorname{Trace}(\boldsymbol{\Sigma}\boldsymbol{A}) + \eta^2 \mu^\top \boldsymbol{A} \mu + \eta \boldsymbol{b}^\top \mu) + \mu^\top \mu + \nu(r - \mu^\top \mu) \tag{20d}$$

where $\boldsymbol{D} = \begin{bmatrix} (1-(1-\eta)^2)\boldsymbol{A} - \boldsymbol{I} & -\eta\epsilon(1-\eta)\boldsymbol{A} \\ -\eta\epsilon(1-\eta)\boldsymbol{A} & -\epsilon\eta^2 \boldsymbol{A} - \nu \boldsymbol{I} \end{bmatrix}$.

The inner supremum is a quadratic expression in $\boldsymbol{z}^{\text{adv}}, \boldsymbol{\theta}$. A finite supremum exists if the Hessian of the expression is negative semifdefinite. Plugging in the tractable maximizer of the quadratic, we get:

$$\inf_{\nu \geq 0} \frac{1}{4} \left\| \begin{bmatrix} 2(1-\epsilon)\eta(1-\eta)\boldsymbol{A}\mu - 2\mu - \eta \boldsymbol{b} \\ \epsilon \eta \boldsymbol{b} + 2\nu\mu \end{bmatrix} \right\|_{\boldsymbol{D}}^2 + (1-\epsilon)(\eta^2 \operatorname{Trace}(\boldsymbol{\Sigma}\boldsymbol{A}) + \eta^2 \mu^\top \boldsymbol{A} \mu + \eta \boldsymbol{b}^\top \mu)$$

$$+ \mu^\top \mu + \eta^2 \operatorname{Trace}(\boldsymbol{A}\boldsymbol{S}) + \nu(r - \mu^\top \mu) \text{ such that } \boldsymbol{D} \succeq 0. \tag{20e}$$

This completes the proof.

□

Lemma B.1. *The stationary distribution in the absence of adversary for the mean estimation problem for* $\mathbb{P}^{\text{data}} = \mathcal{N}(\mu, \boldsymbol{\Sigma})$ *takes the form:*

$$\mathbb{P}\left(\boldsymbol{S}, \mathbb{P}^{\text{data}}\right) = \mathcal{N}(\mu, \eta^2 \boldsymbol{S}).$$

Proof. The stationary distribution is tractable in this case. Recall from Eq. (19), setting $\epsilon = 0$, the transition distribution conditioned on $\boldsymbol{\theta}$ is a Gaussian whose mean is linear in $\boldsymbol{\theta}$. Therefore the stationary distribution:

$$\mathop{\mathbb{E}}_{\boldsymbol{\theta} \sim \mathbb{P}}\left[\mathbb{P}_{\boldsymbol{S}, \mathbb{P}^{\text{data}}}\left(\boldsymbol{\theta}' | \boldsymbol{\theta}\right)\right],$$

will be a Gaussian distribution as a sum of Gaussians is also a Gaussian. Let us assume the distribution has mean $\boldsymbol{m}$. Comparing the means we have:

$$\begin{aligned}\boldsymbol{m}(1-\eta)+\eta\mu &= \boldsymbol{m}\\ \implies \boldsymbol{m} &= \mu.\end{aligned}$$

Moreover, $\mathbb{P}_{\boldsymbol{S},\mathbb{P}^{\text{data}}}(\boldsymbol{\theta}'|\boldsymbol{\theta})$ is a Gaussian with covariance $\eta^2\boldsymbol{S}$ for all $\boldsymbol{\theta}$. Hence the expectation over $\mathbb{P}$ also has covariance $\eta^2\boldsymbol{S}$. This concludes the proof. □

Lemma B.2. *The loss at stationarity of the learning dynamics in the absence of an adversary for the mean estimation problem for $\mathbb{P}^{\text{data}} = \mathcal{N}(\mu, \boldsymbol{\Sigma})$ is given by:*

$$\mathop{\mathbb{E}}_{\boldsymbol{\theta}\sim\mathbb{P}(\boldsymbol{S},\mathbb{P}^{\text{data}})}\left[\ell\left(\boldsymbol{\theta}\right)\right] = \eta^2\,\text{Trace}(\boldsymbol{S}). \tag{21}$$

Proof.

$$\begin{aligned}&\mathop{\mathbb{E}}_{\theta\sim\mathcal{N}(\mu,\eta^2\boldsymbol{S})}\left[\|\theta-\mu\|_2^2\right]\\ &= \mathop{\mathbb{E}}_{\theta\sim\mathcal{N}(0,\eta^2\boldsymbol{S})}\left[\|\theta\|_2^2\right]\\ &= \eta^2\,\text{Trace}(\boldsymbol{S}).\end{aligned}$$

□

Remark B.1. *We use CVXPY [Diamond and Boyd, 2016] to solve the optimization problems in Algorithm 1.*

C Binary Classification

Lemma C.1. *If $\|\boldsymbol{\theta}_0\|_2 \le \frac{1}{\sigma}$, then for all $t > 0$, $\|\boldsymbol{\theta}_t\|_2 \le \frac{1}{\sigma}$.*

Proof. Use Induction and triangle inequality. □

Lemma C.2. *Consider the primal problem (here $\boldsymbol{Q} \succ 0$):*

$$\begin{aligned}&\sup_{\boldsymbol{x},\boldsymbol{y}} -\boldsymbol{x}^\top\boldsymbol{Q}\boldsymbol{x} + \boldsymbol{p}_1^\top\boldsymbol{x} + \boldsymbol{p}_2^\top\boldsymbol{y}\\ &\textit{such that}\\ &\boldsymbol{A}_{1i}^\top\boldsymbol{x} + \boldsymbol{A}_{2i}^\top\boldsymbol{y} \succeq \boldsymbol{c}_i\ \forall i\in[m].\end{aligned}$$

