MAT: MIXED-STRATEGY GAME OF ADVERSARIAL TRAINING IN FINE-TUNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Fine-tuning large-scale models from pre-trained checkpoints has been demonstrated effective for various natural language processing (NLP) tasks. Previous works reveal that leveraging adversarial training methods during the fine-tuning stage significantly enhances the generalization and robustness of the models. However, from the perspective of optimization, the previous adversarial training methods suffer from converging onto local optima due to the non-convexity of the objective. In this work, we reformulate the adversarial training in the view of mixed strategy in game theory and incorporate full strategy space to avoid trapping in local stationarity. Methodologically, we derive the Nash equilibrium of mixed-strategy for adversarial training using entropy mirror descent to establish a novel mixed-strategy adversarial training algorithm (MAT). Numerically, to verify the effectiveness of MAT, we conducted extensive benchmark experiments over the large-scale pre-trained models such as BERT and RoBERTa. The experimental results show that MAT outperforms the previous state-of-the-art on both GLUE and ANLI benchmarks in terms of generalization and robustness.

1 INTRODUCTION

Natural language processing (NLP) has revolutionized in recent years due to the pre-trained language models based on large-scale text data such as BERT (Devlin et al., 2019), GPT (Radford et al., 2018), and T5 (Raffel et al., 2020). This area has emerged as a hot spot for the advancement of artificial intelligence technology. As we saw from previous research, pre-trained language models make up for the lack of labelled data in NLP and have made significant progress in almost all NLP tasks. Fine-tuning training is the critical stage for the pre-trained models to adapt to downstream tasks, which replaces the top layer of the pre-trained model with a task sub-network and re-trains the model with the limited data of the target task. The vast number of parameters enhances the capabilities of the model. However, it also complicates the model's development, training, and use. The main problem is that the pre-trained model may overfit the training data of the target task during the fine-tuning process, resulting in poor generalization performance.

Some recent studies (Liu et al., 2020; Zhu et al., 2020; Jiang et al., 2020; Aghajanyan et al., 2021) have demonstrated that combining fine-tuning with adversarial training can successfully alleviate the problem mentioned above and improve the generalization of the model in downstream tasks. Furthermore, adversarial training in the fine-tuning stage is mainly used as a regularization method to prevent over-fitting rather than to defend models against adversarial attacks, which is widely used in computer vision. Some of the above research has the state-of-the-art experimental results, but adversarial training still has potential for improvement from the game theory perspective.

Our research treats adversarial training as a game and ameliorates it with mixed strategy from game theory. We advocate that adversarial training is a two-player complete-information game between the model and adversarial perturbations. Existing adversarial training is in line with a pure-strategy game in which the strategies of both sides are specific. On the contrary, we expand the adversarial training to a mixed-strategy game in which the strategies are probabilistic. Moreover, in Section 3.1, we will go into further detail on pure strategy and mixed strategy in deep learning. As a simple example, in Figure 1, this is a game of two-person rock-paper-scissors, where are three pure strategies (rock, paper, and scissors) for each side. Under a pure-strategy game, both sides will choose a pure strategy like scissors or paper for the next turn. However, they will plan a probability distribution

over all pure strategies under a mixed-strategy game. According to the Nash theorem (Nash, 1951), a pure-strategy game is a subset of a mixed-strategy game, and the Nash equilibrium (which can be seen as the final result of the game) under a pure-strategy game may not exist. To fully harness the power of adversarial training, we turn to the mixed-strategy game.

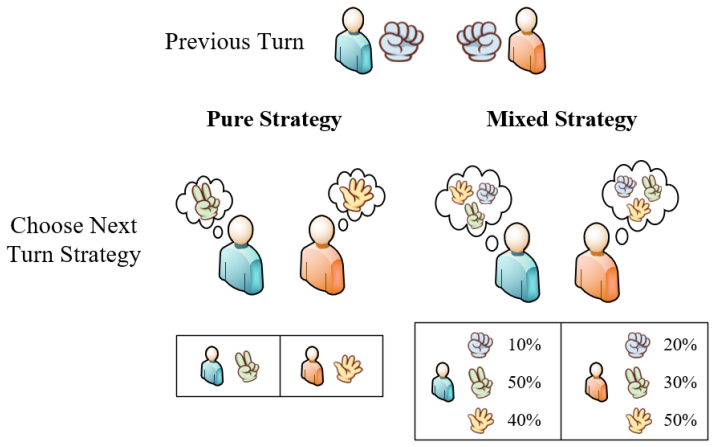

Figure 1: The difference between pure strategy and mixed strategy.

Our main contributions can be summarized as follows:

- We apply game theory to consider adversarial training as a mixed-strategy game. Not only did we deduce the theoretical algorithm by Entropy Mirror Descent, but we also condensed it into an algorithm that can be used for training. Since our adversarial training algorithm is based on the mixed strategy, we name it MAT (**M**ixed-strategy **A**dversarial **T**raining).
- We conduct extensive experiments to verify the effectiveness of MAT and obtain state-of-the-art experimental results. Specifically, the results of the BERT model (Devlin et al., 2019) and the RoBERTa model (Liu et al., 2019) evaluated on the GLUE (Wang et al., 2019) benchmark and the ANLI (Nie et al., 2020) benchmark exceed previous research.

2 Related Work

2.1 Adversarial Training: from CV to NLP

Adversarial training was first proposed together with adversarial examples by Szegedy et al. (2014) and Goodfellow et al. (2015) in the field of computer vision (CV). They explained some reasons for the existence of adversarial examples and defined adversarial training as re-train the model with the datasets of image adversarial examples, which can successfully shield the model from adversarial attacks. Madry et al. (2018) proposed a well-known Min-Max formula to describe the relationship between the model and adversarial examples, and put forward a method named PGD (projected gradient descent) to generate adversarial examples. Then Shafahi et al. (2019) and Zhang et al. (2019) optimized the PGD adversarial training algorithm, mainly to reduce the complexity of training.

Adversarial training came to NLP relying on Miyato et al. (2017), who perturbed the word embeddings to reduce overfitting when training models. After the popularity of pre-trained models, Zhu et al. (2020) extended adversarial training into the fine-tuning stage of pre-trained models and improved the Min-Max formula proposed by Madry et al. (2018). Jiang et al. (2020) advanced that fine-tuning adversarial training could be regarded as a regularization method to control the complexity of the model effectively. Furthermore, their loss function comprehensively considered the loss of normal model training, adversarial training, and model parameters update aggressively. Aghajanyan et al. (2021) split the pre-trained model into a feature extraction part and a classifier of separate constraints, and then modified the adversarial perturbations to the normal or uniformly distributed noise, which dramatically reduces the time to generate adversarial perturbations.

