Towards Optimal Randomized Strategies in Adversarial Example Game

Jiahao Xie[1], **Chao Zhang**[*2], **Weijie Liu**[3,1], **Wensong Bai**[1,2], **Hui Qian**[1,4]

[1]College of Computer Science and Technology, Zhejiang University
[2]Advanced Technology Institute, Zhejiang University
[3]Qiushi Academy for Advanced Studies, Zhejiang University
[4]State Key Lab of CAD&CG, Zhejiang University
xiejh@zju.edu.cn, zczju@zju.edu.cn, westonhunter@zju.edu.cn, wensongb@zju.edu.cn, qianhui@zju.edu.cn

Abstract

The vulnerability of deep neural network models to adversarial example attacks is a practical challenge in many artificial intelligence applications. A recent line of work shows that the use of randomization in adversarial training is the key to find optimal strategies against adversarial example attacks. However, in a fully randomized setting where both the defender and the attacker can use randomized strategies, there are no efficient algorithm for finding such an optimal strategy. To fill the gap, we propose the first algorithm of its kind, called FRAT, which models the problem with a new infinite-dimensional continuous-time flow on probability distribution spaces. FRAT maintains a lightweight mixture of models for the defender, with flexibility to efficiently update mixing weights and model parameters at each iteration. Furthermore, FRAT utilizes lightweight sampling subroutines to construct a random strategy for the attacker. We prove that the continuous-time limit of FRAT converges to a mixed Nash equilibria in a zero-sum game formed by a defender and an attacker. Experimental results also demonstrate the efficiency of FRAT on CIFAR-10 and CIFAR-100 datasets.

Introduction

Deep Neural Network (DNN) models have been shown to be highly vulnerable to adversarial example attacks, which are tiny and imperceptible perturbations of the input designed to fool the model (Biggio et al. 2013; Szegedy et al. 2014; Goodfellow, Shlens, and Szegedy 2015). The vulnerability severely hindered the use of DNNs in safety-critical applications and became one of the main concerns of the artificial intelligence community (Goodfellow, Shlens, and Szegedy 2015). To improve the robustness of the model to adversarial examples, various defense strategies have been proposed in the past few years (Goodfellow, Shlens, and Szegedy 2015; Papernot et al. 2016; Samangouei, Kabkab, and Chellappa 2018; Madry et al. 2018; Cohen, Rosenfeld, and Kolter 2019; Moosavi-Dezfooli et al. 2019; Zhang et al. 2019; Pinot et al. 2020; Meunier et al. 2021; Gowal et al. 2021). Among existing strategies, the adversarial training (AT) approach (Goodfellow, Shlens, and Szegedy 2015; Madry et al. 2018) is widely recognized as the most successful one (Schott et al. 2018; Pang et al. 2020; Maini, Wong, and Kolter 2020; Bai et al. 2021), which usually constructs a virtual attacker that generates the worst adversarial examples maximizing the loss in the neighborhood of clean examples and seeks a robust classification model (classifier) by minimizing the loss on the generated examples.

Recent studies in the AT literature begin to explore randomized strategies for classifiers that probabilistically mix multiple classification models and show that stochastic classifiers are more robust to adversarial examples than a single deterministic classifier (Xie et al. 2018; Wang, Shi, and Osher 2019; Pinot et al. 2019, 2020; Meunier et al. 2021). Pinot et al. (2020) demonstrate from a game-theoretic perspective that randomized classifiers provide better worst-case theoretical guarantees than deterministic ones when attackers use deterministic strategies. They empirically show that the mixture of two classifiers obtained by their proposed Boosted Adversarial Training (BAT) algorithm achieves significant improvement over the state-of-the-art deterministic classifier produced by the Standard Adversarial Training (SAT) algorithm (Madry et al. 2018). Later, Meunier et al. (2021) show that randomized classifiers also outperform deterministic ones when attackers are allowed to use sophisticated randomized attack strategies.

In particular, Meunier et al. (2021) established the existence of an Mixed Nash Equilibrium (MNE) in randomized adversarial training games, i.e., there is an optimal randomized strategy pair of classifier and attacker such that neither of them can benefit from unilaterally changing its strategy, whereas when the player uses a deterministic strategy, a Nash equilibrium may not exist. For problems with discrete classifier parameter spaces, Meunier et al. (2021) propose two theoretically-guaranteed algorithms and then heuristically extend them to problems with continuous parameter. However, their heuristics are not guaranteed to find an MNE, and the efficiency may decrease as the number of mixture components in the randomized classifier increases (see the results in our experiments).

In this paper, we propose an efficient algorithm named Fully Randomized Adversarial Training (FRAT) for finding MNE in the randomized AT game with continuous classifier parameter spaces. In particular, FRAT maintains a weighted mixture of classification models for the classifier, where both model parameters and mixture weights are updated in each round with relatively low computational cost. Further-

*Corresponding author.

more, adversarial examples for the randomized attacker are generated by a restricted sampling subroutine called Projected Langevin algorithm (PLA), which has similar computations to Projected Gradient Descent (PGD) used in existing AT algorithms. Note that the actions of the classifier and the attacker in each round are actually obtained through a continuous-time stream of discrete game objective functions that converge to the MNE of a randomized AT game. Our main contributions are summarized as follows.

1. We propose a new hybrid continuous-time flow that appropriately exploits the bilinear problem structure of the randomized AT game. For the classifier, we adopt the Wasserstein-Fisher-Rao (WFR) flow of the objective because it leads to fast convergence of the objective in the probability space, and efficient update rules for model parameters and mixture weights can be achieved by discretizing this flow with the first order Euler scheme; For the attacker, we first construct a surrogate function by adding a regularizer to the bilinear game, since it is often difficult to construct a proper flow for the inner constrained maximization problem in the original game. Using this surrogate, a convergent Logit Best Response (LBR) flow can be derived, whose first order Euler discretization can be efficiently computed by PLA.
2. We develop analyses for the proposed hybrid continuous-time flow. With a proper regularization parameter, this flow is proved to converge at an $\mathcal{O}(1/T)$ rate to an MNE of the original unregularized AT game under mild assumptions, where T denotes the time.

We conduct numerical experiments on synthetic and real datasets to compare the performance of the proposed algorithm and existing ones. Experimental results demonstrate the efficiency of the proposed algorithm.

Related Works

Randomized strategies in adversarial training. Several works have investigated randomized strategies in adversarial training (Bulò et al. 2016; Perdomo and Singer 2019; Bose et al. 2020; Pinot et al. 2020; Meunier et al. 2021). Notably, Pinot et al. (2020) prove, from a game theoretical point of view, that randomized classifiers offer better worst-case theoretical guarantees than deterministic ones when the attacker uses deterministic strategies. Meunier et al. (2021) further show the existence of MNE in the adversarial example game when both the classifier and the attacker use randomized strategies. Existing methods for finding randomized strategies can be divided into two classes: (i) noise injection methods (Xie et al. 2018; Dhillon et al. 2018; Wang, Shi, and Osher 2019) and (ii) mixed strategy methods (Bulò et al. 2016; Perdomo and Singer 2019; Bose et al. 2020; Pinot et al. 2020; Meunier et al. 2021), both of which suffer from critical limitations. The first class of methods inject random noise into the input data or certain layers of the classification model, which is shown to be effective in practice but generally lacks theoretical guarantees. The second class of methods construct randomized strategies in probability spaces using game theory, and are usually theoretically-guaranteed. However, existing methods of this class apply to restricted settings where the randomized strategies are restricted in certain families of distributions (Bulò et al. 2016; Bose et al. 2020; Pinot et al. 2020), or the strategy spaces are discrete and finite (Perdomo and Singer 2019; Meunier et al. 2021).

