

MAT: Mixed-Strategy Game of Adversarial Training in Fine-tuning

Zhehua Zhong¹, Tianyi Chen² and Zhen Wang^{1*}

¹ School of Cyberspace, Hangzhou Dianzi University

² Microsoft

zhongzhehua@outlook.com, tiachen@microsoft.com, wangzhen@hdu.edu.cn

Abstract

Fine-tuning large-scale pre-trained language models has been demonstrated effective for various natural language processing (NLP) tasks. Previous studies have established that incorporating adversarial training during the fine-tuning stage can significantly enhance model generalization and robustness. However, from the perspective of game theory, such utilizations of adversarial training correspond to pure-strategy games, which are inherently limited in terms of the scope of their strategies, thereby still having room for improvement. In order to push the performance boundaries, we propose a novel **Mixed-strategy Adversarial Training** algorithm (MAT). Methodologically, we derive the Nash equilibrium of a mixed-strategy game for adversarial training using Entropy Mirror Descent to establish MAT by sampling method. To verify the effectiveness of MAT, we conducted extensive benchmark experiments on large-scale pre-trained models, such as BERT and RoBERTa. MAT significantly outperforms the state-of-the-art methods on both the GLUE and ANLI benchmarks in terms of generalization and robustness.

1 Introduction

Recent years have seen significant advancements in the field of natural language processing (NLP) due to the development of large-scale pre-trained language models such as BERT [Devlin *et al.*, 2019], GPT [Radford *et al.*, 2018; Radford *et al.*, 2019; Brown *et al.*, 2020], and T5 [Raffel *et al.*, 2020]. These models have achieved remarkable success across a wide range of NLP tasks, primarily because of their vast number of parameters and the ability to leverage context information. However, the complexity of these models has presented challenges in model development, training, and usage. As we know, fine-tuning is an increasingly popular and fundamental approach, which adapts a pre-trained model to a downstream task through re-training on a small set of task-specific data. Meanwhile, it has been observed

that fine-tuned models are susceptible to overfitting the training data of downstream tasks, which may result in suboptimal generalization performance on unseen data points.

Recent studies [Liu *et al.*, 2020; Zhu *et al.*, 2020; Jiang *et al.*, 2020; Aghajanyan *et al.*, 2021] have demonstrated that incorporating adversarial training during the fine-tuning stage can alleviate issues of overfitting and effectively improve model generalization in downstream applications. Adversarial training optimizes models by augmenting the training set with adversarial examples, which are typically generated by adding small and carefully selected perturbations to input data. However, from the perspective of game theory, the existing adversarial training methods can be mainly interpreted as pure-strategy games between the model and adversarial perturbations, which are limited in their strategy space. According to the Nash theorem [Nash, 1951], the pure-strategy game is a subset of the mixed-strategy game, and the Nash equilibrium (which represents a stable state of the game) in the pure-strategy game may not exist. Therefore, it is natural to question *whether performance can be further improved by interpreting and reformulating adversarial training in light of the theory of the mixed-strategy game*.

In order to address the question, we view adversarial training as a two-player, complete-information game and propose an approach to improve adversarial training under the setting of a mixed-strategy game. Previous adversarial training methods can be mainly seen as pure-strategy games because both model parameters and adversarial perturbations choose deterministically specified strategies. In contrast, our proposed adversarial training method is based on the mixed strategy, which enables a broader search space for model parameters by leveraging the probabilistic nature of the strategies (as discussed in greater detail in Section 3.1). To illustrate this concept, we can consider a two-player game of rock-paper-scissors, as shown in Figure 1. In the mixed-strategy game, the players have a broader range of strategies to choose from.

Our main contributions can be summarized as follows:

- We propose a **Mixed-strategy Adversarial Training** algorithm (MAT) that aims to further improve the generalization and robustness of fine-tuned NLP models. Specifically, we apply game theory to interpret adversarial training as a mixed-strategy game, and derive a theoretical Nash equilibrium using the Entropy Mirror Descent method. Then we simplify this method through

*Corresponding author.

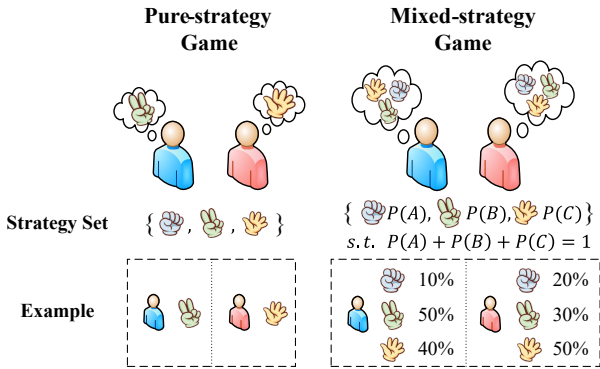


Figure 1: Difference between pure strategy and mixed strategy.

the use of sampling mechanisms to make the training algorithm practical and implementable in reality.

- We perform thorough experiments to demonstrate the effectiveness of MAT and attain state-of-the-art results on the GLUE [Wang *et al.*, 2019] and ANLI [Nie *et al.*, 2020] benchmarks. Our findings show that applying MAT on the BERT [Devlin *et al.*, 2019] and RoBERTa model [Liu *et al.*, 2019] results in significantly superior performance compared to previous methods.

2 Related Work

2.1 Adversarial Training in CV and NLP

Adversarial training was first introduced in the field of computer vision (CV) by Szegedy *et al.*[2014] and Goodfellow *et al.*[2015]. Instead of relying solely on clean datasets for training, researchers discovered that augmenting the training set with adversarial examples can effectively protect models against adversarial attacks. Madry *et al.*[2018] formulated this type of adversarial training as a Min-Max optimization problem, where the adversarial examples are generated by maximizing the deviation of the training objective within the tolerance of visibility. For solving the maximization sub-problems, the projected gradient descent (PGD) algorithm is employed to generate perturbations that satisfy the given constraints. Subsequent research by Shafahi *et al.*[2019] and Zhang *et al.*[2019] have aimed to improve the efficiency of the PGD algorithm by reducing its time complexity.

Adversarial training in natural language processing (NLP) was introduced by Miyato *et al.*[2017], who perturbed the word embedding space to alleviate the issue of overfitting during training models from scratch. With the advent of pre-trained large-scale models, Zhu *et al.*[2020] extended adversarial training to the fine-tuning stage of pre-trained models and proposed the FreeLB algorithm. FreeLB adds adversarial perturbations to word embeddings and minimizes the resultant adversarial risk inside different regions around input samples. Jiang *et al.*[2020] developed the SMART algorithm, which effectively addresses overfitting and aggressive updating issues in the fine-tuning stage through smooth-inducing adversarial regularization and Bregman proximal point optimization. Aghajanyan *et al.*[2021] further improved upon

this by decomposing pre-trained models into a feature extractor and a classifier with separate constraints and modifying adversarial perturbations to be either normal or uniformly distributed noise, significantly reducing the time required to generate adversarial perturbations.