2.2 GAME THEORY IN DEEP LEARNING

Generative adversarial network (GAN) is the game-inspired and widely used deep learning algorithm; accordingly, there are a number of researches on game theory in GAN. Such as, Arora et al. (2017) advocated the mixed strategy GAN training algorithm named the MIX+GAN protocol, which used a small mixture of discriminators and generators. Daskalakis et al. (2018) brought forward to train Wasserstein GANs by Optimistic Mirror Decent (OMD), and they formally showed that the last iteration of the OMD dynamics converges to a Nash equilibrium in the case of this zero-sum game. Hsieh et al. (2019) optimized the set of probability distributions over the pure strategy of the neural networks and suggested using a provably convergent GAN training approach. Ahuja et al. (2020) posed the standard risk minimization paradigm of machine learning when finding the Nash equilibrium of an ensemble game among several environments. Although there is not much work applying game theory in fine-tuning training that we focus on, we found Zuo et al. (2021) proposed SALT (Stackelberg Adversarial Training) formulate adversarial training as a Stackelberg game, in which the follower generates perturbations and the leader trains the model affected by perturbations.

3 ADVERSARIAL TRAINING OF MIXED-STRATEGY GAME

In Section 3.1, we elaborate on pure strategy and mixed strategy in deep learning and explain why we are searching for the mixed-strategy Nash equilibrium. Given that some readers may not be familiar with the mixed-strategy Nash equilibrium in game theory, we review the standard methods in game theory in Section 3.2. On this basis, in Section 3.3, we apply Entropy Mirror Descent to deduce the mixed-strategy Nash equilibrium of adversarial training. Due to the complexity of the above theoretical algorithm, we transform it into a feasible solution in Section 3.4.

Notation: Throughout the paper, we use $f_{\boldsymbol{\theta}}(\boldsymbol{x})$ to denote the output of the model f with parameters $\boldsymbol{\theta}$, which input is word embeddings $\boldsymbol{x}$. $\boldsymbol{\delta}$ denotes adversarial perturbations. D denotes the dataset of the downstream task, and B denotes the batch of the dataset. $\mathcal{D}_{KL}(P\|Q) = \sum_k p_k \log(p_k/q_k)$ denotes the KL-divergence of two discrete distributions P and Q.

3.1 PURE STRATEGY AND MIXED STRATEGY IN DEEP LEARNING

Customarily, the model parameters $\boldsymbol{\theta}$ in deep learning are considered to be determinable general variables, whether in vanilla model training or adversarial training. If it corresponds to game theory, the values of parameters can be referred to as the strategies of models. Because of the continuity of the parameter values, an infinite number of strategies exist for the model. But each update of the parameters is deterministic no matter which optimizer is used, which means models choose only one pure strategy at a time, so that is still the pure strategy. For example, the gradient descent algorithm is used for the optimizations of the model parameters, which chooses the strategy by $\boldsymbol{\theta}^{t+1} = \boldsymbol{\theta}^t - \gamma \times \boldsymbol{g}^t$. And if $\boldsymbol{\theta}^t$ is determined, the strategy for the next step $\boldsymbol{\theta}^{t+1}$ is also determined.

However, if we convert the model from pure strategy to mixed strategy, the straightforward method is having model parameters $\boldsymbol{\theta}$ follow a probability distribution; in other words, $\boldsymbol{\theta}$ becomes a continuous random variable. In this way, the model will update a distribution instead of a value for choosing the next strategy during model training. For example, in a simple assumption, $\boldsymbol{\theta}^t$ may follow the standard normal distribution $\mathcal{N}(0, 1)$, and after parameters updating, $\boldsymbol{\theta}^t$ may follow the normal distribution $\mathcal{N}(3, 5)$. When turning to adversarial training, whose loss function is $l\left(f_{\boldsymbol{\theta}}\left(\boldsymbol{x}+\boldsymbol{\delta}\right), y\right)$, the adversarial perturbation $\boldsymbol{\delta}$ is further taken for a continuous random variable, like above $\boldsymbol{\theta}$. In this manner, the mixed-strategy game of adversarial training is defined as: two game players are the model and adversarial perturbations; strategies of both sides are distributions of their parameters, respectively; and the payoff of the game is the value of the objective function.

3.2 MIXED-STRATEGY NASH EQUILIBRIUM IN GAME THEORY

In reference to game scenarios in Binmore et al. (2007), Player 1 and Player 2 have m and n pure strategies, respectively. Therefore the mixed strategy of Player 1 is an m-dimensional vector $\boldsymbol{p}$, and the mixed strategy of Player 2 is an n-dimensional vector $\boldsymbol{q}$. Then their payoff functions are:

$$\begin{aligned}\Pi_1(\boldsymbol{p},\boldsymbol{q}) &= \boldsymbol{p}^\top \boldsymbol{A}\boldsymbol{q} = \langle \boldsymbol{p}, \boldsymbol{A}\boldsymbol{q}\rangle,\\ \Pi_2(\boldsymbol{p},\boldsymbol{q}) &= \boldsymbol{p}^\top \boldsymbol{B}\boldsymbol{q} = \langle \boldsymbol{p}, \boldsymbol{B}\boldsymbol{q}\rangle,\end{aligned} \tag{1}$$

where $\boldsymbol{A}, \boldsymbol{B}$ are the payoff matrices of Player 1 and Player 2, and each element in $\boldsymbol{p}, \boldsymbol{q}$ is represented the probability of each pure strategy. The game may be described as following functions:

$$\min_{\boldsymbol{p}\in\Delta_m} \max_{\boldsymbol{q}\in\Delta_n} \langle \boldsymbol{p}, \boldsymbol{B}\boldsymbol{q}\rangle - \langle \boldsymbol{p}, \boldsymbol{A}\boldsymbol{q}\rangle, \tag{2}$$

where $\Delta_d := \{\boldsymbol{z} \in \mathbb{R}^d \mid \sum_{i=1}^d z_i = 1\}$ is the probability simplex. In the Δ_d, d is the number of pure strategies, and z_i is the probability of the i-th pure strategy. Beck & Teboulle (2003) proposed a descent algorithm named Entropic Mirror Descent, which can find the Nash equilibrium of the game through the following iterations if we abbreviate Eq.2 with F:

$$\begin{cases} \boldsymbol{p}_{t+1} = \mathrm{MD}_\eta\left(\boldsymbol{p}_t, \partial F/\partial \boldsymbol{p}_t\right) \\ \boldsymbol{q}_{t+1} = \mathrm{MD}_\eta\left(\boldsymbol{q}_t, \partial F/\partial \boldsymbol{q}_t\right) \end{cases} \quad \Rightarrow \quad \left(\overline{\boldsymbol{p}}_T, \overline{\boldsymbol{q}}_T\right), \tag{3}$$

where $\overline{\boldsymbol{p}}_T, \overline{\boldsymbol{q}}_T$ are the average of $\boldsymbol{p}, \boldsymbol{q}$ through T times iterations. As for MD_η is defined as:

$$\boldsymbol{z}^+ = \mathrm{MD}_\eta(\boldsymbol{z}, \boldsymbol{b}) \quad \equiv \quad \boldsymbol{z}^+ = \nabla\Phi^\star\left(\nabla\Phi\left(\boldsymbol{z}\right) - \eta\boldsymbol{b}\right) \quad \equiv \quad z_i^+ = \frac{z_i e^{-\eta b_i}}{\sum_{i=1}^d z_i e^{-\eta b_i}}, \tag{4}$$

where η is the learning rate, and $\Phi(\boldsymbol{z}) := \sum_{i=1}^d z_i \log z_i$ is the entropy function, so its Fenchel dual is $\Phi^\star(\boldsymbol{y}) := \log \sum_{i=1}^d e^{y_i}$. The above MD_η iterations is established in the discrete finite dimensions, and Hsieh et al. (2019) expanded it to the continuous infinite dimensional space:

$$z^+ = \mathrm{MD}_\eta(z, h) \quad \equiv \quad z^+ = \mathrm{d}\Phi^\star(\mathrm{d}\Phi(z) - \eta h) \quad \equiv \quad \mathrm{d}z^+ = \frac{e^{-\eta h}\mathrm{d}z}{\int e^{-\eta h}\mathrm{d}z}, \tag{5}$$

With Entropic Mirror Descent, solving mixed-strategy games is no longer difficult, but the next question is how to transform the adversarial training into a mixed-strategy game.

3.3 Mixed-Strategy Nash Equilibrium in Adversarial Training

The objective functions in previous work using adversarial training for fine-tuning have not been completely unified. As a starting point, we employ the objective function in SMART (Jiang et al., 2020), which giving the pre-trained model $f_{\boldsymbol{\theta}}$ and the downstream dataset D can get:

$$\min_{\boldsymbol{\theta}\in\Theta} \mathbb{E}_{B\sim D} \max_{\boldsymbol{\delta}\in\Delta} \mathbb{E}_{(\boldsymbol{x},y)\sim B}\left[L\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}), y\right) + \lambda l\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}+\boldsymbol{\delta}), f_{\boldsymbol{\theta}}(\boldsymbol{x})\right)\right]. \tag{6}$$

There are two loss functions, L and l, included in this function, where L is the loss function of the target task, and l measures the loss of adversarial training. In classification tasks, l is chosen as the KL-divergence, i.e., $l(P,Q) = \mathcal{D}_{KL}(P\|Q) + \mathcal{D}_{KL}(Q\|P)$; but in regression tasks, l is the squared loss, i.e., $l(p,q) = (p-q)^2$. And λ is a tuning parameter of two loss functions.

Eq.6 shows a pure-strategy game, since the model parameters $\boldsymbol{\theta}$ and the adversarial perturbations $\boldsymbol{\delta}$ are deterministic values, not distributions. Under the pure-strategy game, the strategy sets of the model parameters and adversarial perturbations are Θ and Δ, respectively. As mentioned in Section 3.1, let us transform the game into a mixed-strategy game. In practice, we consider the set of all probability distributions over Θ and Δ. If denote the set of all Borel probability measures on Θ and Δ by $\mathcal{M}(\Theta)$ and $\mathcal{M}(\Delta)$, we will have the following Min-Max function:

$$\min_{\mu\in\mathcal{M}(\Theta)} \mathbb{E}_{B\sim D} \max_{\nu\in\mathcal{M}(\Delta)} \mathbb{E}_{(\boldsymbol{x},y)\sim B}\mathbb{E}_{\boldsymbol{\theta}\sim\mu}\mathbb{E}_{\boldsymbol{\delta}\sim\nu}\left[L\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}), y\right) + \lambda l\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}+\boldsymbol{\delta}), f_{\boldsymbol{\theta}}(\boldsymbol{x})\right)\right]. \tag{7}$$

Finding the optimal probability distributions (μ and ν) of model parameters and adversarial perturbations takes the place of the optimum values ($\boldsymbol{\theta}$ and $\boldsymbol{\delta}$). Therefore, Eq.7 is the objective function of the mixed-strategy game in that we are going to find the Nash equilibrium. It is not difficult to get the partial derivatives of μ and ν concerning this game function if we abbreviate Eq.7 with F:

$$\begin{aligned}\frac{\partial F}{\partial \mu} &= \mathbb{E}_{(\boldsymbol{x},y)\sim D}\left[L\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}),y\right)+\lambda \mathbb{E}_{\boldsymbol{\delta}\sim\nu} l\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}+\boldsymbol{\delta}),f_{\boldsymbol{\theta}}(\boldsymbol{x})\right)\right], \\ \frac{\partial F}{\partial \nu} &= \mathbb{E}_{(\boldsymbol{x},y)\sim B}\mathbb{E}_{\boldsymbol{\theta}\sim\mu} l\left(f_{\boldsymbol{\theta}}(\boldsymbol{x}+\boldsymbol{\delta}),f_{\boldsymbol{\theta}}(\boldsymbol{x})\right).\end{aligned} \tag{8}$$

Theoretically, borrowing from MD_η iterations in Eq.5 of the Entropic Mirror Descent algorithm, Algorithm 1 will provide the Nash equilibrium of the above mixed-strategy game. However, since unable to directly extract the density function of μ and ν, it cannot actually find the Nash equilibrium, so we will apply a feasible simplified sampling approach below to solve the game.

