Efficient Adversarial Training with Robust Early-Bird Tickets

Zhiheng Xi[1*], **Rui Zheng**[1*], **Tao Gui**[2†], **Qi Zhang**[1], **Xuanjing Huang**[1,3]

[1] School of Computer Science, Fudan University, Shanghai, China
[2] Institute of Modern Languages and Linguistics, Fudan University, Shanghai, China
[3] International Human Phenome Institutes (Shanghai), Shanghai, China
zhxi22@m.fudan.edu.cn , {rzheng20,tgui,qz,xjhuang}@fudan.edu.cn

Abstract

Adversarial training is one of the most powerful methods to improve the robustness of pre-trained language models (PLMs). However, this approach is typically more expensive than traditional fine-tuning because of the necessity to generate adversarial examples via gradient descent. Delving into the optimization process of adversarial training, we find that robust connectivity patterns emerge in the early training phase (typically $0.15 \sim 0.3$ epochs), far before parameters converge. Inspired by this finding, we dig out robust early-bird tickets (i.e., subnetworks) to develop an efficient adversarial training method: (1) searching for robust tickets with structured sparsity in the early stage; (2) fine-tuning robust tickets in the remaining time. To extract the robust tickets as early as possible, we design a ticket convergence metric to automatically terminate the searching process. Experiments show that the proposed efficient adversarial training method can achieve up to $7\times \sim 13\times$ training speedups while maintaining comparable or even better robustness compared to the most competitive state-of-the-art adversarial training methods.

1 Introduction

Pre-trained language models (PLMs) have achieved great success in NLP (Devlin et al., 2019a), but they are vulnerable to adversarial examples crafted by performing subtle perturbations on normal examples (Ren et al., 2019; Garg and Ramakrishnan, 2020). Recent studies have shown the prevalence of adversarial vulnerability in NLP tasks (Wallace et al., 2019; Zhang et al., 2021; Lin et al., 2021). Various defense strategies have been proposed to improve the robustness of the model and maintain high accuracy on both normal and adversarial examples (Zhu et al., 2020; Wang et al.,
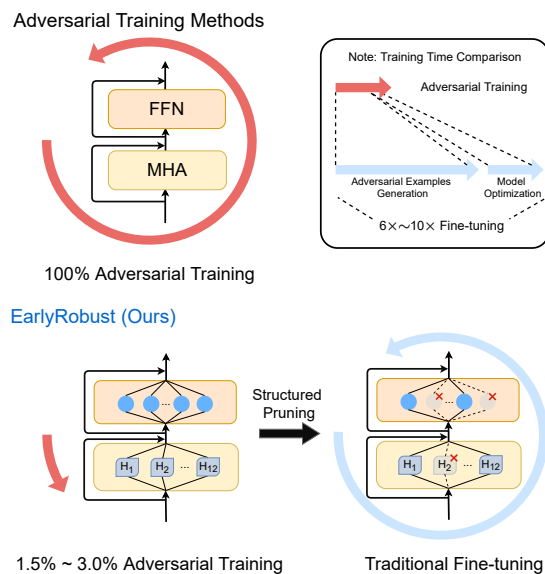

Figure 1: A high-level overview of adversarial training methods and the proposed EarlyRobust. **MHA** means multi-head attention and **FFN** means feed-forward network. **Structured pruning** removes attention heads and intermediate neurons of each transformer layer in BERT. Our method achieves up to $7\times \sim 13\times$ training speedups compared to adversarial training methods.

2021). Adversarial training is one of the most widely used and strongest defense methods (Madry et al., 2018).

One of the main challenges of adversarial training in real-world applications is its high computational cost, as they require multi-step gradient descents to generate adversarial examples (Wong et al., 2020; Andriushchenko and Flammarion, 2020). Some work on this topic found that replacing strong adversarial examples with weaker and cheaper ones does not significantly change the resulting robustness (Wong et al., 2020). Zhang et al. (2019) remove redundant calculations during backpropagation for additional speedup. FreeAT (Shafahi et al., 2019) and FreeLB (Zhu et al., 2020) utilize a "free" strategy to generate diverse adversarial examples at a negligible additional cost.

---

[*]Equal contribution.
[†]Corresponding author.

However, these methods are only alleviating the huge expense of generating adversarial samples, and still require optimizing the adversarial loss objective throughout the training period.

The recently proposed robust lottery ticket hypothesis (Fu et al., 2021; Zheng et al., 2022b) suggests that a deep network contains robust tickets (i.e., subnetworks) that maintain matching accuracy but better robustness than the original network when trained individually. However, robust tickets have to be searched through a tedious process under the guidance of the adversarial loss objective (Zheng et al., 2022b). Moreover, these robust tickets consider unstructured sparsity, which only reduces storage but does not reduce computational overhead (Prasanna et al., 2020).

In this work, we propose a novel **robust early-bird** ticket with **structured sparsity** for efficient adversarial training. We delve into the optimization process of adversarial training, and find that the robust connectivity patterns converge with few training iterations. As shown in Figure 1, we can extract robust tickets at the early stage of adversarial training by pruning the self-attention heads and intermediate neurons that contribute least to accuracy and robustness. A highly robust model can be easily obtained by fine-tuning the proposed robust tickets in the remaining time. In addition, we use a generic ticket convergence metric to automatically terminate the search process without going through each search moment. Experimental results show that the proposed method achieves up to $7\times \sim 13\times$ training speedups while maintaining comparable or even better adversarial robustness compared with traditional adversarial training. Our codes are publicly available at *Github*[1]. The contributions of this work can be summarized as:

- We analyze the optimization process of adversarial training and reveal that the robust connectivity patterns emerge in the early training phase.

- We propose a novel robust early-bird ticket for efficient adversarial training, which consists of two stages: (1) searching for robust tickets using adversarial training in the early stage; and (2) fine-tuning the robust tickets during the remaining time.

- Compared with previous efficient adversarial

training methods, the proposed approach provides a new pathway based on model pruning and robust architecture searching.