2.2 Game Theory Applications in Deep Learning

Generative adversarial networks (GANs) [Goodfellow *et al.*, 2014] generate synthetic data by mimicking genuine samples, which have become increasingly popular and widely used in various applications. GANs can be primarily interpreted as a game theory problem, which has inspired several training approaches in recent years. Arora *et al.*[2017] proposed the MIX+GAN protocol, a mixed-strategy GAN training algorithm that uses a small mixture of discriminators and generators. Daskalakis *et al.*[2018] introduced the Optimistic Mirror Descent (OMD) method for training Wasserstein GANs, and they formally showed that the last iteration of the OMD dynamics converged to a Nash equilibrium in the case of a zero-sum game. Hsieh *et al.*[2019] optimized the set of probability distributions over the pure strategies of neural networks and suggested a provably convergent GAN training approach. Ahuja *et al.*[2020] formulated the standard risk minimization paradigm of machine learning in the context of finding the Nash equilibrium of an ensemble game among several environments. Although there is limited research on applying game theory to fine-tuning pre-trained models, Zuo *et al.*[2021] proposed the SALT approach, which formulates adversarial training as a Stackelberg game in which the follower generates perturbations and the leader optimizes model parameters given these perturbations.

3 Mixed-Strategy Adversarial Training

In this section, we examine adversarial training from a game-theoretic perspective. Specifically, we first generalize the mixed-strategy form of adversarial training in Section 3.1. Then, in Section 3.2, we utilize the Entropy Mirror Descent optimization algorithm to derive a Nash equilibrium for the mixed-strategy adversarial training game. Since the probability distribution of the strategy set is implicitly unknown during model training, we propose a sampling-based simplification of the theoretical algorithm in Section 3.3, resulting in the development of our practical mixed-strategy adversarial training algorithm, referred to as MAT.

Notation. Throughout the paper, we use $f(\theta, \mathbf{x})$ to represent the output of the language model f with parameters θ , given input word embeddings \mathbf{x} . δ represents adversarial perturbations. $\mathcal{D}_{KL}(P||Q) = \sum_k p_k \log(p_k/q_k)$ represents the KL-divergence of two discrete distributions P and Q .

3.1 Defining Adversarial Training Game

Without loss of generality, we define the loss function $L(\theta)$ for the standard NLP fine-tuning task as follows:

$$L(\theta) := \ell(f(\theta, \mathbf{x}), y), \tag{1}$$

where y is the ground truth, and ℓ measures the deviation between the predicted output $f(\theta, \mathbf{x})$ and ground truth y . Moreover, the loss function of adversarial training is defined as:

$$R(\theta, \delta) = \ell_s(f(\theta, \mathbf{x}), f(\theta, \mathbf{x} + \delta)), \tag{2}$$

where ℓ_s measures the deviation brought by adding adversarial perturbations onto the model inputs, which is chosen according to different tasks. For classification tasks, ℓ_s is typically chosen as the symmetrized KL-divergence, *i.e.*, $\ell_s(P, Q) = \mathcal{D}_{KL}(P\|Q) + \mathcal{D}_{KL}(Q\|P)$ given two distributions P and Q . For regression tasks, ℓ_s is chosen as the squared loss. Then, by combining Equation (1) and Equation (2), the target optimization problem for the fine-tuning tasks with adversarial training can be formulated as below:

$$\min_{\theta \in \Theta} \max_{\delta \in \Delta} [L(\theta) + \lambda R(\theta, \delta)], \quad (3)$$

where λ is a tuning coefficient that balances the two individual loss functions $L(\theta)$ and $R(\theta, \delta)$.

Equation (3) can be interpreted as a potential game problem involving two parties: one for the model trainable parameters and the other one for the adversarial perturbations. The feasible domain of the model parameters can be considered as model strategy space, similar to the adversarial perturbations. Adversarial training can be solely viewed as a game where the model parameters select a strategy from the strategy set Θ , while the adversarial perturbations select a strategy from the strategy set Δ . Clearly, this is a pure-strategy game in which the decisions of both parties are deterministic and non-probabilistic. Existing adversarial training methods are limited to pure-strategy games with a limited strategy space, thus, we aim to extend it to a mixed-strategy game.

By using the mixed strategy, we no longer consider specific values of the model parameters and adversarial perturbations, but instead the probability of each possibility. In practice, we consider the set of all probability distributions over Θ and Δ . Let $\mathcal{M}(\Theta)$ and $\mathcal{M}(\Delta)$ denote the set of all Borel probability measures on Θ and Δ , respectively. Then we define the following target problem for a mixed-strategy adversarial training game to find the optimal strategy probability distribution $\mu \in \mathcal{M}(\Theta)$ and $\nu \in \mathcal{M}(\Delta)$:

$$\min_{\mu \in \mathcal{M}(\Theta)} \mathbb{E}_{\theta \sim \mu} \max_{\nu \in \mathcal{M}(\Delta)} \mathbb{E}_{\delta \sim \nu} [L(\theta) + \lambda R(\theta, \delta)]. \quad (4)$$

By redefining adversarial training as a mixed-strategy game, we expand the strategy space, enabling the parties to find the optimal distribution for both sides. More strategy options allow for greater flexibility and robustness in the adversarial training process. The game can be solved when a Nash equilibrium is reached, in which no party can benefit by changing its strategy given the strategy of the other party.

3.2 Finding Nash Equilibrium of Mixed-Strategy Adversarial Training Game

In order to determine the Nash equilibrium of the mixed-strategy adversarial training game defined in Equation (4), we turn to iterative optimization methods as a natural selection for solving such problems. These methods involve iteratively searching for the strategies of the players until an approximate Nash equilibrium is reached. In this work, we utilize the Entropy Mirror Descent (EMD) [Beck and Teboulle, 2003], which is a specific form of the Mirror Descent (MD) [Nemirovski and Yudin, 1983]. EMD uses the entropy function as a measure of distance between probability distributions, known as the Bregman divergence. This measure is employed

to guide the iterative updating of solutions, allowing us to find the Nash equilibrium of mixed-strategy games.

To proceed, we iteratively update the probability distribution of model parameters and adversarial perturbations by EMD, until an approximate Nash equilibrium is found. To simplify the notation, we abbreviate the game function defined in Equation (4) as G . The update mechanisms for the strategy sequences $\{\mu_t\}_{t=1}^T$ and $\{\nu_t\}_{t=1}^T$ can be written as:

$$\begin{cases} \mu_{t+1} = \text{EMD}_\eta(\mu_t, \partial G / \partial \mu_t) \\ \nu_{t+1} = \text{EMD}_\eta(\nu_t, -\partial G / \partial \nu_t) \end{cases} \Rightarrow (\bar{\mu}_T, \bar{\nu}_T), \quad (5)$$

where $\bar{\mu}_T$ and $\bar{\nu}_T$ are the averages of corresponding sequences, *i.e.*, $\bar{\mu}_T := \frac{1}{T} \sum_{t=1}^T \mu_t$ and $\bar{\nu}_T := \frac{1}{T} \sum_{t=1}^T \nu_t$. And $(\bar{\mu}_T, \bar{\nu}_T)$ serves as an approximate Nash equilibrium.

