# Certifying robustness to adaptive data poisoning

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

**Madeleine Udell**
Stanford University
udell@stanford.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 that 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 problem and outline directions for extending this in further work.

## 1 Introduction & Problem Formulation

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) [4, 11], 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 [1]. 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 [15, 21]. Even in [17] 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 [20, 18] 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?

Foundations of RL and Control Connections and New Perspectives Workshop at International Conference on Machine Learning (ICML 2024).

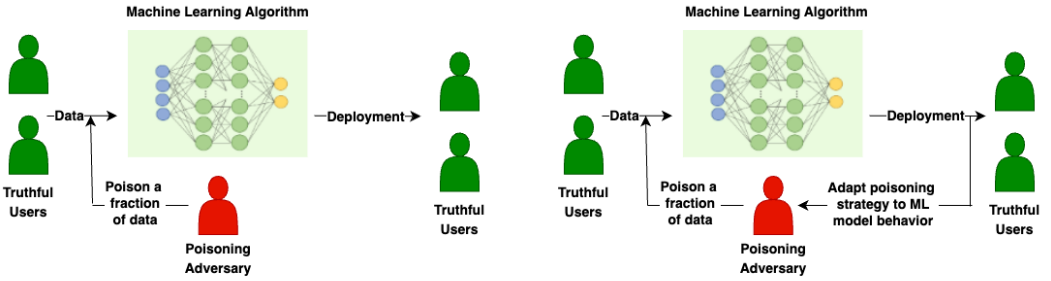


Figure 1: A schematic diagram to highlight the differences between static and dynamic poisoning.

| Attack Type | Adversary adapts poisoning strategy upon observing model behavior | Adversary can poison data for deployed model | Certified robustness |
|---|---|---|---|
| Static / One-shot ([16, 15, 13]) | ✗ | ✗ | ✓ |
| Backdoor([3, 7, 8, 22, 21, 14]) | ✗ | ✓ | ✓ |
| Dynamic attack only ([18, 20] | ✓ | ✓ | ✗ |
| Dynamic attack & defense (Ours) | ✓ | ✓ | ✓ |

Table 1: A comparison with lines of work closest to ours. A detailed description is provided in Appendix A.

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 3) and suggest directions of future work to apply the framework to more realistic learning scenarios.

## 2 Problem setup

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

**Online learning algorithm** We consider learning algorithms that operate online receiving at each step a new datapoint and making an update to parameters being estimated. In

| Notation | Interpretation | Belongs to |
|---|---|---|
| $\theta$ | Parameters of model | $\boldsymbol{\Theta}$ |
| $\phi$ | Hyper-parameters of learning algorithm | $\boldsymbol{\Phi}$ |
| $w$ | Gaussian noise injected into learning algorithm | $\boldsymbol{W}$ |
| $z$ | Datapoint | $\boldsymbol{Z}$ |
| $F$ | Update rule of learning algorithm | $\boldsymbol{\Theta} \times \boldsymbol{W} \times \boldsymbol{Z} \mapsto \boldsymbol{\Theta}$ |
| $z^{\text{adv}}$ | Adversarial data point | $\mathcal{A}$ |

Table 2: Notation

particular, we consider learning algorithms that can be written as

$$\boldsymbol{\theta}_{t+1} \leftarrow F\left(\underbrace{\boldsymbol{\theta}_t}_{\text{Parameter estimate at time } t}, \underbrace{\boldsymbol{w}_t}_{\text{Exogeneous noise input}}, \underbrace{\boldsymbol{z}_t}_{\text{Datapoint received at time } t}\right) \tag{1}$$

where $F : \boldsymbol{\Theta} \times \boldsymbol{W} \times \boldsymbol{Z} \mapsto \boldsymbol{\Theta}$ is an update function that maps the parameters at time $t$ to new parameters, given an exogeneous noise input $\boldsymbol{w}_t$ and a datapoint $\boldsymbol{z}_t$. The exogenous noise input refers to noise artificially injected into the training algorithm in order to make the algorithm more robust to potential poisoning. We further assume that the distribution of $\boldsymbol{w}_t$ is independent of $t$ and each $\boldsymbol{w}_t$ is sampled iid.

**Poisoned learning algorithm** We work in a setting where some of the datapoints received by the learning algorithm are corrupted by an adversary, with the corruption allowed to be a function of the entire trajectory of the learning algorithm up to that point. 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.