2 Related Work

2.1 Textual Adversarial Attack and Defense

Textual adversarial attacks try to fool NLP models with adversarial examples which are constructed by substituting some parts of sentences with their counterparts. Typically, the adversarial examples hold a high similarity with clean ones in semantics or embedding space (Li et al., 2019; Ren et al., 2019; Jin et al., 2020; Li et al., 2020). To improve the robustness of NLP models against textual adversarial attacks, many defense methods have been proposed (Li et al., 2021; Zheng et al., 2022a; Liu et al., 2022). As the most popular one, adversarial training (AT) solves a robust min-max optimization problem by adding norm-bounded perturbations to word embeddings (Madry et al., 2018; Zhu et al., 2020; Li and Qiu, 2021). Likewise, some regularization methods have been proved beneficial to model robustness (Jiang et al., 2020; Wang et al., 2021).

2.2 Efficient Adversarial Training Methods

Adversarial training, represented by PGD (Projected Gradient Descent, Madry et al. (2018)), costs much more computational resources than traditional training due to the need to construct adversarial examples. Therefore, some recent works attempt to improve its efficiency. FreeAT (Shafahi et al., 2019) and YOPO (Zhang et al., 2019) simplify the calculation of gradients to get speedups. Zhu et al. (2020) introduce the above accelerating methods to NLP and propose FreeLB, which adds adversarial perturbations to word embeddings. Our method shapes structured robust tickets in the early adversarial training stage, so it is orthogonal to the above methods.

2.3 Lottery Ticket Hypothesis

Lottery Ticket Hypothesis (LTH) suggests that there exist certain sparse subnetworks (i.e., winning tickets) at initialization that can be trained to achieve competitive performance compared to the full model (Frankle and Carbin, 2019). In NLP, previous work has found that matching subnetworks exist in Transformers, LSTMs, and PLMs (Yu et al., 2020; Renda et al., 2020; Chen et al., 2020). Prasanna et al. (2020) find structured

tickets for BERT, which are efficient in training and inference, by pruning attention heads and MLP layers. Recently, You et al. (2020) and Chen et al. (2021) pioneer to identify Early-Bird tickets at the early stage of training. Fu et al. (2021) and Zheng et al. (2022b) find unstructured robust tickets in neural networks and thus the robust LTH is proposed. However, the search process for winning robust tickets is tedious. A concurrent work in CV jointly identifies and trains unstructured Flying Bird tickets with PGD-AT(Chen et al., 2022), while we explore the convergence of robust connectivity patterns in NLP. Particularly, we identify robust early-bird tickets with structured sparsity and apply them to fine-tuning.

3 Methodology

In this section, we first revisit the robust lottery ticket hypothesis (Sec.3.1). Next, we describe the early emergence of structured robust tickets (Sec.3.2). Based on this key finding, we propose the EarlyRobust method for efficient adversarial training (Sec.3.3). Finally, we make a brief review and discuss how our method accelerates adversarial training (Sec.3.4).

3.1 Revisiting Robust Lottery Ticket Hypothesis

For a network $f(\theta_0)$ initialized with parameters θ_0, we can identify a binary mask m which has the same dimension as θ_0, such that a subnetwork $f(m \odot \theta_0)^2$ can be shaped. Robust lottery ticket hypothesis articulates that, there exists certain subnetwork that can be trained to have matching performance and better robustness compared to the full model following the same training protocol (Zheng et al., 2022b). Such a subnetwork $f(m \odot \theta_0)$ is a so-called *robust winning ticket*, including both the mask m and the initialization θ_0.

3.2 Early Emergence of Robust Tickets with Structured Sparsity

In previous work, robust tickets with unstructured sparsity in PLMs are extracted after a tedious training process. However, when applying a structured pruning strategy, we observe that **robust tickets with structured sparsity emerge in the early training stage.** The structured pruning strategy aims at pruning self-attention heads and intermediate neurons of each transformer layer in

---

$^2\odot$ is the Hadamard product operator.

BERT. Through visualizing the normalized mask distance between different training steps (You et al., 2020) in Figure 2, we learn that the structure of a robust ticket can be identified at a very early adversarial training stage. Inspired by this finding, we propose an efficient adversarial training method that draws robust early-bird tickets at the early adversarial training stage and then fine-tunes these tickets to achieve better robustness.

3.3 Robust Early-bird Tickets

Our method consists of two steps: 1) Searching Stage; 2) Drawing and Fine-tuning Stage.

3.3.1 Searching Stage

BERT is constructed by a stack of transformer encoder layers (Vaswani et al., 2017), and the structure of each layer is identical: a multi-head self-attention (MHA) block followed by a feed-forward network (FFN), with residual connections around each. In each layer, the MHA with N_h heads takes an input x and outputs:

$$\mathrm{MHA}(x) = \sum_{i=1}^{N_h} \mathrm{Att}_{W_K^i, W_Q^i, W_V^i, W_O^i}(x), \quad (1)$$

where W_K^i, W_Q^i, $W_V^i \in \mathbb{R}^{d_h \times d}$, $W_O^i \in \mathbb{R}^{d \times d_h}$, and they denote the query, key, value and output matrices in the l-th attention head. Here d denotes the hidden size (e.g., 768) and $d_h = d/N_h$ denotes the output dimension of each head (e.g., 64).

Next comes an FFN parameterized by $W_U \in \mathbb{R}^{d \times d_f}$ and $W_D \in \mathbb{R}^{d_f \times d}$:

$$\mathrm{FFN}(x) = \mathrm{gelu}(X W_U) \cdot W_D, \quad (2)$$

where $d_f = 4d$.

Learnable Importance Coefficients Recent research reveals that MHAs and FFNs in transformers are redundant (Michel et al., 2019; Voita et al., 2019). Therefore, we adopt the modified network slimming (Liu et al., 2017; Chen et al., 2021), which assigns learnable importance coefficients to self-attention heads and intermediate neurons, to determine which components are essential for robustness:

$$\mathrm{MHA}(x) = \sum_{i=1}^{N_h} c_\mathrm{H}^i \cdot \mathrm{Att}_{W_K^i, W_Q^i, W_V^i, W_O^i}(x), \quad (3)$$

$$\mathrm{FFN}(x) = c_\mathrm{F} \cdot \mathrm{gelu}(X W_U) \cdot W_D, \quad (4)$$

where c_H denotes the coefficients for heads, i is the index of head, and c_F denotes the coefficients for