Algorithms for finding MNE in zero-sum games. In recent years, there has been an increasing interest in finding mixed Nash equilibria in two-player zero-sum continuous games (Perkins and Leslie 2014; Hsieh, Liu, and Cevher 2019; Suggala and Netrapalli 2020; Domingo-Enrich et al. 2020; Liu et al. 2021; Ma and Ying 2021). However, these algorithms are impractical or infeasible for the adversarial example game. Specifically, the algorithms in (Hsieh, Liu, and Cevher 2019; Domingo-Enrich et al. 2020; Liu et al. 2021; Ma and Ying 2021) only apply to games on unconstrained spaces or manifolds, and do not apply to the adversarial example game in which the strategy space of the attacker is a compact convex constraint set. Although Perkins and Leslie (2014) and Suggala and Netrapalli (2020) develop algorithms that apply to zero-sum games with compact convex strategy spaces, their algorithms need store all historical strategies of both the players during the optimization process, which is prohibitive for the adversarial example game with medium to large-scale datasets.

Problem Setting and Algorithm

Problem Setting

The literature of adversarial training usually formulate the problem of adversarial example defense/attack as an Adversarial Training (AT) game, i.e., a zero-sum game between the classifier and the attacker (Shaham, Yamada, and Negahban 2018; Madry et al. 2018). Suppose that we are given a dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$, where $(\mathbf{x}_i, y_i) \in \mathcal{X} \times \mathcal{Y}$ denotes the feature-label pair of the i-th data sample, a classification model parameterized by $\theta \in \Theta$, and a loss function $\ell : \Theta \times (\mathcal{X} \times \mathcal{Y}) \to \mathbb{R}$. The attacker seeks strong adversarial examples by perturbing sample within a given distance ε to maximize the loss function ℓ, while the classifier aims to learn a model that minimizes the loss function defined on the generated adversarial data samples. Specifically, the deterministic AT game is given by

$$\inf_{\theta\in\Theta} \frac{1}{N}\sum_{i=1}^N \Big(\sup_{\hat{\mathbf{x}}_i \in \mathbb{B}_\varepsilon(\mathbf{x}_i)} \ell(\theta, (\hat{\mathbf{x}}_i, y_i)) \Big), \tag{1}$$

where $\mathbb{B}_\varepsilon(\mathbf{x}_i) := \{\mathbf{x} \in \mathcal{X} : d(\mathbf{x}, \mathbf{x}_i) \le \varepsilon\}$ and $d(\cdot,\cdot)$ denotes the distance function on $\mathcal{X}$. Instead of searching deterministic strategies as in (1), we consider a randomized setting of adversarial training, where the classifier (resp., the attacker) searches randomized strategies in the space $\mathcal{M}_1^+(\Theta)$ (resp., $\mathcal{M}_1^+(\mathbb{B}_\varepsilon(\mathbf{x}_i)), i \in \{1,\dots,N\}$). Here, $\mathcal{M}_1^+(\Theta)$ (resp., $\mathcal{M}_1^+(\mathbb{B}_\varepsilon(\mathbf{x}_i))$) denotes the Polish space of Borel probability measures on Θ (resp., $\mathbb{B}_\varepsilon(\mathbf{x}_i)$). Note that a randomized strategy of the attacker can be written as $\nu := (\nu_1, \dots, \nu_N) \in \Sigma$, where Σ stands for the product space $\mathcal{M}_1^+(\mathbb{B}_\varepsilon(\mathbf{x}_1)) \times \dots \times \mathcal{M}_1^+(\mathbb{B}_\varepsilon(\mathbf{x}_N))$. Then, the randomized AT game can be formulated as the following infinite-

dimensional minimax problem on the probability space

$$\inf_{\mu\in\mathcal{M}_1^+(\Theta)}\sup_{\nu\in\Sigma}\left\{\mathcal{L}(\mu,\nu):=\tfrac{1}{N}\textstyle\sum_{i=1}^N\mathbb{E}_{\theta\sim\mu,\mathbf{x}\sim\nu_i}[\ell(\theta,(\mathbf{x},y_i))]\right\} \tag{2}$$

Meunier et al. (2021) prove that under mild assumptions, there exists a mixed Nash equilibrium in (2), that is, there is a pair of strategy (μ^*,ν^*) such that, for any $\mu\in\mathcal{M}_1^+(\Theta)$ and $\nu\in\Sigma$, $\mathcal{L}(\mu^*,\nu)\le\mathcal{L}(\mu^*,\nu^*)\le\mathcal{L}(\mu,\nu^*)$. In contrast, the deterministic formulation (1) does not always have a pure Nash equilibrium (Pinot et al. 2020). This necessitates the use of randomized strategies for finding Nash equilibria (see (Meunier et al. 2021) for more discussions).

Entropy regularization. Instead of directly solving (2), we add an entropy regularization term to make the objective function strongly concave in ν. Note that this is a common technique in the infinite-dimensional optimization literature (Perkins and Leslie 2014; Domingo-Enrich et al. 2020; Ma and Ying 2021; Meunier et al. 2021). Specifically, we define the regularization function as $\mathcal{H}(\nu):=\frac{1}{N}\sum_{i=1}^N\mathrm{KL}(\nu_i\|u_i)$, where u_i is the uniform distribution over $\mathbb{B}_\varepsilon(\mathbf{x}_i)$ and $\mathrm{KL}(\cdot\|\cdot)$ denotes the Kullback-Leibler (KL) divergence, i.e., $\mathrm{KL}(\nu_i\|u_i)=\int\log(\frac{d\nu_i}{du_i})d\nu_i$, if ν_i is absolutely continuous w.r.t. u_i, otherwise $\mathrm{KL}(\nu_i\|u_i)=+\infty$. With this regularization function, we define the regularized adversarial example game as

$$\inf_{\mu\in\mathcal{M}_1^+(\Theta)}\sup_{\nu\in\Sigma}\left\{\mathcal{L}(\mu,\nu)-\beta\mathcal{H}(\nu)\right\}, \tag{3}$$

where $\beta>0$ is the regularization parameter.

The Proposed Algorithm

To solve (3), we propose an algorithm named Fully Randomized Adversarial Training (FRAT). FRAT is derived by discretizing a continuous-time flow on the probability spaces $\mathcal{M}_1^+(\Theta)$ and Σ. Constructing a continuous-time flow (defined by an Ordinary/Partial Differential Equation (ODE/PDE)) and then discretizing it to obtain an algorithm is a common routine for optimization on the probability space (Welling and Teh 2011; Liu 2017; Liutkus et al. 2019; Domingo-Enrich et al. 2020; Ma and Ying 2021). Note that even for this complicated infinite-dimensional space, it is feasible to analyze the convergence of a continuous-time flow using various ODE/PDE analysis tools. With a well-behaved flow, a practical and efficient discrete-time algorithm can be naturally derived using standard discretization techniques. This routine has also been widely used to obtain algorithms with good convergence properties for optimization on $\mathbb{R}^d$ (see (Su, Boyd, and Candes 2014) and references therein). In what follows, we first propose a hybrid continuous-time flow and then derive the FRAT algorithm.

The Continuous-time Flow. Here, we construct a hybrid continuous-time flow of $(\mu(t),\bar\nu(t))\in\mathcal{M}_1^+(\Theta)\times\Sigma$ that guarantees descent on the space $\mathcal{M}_1^+(\Theta)$ and ascent on Σ, where $\mu(t)$ and $\bar\nu(t)$ denote the strategies of the classifier and the attacker, respectively.