Following Beck and Teboulle [2003], the EMD method replaces the Bregman divergence used in MD with an entropy function Φ . This results in the following iterative updates:

$$z_+ = \text{EMD}_\eta(z, h) = \nabla \Phi^*(\nabla \Phi(z) - \eta h). \quad (6)$$

The use of the entropy function in the EMD method allows it to take into account the inherent randomness and uncertainty present in mixed-strategy games, resulting in improved convergence and stability compared to other variants of the MD method. Specifically, the Shannon entropy is used as the measure of distance when the random variable z is discrete, and it is defined as $\Phi(Z) = \sum_{i=1}^n z_i \log z_i$, whereas, the differential entropy is used when z is continuous and it is defined as $\Phi(Z) = \int p(z) \log p(z) dz$, where $p(z)$ is the probability density function. To complete Equation (5), the next step is to compute the derivatives of G with respect to μ and ν , which are given as follows:

$$\begin{aligned} h_\mu(\theta) &:= \frac{\partial G}{\partial \mu} = \mathbb{E}_{\delta \sim \nu} [L(\theta) + \lambda R(\theta, \delta)], \\ h_\nu(\delta) &:= \frac{\partial G}{\partial \nu} = \mathbb{E}_{\theta \sim \mu} \lambda R(\theta, \delta). \end{aligned} \quad (7)$$

To summarize the above steps, we provide an algorithmic pseudocode as Algorithm 1, which allows us to iteratively converge to the Nash equilibrium of the mixed-strategy adversarial training game theoretically. However, since the density functions of μ and ν are implicit, it is not possible to directly find the Nash equilibrium in practice. Therefore, we require a practical method to estimate these distributions, which will be discussed in Section 3.3.

Algorithm 1 EMD Adversarial Training

Input: Initial Distributions μ_1 and ν_1 , Learning Rate η .

- 1: **for** $t = 1, 2, \dots, T - 1$ **do**
- 2: $\nu_{t+1} = \text{EMD}_\eta(\nu_t, -h_\nu^t)$
- 3: $\mu_{t+1} = \text{EMD}_\eta(\mu_t, h_\mu^t)$
- 4: **end for**
- 5: $\bar{\mu}_T = \frac{1}{T} \sum_{t=1}^T \mu_t$
- 6: **return** $\bar{\mu}_T$

Output: Model Parameters Distribution $\bar{\mu}_T$.

3.3 From Theory to Practice

Optimizing the desired risk over the distributions of model parameters and adversarial perturbation as outlined in Algorithm 1 is not practical. To overcome this, the empirical risk needs to be optimized through sampling in order to obtain an approximate distribution, which requires the density function of the sampled distribution.

Following the theorem proposed by Hsieh *et al.*[2019], which illustrated an explicit form of Infinite-Dimensional Mirror Descent, we can recursively apply EMD iterations to obtain the final density functions of μ and ν as below:

$$d\mu_T = \frac{e^{-\sum_{t=1}^T h_\mu^t} d\theta}{\int e^{-\sum_{t=1}^T h_\mu^t} d\theta}, \quad d\nu_T = \frac{e^{\sum_{t=1}^T h_\nu^t} d\delta}{\int e^{\sum_{t=1}^T h_\nu^t} d\delta}. \quad (8)$$

Given Equation (8), the μ_t and ν_t can be sampled and approximated by a set of $\{\theta_t^{(k)}\}_{k=1}^K$ and $\{\delta_t^{(k)}\}_{k=1}^K$. This equation indicates that by starting sampling from μ_1 , it is possible to recursively estimate the distribution of all $\{\mu_t\}_{t=1}^T$.

The remaining preparation is to estimate h_μ and h_ν , which are the expectations of the partial derivatives of G with respect to μ and ν . A commonly-used method to estimate unknown ground-truth expectations is to use the empirical average. To proceed, given K samples, the partial derivatives of μ and ν in Equation (7) can be approximated as follows:

$$\begin{aligned} \hat{h}_\mu(\theta) &\approx \frac{1}{K} \sum_{k=1}^K \left[L(\theta) + \lambda R(\theta, \delta^{(k)}) \right], \\ \hat{h}_\nu(\delta) &\approx \frac{1}{K} \sum_{k=1}^K \lambda R(\theta^{(k)}, \delta). \end{aligned} \quad (9)$$

During model training, maintaining the entire sample set $\{\theta_t^{(k)}\}_{k=1}^K$ and $\{\delta_t^{(k)}\}_{k=1}^K$ can be computationally intensive in terms of both space complexity and time complexity. So we use the mean of samples $\bar{\theta}_t$ and $\bar{\delta}_t$ instead of all sample points. This allows us to further simplify Equation (9) as:

$$\begin{aligned} \hat{h}_\mu(\theta) &\approx L(\theta) + \lambda R(\theta, \bar{\delta}), \\ \hat{h}_\nu(\delta) &\approx \lambda R(\bar{\theta}, \delta), \end{aligned} \quad (10)$$

where $\bar{\theta}$ and $\bar{\delta}$ are efficiently calculated by taking the exponential moving average, like $\bar{z}_{t+1} \leftarrow \beta \bar{z}_t + (1 - \beta) z_t$.

Now we have only one step left, which is collecting samples from distribution μ and ν . A widely used sampling method is Stochastic Gradient Langevin dynamics (SGLD) [Welling and Teh, 2011], which consists of Stochastic Gradient Descent and Langevin Dynamics. For any probability distribution with density function $e^{-h} dz$, such as Equation (8), SGLD iterates as follows:

$$z^{(k+1)} = z^{(k)} - \gamma \nabla h(z^{(k)}) + \sqrt{2\gamma} \epsilon \xi, \quad (11)$$

where γ is the sampling step size, ϵ is the thermal noise, and $\xi \sim \mathcal{N}(0, 1)$ is a standard normal vector. In addition to SGLD, Li *et al.*[2016] demonstrated that gradient-based optimization algorithms, including RMSprop [Hinton *et al.*,