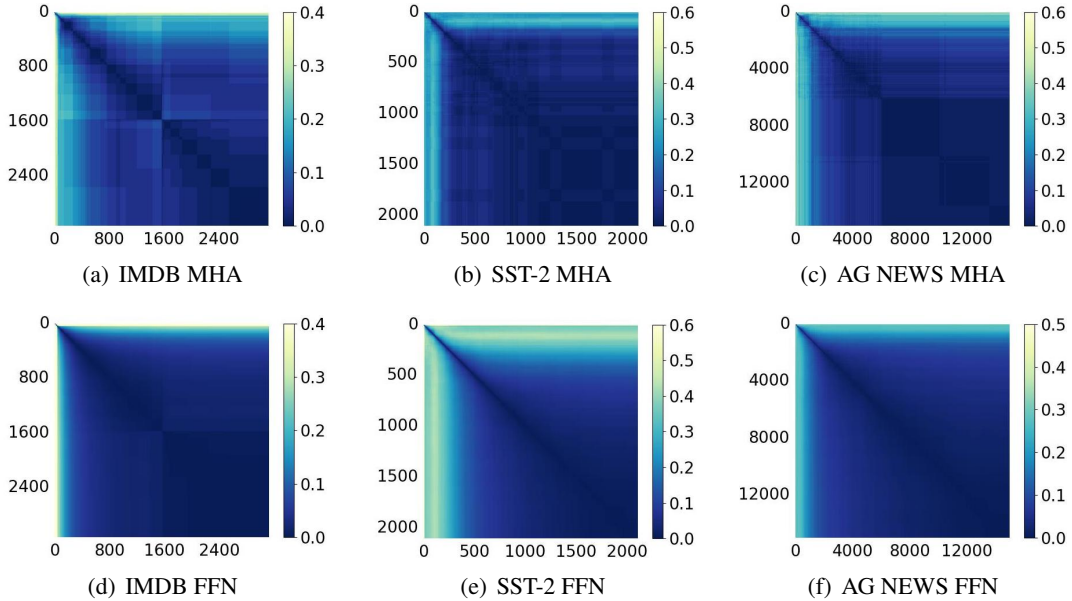| (a) IMDB MHA | (b) SST-2 MHA | (c) AG NEWS MHA |
| (d) IMDB FFN | (e) SST-2 FFN | (f) AG NEWS FFN |

Figure 2: Visualization of mask difference in Hamming distance between different training steps. Top: mask distance in self-attention heads. Bottom: mask distance in intermediate neurons. The axes represent the numbers of **training steps** finished and the color stands for the **normalized distance** between different masks. The darker the color, the smaller the distance. The masks for attention heads and intermediate neurons both converge early.

the FFN. Then, we can jointly train BERT with the importance coefficients but with a regularizer:

$$\mathcal{R}(c) = \lambda_{\mathrm{H}} \|c_{\mathrm{H}}\|_1 + \lambda_{\mathrm{F}} \|c_{\mathrm{F}}\|_1, \qquad (5)$$

where $c = \{c_{\mathrm{H}}, c_{\mathrm{F}}\}$, λ_{H} and λ_{F} denote regularization strength for the two kinds of coefficients respectively.

Adversarial Loss Objective To identify substructures that are responsible for adversarial robustness, we introduce the adversarial loss objective:

$$\min_{\theta, c} \underbrace{\mathbb{E}_{(x,y) \sim \mathcal{D}} \max_{\|\delta\| \leq \epsilon} \mathcal{L}\left(f(x + \delta; \theta, c), y\right),}_{\mathcal{L}_{adv}(\theta, c)} \qquad (6)$$

where (x, y) is a data point from dataset $\mathcal{D}$, δ is the perturbation that is constrained within the ϵ ball. The inner max problem tries to generate worst-case perturbation towards inputs and maximize the model's classification loss, while the outer min problem is to optimize the model on the perturbed data.[3]

Then, by integrating the adversarial training objective and the sparsity regularizer $\mathcal{R}(c)$, we can describe our loss objective as follow:

$$\min_{\theta, c} \mathcal{L}_{adv}(\theta, c) + \mathcal{R}(c). \qquad (7)$$

---
[3]See more details about how to solve the inner max problem in Appendix A.1.

After the joint training, we draw robust tickets, reset the parameters and then train the tickets with traditional fine-tuning.

3.3.2 Drawing and Fine-tuning Stage

Early-stopping Strategy As mentioned before, robust tickets with structured sparsity can be identified in the early adversarial training stage. However, it is difficult to determine the exact search termination time. The difficulty lies mainly in the following two aspects: (1) the termination time varies among datasets; (2) the termination moments of MHA and FFN in the same model are different. Therefore, we design a termination metric that measures the normalized mask distance between consecutive miniepochs (each miniepoch consists of 0.05 epochs). When the termination metric is smaller than the threshold γ both for MHA and FFN, we extract robust early-bird tickets. Similar termination metrics are widely used for the extraction of subnetworks (You et al., 2020). To ensure the consistency of convergence, we end the subnetwork search stage when the termination metric is satisfied five times in a row.

Pruning Strategy At the end of the search process, we prune the unimportant parts according to the magnitudes of the learned importance coefficients. The attention heads and intermediate neurons with the smallest importance coefficients are considered

---

Algorithm 1: EarlyRobust Method

Input: model parameters θ, learnable importance coefficients c, learning rate η and the fine-tuning epoch N.

1 **Procedure** ROBUST TICKETS SEARCHING
Initialize θ, c;
2 **repeat**
3     $\theta = \theta - \eta\nabla_\theta(\mathcal{L}_{adv}(\theta, c) + \mathcal{R}(c))$;
4     $c = c - \eta\nabla_c(\mathcal{L}_{adv}(\theta, c) + \mathcal{R}(c))$;
5 **until** *the convergence condition in Sec.3.3.2 is satisfied*;
6 **Procedure** DRAWING AND FINE-TUNING
Extract robust winning tickets parameterized by θ_{ticket};
7 **for** $epoch \leftarrow 1...N$ **do**
8     $\theta_{ticket} \leftarrow \theta - \eta\nabla_{\theta_{ticket}}\mathcal{L}_{CE}$;
9 **end**

---

to contribute the least to robustness and should be removed. For attention heads, we not only remove them from the computation graph, but also remove the corresponding W_O (see Eqn. (3)). We adopt a modified global approach to prune the heads and ensure that at least one head survives in each layer.[4] As for intermediate neurons, pruning them is equivalent to reducing the size of W_U and W_D in Eqn. (4). We prune intermediate neurons globally as there is a large number of neurons in each FFN layer and empirical analysis suggests that global approach brings better performance.

