

# Balanced Adversarial Training: Balancing Tradeoffs between Fickleness and Obstinance in NLP Models

Hannah Chen, Yangfeng Ji, David Evans

Department of Computer Science

University of Virginia

Charlottesville, VA 22904

{yc4dx,yangfeng,evans}@virginia.edu

## Abstract

Traditional (*fickle*) adversarial examples involve finding a small perturbation that does not change an input’s true label but confuses the classifier into outputting a different prediction. Conversely, *obstinate* adversarial examples occur when an adversary finds a small perturbation that preserves the classifier’s prediction but changes the true label of an input. Adversarial training and certified robust training have shown some effectiveness in improving the robustness of machine learnt models to fickle adversarial examples. We show that standard adversarial training methods focused on reducing vulnerability to fickle adversarial examples may make a model more vulnerable to obstinate adversarial examples, with experiments for both natural language inference and paraphrase identification tasks. To counter this phenomenon, we introduce *Balanced Adversarial Training*, which incorporates contrastive learning to increase robustness against both fickle and obstinate adversarial examples.

## 1 Introduction

Interpreted broadly, an adversarial example is an input crafted intentionally to confuse a model. Most research on adversarial examples, however, focuses on a definition of an adversarial example as an input that is constructed by making minimal perturbations to a normal input that change the model’s output, assuming that the small perturbations preserve the original true label (Goodfellow et al., 2015). Such adversarial examples occur when a model is overly influenced by small changes in the input. Attackers can also target the opposite objective—to find inputs with minimal changes that change the ground truth label but for which the model retains its prior prediction (Jacobsen et al., 2019b).

Various names have been used in the research literature for these two types of adversarial examples including perturbation or sensitivity-based and invariance-based examples (Jacobsen et al.,

2019b,a), and over-sensitive and over-stable examples (Niu and Bansal, 2018; Kumar and Boulanger, 2020). To avoid confusions associated with these names, we refer them as *fickle adversarial examples* (the model changes its output too easily) and *obstinate adversarial examples* (the model doesn’t change its output even though the input has changed in a way that it should).

In NLP, synonym-based word substitution is a common method for constructing fickle adversarial examples (Alzantot et al., 2018; Jin et al., 2020) since synonym substitutions are assumed to not change the true label for an input. These methods target a model’s weakness of being invariant to certain types of changes which makes its predictions insufficiently responsive to small input changes. Attacks based on antonyms and negation have been proposed to create obstinate adversarial examples for dialogue models (Niu and Bansal, 2018).

Adversarial training is considered as the most effective defense strategy yet found against adversarial examples (Madry et al., 2018; Goodfellow et al., 2016). It aims to improve robustness by augmenting the original training set with generated adversarial examples in a way that results in decision boundaries that correctly classify inputs that otherwise would have been fickle adversarial examples. Adversarial training has been shown to improve robustness for NLP models (Yoo and Qi, 2021). Recent works have also studied certified robustness training which gives a stronger guarantee that the model is robust to all possible perturbations of a given input (Jia et al., 2019; Ye et al., 2020).

While prior work on NLP robustness focuses on fickle adversarial examples, we consider both fickle and obstinate adversarial examples. We then further examine the impact of methods designed to improve robustness to fickle adversarial examples on a model’s vulnerability to obstinate adversarial examples. Recent work in the vision domain demonstrated that increasing adversarial robustness of im-

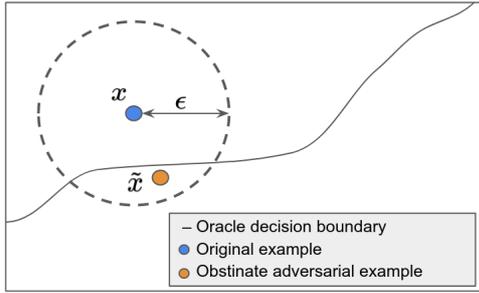


Figure 1: Distance-oracle misalignment (Tramer et al., 2020). While the model is trained to be robust to  $\epsilon$ -bounded perturbation, it becomes too invariant to small changes in the example (obstinate example  $\tilde{x}$ ) that lie on the other side of the oracle decision boundary.

age classification models by training with fickle adversarial examples may increase vulnerability to obstinate adversarial examples (Tramer et al., 2020). Even in cases where the model certifiably guarantees that no adversarial examples can be found within an  $L_p$ -bounded distance, the norm-bounded perturbation does not align with the ground truth decision boundary. This *distance-oracle misalignment* makes it possible to have obstinate adversarial examples located within the same perturbation distance, as depicted in Figure 1. In text, fickle examples are usually generated with a cosine similarity constraint to encourage the representations of the original and the perturbed sentence to be close in the embedding space. However, this similarity measurement may not preserve the actual semantics (Morris et al., 2020) and the model may learn poor representations during adversarial training.

**Contributions.** We study fickle and obstinate adversarial robustness in NLP models with a focus on synonym and antonym-based adversarial examples (Figure 2 shows a few examples). We evaluate both kinds of adversarial robustness on natural language inference and paraphrase identification tasks with BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019) models. We find that there appears to be a tradeoff between robustness to synonym-based and antonym-based attacks. We show that while certified robust training increases robustness against synonym-based adversarial examples, it increases vulnerability to antonym-based attacks (Section 3). We propose a modification to robust training, *Balanced Adversarial Training* (BAT), which uses a contrastive learning objective to help mitigate the distance misalignment problem by learning from both fickle and obstinate examples

(Section 4). We implement two versions of BAT with different contrastive learning objectives, and show the effectiveness in improving both fickleness and obstinacy robustness (Section 4.2).

## 2 Constructing Adversarial Examples

We consider a classification task where the goal of the model  $f$  is to learn to map the textual input  $x$ , a sequence of words,  $x_1, x_2, \dots, x_L$ , to its ground truth label  $y \in \{1, \dots, c\}$ . We assume there is a labeling oracle  $\mathcal{O}$  that corresponds to ground truth and outputs the true label of the given input. We focus on word-level perturbations where the attacker substitutes words in the original input  $x$  with words from a known perturbation set (which we show how we construct it in the following sections). The goal of the attacker is to find an adversarial example  $\tilde{x}$  for input  $x$  such that the output of the model is different from what human would interpret, i.e.  $f(\tilde{x}) \neq \mathcal{O}(\tilde{x})$ .

### 2.1 Fickle Adversarial Examples

For a given input  $(x, y)$  correctly classified by model  $f$  and a set of allowed perturbed sentences  $\mathcal{S}_x$ , an *fickle adversarial example* is defined as an input  $\tilde{x}_f$  such that:

1.  $\tilde{x}_f \in \mathcal{S}_x$
2.  $f(\tilde{x}_f) \neq f(x)$
3.  $\mathcal{O}(\tilde{x}_f) = \mathcal{O}(x)$

There are many different methods for finding fickle adversarial examples. The most common way is to use synonym word substitutions where the target words are replaced with similar words found in the word embedding (Alzantot et al., 2018; Jin et al., 2020) or use known synonyms from WordNet (Ren et al., 2019). Recent work has also explored using masked language models to generate word replacements (Li et al., 2020; Garg and Ramakrishnan, 2020; Li et al., 2021).

We adopt the synonym word substitution method as in Ye et al. (2020). For each word  $x_i$  in an input  $x$ , we create a synonym set  $S_{x_i}$  containing the synonym words of  $x_i$  including itself.  $\mathcal{S}_x$  is then constructed by a set of sentences where each word in  $x$  can be replaced by a word in  $S_{x_i}$ . We consider the case where the attacker does not have a constraint on the number of words that can be perturbed for each input, meaning the attacker can perturb up to  $L$  words which is the length of  $x$ .