Algorithm 1: Entropic Mirror Descent for Adversarial Training

Input: Initial distributions: μ_0, ν_0, learning rate: η.
for $t = 0, 1, \ldots, T-1$ **do**
  $\nu_{t+1} = \mathrm{MD}_\eta\left(\nu_t, \partial F_t/\partial \nu_t\right)$
  $\mu_{t+1} = \mathrm{MD}_\eta\left(\mu_t, \partial F_t/\partial \mu_t\right)$
end
$\bar{\mu}_T = \frac{1}{T}\sum_{t=1}^{T}\mu_t$
Output: Probability distribution of the model parameters: $\bar{\mu}_T$.

3.4 From Theory to Practice

Due to the infeasibility of obtaining density functions of the μ and ν in Eq.8, taking samples to estimate them may be the next best way. As for the expectation of distribution $\mathbb{E}$, a common approach is to replace it with an empirical average. Considering those mentioned above, if we set the sampling times to n' and the batch size to n, the partial derivatives of μ and ν are denoted as follows:

$$\begin{aligned}\frac{\partial F}{\partial \mu} &= \frac{1}{n}\sum_{i=1}^{n}\left[L\left(f_{\boldsymbol{\theta}}\left(\boldsymbol{x}_i\right),y_i\right)+\lambda\frac{1}{n'}\sum_{j=1}^{n'} l\left(f_{\boldsymbol{\theta}}\left(\boldsymbol{x}_i+\boldsymbol{\delta}_j\right),f_{\boldsymbol{\theta}}\left(\boldsymbol{x}_i\right)\right)\right], \\ \frac{\partial F}{\partial \nu} &= \frac{1}{n}\sum_{i=1}^{n}\frac{1}{n'}\sum_{j=1}^{n'} l\left(f_{\boldsymbol{\theta}_j}\left(\boldsymbol{x}_i+\boldsymbol{\delta}\right),f_{\boldsymbol{\theta}_j}\left(\boldsymbol{x}_i\right)\right).\end{aligned} \tag{9}$$

However, in this way, each sample needs to be processed forward and backward propagation, causing the algorithm complexity to be immense. Thus we express a distribution by the mean of samples rather than a collection of samples. Then Eq.9 is simplified to:

$$\begin{aligned}\frac{\partial F}{\partial \mu} &= \frac{1}{n}\sum_{i=1}^{n}\left[L\left(f_{\boldsymbol{\theta}}\left(\boldsymbol{x}_i\right),y_i\right)+\lambda l\left(f_{\boldsymbol{\theta}}\left(\boldsymbol{x}_i+\bar{\boldsymbol{\delta}}\right),f_{\boldsymbol{\theta}}\left(\boldsymbol{x}_i\right)\right)\right], \\ \frac{\partial F}{\partial \nu} &= \frac{1}{n}\sum_{i=1}^{n} l\left(f_{\overline{\boldsymbol{\theta}}}\left(\boldsymbol{x}_i+\boldsymbol{\delta}\right),f_{\overline{\boldsymbol{\theta}}}\left(\boldsymbol{x}_i\right)\right),\end{aligned} \tag{10}$$

where $\overline{\boldsymbol{\theta}}$ and $\overline{\boldsymbol{\delta}}$ can calculated by exponential moving average, like $\overline{z}_{t+1} \leftarrow \beta\overline{z}_t + (1-\beta)z_t$.

The MD_η iteration in Eq.5 also needs to be taken into a more tractable form. By recursively applying, we can get the final results of MD_η with density function of μ and ν through T iterations:

$$\mathrm{d}z_+ = \frac{e^{-\eta h}\mathrm{d}z}{\int e^{-\eta h}\mathrm{d}z} \quad \Rightarrow \quad \mathrm{d}\mu_T = \frac{e^{-\sum_{k=1}^{T} h_k^{\mu}}\mathrm{d}\mu}{\int e^{-\sum_{k=1}^{T} h_k^{\mu}}\mathrm{d}\mu}, \quad \mathrm{d}\nu_T = \frac{e^{-\sum_{k=1}^{T} h_k^{\nu}}\mathrm{d}\nu}{\int e^{-\sum_{k=1}^{T} h_k^{\nu}}\mathrm{d}\nu}. \tag{11}$$

The Stochastic Gradient Langevin Dynamics(Welling & Teh, 2011) is a standard sampling algorithm, and its iterative function for any probability distribution with density function $e^{-h}\mathrm{d}z$ is:

$$\boldsymbol{z}_{t+1} = \boldsymbol{z}_t - \gamma\hat{\nabla}h(\boldsymbol{z}_t) + \sqrt{2\gamma}\epsilon\boldsymbol{\xi}, \tag{12}$$

where γ is the sampling step size, $\hat{\nabla}h$ is the unbiased estimator of ∇h, ϵ is the thermal noise, and $\boldsymbol{\xi} \sim \mathcal{N}(0,1)$ is a standard normal vector. By combining Eq.10, Eq.11 and Eq.12, the following iterative functions express how the model parameters $\boldsymbol{\theta}$ and adversarial perturbations $\boldsymbol{\delta}$ are updated:

$$\begin{aligned}
\boldsymbol{\delta}_t^{(k+1)} &= \boldsymbol{\delta}_t^{(k)} - \gamma_t\nabla_{\boldsymbol{\delta}}\frac{1}{n}\sum_{i=1}^{n} l\left(f_{\boldsymbol{\theta}_t}\left(\boldsymbol{x}_i + \boldsymbol{\delta}_t^{(k)}\right), f_{\boldsymbol{\theta}_t}\left(\boldsymbol{x}_i\right)\right) + \sqrt{2\gamma_t}\epsilon\boldsymbol{\xi}, \\
\boldsymbol{\theta}_t^{(k+1)} &= \boldsymbol{\theta}_t^{(k)} - \gamma_t\nabla_{\boldsymbol{\theta}}\frac{1}{n}\sum_{i=1}^{n}\left[L\left(f_{\boldsymbol{\theta}_t^{(k)}}\left(\boldsymbol{x}_i\right), y_i\right) + \lambda l\left(f_{\boldsymbol{\theta}_t^{(k)}}\left(\boldsymbol{x}_i + \boldsymbol{\delta}_t\right), f_{\boldsymbol{\theta}_t^{(k)}}\left(\boldsymbol{x}_i\right)\right)\right] + \sqrt{2\gamma_t}\epsilon\boldsymbol{\xi}.
\end{aligned} \tag{13}$$

For convenience, we summarize the above method in Algorithm 2, which is named MAT (**M**ixed-strategy **A**dversarial **T**raining). Furthermore, to express our algorithm more intuitively, we draw a schematic to reveal the entire iterative process of MAT, as shown in Figure 2.