Algorithm 2 MAT: Mixed-strategy Adversarial Training

Input: Pre-trained Model Parameters θ_1 ,

Sampling Step Size $\{\gamma_t\}_{t=1}^T$,

Sampling Times K , Thermal Noise ϵ .

```

1: for  $t = 1, 2, \dots, T - 1$  do
2:   Initialize  $\delta_t^{(1)}$ ,  $\bar{\delta}_t \leftarrow \delta_t^{(1)}$ 
3:   for  $k = 1, 2, \dots, K$  do
4:      $\delta_t^{(k+1)} \leftarrow \delta_t^{(k)} + \gamma_t \nabla_\delta \lambda R(\bar{\theta}_t, \delta_t^{(k)}) + \sqrt{2\gamma_t} \epsilon \xi$ 
5:      $\bar{\delta}_t \leftarrow \beta \bar{\delta}_t + (1 - \beta) \delta_t^{(k+1)}$ 
6:   end for
7:    $\theta_t^{(1)} \leftarrow \theta_t$ ,  $\bar{\theta}_t \leftarrow \theta_t$ 
8:   for  $k = 1, 2, \dots, K$  do
9:      $\theta_t^{(k+1)} \leftarrow \theta_t^{(k)} - \gamma_t \nabla_\theta \left[ L(\theta_t^{(k)}) + \lambda R(\theta_t^{(k)}, \bar{\delta}_t) \right] + \sqrt{2\gamma_t} \epsilon \xi$ 
10:     $\bar{\theta}_t \leftarrow \beta \bar{\theta}_t + (1 - \beta) \theta_t^{(k+1)}$ 
11:  end for
12:   $\theta_{t+1} \leftarrow \beta \theta_t + (1 - \beta) \bar{\theta}_t$ 
13: end for
14: return  $\theta_T$ 
    
```

Output: Fine-tuned Model Parameters θ_T .

2012] and Adam [Kingma and Ba, 2015], can also perform dynamic sampling. Their sampling iterations are as follows:

$$\begin{aligned} z^{(k+1)} &= z^{(k)} - \gamma \text{RMSprop} \left(\nabla h(z^{(k)}) \right) + \sqrt{2\gamma} \epsilon \xi, \\ z^{(k+1)} &= z^{(k)} - \gamma \text{Adam} \left(\nabla h(z^{(k)}) \right) + \sqrt{2\gamma} \epsilon \xi, \end{aligned} \quad (12)$$

where RMSprop and Adam refer to the specific variants of gradient calculations used in accordance with these optimization algorithms. If we substitute \hat{h}_μ and \hat{h}_ν from Equation (10) into Equation (11), the SGLD sampling iterations of δ and θ can be written as follows:

$$\begin{aligned} \delta_t^{(k+1)} &= \delta_t^{(k)} + \gamma_t \nabla_\delta \lambda R(\bar{\theta}_t, \delta_t^{(k)}) + \sqrt{2\gamma_t} \epsilon \xi, \\ \theta_t^{(k+1)} &= \theta_t^{(k)} - \gamma_t \nabla_\theta \left[L(\theta_t^{(k)}) + \lambda R(\theta_t^{(k)}, \bar{\delta}_t) \right] + \sqrt{2\gamma_t} \epsilon \xi. \end{aligned}$$

In summary, we entail the substitution of the original distribution with the mean of the distribution, which was approximated through the utilization of samples drawn from said distribution. Upon procuring the approximation of the sequence $\{\mu_t\}_{t=1}^T$, the final Nash equilibrium is executed by calculating the average of this sequence. To provide a clear overview of our method, we outline the whole process in Algorithm 2, which we refer to as MAT (Mixed-strategy Adversarial Training). Additionally, we provide a schematic in Figure 2 to intuitively illustrate the process of MAT.

4 Experiments

In this section, in order to verify the effectiveness of our proposed algorithm, MAT¹, we conducted a comprehensive evaluation of MAT using two widely recognized natural language

¹Code is available at <https://github.com/Charles-Zhong/MAT>.

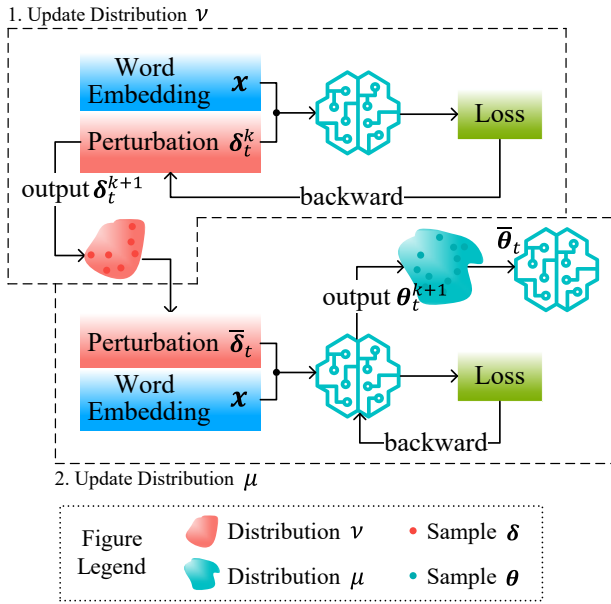


Figure 2: Schematic diagram of MAT algorithm.

understanding benchmarks: GLUE [Wang *et al.*, 2019] and ANLI [Nie *et al.*, 2020]. Additionally, MAT effectively showcase its capacity to enhance model generalization and robustness through a comprehensive comparison of experimental results with previous methods. Further details on the datasets and experiments are provided in the Appendix.

4.1 Models and Benchmarks

BERT Model. BERT [Devlin *et al.*, 2019] is a transformer-based language model widely recognized for its exceptional performance on various NLP tasks. The BERT-base version, which has about 110 Million parameters, is commonly used as a base model for fine-tuning various downstream tasks.

RoBERTa Model. RoBERTa [Liu *et al.*, 2019] is a variant of the BERT model that is trained with more extensive datasets and practical tricks such as dynamic masking, larger batch sizes, and longer training times. The RoBERTa-large version, consisting of a higher number of parameters (356M), is frequently used as a more robust base model for NLP tasks.

GLUE Benchmark. GLUE (General Language Understanding Evaluation) [Wang *et al.*, 2019] benchmark is a collection of nine tasks for natural language understanding model training and evaluation. The nine datasets comprising GLUE are CoLA [Warstadt *et al.*, 2019], SST-2 [Socher *et al.*, 2013], MRPC [Dolan and Brockett, 2005], STS-B [Cer *et al.*, 2017], QQP [Iyer *et al.*, 2017], 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], and WNLI [Levesque, 2011]. With the exception of STS-B, which is a regression task, all of the other tasks fall within the classification category.

ANLI Benchmark. ANLI (Adversarial Natural Language Inference) [Nie *et al.*, 2020] is an adversarial benchmark that is compiled through an iterative, adversarial human-and-

model-in-the-loop process. The dataset is divided into three parts, progressively increasing the challenges for language models. ANLI is typically used to measure the robustness of models, *i.e.*, the performance against adversarial attacks.