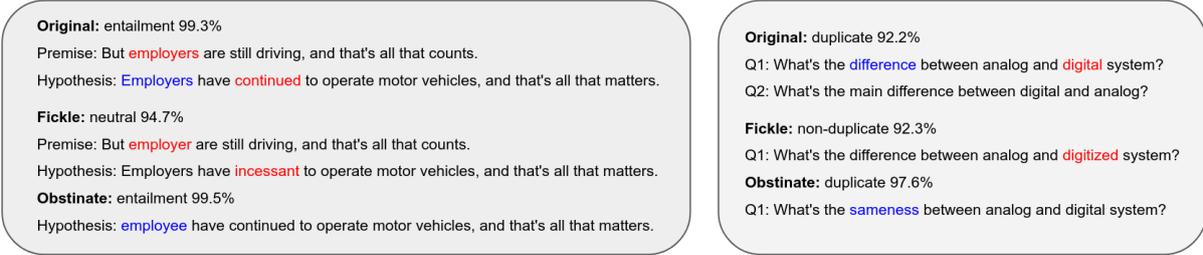


Figure 2: Fickle and obstinate adversarial examples for BERT model fine-tuned on natural language inference (left) and paraphrase identification (right) tasks. Words in red are substituted with their synonyms and words in blue are replaced by their antonyms.

The underlying assumption for fickle examples to work is that the perturbed sentence  $\tilde{x}_f \in S_x$  should have the same ground truth label as the original input  $x$ , i.e.  $\mathcal{O}(\tilde{x}_f) = \mathcal{O}(x) = f(x)$ . However, common practice for constructing fickle examples does not guarantee this is true. Swapping a word with its synonym may change the semantic meaning of the example since even subtle changes in words can have a big impact on meaning, and a word can have different meanings in different context. For instance, “the whole *race* of human kind” and “the whole *competition* of human kind” describe different things. Nonetheless, previous human evaluation has shown that synonym-based adversarial examples still retain the same semantic meaning and label as the original texts most of the time (Jin et al., 2020; Li et al., 2020).

## 2.2 Obstinate Adversarial Examples

For a given input  $(x, y)$  correctly classified by model  $f$  and a set of allowable perturbed sentences  $\mathcal{A}_x$ , an *obstinate adversarial example* is defined as an input  $\tilde{x}_o$  such that:

1.  $\tilde{x}_o \in \mathcal{A}_x$
2.  $f(\tilde{x}_o) = f(x)$
3.  $\mathcal{O}(\tilde{x}_o) \neq \mathcal{O}(x)$

While it is challenging to construct obstinate adversarial examples automatically for image classifiers (Tramer et al., 2020), we are able to automate the process for NLP models. We use a similar antonym word substitution strategy as proposed by Niu and Bansal (2018) to construct obstinate adversarial examples. Similar to synonym word substitutions, for each word  $x_i$  in an input  $x$ , we construct an antonym set  $A_{x_i}$  that consists of the antonyms of  $x_i$ . Since we would like to change the semantic meaning of the input in a way that is likely to flip its label for the task, the attacker is

only allowed to perturb one word with its antonym for each sentence.

The way we construct obstinate adversarial examples may not always satisfy the assumption where the ground truth label of the obstinate example would be different from the original input. The substituted word may not affect the semantic meaning of the input depending on the task. For example, in natural language inference, changing “the weather is *great*, we should go out and have fun” to “the weather is *bad*, ...” does not effect the entailment relationship with “we should have some outdoor activities” since the main argument is in the second part of the sentence. However, we find that antonym substitutions are able to change the semantic meaning of the text most of the time and we choose two tasks that are most likely to change the label under antonym-based attack.

## 3 Robustness Tradeoffs

Normally, adversarial defense methods only target fickle adversarial examples, so there is a risk that such methods increase vulnerability to obstinate adversarial examples. According to the distance-oracle misalignment assumption (Tramer et al., 2020) as depicted in Figure 1, the distance measure for finding adversarial examples and labeling oracle  $\mathcal{O}$  is misaligned if we have  $\mathcal{O}(\tilde{x}_f) = \mathcal{O}(x) = y$  and  $\mathcal{O}(\tilde{x}_o) \neq \mathcal{O}(x)$ , but  $dist(x, \tilde{x}_f) > dist(x, \tilde{x}_o)$ .

### 3.1 Setup

Our experiments are designed to test our hypothesis that optimizing adversarial robustness of NLP models using only fickle examples deteriorates the model’s robustness on obstinate adversarial examples. We use the SAFER certified robust training method proposed by Ye et al. (2020). The idea is to train a smoother model by randomly perturbing

the sentences with words in the synonym substitution set at each training iteration. While common IBP-based certified robust training methods do not scale well onto large pre-trained language models (Jia et al., 2019; Huang et al., 2019), SAFER is a structure-free approach that can be applied to any kind of model architectures. In addition, it gives stronger robustness than traditional adversarial training method (Yoo and Qi, 2021).

We train BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019) models on two different tasks with SAFER training for 15 epochs. We then test the attack success rate for both fickleness and obstinacy attacks at each training epoch. We use the same perturbation method as described in Section 2.1 for both the training and the attack. For each word, the synonym perturbation set is constructed by selecting the top  $k$  nearest neighbors with a cosine similarity constraint of 0.8 in GLOVE embeddings (Pennington et al., 2014), and the antonym perturbation set consists of antonym words found in WordNet (Miller, 1995). We follow the method of Jin et al. (2020) for finding fickle adversarial examples by using word importance ranking and Part-of-Speech (PoS) and sentence semantic similarity constraints as the search criteria. We replace words from the ones with the highest word importance scores to the ones with the least and make sure the new substituted words have the same PoS tags as the original words. For antonym attack, we also use word importance ranking and PoS to search for word substitutions. For comparison, we set up baseline models with normal training on the original training sets.

### 3.2 Tasks

We choose two different tasks from the GLUE benchmark (Wang et al., 2018) that are good candidates for the antonym attack. Antonym-based attacks work well on these tasks since both tasks consist of sentence pairs and changing a word to an opposite meaning is likely to break the relationship between the pairs.

**Natural Language Inference.** We experiment with Multi-Genre Natural Language Inference (MNLI) dataset (Williams et al., 2018) which contains a premise-hypothesis pair for each example. The task is to identify the relation between the sentences in a premise-hypothesis pair and determine whether the hypothesis is true (*entailment*), false (*contradiction*) or undetermined (*neutral*) given the

premise. We consider the case where both premise and hypothesis can be perturbed, but only one word from either premise or hypothesis can be substituted for antonym attack. We exclude examples with a *neutral* label when constructing obstinate adversarial examples since antonym word substitutions may not change their label to a different class.

**Paraphrase Identification.** We use Quora Question Pairs (QQP) (Iyer et al., 2017) which consists of questions extracted from Quora. The goal of the task is to identify duplicate questions. Each question pair is labeled as duplicate or non-duplicate. For our antonym attack strategy, we only target the duplicate class since antonym word substitutions are unlikely to flip an initially non-duplicate pair into a duplicate.

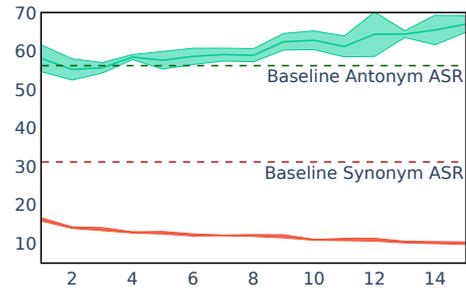
We also conducted experiments using the Wiki Talk Comments (Wulczyn et al., 2017) dataset, a dataset for toxicity detection, by adding or removing toxic words for creating obstinate examples. However, we found adding toxic words can reach almost 100% attack success rate, so there did not seem to be an interesting tradeoff to explore for available models for this task, and we do not include it in our results.