Its dual is the following oprimization problem:

$$\begin{aligned}&\inf_{\nu_1,\ldots,\nu_m\succeq 0} \frac{1}{4}\|\boldsymbol{p}_1 + \sum_{i\in[m]}\boldsymbol{A}_{1i}\nu_i\|^2_{\boldsymbol{Q}^{-1}} - \sum_{i\in[m]}\nu_i^\top\boldsymbol{c}_i\\ &\textit{such that}\\ &\boldsymbol{p}_2 + \sum_{i\in[m]}\boldsymbol{A}_{2i}\nu_i = \boldsymbol{0}.\end{aligned}$$

Proof. We write the primal objective's dual function with Lagrange parameters $\nu_1,\ldots,\nu_m \succeq 0$ as follows:

$$\sup_{\boldsymbol{x},\boldsymbol{y}} -\boldsymbol{x}^\top\boldsymbol{Q}\boldsymbol{x} + (\boldsymbol{p}_1 + \sum_{i\in[m]}\boldsymbol{A}_{1i}\nu_i)^\top\boldsymbol{x} + (\boldsymbol{p}_2 + \sum_{i\in[m]}\boldsymbol{A}_{2i}\nu_i)^\top\boldsymbol{y} - \sum_{i\in[m]}\nu_i^\top\boldsymbol{c}_i.$$

The supremum is maximized for:

$$\boldsymbol{x}^* = \frac{1}{2}\boldsymbol{Q}^{-1}(\boldsymbol{p}_1 + \sum_{i\in[m]}\boldsymbol{A}_{1i}\nu_i),$$

and since we don't have a lower bound on $\boldsymbol{y} \succeq \boldsymbol{0}$, we need $\boldsymbol{p}_2 + \sum_{i\in[m]}\boldsymbol{A}_{2i}\nu_i = \boldsymbol{0}$.

Plugging these in, the dual function completes the proof. □

Theorem 3. *Choosing $\lambda : \mathbb{R}^d \to \mathbb{R}$ in Theorem 1 to be quadratic, i.e., $\lambda(\boldsymbol{\theta}) = \boldsymbol{\theta}^\top \boldsymbol{A}\boldsymbol{\theta} + \boldsymbol{b}^\top\boldsymbol{\theta}$, parameter space $\boldsymbol{\Theta} = \{\boldsymbol{\theta} \mid \|\boldsymbol{\theta}\|_2 \leq \frac{1}{\sigma}\}$ and the adversarial constraint set of the form $\mathcal{A} = \{\boldsymbol{z}^{\text{adv}} \mid \|\boldsymbol{z}^{\text{adv}}\|_2 \leq 1\}$, the certificate for the binary classification problem for $\mathbb{P}^{\text{data}}(\boldsymbol{z}) = \{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_N\}$ for a learning algorithm with regularization parameter σ, and learning rate η is upper bounded by*

$$\max\left(\mathsf{OPT}_1, \mathsf{OPT}_2\right)$$

$$\begin{aligned}
\mathsf{OPT}_1 = \inf_{\nu, \boldsymbol{A}, \boldsymbol{b}} \quad & \|\boldsymbol{p}(\boldsymbol{b}, \nu)\|^2_{\boldsymbol{D}(\boldsymbol{A},\nu)^{-1}} + q(\nu) \\
\text{s.t.} \quad & \boldsymbol{D}(\boldsymbol{A}, \nu) \succeq 0, \\
& \boldsymbol{r}_i(\nu) + \boldsymbol{s}(z_i, \boldsymbol{A}, \boldsymbol{b}) = 0, \ \ \forall i \in [N].
\end{aligned}$$

$$\begin{aligned}
\mathsf{OPT}_2 = \inf_{\nu, \boldsymbol{A}, \boldsymbol{b}} \quad & \|\boldsymbol{p}'(\boldsymbol{b}, \nu)\|^2_{\boldsymbol{D}'(\boldsymbol{A},\nu)^{-1}} + q'(\nu) \\
\text{s.t.} \quad & \boldsymbol{D}'(\boldsymbol{A}, \nu) \succeq 0, \\
& \boldsymbol{r}_i(\nu) + \boldsymbol{s}(z_i, \boldsymbol{A}, \boldsymbol{b}) = 0, \ \ \forall i \in [N].
\end{aligned}$$

where $\boldsymbol{p}()$, $\boldsymbol{p}'()$, $q()$, $q'()$, $\boldsymbol{D}()$, $\boldsymbol{D}'()$, $\boldsymbol{r}_1(), \ldots, \boldsymbol{r}_N()$, $\boldsymbol{s}()$ are affine functions of the optimization variables $\nu, \boldsymbol{A}, \boldsymbol{b}$ as defined below and $\nu = \{\nu_1, \ldots, \nu_{10}\}$:

$$\begin{aligned}
\boldsymbol{p}(\boldsymbol{b}, \nu) &= \frac{1}{2}\begin{bmatrix} -\sigma\eta\boldsymbol{b} + \sum_{i\in[N]}\left((\nu_{1i} - \nu_{2i})\boldsymbol{z}_i - \nu_{4i} + \nu_{6i}\right) \\ \epsilon\eta\boldsymbol{b} \end{bmatrix}, \\
\boldsymbol{D}(\boldsymbol{A}, \nu) &= \begin{bmatrix} [1-(1-\sigma\eta)^2]\boldsymbol{A} + \nu_8\boldsymbol{I} & -\epsilon(1-\sigma\eta)\eta\boldsymbol{A} + \nu_9\boldsymbol{I} \\ -\epsilon(1-\sigma\eta)\eta\boldsymbol{A} + \nu_9\boldsymbol{I} & -\epsilon\eta^2\boldsymbol{A} + \nu_{10}\boldsymbol{I} \end{bmatrix}, \\
q(\nu) &= q'(\nu) + 2\nu_9 + \nu_{10}, \\
\boldsymbol{p}'(\boldsymbol{b}, \nu) &= \frac{1}{2}\left[-\sigma\eta\boldsymbol{b} + \sum_{i\in[N]}\left((\nu_{1i} - \nu_{2i})\boldsymbol{z}_i - \nu_{4i} + \nu_{6i}\right)\right], \\
\boldsymbol{D}'(\boldsymbol{A}, \nu) &= [1-(1-\sigma\eta)^2]\boldsymbol{A} + \nu_8\boldsymbol{I}, \\
q'(\nu) &= -\nu_1^\top\mathbf{1} + (2 + \tfrac{1}{\sigma})\nu_2^\top\mathbf{1} + \tfrac{(\nu_4+\nu_6)^\top\mathbf{1}}{\sigma} + \mathbf{1}^\top\nu_7 + \frac{\nu_8}{\sigma^2}, \\
\boldsymbol{r}_i(\nu) &= \begin{bmatrix} (1+\frac{1}{\sigma})(\nu_{1i} - \nu_{2i}) - \frac{\mathbf{1}^\top(\nu_{4i}+\nu_{6i})}{\sigma} - \nu_{7i} \\ \nu_{3i} + \nu_{4i} - \nu_{5i} - \nu_{6i} \end{bmatrix}, \\
\boldsymbol{s}(\boldsymbol{z}_i, \boldsymbol{A}, \boldsymbol{b}) &= \frac{1}{N}\begin{bmatrix} (1-\epsilon)\eta^2\boldsymbol{z}_i^\top\boldsymbol{A}\boldsymbol{z}_i + (1-\epsilon)\eta\boldsymbol{b}^\top\boldsymbol{z}_i + 1 \\ 2(1-\epsilon)\eta(1-\sigma\eta)\boldsymbol{A}\boldsymbol{z}_i - \boldsymbol{z}_i \end{bmatrix}.
\end{aligned}$$

Proof. Since ϵ fraction of the samples are corrupted by the adversary, the transition distribution conditioned on $\boldsymbol{z}^{\text{adv}}$, the benign distribution $\mathbb{P}^{\text{data}}$ and the defense parameter σ is given by:

$$\begin{aligned}
\mathbb{P}_{\sigma, \mathbb{P}^{\text{data}}, \boldsymbol{z}^{\text{adv}}}(\boldsymbol{\theta}' | \boldsymbol{\theta}) &= \epsilon F(\boldsymbol{\theta}, \boldsymbol{z}^{\text{adv}}) + (1-\epsilon)\mathop{\mathbb{E}}_{\boldsymbol{z}\sim\mathbb{P}^{\text{data}}}[F(\boldsymbol{\theta}, \boldsymbol{z})] \\
&= \epsilon((1-\sigma\eta)\boldsymbol{\theta} + \eta\mathbb{I}[\boldsymbol{\theta}^\top\boldsymbol{z}^{\text{adv}} \leq 1]\boldsymbol{z}^{\text{adv}}) + (1-\epsilon)((1-\sigma\eta)\boldsymbol{\theta} + \eta\mathop{\mathbb{E}}_{\boldsymbol{z}\sim\mathbb{P}^{\text{data}}}\left[\mathbb{I}[\boldsymbol{\theta}^\top\boldsymbol{z} \leq 1]\boldsymbol{z}\right]).
\end{aligned}$$

We wish to analyse the adversarial loss at stationarity. We consider the following 2 cases at stationarity. Consider $\boldsymbol{\Theta}_1, \mathcal{A}_1$ as the space such that (i) $\mathbb{I}[\boldsymbol{\theta}^\top\boldsymbol{z}^{\text{adv}} \leq 1] = 0$, the transition distribution at stationarity looks like:

$$\begin{aligned}
\mathbb{P}_{\sigma, \mathbb{P}^{\text{data}}, \boldsymbol{z}^{\text{adv}}}(\boldsymbol{\theta}' | \boldsymbol{\theta}) &= \epsilon F(\boldsymbol{\theta}, \boldsymbol{z}^{\text{adv}}) + (1-\epsilon)\mathop{\mathbb{E}}_{\boldsymbol{z}\sim\mathbb{P}^{\text{data}}}[F(\boldsymbol{\theta}, \boldsymbol{z})] \\
&= \epsilon((1-\sigma\eta)\boldsymbol{\theta} + \eta\mathbb{I}[\boldsymbol{\theta}^\top\boldsymbol{z}^{\text{adv}} \leq 1]\boldsymbol{z}^{\text{adv}}) + (1-\epsilon)((1-\sigma\eta)\boldsymbol{\theta} + \eta\mathop{\mathbb{E}}_{\boldsymbol{z}\sim\mathbb{P}^{\text{data}}}\left[\mathbb{I}[\boldsymbol{\theta}^\top\boldsymbol{z} \leq 1]\boldsymbol{z}\right]) \\
&= (1 - \frac{\sigma}{1-\epsilon}(1-\epsilon)\eta)\boldsymbol{\theta} + (1-\epsilon)\eta\mathop{\mathbb{E}}_{\boldsymbol{z}\sim\mathbb{P}^{\text{data}}}\left[\mathbb{I}[\boldsymbol{\theta}^\top\boldsymbol{z} \leq 1]\boldsymbol{z}\right]
\end{aligned}$$

This can be interpreted as the stationary distribution in the absence of an adversary with learning rate $(1-\epsilon)\eta$ and regularization $\frac{\sigma}{(1-\epsilon)}$.