4.2 Implementation Details

Our implementation of the MAT algorithm is based on the PyTorch framework [Paszke *et al.*, 2019]. In addition, we leverage the Huggingface library, namely *transformers* [Wolf *et al.*, 2020] and *datasets* [Lhoest *et al.*, 2021], for loading pre-trained models and datasets. The text data in all the datasets are tokenized, and we limit the maximum length to 512 tokens. The hyperparameter settings are determined by combining empirical selection with the use of AutoML toolkits like NNI². The checkpoints of pre-trained models, including BERT-base³ and RoBERTa-large⁴, are obtained from the Huggingface repository.

During the model training phase, we utilize a range of sampling techniques, including Stochastic Gradient Langevin Dynamics (SGLD) sampling, along with its preconditioned variants such as the RMSProp-preconditioned and Adam-preconditioned versions of SGLD. Additionally, we experiment with a range of learning rates, specifically within the interval $\{1 \times 10^{-5}, 3 \times 10^{-5}, 5 \times 10^{-5}\}$, and employ various batch sizes, including 8, 16, 32, and 64. To ensure the robustness, we set a maximum norm of adversarial perturbation at 1×10^{-5} , and implement a clipping mechanism to curtail any perturbations that exceed this threshold. Furthermore, in accordance with previous works, we employ the single-task fine-tuning procedure for the GLUE benchmark, without utilizing any out-of-domain data. With regard to the ANLI benchmark, we follow the experimental setup of Nie *et al.*[2020] and Jiang *et al.*[2020] for facilitating comparison with their results, and train RoBERTa-larger model on the combined NLI datasets comprising of MNLI [Williams *et al.*, 2018], SNLI [Bowman *et al.*, 2015], FEVER [Thorne *et al.*, 2018], and ANLI [Nie *et al.*, 2020].

4.3 Generalization Results

In order to evaluate the performance of MAT, we conducted experiments on both BERT-base and RoBERTa-large models, utilizing the GLUE benchmark as the evaluation platform. The obtained results are compared to state-of-the-art adversarial training fine-tuning methods, including FreeLB [Zhu *et al.*, 2020], SMART [Jiang *et al.*, 2020], SALT [Zuo *et al.*, 2021], and R3F [Aghajanyan *et al.*, 2021]. Additionally, we also include several vanilla fine-tuned baseline models from Devlin *et al.*[2019], Jiang *et al.*[2020], and Liu *et al.*[2019] as a point of comparison. The numerical results are present in Table 1 and Table 2, respectively, for BERT-base and RoBERTa-large models. The metrics shown in the tables are Mcc (Matthews correlation coefficient), Acc (Accuracy), F1 (F1 score), and P/SCorr (Pearson and Spearman correlation coefficient), which are 100x scaled up from percentage representations. To facilitate the comparison, the best results

²<https://github.com/microsoft/nni>

³<https://huggingface.co/bert-base-uncased>

⁴<https://huggingface.co/roberta-large>

Dataset	CoLA	SST-2	MRPC	QQP	STS-B	MNLI	QNLI	RTE	Score
Size	8.6k	67.3k	3.7k	364k	5.7k	393k	104.7k	2.5k	
Metric	Mcc	Acc	Acc/F1	Acc/F1	P/SCorr	Acc	Acc	Acc	
BERT ¹	-	92.7	86.7/-	-	-	84.4/-	88.4	-	-
BERT ²	54.7	92.9	84.1/89.0	90.9/88.3	89.2/88.8	84.5/84.4	91.1	63.5	81.48
+SMART ²	59.1	93.0	87.7/91.3	91.5/88.5	90.0/89.4	85.6/86.0	91.7	71.2	83.75
+SALT ³	61.0	93.6	88.4/91.8	91.7/88.6	90.4/90.0	86.1/85.8	92.0	72.9	84.48
+MAT (Ours)	62.6	93.8	89.0/92.1	91.8/89.0	90.5/90.2	86.2/86.0	92.4	74.7	85.11

References: ¹ [Devlin *et al.*, 2019], ² [Jiang *et al.*, 2020], ³ [Zuo *et al.*, 2021].

Table 1: GLUE results based on the BERT-base model.

Dataset	CoLA	SST-2	MRPC	QQP	STS-B	MNLI	QNLI	RTE	Score
Size	8.6k	67.3k	3.7k	364k	5.7k	393k	104.7k	2.5k	
Metric	Mcc	Acc	Acc/F1	Acc/F1	P/SCorr	Acc	Acc	Acc	
RoBERTa ¹	68.0	96.4	90.9/-	92.2/-	92.4/-	90.2/-	94.7	86.6	88.93
+FreeLB ²	71.1	96.7	91.4/-	92.6/-	92.7/-	90.6/-	95.0	88.1	89.78
+SMART ³	70.6	96.9	89.2/92.1	92.4/89.8	92.8/92.6	91.1/91.3	95.6	92.0	90.09
+R3F ⁴	71.2	97.0	91.6/-	92.4/89.8	-	91.1/91.3	95.3	88.5	-
+MAT (Ours)	71.3	97.1	91.7/93.9	92.5/90.0	92.9/92.6	91.3/91.4	95.7	90.6	90.35

References: ¹ [Liu *et al.*, 2019], ² [Zhu *et al.*, 2020], ³ [Jiang *et al.*, 2020], ⁴ [Aghajanyan *et al.*, 2021].

Table 2: GLUE results based on the RoBERTa-large model.

are highlighted in bold. Additionally, we also calculate and present the overall GLUE scores for each method, where a higher number indicates superior overall performance.

BERT-base on GLUE. As demonstrated in Table 1, our proposed MAT exhibits a significant improvement in model generalization compared to both vanilla fine-tuning methods and existing state-of-the-art methods such as SMART and SALT, across all individual datasets. Specifically, MAT significantly enhances the model performance on the CoLA and RTE datasets, achieving scores of 62.6 and 74.7, respectively, which represent an improvement of 1.6-7.9 and 1.8-11.2 points compared to competing methods. Furthermore, on the other datasets, MAT also demonstrates a notable superiority in performance. For instance, on large-scale datasets such as QQP and MNLI, MAT achieves the highest scores of 91.8/89.0 and 86.2/86.0, respectively, representing an improvement of 1.9/0.7 and 1.7/1.6 points over vanilla fine-tuning methods, as well as an improvement of 0.1/0.4 and 0.1/0.2 over the previous state-of-the-art method (SALT). Overall, MAT achieves a final score of 85.11, which is substantially ahead of competitors by 0.63 to 3.63 points.

RoBERTa-large on GLUE. As drawn in Table 2, the superiority of MAT still holds on RoBERTa-large model. MAT achieves the best performance on seven-eight datasets, with the remaining dataset ranking second. Remarkably, MAT achieves substantial improvements of 0.26-1.42 points on average and obtains the highest overall score among all methods. These results further demonstrate the effectiveness of MAT in robustly optimizing large-scale transformer models.

