Efficient Generative Adversarial Training for Language Models via Multi-task Feature Transfer

Anonymous ACL submission

Abstract

Adversarial training is a well-known methodology for enhancing language models and avoiding harmful responses and misclassification. Although adversarial training has gained empirical success, many existing methods to create embeddings via query-based adversarial samples that are different from actual realistic text adversarial features during the training process. In this work, we propose UnGAT and MulGAT, new approaches for adversarial training. They produce perturbations as discrete tokens rather than apply perturbations to embedding representations during whole training process. In particular, both UnGAT and MulGAT consist of a generator that produces adversarial text and a victim model fine-tuned on both original and adversarial text. While UnGAT's generator is fine-tuned to fool victim model without adversarial dataset, MulGAT transfers adversarial features from source tasks to unseen tasks via a generator fine-tuned on multi-task adversarial dataset. Experiments on text classification and dialogue generation demonstrate the effectiveness of our approaches over many state-of-the-art baselines.

1 Introduction

The vulnerability of deep learning models to adversarial attacks and samples has been well known for recent years (Zhu et al., 2020; Wu et al., 2023; Madry et al., 2018). The performance of a language model is significantly reduced in the evaluation of the robustness benchmark (Wang et al., 2021) and query-based adversarial attacks (Li et al., 2023b). In particular, the perturbation of one or some words or characters in the original input can mislead language models without changing semantics and meanings. Adversarial training is a methodology to make language models less brittle against these attacks (Madry et al., 2018; Zhu et al., 2020; Raman et al., 2023).

Despite the achievement of many adversarial training methods, they need to construct the embedding perturbation in the latent space, necessitating multiple iterations of gradient descent for each sample (Zhu et al., 2020; Madry et al., 2018). This drastically increases the computation cost and has an existing gap between embedding perturbation and real adversarial feature (Zhao and Mao, 2023). In this paper, we propose two methods named **Un**superised **G**enerative **A**dversiaral **T**rainining (**UnGAT**), and **Mul**ti-task **G**enerative **A**dversarial **T**raining (**MulGAT**). Both UnGAT and MulGAT combine two transformer models, a generator and a victim model. The generator and victim models are trained simultaneously as a two-player mini-max game in the UnGAT training process. The aim of the generator is to change a clean input into an adversarial text to be against the victim model. The robustness of the victim model is gained through both adversarial text and cleaned input. Compared to previous works (Li et al., 2023a; Zhao and Mao, 2023), the major advantages of the proposed UnGAT are: **(1)** No require special input tokens (e.g. [MASK]) to perturb text for adversarial training. This helps our UnGAT and MulGAT be easily adapted to many kinds of pre-trained models. **(2)** The UnGAT does not utilize adversarial gradients, requesting a number of model's query to construct a noise per clean text, to inject embedding representation of text. Our method optimizes the generator objectives by using the victim loss gradients instead. In this paper, we conduct comprehensive experiments on the AdvGLUE benchmark (Wang et al., 2018, 2021) and dialogue generation. We validate UnGAT and MulGAT with many state-of-the-art methods (Zhu et al., 2020; Aghajanyan et al., 2021; Xu et al., 2021; Tong et al., 2022; Raman et al., 2023; Wu et al., 2023; Zhong et al., 2023; Madry et al., 2018; Ishida et al., 2020; Hoang et al., 2024) with BERT (Devlin et al., 2019), BART(Lewis et al., 2020), and T5 (Raffel

et al., 2020) models. In addition, we also perform an ablation study to show the necessity of each component and the effect of the hyper-parameters on the robustness of UnGAT and MulGAT. The empirical results on five datasets of the AdvGLUE benchmark and four dialogue generation datasets demonstrate the effectiveness of our generative adversarial training framework.

2 Related work

2.1 Pre-trained transformer models

Over the past many years, the number of pre-trained transformer language models has increased drastically, achieving remarkable performance on many NLP benchmarks. BERT (Devlin et al., 2019) is the first encoder transformer model, pre-trained by optimizing mask language model and next sentence prediction objectives. In contrast, GPT2 (Radford et al., 2019) differs from BERT in the pre-training approach, it is pre-trained on a causal language modeling objective, where the model predicts the next token in a sequence without any token masking. BART (Lewis et al., 2020) and T5 (Raffel et al., 2020) are typical pre-trained encoder-decoder transformer models, that combine the advantages of bidirectional encoding, as BERT, with autoregressive decoding for generation tasks. In contrast, ELECTRA (Clark et al., 2020) implements replaced token detection that is demonstrated more efficiently than conventional mask language modeling methods.

2.2 Adversarial training

Adversarial training is a well-known method to alleviate the brittleness of models to adversarial examples (Madry et al., 2018). One of the initial method, (Madry et al., 2018) construct continuous adversarial text based project gradient descent via model-query in multi-steps. Several works (Zhu et al., 2020; Wu et al., 2023) introduce other embedding perturbation via projected gradient descent. (Aghajanyan et al., 2021) analyzes use of trust region methods and representational collapse to keep generalizable representations during training process. Self-evolution learning (Zhong et al., 2023) proposes a token masking method and learn data distribution to improve model's performance on natural language understanding tasks. AdvFooler (Hoang et al., 2024) enhances model's robustness by randomizing the latent representation of the input through many layers at inference time. Other works (Li et al., 2023a; Zhao and Mao, 2023) focus on combining projected gradient descent with replaced token detection to enhance efficiency in the training process. In contrast to these methods, our methods use a generator to produce adversarial text for adversarial training, that can transfer adversarial features from source tasks to unseen tasks via MulGAT procedure.

2.3 Generative Adversarial Networks

Generative Adversarial Networks (GANs) (Goodfellow et al., 2014) can be used to improve the quality of synthetic data in image generation tasks. GANs include two components: (1) generator for sythetic data generation, and (2) discriminator to distinguish human data and synthetic data. In transformer model pre-training, some methods inspired by GANs are adapted to pre-train language models (Clark et al., 2020). Taking this inspiration, we propose an adversarial training framework, UnGAT and MultGAT, as a two-player min-max game. Instead of utilizing loss gradient for perturbing word embeddings as (Li et al., 2023a; Wu et al., 2023), UnGAT uses loss gradient to update generator for making adversarial input.