The other case is $\boldsymbol{\Theta}_2, \mathcal{A}_2$ such that (ii) $\mathbb{I}[\boldsymbol{\theta}^\top\boldsymbol{z}^{\text{adv}} \leq 1] = 1$.

Note that $\{\boldsymbol{\Theta}_1 \times \mathcal{A}_1\} \cup \{\boldsymbol{\Theta}_2 \times \mathcal{A}_2\} = \boldsymbol{\Theta} \times \mathcal{A}$. We can treat both these cases separately in our original problem Eq. 8 before going to the formulation in Eq. 9. Thus we aim to find a certificate for each of these cases and we do so via the formulation in Eq. 9 and take the max of these upper bounds.

Choosing $\lambda(\boldsymbol{\theta}) = \boldsymbol{\theta}^\top \boldsymbol{A}\boldsymbol{\theta} + \boldsymbol{\theta}^\top \boldsymbol{b}$ to be a quadratic function, we derive the certificate for a fixed σ:

$$\begin{aligned}
\text{OPT} = &\sup_{\substack{\boldsymbol{z}^{\text{adv}} \in \mathcal{A} \\ \boldsymbol{\theta}:\|\boldsymbol{\theta}\|_2 \le \frac{1}{\sigma}}} \epsilon\left(\lambda\left(F\left(\boldsymbol{\theta}, \boldsymbol{z}^{\text{adv}}\right)\right)\right) + (1-\epsilon) \mathop{\mathbb{E}}_{\boldsymbol{z}\sim\mathbb{P}^{\text{data}}}\left[\lambda\left(F\left(\boldsymbol{\theta}, \boldsymbol{z}\right)\right)\right] - \lambda(\boldsymbol{\theta}) + \ell_{\text{adv}}(\boldsymbol{\theta}) \\
= &\sup_{\substack{\boldsymbol{z}^{\text{adv}} \in \mathcal{A} \\ \boldsymbol{\theta}:\|\boldsymbol{\theta}\|_2 \le \frac{1}{\sigma}}} [(1-\sigma\eta)^2 - 1]\boldsymbol{\theta}^\top \boldsymbol{A}\boldsymbol{\theta} + \epsilon\eta^2 \mathbb{I}[\boldsymbol{\theta}^\top \boldsymbol{z}^{\text{adv}} \le 1] {\boldsymbol{z}^{\text{adv}}}^\top \boldsymbol{A}\boldsymbol{z}^{\text{adv}} + 2\epsilon(1-\sigma\eta)\eta \mathbb{I}[\boldsymbol{\theta}^\top \boldsymbol{z}^{\text{adv}} \le 1]\boldsymbol{\theta}^\top \boldsymbol{A}\boldsymbol{z}^{\text{adv}} \\
&+ \epsilon\eta \mathbb{I}[\boldsymbol{\theta}^\top \boldsymbol{z}^{\text{adv}} \le 1]\boldsymbol{b}^\top \boldsymbol{z}^{\text{adv}}] + (1-\epsilon) \mathop{\mathbb{E}}_{\boldsymbol{z}\sim\mathbb{P}^{\text{data}}}\left[\eta^2 \mathbb{I}[\boldsymbol{\theta}^\top \boldsymbol{z} \le 1]\boldsymbol{z}^\top \boldsymbol{A}\boldsymbol{z} + 2\eta(1-\sigma\eta)\mathbb{I}[\boldsymbol{\theta}^\top \boldsymbol{z} \le 1]\boldsymbol{\theta}^\top \boldsymbol{A}\boldsymbol{z} + \eta \mathbb{I}[\boldsymbol{\theta}^\top \boldsymbol{z} \le 1]\boldsymbol{b}^\top \boldsymbol{z}\right] \\
&- \sigma\eta \boldsymbol{b}^\top \boldsymbol{\theta} + \mathop{\mathbb{E}}_{z\sim\mathbb{P}^{\text{data}}}\left[\max\{0, 1 - \boldsymbol{\theta}^\top \boldsymbol{z}\}\right]
\end{aligned}$$

(Using sample average approximation for $\mathbb{P}^{\text{data}}$ with data points from the training data set we get:)

$$\begin{aligned}
= &\sup_{\substack{\boldsymbol{z}^{\text{adv}} \in \mathcal{A} \\ \boldsymbol{\theta}:\|\boldsymbol{\theta}\|_2 \le \frac{1}{\sigma} \\ q_i \in \{0,1\} \forall i \in [N]}} [(1-\sigma\eta)^2 - 1]\boldsymbol{\theta}^\top \boldsymbol{A}\boldsymbol{\theta} + \epsilon\eta^2 \mathbb{I}[\boldsymbol{\theta}^\top \boldsymbol{z}^{\text{adv}} \le 1] {\boldsymbol{z}^{\text{adv}}}^\top \boldsymbol{A}\boldsymbol{z}^{\text{adv}} + 2\epsilon(1-\sigma\eta)\eta \mathbb{I}[\boldsymbol{\theta}^\top \boldsymbol{z}^{\text{adv}} \le 1]\boldsymbol{\theta}^\top \boldsymbol{A}\boldsymbol{z}^{\text{adv}} \\
&+ \epsilon\eta \mathbb{I}[\boldsymbol{\theta}^\top \boldsymbol{z}^{\text{adv}} \le 1]\boldsymbol{b}^\top \boldsymbol{z}^{\text{adv}} + (1-\epsilon)\frac{1}{N}\sum_{i\in[N]}\left[\eta^2 q_i \boldsymbol{z}_i^\top \boldsymbol{A}\boldsymbol{z}_i + 2\eta(1-\sigma\eta)q_i\boldsymbol{\theta}^\top \boldsymbol{A}\boldsymbol{z}_i + \eta q_i \boldsymbol{b}^\top \boldsymbol{z}_i\right] + \\
&\frac{1}{N}\sum_{i\in[N]}\left[q_i(1 - \boldsymbol{\theta}^\top \boldsymbol{z}_i)\right] - \sigma\eta\boldsymbol{b}^\top\boldsymbol{\theta} \\
&\text{such that } 1 - \boldsymbol{\theta}^\top \boldsymbol{z}_i \le (1 + \frac{d}{\sigma})q_i, 1 - \boldsymbol{\theta}^\top \boldsymbol{z}_i \ge -(1+\frac{d}{\sigma})(1-q_i) \forall i \in [N].
\end{aligned}$$

