Adversarial Training with Fast Gradient Projection Method against Synonym Substitution based Text Attacks

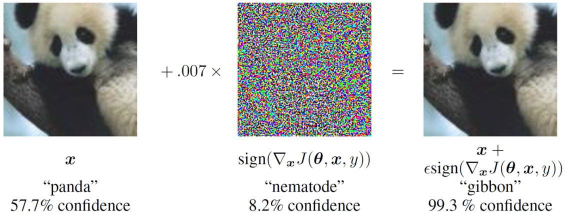
Xiaosen Wang[1], Yichen Yang[1], Yihe Deng[2] and Kun He[1]

[1] School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, China
[2] Computer Science Department, University of California, Los Angeles
Contact: xiaosen@hust.edu.cn
Homepage: `https://xiaosen-wang.github.io/`

Outline

1. Background
2. Fast Gradient Projection Method (FGPM)
3. Adversarial Training with FGPM enhanced by Logit pairing (ATFL)
4. Experiment
5. Conclusion

Adversarial Example



Adversarial Example for Image Classification [4].

Prediction	Confidence	Texts
Positive	99.7%	This is a **unique** masterpiece made by the best director ever lived in the ussr. He knows the art of film making and can use it very well. If you find this movie, buy or copy it!
Negative	86.2%	This is a **sole** masterpiece made by the best director ever lived in the ussr. He knows the art of film making and can use it very well. If you find this movie, buy or copy it!

Adversarial Example for Text Classification [14].

Definition of Adversarial Examples

Image Adversarial Examples

Given an image classifier f and a constant ϵ, the image adversarial example for input x can be defined as finding an example x_{adv} which satisfies $\|x - x_{adv}\|_p < \epsilon$ and $f(x_{adv}) \neq f(x) = y$, where $\|\cdot\|_p$ denotes ℓ_p norm and y is the ground true label.

Textual Adversarial Examples

Given a text classifier ϕ and a constant ϵ, the textual adversarial example for input x can be defined as finding an example x_{adv} which satisfies $R(x, x_{adv}) < \epsilon$ and $\phi(x_{adv}) \neq \phi(x) = y$, where $R(a, b)$ evaluates the dissimilarity between a and b.

Definition of Adversarial Examples

Image Adversarial Examples

Given an image classifier f and a constant ϵ, the image adversarial example for input x can be defined as finding an example x_{adv} which satisfies $\|x - x_{adv}\|_p < \epsilon$ and $f(x_{adv}) \neq f(x) = y$, where $\|\cdot\|_p$ denotes ℓ_p norm and y is the ground true label.

Textual Adversarial Examples

Given a text classifier ϕ and a constant ϵ, the textual adversarial example for input x can be defined as finding an example x_{adv} which satisfies $R(x, x_{adv}) < \epsilon$ and $\phi(x_{adv}) \neq \phi(x) = y$, where $R(a, b)$ evaluates the dissimilarity between a and b.

It is hard for textual adversarial attack and defense due to the lexical, grammatical and semantic constraints.

Various type of Textual Adversarial Attacks

Based on the metrics to evaluate the dissimilarity of two texts, current adversarial attacks can be split into three categories.

- Character Level Attack [10, 3, 9]
    - Flipping/deleting/inserting characters: A spell checker can fix the perturbations.
- Sentence Level Attack [5, 13]
    - Paraphrasing: Very time consuming.
- Word Level Attack
    - Embedding perturbation or adding/removing words [11]: Hurting semantic consistency and grammatical correctness.
    - **Synonym substitution** [1, 12, 14]: A good and popular way for generating textural adversarial examples.

Existing Synonym Substitution Based Adversarial Attack Methods

- **Greedy Search Algorithm (GSA)** [8] greedily substitutes the word in the input with the word in the synonym set which minimizes the confidence.
- **Genetic Algorithm (GA)** [1] and **Improved Genetic Algorithm (IGA)** [14] adopt a population for replacing word with their synonym which minimizes the confidence.
- **Probability Weighted Word Saliency (PWWS)** [12] considers the word saliency as well as the classification confidence for substituting the word.
- **Particle Swarm Optimization (PSO)** [16] treats the text as a particle and substitutes the word with sememe word.