We let the strategy $\mu(t)$ of the classifier follow the Wasserstein-Fisher-Rao (WFR) flow

$$\dot\mu(t)=\gamma\nabla\cdot\left(\tfrac{\mu(t)}{N}\textstyle\sum_{i=1}^N\mathbb{E}_{\mathbf{x}\sim\nu_i(t)}[\nabla_\theta\ell(\theta,(\mathbf{x},y_i))]\right)$$
$$+\alpha\mu(t)\left(\mathcal{L}(\mu(t),\nu(t))-\mathcal{L}(\delta_\theta,\nu(t))\right), \tag{4}$$

with the initial condition $\mu(0)=\mu^0$ for some $\mu^0\in\mathcal{M}_1^+(\Theta)$, where γ and α are non-negative constants and $\nu(t)=(\nu_1(t),\dots,\nu_N(t))$ will be defined later. Actually, the WFR flow (4) is the gradient flow of the objective function $\mathcal{L}$ on the Wasserstein-Fisher-Rao space, and $\mathcal{L}$ descends following this flow when the strategy $\nu(t)$ is kept fixed. Note that the WFR flow has been widely used in optimization problems on the probability space, such as over-parameterized network training (Liero, Mielke, and Savaré 2018; Rotskoff et al. 2019; Chizat 2022) and unconstrained randomized zero-sum games (Domingo-Enrich et al. 2020).

The attacker uses the Logit Best Response (LBR) flow

$$\dot{\bar\nu}(t)=\tfrac{1}{t}\left(\nu(t)-\bar\nu(t)\right) \tag{5}$$

with the initial condition $\bar\nu(t)=\nu^0$ for $t\in[0,1]$. Here, $\nu(t)$ is the best response to the time-average strategy $\bar\mu(t):=\frac{1}{t}\int_0^t\mu(s)ds$ of the classifier for $t\ge1$, i.e.,

$$\nu(t):=\begin{cases}\nu^0, & \text{if } t\in[0,1)\\ \operatorname*{argmax}_{\nu\in\Sigma}\mathcal{L}(\bar\mu(t),\nu)-\beta\mathcal{H}(\nu), & \text{otherwise.}\end{cases} \tag{6}$$

The LBR flow has been widely used in constrained games, where the minimization/maximization sub-problem is defined on constrained sets (Hofbauer and Sandholm 2002; Perkins and Leslie 2014; Lahkar and Riedel 2015). As we shall see in the next section, (6) results in increasing of certain potential function and induces convergence to MNE when combined with (4). We call the hybrid flow of $(\mu(t),\bar\nu(t))$ following (4)-(6) as the WFR-LBR flow.

The Discrete-time Algorithm. To obtain a practical algorithm, we discretize the WFR-LBR flow in both space and time. The discretization steps are detailed as follows.

1. Discretization of the WFR flow (4). First, we use a weighted mixture $\hat\mu(t):=\sum_{j=1}^M w_j(t)\delta_{\theta_j(t)}\in\mathcal{M}_1^+(\Theta)$ to approximate $\mu(t)$ defined on the whole space, where M is a fixed integer, $\delta_{\theta_j(t)}$ is the Dirac measure of mass 1 at the particle $\theta_j(t)$, $w_j(t)\ge0$ is the mixing weight such that $\sum_{j=1}^M w_j(t)=1$. Then, we construct the following continuous-time flow for $\theta_j(t)$ and $w_j(t)$

$$\begin{cases}\dot\theta_j(t)=-\frac{\gamma}{N}\sum_{i=1}^N\mathbb{E}_{\mathbf{x}\sim\nu_i(t)}[\nabla_\theta\ell(\theta_j(t),(\mathbf{x},y_i))]\\ \dot w_j(t)=\alpha\big(\mathcal{L}(\mu(t),\nu(t))-\mathcal{L}(\delta_{\theta_j(t)},\nu(t))\big)\,w_j(t).\end{cases} \tag{7}$$

   Note that (4) is actually derived from (7) and the mean field limit of (7) converges to (4) (Domingo-Enrich et al. 2020). By applying the first order Euler discretization to (7), we obtain the following update rule

$$\begin{cases}\theta_j^{(t+1)}=\theta_j^{(t)}-\frac{\eta}{N}\sum_{i=1}^N\nabla_\theta\ell(\theta_j^{(t)},(\hat{\mathbf{x}}_i^{(t)},y_i))\\ w_j^{(t+1)}=\frac{w_j^{(t)}\exp(-\frac{\eta'}{N}\sum_{i=1}^N\ell(\theta_j^{(t)},(\hat{\mathbf{x}}_i^{(t)},y_i)))}{\sum_{j=1}^M w_j^{(t)}\exp(-\frac{\eta'}{N}\sum_{i=1}^N\ell(\theta_j^{(t)},(\hat{\mathbf{x}}_i^{(t)},y_i)))},\end{cases} \tag{8}$$

   where η and η' are positive step sizes and the superscript (t) denotes the discrete time step. This update rule moves each particle $\theta_j(t)$ along the negative gradient direction to decrease the loss value and adjusts the weights so that particles with lower loss values have larger weights.

2. Discretization of the LBR flow of (5). The first order Euler discretization of (5) leads to

$$\bar{\nu}_i^{(t+1)} \leftarrow \tfrac{t+1}{t+2}\bar{\nu}_i^{(t)} + \tfrac{1}{t+2}\nu_i^{(t+1)}, \quad (9)$$

As both $\bar{\nu}$ and ν_i are infinite-dimensional variables, it is generally hard to compute the above Euler discretization. It can be verified that the maximization problem in (6) has a unique solution $\nu(t)$, and each $\nu_i(t)$ has the density

$$p_{\nu_i}(\mathbf{x}) = \frac{\exp(\beta^{-1}\mathbb{E}_{\theta\sim\bar{\mu}(t)}[\ell(\theta,(\mathbf{x},y_i))])}{\int_{\mathbb{B}_\varepsilon(\mathbf{x}_i)}\exp(\beta^{-1}\mathbb{E}_{\theta\sim\bar{\mu}(t)}[\ell(\theta,(\mathbf{x},y_i))])d\mathbf{x}}. \quad (10)$$

Thus, we can use the stochastic approximation technique to approximate $\nu_i(t)$ by drawing a sample from $\nu_i^{(t+1)}$ (10), and obtain the following update rule

$$\bar{\nu}_i^{(t+1)} \leftarrow \tfrac{t+1}{t+2}\bar{\nu}_i^{(t)} + \tfrac{1}{t+2}\delta_{\hat{\mathbf{x}}_i^{(t+1)}}, \quad (11)$$

where the initial distribution $\bar{\nu}_i^{(0)} = \delta_{\hat{\mathbf{x}}_i^{(0)}}$ for some $\hat{\mathbf{x}}_i^{(0)} \in \mathbb{B}_\varepsilon(\mathbf{x}_i)$, and $\hat{\mathbf{x}}_i^{(t+1)}$ is sampled from (10). Note that the objective function in (6) is the sum of a linear function and a nonlinear regularization function. Thereby, the update rule of $\bar{\nu}$ can be viewed as the Generalized Frank-Wolfe (aka Generalized Conditional Gradient) algorithm (Bonesky et al. 2007; Bredies, Lorenz, and Maass 2009) on the probability measure space. In addition, the update rule of $\bar{\nu}$ is an extension of the stochastic fictitious play (Hofbauer and Sandholm 2002; Perkins and Leslie 2014) from the one-dimensional space to a high-dimensional space.