To get rid of the indicator variable $\mathbb{I}[\boldsymbol{\theta}^\top \boldsymbol{z}^{\text{adv}} \le 1]$, we can write the certificate as the maximum of 2 optimization problems, (i) OPT_1 with $\mathbb{I}[\boldsymbol{\theta}^\top \boldsymbol{z}^{\text{adv}} \le 1] = 0$ and, (ii) OPT_2 with $\mathbb{I}[\boldsymbol{\theta}^\top \boldsymbol{z}^{\text{adv}} \le 1] = 1$. Note that the optimization problem in OPT_1 doesn't have $\boldsymbol{z}^{\text{adv}}$ as a decision variable. We first focus on the relaxations on the optimization problem in OPT_2 and a bound on the optimization problem in OPT_1 can be obtained by dropping the terms in $\boldsymbol{z}^{\text{adv}}$ from OPT_2.

$$\begin{aligned}
\mathsf{OPT}_2 = &\sup_{\substack{\boldsymbol{z}^{\text{adv}} \in \mathcal{A} \\ \boldsymbol{\theta}:\|\boldsymbol{\theta}\|_2 \le \frac{1}{\sigma} \\ q_i \in \{0,1\} \forall i \in [N]}} [(1-\sigma\eta)^2 - 1]\boldsymbol{\theta}^\top \boldsymbol{A}\boldsymbol{\theta} + \epsilon\eta^2 {\boldsymbol{z}^{\text{adv}}}^\top \boldsymbol{A}\boldsymbol{z}^{\text{adv}} + 2\epsilon(1-\sigma\eta)\eta\boldsymbol{\theta}^\top \boldsymbol{A}\boldsymbol{z}^{\text{adv}} + \epsilon\eta\boldsymbol{b}^\top \boldsymbol{z}^{\text{adv}} \\
&+ (1-\epsilon)\frac{1}{N}\sum_{i\in[N]}\left[\eta^2 q_i \boldsymbol{z}_i^\top \boldsymbol{A}\boldsymbol{z}_i + 2\eta(1-\sigma\eta)q_i\boldsymbol{\theta}^\top \boldsymbol{A}\boldsymbol{z}_i + \eta q_i \boldsymbol{b}^\top \boldsymbol{z}_i\right] + \frac{1}{N}\sum_{i\in[N]}\left[q_i(1 - \boldsymbol{\theta}^\top \boldsymbol{z}_i)\right] - \sigma\eta\boldsymbol{b}^\top\boldsymbol{\theta} \\
&\text{such that } \boldsymbol{\theta}^\top \boldsymbol{z}^{\text{adv}} \le 1, 1 - \boldsymbol{\theta}^\top \boldsymbol{z}_i \le (1 + \frac{1}{\sigma})q_i, 1 - \boldsymbol{\theta}^\top \boldsymbol{z}_i \ge -(1+\frac{1}{\sigma})(1-q_i) \forall i \in [N] \\
&\text{(Relaxing integer variables } q_i\text{'s to continuous variables)} \\
\le &\sup_{\substack{\boldsymbol{z}^{\text{adv}} \in \mathcal{A} \\ \boldsymbol{\theta}:\|\boldsymbol{\theta}\|_2 \le \frac{1}{\sigma} \\ q_i \in [0,1] \forall i \in [N]}} [(1-\sigma\eta)^2 - 1]\boldsymbol{\theta}^\top \boldsymbol{A}\boldsymbol{\theta} + \epsilon\eta^2 {\boldsymbol{z}^{\text{adv}}}^\top \boldsymbol{A}\boldsymbol{z}^{\text{adv}} + 2\epsilon(1-\sigma\eta)\eta\boldsymbol{\theta}^\top \boldsymbol{A}\boldsymbol{z}^{\text{adv}} + \epsilon\eta\boldsymbol{b}^\top \boldsymbol{z}^{\text{adv}} \\
&+ (1-\epsilon)\frac{1}{N}\sum_{i\in[N]}\left[\eta^2 q_i \boldsymbol{z}_i^\top \boldsymbol{A}\boldsymbol{z}_i + 2\eta(1-\sigma\eta)q_i\boldsymbol{\theta}^\top \boldsymbol{A}\boldsymbol{z}_i + \eta q_i \boldsymbol{b}^\top \boldsymbol{z}_i\right] + \frac{1}{N}\sum_{i\in[N]}\left[q_i(1 - \boldsymbol{\theta}^\top \boldsymbol{z}_i)\right] - \sigma\eta\boldsymbol{b}^\top\boldsymbol{\theta} \\
&\text{such that } \boldsymbol{\theta}^\top \boldsymbol{z}^{\text{adv}} \le 1, 1 - \boldsymbol{\theta}^\top \boldsymbol{z}_i \le (1 + \frac{1}{\sigma})q_i, 1 - \boldsymbol{\theta}^\top \boldsymbol{z}_i \ge -(1+\frac{1}{\sigma})(1-q_i) \forall i \in [N].
\end{aligned}$$