3 Proposed methods

3.1 Preliminaries

Our proposed adversarial training process includes two language models: a generator G with parameter ϕ and a victim f with parameter θ. Both the generator and the victim model are required to share the same vocabulary V in UnGAT. Similar to (Goodfellow et al., 2014), they are jointly trained by optimizing a two-player min-max game with the value function $V(G, f)$:

$$\min_G \max_f V(G, f) = \mathop{\mathbb{E}}_{x \sim \mathcal{D}}[f(x)] + \mathop{\mathbb{E}}_{x \sim \mathcal{D}}[f(G(x))]$$

where $f(x)$ and $f(G(x))$ are probability of correct label y of clean input and noise input respectively. We train the generator G to minimize $f(G(x))$, with the aim of making the victim model misclassify. Simultaneously, we train victim f to maximize both $f(x)$ and $f(G(x))$.

3.2 UnGAT

The overall procedure of UnGAT is shown in Algorithm 1. A generator maps an original text $x = [x_1, x_2, ..., x_n], x_i \in V$ to a noise text $\hat{x} = [\hat{x}_1, \hat{x}_2, ..., \hat{x}_n], \hat{x}_i \in V$ that satisfies $f(x, \theta) \neq$

Algorithm 1: Pseudocode of UnGAT

Require: Training dataset D, generator G, victim f, epoch n
for *epoch = 1,2,3,...,n* **do**
  for *batch* $(x, y) \subset D$ **do**
    /* Train generator */
    $\hat{t}$ = gumbel_softmax(G(x))
    Compute $\mathcal{L}_{adv}$, $\mathcal{L}_{div}$ from $\hat{t}$ and y
    Compute Eq. 4 from $\mathcal{L}_{adv}$, $\mathcal{L}_{div}$
    Update G by gradient of Eq. 4
    /* Train victim */
    $\hat{t}$ = gumbel_softmax(G(x))
    Compute Eq. 5 from $x, \hat{t}, y$
    Update f by gradient of Eq. 5
  end
end

$f(\hat{x}, \theta)$. To construct a noise text $\hat{x}$, the generator maps input text $x = [x_1, x_2, ..., x_n]$ into its output logits , $h = [h_1, h_2, ..., h_n], h \in \mathbb{R}^{|V|}$, $|V|$ is the size of vocabulary V. Then, new tokens for noise text can be easily sampled or searched via softmax distribution. However, using `sample(`h`)` or `argmax(`h`)` operators cause the problem of non-differentiability and the generator's parameter ϕ cannot be updated by gradient-based optimization because of the natural discrete of words (Nie et al., 2019). To address this issue, the Gumbel-Max trick (Maddison et al., 2017) is adopted to approximate the discrete distribution of the generator's logits. The Gumbel-Max trick samples the discrete token $\hat{t}_i$ following:

$$\hat{t}_i = softmax(\tau(h_i + g_i)) \tag{1}$$

where $\hat{t}_i$ is one-hot vector, $g_i^{(k)} = -log(-log(U_i))$ and $U_i \sim Uniform(0, 1)$. τ is the temperature, which is set to 1. Now, the noise text is $\hat{x} \sim \hat{t} = [\hat{t}_1, \hat{t}_2, ..., \hat{t}_n]$, that is differentiable with respect to h. Then, we can calculate the loss value of the adversarial text $\hat{x}$:

$$\begin{aligned} \mathcal{L}_{adv} &= loss(f(G(x), \theta), y) \\ &= loss(f(\hat{t}, \theta), y) \end{aligned} \tag{2}$$

However, solely optimize Equation 2 can make model get in stuck with text degeneration, which a word/token is generated repeatedly. It can lead our training process to sub-optimal convergence. From (Ji and Huang, 2021; Yang et al., 2022) exploration, we adapt **diversity loss** as **regularization**. Diversity loss encourage model to use all token in vocabulary equally, by minimizing the negative entropy of token's logits following the below function:

$$\begin{aligned} \hat{h} &= max(h) \\ \mathcal{L}_{div} &= \frac{1}{n} \sum_{k=1}^{n} \hat{h}_n log(\hat{h}_n) \end{aligned} \tag{3}$$

Finally, combing Equation 2, and Equation 3, our loss for fine-tuning generator is:

$$\mathcal{L}_{gen} = \lambda \mathcal{L}_{div} - (1 - \lambda)\mathcal{L}_{adv} \tag{4}$$

where λ is the diversity rate $(0.0 \leq \lambda \leq 1.0)$.

For a sample $(x, y) \in \mathcal{D}$, generator G is fixed [1], we construct an noise sample $(\hat{x}, y), \hat{x} = G(x)$, then the victim model will predict the correct label of both x and $\hat{x}$. The victim model is updated by the gradient of the below objectives:

$$\mathcal{L}_{vic} = \mathcal{L}(f(x, \theta), y) + \mathcal{L}(f(\hat{x}, \theta), y) \tag{5}$$

In construction of noise sample $\hat{x}$, we still adapt Gumbel-Max trick similar generator's training instead of argmax operator. The randomness is useful for alleviating the effect of adversarial samples (Xie et al., 2018). We do not implement any balance weight as diversity rate in victim training, because the importance of clean and noise samples is equivalent in optimization of Equation 5.

3.3 MulGAT: generator as multi-task learner

UnGAT faces some challenges: **(1) waste of memory**: UnGAT will create many copies of the pre-trained model for each tasks, **(2) lack of inheriting adversarial features**: the generator in UnGAT is learned to maximize the loss (e.g. cross-entropy loss) of victim model via back-propagation. As a result, UnGAT's generator cannot produce adversarial sequences mimicking the adversarial sentences of query-based attack tools.