Efficient Robust Fine-tuning After pruning attention heads and intermediate neurons, we then reset the weights to the pre-trained weights and utilize traditional fine-tuning to train the robust tickets.

3.4 A Brief Review

The proposed method searches for the key connectivity patterns for robustness , draws the winning tickets and trains them with traditional fine-tuning. We summarize it in Algorithm 1.

How does EarlyRobust accelerate adversarial training? Firstly, the method draws robust tickets with *structured sparsity*, which reduces the size of models and speeds up training and inference. Secondly, the *early-stopping strategy* prevents the

---

[4]The attention heads in some layers may be completely removed if we use a global pruning approach, leaving the network un-trainable. While our experiments (see Sec.5.6) show that global pruning is more effective than layer-wise pruning. The modified global approach is a trade-off between trainability and effectiveness.

tedious searching process for robust tickets. Finally, the method trains the robust early-bird tickets with *traditional fine-tuning* instead of adversarial training methods, getting rid of the time-consuming process to construct adversarial examples.

4 Experimental Settings

4.1 Backbone Model and Datasets

We take BERT$_{BASE}$ (12 transformer layers, 12 attention heads, 3,072 intermediate neurons per layer, hidden size 768 and 110M parameters in total) as the backbone model. We follow the BERT implementation in (Devlin et al., 2019b; Wolf et al., 2019). Mainly, we evaluate the proposed method on three text classification datasets, including IMDB (Maas et al., 2011), SST-2 (Socher et al., 2013), and AG NEWs (Zhang et al., 2015). We also include other datasets of more tasks in GLUE (Wang et al., 2019) as a supplement, such as QNLI and QQP.

4.2 Baselines

We compare our method with fine-tuned BERT, EarlyBERT, and several strong robust baselines. **Fine-tune** (Devlin et al., 2019b): Training full BERT on downstream tasks. **EarlyBERT** (Chen et al., 2021): A framework that draws early-bird tickets from BERT for efficient fine-tuning. **PGD** (Madry et al., 2018): An adversarial training algorithm that minimizes the empirical loss on worst-case adversarial examples. **FreeLB** (Zhu et al., 2020): An enhanced PGD-based adversarial training method for natural language understanding. **InfoBERT** (Wang et al., 2021): A framework that improves model's robustness from an information-theoretic perspective. **RobusT** (Zheng et al., 2022b): An approach that identifies robust tickets from the original PLMs through learning binary masks.

4.3 Evaluation Settings

Robust Evaluation We apply three popular textual attack methods (Li et al., 2021) to evaluate the model's defensive capability. **TextBugger** (Li et al., 2019) generates misspelled words by using character-level and word-level perturbations. **TextFooler** (Jin et al., 2020) constructs adversarial examples by substituting the most important words in a sentence with synonyms. **BERT-Attack** (Li et al., 2020) employs BERT to generate adversarial

Dataset	Method	Speedup	Params	Clean%	TextFooler		TextBugger		BERT-Attack	
					Aua%	#Query	Aua%	#Query	Aua%	#Query
IMDB	Fine-tune†	9.4×	85M	92.05	11.8	922.4	23.2	695.2	7.8	1187.0
	EarlyBERT†	**15.6×**	**58M**	91.9	3.4	874.9	12.6	649.7	0.8	1099.9
	PGD‡	1.0×	85M	**93.2**	30.2	1562.8	41.6	905.8	21.8	2114.6
	FreeLB‡	1.2×	85M	**93.2**	35.0	1736.9	53.0	**1110.9**	29.0	2588.8
	InfoBERT‡	0.3×	85M	93.3	49.6	1932.3	53.8	1070.4	47.2	3088.8
	RobusT‡	0.4×	**68M**	91.8	**58.6**	**1994.7**	**63.6**	1153.3	**58.0**	3120.2
	EarlyRobust	12.6×	58M	91.8	63.8	2012.8	63.8	1107.1	63.0	3125.6
SST-2	Fine-tune†	10.1×	85M	92.2	17.9	123.4	40.6	53.4	13.3	158.3
	EarlyBERT†	**18.6×**	63M	92.3	13.4	111.5	37.7	52.9	9.9	138.9
	PGD‡	1.0×	85M	93.2	18.1	118.5	44.2	56.0	13.4	151.3
	FreeLB‡	1.4×	85M	91.7	29.4	152.6	**49.7**	**58.6**	23.8	174.7
	InfoBERT‡	0.6×	85M	92.4	18.3	126.8	42.4	54.6	15.0	160.4
	RobusT‡	0.4×	60M	90.9	28.6	**149.8**	43.1	53.9	20.8	169.2
	EarlyRobust	13.7×	63M	**92.8**	**28.8**	142.8	46.3	61.9	**22.7**	187.1
AG NEWS	Fine-tune†	6.8×	85M	94.6	28.6	383.3	45.2	192.5	17.6	556.0
	EarlyBERT†	**7.2×**	69M	94.5	32.8	392.3	46.6	195.4	21.2	571.1
	PGD‡	1.0×	85M	**95.0**	**36.8**	414.9	56.4	201.8	21.6	616.1
	FreeLB‡	1.3×	85M	**95.0**	34.8	408.5	**54.2**	**210.3**	20.4	596.2
	InfoBERT‡	0.4×	85M	94.5	33.8	395.6	49.6	194.1	**23.4**	**618.9**
	RobusT‡	0.4×	60M	94.9	35.2	**415.6**	49.0	206.9	21.8	617.5
	EarlyRobust	7.0×	69M	94.4	41.0	416.2	50.0	216.4	32.2	620.1

Table 1: Experimental results of adversarial robustness evaluation. The best performance is marked in **bold** and underline; the second is marked in **bold**. **Speedup** means training speedup, which is reported against adversarial training method PGD. **Params** is the number of model parameters.[5] Methods labeled by † are fine-tuning baselines without considering adversarial robustness, and methods labeled by ‡ are adversarial defense baselines. Our method achieves high adversarial robustness while maintaining a low computational and storage consumption.

texts, and thus the semantic consistency and the language fluency are preserved.