By combining the update rules (8) and (11), we obtain the FRAT algorithm, which is summarized in Algorithm 1. On line 1 of Algorithm 1, the strategies of the classifier and the attacker are initialized. Lines 3-6 computes the update direction of the classifier's strategy. Specifically, line 3 (resp., lines 4-5) computes model parameters (resp., weights) of the classifier's update direction, and line 6 combines the weights and models to obtain the update direction. Line 7 constructs the attacker's update direction by sampling from the distribution in (10). Finally, the two players update their strategies on lines 8 and 9, respectively.

To sample from the distribution $\nu_i^{(t)}$ which is supported on the constraint set $\mathbb{B}_\varepsilon(\mathbf{x}_i)$, we resort to an efficient constrained sampling method called the Projected Langevin Algorithm (PLA) (Bubeck, Eldan, and Lehec 2018). PLA produces an approximate sample by performing the following update step for multiple iterations

$$\hat{\mathbf{x}}_i \leftarrow \Pi_{\mathbb{B}_\varepsilon(\mathbf{x}_i)}(\hat{\mathbf{x}}_i + \tfrac{\lambda}{2\beta}\mathbb{E}_{\theta\sim\bar{\mu}^{(t)}}\nabla_{\mathbf{x}}\ell(\theta,(\hat{\mathbf{x}}_i,y_i)) + \omega\sqrt{\lambda}\xi), \quad (12)$$

where $\Pi_{\mathbb{B}_\varepsilon(\mathbf{x}_i)}(\cdot)$ denotes the projection onto $\mathbb{B}_\varepsilon(\mathbf{x}_i)$, λ is the step size, ω is a constant, and ξ is sampled in each iteration from $\mathcal{N}(\mathbf{0},\mathbf{I})$, i.e., the Gaussian distribution with mean $\mathbf{0}$ and covariance matrix $\mathbf{I}$. Actually, the PLA algorithm can be viewed as a variant of Projected Gradient Descent (PGD) with the only addition of Gaussian noise perturbations. Thus, the computational cost of PLA is similar to PGD which is used in existing AT algorithms.

In what follows, we discuss some tricks and techniques to speed up the proposed algorithm.

Algorithm 1: Fully Randomized Adversarial Training.

Input: IID samples $\theta_1^{(0)},\dots,\theta_M^{(0)}$ from $\mu^{(0)} \in \mathcal{M}_1^+(\Theta)$, an IID sample $\hat{\mathbf{x}}_i^{(0)}$ from $\nu_i^{(0)} \in \mathcal{M}_1^+(\mathbb{B}_\varepsilon(\mathbf{x}_i))$ for each $i \in \{1,\dots,N\}$, initial weights $w_j^{(0)} = 1/M$ for $j \in \{1,\dots,M\}$, and step sizes η and η'.

1: $\bar{\mu}^{(0)} \leftarrow \sum_{j=1}^M w_j^{(0)}\delta_{\theta_j^{(0)}}$, $\bar{\nu}_i^{(0)} \leftarrow \delta_{\hat{\mathbf{x}}_i^{(0)}}$ for each $i \in \{1,\dots,N\}$;
2: **for** $t = 0,\dots,T-1$ **do**
3: $\quad\theta_j^{(t+1)} \leftarrow \theta_j^{(t)} - \frac{\eta}{N}\sum_{i=1}^N \nabla_\theta \ell(\theta_j^{(t)},(\hat{\mathbf{x}}_i^{(t)},y_i))$ for each $j \in \{1,\dots,M\}$;
4: $\quad\hat{w}_j^{(t+1)} \leftarrow w_j^{(t)}\exp(-\frac{\eta'}{N}\sum_{i=1}^N \ell(\theta_j^{(t)},(\hat{\mathbf{x}}_i^{(t)},y_i)))$ for each $j \in \{1,\dots,M\}$;
5: $\quad w_j^{(t+1)} \leftarrow \hat{w}_j^{(t+1)}/\sum_{j=1}^M \hat{w}_j^{(t+1)}$ for each $j \in \{1,\dots,M\}$;
6: $\quad\mu^{(t+1)} \leftarrow \sum_{j=1}^M w_j^{(t+1)}\delta_{\theta_j^{(t+1)}}$;
7: $\quad$Sample $\hat{\mathbf{x}}_i^{(t+1)}$ from $\nu_i^{(t+1)}$ defined in (10) for each $i \in \{1,\dots,N\}$;
8: $\quad\bar{\mu}^{(t+1)} \leftarrow \frac{t+1}{t+2}\bar{\mu}^{(t)} + \frac{1}{t+2}\mu^{(t+1)}$;
9: $\quad\bar{\nu}_i^{(t+1)} \leftarrow \frac{t+1}{t+2}\bar{\nu}_i^{(t)} + \frac{1}{t+2}\delta_{\hat{\mathbf{x}}_i^{(t+1)}}$ for each $i \in \{1,\dots,N\}$;
10: **end for**
Return $\bar{\mu}^{(T)}$ and $\{\bar{\nu}_i^{(T)}\}_{i=1}^N$.

1. In practice, it is not necessary to sample $\{\hat{\mathbf{x}}_1^{(t)},\dots,\hat{\mathbf{x}}_N^{(t)}\}$ in step 7 and store them for calculating $\bar{\nu}^{(t)}$ in step 9 as in a defense task, the defender is only concerned with finding an optimal classifier, and thus the output strategy of the attacker is negligible. We only need to sample $\hat{\mathbf{x}}_i^{(t)}$'s in step 3 and 4 according to (10) with $\bar{\mu}^{(t)}$. This greatly reduces the storage cost.
2. The second technique is to use stochastic minibatch gradients to approximate the full gradient. Specifically, in the update steps of $\theta_j^{(t)}$ and $w_j^{(t)}$, we can sample a minibatch $\mathcal{B}_\mathbf{x}$ of the perturbed data samples from $\{\hat{\mathbf{x}}_1^{(t)},\dots,\hat{\mathbf{x}}_N^{(t)}\}$ to estimate the exact loss value $\frac{1}{N}\sum_{i=1}^N \ell(\theta_j^{(t)},(\hat{\mathbf{x}}_i^{(t)},y_i))$ and its gradient. In this way, we can perform the sampling subroutine on line 7 of Algorithm 1 for only a minibatch of data points in each iteration because the rest are unused.
3. In each step (12) of PLA, we can also randomly select a minibatch $\mathcal{B}_\mu$ from $\{\mu^{(0)},\dots,\mu^{(t)}\}$ to approximate the full gradient $\mathbb{E}_{\theta\sim\bar{\mu}^{(t)}}[\nabla\ell(\theta,(\mathbf{x},y_i))]$. Furthermore, to reduce the space complexity, we can maintain a sliding window $\{\mu^{(t-|\mathcal{B}_\mu|+1)},\dots,\mu^{(t)}\}$ and compute $\frac{1}{|\mathcal{B}_\mu|}\sum_{s=t-|\mathcal{B}_\mu|+1}^t \mathbb{E}_{\theta\sim\mu^{(s)}}[\nabla\ell(\theta,(\mathbf{x},y_i))]$ as a surrogate of $\mathbb{E}_{\theta\sim\bar{\mu}^{(t)}}[\nabla\ell(\theta,(\mathbf{x},y_i))]$.

By using the above techniques and an S-step PLA algorithm as the sampling subroutine, FRAT computes $(M|\mathcal{B}_\mathbf{x}| + MS|\mathcal{B}_\mu||\mathcal{B}_\mathbf{x}|)$ gradients in each iteration, where the computation over $\mathcal{B}_\mu$ can be parallelized. In comparison, the

SAT algorithm (Madry et al. 2018) with S steps of PGD attack in each iteration computes $(S+1)|\mathcal{B}_{\mathbf{x}}|$ gradients. Thus, when M and $|\mathcal{B}_\mu|$ are small, our algorithm has a similar per-iteration computation time to SAT.