### 3.3 Results

We visualize the attack success rates for fickleness (synonym attack) and obstinacy (antonym attack) attacks in Figure 3. The results are consistent with our hypothesis that optimizing adversarial robustness of NLP models using only fickle examples can result in models that are more vulnerable to obstinacy attacks. Robustness training for the BERT model on MNLI improves fickleness robustness, reducing the synonym attack success rate from 36% to 11% (a 69% decrease) after training for 15 epochs (Figure 3a), but antonym attack success rate increases from 56% to 63% (a 13% increase). The antonym attack success rate increases even more for the RoBERTa model (Figure 3b), increasing from 56% to 67% (a 20% increase) while the synonym attack success rate decreases from 31.2% to 10% (a 68% decrease). The RoBERTa model is pre-trained to be more robust than the BERT model with dynamic masking, which perhaps explains the difference. We observe a robustness tradeoff for QQP dataset as well (see Appendix A.1). In addition, the fickle adversarial training does not sacrifice the performance on the original examples

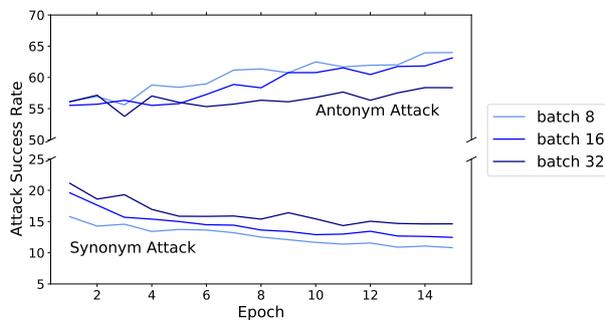


(a) MNLi (BERT)

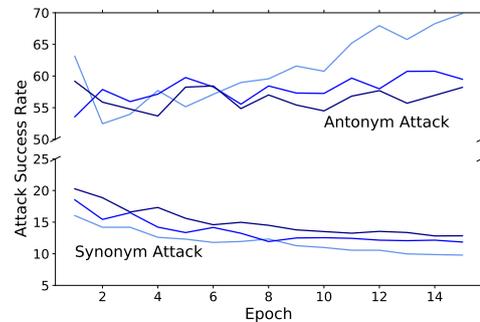


(b) MNLi (RoBERTa)

Figure 3: Fickleness and obstinacy tradeoff where fickleness attack success rate increases as obstinacy attack success rate decreases. The figure shows the results on MNLi matched validation set with average and standard deviation across three different runs. Dash lines show the synonym/antonym attack success rate on baseline model with normal training.



(a) MNLi (BERT)



(b) MNLi (RoBERTa)

Figure 4: The synonym and antonym attack success rate at each SAFER training epoch with varying batch size. When the model is trained with smaller batch size, the synonym attack success rate is lower and the antonym success rate is higher.

and increases consistently throughout training (see Figure 9 in the appendix).

**Impact of Batch Size.** We experiment with different batch sizes for fickle-based robust training. Figure 4 shows the results on MNLi dataset. When the model is trained with a smaller batch size, the synonym attack success rate becomes lower, but the antonym success rate gets higher. This means that the model may overfit on the fickle examples due to smaller training batch size, exacerbating the impact of the unbalanced adversarial training. Similar observation is found on the QQP task (see Figure 8 in the appendix). We found similar evidence on the evaluation accuracy on the original validation set (see Figure 9 in the appendix). While models with smaller batch sizes converge faster, they lead to lower performance and poorer generalization. In Appendix C.4, we show that our proposed method is not affected by the training batch size.

## 4 Balanced Adversarial Training

In previous section, we argued that the tradeoff between fickleness and obstinacy can be attributed to distance-oracle misalignment. This section proposes and evaluates a modification to adversarial training that balances both kinds of adversarial examples.

### 4.1 Approach

The most intuitive way to make the semantic distance in the representation space align better with human perception is to move the fickle example closer to the original input and push the original input apart from the obstinate example in the representation space. This goal matches the objective of contrastive learning, a type of self-supervised learning that learns representations with positive (similar) examples close together and negative (dissimilar) examples far apart (Hadsell et al., 2006;

Model	Method	Eval Acc (%)	Antonym ASR (%)	Synonym ASR (%)
BERT	Normal Training	84.01 ±0.32	57.48 ±0.14	36.63 ±0.34
	A2T	84.44	56.51	21.67
	SAFER	83.81 ±0.40	63.33 ±1.55	10.61 ±0.17
	BAT-Pairwise	84.03 ±0.21	35.47 ±1.51	26.70 ±0.78
	BAT-Triplet	84.66 ±0.12	34.25 ±2.19	25.70 ±0.13
RoBERTa	Normal Training	87.74 ±0.09	55.86 ±0.47	30.86 ±0.50
	A2T	86.98	56.84	19.78
	SAFER	86.18 ±0.18	67.15 ±2.17	10.00 ±0.34
	BAT-Pairwise	87.23 ±0.38	38.68 ±1.61	27.29 ±0.20
	BAT-Triplet	87.44 ±0.12	33.02 ±0.81	27.54 ±0.45

Table 1: Balanced Adversarial Training evaluation results on MNLI matched validation set. Results shown with standard deviations are average across three different runs.

Schroff et al., 2015). Positive examples are usually generated with data augmentation such as spatial transformation, and negative examples are sampled from other examples (Chen et al., 2020).

We adapt contrastive learning to balance adversarial training by treating fickle adversarial examples as positive examples and obstinate adversarial examples as negative examples. The idea is to minimize the distance between the positive pairs and maximize the distance between the negative pairs. We construct positive pairs by pairing the original input with a corresponding fickle example, and negative pairs as the original input paired with an obstinate example. We generate fickle examples by applying synonym transformations, and obstinate examples by applying antonym transformations.

We combine normal training with a contrastive learning objective and experiment with two different approaches for contrastive loss: pairwise and triplet loss. While recent contrastive learning incorporates multiple positive and negative examples for each input, we use these two methods as they consider the simplest case where only a positive and a negative example is needed for each input. Similarly to SAFER certified robust training, we use an augmented approach without querying the model to check if the attack succeeds. We choose this approach over traditional adversarial training since it is computationally less expensive.

Given an input  $(x, y)$ , we generate an example  $\tilde{x}_o$  by applying synonym perturbations and an example  $\tilde{x}_u$  by applying antonym perturbations. Let  $d(x_1, x_2)$  denote the distance measure between  $x_1$  and  $x_2$  in the representation space.

**BAT-Pairwise.** For the pairwise approach, we inde-

pendently optimize the distance for the fickle pair  $(x, \tilde{x}_f)$  and the obstinate pair  $(x, \tilde{x}_o)$ :

$$\mathcal{L}^{BAT_{pair}} = \mathcal{L}_{ML} + \mathcal{L}_{pair}$$

$$\mathcal{L}_{ML} = \log f(y | x)$$

$$\mathcal{L}_{pair} = \alpha d(x, \tilde{x}_f) + \beta \max(0, m - d(x, \tilde{x}_o))$$

The hyperparameters  $\alpha$  and  $\beta$  control the weighting of the fickle and obstinate pairs, and  $m$  is the margin. The  $\mathcal{L}_{pair}$  loss term is designed to minimize the distance to the fickle adversarial example and maximize the distance to the obstinate adversarial example. The margin  $m$  penalizes the model when the obstinate example within  $m$  distance of the original input ( $d(x, \tilde{x}_o) < m$ ). We use cosine similarity for distance measure (ranges from 0 to 1), and set the margin as 1 as we find it gives the best performance (see Appendix C.2.2). For the case where we are unable to find a valid fickle and obstinate adversarial example, we set the corresponding term, either  $d(x, \tilde{x}_f)$  or  $m - d(x, \tilde{x}_o)$ , to 0.

