Certifying robustness to adaptive data poisoning

Anonymous Author(s)
Affiliation
Address
email

Abstract

The rise of foundational models fine-tuned with 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. While existing research focuses on certifying robustness for static adversaries acting on offline datasets, dynamic attack algorithms have shown to be more effective. Relevant for models with periodic updates where an adversary can adapt based on the algorithm's behavior, such as those in RLHF, 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 problem.

1 Introduction & Problem Formulation

With the advent of foundational models fine tuned using human feedback gathered from potentially untrusted users (for example, users of a publicly available language model) [3, 7], 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 [2]. 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 [9, 13]. Even in [10] 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 [12, 11] showing that these attacks can indeed be more powerful than static adversaries. 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 learning 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. We provide (to the best of our knowledge) the first general framework for computing certified bounds on the worst case impact of a data poisoning attacker, and further, use this certificate to design robust learning algorithms. We given an illustration of the framework for the mean-estimation problem (see Section 2), and aim to leverage this framework for regression, classification and generative models in future work.

Submitted to 38th Conference on Neural Information Processing Systems (NeurIPS 2024). Do not distribute.

Learning objective We study learning problems where the goal is to minimize

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

where $\boldsymbol{\theta} \in \mathbb{R}^d$ are parameters to be estimated (for example parameters of a generative model, classification model or a regression model), $\ell : \mathbb{R}^d \times \mathbb{R}^n \mapsto \mathbb{R}$ is a loss function and $\boldsymbol{z} \in \mathbb{R}^n$ are i.i.d. samples from an underlying data distribution $\mathbb{P}^{\text{data}}$.

Adversarially corrupted learning algorithm We work in a setting where the learning is being done in an online fashion and the corrupted datapoint can be updated after every step of learning, based on the trajectory of the learning process observed by the adversary. We consider learning algorithms of the form

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

where $\boldsymbol{w}_t \sim \mathcal{N}\left(0, \boldsymbol{I}\right)$ is chosen iid at each t and $\boldsymbol{z}_t \sim \epsilon \text{Dirac}(\boldsymbol{z}_t^{\text{adv}}) + (1-\epsilon)\mathbb{P}^{\text{data}}$ and $\boldsymbol{B} \in \mathbb{R}^{d \times d}$ is a design parameter of the learning algorithm that is described below (see Potential Defense) and ϵ is a parameter that controls the "level" of poisoning (analogous to the fraction of poisoned samples). This is a special case of Huber's contamination model, which is used in the robust statistics literature [4] (with the contamination model being a Dirac distribution). Further, there are typically allowed ranges for the datapoints that come from the learning algorithm normalizing inputs or by an outlier detection system used to filter potential adversarial data. In this preliminary work, we restrict ourselves to norm balls, $\mathcal{A} = \{\boldsymbol{z} : \|\boldsymbol{z}\| \leq r\}$.

Potential Defense Inspired by differentially private learning algorithms like DP-SGD [1], we propose adding Gaussian noise to the learning process as a way of smoothing the learning algorithm against impacts of the poisoning adversary. In particular, we add $\boldsymbol{B}\boldsymbol{w}_t$ where $\boldsymbol{w}_t$ is iid noise in each step sampled from the standard Gaussian, and $\boldsymbol{B}$ is a design parameter of the learning algorithm. Subsequently, we will choose $\boldsymbol{B}$ so as to minimize the worst case impact of the poisoning adversary. We denote by $\boldsymbol{S} = \boldsymbol{B}\boldsymbol{B}^\top$ the covariance matrix of the noise added.

Adversarial objective We assume that the poisoning adversary is interested in maximizing some adversarial objective $\ell_{\text{adv}}\left(\boldsymbol{\theta}\right)$ on target data:

$$\ell_{\text{adv}}\left(\boldsymbol{\theta}\right) = \mathop{\mathbb{E}}_{z \sim \mathbb{P}^{\text{target}}} \left[\ell\left(\boldsymbol{\theta}, \boldsymbol{z}\right)\right] \text{ Maximize loss on some target data}$$

Dynamics as a Markov Chain By (1), we have that, conditioned on $\boldsymbol{\theta}_t$ and $\boldsymbol{z}_t$, $\boldsymbol{\theta}_{t+1}$ follows a Gaussian distribution with mean $\boldsymbol{\theta}_t - \eta \nabla \ell\left(\boldsymbol{\theta}_t, \boldsymbol{z}_t\right)$.