Existing Synonym Substitution Based Adversarial Attack Methods

- **Greedy Search Algorithm (GSA)** [8] greedily substitutes the word in the input with the word in the synonym set which minimizes the confidence.
- **Genetic Algorithm (GA)** [1] and **Improved Genetic Algorithm (IGA)** [14] adopt a population for replacing word with their synonym which minimizes the confidence.
- **Probability Weighted Word Saliency (PWWS)** [12] considers the word saliency as well as the classification confidence for substituting the word.
- **Particle Swarm Optimization (PSO)** [16] treats the text as a particle and substitutes the word with sememe word.

All the above attacks are black-box attack and time-consuming!

Revisting Adversarial Attack in Image Domain

Fast Gradient Sign Method (FGSM) [4] crafts adversarial example by adding perturbation in the gradient direction of the loss function $J(x, y; \theta)$ as follows:

$$x^{adv} = x + \epsilon \cdot \mathrm{sign}(\nabla_x J(x, y; \theta)),$$

where $\mathrm{sign}(\cdot)$ denotes the sign function and $\nabla_x J(x, y; \theta)$ is the gradient of the loss function w.r.t.x.

FGSM is very **fast** because it only needs one forward propagation and backpropagation to craft adversarial example.

Revisting Adversarial Attack in Image Domain

Fast Gradient Sign Method (FGSM) [4] crafts adversarial example by adding perturbation in the gradient direction of the loss function $J(x, y; \theta)$ as follows:

$$x^{adv} = x + \epsilon \cdot \text{sign}(\nabla_x J(x, y; \theta)),$$

where $\text{sign}(\cdot)$ denotes the sign function and $\nabla_x J(x, y; \theta)$ is the gradient of the loss function w.r.t.x.

FGSM is very **fast** because it only needs one forward propagation and backpropagation to craft adversarial example.

Could we generate textual adversary by FGSM?

Why FGSM cannot be Applied in Text Domain?

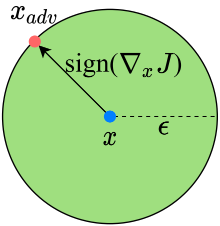
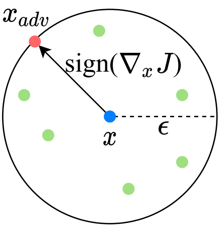


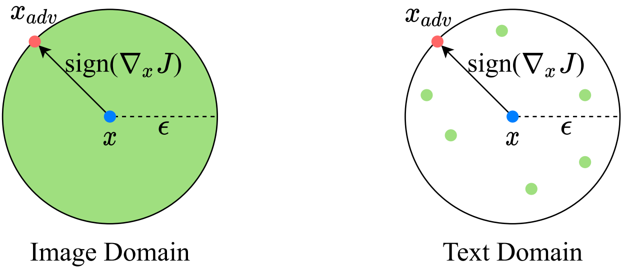
Image Domain



Text Domain

● denotes the possible inputs for the deep neural model.

Why FGSM cannot be Applied in Text Domain?



● denotes the possible inputs for the deep neural model.

Even we fortunately find a possible input by FGSM, it might also violate the lexical, grammatical and semantic constrains.

Fast Gradient Projection Method (FGPM)

Revisiting Synonym Substitution based Text Attacks

Given a target classifier ϕ and input text $x = \langle w_1, \cdots, w_i, \cdots, w_n \rangle$, there are generally three procedures for crafting the adversarial example x_{adv}:

- Constructing the Synonym set for each word w_i
- Finding the optimal synonym for each word w_i
- Determining the substitution order

Fast Gradient Projection Method (FGPM)

Constructing the Synonym Set

To align with previous works, we construct the synonym set based on GloVe vector space.