Adversarial transferability (Yuan et al., 2021; Lv et al., 2023) shows that an adversarial feature can fool many types of language models on a unique task. Inspired by this exploration, we implement MulGAT so that the generator can learn about adversarial features with a small number of tasks and then transfer their knowledge to adapt unseen tasks. Figure 1 illustrates a clear comparison between UnGAT and MulGAT in generator training, both of

[1]The generator G is not updated during victim model training process.

them have a generator and victim model. We fine-tune the generator on the TCAB dataset (Asthana et al., 2022), which contains six distinct datasets in the field of hate speech detection and sentiment analysis. MulGAT's generator learns diverse adversarial features in both of word and token perturbation, produced by attacking many transformer-encoder models. Compared to UnGAT, the generator's weights are not updated during MulGAT's fine-tuning process. Consequently, MulGAT significantly reduces the computational cost, an issue of UnGAT as well as many previous adversarial training methods (Madry et al., 2018).

4 Experimental setup

4.1 Datasets

We perform experiments on the AdvGLUE benchmark (Wang et al., 2021) constructed from the GLUE benchmark (Wang et al., 2019). AdvGLUE consists of six datasets: Sentiment Analysis (SST-2), Paraphrase (QQP), Natual Language Inference (MNLI, QNLI), Textual Entailment (RTE). We use the original training dataset of each one in AdvGLUE, and then evaluate the performance of fine-tuned models on AdvGLUE by using accuracy metric (see Table 6 for more details of number of samples for training and testing). The results of our method and the baselines are reported in Table 1 with two BERT backbones. For dialogue generation, we evaluate the effectiveness of both UnGAT and MulGAt on four different datasets, including Persona-Chat (PC) (Zhang et al., 2018), Blended-Skill-Talk (BST) (Smith et al., 2020), Empathetic-Dialogue (ED) (Rashkin et al., 2019), and Conv-AI-2 (CV2) (Dinan et al., 2019). Following (Li et al., 2023b), the dialogue model takes an input that includes the entire dialogue history between two people and the current utterance of one person, and produces the utterance of the other person. See Appendix A for more details.

4.2 Metrics

We evaluate the robustness of adversarial training methods for dialogue generation via four metrics **(1)** The length of generation text **(Length)**: DGSlow (Li et al., 2023b) fools a model to generate tokens as much as possible, so shorter length is better. **(2) METEOR** (Banerjee and Lavie, 2005) is to compute the performance (i.e., the match between ground truth and model output) of model under attack. **(3) Cos** indicates the cosine similarity of original input and adversarial input made by attack tools, low Cos means that the original input is textualy different from adversarial input.**(4)** Attack success rate **(ASR)** defines the success percentage of a attack tool, following Equation 6:

$$
\begin{aligned}
ASR =& \frac{\sum_i^N \mathbb{1}[s(x,\hat{x}) > \alpha] \wedge \mathbb{1}[E(y,\hat{y})]}{N} \\
\text{s.t. } E(y,\hat{y}) =& (B(y, y_{ref}) - B(\hat{y}, y_{ref})) > \beta \\
& \vee (R(y, y_{ref}) - R(\hat{y}, y_{ref})) > \beta \\
& \vee (M(y, y_{ref}) - M(\hat{y}, y_{ref})) > \beta
\end{aligned}
\tag{6}
$$

where $s(x, \hat{x})$ denotes the cosine similarity between embeddings of original input x and crafted input $\hat{x}$. $B(.,.)$, $R(.,.)$, and $M(.,.)$ stand for BLEU(Papineni et al., 2002), ROUGE(Lin, 2004), and METEOR (Banerjee and Lavie, 2005) metric respectively. An attack fails when the semantic meaning of both x and $\hat{x}$ is irrelevant or adversarial text cannot fool the model to generate outputs that do not relate to the inputs. In our work, we set perturbation threshold α as 0.7 and performance threshold β as 0.0 for all experiments on dialogue generation. For the text classification benchmark, we mainly use accuracy to compare our proposed approaches with many strong baselines.

4.3 Attack tool

In our work, we use DGSlow (Li et al., 2023b), a state-of-the-art tool for adversarial attack on dialogue generation system. DGSlow iteratively searches and substitutes vulnerable words in order to maximize generation output length and minimize generation accuracy by gradient-based multi-objective optimization. Following (Li et al., 2023b), we use pre-trained BERT-Large-Cased model for word perturbation with the number of candidates set as 50 for mutation. We restrict maximum number of iterations to 5, meaning that no more than 5 words changed for each input sentence.

5 Results

5.1 AdvGLUE result

Table 1 shows that our proposed methods consistently outperform all baselines across different diversity datasets of Adversarial GLUE. Our average accuracy for both UnGAT and MulGAT is much higher than that of many previous methods. In experiments with BERT-Base, compared to the