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

$$\mathbb{P}_{t+1}\left(\boldsymbol{\theta}\right) = \int \mathbb{P}_{\boldsymbol{S}, \mathbb{P}^{\text{data}}, \boldsymbol{z}^{\text{adv}}}\left(\boldsymbol{\theta} | \boldsymbol{\theta}'\right) \mu_t\left(\boldsymbol{\theta}'\right) d\boldsymbol{\theta}',$$

where $\mathbb{P}_{\boldsymbol{S}, \mathbb{P}^{\text{data}}, \boldsymbol{z}^{\text{adv}}}$ is the transition kernel induced by (1), explicitly given by

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

where $\mathcal{N}\left(\boldsymbol{x} | \mu, \Sigma\right)$ denotes the Gaussian density at x for a Gaussian with mean μ and covariance matrix Σ.

A certificate for the adversarial loss (Analysis) Since this is a Markov process, the optimal sequence of actions for the adversary (ie choices of $\boldsymbol{z}^{\text{adv}}$) constitute a Markov Decision Process with

$$\text{States } \boldsymbol{\theta}, \text{Actions } \boldsymbol{z}^{\text{adv}}, \text{Transition Kernel } \mathbb{P}^{\text{trans}}\left(\boldsymbol{\theta}' | \boldsymbol{\theta}, \boldsymbol{z}^{\text{adv}}\right) = \mathbb{P}_{\boldsymbol{S}, \mathbb{P}^{\text{data}}, \boldsymbol{z}^{\text{adv}}}\left(\boldsymbol{\theta}' | \boldsymbol{\theta}\right)$$

and hence, can be formulated as an infinite dimensional linear program [8]. In particular, for the infinite horizon average reward setting [6], the LP can be written as

$$\sup_{\mathbb{P}\in\mathcal{P}[\mathbb{R}^d\times\mathbb{R}^n]} \quad \mathop{\mathbb{E}}_{\boldsymbol{\theta},\boldsymbol{z}^{\mathrm{adv}}\sim\mathbb{P}}\left[\ell_{\mathrm{adv}}\left(\boldsymbol{\theta}\right)\right] \tag{3a}$$

$$\text{subject to} \quad \mathop{\mathbb{E}}_{\boldsymbol{\theta},\boldsymbol{z}^{\mathrm{adv}}\sim\mathbb{P}}\left[\mathbb{P}_{\boldsymbol{S},\mathbb{P}^{\mathrm{data}},\boldsymbol{z}^{\mathrm{adv}}}\left(\boldsymbol{\theta}'|\boldsymbol{\theta}\right)\right] \quad = \mathop{\mathbb{E}}_{\boldsymbol{\theta},\boldsymbol{z}^{\mathrm{adv}}\sim\mathbb{P}}\left[\mathbb{I}[\boldsymbol{\theta}'=\boldsymbol{\theta}]\right] \quad \forall\boldsymbol{\theta}'\in\mathbb{R}^d. \tag{3b}$$

where $\mathcal{P}[\mathbb{R}^d\times\mathbb{R}^n]$ denotes the space of probability measures on $\mathbb{R}^d\times\mathbb{R}^n$ and $\mathbb{I}$ denotes the indicator function that equals 1 if its argument is true and 0 otherwise.

Theorem 1. *For any function $\lambda:\mathbb{R}^d\mapsto\mathbb{R}$, we can upper bound the optimal value of (3) by*

$$\sup_{\substack{\boldsymbol{\theta}\in\mathbb{R}^d\\ \boldsymbol{z}^{adv}\in\mathcal{A}}} \quad \mathop{\mathbb{E}}_{\boldsymbol{\theta}'\sim\mathbb{P}_{\boldsymbol{S},\mathbb{P}^{\mathrm{data}},\boldsymbol{z}^{adv}}(\cdot|\boldsymbol{\theta})}\left[\lambda\left(\boldsymbol{\theta}'\right)\right]+\ell_{adv}\left(\boldsymbol{\theta}\right)-\lambda\left(\boldsymbol{\theta}\right). \tag{4}$$

Proof. Follows by weak duality for the LP (5a).