- Measuring semantic similarity: Euclidean distance in GloVe vector space after counter-fitting which removes antonyms.
- Defining a synonym set for each word $w_i \in x$ in the embedding space as follows:

$$S(w_i, \delta) = \{\hat{w}_i \in \mathcal{D} \mid \|\hat{w}_i - w_i\|_2 \leq \delta\}, \tag{1}$$

where δ is a hyper-parameter that constrains the maximum Euclidean distance for synonyms in the embedding space and we set $\delta = 0.5$.

Fast Gradient Projection Method (FGPM)

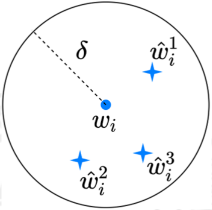
Finding the Optimal Synonym for Each Word

For each word w_i, we expect to pick a word $\hat{w}_i^* \in S(w_i, \delta)$ that earns the most benefit to the overall substitution process of adversary generation.

Previous works greedily pick a synonym $\hat{w}_i^* \in S(w_i, \delta)$ that minimizes the classification confidence:

$$\hat{w}_i^* = \underset{\hat{w}_i^j \in S(w_i,\delta)}{\arg\max} \left(F(x, y) - F(\hat{x}_i^j, y)\right),$$

where $\hat{x}_i^j = \langle w_1, \cdots, w_{i-1}, \hat{w}_i^j, w_{i+1}, \cdots, w_n \rangle$. The selection process is time consuming as picking such a $\hat{w}_i^*$ needs $|S(w_i, \delta)|$ queries on the model.

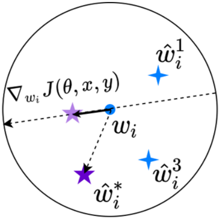


Fast Gradient Projection Method (FGPM)

Finding the Optimal Synonym for Each Word

Based on the local linearity of deep models, we first calculate the gradient $\nabla_{w_i} J(\theta, x, y)$ for each word w_i where $J(\theta, x, y)$ is the loss function used for training. Then, we estimate the change by calculating $(\hat{w}_i^j - w_i) \cdot \nabla_{w_i} J(\theta, x, y)$ and choose a synonym with the maximum product value:

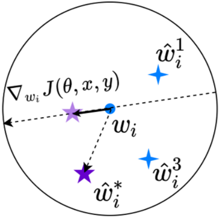
$$\hat{w}_i^* = \underset{\hat{w}_i^j \in S(w_i, \delta)}{\arg\max} \, (\hat{w}_i^j - w_i) \cdot \nabla_{w_i} J(\theta, x, y). \qquad (2)$$



Fast Gradient Projection Method (FGPM)

Finding the Optimal Synonym for Each Word

Based on the local linearity of deep models, we first calculate the gradient $\nabla_{w_i} J(\theta, x, y)$ for each word w_i where $J(\theta, x, y)$ is the loss function used for training. Then, we estimate the change by calculating $(\hat{w}_i^j - w_i) \cdot \nabla_{w_i} J(\theta, x, y)$ and choose a synonym with the maximum product value:



$$\hat{w}_i^* = \mathop{\arg\max}_{\hat{w}_i^j \in S(w_i, \delta)} (\hat{w}_i^j - w_i) \cdot \nabla_{w_i} J(\theta, x, y). \qquad (2)$$

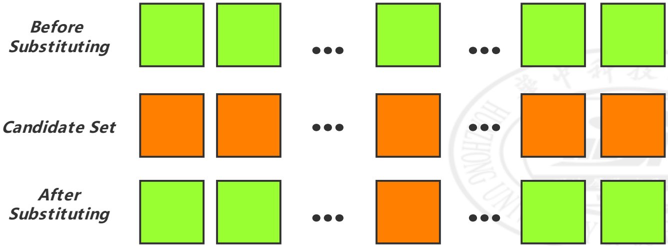
Only one query needed for choosing $\hat{w}_i^*$.