Mathematically, we have that at time $t$ the learning algorithm receives a datapoint $\boldsymbol{z}_t \sim \epsilon\text{Dirac}(\boldsymbol{z}_t^{\text{adv}}) + (1-\epsilon)\mathbb{P}^{\text{data}}$ and $\epsilon$ is a parameter that controls the "level" of poisoning (analogous to the fraction of poisoned samples in static poisoning settings [15]). This is a special case of Huber's contamination model, which is used in the robust statistics literature [5] (with the contamination model being a Dirac distribution). We restrict the adversary to choose $\boldsymbol{z}_t^{\text{adv}} \in \mathcal{A}$ which reflects the allowed range of datapoints due to input feature normalization or outlier detection systems.

**Adversarial objective** 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.

**Dynamics as a Markov Chain** 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

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

where $\mathbb{P}_{F,\mathbb{P}^{\text{data}},\boldsymbol{z}^{\text{adv}}}$ is the transition kernel induced, i.e., the conditional probability distribution of $\boldsymbol{\theta}' = F(\boldsymbol{\theta}, \boldsymbol{w}_t, \boldsymbol{z}_t)$ by (1) given $\boldsymbol{\theta}$.

### 2.1 Technical Approach: Certificate of Robustness

We are now ready to present our technical result, a certificate of robustness against dynamic data poisoning adversaries. Since the learning algorithm is a Markov process, the optimal sequence of actions for the adversary (i.e., choices of $\boldsymbol{z}^{\text{adv}}$) constitute a Markov Decision Process with

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

and hence, can be formulated as an *infinite dimensional* linear program [12]. In particular, for the infinite horizon average reward setting [9], the LP can be written as

$$\sup_{\mathbb{P}\in\mathcal{P}[\Theta\times\boldsymbol{Z}]} \mathop{\mathbb{E}}_{\boldsymbol{\theta},\boldsymbol{z}^{\text{adv}}\sim\mathbb{P}} [\ell_{\text{adv}}(\boldsymbol{\theta})] \tag{2a}$$

$$\text{subject to } \mathop{\mathbb{E}}_{\boldsymbol{\theta},\boldsymbol{z}^{\text{adv}}\sim\mathbb{P}} \left[\mathbb{P}_{F,\mathbb{P}^{\text{data}},\boldsymbol{z}^{\text{adv}}}(\boldsymbol{\theta}'|\boldsymbol{\theta})\right] = \mathop{\mathbb{E}}_{\boldsymbol{\theta},\boldsymbol{z}^{\text{adv}}\sim\mathbb{P}} [\mathbb{I}[\boldsymbol{\theta}' = \boldsymbol{\theta}]] \quad \forall \boldsymbol{\theta}' \in \mathbb{R}^d. \tag{2b}$$

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

**Theorem 1.** *For any function $\lambda : \Theta \mapsto \mathbb{R}$, we can upper bound the optimal value of* (2) *by*

$$\sup_{\substack{\boldsymbol{\theta}\in\boldsymbol{\Theta}\\ \boldsymbol{z}^{adv}\in\mathcal{A}}} \quad \mathbb{E}_{\boldsymbol{\theta}'\sim\mathbb{P}_{F,\mathbb{P}^{\text{data}},\boldsymbol{z}^{adv}}(\cdot|\boldsymbol{\theta})} \left[\lambda\left(\boldsymbol{\theta}'\right)\right] + \ell_{adv}\left(\boldsymbol{\theta}\right) - \lambda\left(\boldsymbol{\theta}\right). \tag{3}$$

*Proof.* Follows by weak duality for the LP (4a).

If strong duality holds, we further have that the optimal value of (2) is exactly equal to

$$\inf_{\lambda:\boldsymbol{\Theta}\mapsto\mathbb{R}} \quad \sup_{\boldsymbol{\theta}\in\boldsymbol{\Theta},\boldsymbol{z}^{\text{adv}}\in\mathcal{A}} \quad \mathbb{E}_{\boldsymbol{\theta}'\sim\mathbb{P}_{F,\mathbb{P}^{\text{data}},\boldsymbol{z}^{\text{adv}}}(\cdot|\boldsymbol{\theta})} \left[\lambda\left(\boldsymbol{\theta}'\right)\right] + \ell_{\text{adv}}\left(\boldsymbol{\theta}\right) - \lambda\left(\boldsymbol{\theta}\right). \tag{4a}$$