Using McCormick relaxations for bilinear terms $q_i\boldsymbol{\theta}$, we get:

$$\begin{aligned}
\mathsf{OPT}_2 \leq \sup_{\substack{\boldsymbol{z}^{\text{adv}}\in\mathcal{A},\boldsymbol{\theta}:\|\boldsymbol{\theta}\|_2\leq\frac{1}{\sigma},\\ \boldsymbol{q}\in\mathbb{R}^N,\boldsymbol{w}\in\mathbb{R}^{dN}}} & -\begin{bmatrix}\boldsymbol{\theta}\\ \boldsymbol{z}^{\text{adv}}\end{bmatrix}^\top \begin{bmatrix}[1-(1-\sigma\eta)^2]\boldsymbol{A} & -\epsilon(1-\sigma\eta)\eta\boldsymbol{A}\\ -\epsilon(1-\sigma\eta)\eta\boldsymbol{A} & -\epsilon\eta^2\boldsymbol{A}\end{bmatrix}\begin{bmatrix}\boldsymbol{\theta}\\ \boldsymbol{z}^{\text{adv}}\end{bmatrix} + \\
& \begin{bmatrix}-\sigma\eta\boldsymbol{b}\\ \epsilon\eta\boldsymbol{b}\end{bmatrix}^\top \begin{bmatrix}\boldsymbol{\theta} & \boldsymbol{z}^{\text{adv}}\end{bmatrix} + \frac{1}{N}\sum_{1=1}^{N}\begin{bmatrix}(1-\epsilon)\eta^2\boldsymbol{z}_i^\top\boldsymbol{A}\boldsymbol{z}_i+(1-\epsilon)\eta\boldsymbol{b}^\top\boldsymbol{z}_i+1\\ 2(1-\epsilon)\eta(1-\sigma\eta)\boldsymbol{A}\boldsymbol{z}_i-\boldsymbol{z}_i\end{bmatrix}^\top \begin{bmatrix}q_i & \boldsymbol{w}_i\end{bmatrix}\\
& \text{such that}\\
& \boldsymbol{z}_i^\top\boldsymbol{\theta}+(1+\frac{1}{\sigma})q_i\geq 1\ \forall i\in[N],\\
& -\boldsymbol{z}_i^\top\boldsymbol{\theta}-(1+\frac{1}{\sigma})q_i\geq -(2+\frac{1}{\sigma})\ \forall i\in[N],\\
& q_i\mathbf{1}/\sigma+\boldsymbol{w}_i\succeq\mathbf{0}\ \forall i\in[N],\\
& -\boldsymbol{\theta}-q_i\mathbf{1}/\sigma+\boldsymbol{w}_i\succeq-\mathbf{1}/\sigma\ \forall i\in[N],\\
& q_i\mathbf{1}/\sigma-\boldsymbol{w}_i\succeq\mathbf{0}\ \forall i\in[N],\\
& \boldsymbol{\theta}-q_i\mathbf{1}/\sigma-\boldsymbol{w}_i\succeq-\mathbf{1}/\sigma\ \forall i\in[N],\\
& \boldsymbol{q}\succeq\mathbf{0},\\
& -\boldsymbol{q}\succeq-\mathbf{1},\\
& 1-\boldsymbol{\theta}^\top\boldsymbol{z}^{\text{adv}}\geq 0.
\end{aligned}$$

From the Mccormick envelope constraints, and the constraints on $\boldsymbol{\theta}$ and $\boldsymbol{q}$, we can derive that $\boldsymbol{w}_i\geq 0\ \forall i\in[N]$. Plugging $\mathcal{A}=\{\boldsymbol{z}^{\text{adv}}|\|\boldsymbol{z}^{\text{adv}}\|_2\leq 1\}$. Now we utilize Lemma C to write the dual problem as:

$$\begin{aligned}
\inf_{\substack{\nu_1,\nu_2,\nu_7\in\mathbb{R}_+^N\\ \nu_3,\nu_4,\nu_5,\nu_6\in\mathbb{R}_+^{N\times d}\\ \nu_8,\nu_9,\nu_{10}\in\mathbb{R}_+}} & \frac{1}{4}\|\begin{bmatrix}-\sigma\eta\boldsymbol{b}+\sum_{i\in[N]}((\nu_{1i}-\nu_{2i})\boldsymbol{z}_i-\nu_{4i}+\nu_{6i})\\ \epsilon\eta\boldsymbol{b}\end{bmatrix}\|^2_{\boldsymbol{D}^{-1}}\\
& -\nu_1^\top\mathbf{1}+(2+\frac{1}{\sigma})\nu_2^\top\mathbf{1}+\frac{(\nu_4+\nu_6)^\top\mathbf{1}}{\sigma}+\mathbf{1}^\top\nu_7+\frac{\nu_8}{\sigma^2}+2\nu_9+\nu_{10}.\\
& \text{such that}\\
& \begin{bmatrix}(1+\frac{1}{\sigma})(\nu_{1i}-\nu_{2i})-\frac{\mathbf{1}^\top(\nu_{4i}+\nu_{6i})}{\sigma}-\nu_{7i}\\ \nu_{3i}+\nu_{4i}-\nu_{5i}-\nu_{6i}\end{bmatrix}+\boldsymbol{s}(\boldsymbol{z}_i,\boldsymbol{A},\boldsymbol{b})\preceq\mathbf{0}\ \forall i\in[N].\\
& \boldsymbol{D}\succeq 0
\end{aligned}$$

where

$$\boldsymbol{D}=\begin{bmatrix}[1-(1-\sigma\eta)^2]\boldsymbol{A}+\nu_8\boldsymbol{I} & -\epsilon(1-\sigma\eta)\eta\boldsymbol{A}+\nu_9\\ -\epsilon(1-\sigma\eta)\eta\boldsymbol{A}+\nu_9 & -\epsilon\eta^2\boldsymbol{A}+\nu_{10}\boldsymbol{I}\end{bmatrix}.$$