Analysis

In this section, we show that the continuous-time flow (4) and (5) converges to an approximate MNE. Here, we only present the major results and defer the detailed analyses to the long version of this paper. Throughout our analysis, the following three assumptions are required.

Assumption 1. *The loss function $\ell : \Theta \times (\mathcal{X} \times \mathcal{Y}) \to \mathbb{R}$ satisfies the following conditions: (i) the function ℓ is non-negative and Borel measurable; (ii) $\ell(\theta, (\mathbf{x}, y))$ is continuous differentiable and G-Lipschitz w.r.t. $(\theta, \mathbf{x})$; (iii) $\exists U > 0$, $\forall \theta \in \Theta$ and $(\mathbf{x}, y) \in \mathcal{X} \times \mathcal{Y}$, $0 \le \ell(\theta, (\mathbf{x}, y)) \le U$.*

Assumption 2. *Θ is a compact Riemannian manifold without boundary of dimension d_θ embedded in $\mathbb{R}^{D_\theta}$. For all $\theta \in \Theta$, Vol$(\mathbb{B}_\varepsilon(\theta)) \ge e^{-K}\varepsilon^{d_\theta}$, where the volume is defined in terms of the Borel measure of Θ.*

Assumption 3. *The initial distribution μ^0 of the classifier has a Radon-Nikodym derivative $\rho := \frac{d\mu^0}{d\theta}$ with respect to the Borel measure of Θ and $\rho(\theta) \ge e^{-K'}$ for all $\theta \in \Theta$. Similarly, ν_i^0 also has a Radon-Nikodym derivative $q_i := \frac{d\nu_i^0}{d\mathbf{x}}$ with respect to the Lebesgue measure of $\mathbb{B}_\varepsilon(\mathbf{x}_i)$ for all $i \in \{1, \dots, N\}$ and $q_i(\mathbf{x}) > 0$ for all $\mathbf{x} \in \mathbb{B}_\varepsilon(\mathbf{x}_i)$.*

Under Assumptions 1 and 2, the infimum and supremum in both the problems (2) and (3) can be achieved, and there exists MNE in these problems. We refer the reader to the long version of this paper for details. A natural metric for measuring the quality of a candidate solution $(\tilde{\mu}, \tilde{\nu})$ of the regularized problem (3) is the primal-dual gap

$$\mathcal{G}_\beta(\tilde{\mu}, \tilde{\nu}) := \sup_{\nu \in \times_{i=1}^N \mathcal{M}_1^+(\mathbb{B}_\varepsilon(\mathbf{x}_i))} \{\mathcal{L}(\tilde{\mu}, \nu) - \beta\mathcal{H}(\nu)\} - \inf_{\mu \in \mathcal{M}_1^+(\Theta)} \{\mathcal{L}(\mu(T), \tilde{\nu}) - \beta\mathcal{H}(\tilde{\nu})\}. \quad (13)$$

Note that when $\mathcal{G}_\beta(\tilde{\mu}, \tilde{\nu}) = 0$, $(\tilde{\mu}, \tilde{\nu})$ is an MNE of (3). In the case $\beta = 0$, $\mathcal{G}_\beta$ is the primal-dual gap for the problem (2).

Regularization Error Analysis

In the following theorem, we provide an upper bound of the approximation error due to the entropy regularization.

Theorem 1. *Let $\mathcal{X}$ be $\mathbb{R}^{d_{\mathbf{x}}}$ equipped with the ℓ_∞ norm and let $\mathbb{B}_\varepsilon(\mathbf{x}_i) = \{\mathbf{x} \in \mathcal{X} : \|\mathbf{x} - \mathbf{x}_i\|_\infty\} \le \varepsilon$. Under Assumption 1 and the condition $0 < \beta \le \varepsilon / d_{\mathbf{x}}$, we have*

$$\mathcal{G}_0(\tilde{\mu}, \tilde{\nu}) \le \mathcal{G}_\beta(\tilde{\mu}, \tilde{\nu}) + \beta d_{\mathbf{x}} \log \tfrac{2\varepsilon G}{\beta d_{\mathbf{x}}} + \beta d_{\mathbf{x}}.$$

Theorem 1 shows that an (approximate) MNE of (3) is still an approximate MNE of (2) when the regularization level β is sufficiently small.

Convergence Analysis

In what follows, we analyze the convergence rate of the proposed WFR-LBR dynamics. By the definition of $\nu(T)$ in (6), when $T \ge 1$, the primal-dual gap of $(\bar{\mu}(T), \bar{\nu}(T))$ produced by the WFR-LBR flow can be written as

$$\mathcal{G}_\beta(\bar{\mu}(T), \bar{\nu}(T)) = \mathcal{L}(\bar{\mu}(T), \nu(T)) - \beta\mathcal{H}(\nu(T)) - \inf_{\mu \in \mathcal{M}_1^+(\Theta)} \{\mathcal{L}(\mu(T), \bar{\nu}(T)) - \beta\mathcal{H}(\bar{\nu}(T))\}, \quad (14)$$

We construct two new potential functions $\mathcal{R}_\mu(T)$ and $\mathcal{R}_\nu(T)$ that allow us to separately analyzing the convergence of the classifier and the attacker

$$\begin{cases} \mathcal{R}_\mu(T) := \frac{1}{T}\int_0^T \mathcal{L}(\mu(t), \nu(t))dt - \mathcal{L}(\mu^*(T), \bar{\nu}(T)) \\ \mathcal{R}_\nu(T) := \mathcal{L}(\bar{\mu}(T), \nu(T)) - \beta\mathcal{H}(\nu(T)) \\ \qquad\qquad -\frac{1}{T}\int_0^T \mathcal{L}(\mu(t), \nu(t))dt + \beta\mathcal{H}(\bar{\nu}(T)), \end{cases} \quad (15)$$

where $\mu^*(T)$ is defined as $\mu^*(T) \in \operatorname{argmin}_{\mu \in \mathcal{M}_1^+(\Theta)} \mathcal{L}(\mu, \bar{\nu}(T))$. Note that $\mathcal{G}_\beta(\bar{\mu}(T), \bar{\nu}(T)) = \mathcal{R}_\mu(T) + \mathcal{R}_\nu(T)$ when $T \ge 1$. Therefore, to analyze the convergence of $\mathcal{G}_\beta(\bar{\mu}(T), \bar{\nu}(T))$, it suffices to bound the two potential functions. The upper bounds of $\mathcal{R}_\mu(T)$ and $\mathcal{R}_\nu(T)$ are provided in Lemmas 1 and 2 below, respectively.

Lemma 1. *Under Assumptions 1-3, $\mathcal{R}_\mu(T) \le \frac{1}{\alpha T}(K + K' + d_\theta(1 - \log d_\theta + \log(\alpha(U+G)T))) + \frac{\gamma}{2}(U+G)^2 T$.*

By Lemma 1, with sufficiently small γ and large enough α and T, we can bound $\mathcal{R}_\mu(T)$ by any desired accuracy.

Lemma 2. *Under Assumptions 1-3, $\forall T \ge 1$, we have $\mathcal{R}_\nu(T) \le \frac{1}{T}\mathcal{R}_\nu(1)$, where $\mathcal{R}_\nu(1) = \max_{\nu \in \Sigma}\{\mathcal{L}(\bar{\mu}(1), \nu) - \beta\mathcal{H}(\nu)\} - (\mathcal{L}(\bar{\mu}(1), \nu^0) - \beta\mathcal{H}(\nu^0)) \ge 0$.*

Lemma 2 indicates that the potential function $\mathcal{R}_\nu(T)$ decreases at an $\mathcal{O}(1/T)$ rate. By combining Lemmas 1 and 2, we obtain the following convergence result.