4.4 Robustness Results

Besides generalization, the robustness of models has also gained increasing attention [Liang *et al.*, 2018; Jin *et al.*, 2020; Zhou *et al.*, 2021; Wang *et al.*, 2022]. It refers to the ability of a model to maintain its performance when faced with unexpected or malicious inputs. Thus, we evaluate the robustness gained from MAT on the ANLI benchmark.

The experiments were conducted on the dev-set and test-set of ANLI, which results are in Table 3. It is evident that MAT performs significantly better than vanilla fine-tuning [Nie *et al.*, 2020] and SMART [Jiang *et al.*, 2020] across all levels of adversarial challenges. Specifically, MAT reaches 49.3 and 74.7 on dev-sets R3 and test-sets R1, substantially ahead of others by 1.7 and 2.3 points. These results indicate that MAT fine-tuned models exhibit superior robustness against adversarial texts across a wide range of testing scenarios.

Datasets	MNLI + SNLI + FEVER + ANLI					
Evaluation ANLI Part	ANLI-Dev			ANLI-Test		
	R1	R2	R3	R1	R2	R3
RoBERTa ¹	73.8	48.9	44.4	-	-	-
+SMART ²	74.5	50.9	47.6	72.4	49.8	50.3
+MAT (Ours)	74.8	51.0	49.3	74.7	51.1	50.5

References: ¹ [Nie *et al.*, 2020], ² [Jiang *et al.*, 2020].

Table 3: ANLI results based on the RoBERTa-large model.

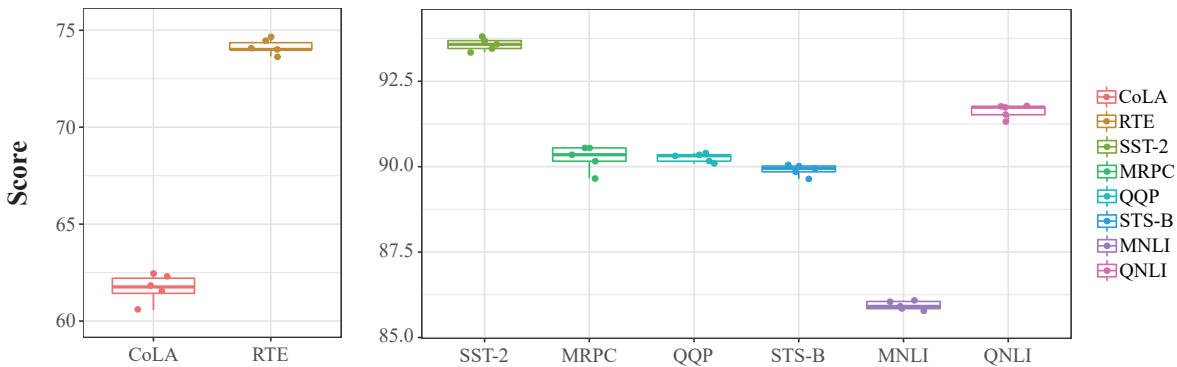


Figure 3: Box plot error bar of MAT on the GLUE benchmark.

4.5 Error Bar Analysis

Training deep learning models is essentially solving a non-convex optimization problem that may have numerous local optima. Therefore, it is essential to evaluate the reliability of MAT methods by conducting sufficiently repeated experiments with different random starting points.

To ensure fair comparisons with other methods, we conducted five experiments applying MAT to the BERT-base model on GLUE benchmarks, each with different random seeds. To visualize our experimental results, we represented them as a box plot in Figure 3. In order to present the results more effectively, we divided all datasets into two parts, displayed on the left and right sides of the graph, with distinct vertical coordinate ranges. For datasets with multiple metrics, such as MRPC, QQP, STS-B, and MNLI, we calculated the average of these metrics to obtain the final score. The results consistently demonstrate that across all five random seeds, both the maximum and median values of our runs are consistently higher than those of other methods. These findings provide strong evidence of the reliability of MAT.

4.6 Ablation Study

In this subsection, we analyze the impact of several vital hyperparameters on the performance of the MAT algorithm. Specifically, we focus on the number of samples K in Equation (9), which is used to approximate the probability distributions, as well as the regularization coefficient λ in Equation (4) that balances the loss of regular training and adversarial training. To investigate the effect of these hyperparameters, we conducted experiments on the BERT-base model using the CoLA, SST-2, and MRPC datasets from the GLUE benchmark, while keeping all other hyperparameters constant.

The impact of the number of samples K on MAT performance is shown in Table 4. Since the exponential moving averages are leveraged in MAT, the model performance may be negatively impacted by both too small and too large values of K . When K is too small, the distributions may not be accurately approximated. On the other hand, if K is set to a large value, the estimated distributions may be overly dependent on the neighbors of the latest iterate, while ignoring other possibilities that may have occurred in previous iterates. Based on these observations, it is recommended to select an

MAT- K	1	5	10	20	30	50
CoLA (Mcc)	61.5	61.7	62.6	61.6	62.2	61.4
SST-2 (Acc)	93.1	93.8	93.8	93.2	93.0	93.0
MRPC (Acc)	85.7	87.0	89.0	87.3	86.5	86.3
MRPC (F1)	90.0	90.9	92.1	91.1	90.3	90.2

Table 4: Ablation study result of K on the BERT-base model.

MAT- λ	0.1	1	3	5	10	50
CoLA (Mcc)	60.8	60.1	62.6	61.9	62.4	57.8
SST-2 (Acc)	93.7	93.7	93.8	93.5	93.5	92.4
MRPC (Acc)	86.5	88.2	88.7	89.0	86.3	85.5
MRPC (F1)	90.3	91.5	92.0	92.1	90.2	90.0

Table 5: Ablation study result of λ on the BERT-base model.

appropriate number of samples based on the coefficient of the exponential moving average to achieve the best results.

Table 5 presents the experimental results for different values of λ s. Based on our hyperparameter tuning experience, we find that choosing a larger λ value is generally beneficial for smaller datasets. Conversely, larger λ values tend to degrade performance on larger datasets. From the results, it can be inferred that the use of adversarial training leads to better generalization of the model over unseen samples.

5 Conclusion

This work introduces a mixed-strategy game into adversarial training for fine-tuning large-scale pre-trained models, by building a game between the model and adversarial perturbations, incorporating the Entropic Mirror Descent for Adversarial Training to derive the Nash equilibrium. To simplify this method for practical usage, we propose the MAT algorithm by a sampling method. The MAT algorithm is evaluated on multiple benchmarks with pre-trained models and compared with previous state-of-the-art methods. The experimental results demonstrate that MAT leads to better performance in both model generalization and robustness. Our work also provides solid practical support for the research of introducing game theory into adversarial training.