We adopt three evaluating metrics as follows: **Clean accuracy (Clean%)** is the model's accuracy on the clean test set. **Accuracy under attack (Aua%)** denotes the model's prediction accuracy under certain adversarial attack methods. **Number of Queries (#Query)** refers to the average attempts an attacker queries the model, and the larger the number of queries is, the more difficult the model is to be fooled.

Training Time Measurement Protocol We measure the training time of each method on GPU and exclude the time for data I/O. To get rid of randomness, we run each method five times and report the average time. Notably, the training time of LTH-based methods (e.g., EarlyRobust, EarlyBERT, and RobusT) includes both the searching stage and the fine-tuning stage.

4.4 Implementation Details

We re-implement baseline methods based on their open-source codes and the results are competitive. We employ FreeLB to implement the adversarial

loss objective in the searching stage of EarlyRobust because compared to standard K-step PGD, it accumulates gradients of parameters in multiple forward passes and passes gradients backward once. **Clean%** is tested on the whole test set. **Aua%** and **#Query** are evaluated on the whole test dataset for SST-2, and 500 randomly chosen samples for other datasets. We set the early-stopping threshold γ to 0.1 and fine-tune EarlyRobust tickets 10 epochs. For fine-tuning, EarlyBERT, and other adversarial training methods, we also set the training epoch to 10, which is a trade-off between training cost and performance. The sparsity for EarlyBERT is the same as that of EarlyRobust. We prune 40%, 30%, 20%, 30% and 30% intermediate neurons on IMDB, SST-2, AG NEWS, QNLI and QQP, respectively; the pruning ratio for attention heads is $1/6$. More implementation details and hyperparameters can be found in Appendix A.2.

5 Experimental Results and Discussion

In this section, we illustrate the effectiveness and efficiency of our method with experimental results.

---

[5] We exclude embedding matrices when calculating the number of parameters following previous work.

Dataset	Method	Speedup	Params	Clean%	Aua%
QNLI	Fine-tune[†]	9.6×	85M	**91.6**	5.8
	PGD[‡]	1.0×	85M	91.2	12.2
	FreeLB[‡]	1.3×	85M	90.5	12.8
	EarlyRobust	**12.2×**	**63M**	91.4	**18.8**
QQP	Fine-tune[†]	9.8×	85M	**91.3**	24.8
	PGD[‡]	1.0×	85M	91.2	27.0
	FreeLB[‡]	1.3×	85M	91.2	27.4
	EarlyRobust	**13.6×**	**63M**	90.9	**32.6**

Table 2: Experimental results on QNLI and QQP. **Aua**% is obtained after using TextFooler attack. Our method brings robustness improvement and training speedups on different tasks.

Dataset	Method	Clean%	Aua%
IMDB	**EarlyRobust**	91.8	**63.8**
	w/o Adv	**92.0**	6.4
	w/o Regularizer	87.5	31.2
	Random Tickets	91.1	6.2
SST-2	**EarlyRobust**	**92.8**	**28.8**
	w/o Adv	92.2	13.9
	w/o Regularizer	91.3	8.5
	Random Tickets	90.9	9.1
AG NEWS	**EarlyRobust**	94.4	**41.0**
	w/o Adv	**94.6**	31.0
	w/o Regularizer	92.5	31.2
	Random Tickets	94.0	19.4

Table 3: Ablation study. **Aua**% is obtained after using TextFooler attack. Results of randomly pruned tickets are the average of 5 trials with different random seeds. Both Adversarial loss objective and sparsity-inducing regularizer play a fundamental role for high-quality tickets. Random tickets suffer a large degradation in robustness, indicating that robust tickets are non-trivial.
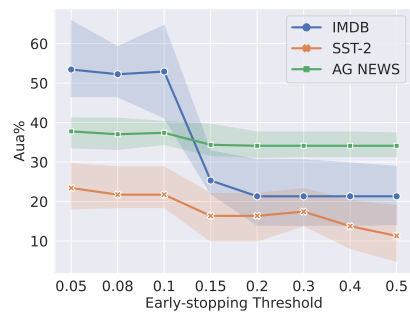
5.1 Main Results

The main results of EarlyRobust and other baselines are summarized in Table 1. We can observe that: 1) Our method achieves high robustness with a little sacrifice in accuracy. It performs slightly worse than FreeLB on SST-2, but achieves the best robustness on both IMDB and AG NEWS, suggesting that the subnetworks shaped by our method own inborn robustness which can be exerted through traditional fine-tuning. 2) Our method achieves sizable training accelerations compared to robust baselines. This proves our method can deliver highly robust models under different kinds of adversarial attacks with limited computational resources. Though EarlyBERT and fine-tuning are also fast, they do not take robustness into account and thus have weak defensive capabilities. 3) Our method also brings storage savings, and thus robust early-bird tickets are suitable to be deployed on mobile devices and edge devices.

We evaluate our method on paraphrase and inference tasks to verify its generality. Results in Table 2 show that our method consistently performs well on more difficult tasks.

5.2 Ablation Study

To investigate the contribution of different components to our method, we perform ablation experiments by removing adversarial loss objective (Adv) and sparsity-inducing regularizer. We also evaluate the performance of randomly pruned tickets. Results in Table 3 show that: 1) Adversarial loss objective is important for robustness, but not for clean accuracy. 2) There exists degradation in both accuracy and robustness if we remove the regularizer, indicating that the sparsity it

Figure 3: Influence of different early-stopping thresholds (i.e., γ) on model robustness. **Aua**% is obtained after using TextFooler attack and the results are obtained from 5 trials of different random seeds. Model robustness converges when γ decreases to around 0.1.

induces is indispensable for high-quality tickets. 3) Randomly pruned tickets have a competitive accuracy performance, which is also observed by (Prasanna et al., 2020). However, random tickets consistently suffer a large drop in robustness, which proves that robust tickets are non-trivial.

5.3 Different Early-stopping Thresholds

Here we study how different early-stopping thresholds affect the model's robustness. Results in Figure 3 show that the model's robustness converges when γ is reduced to around 0.1. Considering training efficiency, we think 0.1 is a reasonable threshold.