Theorem 2. *Under Assumptions 1-3, for any $T \ge 1$, we have $\mathcal{G}_\beta(\bar{\mu}(T), \bar{\nu}(T)) \le \frac{1}{\alpha T}(K + K' + d_\theta(1 - \log d_\theta + \log(\alpha(U+G)T))) + \frac{\gamma}{2}(U+G)^2 T + \frac{1}{T}\mathcal{R}_\nu(1)$.*

Theorem 2 shows that if we set $\alpha = \mathcal{O}(1)$ and $\gamma = \mathcal{O}(1/T^2)$, $(\bar{\mu}(T), \bar{\nu}(T))$ is an $\tilde{\mathcal{O}}(1/T)$-approximate mixed Nash equilibrium of the regularized game (3). Further, under the condition of Theorem 1 and the additional condition that $\beta \le \mathcal{O}(1/T)$, $(\bar{\mu}(T), \bar{\nu}(T))$ is also an $\tilde{\mathcal{O}}(1/T)$-approximate mixed Nash equilibrium of the original game (2).

Numerical Experiments

In this section, we conduct numerical experiments on synthetic and real datasets to demonstrate the efficiency of the proposed algorithm.[1] We compare the proposed FRAT algorithm with SAT (Madry et al. 2018), Boosted Adversarial Training (BAT) (Pinot et al. 2020), and the algorithms in (Meunier et al. 2021). Note that SAT produces a single deterministic classification model, BAT produces a mixture of 2 models, and FRAT and those in (Meunier et al. 2021) produce mixtures of classification models with tunable sizes.

[1] Source code: https://github.com/xjiajiahao/fully-randomized-adversarial-training

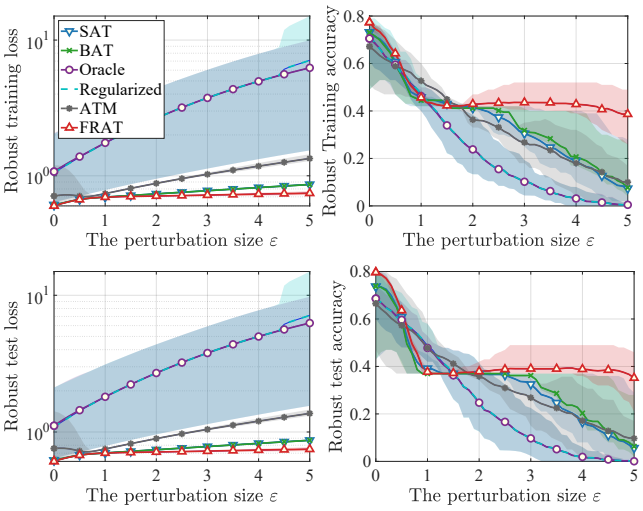

Figure 1: Results on the synthetic dataset. The first (resp., second) row reports the robust loss and accuracies on the training (resp., test) dataset. Each curve corresponds to the mean value over 6 runs with the shaded area covering the minimum/maximum values.

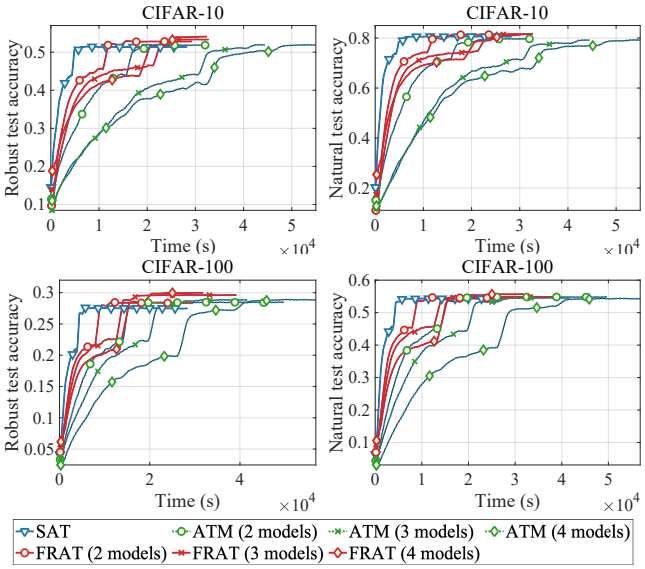

Figure 2: Results on the CIFAR-10 (top) and CIFAR-100 (bottom) datasets. The left (resp., right) column shows the test accuracy against 20 steps of PGD attack (resp, the natural accuracy on the clean test data).

Experiment on Synthetic Data

In the first experiment, we consider training adversarially robust classifiers on a synthetic dataset following (Meunier et al. 2021). The synthetic data distribution $P(\mathbf{x}, y)$ is constructed as follows. First, we let the feature space $\mathcal{X}$ be $\mathbb{R}^2$ and the label space $\mathcal{Y} = \{-1, +1\}$. Then, we set the marginal distribution of the label as $P(y = +1) = P(y = -1) = 1/2$ and set the conditional distribution of the features as $P(\mathbf{x}|y = +1) = \frac{3}{4}\mathcal{N}((3, 0), \mathbf{I}_2) + \frac{1}{4}\mathcal{N}((-3, 0), \mathbf{I}_2)$ and $P(\mathbf{x}|y = -1) = \mathcal{N}((0, 0), \mathbf{I}_2)$. Here, $\mathbf{I}_2$ denotes the identity matrix in $\mathbb{R}^{2\times 2}$ and $\mathcal{N}(\mathbf{x}, \mathbf{I}_2)$ denotes the Gaussian distribution with mean $\mathbf{x}$ and covariance matrix $\mathbf{I}_2$. We generate the training dataset by randomly drawing $N = 100$ data samples from the distribution $P(\mathbf{x}, y)$. The test dataset is independently generated in the same way. We use linear classification models of the form $\theta = (\mathbf{w}^T, b)^T \in \Theta = \mathbb{R}^3$, where $\mathbf{w} \in \mathbb{R}^2$ and $b \in \mathbb{R}$ are the weight and bias parameters, respectively. The loss function $\ell(\theta, (\mathbf{x}, y))$ and the constraint set $\mathbb{B}_\varepsilon$ in (1) are defined as the logistic loss function and the ℓ_2 norm ball with radius ε, respectively.

We compare FRAT with five baselines: SAT, BAT, and the oracle algorithm, the regularized algorithm, and the Adversarial Training of Mixtures (ATM) algorithm in (Meunier et al. 2021). Note that the oracle algorithm and the regularized algorithm are restricted to the case where Θ is finite discrete space. To apply these two algorithms in our experiment, we follow (Meunier et al. 2021) and generate a finite discrete model space by randomly sampling 20 linear models from $[-7, 7]^2$ with accuracies higher than 0.6 on the clean training data. For a fair comparison, we set the size of the mixture of models in FRAT and ATM to 20 and initialize the 20 models randomly. In the implementation of FRAT, we use the PLA algorithm described previously as the sampling subroutine, and the expectation in (12) is estimated by drawing 100 models from $\{\mu^{(0)}, \ldots, \mu^{(t)}\}$ when $t \geq 100$. The regularization parameter in both FRAT and the regularized algorithm in (Meunier et al. 2021) is set to 0.01. In addition, to approximate the inner maximization problem in (1), which is required by SAT, BAT, the oracle algorithm, and ATM, we first uniformly sample 1000 points from $\mathbb{B}_\varepsilon(\mathbf{x}_i)$ and then selects the one maximizing the loss, where $\mathbf{x}_i$ is the feature vector of a sample to be attacked. We run each algorithm until convergence under different perturbation sizes from the range $\{0, 0.1, 0.2, \ldots, 5.0\}$.