A Datasets

In this supplementary section, we collate and organize the information pertaining to the datasets employed in our experiments. Table 6 presents an overview of the task type, the number of classification categories, and the sample size for each dataset. Additional details concerning the datasets are provided in the following paragraphs.

CoLA Dataset. The Corpus of Linguistic Acceptability (CoLA) is an assemblage of English acceptability judgments extracted from a corpus of literature on linguistic theory. Each example within the dataset is comprised of a sentence and accompanied by a binary label, indicating whether the sentence is deemed grammatically acceptable.

SST-2 Dataset. The Stanford Sentiment Treebank (SST-2) dataset is comprised of sentences sourced from movie reviews, along with corresponding human-annotated sentiments. The task is to predict the sentiment of a given sentence, where the dataset utilizes two classification categories (positive/negative) provided labels at the sentence level.

MRPC Dataset. The Microsoft Research Paraphrase Corpus (MRPC) is a corpus of sentence pairs that are automatically obtained from various online news sources and manually annotated to indicate the presence or absence of semantic equivalence within the sentence pairs.

QQP Dataset. The Quora Question Pairs (QQP) dataset is a compilation of question pairs that are extracted from the community question-answering website Quora. The task of QQP dataset is to identify whether a given pair of questions are semantically equivalent or not.

STS-B Dataset. The Semantic Textual Similarity Benchmark (STS-B) dataset is a collection of sentence pairs that are sourced from news headlines, video and image captions, as well as natural language inference data. Each sentence pair within the dataset is annotated by a human, which is accompanied with a similarity score from 0 to 5.

MNLI Dataset. The Multi-genre Natural Language Inference corpus (MNLI) is a comprehensive dataset that is comprised of sentence pairs which are crowd-sourced and annotated with textual entailment labels. The task is to predict the relationship between a given premise sentence and a hypothesis sentence, precisely whether the premise entails the hypothesis (entailment), contradicts the hypothesis (contradiction), or neither (neutral). The premise sentences in this dataset are collected from ten different sources, including transcribed speeches, novels, and government reports.

QNLI Dataset. The Question Natural Language Inference (QNLI) dataset is an adaptation of the Stanford Question Answering Dataset (SQuAD), which comprises question and paragraph pairs. In the original dataset, a sentence within a paragraph sourced from Wikipedia contains the answer to the related question, which is written by an annotator. The transformed task in QNLI is determining whether the context sentence contains the answer to the question.

RTE Dataset. The Recognizing Textual Entailment (RTE) dataset comes from a series of annual textual entailment challenges, specifically RTE1, RTE2, RTE3, and RTE5. This dataset is converted into a binary classification task, where the three classes of the original datasets (neutrality, contradiction, and entailment) are collapsed into two classes (entailment and not_entailment) for consistency.

ANLI Dataset. The Adversarial Natural Language Inference (ANLI) dataset is a large-scale benchmark that is constructed through an iterative process of cyclic human-model confrontation. This dataset is divided into three parts, each of which is designed to increase the difficulty progressively for deep neural network models. ANLI is commonly utilized to evaluate the robustness of a model, specifically its ability to perform when faced with adversarial attacks.

B Hyperparameters

Our experiments were conducted on multiple servers that are equipped with NVIDIA V100 and NVIDIA 3090 GPUs. As detailed in the main text of the paper, a combination of empirical methods and the AutoML toolkit (NNI) were employed for the hyperparameters search. As for the sensitivity of the hyperparameters, it is worth noting that the proposed MAT algorithm is not dependent on the meticulous tuning of the hyperparameters, as results that surpass the previous state-of-the-art were achieved with tiny hyperparameter adjustments. The value ranges of some hyperparameters that were explored during the training process are presented in Table 7.

Hyperparameter	Search Range
Sampling Times K	{5, 10, 15, 20, 30}
Sampling Step Size γ	$\{1 \times 10^{-5}, 3 \times 10^{-5}, 5 \times 10^{-5}\}$
Beta β	{0.1, 0.3, 0.5, 0.7, 0.9}
Lambda λ	{0.1, 1, 2, 3, 4, 5}

Table 7: Search ranges of the hyperparameters.

Dataset	CoLA	SST-2	MRPC	QQP	STS-B	MNLI	QNLI	RTE	ANLI
Task Type	<i>class.</i>	<i>class.</i>	<i>class.</i>	<i>class.</i>	<i>regress.</i>	<i>class.</i>	<i>class.</i>	<i>class.</i>	<i>class.</i>
Lables	2	2	2	2	-	3	2	2	3
Train-set	8,551	67,349	3,668	363,846	5,749	392,702	104,743	2,490	162,901
Dev-set	1,043	872	408	40,430	1,500	19,647	5,463	277	3,200
Test-set	1,063	1,821	1,752	390,965	1,379	19,643	5,463	3,000	3,200

Abbreviations: *class.* denotes a classification task and *regress.* denotes a regression task.

Table 6: Summary of the datasets in the GLUE and ANLI benchmark.

Acknowledgments

We would like to thank Xinrun Wang and the anonymous reviewers for their helpful discussions and support. This work is supported by the Key R&D Programs of Zhejiang (2022C01018), the Joint Funds of National Natural Science Foundation of China (U21B2001), and the National Natural Science Foundation of China (62176080).

Contribution Statements

We state that Zhehua Zhong and Tianyi Chen contributed equally to this work.