**BAT-Triplet.** For the triplet approach, the original input  $x$  acts as an anchor and a triplet,  $(x, \tilde{x}_f, \tilde{x}_o)$ , is considered instead of pairs. The triplet loss aims to make the distance between the obstinate pair larger than the distance between the fickle pair, with at least a margin  $m$ :  $d(x, \tilde{x}_o) > d(x, \tilde{x}_f) + m$ . The training loss can be formalized as:

$$\mathcal{L}^{BAT_{triplet}} = \mathcal{L}_{ML} + \lambda \mathcal{L}_{triplet}$$

$$\mathcal{L}_{triplet} = \max(0, d(x, \tilde{x}_f) + (m - d(x, \tilde{x}_o)))$$

where the hyperparameter  $\lambda$  controls the weight of the contrastive loss term. Similarly to the pairwise loss, if no fickle and obstinate example is available, we mask out  $d(x, \tilde{x}_f)$  or  $m - d(x, \tilde{x}_o)$  in  $\mathcal{L}_{triplet}$ .

Model	Method	Eval Acc (%)	F1	Antonym ASR (%)	Synonym ASR (%)
BERT	Normal Training	90.78 $\pm$ 0.12	87.67 $\pm$ 0.12	44.94 $\pm$ 1.24	21.16 $\pm$ 0.36
	A2T	90.70	87.52	36.27	14.71
	SAFER	90.85 $\pm$ 0.19	87.65 $\pm$ 0.23	50.10 $\pm$ 0.44	4.83 $\pm$ 0.15
	BAT-Pairwise	90.31 $\pm$ 0.03	86.97 $\pm$ 0.09	22.46 $\pm$ 1.08	16.00 $\pm$ 0.73
	BAT-Triplet	90.82 $\pm$ 0.11	87.79 $\pm$ 0.04	16.24 $\pm$ 1.41	15.73 $\pm$ 0.09
RoBERTa	Normal Training	91.18 $\pm$ 0.09	88.29 $\pm$ 0.12	40.52 $\pm$ 0.71	18.9 $\pm$ 0.12
	A2T	91.14	88.04	42.42	13.04
	SAFER	91.26 $\pm$ 0.10	88.27 $\pm$ 0.15	46.07 $\pm$ 1.44	4.87 $\pm$ 0.45
	BAT-Pairwise	90.19 $\pm$ 0.29	86.94 $\pm$ 0.40	15.29 $\pm$ 2.62	16.58 $\pm$ 0.44
	BAT-Triplet	91.04	88.21	13.02	16.89

Table 2: Balanced Adversarial Training evaluation results on QQP validation set.

We show the training details and how we find the best hyperparameters in Appendix C.2.

## 4.2 Results

Table 1 shows BAT training results on the MNLI validation sets. We use normal training as the non-robust baseline, and include two robust baselines: certified robust training (SAFER), and traditional adversarial training (A2T) (Yoo and Qi, 2021). Balanced Adversarial Training increases the model’s adversarial robustness against both antonym and synonym attacks, while preserving its performance on the original validation set.

While both of the robust baselines (SAFER and A2T), which only consider fickle adversarial examples, perform best when evaluated solely based on fickleness robustness, they are more vulnerable to obstinate adversarial examples. We found that BAT-Triplet performs better than BAT-Pairwise in terms of improving robustness against antonym attacks. With BAT-Triplet, the antonym attack success rate on BERT decreases from 57% to 34% (a 40% decrease) comparing to normal training, and the synonym attack success rate decreases from 36% to 26% (a 28% decrease).

Results for the QQP dataset are shown in Table 2. While the antonym attack success rates drop more than half (around 67% decrease) after BAT training, the synonym attack success rate has a 24% decrease on BERT and only 10% on RoBERTa, as the synonym attack success rate is already low on the model with normal training.

## 4.3 Representation Analysis

We compare the learned representations of models trained with BAT to normal training and SAFER. We sample 500 examples from MNLI dataset (ex-

cluding the neutral class) and apply synonym and antonym perturbations for each input. We then project the model representations before the last classification layer to 2 dimensional space with t-SNE (van der Maaten and Hinton, 2008) and visualize the results in Figure 5.

When training with normal training or SAFER, we can see that both fickle and obstinate adversarial examples are fairly close to the original examples. However, with BAT-Pairwise or BAT-Triplet, obstinate examples are pushed further away from both original and fickle examples. This matches with BAT’s training goal where the distance between obstinate and original examples is maximized and the distance between fickle and original examples is minimized. This also shows how BAT is able to fix the distance-oracle misalignment, making the semantic distance in the representation space aligns better with human perception, and further improve robustness against both types of adversarial examples.

## 5 Related Work

Compared to fickle adversarial examples, obstinacy has been less studied in NLP as well as other domains. Feng et al. (2018) delete words iteratively from the input to create examples that appear rubbish to human but retain the model’s prediction with high confidence. Welbl et al. (2020) use Part-of-Speech and Name Entity based perturbations against reading comprehension models. Niu and Bansal (2018) study both types of attack strategies for dialogue models. They create obstinate adversarial examples by substituting words with antonyms or adding negation words to the input.