If strong duality holds, we further have that the optimal value of (3) is exactly equal to

$$\inf_{\lambda:\mathbb{R}^d\mapsto\mathbb{R}} \quad \sup_{\boldsymbol{\theta},\boldsymbol{z}^{\mathrm{adv}}} \quad \mathop{\mathbb{E}}_{\boldsymbol{\theta}'\sim\mathbb{P}_{\boldsymbol{S},\mathbb{P}^{\mathrm{data}},\boldsymbol{z}^{\mathrm{adv}}}(\cdot|\boldsymbol{\theta})}\left[\lambda\left(\boldsymbol{\theta}'\right)\right]+\ell_{\mathrm{adv}}\left(\boldsymbol{\theta}\right)-\lambda\left(\boldsymbol{\theta}\right). \tag{5a}$$

A design principle for robust learning algorithms (aka meta-learning a robust learning algorithm) Based on the above analysis, we can attempt to design the parameters of the learning algorithm (in this case $\boldsymbol{S}=\boldsymbol{B}\boldsymbol{B}^\top$) to trade-off performance and robustness. In particular, in the absence of poisoned data, the updates (1) result in a stationary distribution $\mathbb{P}\left(\boldsymbol{S},\mathbb{P}^{\mathrm{data}}\right)$ over model parameters $\boldsymbol{\theta}$:

$$\mathbb{P}\left(\boldsymbol{S},\mathbb{P}^{\mathrm{data}}\right)=\mathbb{P} \text{ that satisfies } \mathbb{P}\left(\boldsymbol{\theta}'\right)=\mathop{\mathbb{E}}_{\boldsymbol{\theta}\sim\mathbb{P}}\left[\mathop{\mathbb{E}}_{\boldsymbol{z}\sim\mathbb{P}^{\mathrm{data}}}\left[\mathcal{N}\left(\boldsymbol{\theta}'|\boldsymbol{\theta}_t-\eta\nabla\ell\left(\boldsymbol{\theta},\boldsymbol{z}\right),\eta^2\boldsymbol{S}\right)\right]\right]. \tag{6}$$

Given some space of data distributions $\mathcal{P}$ we can sample from (in a meta learning sense), we can propose the following criterion:

$$\inf_{\substack{\boldsymbol{S}\in\mathbb{S}^d_+\\ \lambda:\mathbb{R}^d\mapsto\mathbb{R}}} \mathop{\mathbb{E}}_{\mathbb{P}^{\mathrm{data}}\sim\mathcal{P}}\left[\mathop{\mathbb{E}}_{\boldsymbol{\theta}\sim\mathbb{P}\left(\boldsymbol{S},\mathbb{P}^{\mathrm{data}}\right)}\left[\ell\left(\boldsymbol{\theta}\right)\right]+\kappa\left(\sup_{\boldsymbol{\theta}\in\mathbb{R}^d,\boldsymbol{z}^{\mathrm{adv}}\in\mathcal{A}}\mathop{\mathbb{E}}_{\boldsymbol{\theta}'\sim\mathbb{P}_{\boldsymbol{S},\mathbb{P}^{\mathrm{data}},\boldsymbol{z}^{\mathrm{adv}}}(\cdot|\boldsymbol{\theta})}\left[\lambda\left(\boldsymbol{\theta}'\right)\right]+\ell_{\mathrm{adv}}\left(\boldsymbol{\theta}\right)-\lambda\left(\boldsymbol{\theta}\right)\right)\right], \tag{7}$$

where $\kappa>0$ is a trade-off parameter. The outer expectation is a meta-learning inspired formulation, where we are designing a learning algorithm that is good "in expectation" under a meta-distribution over distributions. The first term in the outer expectation 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 worse case loss incurred by the learning algorithm in the presence of the adversary.

2 Mean estimation

Consider the mean estimation problem, where we aim to learn the parameter $\boldsymbol{\theta}$ to estimate the mean $\mu=\mathbb{E}_{\boldsymbol{z}\sim\mathbb{P}^{\mathrm{data}}}\left[\boldsymbol{z}\right]$ of a distribution $\mathbb{P}^{\mathrm{data}}$. The adversarial loss is given by:

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

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 $\|\boldsymbol{z}^{adv}-\mu\|_2^2\leq r$, the certificate for the mean estimation problem for $\mathbb{P}^{\mathrm{data}}(\boldsymbol{z})=\mathcal{N}(\boldsymbol{z}|\mu,\boldsymbol{\Sigma})$ for a fixed learning algorithm (i.e. $\boldsymbol{S}$ is fixed) is given by:*

$$\inf_{\boldsymbol{A}\in\mathbb{S}^d,\boldsymbol{b}\in\mathbb{R}^d,\nu\geq 0} g(\boldsymbol{A},\boldsymbol{b},\nu,\boldsymbol{S},\mu,\boldsymbol{\Sigma}), \tag{8}$$