### 2.2 Meta-learning a robust learning algorithm

Consider a parameterized family of learning algorithms $F_\phi$ with tunable parameters $\phi \in \boldsymbol{\Phi}$. Based on the above analysis, we can attempt to design the parameters $\phi$ of the learning algorithm to trade-off performance and robustness. In particular, in the absence of poisoned data, assume that the updates (1) result in a stationary distribution $\mathbb{P}\left(\phi, \mathbb{P}^{\text{data}}\right)$ over model parameters $\boldsymbol{\theta}$.

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

$$\inf_{\substack{\phi\in\boldsymbol{\Phi}\\ \lambda:\boldsymbol{\Theta}\mapsto\mathbb{R}}} \mathbb{E}_{\mathbb{P}^{\text{data}}\sim\mathcal{P}} \left[ \mathbb{E}_{\boldsymbol{\theta}\sim\mathbb{P}\left(\phi,\mathbb{P}^{\text{data}}\right)} \left[\ell\left(\boldsymbol{\theta}\right)\right] + \kappa \left( \sup_{\boldsymbol{\theta}\in\boldsymbol{\Theta},\boldsymbol{z}^{\text{adv}}\in\mathcal{A}} \mathbb{E}_{\boldsymbol{\theta}'\sim\mathbb{P}_{S,\mathbb{P}^{\text{data}},\boldsymbol{z}^{\text{adv}}}(\cdot|\boldsymbol{\theta})} \left[\lambda\left(\boldsymbol{\theta}'\right)\right] + \ell_{\text{adv}}\left(\boldsymbol{\theta}\right) - \lambda\left(\boldsymbol{\theta}\right) \right) \right], \tag{5}$$

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.

## 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:

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

where $\boldsymbol{S} = \boldsymbol{B}\boldsymbol{B}^\top \in \mathbb{S}_+^d$ is the tunable defense parameter and $\boldsymbol{w}_t \sim \mathcal{N}(0, \boldsymbol{I})$ is Gaussian noise. The adversarial loss is given by:

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

**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 $\|\boldsymbol{z}^{adv} - \mu\|_2^2 \le r$, 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:*

$$\inf_{\boldsymbol{A}\in\mathbb{S}^d, \boldsymbol{b}\in\mathbb{R}^d, \nu\ge 0} g(\boldsymbol{A}, \boldsymbol{b}, \nu, \boldsymbol{S}, \mu, \boldsymbol{\Sigma}), \tag{6}$$

*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:*

$$g(\boldsymbol{A}, \boldsymbol{b}, \nu, \boldsymbol{S}, \mu, \boldsymbol{\Sigma}) = \begin{cases} \frac{1}{4}\left\| \begin{bmatrix} 2(1-\epsilon)\eta(1-\eta)\boldsymbol{A}\mu - 2\mu - \eta\boldsymbol{b} \\ \epsilon\eta\boldsymbol{b} + 2\nu\mu \end{bmatrix} \right\|_{\boldsymbol{D}}^2 + (1-\epsilon)(\eta^2 Trace(\boldsymbol{\Sigma}\boldsymbol{A}) + \eta^2\mu^\top \boldsymbol{A}\mu + \eta\boldsymbol{b}^\top\mu) \\ +\mu^\top\mu + \eta^2 Trace(\boldsymbol{A}\boldsymbol{S}) + \nu(r - \mu^\top\mu) \text{ if } \nu \ge 0;\ \boldsymbol{D} \succeq 0 \\ -\infty \quad else \end{cases} \tag{7}$$

**Algorithm 1** Meta learning a robust learning algorithm for mean estimation

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

*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* $\|\boldsymbol{x}\|^2_{\boldsymbol{D}} = \boldsymbol{x}^\top \boldsymbol{D}^{-1}\boldsymbol{x}$.

**Meta-Learning Algorithm** Following the formulation in Eq. (5), 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:

$$\inf_{\boldsymbol{S}\in\mathbb{S}^d_+} \quad \eta^2 \mathrm{Trace}(\boldsymbol{S}) + \kappa \mathbb{E}_{\mu,\mathbf{\Sigma}\sim\mathcal{P}}[\inf_{\substack{\nu\geq 0 \\ \boldsymbol{A}\in\mathbb{S}^d, \boldsymbol{b}\in\mathbb{R}^d}} g(\boldsymbol{A}, \boldsymbol{b}, \nu, \boldsymbol{S}, \mu, \mathbf{\Sigma})]. \tag{8}$$

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.