Dataset	Method	Clean%	Aua%
	EarlyRobust	<u>**91.8**</u>	<u>**63.8**</u>
IMDB	w/o Initialization	82.4	0.0
	w/o Structure	89.2	4.0
	EarlyRobust	<u>**92.8**</u>	<u>**28.8**</u>
SST-2	w/o Initialization	81.5	0.8
	w/o Structure	88.4	20.0
	EarlyRobust	<u>**94.4**</u>	<u>**41.0**</u>
AG NEWS	w/o Initialization	92.2	0.6
	w/o Structure	<u>**94.4**</u>	36.6

Table 4: Importance of EarlyRobust ticket initialization and structure. **Aua**% is obtained under TextFooler attack. The pre-trained initialization and the structure are both indispensable for EarlyRobust tickets. The pre-trained initialization seems more important than the structure.

5.4 Importance of EarlyRobust Tickets Initialization and Structure

According to LTH, the winning tickets can not be trained effectively without original initialization and the ticket structure also plays a role (Frankle and Carbin, 2019). Therefore, we follow a similar way in (Zheng et al., 2022b) to study the importance of the initialization and the structure of EarlyRobust tickets. Specifically, we re-initialize the weights of EarlyRobust tickets to exclude the effect of the initialization. We employ the full BERT and re-initialize the weights which are not contained in the robust tickets to exclude the effect of the structure while preserving the effect of the initialization. Table 4 demonstrates that the pre-trained initialization and the structure are both fundamental for model performance. Another observation is that without the initialization, robust tickets suffer a greater performance drop than without the structure.

5.5 Impact of Sparsity

In our method, structured sparsity can not only bring speedups but also save the storage. Figure 4 illustrates that as the sparsity (both for heads and neurons) increases, the accuracy shows a gradually decreasing trend, but the robustness is different. The robustness first increases and then degrades sharply as sparsity grows, which is similar to the observations in (Fu et al., 2021). To extract high-quality tickets, $1/6 \sim 1/4$ and $20\% \sim 40\%$ are suitable pruning ratio ranges for heads and neurons, respectively. We also visualize the sparsity patterns in Appendix A.3.
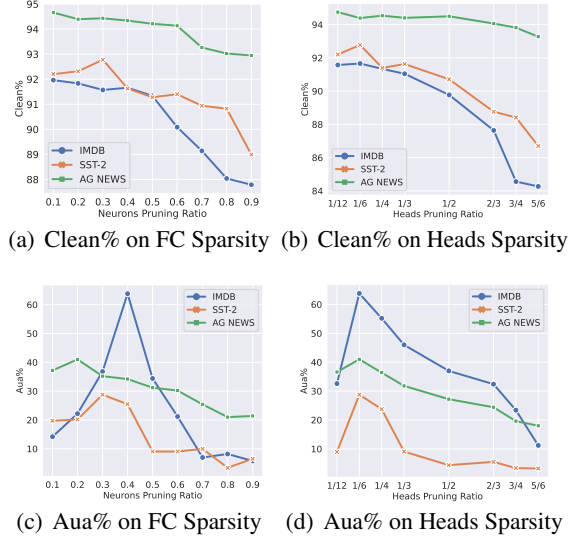

(a) Clean% on FC Sparsity (b) Clean% on Heads Sparsity

(c) Aua% on FC Sparsity (d) Aua% on Heads Sparsity

Figure 4: Impact of Sparsity on **Clean**% and **Aua**%. **Aua**% is obtained under TextFooler attack.

Clean%	Heads-L	Heads-G
Aua%		
Neurons-L	91.5	91.6
	1.2	35.4
Neurons-G	91.4	<u>**91.7**</u>
	56.0	<u>**63.8**</u>

Table 5: Influence of global and layer-wise pruning on model performance on IMDB. **-L** means layer-wise while **-G** means global. **Aua**% is obtained under TextFooler attack. Global pruning outperforms layer-wise pruning in terms of robustness.

5.6 Global Pruning vs. Layer-wise Pruning

As mentioned in Sec.3.3.2, we prune self-attention heads and intermediate neurons separately with global approach. Here we study how global pruning and layer-wise pruning influence model performance. Table 5 demonstrates the results on IMDB.[6] We can observe that: 1) Layer-wise pruning performs as well as global pruning in accuracy, but it induces a degradation in robustness, revealing that different layers have different degrees of contribution to robustness. So different layers should not be pruned with the same ratio. 2) The model's robustness drops more dramatically when we use layer-wise pruning for neurons than for heads, suggesting that intermediate neurons are more sensitive to pruning approaches.

---

[6]See Appendix A.4 for more results.

6 Conclusion

In this paper, we delve into the optimization process of adversarial training on BERT, and observe the early convergence of robust connectivity patterns. Based on this, we propose EarlyRobust, a method that shapes structured robust early-bird tickets under the guidance of the adversarial loss objective and then utilizes efficient fine-tuning to train them. Experiments on various datasets demonstrate that our method achieves competitive or even better robustness compared to other strong baselines while maintaining a very low computational and storage consumption.

Limitation

In this work, we propose a method to draw structured robust early-bird tickets, which can be used as an efficient alternative to adversarial training. However, there are still limitations to be explored in future work: 1) Model robustness drops sharply when the pruning ratio is too large, so our robust tickets do not have high sparsity. We expect to compress the robust early-bird tickets to a smaller size while maintaining a high quality in the future. 2) We draw the robust early-bird tickets from pre-trained models, and future work includes searching for robust early-bird tickets from randomly initialized networks.

Ethics Statement

Our method is proposed for efficient adversarial training, so it is environmental-friendly. It achieves up to $7\times \sim 13\times$ speedups compared to standard adversarial training, saving computational resources by a significant margin. Moreover, the proposed robust early-bird tickets with structured sparsity bring savings in storage, so it is easy to deploy them on mobile devices and edge devices.

Acknowledgements

References

Maksym Andriushchenko and Nicolas Flammarion. 2020. Understanding and improving fast adversarial training. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Tianlong Chen, Jonathan Frankle, Shiyu Chang, Sijia Liu, Yang Zhang, Zhangyang Wang, and Michael Carbin. 2020. The lottery ticket hypothesis for pre-trained BERT networks. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Tianlong Chen, Zhenyu Zhang, Pengjun Wang, Santosh Balachandra, Haoyu Ma, Zehao Wang, and Zhangyang Wang. 2022. Sparsity winning twice: Better robust generalization from more efficient training. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Xiaohan Chen, Yu Cheng, Shuohang Wang, Zhe Gan, Zhangyang Wang, and Jingjing Liu. 2021. Earlybert: Efficient BERT training via early-bird lottery tickets. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 2195–2207. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019a. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019b. BERT: pre-training of deep bidirectional transformers for language understanding. In *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)*, pages 4171–4186. Association for Computational Linguistics.