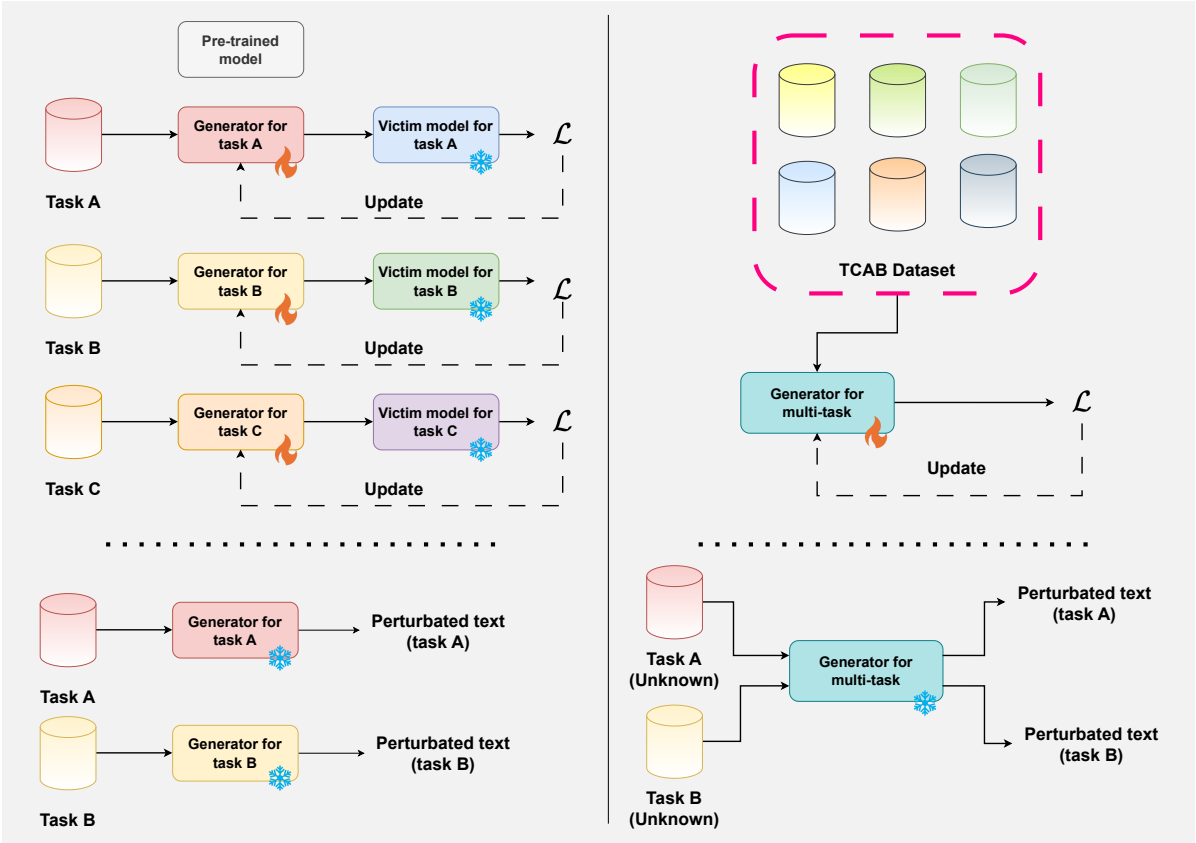

Figure 1: Illustration of training and using generator of UnGAT (left) and MulGAT (right). On the top left, each generator is trained for a specific task by unsupervised learning to maximize the loss of victim model, and then each generator perturbs its task (bottom left). In contrast, multi-task generator learns perturbation features from various tasks in TCAB, an attack benchmark dataset (on the top right). And then multi-task generator transfers knowledge of TCAB dataset to unknown dataset without victim model feedback like UnGAT (on the bottom right).

second-best method (BERT-CreAT), our proposed method improves the accuracy by 6. 1% on average. Specifically, our proposed method increases the accuracy of vanilla fine-tuning from 38.6% to 48.7% and 55.9%, demonstrating the effectiveness of adversarial training without vector perturbation with projected gradient descent. TCAB dataset contains SST-2 dataset, used to trained MulGAT's generator, we evaluate our methods by the average accuracy of MNLI, QNLI, QQP and RTE, named Avg-4. It is apparent that MulGAT can effectively transfer adversarial features to unseen tasks (i.e., from sentiment analysis and hate speech detection to natural language inference and textual entailment).

5.2 Dialogue generation result

The main results are shown in Table 2 on BART and T5 models in four benchmark datasets. First, we evaluate the robustness of using UnGAT for language models in the query-based attack scenario. For the PC task, we find that fine-tuning BART with UnGAT achieves the ASR of 28. 6%, while other state-of-the-art baselines, AdvFooler and Flooding, reach the ASR of 33. 3%. Moreover, MulGAT can prevent dialogue models from generating longer output sentences due to attack of DGSlow, this can be witnessed via experiments of fine-tuning T5 and BART with MulGAT on BST, ED dataset. In some cases, UnGAT or MulGAT can prevent the model from generating a longer output, which is one of the objectives of the attack tool (i.e., DGSlow). They achieve the shortest or second-shortest output lengths on ED and BST datasets, showing that the models produce less irrelevant output due to adversarial input. In summary, UnGAT and MulGAT reduce attack success rate significantly compared to many baselines.

6 Ablation study

6.1 Different backbones

To evaluate the effectiveness of UnGAT, we conduct the ablation study on different pre-trained transformer backbones: **(1)** BERT, **(2)** ROBERTA

Method	Adv-SST2	Adv-MNLI	Adv-QNLI	Adv-QQP	Adv-RTE	Avg	Avg-4
BERT-Base							
FT	32.3	32.6	40.1	50.8	37.0	38.6	40.1
FreeLB	31.6	33.5	45.4	51.0	42.0	40.7	43.0
BERT-MLM	32.0	27.6	43.4	48.5	45.9	39.5	41.4
BERT-CreAT	35.3	36.0	44.8	51.5	45.2	42.6	44.4
SE	28.4	23.5	42.6	42.3	33.3	34.0	35.4
MVP	28.4	28.9	36.5	52.6	39.5	37.2	39.4
UnGAT	39.2	35.6	**51.4**	51.3	**68.3**	48.7	51.7
MulGAT	**58.1**	**43.8**	47.3	**66.7**	63.4	**55.9**	**55.3**
BERT-Large							
FT	47.6	35.0	46.4	38.5	37.0	41.8	39.2
R3F	38.5	35.8	47.5	40.6	50.1	42.5	43.5
ChildTuning$_F$	34.5	33.9	47.5	40.4	42.0	39.6	41.0
ChildTuning$_D$	39.2	34.1	49.6	40.7	46.2	41.9	42.7
Match-Tuning	54.1	35.5	47.5	41.5	52.5	45.7	46.8
SE	35.1	24.7	45.3	50.0	53.1	41.6	43.3
UnGAT	52.7	37.7	52.7	59.0	**73.2**	55.1	55.7
MulGAT	**62.2**	**45.5**	**55.4**	**73.1**	60.5	**59.3**	**58.6**

Table 1: Accuracy results on the AdvGLUE benchmark. We report accuracy of each method in five datasets. Avg and Avg-4 stands for the accuracy average of five datasets and four datasets, excluding SST-2, respectively. The best results of each model is **bold**.