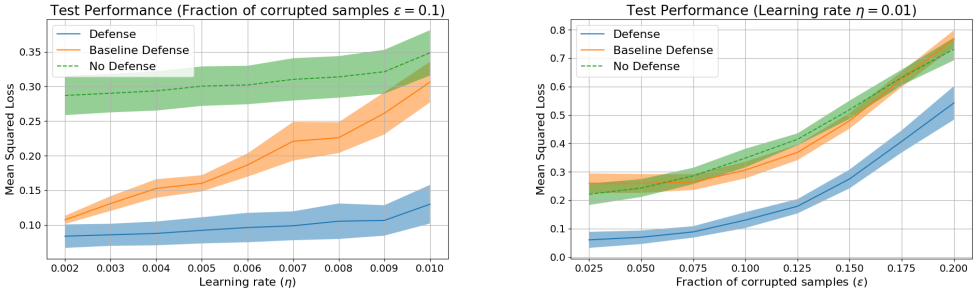


Figure 2: Test performance (mean squared error between true and estimated means) on 50 ($d = 20$ dimensional) Gaussian distributions drawn from Gaussian prior for the mean and Inverse Wishart prior for the covariance. The defense parameter $\boldsymbol{S}$ was trained with 10 such randomly chosen Gaussians 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] Battista Biggio, Blaine Nelson, and Pavel Laskov. Poisoning attacks against support vector machines. *arXiv preprint arXiv:1206.6389*, 2012.

[2] 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.

[3] 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.

[4] 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.

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

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

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

[8] 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.

[9] 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.

[10] 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.

[11] 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.

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

[13] 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.

[14] 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.

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

[16] 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.

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

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

[19] 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.

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

[21] 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.

[22] 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 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 [1, 10, 2] 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 [16] 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 [3, 7, 8, 22]. 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, a dynamic poisoning adversary [18, 20] 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. [15] provide certificates for linear models trained with gradient descent, [13] present a statistical upper-bound on the effectiveness of $\ell$-2 perturbations on training labels for linear models using randomized smoothing, [21, 14] present a model-agnostic certified approach that can effectively defend against both trigger-less and backdoor attacks, [19] observe that differential privacy, which usually covers addition or removal of data points, can also provide statistical guarantees in some limited poisoning settings. Even in [17] 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.

## B 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 $\|\boldsymbol{z}^{adv} - \mu\|_2^2 \leq r$, 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:*

$$\inf_{\boldsymbol{A}\in\mathbb{S}^d, \boldsymbol{b}\in\mathbb{R}^d, \nu\geq 0} g(\boldsymbol{A}, \boldsymbol{b}, \nu, \boldsymbol{S}, \mu, \boldsymbol{\Sigma}), \tag{6}$$

*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:*

$$g(\boldsymbol{A}, \boldsymbol{b}, \nu, \boldsymbol{S}, \mu, \boldsymbol{\Sigma}) = \begin{cases} \frac{1}{4}\left\| \begin{bmatrix} 2(1-\epsilon)\eta(1-\eta)\boldsymbol{A}\mu - 2\mu - \eta\boldsymbol{b} \\ \epsilon\eta\boldsymbol{b} + 2\nu\mu \end{bmatrix} \right\|_{\boldsymbol{D}}^2 + (1-\epsilon)(\eta^2 \mathit{Trace}(\boldsymbol{\Sigma}\boldsymbol{A}) + \eta^2\mu^\top\boldsymbol{A}\mu + \eta\boldsymbol{b}^\top\mu) \\ +\mu^\top\mu + \eta^2 \mathit{Trace}(\boldsymbol{A}\boldsymbol{S}) + \nu(r - \mu^\top\mu) \text{ if } \nu \geq 0;\ \boldsymbol{D} \succeq 0 \\ -\infty \quad \text{else} \end{cases} \tag{7}$$