Fast Gradient Projection Method (FGPM)

Determining the Substitution Order

For each word w_i in text $x = \langle w_1, \cdots, w_i, \cdots, w_n \rangle$, we use the above word substitution strategy to choose its optimal substitution synonym and obtain a candidate set $C_s = \{\hat{w}_1^*, \cdots, \hat{w}_i^*, \cdots, \hat{w}_n^*\}$. Then we pick a word $\hat{w}_i^* \in C_s$ that leads to the biggest value:

$$\hat{w}_* = \underset{\hat{w}_i^* \in C_s}{\arg\max} (\hat{w}_i^* - w_i) \cdot \nabla_{w_i} J(\theta, x, y). \tag{3}$$



Fast Gradient Projection Method (FGPM)
Algorithm

Algorithm 1 The FGPM Algorithm

Input: Benign sample $x = \langle w_1, \cdots, w_i, \cdots, w_n \rangle$; True label y for x; Target classifier ϕ; Upper bound distance for synonyms δ; Maximum number of iterations N; Upper bound for word substitution ratio ϵ

Output: Adversarial example x_{adv}

1: Initialize $x^0_{adv} = x$
2: Calculate $S(w_i, \delta)$ by Eq. (1) for $w_i \in x^0_{adv}$
3: **for** $k = 1 \rightarrow N$ **do**
4: Construct candidate set $C_s = \{\hat{w}^*_1, \cdots, \hat{w}^*_i, \cdots, \hat{w}^*_n\}$ by Eq. (2)
5: Calculate optimal word $\hat{w}_*$ by Eq. (3)
6: Substitute $w_* \in x^{k-1}_{adv}$ with $\hat{w}_*$ to obtain x^k_{adv}
7: **if** $\phi(x^k_{adv}) \neq y$ and $R(x^k_{adv}, x) < \epsilon$ **then**
8: **return** x^k_{adv} ▷ Succeed
9: **end if**
10: **end for**
11: **return** None ▷ Failed

$$R(x, x_{adv}) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}_{w_i \neq w'_i}(w_i, w'_i)$$

Adversarial Training with FGPM enhanced by Logit pairing (ATFL)

Variants of Adversarial Training in Image Domain

Adversarial training (AT), which injects adversarial examples into training data, is one of the most efficacious defense methods in image domain and has been widely investigated.

Defense Method	Loss Function
Standard [4]	$\alpha CE(F(x,\cdot),y) + (1-\alpha)CE(F(x_{adv},\cdot),y)$
TRADES [17]	$CE(F(x,\cdot),y) + \lambda \cdot \|F(x,\cdot) - F(x_{adv},\cdot)\|$
MMA [2]	$CE(F(x,\cdot),y) \cdot \mathbb{1}(\phi(x) \neq y) + CE(F(x_{adv},\cdot),y) \cdot \mathbb{1}(\phi(x) = y)$
MART [15]	$BCE(F(x_{adv},\cdot),y) + \lambda \cdot KL(F(x,\cdot)\|F(x_{adv},\cdot)) \cdot (1 - F(x,y))$
CLP [7]	$CE(F(x,\cdot),y) + \lambda \cdot \|F(x,\cdot) - F(x',\cdot)\|$
ALP [7]	$\alpha CE(F(x,\cdot),y) + (1-\alpha)CE(F(x_{adv},\cdot),y) + \lambda \cdot \|F(x,\cdot) - F(x_{adv},\cdot)\|$

Table: The loss functions for different variations of adversarial training.

Adversarial Training with FGPM enhanced by Logit pairing (ATFL)

Why AT has not been implemented as an effective defense method against synonym substitution based attacks?

- AT needs a large number of adversaries for training.
- Due to the discrete input space, existing attacks do not adopt gradient and are very slow.

Such inefficiency of existing adversary generation methods holds back adversarial training in text domain.

Adversarial Training with FGPM enhanced by Logit pairing (ATFL)

Why AT has not been implemented as an effective defense method against synonym substitution based attacks?