Algorithm 2: MAT: Mixed-strategy Adversarial Training

Input: Pre-trained model parameters: $\boldsymbol{\theta}_0$, sampling step size: $\{\gamma_t\}_{t=0}^{T-1}$, sampling times: K, thermal noise: ϵ, coefficient of exponential moving average: β.

for $t = 0, 1, \ldots, T-1$ **do**
  init $\boldsymbol{\delta}_t^{(0)}$, $\quad\overline{\boldsymbol{\delta}}_t \leftarrow \boldsymbol{\delta}_t^{(0)}$
  for $k = 0, 1, \ldots, K-1$ **do**
    $\boldsymbol{\delta}_t^{(k+1)} \leftarrow \boldsymbol{\delta}_t^{(k)} - \gamma_t\nabla_{\boldsymbol{\delta}}\frac{1}{n}\sum_{i=1}^{n} l(f_{\boldsymbol{\theta}_t}(\boldsymbol{x}_i + \boldsymbol{\delta}_t^{(k)}), f_{\boldsymbol{\theta}_t}(\boldsymbol{x}_i)) + \sqrt{2\gamma_t}\epsilon\boldsymbol{\xi}$
    $\overline{\boldsymbol{\delta}}_t \leftarrow \beta\overline{\boldsymbol{\delta}}_t + (1-\beta)\boldsymbol{\delta}_t^{(k+1)}$
  end
  $\boldsymbol{\theta}_t^{(0)} \leftarrow \boldsymbol{\theta}_t, \quad \overline{\boldsymbol{\theta}}_t \leftarrow \boldsymbol{\theta}_t$
  for $k = 0, 1, \ldots, K-1$ **do**
    $\boldsymbol{\theta}_t^{(k+1)} \leftarrow$
    $\boldsymbol{\theta}_t^{(k)} - \gamma_t\nabla_{\boldsymbol{\theta}}\frac{1}{n}\sum_{i=1}^{n}\left[L(f_{\boldsymbol{\theta}_t^{(k)}}(\boldsymbol{x}_i), y_i) + \lambda l(f_{\boldsymbol{\theta}_t^{(k)}}(\boldsymbol{x}_i + \overline{\boldsymbol{\delta}}_t), f_{\boldsymbol{\theta}_t^{(k)}}(\boldsymbol{x}_i))\right] + \sqrt{2\gamma_t}\epsilon\boldsymbol{\xi}$
    $\overline{\boldsymbol{\theta}}_t \leftarrow \beta\overline{\boldsymbol{\theta}}_t + (1-\beta)\boldsymbol{\theta}_t^{(k+1)}$
  end
  $\boldsymbol{\theta}_{t+1} \leftarrow \beta\boldsymbol{\theta}_t + (1-\beta)\overline{\boldsymbol{\theta}}_t$
end

Output: Fine-tuned model parameters: $\boldsymbol{\theta}_T$.

4 EXPERIMENTS

In order to verify the effectiveness of our proposed algorithm, we comprehensively evaluate MAT with two Natural Language Understanding benchmarks, namely GLUE (Wang et al., 2019) and ANLI (Nie et al., 2020). The primary purpose is to analyze the generalization and robustness improvement of the pre-trained model after fine-tuning with MAT.

4.1 DATASETS INTRODUCTION

GLUE Benchmark. The General Language Understanding Evaluation (GLUE) benchmark (Wang et al., 2019) is a collection of nine tasks for natural language understanding system training, evaluation, and analysis. GLUE has nine datasets which are CoLA (Warstadt et al., 2019), SST-2 (Socher et al., 2013), MRPC (Dolan & Brockett, 2005), STS-B (Agirre et al., 2007), QQP (Iyer et al., 2017),

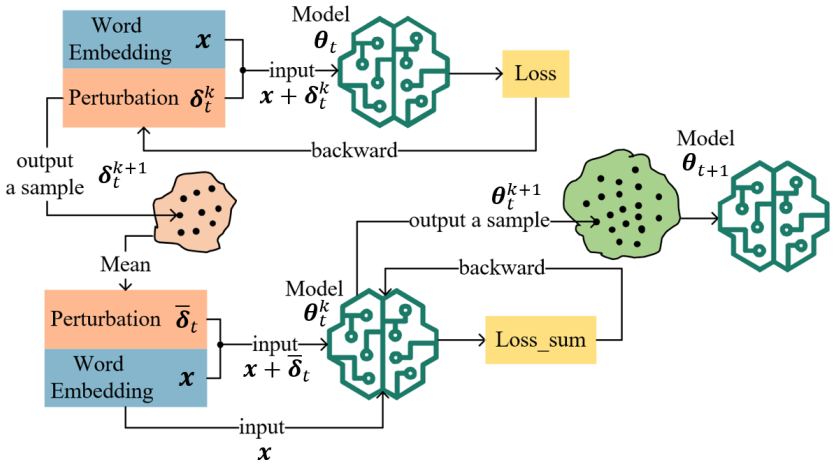

Figure 2: Schematic diagram of MAT algorithm.

MNLI (Williams et al., 2018), QNLI (Rajpurkar et al., 2016), RTE (Dagan et al., 2005; Bar Haim et al., 2006; Giampiccolo et al., 2007; Bentivogli et al., 2009), WNLI (Levesque, 2011). Except that STS-B is a regression task, all the other tasks fall within the classification tasks.

ANLI Benchmark. The Adversarial Natural Language Inference (ANLI) (Nie et al., 2020) is a new large-scale NLI benchmark dataset, collected via an iterative, adversarial human-and-model-in-the-loop procedure. There are three parts in the dataset, and their difficulty gradually increases for deep language models. ANLI is used to measure the robustness of the model, that is, the performance of the model in the face of adversarial attacks. Therefore, the improvement of model robustness will also be one of the keys for us in evaluating the MAT algorithm.

4.2 Experiment Setting

All our experimental code is based on the PyTorch (Paszke et al., 2019) framework. As for pre-trained models, we choose the BERT-base[1] and the RoBERTa-large[2], with parameter sizes of 110M and 340M respectively, which checkpoints are from the Huggingface repository. The open source libraries of Transformers (Wolf et al., 2020) and Datasets (Lhoest et al., 2021) are used for loading pre-trained models and preprocessing datasets. Since the MAT algorithm is based on a mixed strategy, we use the Stochastic Gradient Langevin Dynamics sampling to update the model parameters instead of any existing optimizers. The hyper-parameter search is based on experience and NNI[3], an AutoML library, where using the TPE algorithm for parameter searching.