Algorithm 1 Meta learning

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

where $g(\boldsymbol{A}, \boldsymbol{b}, \nu, \boldsymbol{S}, \mu, \boldsymbol{\Sigma})$ *is a convex objective in* $\boldsymbol{A}, \boldsymbol{b}, \nu$ *(matrix fractional objective with Linear Matrix Inequality (LMI) constraint) as defined below:*

$$g(\boldsymbol{A}, \boldsymbol{b}, \nu, \boldsymbol{S}, \mu, \boldsymbol{\Sigma}) = \begin{cases} \frac{1}{4} \left\| \begin{bmatrix} 2(1-\epsilon)\eta(1-\eta)\boldsymbol{A}\mu - 2\mu - \eta\boldsymbol{b} \\ \epsilon\eta\boldsymbol{b} + 2\nu\mu \end{bmatrix} \right\|_{\boldsymbol{D}}^2 + (1-\epsilon)(\eta^2 \mathit{Trace}(\boldsymbol{\Sigma}\boldsymbol{A}) + \eta^2\mu^\top \boldsymbol{A}\mu + \eta\boldsymbol{b}^\top\mu) \\ +\mu^\top\mu + \eta^2 \mathit{Trace}(\boldsymbol{A}\boldsymbol{S}) + \nu(r - \mu^\top\mu) \;\; \mathit{if} \;\; \nu \geq 0;\; \boldsymbol{D} \succeq 0 \\ -\infty \quad \mathit{else} \end{cases} \tag{9}$$

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

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 $\mathbb{P}^{\mathrm{data}}$ from the meta distribution $\mathcal{P}$). For the mean estimation problem this boils down to solving:

$$\inf_{\boldsymbol{S}\in\mathbb{S}_+^d} \quad \eta^2 \mathrm{Trace}(\boldsymbol{S}) + \kappa\mathbb{E}_{\mu,\boldsymbol{\Sigma}\sim\mathcal{P}}[\inf_{\substack{\nu\geq 0 \\ \boldsymbol{A}\in\mathbb{S}^d, \boldsymbol{b}\in\mathbb{R}^d}} g(\boldsymbol{A}, \boldsymbol{b}, \nu, \boldsymbol{S}, \mu, \boldsymbol{\Sigma})]. \tag{10}$$

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.

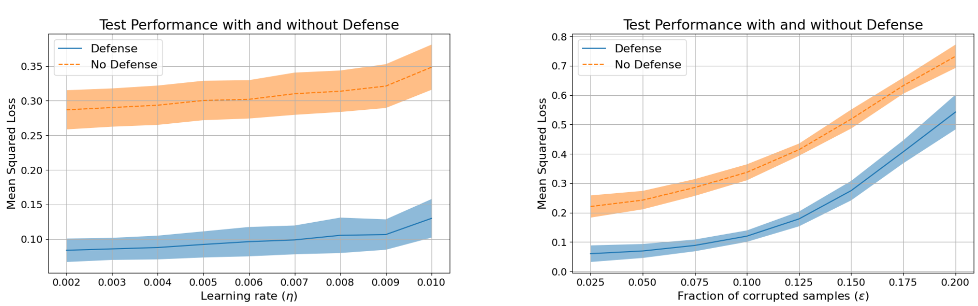

Figure 1: Test performance (mean squared error between true and estimated means) on 50 ($d = 20$ dimensional) Gaussian distributions drawn from Gaussian prior for the mean and Inverse Wishart prior for the covariance. The defense parameter $\boldsymbol{S}$ was trained with 10 such randomly chosen Gaussians via Algorithm 1. We varied the learning rates (left) and the the fraction of samples corrupted by the dynamic adversary (right) and observe that our defense beats training without defense significantly.

References

[1] Raef Bassily, Adam Smith, and Abhradeep Thakurta. Private empirical risk minimization: Efficient algorithms and tight error bounds. In *2014 IEEE 55th annual symposium on foundations of computer science*, pages 464–473. IEEE, 2014.

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

[3] 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.

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

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

[6] 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.

[7] 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.

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

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

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

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

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

[13] 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.