- AT needs a large number of adversaries for training.
- Due to the discrete input space, existing attacks do not adopt gradient and are very slow.

Such inefficiency of existing adversary generation methods holds back adversarial training in text domain.

The high efficiency of FGPM makes it possible for AT against synonym substitution based attacks. We further propose Adversarial Training with FGPM enhanced by Logit pairing (ATFL):

$$\tilde{J}(\theta, x, y) = \alpha J(\theta, x, y) + (1-\alpha) J(\theta, x_{adv}, y) + \lambda \| F(x, \cdot) - F(x_{adv}, \cdot) \|.$$

Experiments
Experimental Setup

- **Baselines**
  - Attacks: *Papernot'* [11], GSA [8], PWWS [12] and IGA [14]
  - Defenses: IBP [6], SEM [14]
- **Datasets**: *AG's News*, *DBPedia* and *Yahoo! Answers*
- **Models**: CNN, LSTM and Bi-LSTM
- **Hyper-parameters**: $\epsilon = 0.25$, $\alpha = 0.5$, $\lambda = 0.5$

Due to the low efficiency of attack baselines, we craft adversarial examples on 200 randomly sampled examples on each dataset.

Experiments

Evaluation on FGPM — Classification Accuracy under Attacks

	AG's News			*DBPedia*			*Yahoo! Answers*		
	CNN	LSTM	Bi-LSTM	CNN	LSTM	Bi-LSTM	CNN	LSTM	Bi-LSTM
No Attack†	92.3	92.6	92.5	98.7	98.8	99.0	72.3	75.1	74.9
No Attack	87.5	90.5	88.5	99.5	99.0	99.0	71.5	72.5	73.5
Papernot'	72.0	61.5	65.0	80.5	77.0	83.5	38.0	43.0	36.5
GSA	45.5	35.0	40.0	52.0	49.0	53.5	21.5	19.5	19.0
PWWS	<u>37.5</u>	<u>30.0</u>	<u>29.0</u>	55.5	52.5	50.0	<u>5.5</u>	<u>12.5</u>	11.0
IGA	**30.0**	**26.5**	**25.5**	**36.5**	**38.5**	**37.0**	**3.5**	**5.5**	**7.0**
FGPM	<u>37.5</u>	31.0	32.0	<u>40.0</u>	<u>45.5</u>	<u>47.5</u>	6.0	17.0	<u>10.5</u>

Table: The classification accuracy (%) of different models under various competitive adversarial attacks.

Experiments

Evaluation on FGPM — Classification Accuracy under Attacks

	AG's News			*DBPedia*			*Yahoo! Answers*		
	CNN	LSTM	Bi-LSTM	CNN	LSTM	Bi-LSTM	CNN	LSTM	Bi-LSTM
No Attack†	92.3	92.6	92.5	98.7	98.8	99.0	72.3	75.1	74.9
No Attack	87.5	90.5	88.5	99.5	99.0	99.0	71.5	72.5	73.5
Papernot'	72.0	61.5	65.0	80.5	77.0	83.5	38.0	43.0	36.5
GSA	45.5	35.0	40.0	52.0	49.0	53.5	21.5	19.5	19.0
PWWS	<u>37.5</u>	<u>30.0</u>	<u>29.0</u>	55.5	52.5	50.0	<u>5.5</u>	<u>12.5</u>	11.0
IGA	**30.0**	**26.5**	**25.5**	**36.5**	**38.5**	**37.0**	**3.5**	**5.5**	**7.0**
FGPM	<u>37.5</u>	31.0	32.0	<u>40.0</u>	<u>45.5</u>	<u>47.5</u>	6.0	17.0	<u>10.5</u>

Table: The classification accuracy (%) of different models under various competitive adversarial attacks.

Compared with other attacks, FGPM achieves the attack performance on par with other attacks.