References

- [Aghajanyan *et al.*, 2021] Armen Aghajanyan, Akshat Shrivastava, Anchit Gupta, Naman Goyal, Luke Zettlemoyer, and Sonal Gupta. Better fine-tuning by reducing representational collapse. In *ICLR*, 2021.
- [Ahuja *et al.*, 2020] Kartik Ahuja, Karthikeyan Shanmugam, Kush R. Varshney, and Amit Dhurandhar. Invariant risk minimization games. In *ICML*, 2020.
- [Arora *et al.*, 2017] Sanjeev Arora, Rong Ge, Yingyu Liang, Tengyu Ma, and Yi Zhang. Generalization and equilibrium in generative adversarial nets (gans). In *ICML*, 2017.
- [Bar-Haim *et al.*, 2006] Roy Bar-Haim, Ido Dagan, Bill Dolan, Lisa Ferro, Danilo Giampiccolo, Bernardo Magnini, et al. The second PASCAL recognizing textual entailment challenge. In *Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*, 2006.
- [Beck and Teboulle, 2003] Amir Beck and Marc Teboulle. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Operations Research Letters*, 2003.
- [Bentivogli *et al.*, 2009] 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*, 2009.
- [Bowman *et al.*, 2015] Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. A large annotated corpus for learning natural language inference. In *EMNLP*, 2015.
- [Brown *et al.*, 2020] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, et al. Language models are few-shot learners. In *NeurIPS*, 2020.
- [Cer *et al.*, 2017] Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, 2017.
- [Dagan *et al.*, 2005] Ido Dagan, Oren Glickman, and Bernardo Magnini. The PASCAL recognizing textual entailment challenge. In *First PASCAL Machine Learning Challenges Workshop*, 2005.
- [Daskalakis *et al.*, 2018] Constantinos Daskalakis, Andrew Ilyas, Vasilis Syrgkanis, and Haoyang Zeng. Training gans with optimism. In *ICLR*, 2018.
- [Devlin *et al.*, 2019] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, 2019.
- [Dolan and Brockett, 2005] William B. Dolan and Chris Brockett. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing*, 2005.
- [Giampiccolo *et al.*, 2007] Danilo Giampiccolo, Bernardo Magnini, Ido Dagan, and Bill Dolan. The third PASCAL recognizing textual entailment challenge. In *Proceedings of the ACL-PASCAL@ACL 2007 Workshop on Textual Entailment and Paraphrasing*, 2007.
- [Goodfellow *et al.*, 2014] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, et al. Generative adversarial nets. In *NeurIPS*, 2014.
- [Goodfellow *et al.*, 2015] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *ICLR*, 2015.
- [Hinton *et al.*, 2012] Geoffrey Hinton, Nitish Srivastava, and Kevin Swersky. Rmsprop: Divide the gradient by a running average of its recent magnitude. *Coursera*, 2012.
- [Hsieh *et al.*, 2019] Ya-Ping Hsieh, Chen Liu, and Volkan Cevher. Finding mixed nash equilibria of generative adversarial networks. In *ICML*, 2019.
- [Iyer *et al.*, 2017] Shankar Iyer, Nikhil Dandekar, and Kornel Csernai. First quora dataset release: Question pairs. Technical report, Quora, 2017.
- [Jiang *et al.*, 2020] 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 *ACL*, 2020.
- [Jin *et al.*, 2020] Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. Is BERT really robust? A strong baseline for natural language attack on text classification and entailment. In *AAAI*, 2020.
- [Kingma and Ba, 2015] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.
- [Levesque, 2011] Hector J. Levesque. The winograd schema challenge. In *AAAI Spring Symposium*, 2011.
- [Lhoest *et al.*, 2021] Quentin Lhoest, Albert Villanova del Moral, Yacine Jernite, Abhishek Thakur, Patrick von Platen, Suraj Patil, et al. Datasets: A community library for natural language processing. In *EMNLP*, 2021.
- [Li *et al.*, 2016] Chunyuan Li, Changyou Chen, David E. Carlson, and Lawrence Carin. Preconditioned stochastic

- gradient langevin dynamics for deep neural networks. In *AAAI*, 2016.
- [Liang *et al.*, 2018] Bin Liang, Hongcheng Li, Miaoqiang Su, Pan Bian, Xirong Li, and Wenchang Shi. Deep text classification can be fooled. In *IJCAI*, 2018.
- [Liu *et al.*, 2019] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, et al. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692, 2019.
- [Liu *et al.*, 2020] Xiaodong Liu, Hao Cheng, Pengcheng He, Weizhu Chen, Yu Wang, Hoifung Poon, et al. Adversarial training for large neural language models. *CoRR*, abs/2004.08994, 2020.
- [Madry *et al.*, 2018] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *ICLR*, 2018.
- [Miyato *et al.*, 2017] Takeru Miyato, Andrew M. Dai, and Ian J. Goodfellow. Adversarial training methods for semi-supervised text classification. In *ICLR*, 2017.
- [Nash, 1951] John Nash. Non-cooperative games. *Annals of Mathematics*, 54(2):286–295, 1951.
- [Nemirovski and Yudin, 1983] Arkadi Nemirovski and D. Yudin. *Problem complexity and method efficiency in optimization*. Wiley, 1983.
- [Nie *et al.*, 2020] Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. Adversarial NLI: A new benchmark for natural language understanding. In *ACL*, 2020.
- [Paszke *et al.*, 2019] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, et al. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, 2019.
- [Radford *et al.*, 2018] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. *OpenAI blog*, 2018.
- [Radford *et al.*, 2019] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 2019.
- [Raffel *et al.*, 2020] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, et al. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 2020.
- [Rajpurkar *et al.*, 2016] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100, 000+ questions for machine comprehension of text. In *EMNLP*, 2016.
- [Shafahi *et al.*, 2019] Ali Shafahi, Mahyar Najibi, Amin Ghiasi, Zheng Xu, John P. Dickerson, Christoph Studer, et al. Adversarial training for free! In *NeurIPS*, 2019.
- [Socher *et al.*, 2013] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng, et al. Recursive deep models for semantic compositionality over a sentiment treebank. In *EMNLP*, 2013.
- [Szegedy *et al.*, 2014] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, et al. Intriguing properties of neural networks. In *ICLR*, 2014.
- [Thorne *et al.*, 2018] James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. FEVER: a large-scale dataset for fact extraction and verification. In *NAACL-HLT*, 2018.
- [Wang *et al.*, 2019] 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 *ICLR*, 2019.
- [Wang *et al.*, 2022] Zhen Wang, Yitao Zheng, Hai Zhu, Chang Yang, and Tianyi Chen. Transferable adversarial examples can efficiently fool topic models. *Computers & Security*, 118:102749, 2022.
- [Warstadt *et al.*, 2019] Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 2019.
- [Welling and Teh, 2011] Max Welling and Yee Whye Teh. Bayesian learning via stochastic gradient langevin dynamics. In *ICML*, 2011.
- [Williams *et al.*, 2018] Adina Williams, Nikita Nangia, and Samuel R. Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *NAACL-HLT*, 2018.
- [Wolf *et al.*, 2020] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, et al. Transformers: State-of-the-art natural language processing. In *EMNLP*, 2020.
- [Zhang *et al.*, 2019] Dinghuai Zhang, Tianyuan Zhang, Yiping Lu, Zhanxing Zhu, and Bin Dong. You only propagate once: Accelerating adversarial training via maximal principle. In *NeurIPS*, 2019.
- [Zhou *et al.*, 2021] Qi Zhou, Haipeng Chen, Yitao Zheng, and Zhen Wang. Evalda: Efficient evasion attacks towards latent dirichlet allocation. In *AAAI*, 2021.
- [Zhu *et al.*, 2020] Chen Zhu, Yu Cheng, Zhe Gan, Siqi Sun, Tom Goldstein, and Jingjing Liu. FreeLB: Enhanced adversarial training for natural language understanding. In *ICLR*, 2020.
- [Zuo *et al.*, 2021] Simiao Zuo, Chen Liang, Haoming Jiang, Xiaodong Liu, Pengcheng He, Jianfeng Gao, et al. Adversarial regularization as stackelberg game: An unrolled optimization approach. In *EMNLP*, 2021.