(Liu et al., 2020), **(3)** ELECTRA (Clark et al., 2020), and **(4)** GPT2 (Radford et al., 2019). We choose three datasets for ablation study, including SST-2, QNLI, and RTE. For fair comparison among different backbones, we use the base version of each pre-trained model. Overall, ELECTRA achieves the highest accuracy on average, the second-best is BERT. In the SST-2, QNLI and RTE tasks, the accuracies of ELECTRA are 56.8%, 62.2% and 70.8%, respectively, which are significantly higher than the results of other models.

6.2 Effect of diversity rate

We compare our proposed method with BERT-Base in different diversity rates on the SST-2 and RTE datasets. The diversity rate is from 0.1 to 0.9, exhibiting the trade-off between cross-entropy loss and diversity loss (regularization) in Equation 4. Figure 2 gives the results of the models through various diversity rates. In RTE task, we observe that the accuracy of BERT-Base on AdvGLUE peaks at 68.3% with $\lambda = 0.5$, outperforming many state-of-the-art baselines. In SST-2 task, the BERT-Base's result exhibits that the model has high accuracy on adversarial benchmark subject to $\lambda \geq 0.7$. However, experimental results for BERT-Large show that the lower diversity boosts the model's accuracy on benchmark. The accuracy of BERT-Large reaches the highest point of 52.7 at $\lambda = 0.2$, and then the accuracy decreases when $\lambda > 0.2$. For BERT-Large in the RTE task, the accuracy on the AdvGLUE benchmark fluctuates significantly between 24.4% and 73.2% over various diversity rates. On the GLUE benchmark, there are no strong change in accuracy on both BERT-Base and BERT-Large backbones. On the other hand, the results of RTE task are unstable. In conclusion, our proposed adversarial training methods are affected remarkably by the different diversity rate.

6.3 Training objectives

We compare the performance of our proposed methods via many variants. In this ablation study, we evaluate UnGAT with different settings: **(1) Freeze generator (FG)**: we do not train the generator, and only the victim model is fine-tuned during the training process.**(2) No diversity loss (w/o diversity loss)**: we set zero diversity rate ($\lambda = 0$) in Equation 4. Overall, removing some elements of our proposed methods leads to a declination of model accuracy on the test benchmark, showing via Table 3. In the experiments with BERT-Base backbone, the zero diversity rate does not have significant impact on the result of the model on adver-

Dataset	Method	BART				T5			
		ASR	Length	Cosine	ME	ASR	Length	Cosine	ME
PC	FT	47.6	19.8	0.66	28.6	38.1	12.7	0.71	23.0
	PGD	57.1	41.5	0.72	28.6	23.8	15.4	0.63	24.3
	Flooding	33.3	21.4	0.58	28.2	**14.3**	13.7	**0.61**	24.3
	FreeLB	38.1	20.9	0.65	25.7	38.1	13.2	0.68	**25.3**
	AdvFooler (En)	33.3	13.6	**0.59**	25.5	38.1	12.7	0.74	23.3
	AdvFooler (De)	53.3	**9.0**	**0.59**	21.3	38.1	**12.5**	0.80	23.3
	AdvFooler (All)	58.3	52.3	0.86	23.6	47.6	12.9	0.80	23.1
	UnGAT	**28.6**	17.8	0.66	29.9	38.1	16.0	0.73	24.6
	MulGAT	33.3	19.0	0.72	**30.2**	19.1	14.2	0.71	23.5
BST	FT	76.7	29.0	0.82	26.2	63.3	27.2	0.74	15.3
	PGD	63.3	32.4	0.73	24.3	46.7	27.2	0.68	17.3
	Flooding	70.0	67.9	0.76	22.7	50.0	24.7	0.67	16.5
	FreeLB	70.0	33.4	0.73	26.1	53.3	28.6	0.71	15.0
	AdvFooler (En)	56.7	53.0	0.73	22.7	47.7	23.4	0.84	20.6
	AdvFooler (De)	69.2	89.1	0.92	9.6	60.0	20.0	0.89	17.9
	AdvFooler (All)	60.0	27.6	0.89	11.1	50.0	**18.6**	0.87	**20.7**
	UnGAT	56.7	28.1	**0.67**	24.3	50.0	33.0	**0.66**	16.0
	MulGAT	**53.3**	**26.0**	0.72	**28.2**	**46.7**	36.6	0.67	15.4
ED	FT	60.0	54.4	0.71	11.8	60.0	32.2	0.76	8.9
	PGD	60.0	44.7	0.67	**14.6**	30.0	25.7	0.62	13.8
	Flooding	50.0	62.8	0.74	14.1	60.0	48.5	0.72	8.4
	FreeLB	30.0	143.5	**0.65**	12.8	30.0	38.0	**0.61**	7.8
	AdvFooler (En)	30.0	14.6	0.70	14.2	60.0	32.2	0.76	8.9
	AdvFooler (De)	60.0	54.7	0.71	14.2	60.0	32.2	0.76	8.9
	AdvFooler (All)	60.0	54.7	0.71	14.2	70.0	14.9	0.83	9.3
	UnGAT	30.0	**17.9**	0.76	13.5	30.0	18.7	0.76	10.0
	MulGAT	**10.0**	20.7	0.76	13.1	**20.0**	**14.8**	0.83	**15.1**
CV2	FT	43.8	17.9	0.69	**16.3**	37.5	18.3	0.64	12.6
	PGD	**12.5**	20.1	**0.45**	12.6	31.3	11.6	0.68	**17.1**
	Flooding	37.5	20.9	0.56	13.9	43.8	20.1	0.63	15.3
	FreeLB	37.5	23.0	0.62	2.0	43.8	30.4	0.62	17.8
	AdvFooler (En)	25.0	**11.4**	0.52	11.8	31.3	17.6	0.62	12.8
	AdvFooler (De)	46.2	43.6	0.86	8.4	50.0	22.1	0.79	14.6
	AdvFooler (All)	58.3	89.3	0.88	6.3	50.0	21.6	0.78	14.4
	UnGAT	**12.5**	14.6	0.72	11.9	18.8	43.4	**0.61**	10.7
	MulGAT	**12.5**	12.8	0.78	8.5	**12.5**	**11.3**	0.75	10.0