Our work is the first to study tradeoffs between

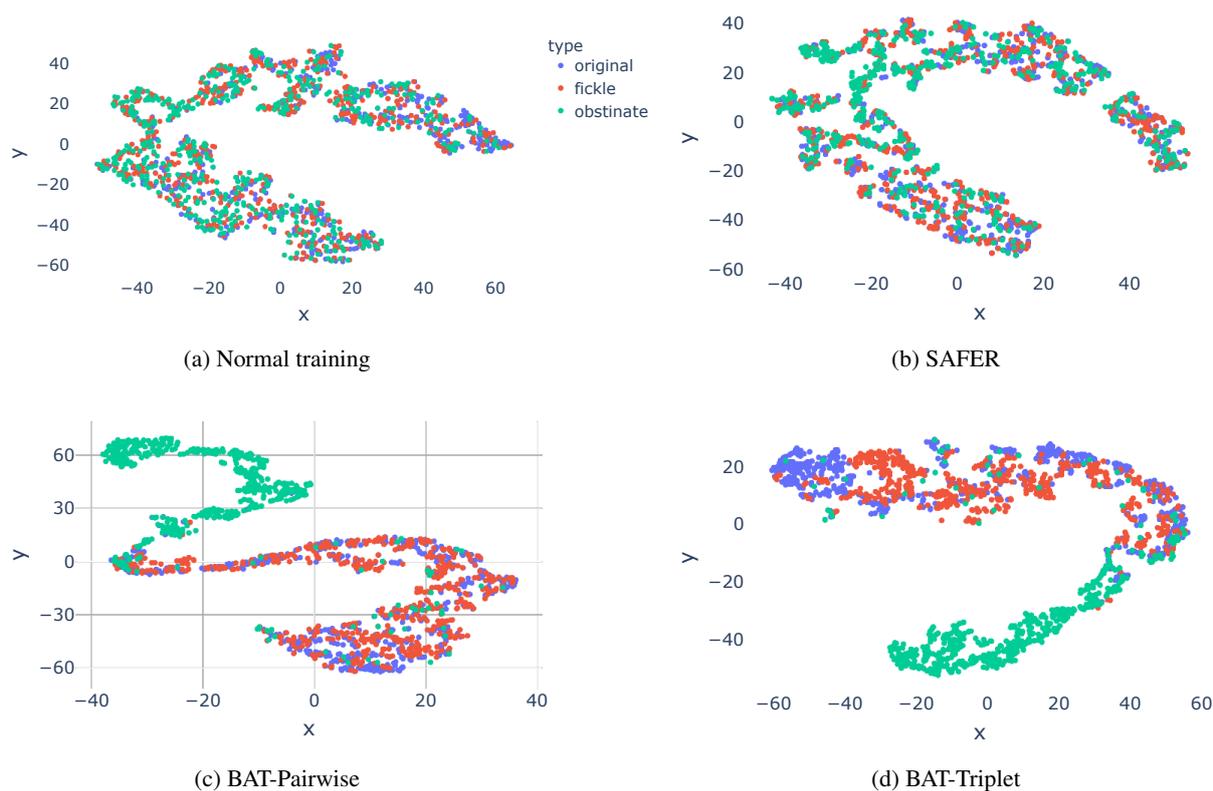


Figure 5: 2D projection of model representation for RoBERTa MNL models trained with normal training, certified robust training with fickle adversarial examples (SAFER), BAT-Pairwise, and BAT-Triplet.

fickle and obstinate adversarial examples in NLP, but a few previous works have considered these tradeoffs in the vision domain. [Jacobsen et al. \(2019b\)](#) show that adversary can not only target the model’s excessive sensitivity but its excessive invariance to small changes in the input. They propose an alternative training objective based on information theory to make the model less invariant to semantically meaningful changes. [Tramer et al. \(2020\)](#) study the tradeoff between the two types of adversarial examples for image classifiers. They show that data augmentation can help increase robustness against obstinacy attacks, but is not sufficient to impede both types of attacks. Our work differs in that we propose a new adversarial training method that improves model robustness against both types of adversarial examples. In addition, unlike images where human inspection is usually required to check whether the perturbed pixels would change the true label of the image, we are able to automate the process of generating obstinate examples for text.

Recent work introduce contrastive learning for image classifiers in the adversarial learning set-

ting where an fickle adversarial augmentation is used to generate positive examples and negative examples are sampled from other images. [Kim et al. \(2020\)](#) generate diverse positive examples by launching instance-wise attack on augmented images and show that it improves model’s fickleness robustness. [Ho and Nvasconcelos \(2020\)](#) create challenging positive pairs by using the gradients of the contrastive loss to generate fickle adversarial examples and they show that it improves model performance.

## 6 Conclusion

We demonstrate the tradeoff between vulnerability to synonym-based (fickle) and antonym-based (obstinate) adversarial examples for NLP models and show that increasing robustness against synonym based attacks also increases vulnerability to antonym-based attacks. To manage this tension, we introduce a new adversarial training method, BAT, which targets the distance-oracle misalignment problem and can help balance the fickleness and obstinacy in adversarial training.

## Limitations

We showed robustness tradeoffs exist between synonym and antonym-based adversarial examples. Since there are numerous ways to construct adversarial examples for NLP models, further investigation is needed to show if this holds true for any kind of fickleness and obstinacy attacks for NLP models and we will leave it for future work. In order to launch antonym attack automatically, we are also limited to sentence pair tasks that are more prone to changing the ground truth label when replacing a word with its antonym. In addition, BAT sacrifices the performance on the synonym-based attack success rate for robustness to antonym-based attack when comparing to fickle adversarial training methods. We show that there is a tradeoff between robustness against synonym and antonym based attacks and our goal is to achieve a better tradeoff between them.

## Availability

Code for reproducing our experiments is available at: <https://github.com/hannahxchen/balanced-adversarial-training>.

## Acknowledgements

This work was partially supported by a grant from the National Science Foundation (#1804603).

## References

- Moustafa Alzantot, Yash Sharma, Ahmed Elgohary, Bo-Jhang Ho, Mani Srivastava, and Kai-Wei Chang. 2018. [Generating natural language adversarial examples](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.
- Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. 2020. [A simple framework for contrastive learning of visual representations](#). *CoRR*, abs/2002.05709.
- 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)*.
- Shi Feng, Eric Wallace, Alvin Grissom II, Mohit Iyyer, Pedro Rodriguez, and Jordan Boyd-Graber. 2018. [Pathologies of neural models make interpretations difficult](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.
- 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)*.
- Ian J. Goodfellow, Nicolas Papernot, and Patrick D. McDaniel. 2016. [cleverhans v0.1: an adversarial machine learning library](#). *CoRR*, abs/1610.00768.
- Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. [Explaining and harnessing adversarial examples](#). *CoRR*, abs/1412.6572.
- R. Hadsell, S. Chopra, and Y. LeCun. 2006. [Dimensionality reduction by learning an invariant mapping](#). In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2.
- Chih-Hui Ho and Nuno Vasconcelos. 2020. [Contrastive learning with adversarial examples](#). In *Advances in Neural Information Processing Systems*, volume 33.
- Po-Sen Huang, Robert Stanforth, Johannes Welbl, Chris Dyer, Dani Yogatama, Sven Gowal, Krishnamurthy Dvijotham, and Pushmeet Kohli. 2019. [Achieving verified robustness to symbol substitutions via interval bound propagation](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*.
- Shankar Iyer, Nikhil Dandekar, and Kornél Csernai. 2017. First quora dataset release: Question pairs. <https://www.quora.com/q/quoradata/First-Quora-Dataset-Release-Question-Pairs>.
- Joern-Henrik Jacobsen, Jens Behrmann, Nicholas Carlini, Florian Tramèr, and Nicolas Papernot. 2019a. Exploiting excessive invariance caused by norm-bounded adversarial robustness. *Safe Machine Learning workshop at ICLR*.
- Joern-Henrik Jacobsen, Jens Behrmann, Richard Zemel, and Matthias Bethge. 2019b. [Excessive invariance causes adversarial vulnerability](#). In *International Conference on Learning Representations*.
- Robin Jia, Aditi Raghunathan, Kerem Göksel, and Percy Liang. 2019. [Certified robustness to adversarial word substitutions](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*.
- 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](#). *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05).