A Proofs

Theorem 2. *Choosing $\lambda : \mathbb{R}^d \to \mathbb{R}$ in Theorem 1 to be quadratic, i.e. $\lambda(\boldsymbol{\theta}) = \boldsymbol{\theta}^\top \boldsymbol{A}\boldsymbol{\theta} + \boldsymbol{\theta}^\top \boldsymbol{b}$, the adversarial constraint set of the form $\|\boldsymbol{z}^{adv} - \mu\|_2^2 \le r$, 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:*

$$\inf_{\boldsymbol{A}\in\mathbb{S}^d, \boldsymbol{b}\in\mathbb{R}^d, \nu\ge 0} g(\boldsymbol{A}, \boldsymbol{b}, \nu, \boldsymbol{S}, \mu, \boldsymbol{\Sigma}), \tag{8}$$

where $g(\boldsymbol{A}, \boldsymbol{b}, \nu, \boldsymbol{S}, \mu, \boldsymbol{\Sigma})$ is a convex objective in $\boldsymbol{A}, \boldsymbol{b}, \nu$ (matrix fractional objective with Linear Matrix Inequality (LMI) constraint) as defined below:

$$g(\boldsymbol{A}, \boldsymbol{b}, \nu, \boldsymbol{S}, \mu, \boldsymbol{\Sigma}) = \begin{cases} \frac{1}{4}\left\| \begin{bmatrix} 2(1-\epsilon)\eta(1-\eta)\boldsymbol{A}\mu - 2\mu - \eta\boldsymbol{b} \\ \epsilon\eta\boldsymbol{b} + 2\nu\mu \end{bmatrix} \right\|_{\boldsymbol{D}}^2 + (1-\epsilon)(\eta^2 Trace(\boldsymbol{\Sigma}\boldsymbol{A}) + \eta^2\mu^\top\boldsymbol{A}\mu + \eta\boldsymbol{b}^\top\mu) \\ +\mu^\top\mu + \eta^2 Trace(\boldsymbol{A}\boldsymbol{S}) + \nu(r - \mu^\top\mu) \text{ if } \quad \nu \ge 0;\ \boldsymbol{D} \succeq 0 \\ -\infty \quad else \end{cases} \tag{9}$$

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

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

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

The transition distribution for the parameter is given by:

$$\mathbb{P}_{\boldsymbol{S},\mathbb{P}^{\text{data}},\boldsymbol{z}^{\text{adv}}}(\boldsymbol{\theta}'|\boldsymbol{\theta}) = \epsilon\mathcal{N}\left(\boldsymbol{\theta}'|F(\boldsymbol{\theta}, \boldsymbol{z}^{\text{adv}}), \eta^2\boldsymbol{S}\right) + (1-\epsilon)\mathop{\mathbb{E}}_{\boldsymbol{z}\sim\mathbb{P}^{\text{data}}}\left[\mathcal{N}\left(\boldsymbol{\theta}'|F(\boldsymbol{\theta}, \boldsymbol{z}), \eta^2\boldsymbol{S}\right)\right] \tag{11}$$

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

Then, we have from Eq. (5a) that the certified bound on the adversarial objective is given by:

$$\sup_{\substack{z^{\mathrm{adv}}\in\mathcal{A}\\ \boldsymbol{\theta}}} \epsilon \mathop{\mathbb{E}}_{\boldsymbol{\theta}'\sim\mathcal{N}\left(F\left(\boldsymbol{\theta},z^{\mathrm{adv}}\right),\eta^2 \boldsymbol{S}\right)}\left[\lambda\left(\boldsymbol{\theta}'\right)\right] + (1-\epsilon)\mathop{\mathbb{E}}_{z\sim\mathbb{P}^{\mathrm{data}}}\left[\mathop{\mathbb{E}}_{\boldsymbol{\theta}'\sim\mathcal{N}(F(\boldsymbol{\theta},z),\eta^2\boldsymbol{S})}\left[\lambda\left(\boldsymbol{\theta}'\right)\right]\right] - \lambda\left(\boldsymbol{\theta}\right) + \ell_{\mathrm{adv}}\left(\boldsymbol{\theta}\right) \tag{12a}$$