Table 2: Evaluation of adversarial training methods in four dialogue generation benchmark datasets. ASR and Length denotes the attack success rate and average generation output length, respectively. Cosine denotes the cosine similarity between original and adversarial sentences. ME stands for the METEOR metric.

sarial benchmarks. The average accuracy declines slightly by 0.5%, which means that the generator can be trained by optimizing solely the classification loss. In contrast, if there is no diversity rate in Equation 4, the results of BERT-Large are much worse than those of UnGAT. Specifically, when diversity loss is removed, the average accuracy declines to 48.0%, in which the accuracy on RTE task considerably decreases from 73.2% to 43.9%. On the other hand, freezing the generator's parameters during training steps causes deterioration in the result of BERT-Base. With BERT-Large backbone, the accuracy of generator freezing increases by 1.4% on QNLI and 7.3% on RTE datasets. This ablation study demonstrates the importance of each component of our UnGAT.

Setting	Adv-SST2	Adv-QNLI	Adv-RTE	Average	Δ ↓
BERT-Base					
UnGAT (baseline)	39.2	51.4	68.3	53.0	-
Freeze generator	33.8	46.5	58.5	46.3	6.7
W/o diversity loss	**37.8**	**51.4**	**68.3**	**52.5**	**0.5**
BERT-Large					
UnGAT (baseline)	52.7	52.7	73.2	59.5	-
Freeze generator	39.2	**48.7**	**51.2**	46.4	13.2
W/o diversity loss	**52.7**	47.3	43.9	**48.0**	**11.5**

Table 3: Result of UnGAT with different settings. Δ ↓ denotes the accuracy drop compared to UnGAT.

Model	SST2	QNLI	RTE	Avg	↑
BERT					
FT	32.3	40.1	37.0	36.5	-
UnGAT	39.2	51.4	65.9	52.0	15.5
MulGAT	58.1	47.3	63.4	56.3	19.8
RoBERTa					
FT	31.1	33.8	37.0	34.0	-
UnGAT	32.4	43.2	46.3	40.6	6.6
MulGAT	61.5	41.9	35.8	46.4	12.4
GPT2					
FT	43.2	46.6	43.2	44.3	-
UnGAT	51.4	46.0	48.8	48.7	4.4
MulGAT	54.1	51.4	50.6	52.0	7.7
ELECTRA					
FT	63.5	57.4	53.1	58	-
UnGAT	56.8	**62.2**	**70.8**	**63.3**	5.3
MulGAT	**78.4**	59.5	58.0	65.3	7.3

Table 4: Results of UnGAT and MulGAT on backbone models on SST-2, QNLI, RTE datasets of AdvGLUE.

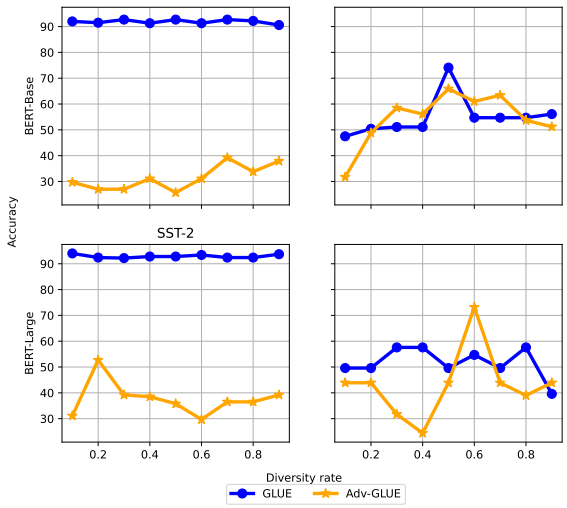

Figure 2: The accuracy of BERT-Base and BERT-Large model through various diversity rates. "GLUE" and "AdvGLUE" denote accuracy on the validation set of each task on each benchmarks.

Dataset	Token	Word	Word+Token
PC	**14.3**	28.6	19.1
BST	50.0	50.0	**12.1**
ED	50.0	30.0	**20.0**
CV2	18.6	**12.5**	**12.5**

Table 5: ASR of each adversarial feature on four benchmark datasets.

6.4 Adversarial Features

In this section, we validate the effectiveness of MulGAT via training generator with diverse adversarial features **(1) token-level**, **(2) word-level**, and **(3) combination of word-level and token-level**. We divide TCAB into two parts: token-level features and word-level features, and then fine-tune generator in each part. The ASR of each feature type is reported in Table 5, we fine-tune T5 with different adversarial features of MulGAT on PC, BST, ED and CV2 datasets. Overall, the attack success rate of MulGAT with combination of token-level and word-level features is significantly better compared to the others, except token-level features on PC task. It shows that fine-tuning language models with both word and token features has beneficial effects against query-based adversarial attacks.

7 Conclusion

In this work, we have introduced new adversarial training approaches, UnGAT and MulGAT, to improve the model performance. Both use a language model to generate adversarial examples for training, instead of aadding perturbations to continuous word embeddings by gradient descent. Our proposed methods improve many language models (e.g. BERT, BART, and T5) on adversarial benchmarks demonstrated via empirical experiments.

Limitations

Due to high computational cost, we do not conduct experiments with our proposed method to larger language models, that are raised in recent years. In UnGAT, we do not train generator under any linguistic or semantic similarity constraints, so the adversarial text of generator should be investigated in an insightful future work. Furthermore, our work limits in the context of AdvGLUE and dialogue generation benchmark, the potential tasks and benchmarks could be explored such as machine translation, machine reading comprehension tasks.