The experimental results are shown in Figure 1. Generally, it can be observed that FRAT achieves the lowest robust training/test loss and highest training/test accuracies among the 6 algorithms, where the robust training/test loss is the loss value of the classifier on the perturbed training/test data samples generated by the adversary described above, and the robust training/test accuracy is defined accordingly. Moreover, our FRAT algorithm constantly outperforms other methods by a large margin in terms of the robust training/test accuracies for large perturbation level ($\varepsilon > 2.5$). We also observe that BAT slightly outperforms SAT in terms of the robust training and test accuracies, and ATM is slightly inferior to SAT under a large range of ε. The oracle and the regularized algorithms, which only update the mixing weights of the initialized models and keep the model parameters fixed during training, are significantly inferior to other algorithms. These two algorithms also have high variance among different runs, implying that their performance is sensitive to the quality of the initialized models.

Experiments on Real Data

To demonstrate the efficiency of the proposed algorithm in practice, we conduct experiments on CIFAR-10 and CIFAR-

Dataset	Algorithm	Natural Test Accuracy	APGD_{CE}	APGD_{DLR}	APGD_{CE} & APGD_{DLR}
CIFAR-10	SAT	80.6	49.7	48.1	47.6
	BAT	80.6	**53.3**	45.2	45.0
	ATM (2 models)	79.6	50.2	47.7	47.3
	FRAT (2 models)	81.1	50.9	49.3	48.5
	ATM (3 models)	79.2	50.4	47.8	47.3
	FRAT (3 models)	81.1	51.5	49.7	48.9
	ATM (4 models)	79.2	50.4	47.7	47.3
	FRAT (4 models)	**81.6**	52.1	**50.1**	**49.4**
CIFAR-100	SAT	54.1	26.5	23.4	23.2
	BAT	52.5	**28.9**	22.6	22.5
	ATM (2 models)	54.9	27.5	24.1	23.7
	FRAT (2 models)	54.6	27.5	24.3	23.9
	ATM (3 models)	54.5	27.9	24.4	24.1
	FRAT (3 models)	54.8	28.4	25.5	24.9
	ATM (4 models)	54.2	27.9	24.6	24.2
	FRAT (4 models)	**55.6**	28.7	**26.1**	**25.5**

Table 1: Results of the compared algorithms on CIFAR-10 and CIFAR-100 datasets. The third column shows the natural accuracy on the clean test data. The fourth (resp., fifth) column correspond to the robust test accuracies against $\text{AutoPGD}_{\text{CE}}$ (resp., APGD_{DLR}). The last column presents the robust accuracy against the combination of $\text{AutoPGD}_{\text{CE}}$ and APGD_{DLR} attacks.

100 datasets (Krizhevsky, Hinton et al. 2009). We compare FRAT with SAT, BAT, and ATM. The oracle and regularized algorithms in (Meunier et al. 2021) are excluded from our baselines as they are impractical in high-dimensional spaces. For ATM and FRAT, we test different sizes M of the mixture in the range $\{2, 3, 4\}$, and denote the corresponding algorithms as ATM/FRAT (M models). We basically follow the experimental setting in (Meunier et al. 2021). The detailed setting is deferred to the long version of this paper. For FRAT, we implement the sampling subroutine with 10 steps of PLA (12), where the noise level γ is set to 0.0001 and the expectation over $\bar{\mu}^{(t)}$ is approximated with the sliding window trick described previously. We set the size of the sliding window to 1, which already performs well. The average runtime per iteration of SAT is 0.72 s on CIFAR-10 (resp., 1.63 s on CIFAR-100); FRAT with M = 2, 3, and 4 models take 1.50 s, 2.24 s, and 3.19 s on CIFAR-10 (resp., 1.63 s, 2.30 s, and 2.87 s on CIFAR-100), which are about M times of SAT and corroborate our analysis.

After training, we evaluate the classifier obtained by each algorithm using an adapted version of AutoPGD untargeted attacks (Croce and Hein 2020) with both Cross Entropy (CE) and Difference of Logits Ratio (DLR) loss. We refer to these two attacks as APGD_{CE} and APGD_{DLR}.

The training curves of SAT, ATM, and FRAT are shown in Figure 2, where the robust test accuracy is the accuracy of a classifier on perturbed data samples generated by 20 steps of PGD attack (PGD_{20} for short), and the natural test accuracy is the accuracy on clean data samples. Note that the result of BAT is excluded, because it is not an iterative algorithm but rather a one-step boosting method based on SAT. In addition, we terminate the training when the performance plateaus. We observe that both ATM and FRAT with mixture sizes in the range $\{2, 3, 4\}$ achieve higher robust test accuracies than that of SAT, while the natural test accuracies of all algorithms are similar. This demonstrates the efficiency of using randomized strategies. Moreover, FRAT converges faster than ATM in terms of both the robust and natural accuracies when they use the same mixture size, and the robust test accuracy of FRAT at convergence is higher than that of ATM. Note that as the size M of the mixture increases, the convergence rate of ATM slows down significantly, whereas the convergence rate of FRAT is less affected by the size M.

Table 1 compares the performance of SAT, BAT, ATM, and FRAT after training. We can see that as M increases, the performance of FRAT improves, and FRAT with $M = 4$ achieves the best natural test accuracy, the accuracy against APGD_{DLR}, and the accuracy against APGD_{CE} & APGD_{DLR}. We also observe that FRAT outperforms ATM when they use the same size M, and FRAT with $M = 2$ already outperforms SAT and BAT. These results demonstrate the robustness of FRAT against strong attacks, and the superiority of FRAT over SAT, BAT, and ATM. Note that as the training procedure of BAT uses the CE loss to generate adversarial examples, BAT achieves the best performance in terms of the accuracy against APGD_{CE}. However, BAT performs worse in terms of the accuracy against the APGD_{DLR} attack and the APGD_{CE} & APGD_{CE} attack, which indicates that BAT is vulnerable to general attacks other than CE.

Conclusion

In this paper, we propose an efficient algorithm for finding optimal randomized strategies in the AT game. Our algorithm FRAT is obtained by discretizing a new continuous-time and interacting flow that properly exploits the problem structure. We prove that this flow converges to an MNE at a sublinear rate. Experimental results demonstrate the advantages of FRAT over existing ones.

Acknowledgements

This work is supported by National Key Research and Development Program of China under Grant 2020AAA0107400, Zhejiang Provincial Natural Science Foundation of China under Grant No. LZ18F020002, and National Natural Science Foundation of China (Grant No: 62206248).

References

Bai, T.; Luo, J.; Zhao, J.; Wen, B.; and Wang, Q. 2021. Recent Advances in Adversarial Training for Adversarial Robustness. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, 4312–4321.

Biggio, B.; Corona, I.; Maiorca, D.; Nelson, B.; Šrndić, N.; Laskov, P.; Giacinto, G.; and Roli, F. 2013. Evasion attacks against machine learning at test time. In *Joint European conference on machine learning and knowledge discovery in databases*, 387–402. Springer.

Bonesky, T.; Bredies, K.; Lorenz, D. A.; and Maass, P. 2007. A generalized conditional gradient method for nonlinear operator equations with sparsity constraints. *Inverse Problems*, 23(5): 2041.

Bose, J.; Gidel, G.; Berard, H.; Cianflone, A.; Vincent, P.; Lacoste-Julien, S.; and Hamilton, W. 2020. Adversarial example games. *Advances in neural information processing systems*, 33: 8921–8934.