Jonathan Frankle and Michael Carbin. 2019. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Yonggan Fu, Qixuan Yu, Yang Zhang, Shang Wu, Xu Ouyang, David D. Cox, and Yingyan Lin.

2021. Drawing robust scratch tickets: Subnetworks with inborn robustness are found within randomly initialized networks. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 13059–13072.

Siddhant Garg and Goutham Ramakrishnan. 2020. BAE: BERT-based adversarial examples for text classification. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6174–6181, Online. Association for Computational Linguistics.

Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Tuo Zhao. 2020. SMART: robust and efficient fine-tuning for pretrained natural language models through principled regularized optimization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 2177–2190. Association for Computational Linguistics.

Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. 2020. Is BERT really robust? A strong baseline for natural language attack on text classification and entailment. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 8018–8025. AAAI Press.

Jinfeng Li, Shouling Ji, Tianyu Du, Bo Li, and Ting Wang. 2019. Textbugger: Generating adversarial text against real-world applications. In *26th Annual Network and Distributed System Security Symposium, NDSS 2019, San Diego, California, USA, February 24-27, 2019*. The Internet Society.

Linyang Li, Ruotian Ma, Qipeng Guo, Xiangyang Xue, and Xipeng Qiu. 2020. BERT-ATTACK: adversarial attack against BERT using BERT. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 6193–6202. Association for Computational Linguistics.

Linyang Li and Xipeng Qiu. 2021. Token-aware virtual adversarial training in natural language understanding. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 8410–8418. AAAI Press.

Zongyi Li, Jianhan Xu, Jiehang Zeng, Linyang Li, Xiaoqing Zheng, Qi Zhang, Kai-Wei Chang, and Cho-Jui Hsieh. 2021. Searching for an effective defender: Benchmarking defense against adversarial word substitution. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 3137–3147. Association for Computational Linguistics.

Jieyu Lin, Jiajie Zou, and Nai Ding. 2021. Using adversarial attacks to reveal the statistical bias in machine reading comprehension models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 333–342, Online. Association for Computational Linguistics.

Qin Liu, Rui Zheng, Bao Rong, Jingyi Liu, Zhihua Liu, Zhanzhan Cheng, Liang Qiao, Tao Gui, Qi Zhang, and Xuanjing Huang. 2022. Flooding-x: Improving bert's resistance to adversarial attacks via loss-restricted fine-tuning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 5634–5644. Association for Computational Linguistics.

Zhuang Liu, Jianguo Li, Zhiqiang Shen, Gao Huang, Shoumeng Yan, and Changshui Zhang. 2017. Learning efficient convolutional networks through network slimming. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 2755–2763. IEEE Computer Society.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference, 19-24 June, 2011, Portland, Oregon, USA*, pages 142–150. The Association for Computer Linguistics.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2018. 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.

Paul Michel, Omer Levy, and Graham Neubig. 2019. Are sixteen heads really better than one? In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 14014–14024.

John X. Morris, Eli Lifland, Jin Yong Yoo, Jake Grigsby, Di Jin, and Yanjun Qi. 2020. Textattack: A framework for adversarial attacks, data augmentation, and adversarial training in nlp.

Sai Prasanna, Anna Rogers, and Anna Rumshisky. 2020. When BERT plays the lottery, all tickets are winning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 3208–3229. Association for Computational Linguistics.

Shuhuai Ren, Yihe Deng, Kun He, and Wanxiang Che. 2019. Generating natural language adversarial examples through probability weighted word saliency. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 1085–1097. Association for Computational Linguistics.

Alex Renda, Jonathan Frankle, and Michael Carbin. 2020. Comparing rewinding and fine-tuning in neural network pruning. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Ali Shafahi, Mahyar Najibi, Amin Ghiasi, Zheng Xu, John P. Dickerson, Christoph Studer, Larry S. Davis, Gavin Taylor, and Tom Goldstein. 2019. Adversarial training for free! In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 3353–3364.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng, and Christopher Potts. 2013. 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*, pages 1631–1642. ACL.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. 2019. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28-August 2, 2019, Volume 1: Long Papers*, pages 5797–5808. Association for Computational Linguistics.

Eric Wallace, Pedro Rodriguez, Shi Feng, Ikuya Yamada, and Jordan Boyd-Graber. 2019. Trick me if you can: Human-in-the-loop generation of adversarial examples for question answering. *Transactions of the Association for Computational Linguistics*, 7:387–401.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. 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.

Boxin Wang, Shuohang Wang, Yu Cheng, Zhe Gan, Ruoxi Jia, Bo Li, and Jingjing Liu. 2021. Infobert: Improving robustness of language models from an information theoretic perspective. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *CoRR*, abs/1910.03771.

Eric Wong, Leslie Rice, and J. Zico Kolter. 2020. Fast is better than free: Revisiting adversarial training. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Haoran You, Chaojian Li, Pengfei Xu, Yonggan Fu, Yue Wang, Xiaohan Chen, Richard G. Baraniuk, Zhangyang Wang, and Yingyan Lin. 2020. Drawing early-bird tickets: Toward more efficient training of deep networks. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Haonan Yu, Sergey Edunov, Yuandong Tian, and Ari S. Morcos. 2020. Playing the lottery with rewards and multiple languages: lottery tickets in RL and NLP. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Dinghuai Zhang, Tianyuan Zhang, Yiping Lu, Zhanxing Zhu, and Bin Dong. 2019. You only propagate once: Accelerating adversarial training via maximal principle. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 227–238.

Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 649–657.

Xinze Zhang, Junzhe Zhang, Zhenhua Chen, and Kun He. 2021. Crafting adversarial examples for neural machine translation. In *Proceedings of the 59th Annual Meeting of the Association for*

Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 1967–1977, Online. Association for Computational Linguistics.

Rui Zheng, Rong Bao, Qin Liu, Tao Gui, Qi Zhang, Xuanjing Huang, Rui Xie, and Wei Wu. 2022a. Plugat: A plug and play module to defend against textual adversarial attack. In *Proceedings of the 29th International Conference on Computational Linguistics, COLING 2022, Gyeongju, Republic of Korea, October 12-17, 2022*, pages 2873–2882. International Committee on Computational Linguistics.