- Minseon Kim, Jihoon Tack, and Sung Ju Hwang. 2020. [Adversarial self-supervised contrastive learning](#). In *Advances in Neural Information Processing Systems*, volume 33.
- Vivekanandan S. Kumar and David Boulanger. 2020. Explainable automated essay scoring: Deep learning really has pedagogical value. In *Frontiers in Education*.
- Dianqi Li, Yizhe Zhang, Hao Peng, Liqun Chen, Chris Brockett, Ming-Ting Sun, and Bill Dolan. 2021. [Contextualized perturbation for textual adversarial attack](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.
- 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)*.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. [RoBERTa: A robustly optimized BERT pretraining approach](#). *CoRR*, abs/1907.11692.
- 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*.
- George A. Miller. 1995. [WordNet: A lexical database for English](#). *Communications of the ACM*, 38(11).
- John Morris, Eli Lifland, Jack Lanchantin, Yangfeng Ji, and Yanjun Qi. 2020. [Reevaluating adversarial examples in natural language](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020*.
- Tong Niu and Mohit Bansal. 2018. [Adversarial over-sensitivity and over-stability strategies for dialogue models](#). In *Proceedings of the 22nd Conference on Computational Natural Language Learning*.
- Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. [GloVe: Global vectors for word representation](#). In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.
- 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 Annual Meeting of the Association for Computational Linguistics*.
- Florian Schroff, Dmitry Kalenichenko, and James Philbin. 2015. [FaceNet: A unified embedding for face recognition and clustering](#). In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Florian Tramer, Jens Behrmann, Nicholas Carlini, Nicolas Papernot, and Joern-Henrik Jacobsen. 2020. [Fundamental tradeoffs between invariance and sensitivity to adversarial perturbations](#). In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*.
- Laurens van der Maaten and Geoffrey Hinton. 2008. [Visualizing data using t-SNE](#). *Journal of Machine Learning Research*, 9(86).
- 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*.
- Johannes Welbl, Pasquale Minervini, Max Bartolo, Pontus Stenetorp, and Sebastian Riedel. 2020. [Under-sensitivity in neural reading comprehension](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020*.
- Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. [A broad-coverage challenge corpus for sentence understanding through inference](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*.
- Ellery Wulczyn, Nithum Thain, and Lucas Dixon. 2017. [Ex Machina: Personal attacks seen at scale](#). In *Proceedings of the 26th International Conference on World Wide Web*.
- Mao Ye, Chengyue Gong, and Qiang Liu. 2020. [SAFER: A structure-free approach for certified robustness to adversarial word substitutions](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.
- Jin Yong Yoo and Yanjun Qi. 2021. [Towards improving adversarial training of NLP models](#). In *Findings of the Association for Computational Linguistics: EMNLP 2021*.

## A Fickleness and Obstinacy Robustness Tradeoffs

### A.1 Synonym and Antonym Attack Robustness Tradeoffs

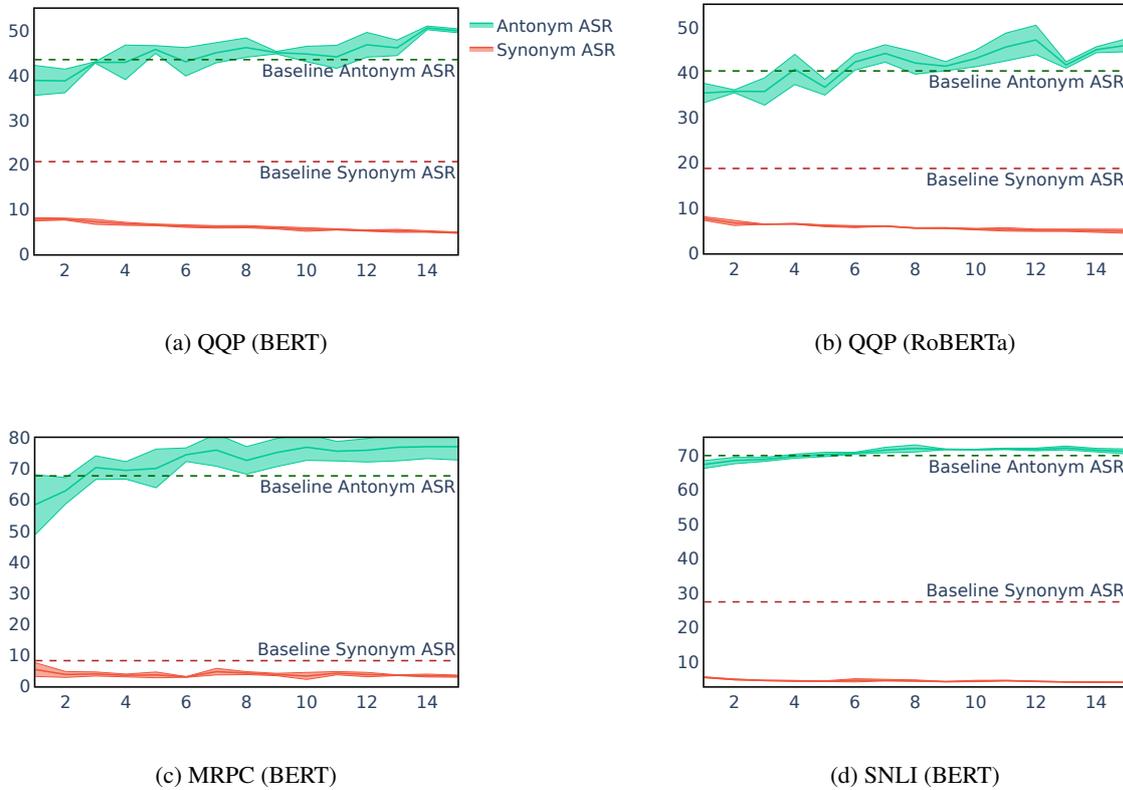


Figure 6: Robustness tradeoffs between synonym and antonym based attacks on QQP and MRPC dataset. The figure shows the average and standard deviation across 3 different runs.

## A.2 Synonym and Negation Attack Robustness Tradeoffs

We test how negation attack success rate would change as the model robustness against synonym attack increases. For negation attack, we add negation to a verb in the sentence, i.e., “*I can do it*” to “*I can’t do it*”, or remove negation from a sentence, i.e., “*I am not going*” to “*I am going*”. We follow similar setup as in Section 3. We found that there exists a tradeoff between synonym-based adversarial examples and negation-based adversarial examples on QQP task, but found no significant tradeoff on MNLI task, as shown in Figure 7.

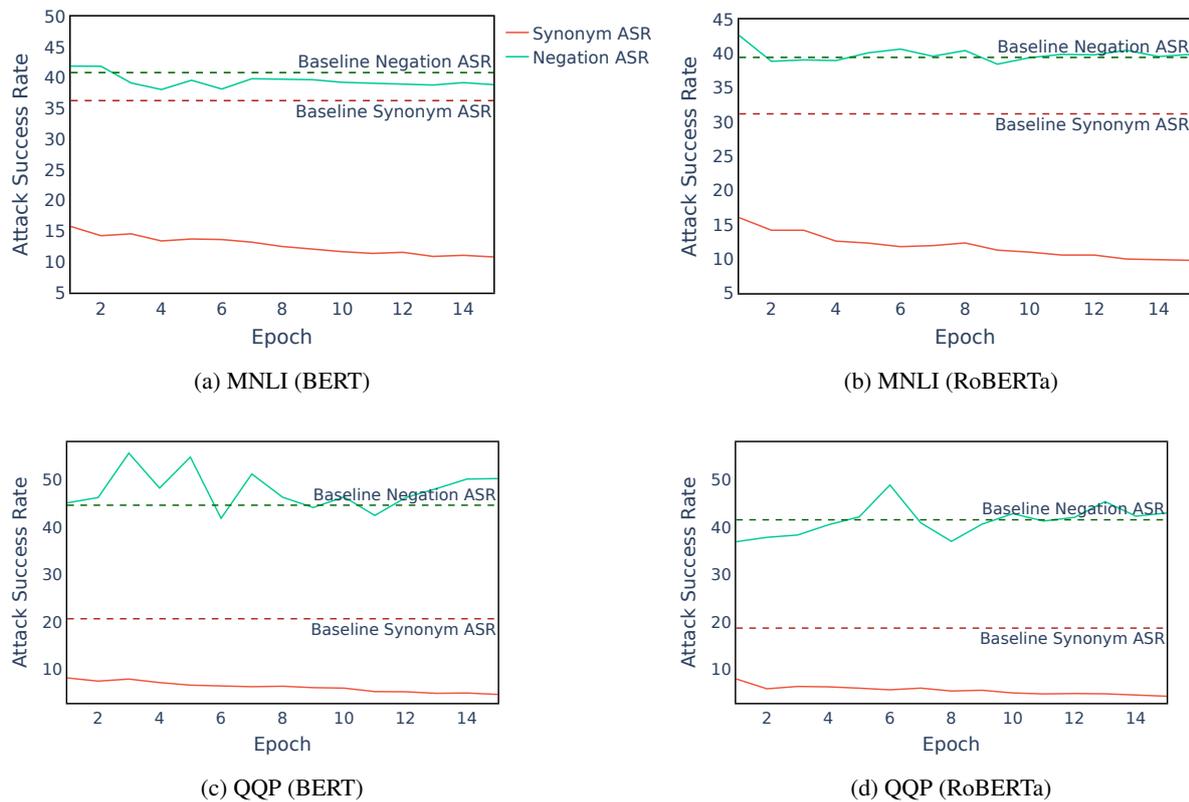


Figure 7: Negation attack success rate on models at each epoch when training with SAFER.