Bredies, K.; Lorenz, D. A.; and Maass, P. 2009. A generalized conditional gradient method and its connection to an iterative shrinkage method. *Computational Optimization and applications*, 42(2): 173–193.

Bubeck, S.; Eldan, R.; and Lehec, J. 2018. Sampling from a log-concave distribution with projected langevin monte carlo. *Discrete & Computational Geometry*, 59(4): 757–783.

Bulò, S. R.; Biggio, B.; Pillai, I.; Pelillo, M.; and Roli, F. 2016. Randomized prediction games for adversarial machine learning. *IEEE transactions on neural networks and learning systems*, 28(11): 2466–2478.

Chizat, L. 2022. Sparse optimization on measures with over-parameterized gradient descent. *Mathematical Programming*, 194(1): 487–532.

Cohen, J.; Rosenfeld, E.; and Kolter, Z. 2019. Certified adversarial robustness via randomized smoothing. In *International Conference on Machine Learning*, 1310–1320. PMLR.

Croce, F.; and Hein, M. 2020. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *International conference on machine learning*, 2206–2216. PMLR.

Dhillon, G. S.; Azizzadenesheli, K.; Lipton, Z. C.; Bernstein, J. D.; Kossaifi, J.; Khanna, A.; and Anandkumar, A. 2018. Stochastic Activation Pruning for Robust Adversarial Defense. In *International Conference on Learning Representations*.

Domingo-Enrich, C.; Jelassi, S.; Mensch, A.; Rotskoff, G.; and Bruna, J. 2020. A mean-field analysis of two-player zero-sum games. *Advances in neural information processing systems*, 33: 20215–20226.

Goodfellow, I. J.; Shlens, J.; and Szegedy, C. 2015. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*.

Gowal, S.; Rebuffi, S.-A.; Wiles, O.; Stimberg, F.; Calian, D. A.; and Mann, T. A. 2021. Improving robustness using generated data. *Advances in Neural Information Processing Systems*, 34: 4218–4233.

Hofbauer, J.; and Sandholm, W. H. 2002. On the global convergence of stochastic fictitious play. *Econometrica*, 70(6): 2265–2294.

Hsieh, Y.-P.; Liu, C.; and Cevher, V. 2019. Finding mixed nash equilibria of generative adversarial networks. In *International Conference on Machine Learning*, 2810–2819. PMLR.

Krizhevsky, A.; Hinton, G.; et al. 2009. Learning multiple layers of features from tiny images. Citeseer.

Lahkar, R.; and Riedel, F. 2015. The logit dynamic for games with continuous strategy sets. *Games and Economic Behavior*, 91: 268–282.

Liero, M.; Mielke, A.; and Savaré, G. 2018. Optimal entropy-transport problems and a new Hellinger–Kantorovich distance between positive measures. *Inventiones mathematicae*, 211(3): 969–1117.

Liu, L.; Zhang, Y.; Yang, Z.; Babanezhad, R.; and Wang, Z. 2021. Infinite-Dimensional Optimization for Zero-Sum Games via Variational Transport. In *International Conference on Machine Learning*, 7033–7044. PMLR.

Liu, Q. 2017. Stein variational gradient descent as gradient flow. *Advances in neural information processing systems*, 30.

Liutkus, A.; Simsekli, U.; Majewski, S.; Durmus, A.; and Stöter, F.-R. 2019. Sliced-Wasserstein flows: Nonparametric generative modeling via optimal transport and diffusions. In *International Conference on Machine Learning*, 4104–4113. PMLR.

Ma, C.; and Ying, L. 2021. Provably convergent quasistatic dynamics for mean-field two-player zero-sum games. In *International Conference on Learning Representations*.

Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; and Vladu, A. 2018. Towards Deep Learning Models Resistant to Adversarial Attacks. In *International Conference on Learning Representations*.

Maini, P.; Wong, E.; and Kolter, Z. 2020. Adversarial robustness against the union of multiple perturbation models. In *International Conference on Machine Learning*, 6640–6650. PMLR.

Meunier, L.; Scetbon, M.; Pinot, R. B.; Atif, J.; and Chevaleyre, Y. 2021. Mixed nash equilibria in the adversarial examples game. In *International Conference on Machine Learning*, 7677–7687. PMLR.

Moosavi-Dezfooli, S.-M.; Fawzi, A.; Uesato, J.; and Frossard, P. 2019. Robustness via curvature regularization, and vice versa. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9078–9086.

Pang, T.; Yang, X.; Dong, Y.; Su, H.; and Zhu, J. 2020. Bag of Tricks for Adversarial Training. In *International Conference on Learning Representations*.

Papernot, N.; McDaniel, P.; Wu, X.; Jha, S.; and Swami, A. 2016. Distillation as a defense to adversarial perturbations against deep neural networks. In *2016 IEEE symposium on security and privacy (SP)*, 582–597. IEEE.

Perdomo, J. C.; and Singer, Y. 2019. Robust attacks against multiple classifiers. *arXiv preprint arXiv:1906.02816*.

Perkins, S.; and Leslie, D. S. 2014. Stochastic fictitious play with continuous action sets. *Journal of Economic Theory*, 152: 179–213.

Pinot, R.; Ettedgui, R.; Rizk, G.; Chevaleyre, Y.; and Atif, J. 2020. Randomization matters how to defend against strong adversarial attacks. In *International Conference on Machine Learning*, 7717–7727. PMLR.

Pinot, R.; Meunier, L.; Araujo, A.; Kashima, H.; Yger, F.; Gouy-Pailler, C.; and Atif, J. 2019. Theoretical evidence for adversarial robustness through randomization. *Advances in Neural Information Processing Systems*, 32.

Rotskoff, G.; Jelassi, S.; Bruna, J.; and Vanden-Eijnden, E. 2019. Global convergence of neuron birth-death dynamics. In *International Conference on Machine Learning*.

Samangouei, P.; Kabkab, M.; and Chellappa, R. 2018. Defense-GAN: Protecting Classifiers Against Adversarial Attacks Using Generative Models. In *International Conference on Learning Representations*.

Schott, L.; Rauber, J.; Bethge, M.; and Brendel, W. 2018. Towards the first adversarially robust neural network model on MNIST. In *International Conference on Learning Representations*.

Shaham, U.; Yamada, Y.; and Negahban, S. 2018. Understanding adversarial training: Increasing local stability of supervised models through robust optimization. *Neurocomputing*, 307: 195–204.

Su, W.; Boyd, S.; and Candes, E. 2014. A differential equation for modeling Nesterov's accelerated gradient method: theory and insights. *Advances in neural information processing systems*, 27.

Suggala, A.; and Netrapalli, P. 2020. Follow the perturbed leader: Optimism and fast parallel algorithms for smooth minimax games. *Advances in Neural Information Processing Systems*, 33: 22316–22326.

Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; and Fergus, R. 2014. Intriguing properties of neural networks. In *International Conference on Learning Representations*.

Wang, B.; Shi, Z.; and Osher, S. 2019. Resnets ensemble via the feynman-kac formalism to improve natural and robust accuracies. *Advances in Neural Information Processing Systems*, 32.

Welling, M.; and Teh, Y. W. 2011. Bayesian learning via stochastic gradient Langevin dynamics. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, 681–688. Citeseer.

Xie, C.; Wang, J.; Zhang, Z.; Ren, Z.; and Yuille, A. 2018. Mitigating Adversarial Effects Through Randomization. In *International Conference on Learning Representations*.

Zhang, H.; Yu, Y.; Jiao, J.; Xing, E.; El Ghaoui, L.; and Jordan, M. 2019. Theoretically principled trade-off between robustness and accuracy. In *International conference on machine learning*, 7472–7482. PMLR.