Adversarial training is more time-consuming than vanilla fine-tuning training, and our algorithm further complicates parameter updating process. Therefore we abandoned training MNLI (Williams et al., 2018) and QQP (Iyer et al., 2017) on the GLUE benchmark, because their size is too big. Nonetheless, it will not affect our final experimental conclusion, since the results of the rest datasets have shown excellent performance with the MAT algorithm. For the ANLI benchmark, we follow the experimental setup of Nie et al. (2020) and Jiang et al. (2020), training RoBERTa-lager on the combined NLI datasets (MNLI (Williams et al., 2018) + SNLI (Bowman et al., 2015) + FEVER (Thorne et al., 2018) + ANLI (Nie et al., 2020)). And we employ the checkpoints[4] published by Nie et al. (2020), then continue training the model on ANLI with the MAT algorithm.

[1]https://huggingface.co/bert-base-uncased/tree/main
[2]https://huggingface.co/roberta-large/tree/main
[3]https://github.com/microsoft/nni
[4]https://huggingface.co/ynie/roberta-large-snli_mnli_fever_anli_R1_R2_R3-nli/tree/main

Table 1: Results of GLUE based on the BERT-base model

Dataset	**CoLA**	**SST-2**	**MRPC**	**STS-B**	**QNLI**	**RTE**
Size	8.6k	67.3k	3.7k	5.7k	104.7k	2.5k
Metric	Mcc	Acc	Acc/F1	P/SCorr	Acc	Acc
BERT (Devlin et al., 2019)	-	92.7	86.7/-	-	88.4	-
BERT (Jiang et al., 2020)	54.7	92.9	84.1/89.0	89.2/88.8	91.1	63.5
+SMART (Jiang et al., 2020)	59.1	93.0	87.7/91.3	90.0/89.4	91.7	71.2
+MAT (Ours)	**61.3**	**93.1**	**89.0/92.1**	**90.2/89.9**	**91.8**	**72.6**

Table 2: Results of GLUE based on the RoBERTa-large model

Dataset	**CoLA**	**SST-2**	**MRPC**	**STS-B**	**QNLI**	**RTE**
Size	8.6k	67.3k	3.7k	5.7k	104.7k	2.5k
Metric	Mcc	Acc	Acc/F1	P/SCorr	Acc	Acc
RoBERTa (Liu et al., 2019)	68.0	96.4	90.9/-	92.4/-	94.7	86.6
+FreeLB (Zhu et al., 2020)	71.1	96.7	91.4/-	92.7/-	95.0	88.1
+SMART (Jiang et al., 2020)	70.6	96.9	89.2/92.1	92.8/92.6	95.6	**92.0**
+R3F (Aghajanyan et al., 2021)	71.2	**97.0**	91.6/-	-/-	95.3	88.5
+MAT (Ours)	**71.3**	**97.0**	**91.7/93.9**	**92.9/92.6**	**95.7**	90.6

4.3 Generalization Evaluation Experimental Results

In order to assess the improvement of the model generalization performance after fine-tuning with MAT, we compare the experimental results of MAT on the GLUE benchmark with a series of previous works, including FreeLB (Zhu et al., 2020), SMART (Jiang et al., 2020), and R3F (Aghajanyan et al., 2021). Moreover, since the MAT algorithm borrows the objective function in SMART as a starting point, SMART naturally become our main comparison object. Further clarification is that MAT only selects the part of the source objective function in SMART, which may degrade performance, as shown in the ablation study of their experiment. Nonetheless, the MAT algorithm not only compensates for the performance loss caused by the incomplete objective function, but also achieves higher performance than the results of SMART on various datasets.

The experimental results of fine-tuning the BERT-base model with the MAT algorithm are organized in Table 1. We compare the results of MAT with vanilla fine-tuning and SMART, and the MAT algorithm outperforms previous work on all datasets. It demonstrates the benefit of adversarial training and highlights that our mixed-strategy adversarial training further improves the generalization performance of the model. In comparison to SMART, MAT performs significantly higher on CoLA (**61.3** vs 59.1), MRPC (**89.0/92.1** vs 87.7/91.3), and RTE (**72.6** vs 71.2), and also marginally better on SST-2 (**93.1** vs 93.0), STS-B (**90.2/89.9** vs 90.0/89.4), and QNLI (**91.8** vs 91.7).

In Table 2, we summarized the experimental results of the MAT algorithm on fine-tuning the RoBERTa-larger model. Compared with the vanilla fine-tuning, FreeLB, SMART, and R3F, the experimental results of MAT reach SOTA on five datasets. Precisely speaking, when the results are compared with SMART, MAT has a clear lead on CoLA (**71.3** vs 70.6) and MRPC (**91.7/93.9** vs 89.2/92.1), and a slight improvement on SST-2 (**97.0** vs 96.9), STS-B (**92.9/92.6** vs 92.8/92.6), and QNLI (**95.7** vs 95.6), only lagging behind on RTE (90.6 vs **92.0**).[5]

4.4 Robustness Evaluation Experimental Results

When deploying models, the generalization of the model is undoubtedly essential, but robustness is also receiving more and more attention. Therefore, we further evaluate the robustness gained from

[5]The abbreviations of the metric in the Table 1 and Table 2 are as follows: Mcc (Matthews correlation coefficient), Acc (Accuracy), F1 (F1 Score), P/SCorr (Pearson and Spearman Correlation coefficient).

Table 3: Results of ANLI based on the RoBERTa-large model

Combined Training Datasets	**MNLI + SNLI + FEVER + ANLI**					
Evaluation Dataset	**ANLI-Dev**			**ANLI-Test**		
ANLI Part	**R1**	**R2**	**R3**	**R1**	**R2**	**R3**
RoBERTa (Nie et al., 2020)	73.8	48.9	44.4	-	-	-
+SMART (Jiang et al., 2020)	74.5	50.9	47.6	72.4	49.8	**50.3**
+MAT (Ours)	**74.8**	**51.0**	**49.3**	**74.7**	**51.1**	49.5

the MAT algorithm, that is, the performance of the model in the face of the ANLI benchmark. For the convenience of comparison, we use the same experimental setup as SMART and train the RoBERTa-Large model with the combined NLI dataset (MNLI (Williams et al., 2018) + SNLI (Bowman et al., 2015) + FEVER (Thorne et al., 2018) + ANLI (Nie et al., 2020)).