We use $\nu\succeq\mathbf{0}$ to compactly denote $\{\nu_1,\ldots,\nu_{10}\}$. Define the following affine functions in $\nu,\boldsymbol{A},\boldsymbol{b}$:

$$\begin{aligned}
\boldsymbol{p}(\boldsymbol{b},\nu) &= \frac{1}{2}\begin{bmatrix}-\sigma\eta\boldsymbol{b}+\sum_{i\in[N]}((\nu_{1i}-\nu_{2i})\boldsymbol{z}_i-\nu_{4i}+\nu_{6i})\\ \epsilon\eta\boldsymbol{b}\end{bmatrix}\in\mathbb{R}^{2d},\\
\boldsymbol{D}(\boldsymbol{A},\nu) &= \begin{bmatrix}[1-(1-\sigma\eta)^2]\boldsymbol{A}+\nu_8\boldsymbol{I} & -\epsilon(1-\sigma\eta)\eta\boldsymbol{A}+\nu_9\\ -\epsilon(1-\sigma\eta)\eta\boldsymbol{A}+\nu_9 & -\epsilon\eta^2\boldsymbol{A}+\nu_{10}\boldsymbol{I}\end{bmatrix}\in\mathbb{S}^{2d}_+,\\
q(\nu) &= -\nu_1^\top\mathbf{1}+(2+\frac{1}{\sigma})\nu_2^\top\mathbf{1}+\frac{(\nu_4+\nu_6)^\top\mathbf{1}}{\sigma}+\mathbf{1}^\top\nu_7+\frac{\nu_8}{\sigma^2}+2\nu_9+\nu_{10}\in\mathbb{R},\\
\boldsymbol{r}_i(\nu) &= \begin{bmatrix}(1+\frac{1}{\sigma})(\nu_{1i}-\nu_{2i})-\frac{\mathbf{1}^\top(\nu_{4i}+\nu_{6i})}{\sigma}-\nu_{7i}\\ \nu_{3i}+\nu_{4i}-\nu_{5i}-\nu_{6i}\end{bmatrix}\in\mathbb{R}^{d+1},\\
\boldsymbol{s}(\boldsymbol{z}_i,\boldsymbol{A},\boldsymbol{b}) &= \frac{1}{N}\begin{bmatrix}((1-\epsilon)\eta^2\boldsymbol{z}_i^\top\boldsymbol{A}\boldsymbol{z}_i+(1-\epsilon)\eta\boldsymbol{b}^\top\boldsymbol{z}_i+1)\\ (2(1-\epsilon)\eta(1-\sigma\eta)\boldsymbol{A}\boldsymbol{z}_i-\boldsymbol{z}_i)\end{bmatrix}\in\mathbb{R}^{d+1}.
\end{aligned}$$

The certificate can thus be written compactly as:

$$\begin{aligned}\mathsf{OPT}_2 \leq \inf_{\nu \succeq \mathbf{0}} \; & \|\boldsymbol{p}(\boldsymbol{b}, \nu)\|^2_{\boldsymbol{D}(\boldsymbol{A},\nu)^{-1}} + q(\nu) \\ & \text{such that} \\ & \boldsymbol{r}_i(\nu) + \boldsymbol{s}(z_i, \boldsymbol{A}, \boldsymbol{b}) \preceq 0 \; \forall i \in [N], \boldsymbol{D}(\boldsymbol{A}, \nu) \succ \mathbf{0}.\end{aligned}$$

Similarly OPT_1 can be upper bounded as:

$$\begin{aligned}\mathsf{OPT}_1 \leq \inf_{\nu \succeq \mathbf{0}} \; & \|\boldsymbol{p}'(\boldsymbol{b}, \nu)\|^2_{\boldsymbol{D}'(\boldsymbol{A},\nu)^{-1}} + q'(\nu) \\ & \text{such that} \\ & \boldsymbol{r}_i(\nu) + \boldsymbol{s}(z_i, \boldsymbol{A}, \boldsymbol{b}) \preceq 0 \; \forall i \in [N], \boldsymbol{D}'(\boldsymbol{A}, \nu) \succ \mathbf{0}.\end{aligned}$$

where:

$$\begin{aligned}&\boldsymbol{p}'(\boldsymbol{b}, \nu) = \frac{1}{2}\left[-\sigma\eta\boldsymbol{b} + \textstyle\sum_{i\in[N]}\left((\nu_{1i} - \nu_{2i})\boldsymbol{z}_i - \nu_{4i} + \nu_{6i}\right)\right] \in \mathbb{R}^d, \\ &\boldsymbol{D}'(\boldsymbol{A}, \nu) = [1 - (1 - \sigma\eta)^2]\boldsymbol{A} + \nu_8 \boldsymbol{I} \in \mathbb{S}^d_+, \\ &q'(\nu) = -\nu_1^\top \mathbf{1} + (1 + \frac{1}{\sigma})\nu_2^\top \mathbf{1} + \frac{(\nu_4 + \nu_6)^\top \mathbf{1}}{\sigma} + \frac{\nu_8}{\sigma^2} \in \mathbb{R}.\end{aligned}$$