Rui Zheng, Bao Rong, Yuhao Zhou, Di Liang, Sirui Wang, Wei Wu, Tao Gui, Qi Zhang, and Xuanjing Huang. 2022b. Robust lottery tickets for pre-trained language models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2211–2224, Dublin, Ireland. Association for Computational Linguistics.

Chen Zhu, Yu Cheng, Zhe Gan, Siqi Sun, Tom Goldstein, and Jingjing Liu. 2020. 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.

A Appendix

A.1 Solving the Inner Max Problem of Adversarial Loss Objective

We solve the inner max problem(generate the worst-case perturbation, i.e., $\mathcal{L}_{adv}(\theta, c)$) with K-step projected gradient descent strategy (Madry et al., 2018), and the $(k + 1)$-th perturbation is defined as :

$$\delta_{k+1} = \prod_{\|\delta\| \leq \epsilon} \left(\delta_k + \eta \frac{g(\delta_k)}{\|g(\delta_k)\|} \right), \qquad (8)$$

where $g(\delta_k) = \nabla_{\delta_k} \mathcal{L}(f(x + \delta_k; \theta, c), y)$ is the gradient of the loss with respect to δ, and $\prod_{\|\delta\| \leq \epsilon}(\cdot)$ performs a projection onto the ϵ-ball, which is a Frobenius normalization ball.

A.2 Implementation Details

A.2.1 Details for Adversarial Attack

We adopt textattack (Morris et al., 2020) to implement adversarial attack methods. **Aua%** and **#Query** are evaluated on all 872 test examples for SST-2, 500 randomly selected test samples for IMDB, AG NEWS, QNLI and QQP. For BERT-Attack, we set the neighbor vocabulary size to 50, and set the sentence similarity to 0.2; for the other two attack methods, we use the default parameters of third-party libraries.

A.2.2 Hardware Details

Experiments on SST-2, QNLI and QQP are run on seven 1080Ti GPUs and the sequence length is set to 128, while experiments on AG NEWS and IMDB are run on eight 2080Ti GPUs and the sequence length is set to 256.

A.2.3 Hyperparameters

There are four hyperparameters introduced by the adversarial loss objective: the initial magnitude of perturbations ϵ_0, the number of gradient descent steps for adversary s, the perturbation step size τ, and we do not constrain the bound of perturbations. We report the learning rate η and the learning rate is the same at both the searching and fine-tuning stages. Moreover, we report the regularization penalties λ_{H} and λ_{F}. We list the hyperparameters used for each task in Table 6.

Datasets	ϵ_0	s	τ	η	λ_{H}	λ_{F}
IMDB	0.05	5	0.01	$2e-5$	$1e-5$	$2e-4$
SST2	0.05	5	0.02	$2e-5$	$1e-4$	$3e-4$
AGNEWS	0.05	5	0.01	$2e-5$	$3e-4$	$5e-5$
QNLI	0.05	5	0.01	$2e-5$	$2e-4$	$3e-4$
QQP	0.05	5	0.01	$2e-5$	$1e-5$	$1e-5$

Table 6: Hyperparameters in the proposed method.

A.3 Sparsity Pattern

Figure 5 illustrates the sparsity pattern of robust tickets on three datasets. We can observe that for different datasets, the sparsity pattern varies greatly. So we speculate that it is dataset-dependent. Another interesting finding is that on a given dataset, the pruning for attention heads and intermediate neurons is concentrated in different layers, i.e., if attention heads are pruned centrally in some layers, the pruning on intermediate neurons is concentrated in other layers. For example, on SST-2, our method prunes heads mainly in bottom layers, while pruning intermediate neurons mainly in top layers.

A.4 More Results of Global Pruning vs. Layer-wise Pruning

Table 7 and Table 8 show the results on SST-2 and AG NEWS, respectively. We can consistently find that *global* pruning performs better than *layer-wise* pruning in terms of robustness.

Clean% Aua%	Heads-L	Heads-G
Neurons-L	90.1	91.5
	11.7	12.0
Neurons-G	91.6	**92.8**
	24.3	**28.8**

Table 7: Global Pruning vs. Layer-wise Pruning on SST-2.

Clean% Aua%	Heads-L	Heads-G
Neurons-L	94.5	**94.6**
	36.8	38.4
Neurons-G	94.4	94.5
	39.6	**41.0**

Table 8: Global Pruning vs. Layer-wise Pruning on AG NEWS.

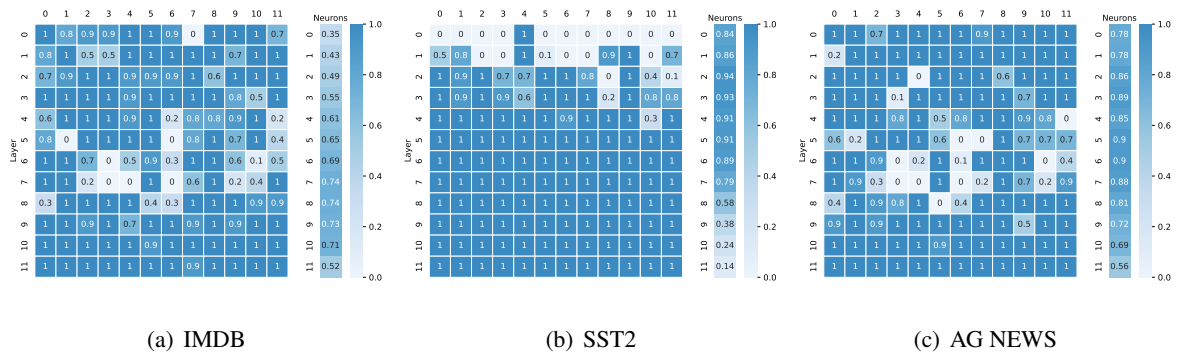(a) IMDB   (b) SST2   (c) AG NEWS

Figure 5: Sparsity pattern of EarlyRobust tickets. For each figure, cells in the left give the average number of random seeds where a given head survives; cells in the right give the average percentage of surviving weights across 10 random seeds. We can observe that the sparsity pattern is dataset-dependent and interestingly, on a given dataset, the pruning for heads and intermediate neurons is concentrated in different layers.