The experimental results are compared on the development set and test set of ANLI, as shown in Table 3. We choose vanilla fine-tuning and SMART as the baseline, and the comparison result shows that MAT is ahead in almost all parts of ANLI, indicating that MAT can bring better robustness to the model. In contrast to SMART, MAT has an advantage on dev sets, such as the R1 (**74.8** vs 74.5), R2 (**51.0** vs 50.9), and R3 (**48.7** vs 47.6). For test sets, MAT is obviously leading on the R1 (**74.7** vs 72.4), R2 (**51.1** vs 49.8), and slightly behind on the R3 (49.5 vs **50.3**).

5 Conclusion

This work introduces a mixed-strategy game into adversarial training for fine-tuning pre-trained models. We deduced the Nash equilibrium with a traditional game theory method, the Entropic Mirror Descent for Adversarial Training to solve the game. Furthermore, we simplified this method into a practical algorithm named MAT. We evaluate the MAT algorithm on multiple benchmarks with pre-trained models, comparing it with previous work. The experimental results proved that MAT leads to better performance of model generalization and robustness, which also provide solid practical support for the research of introducing game theory into adversarial training. However, our algorithm needs to use sampling to explore the distributions of model parameters and adversarial perturbations, which requires a significant amount of computational overhead, resulting taking a significantly longer training time than the pure-strategy algorithm. We believe there will be more solutions to solve the mixed-strategy adversarial training efficiently in the future.

References

Armen Aghajanyan, Akshat Shrivastava, Anchit Gupta, Naman Goyal, Luke Zettlemoyer, and Sonal Gupta. Better fine-tuning by reducing representational collapse. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=OQ08SN70M1V.

Eneko Agirre, Lluís Màrquez i Villodre, and Richard Wicentowski (eds.). *Proceedings of the 4th International Workshop on Semantic Evaluations, SemEval@ACL 2007, Prague, Czech Republic, June 23-24, 2007*, 2007. The Association for Computer Linguistics. URL https://aclanthology.org/volumes/S07-1/.

Kartik Ahuja, Karthikeyan Shanmugam, Kush R. Varshney, and Amit Dhurandhar. Invariant risk minimization games. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pp. 145–155. PMLR, 2020. URL http://proceedings.mlr.press/v119/ahuja20a.html.

Sanjeev Arora, Rong Ge, Yingyu Liang, Tengyu Ma, and Yi Zhang. Generalization and equilibrium in generative adversarial nets (gans). In Doina Precup and Yee Whye Teh (eds.), *Proceedings of*

the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017, volume 70 of *Proceedings of Machine Learning Research*, pp. 224–232. PMLR, 2017. URL http://proceedings.mlr.press/v70/arora17a.html.

Roy Bar Haim, Ido Dagan, Bill Dolan, Lisa Ferro, Danilo Giampiccolo, Bernardo Magnini, and Idan Szpektor. The second PASCAL recognising textual entailment challenge. *Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*, 01 2006.

Amir Beck and Marc Teboulle. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Oper. Res. Lett.*, 31(3):167–175, 2003. doi: 10.1016/S0167-6377(02)00231-6. URL https://doi.org/10.1016/S0167-6377(02)00231-6.

Luisa Bentivogli, Bernardo Magnini, Ido Dagan, Hoa Trang Dang, and Danilo Giampiccolo. The fifth PASCAL recognizing textual entailment challenge. In *Proceedings of the Second Text Analysis Conference, TAC 2009, Gaithersburg, Maryland, USA, November 16-17, 2009*. NIST, 2009. URL https://tac.nist.gov/publications/2009/additional.papers/RTE5_overview.proceedings.pdf.

Ken Binmore et al. *Playing for real: a text on game theory*. Oxford university press, 2007.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. A large annotated corpus for learning natural language inference. In Lluís Màrquez, Chris Callison-Burch, Jian Su, Daniele Pighin, and Yuval Marton (eds.), *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pp. 632–642. The Association for Computational Linguistics, 2015. doi: 10.18653/v1/d15-1075. URL https://doi.org/10.18653/v1/d15-1075.

Ido Dagan, Oren Glickman, and Bernardo Magnini. The PASCAL recognising textual entailment challenge. In Joaquin Quiñonero Candela, Ido Dagan, Bernardo Magnini, and Florence d'Alché-Buc (eds.), *Machine Learning Challenges, Evaluating Predictive Uncertainty, Visual Object Classification and Recognizing Textual Entailment, First PASCAL Machine Learning Challenges Workshop, MLCW 2005, Southampton, UK, April 11-13, 2005, Revised Selected Papers*, volume 3944 of *Lecture Notes in Computer Science*, pp. 177–190. Springer, 2005. doi: 10.1007/11736790_9. URL https://doi.org/10.1007/11736790_9.

Constantinos Daskalakis, Andrew Ilyas, Vasilis Syrgkanis, and Haoyang Zeng. Training gans with optimism. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL https://openreview.net/forum?id=SJJySbbAZ.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pp. 4171–4186. Association for Computational Linguistics, 2019. doi: 10.18653/v1/n19-1423. URL https://doi.org/10.18653/v1/n19-1423.

William B. Dolan and Chris Brockett. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing, IWP@IJCNLP 2005, Jeju Island, Korea, October 2005, 2005*. Asian Federation of Natural Language Processing, 2005. URL https://aclanthology.org/I05-5002/.

Danilo Giampiccolo, Bernardo Magnini, Ido Dagan, and Bill Dolan. The third PASCAL recognizing textual entailment challenge. In Satoshi Sekine, Kentaro Inui, Ido Dagan, Bill Dolan, Danilo Giampiccolo, and Bernardo Magnini (eds.), *Proceedings of the ACL-PASCAL@ACL 2007 Workshop on Textual Entailment and Paraphrasing, Prague, Czech Republic, June 28-29, 2007*, pp. 1–9. Association for Computational Linguistics, 2007. URL https://aclanthology.org/W07-1401/.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In Yoshua Bengio and Yann LeCun (eds.), *3rd International Conference on Learning*

Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings, 2015. URL http://arxiv.org/abs/1412.6572.

Ya-Ping Hsieh, Chen Liu, and Volkan Cevher. Finding mixed nash equilibria of generative adversarial networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2810–2819. PMLR, 2019. URL http://proceedings.mlr.press/v97/hsieh19b.html.