□

$$\begin{aligned}\mathsf{OPT}_2 \leq \sup_{\substack{\boldsymbol{z}^{\text{adv}} \in \mathcal{A}, \boldsymbol{\theta}: \|\boldsymbol{\theta}\|_2 \leq \frac{1}{\sigma}, \\ \boldsymbol{q} \in \mathbb{R}^N, \boldsymbol{w} \in \mathbb{R}^{dN}}} & -\begin{bmatrix}\boldsymbol{\theta} \\ \boldsymbol{z}^{\text{adv}}\end{bmatrix}^\top \begin{bmatrix}[1 - (1 - \sigma\eta)^2]\boldsymbol{A} & -\epsilon(1 - \sigma\eta)\eta\boldsymbol{A} \\ -\epsilon(1 - \sigma\eta)\eta\boldsymbol{A} & -\epsilon\eta^2\boldsymbol{A}\end{bmatrix}\begin{bmatrix}\boldsymbol{\theta} \\ \boldsymbol{z}^{\text{adv}}\end{bmatrix} + \\ & \begin{bmatrix}-\sigma\eta\boldsymbol{b} \\ \epsilon\eta\boldsymbol{b}\end{bmatrix}^\top \begin{bmatrix}\boldsymbol{\theta} \\ \boldsymbol{z}^{\text{adv}}\end{bmatrix} + \frac{1}{N}\sum_{1=1}^{N}\begin{bmatrix}1 \\ (1 - \epsilon)\eta^2 \boldsymbol{z}_i^\top \boldsymbol{A}\boldsymbol{z}_i + (1 - \epsilon)\eta\boldsymbol{b}^\top \boldsymbol{z}_i \\ 2(1 - \epsilon)\eta(1 - \sigma\eta)\boldsymbol{A}\boldsymbol{z}_i\end{bmatrix}^\top \begin{bmatrix}p_i \\ q_i \\ \boldsymbol{w}_i\end{bmatrix} \\ & \text{such that} \\ & \boldsymbol{z}_i^\top \boldsymbol{\theta} + (1 + \frac{1}{\sigma})q_i \geq 1 \; \forall i \in [N], \\ & -\boldsymbol{z}_i^\top \boldsymbol{\theta} - (1 + \frac{1}{\sigma})q_i \geq -(2 + \frac{1}{\sigma}) \; \forall i \in [N], \\ & q_i \mathbf{1}/\sigma + \boldsymbol{w}_i \succeq \mathbf{0} \; \forall i \in [N], \\ & -\boldsymbol{\theta} - q_i \mathbf{1}/\sigma + \boldsymbol{w}_i \succeq -\mathbf{1}/\sigma \; \forall i \in [N], \\ & q_i \mathbf{1}/\sigma - \boldsymbol{w}_i \succeq \mathbf{0} \; \forall i \in [N], \\ & \boldsymbol{\theta} - q_i \mathbf{1}/\sigma - \boldsymbol{w}_i \succeq -\mathbf{1}/\sigma \; \forall i \in [N], \\ & \boldsymbol{q} \succeq \mathbf{0}, \\ & -\boldsymbol{q} \succeq -\mathbf{1}, \\ & \boldsymbol{p} \succeq \mathbf{0}, \\ & -\boldsymbol{p} \succeq -\mathbf{1}, \\ & 1 - \boldsymbol{\theta}^\top \boldsymbol{z}^{\text{adv}} \geq 0, \\ & \boldsymbol{\theta}^\top \boldsymbol{z}_i + \frac{1}{\sigma}p_i \geq 0 \\ & -\boldsymbol{\theta}^\top \boldsymbol{z}_i - \frac{1}{\sigma}p_i \geq -\frac{1}{\sigma}\end{aligned}$$

D Experiment Details

D.1 Vision Datasets

We use the following pre-processing for all datasets. Use pretrained ResNet18 on ImageNET to extract features as the output of the pre-final layer. Perform Singular Value Decomposition (SVD) to select the top $d = 30$ directions with largest feature variation. We normalise our dataset to ensure zero mean, append 1 to each feature vector to capture the bias term, and scale to ensure each datapoint has ℓ_2 norm less than or equal to 1. We multiply the $\{+1, -1\}$ labels to our features.

We split the training set into $\mathbb{P}^{\text{data}}$ and $\mathbb{P}^{\text{target}}$ which are used to compute the certificates. For the simulation of the learning algorithms with various attacks, the same $\mathbb{P}^{\text{data}}$ and $\mathbb{P}^{\text{target}}$ are used. The evaluation of these learning algorithms is done on a test set which is distributionally similar to $\mathbb{P}^{\text{target}}$.

D.2 Reward Modeling

We use the following pre-processing the HelpSteer dataset. We pass as inputs a concatenated text of the prompt, response pair to a pretrained BERT model to extract features as the output of the pre-final layer. We then perform Singular Value Decomposition (SVD) to select the top $d = 30$ directions with largest feature variation. We normalise our dataset to ensure zero mean, append 1 to each feature vector to capture the bias term, and scale to ensure each datapoint has ℓ_2 norm less than or equal to 1.

The scores on each attribute in the raw dataset are natural numbers between $[0, 4]$. We linearly scale it to lie in the range $[-1, 1]$. We then multiply the labels to our features.

We split the training set into $\mathbb{P}^{\text{data}}$ and $\mathbb{P}^{\text{target}}$ which are used to compute the certificates. For the simulation of the learning algorithms with various attacks, the same $\mathbb{P}^{\text{data}}$ and $\mathbb{P}^{\text{target}}$ are used. The evaluation of these learning algorithms is done on a test set which is distributionally similar to $\mathbb{P}^{\text{target}}$.