## B Fickleness Robust Training with Varying Batch Size

### B.1 Synonym and Antonym Robustness Tradeoffs on QQP Task

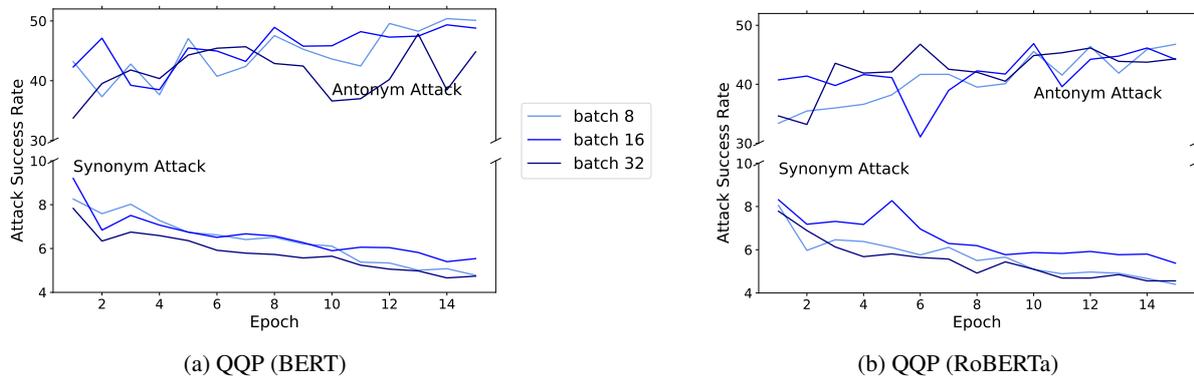


Figure 8: The synonym and antonym attack success rate at each SAFER training epoch with varying batch size.

### B.2 Evaluation Accuracy at Each SAFER Training Epoch

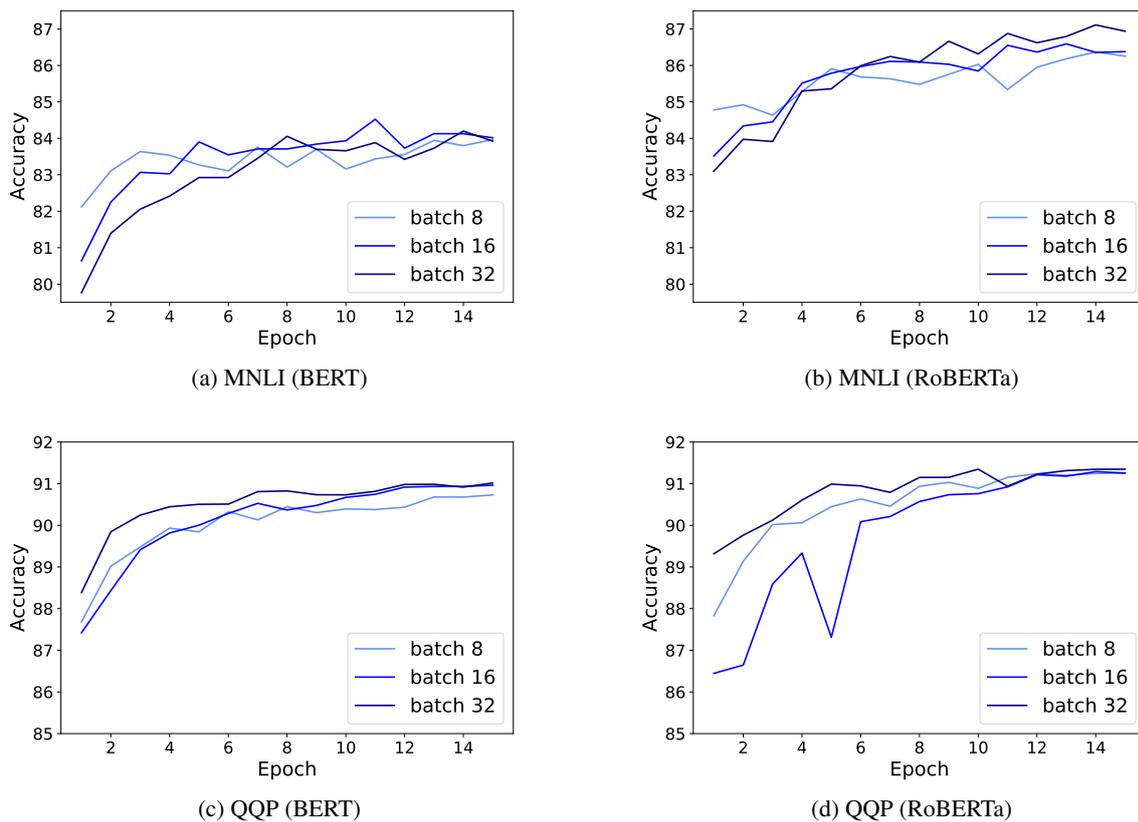


Figure 9: The evaluation accuracy on original validation set at each SAFER training epoch with varying batch size.

## C Balanced Adversarial Training

### C.1 Balanced Adversarial Training Details

We implement BAT similarly to the SAFER training method as described in Section 3.1 where we randomly perturb the inputs with words from the synonym/antonym substitution sets.

We train the BERT and RoBERTa models for 2 or 3 epochs with a learning rate of  $2 \times 10^{-5}$  or  $3 \times 10^{-5}$  and batch size of 32. For contrastive loss weights and margin in BAT-Pairwise and BAT-Triplet, we perform hyperparameter search and choose the ones with best performance (see Appendix C.2).

### C.2 BAT Hyperparameter Search

#### C.2.1 Contrastive Loss Weights

In Figure 10, we show varying fickle loss weights ( $\alpha$ ) with obstinate loss weight fixed ( $\beta = 1.0$ ) and vice versa when training with BAT-Pairwise method. As the value of  $\alpha$  increases, antonym attack success rate increases. On the other hand, as the value of  $\beta$  increases, synonym attack success rate has a small increase as well. We found  $\alpha = 1.0$  and  $\beta = 1.2$  gives the best performance for BERT MNLI model. We test different contrastive loss weights ( $\lambda$ ) when training with BAT-Triplet and show the results in Figure 11. We found that as we increase  $\lambda$ , model accuracy on the validation set decreases.