Ethics Statement

In this paper, the authors introduce UnGAT and MulGAT, adversarial training methods to improve model robustness. The methods and outcomes are intended for purely academic and constructive purposes, with no foreseeable risk of misuse or negative societal impact. We acknowledge the ACL Policy on Publication Ethics.

Acknowledgments

References

Armen Aghajanyan, Akshat Shrivastava, Anchit Gupta, Naman Goyal, Luke Zettlemoyer, and Sonal Gupta. 2021. Better fine-tuning by reducing representational collapse. In *International Conference on Learning Representations*.

Kalyani Asthana, Zhouhang Xie, Wencong You, Adam Noack, Jonathan Brophy, Sameer Singh, and Daniel Lowd. 2022. Tcab: A large-scale text classification attack benchmark. *arXiv preprint arXiv:2210.12233*.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. 2018. JAX: composable transformations of Python+NumPy programs.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. In *International Conference on Learning Representations*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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.

Emily Dinan, Varvara Logacheva, Valentin Malykh, Alexander H. Miller, Kurt Shuster, Jack Urbanek, Douwe Kiela, Arthur Szlam, Iulian Serban, Ryan Lowe, Shrimai Prabhumoye, Alan W. Black, Alexander I. Rudnicky, Jason Williams, Joelle Pineau, Mikhail S. Burtsev, and Jason Weston. 2019. The second conversational intelligence challenge (convai2). *CoRR*, abs/1902.00098.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc.

Duy Hoang, Nguyen Hung-Quang, Saurav Manchanda, Minlong Peng, Kok-Seng Wong, and Khoa Doan. 2024. Fooling the textual fooler via randomizing latent representations. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 14403–14421, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics.

Takashi Ishida, Ikko Yamane, Tomoya Sakai, Gang Niu, and Masashi Sugiyama. 2020. Do we need zero training loss after achieving zero training error? In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 4604–4614. PMLR.

Haozhe Ji and Minlie Huang. 2021. DiscoDVT: Generating long text with discourse-aware discrete variational transformer. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4208–4224, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Linyang Li, Demin Song, and Xipeng Qiu. 2023a. Text adversarial purification as defense against adversarial attacks. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 338–350, Toronto, Canada. Association for Computational Linguistics.

Yufei Li, Zexin Li, Yingfan Gao, and Cong Liu. 2023b. White-box multi-objective adversarial attack on dialogue generation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1778–1792, Toronto, Canada. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Ro{bert}a: A robustly optimized {bert} pretraining approach.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Minxuan Lv, Chengwei Dai, Kun Li, Wei Zhou, and Songlin Hu. 2023. CT-GAT: Cross-task generative adversarial attack based on transferability. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5581–5591, Singapore. Association for Computational Linguistics.

Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. 2017. The concrete distribution: A continuous relaxation of discrete random variables. In *International Conference on Learning Representations*.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2018. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*.

Weili Nie, Nina Narodytska, and Ankit Patel. 2019. RelGAN: Relational generative adversarial networks for text generation. In *International Conference on Learning Representations*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Mrigank Raman, Pratyush Maini, J Kolter, Zachary Lipton, and Danish Pruthi. 2023. Model-tuning via prompts makes NLP models adversarially robust. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9266–9286, Singapore. Association for Computational Linguistics.

Hannah Rashkin, Eric Michael Smith, Margaret Li, and Y-Lan Boureau. 2019. Towards empathetic open-domain conversation models: A new benchmark and dataset. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5370–5381, Florence, Italy. Association for Computational Linguistics.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *ArXiv*, abs/1910.01108.

Eric Michael Smith, Mary Williamson, Kurt Shuster, Jason Weston, and Y-Lan Boureau. 2020. Can you put it all together: Evaluating conversational agents' ability to blend skills. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2021–2030, Online. Association for Computational Linguistics.

Bowen Tan, Yun Zhu, Lijuan Liu, Hongyi Wang, Yonghao Zhuang, Jindong Chen, Eric Xing, and Zhiting Hu. 2024. RedCoast: A lightweight tool to automate distributed training of LLMs on any GPU/TPUs. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 3: System Demonstrations)*, pages 137–147, Mexico City, Mexico. Association for Computational Linguistics.

Shoujie Tong, Qingxiu Dong, Damai Dai, Yifan Song, Tianyu Liu, Baobao Chang, and Zhifang Sui. 2022. Robust fine-tuning via perturbation and interpolation from in-batch instances. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pages 4397–4403. International Joint Conferences on Artificial Intelligence Organization. Main Track.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing*

and Interpreting Neural Networks for NLP, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

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 *International Conference on Learning Representations*.

Boxin Wang, Chejian Xu, Shuohang Wang, Zhe Gan, Yu Cheng, Jianfeng Gao, Ahmed Hassan Awadallah, and Bo Li. 2021. Adversarial GLUE: A multi-task benchmark for robustness evaluation of language models. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Hongqiu Wu, Yongxiang Liu, Hanwen Shi, hai zhao, and Min Zhang. 2023. Toward adversarial training on contextualized language representation. In *The Eleventh International Conference on Learning Representations*.

Cihang Xie, Jianyu Wang, Zhishuai Zhang, Zhou Ren, and Alan Yuille. 2018. Mitigating adversarial effects through randomization. In *International Conference on Learning Representations*.

Runxin Xu, Fuli Luo, Zhiyuan Zhang, Chuanqi Tan, Baobao Chang, Songfang Huang, and Fei Huang. 2021. Raise a child in large language model: Towards effective and generalizable fine-tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9514–9528, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Erguang Yang, Mingtong Liu, Deyi Xiong, Yujie Zhang, Yufeng Chen, and Jinan Xu. 2022. Long text generation with topic-aware discrete latent variable model. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8100–8107, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Liping Yuan, Xiaoqing Zheng, Yi Zhou, Cho-Jui Hsieh, and Kai-Wei Chang. 2021. On the transferability of adversarial attacks against neural text classifier. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1612–1625, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. 2018. Personalizing dialogue agents: I have a dog, do you have pets too? In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2204–2213, Melbourne, Australia. Association for Computational Linguistics.