Shankar Iyer, Nikhil Dandekar, and Kornel Csernai. First quora dataset release: Question pairs. Technical report, Quora, 2017. URL https://data.quora.com/First-Quora-Dataset-Release-Question-Pairs.

Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Tuo Zhao. SMART: robust and efficient fine-tuning for pre-trained natural language models through principled regularized optimization. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel R. Tetreault (eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pp. 2177–2190. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.acl-main.197. URL https://doi.org/10.18653/v1/2020.acl-main.197.

Hector J. Levesque. The winograd schema challenge. In *Logical Formalizations of Commonsense Reasoning, Papers from the 2011 AAAI Spring Symposium, Technical Report SS-11-06, Stanford, California, USA, March 21-23, 2011*. AAAI, 2011. URL http://www.aaai.org/ocs/index.php/SSS/SSS11/paper/view/2502.

Quentin Lhoest, Albert Villanova del Moral, Yacine Jernite, Abhishek Thakur, Patrick von Platen, Suraj Patil, Julien Chaumond, Mariama Drame, Julien Plu, Lewis Tunstall, Joe Davison, Mario Sasko, Gunjan Chhablani, Bhavitvya Malik, Simon Brandeis, Teven Le Scao, Victor Sanh, Canwen Xu, Nicolas Patry, Angelina McMillan-Major, Philipp Schmid, Sylvain Gugger, Clément Delangue, Théo Matussière, Lysandre Debut, Stas Bekman, Pierric Cistac, Thibault Goehringer, Victor Mustar, François Lagunas, Alexander M. Rush, and Thomas Wolf. Datasets: A community library for natural language processing. In Heike Adel and Shuming Shi (eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, EMNLP 2021, Online and Punta Cana, Dominican Republic, 7-11 November, 2021*, pp. 175–184. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.emnlp-demo.21. URL https://doi.org/10.18653/v1/2021.emnlp-demo.21.

Xiaodong Liu, Hao Cheng, Pengcheng He, Weizhu Chen, Yu Wang, Hoifung Poon, and Jianfeng Gao. Adversarial training for large neural language models. *CoRR*, abs/2004.08994, 2020. URL https://arxiv.org/abs/2004.08994.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692, 2019. URL http://arxiv.org/abs/1907.11692.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL https://openreview.net/forum?id=rJzIBfZAb.

Takeru Miyato, Andrew M. Dai, and Ian J. Goodfellow. Adversarial training methods for semi-supervised text classification. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL https://openreview.net/forum?id=r1X3g2_xl.

John Nash. Non-cooperative games. *Annals of Mathematics*, 54(2):286–295, 1951. ISSN 0003486X. URL http://www.jstor.org/stable/1969529.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. Adversarial NLI: A new benchmark for natural language understanding. In Dan Jurafsky, Joyce Chai,

Natalie Schluter, and Joel R. Tetreault (eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pp. 4885–4901. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.acl-main.441. URL https://doi.org/10.18653/v1/2020.acl-main.441.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Z. Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 8024–8035, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/bdbca288fee7f92f2bfa9f7012727740-Abstract.html.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. *OpenAI blog*, 2018. URL https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67, 2020. URL http://jmlr.org/papers/v21/20-074.html.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100, 000+ questions for machine comprehension of text. In Jian Su, Xavier Carreras, and Kevin Duh (eds.), *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pp. 2383–2392. The Association for Computational Linguistics, 2016. doi: 10.18653/v1/d16-1264. URL https://doi.org/10.18653/v1/d16-1264.

Ali Shafahi, Mahyar Najibi, Amin Ghiasi, Zheng Xu, John P. Dickerson, Christoph Studer, Larry S. Davis, Gavin Taylor, and Tom Goldstein. Adversarial training for free! In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 3353–3364, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/7503cfacd12053d309b6bed5c89de212-Abstract.html.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*, pp. 1631–1642. ACL, 2013. URL https://aclanthology.org/D13-1170/.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In Yoshua Bengio and Yann LeCun (eds.), *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014. URL http://arxiv.org/abs/1312.6199.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. FEVER: a large-scale dataset for fact extraction and verification. In Marilyn A. Walker, Heng Ji, and Amanda Stent (eds.), *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pp. 809–819. Association for Computational Linguistics, 2018. doi: 10.18653/v1/n18-1074. URL https://doi.org/10.18653/v1/n18-1074.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=rJ4km2R5t7.

Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. Neural network acceptability judgments. *Trans. Assoc. Comput. Linguistics*, 7:625–641, 2019. doi: 10.1162/tacl_a_00290. URL https://doi.org/10.1162/tacl_a_00290.

Max Welling and Yee Whye Teh. Bayesian learning via stochastic gradient langevin dynamics. In Lise Getoor and Tobias Scheffer (eds.), *Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28 - July 2, 2011*, pp. 681–688. Omnipress, 2011. URL https://icml.cc/2011/papers/398_icmlpaper.pdf.

Adina Williams, Nikita Nangia, and Samuel R. Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In Marilyn A. Walker, Heng Ji, and Amanda Stent (eds.), *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pp. 1112–1122. Association for Computational Linguistics, 2018. doi: 10.18653/v1/n18-1101. URL https://doi.org/10.18653/v1/n18-1101.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-art natural language processing. In Qun Liu and David Schlangen (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, EMNLP 2020 - Demos, Online, November 16-20, 2020*, pp. 38–45. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.emnlp-demos.6. URL https://doi.org/10.18653/v1/2020.emnlp-demos.6.

Dinghuai Zhang, Tianyuan Zhang, Yiping Lu, Zhanxing Zhu, and Bin Dong. You only propagate once: Accelerating adversarial training via maximal principle. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 227–238, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/812b4ba287f5ee0bc9d43bbf5bbe87fb-Abstract.html.

Chen Zhu, Yu Cheng, Zhe Gan, Siqi Sun, Tom Goldstein, and Jingjing Liu. Freelb: Enhanced adversarial training for natural language understanding. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL https://openreview.net/forum?id=BygzbyHFvB.

Simiao Zuo, Chen Liang, Haoming Jiang, Xiaodong Liu, Pengcheng He, Jianfeng Gao, Weizhu Chen, and Tuo Zhao. Adversarial training as stackelberg game: An unrolled optimization approach. *CoRR*, abs/2104.04886, 2021. URL https://arxiv.org/abs/2104.04886.