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

$$= \sup_{\substack{z^{\mathrm{adv}}\in\mathcal{A}\\ \boldsymbol{\theta}}} \epsilon\left(\lambda\left(F\left(\boldsymbol{\theta},z^{\mathrm{adv}}\right)\right) + \eta^2\left\langle\nabla^2\lambda\left(0\right),\boldsymbol{S}\right\rangle\right) + (1-\epsilon)\mathop{\mathbb{E}}_{z\sim\mathbb{P}^{\mathrm{data}}}\left[\lambda\left(F\left(\boldsymbol{\theta},z\right)\right) + \eta^2\left\langle\nabla^2\lambda\left(0\right),\boldsymbol{S}\right\rangle\right] - \lambda\left(\boldsymbol{\theta}\right) + \ell_{\mathrm{adv}}\left(\boldsymbol{\theta}\right) \tag{12b}$$

$$= \sup_{\substack{z^{\mathrm{adv}}:\|z^{\mathrm{adv}}-\mu\|_2^2\leq r\\ \boldsymbol{\theta}}} -\left\|\begin{bmatrix}\boldsymbol{\theta}\\ z^{\mathrm{adv}}\end{bmatrix}\right\|_{\boldsymbol{E}^{-1}}^2 + \begin{bmatrix}\boldsymbol{\theta}\\ z^{\mathrm{adv}}\end{bmatrix}^\top\begin{bmatrix}2(1-\epsilon)\eta(1-\eta)\boldsymbol{A}\mu - 2\mu - \eta\boldsymbol{b}\\ \epsilon\eta\boldsymbol{b}\end{bmatrix}$$
$$+ (1-\epsilon)(\eta^2\mathrm{Trace}(\boldsymbol{\Sigma}\boldsymbol{A}) + \eta^2\mu^\top\boldsymbol{A}\mu + \eta\boldsymbol{b}^\top\mu) + \mu^\top\mu, \tag{12c}$$

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:

$$= \inf_{\nu\geq 0}\sup_{\substack{z^{\mathrm{adv}}\\ \boldsymbol{\theta}}} -\left\|\begin{bmatrix}\boldsymbol{\theta}\\ z^{\mathrm{adv}}\end{bmatrix}\right\|_{\boldsymbol{D}^{-1}}^2 + \begin{bmatrix}\boldsymbol{\theta}\\ z^{\mathrm{adv}}\end{bmatrix}^\top\begin{bmatrix}2(1-\epsilon)\eta(1-\eta)\boldsymbol{A}\mu - 2\mu - \eta\boldsymbol{b}\\ \epsilon\eta\boldsymbol{b}\end{bmatrix}$$
$$+ (1-\epsilon)(\eta^2\mathrm{Trace}(\boldsymbol{\Sigma}\boldsymbol{A}) + \eta^2\mu^\top\boldsymbol{A}\mu + \eta\boldsymbol{b}^\top\mu) + \mu^\top\mu + \nu(r - \mu^\top\mu) \tag{12d}$$

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

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

$$\inf_{\nu\geq 0}\frac{1}{4}\left\|\begin{bmatrix}2(1-\epsilon)\eta(1-\eta)\boldsymbol{A}\mu - 2\mu - \eta\boldsymbol{b}\\ \epsilon\eta\boldsymbol{b} + 2\nu\mu\end{bmatrix}\right\|_{\boldsymbol{D}}^2 + (1-\epsilon)(\eta^2\mathrm{Trace}(\boldsymbol{\Sigma}\boldsymbol{A}) + \eta^2\mu^\top\boldsymbol{A}\mu + \eta\boldsymbol{b}^\top\mu)$$
$$+ \mu^\top\mu + \eta^2\mathrm{Trace}(\boldsymbol{A}\boldsymbol{S}) + \nu(r - \mu^\top\mu) \text{ such that } \boldsymbol{D}\succeq 0. \tag{12e}$$

This completes the proof.

□

Lemma A.1. *The stationary distribution in the absence of adversary in Eq.* (6) *for the mean estimation problem for* $\mathbb{P}^{\mathrm{data}} = \mathcal{N}(\mu,\boldsymbol{\Sigma})$ *takes the form:*

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

Proof. The stationary distribution is tractable in this case. Recall from Eq. (11), setting $\epsilon = 0$, the transition distribution conditioned on $\boldsymbol{\theta}$ is a Gaussian whose mean is linear in $\boldsymbol{\theta}$. Therefore the stationary distribution:

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

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:

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

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. $\square$

Lemma A.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:*

$$\mathop{\mathbb{E}}_{\boldsymbol{\theta}\sim\mathbb{P}\left(\boldsymbol{S},\mathbb{P}^{\text{data}}\right)}\left[\ell\left(\boldsymbol{\theta}\right)\right] = \eta^2 \mathit{Trace}(\boldsymbol{S}). \tag{13}$$

Proof.

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

$\square$

Remark A.1. *We use CVXPY [5] to solve the optimization problems in Algorithm 1.*