Jiahao Zhao and Wenji Mao. 2023. Generative adversarial training with perturbed token detection for model robustness. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13012–13025, Singapore. Association for Computational Linguistics.

Qihuang Zhong, Liang Ding, Juhua Liu, Bo Du, and Dacheng Tao. 2023. Self-evolution learning for discriminative language model pretraining. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 4130–4145, Toronto, Canada. 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 *International Conference on Learning Representations*.

A Implementation details

A.1 Backbones

The detailed pre-trained models for generator and victim model in all experiments are showed in Table 7. In each task, we set different hyper-parameter, the detailed hyper-parameter is provided in Table 8. For BERT and ROBERTA backbones, we use distil version of them (Sanh et al., 2019) for generator to reduce computational cost. In experiments with four benchmark datasets for dialogue generation, we mainly use BART-Base and T5-Small checkpoint. In experiments with MulGAT, We fine-tune these generative language models with learning rate 5e-5, linear learning rate scheduler within 100 epochs. For UnGAT, we use a fixed diversity rate 0.5 and fine-tune these models within 15 epochs

Dataset	Train	Dev	Test
SST-2	67.3k	0.87k	0.15k
MNLI	393k	9.82k	0.12k
QNLI	116k	5.46k	0.15k
QQP	795k	40.4k	0.08k
RTE	2.49k	0.28k	0.08k

Table 6: Data statistics

Generator	Victim
DistilBERT-Base-Uncased	BERT-Base-Uncased
DistilBERT-Base-Uncased	BERT-Large-Uncased
DistilRoBERTa-Base	RoBERTa-Base
GPT2	GPT2
ELECTRA-Base	ELECTRA-Base

Table 7: Pre-trained models of generator and victim

A.2 Baselines

We fine-tune pre-trained BERT models, including BERT-Base and BERT-Large on the GLUE benchmark (Wang et al., 2019), by Huggingface Transformers (Wolf et al., 2020) and Pytorch (Paszke et al., 2019). To evaluate the robustness of our proposed method, we compare our methods with the following baseline: **Vanilla fine-tuning (FT)**: We fine-tune pre-trained models and evaluate following (Wang et al., 2021) on the GLUE benchmark. **Prompt-based fine-tuning (MVP)** (Raman et al., 2023): finds that fine-tune model via prompts help model against adversarial examples. **Self-Evolution Learning (SE)** (Zhong et al., 2023) continues pretraining masked language model (e.g., BERT) with linguistically-motivated masking strategies and then fine-tune these models on downstream datasets. **CreAT** (Wu et al., 2023) (BERT-CreAT) is an adversarial training that finding perturbations based on the deviation of output distribution and contextualized representation. Besides, BERT-MLM pre-trains BERT on subsets of C4 dataset (Raffel et al., 2020) and then fine-tunes on downstream datasets. **R3F** (Aghajanyan et al., 2021) fine-tunes language models in trust region to alleviate the degradation of generalizable representations of language models. **ChildTuning**$_F$ and **ChildTuning**$_D$ (Xu et al., 2021) update the subset of model parameters by multiplying the gradients corresponding to the model parameter by binary matrix (mask) for generalizable fine-tuning. **Match-Tuning** (Tong et al., 2022) determines how to utilize the in-batch instances during the whole training process.

For the generation of benchmark data sets, we consider using several state-of-the-art adversarial training methods. Projected Gradient Descent **(PGD)** (Madry et al., 2018) is implemented by injecting word embeddings with adversarial perturbations in the embedding space. **(Flooding)** (Ishida et al., 2020) avoids training loss reaching zero during training by maintaining training loss value higher than flood level. Free large batch adversarial training using projected gradient descent **(FreeLB)** (Zhu et al., 2020) also adds adversarial perturbations, generated in a region around input samples, to embeddings. (Hoang et al., 2024) proposes **(AdvFooler)**, a method to randomize the latent representation of the input and layer's output to prevent query-based attack tools from finding important words in input. We extend AdvFooler to sequence-to-sequence (seq2seq) models (e.g., BART, T5) for generation tasks, resulting three baseline **(AdvFooler(En), AdvFooler(De), AdvFooler(All))**. AdvFooler(En) indicates that we just use AdvFooler for Encoder block of seq2seq model. Similarly, AdvFooler(De) and AdvFooler(All) are used for Decoder and both Encoder-Decoder block of seq2seq model respectively.

All experiments are conducted on an A100 GPU and two T4 GPUs.

A.3 Multi-task generator for MulGAT

We adapt encoder-decoder Transformer architecture rather than encoder-only and decoder-only model architecture for original examples to adversarial. The encoder will capture the feature and information of input sentence, and then the decoder generate the output with variable length. This is easily adapt model for different kind of perturbation: token-level perturbation, and word-level perturbation.

The TCAB dataset contains more than 1.4 million samples, including word-level adversarial samples and token-level adversarial samples. Similar to (Lv et al., 2023), we fine-tune BART-Base on TACB with learning rate 0.001 and AdamW optimizer (Loshchilov and Hutter, 2019) within 20 epochs. To speed up the fine-tuning process, we use Jax (Bradbury et al., 2018) and RedCoast (Tan et al., 2024) framework to fine-tune BART on TPU VM V3 (8 cores). The batch size per core is 64, so the total batch size is 512. The total fine-tuning time is about 24 hours. During inference, there are no tools used to restrict grammar or limit the bound for adversarial texts.

	BERT-Base					BERT-Large				
Hyper-parameter	SST-2	MNLI	QNLI	QQP	RTE	SST-2	MNLI	QNLI	QQP	RTE
Learning rate	2e-5									
Batch size	32									
Optimizer	AdamW (Loshchilov and Hutter, 2019)									
Max length	128	256	128	320	320	128	256	128	320	320
Epoch	9	1	6	2	7	5	3	3	1	3
λ	0.7	0.5	0.5	0.5	0.6	0.2	0.5	0.3	0.3	0.6

Table 8: The hyper-parameter for BERT-Base and BERT-Large in our experiments