*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:

$$\mathbb{P}_{\boldsymbol{S}, \mathbb{P}^{\text{data}}, \boldsymbol{z}^{\text{adv}}}\left(\boldsymbol{\theta}' | \boldsymbol{\theta}\right) = \epsilon \mathcal{N}\left(\boldsymbol{\theta}' | F(\boldsymbol{\theta}, \boldsymbol{z}^{\text{adv}}), \eta^2 \boldsymbol{S}\right) + (1-\epsilon) \mathop{\mathbb{E}}_{\boldsymbol{z} \sim \mathbb{P}^{\text{data}}}\left[\mathcal{N}\left(\boldsymbol{\theta}' | F(\boldsymbol{\theta}, \boldsymbol{z}), \eta^2 \boldsymbol{S}\right)\right] \quad (9)$$

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

Then, we have from Eq. (4a) that the certified bound on the adversarial objective is given by:

$$\sup_{\substack{\boldsymbol{z}^{\text{adv}} \in \mathcal{A} \\ \boldsymbol{\theta}}} \epsilon \mathop{\mathbb{E}}_{\boldsymbol{\theta}' \sim \mathcal{N}\left(F\left(\boldsymbol{\theta}, \boldsymbol{z}^{\text{adv}}\right), \eta^2 \boldsymbol{S}\right)}\left[\lambda\left(\boldsymbol{\theta}'\right)\right] + (1-\epsilon) \mathop{\mathbb{E}}_{\boldsymbol{z} \sim \mathbb{P}^{\text{data}}}\left[\mathop{\mathbb{E}}_{\boldsymbol{\theta}' \sim \mathcal{N}(F(\boldsymbol{\theta}, \boldsymbol{z}), \eta^2 \boldsymbol{S})}\left[\lambda\left(\boldsymbol{\theta}'\right)\right]\right] - \lambda(\boldsymbol{\theta}) + \ell_{\text{adv}}(\boldsymbol{\theta}) \quad (10a)$$

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:

$$= \sup_{\substack{\boldsymbol{z}^{\text{adv}} \in \mathcal{A} \\ \boldsymbol{\theta}}} \epsilon\left(\lambda\left(F\left(\boldsymbol{\theta}, \boldsymbol{z}^{\text{adv}}\right)\right) + \eta^2 \left\langle \nabla^2 \lambda(0), \boldsymbol{S} \right\rangle\right) + (1-\epsilon) \mathop{\mathbb{E}}_{\boldsymbol{z} \sim \mathbb{P}^{\text{data}}}\left[\lambda(F(\boldsymbol{\theta}, \boldsymbol{z})) + \eta^2 \left\langle \nabla^2 \lambda(0), \boldsymbol{S} \right\rangle\right] - \lambda(\boldsymbol{\theta}) + \ell_{\text{adv}}(\boldsymbol{\theta}) \quad (10b)$$

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

$$+ (1-\epsilon)(\eta^2 \text{Trace}(\boldsymbol{\Sigma}\boldsymbol{A}) + \eta^2 \mu^\top \boldsymbol{A} \mu + \eta \boldsymbol{b}^\top \mu) + \mu^\top \mu, \quad (10c)$$

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 $\nu$) can be written as:

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

$$+ (1-\epsilon)(\eta^2 \text{Trace}(\boldsymbol{\Sigma}\boldsymbol{A}) + \eta^2 \mu^\top \boldsymbol{A} \mu + \eta \boldsymbol{b}^\top \mu) + \mu^\top \mu + \nu(r - \mu^\top \mu) \quad (10d)$$

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:

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

$$+ \mu^\top \mu + \eta^2 \text{Trace}(\boldsymbol{A}\boldsymbol{S}) + \nu(r - \mu^\top \mu) \text{ such that } \boldsymbol{D} \succeq 0. \quad (10e)$$

This completes the proof.

□

**Lemma B.1.** *The stationary distribution in the absence of adversary in Eq.* (**??**) *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. (9), 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, $\mathbb{P}_{\boldsymbol{S},\mathbb{P}^{\text{data}}}\left(\boldsymbol{\theta}'|\boldsymbol{\theta}\right)$ 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:*

$$\mathop{\mathbb{E}}_{\boldsymbol{\theta}\sim\mathbb{P}\left(\boldsymbol{S},\mathbb{P}^{\text{data}}\right)}\left[\ell\left(\boldsymbol{\theta}\right)\right] = \eta^2 \mathit{Trace}(\boldsymbol{S}). \tag{11}$$

*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 [6] to solve the optimization problems in Algorithm 1.*