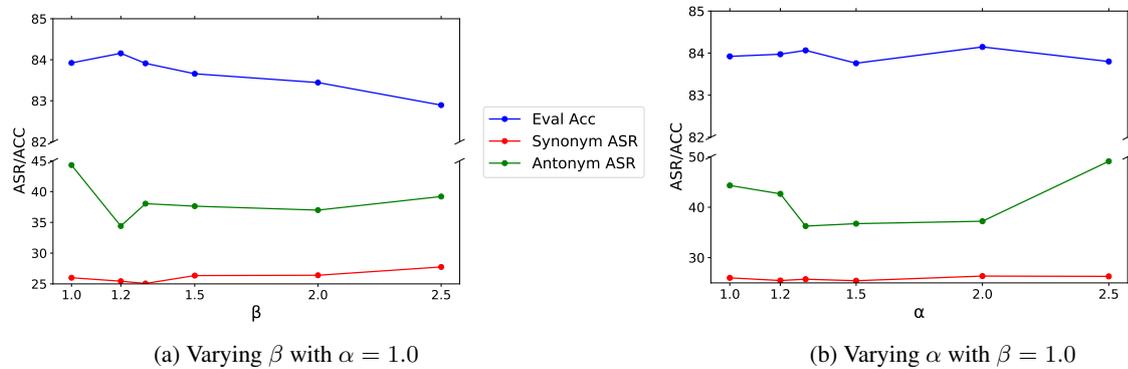


Figure 10: Performance of BERT models trained on MNLI tasks with different fickle ( $\alpha$ ) and obstinate ( $\beta$ ) loss weights in BAT-Pairwise.

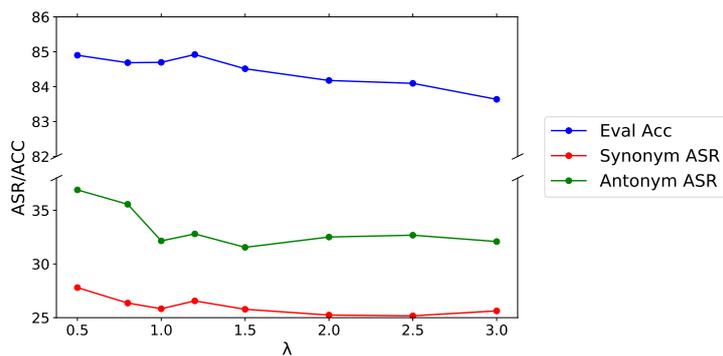


Figure 11: Performance of BERT models trained on MNLI tasks with different contrastive loss weights ( $\lambda$ ) in BAT-Triplet.

### C.2.2 Margin

We test varying margin  $m$  from 0.3 to 1.0 as cosine similarity ranges between 0 to 1. In Figure 12, we can see that as margin approaches 1, both synonym and antonym attack success rates decrease. Model achieves best performance when margin is closer to 1. We also found that margin has larger effect on BAT-Triplet.

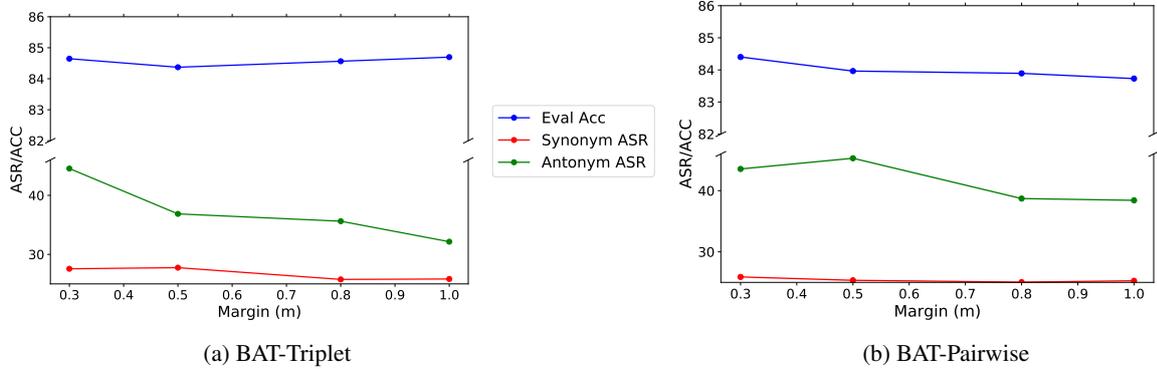


Figure 12: Performance of BERT models trained on MNLI tasks with different margin ( $m$ ).

### C.3 BAT Evaluation on MNLI Mismatched Validation Set

Model	Method	Eval Acc (%)	Antonym ASR (%)	Synonym ASR (%)
BERT	Normal Training	84.28 $\pm$ 0.51	58.59 $\pm$ 0.28	41.15 $\pm$ 0.45
	A2T	85.00	57.86	24.84
	SAFER	83.73 $\pm$ 0.12	65.13 $\pm$ 1.64	12.59 $\pm$ 0.78
	BAT-Pairwise	84.52 $\pm$ 0.06	36.71 $\pm$ 1.31	29.63 $\pm$ 0.95
	BAT-Triplet	84.99 $\pm$ 0.06	34.55 $\pm$ 2.52	29.02 $\pm$ 0.17
RoBERTa	Normal Training	87.49 $\pm$ 0.05	57.66 $\pm$ 0.95	34.56 $\pm$ 0.16
	A2T	86.52	58.19	21.07
	SAFER	86.26 $\pm$ 0.23	69.52 $\pm$ 1.76	11.12 $\pm$ 0.22
	BAT-Pairwise	87.16 $\pm$ 0.41	39.34 $\pm$ 2.09	30.58 $\pm$ 0.28
	BAT-Triplet	87.19 $\pm$ 0.18	33.60 $\pm$ 0.78	30.19 $\pm$ 0.91

Table 3: Balanced Adversarial Training evaluation results on MNLI mismatched validation set.

### C.4 BAT-Triplet with Varying Batch Size

In Section 3.3, we observed that certified robust training with smaller batch sizes results in larger gaps in robustness tradeoff. We test BAT-Triplet with varying batch size when training BERT on MNLI task and we find that it gives consistent improvement on robustness regardless of the batch size, as shown in Table 4.

Batch Size	Accuracy (%)	Antonym ASR (%)	Synonym ASR (%)
8	84.45	34.59	25.66
16	84.08	31.05	25.89
32	84.70	32.15	25.83

Table 4: BAT-Triplet with BERT training, varying batch size, evaluated on MNLI matched validation set.

## C.5 Additional BAT Results

Model	Method	Eval Acc (%)	F1	Antonym ASR (%)	Synonym ASR (%)
BERT	Normal Training	85.37 $\pm$ 0.42	89.66 $\pm$ 0.30	67.79 $\pm$ 1.71	8.23 $\pm$ 0.26
	SAFER	85.62 $\pm$ 0.46	90.01 $\pm$ 0.38	77.19 $\pm$ 4.28	3.24 $\pm$ 0.34
	BAT-Pairwise	85.54 $\pm$ 0.40	89.68 $\pm$ 0.46	44.34 $\pm$ 3.96	6.02 $\pm$ 0.43
	BAT-Triplet	85.13 $\pm$ 0.31	89.50 $\pm$ 0.20	45.22 $\pm$ 0.63	6.62 $\pm$ 0.25

Table 5: Balanced Adversarial Training evaluation results on MRPC validation set.

Model	Method	Eval Acc (%)	Antonym ASR (%)	Synonym ASR (%)
BERT	Normal Training	90.82 $\pm$ 0.33	69.90 $\pm$ 0.23	27.46 $\pm$ 0.18
	SAFER	90.00 $\pm$ 0.24	72.10 $\pm$ 0.56	4.21 $\pm$ 0.07
	BAT-Pairwise	90.58 $\pm$ 0.31	54.01 $\pm$ 1.15	15.10 $\pm$ 0.04
	BAT-Triplet	90.67 $\pm$ 0.39	44.19 $\pm$ 1.31	15.49 $\pm$ 0.37

Table 6: Balanced Adversarial Training evaluation results on SNLI validation set.