Experiments

Evaluation on FGPM — Transferability

	CNN	LSTM	Bi-LSTM	CNN	LSTM	Bi-LSTM	CNN	LSTM	Bi-LSTM
Papernot'	72.0*	80.5	82.5	83.5	61.5*	78.5	79.5	74.5	65.0*
GSA	45.5*	80.0	80.0	84.5	35.0*	73.0	81.5	72.5	40.0*
PWWS	37.5*	**70.5**	**70.0**	<u>83.0</u>	30.0*	**67.5**	80.0	**67.5**	29.0*
IGA	30.0*	74.5	<u>74.5</u>	84.0	26.5*	<u>71.5</u>	<u>79.0</u>	<u>71.0</u>	25.5*
FGPM	37.5*	<u>72.5</u>	<u>74.5</u>	**81.0**	31.0*	73.5	**77.5**	**67.5**	32.0*

Table: The classification accuracy (%) of different models for adversaries generated on other models on *AG's News* for transferability evaluation. * indicates that the adversaries are generated based on this model.

Experiments

Evaluation on FGPM — Transferability

	CNN	LSTM	Bi-LSTM	CNN	LSTM	Bi-LSTM	CNN	LSTM	Bi-LSTM
Papernot'	72.0*	80.5	82.5	83.5	61.5*	78.5	79.5	74.5	65.0*
GSA	45.5*	80.0	80.0	84.5	35.0*	73.0	81.5	72.5	40.0*
PWWS	37.5*	**70.5**	**70.0**	83.0	30.0*	**67.5**	80.0	**67.5**	29.0*
IGA	30.0*	74.5	74.5	84.0	26.5*	71.5	79.0	71.0	25.5*
FGPM	37.5*	72.5	74.5	**81.0**	31.0*	73.5	**77.5**	**67.5**	32.0*

Table: The classification accuracy (%) of different models for adversaries generated on other models on *AG's News* for transferability evaluation. * indicates that the adversaries are generated based on this model.

The adversarial examples crafted by FGPM is on par with the best transferability performance among the baselines.

Experiments
Evaluation on FGPM — Attack Efficiency

	AG's News			*DBPedia*			*Yahoo! Answers*		
	CNN	LSTM	Bi-LSTM	CNN	LSTM	Bi-LSTM	CNN	LSTM	Bi-LSTM
Papernot'	74	1,676	4,401	145	2,119	6,011	120	9,719	19,211
GSA	276	643	713	616	1,006	1,173	1,257	2,234	2,440
PWWS	122	28,203	28,298	204	34,753	35,388	643	98,141	100,314
IGA	965	47,142	91,331	1,369	69,770	74,376	893	132,044	123,976
FGPM	**8**	**29**	**29**	**8**	**34**	**33**	**26**	**193**	**199**

Table: Comparison on the total running time (in seconds) for generating 200 adversarial instances.

Experiments
Evaluation on FGPM — Attack Efficiency

	AG's News			*DBPedia*			*Yahoo! Answers*		
	CNN	LSTM	Bi-LSTM	CNN	LSTM	Bi-LSTM	CNN	LSTM	Bi-LSTM
Papernot'	74	1,676	4,401	145	2,119	6,011	120	9,719	19,211
GSA	276	643	713	616	1,006	1,173	1,257	2,234	2,440
PWWS	122	28,203	28,298	204	34,753	35,388	643	98,141	100,314
IGA	965	47,142	91,331	1,369	69,770	74,376	893	132,044	123,976
FGPM	**8**	**29**	**29**	**8**	**34**	**33**	**26**	**193**	**199**

Table: Comparison on the total running time (in seconds) for generating 200 adversarial instances.

FGPM is at least 20 times faster than the fastest baseline method GSA, while maintaining a high attack success rate.

Experiments

Evaluation on ATFL — Defense against Adversarial Attacks

Dataset	Attack	CNN				LSTM				Bi-LSTM			
		NT	SEM	IBP	ATFL	NT	SEM	IBP	ATFL	NT	SEM	IBP	ATFL
AG's News	No Attack†	**92.3**	89.7	89.4	91.8	**92.6**	90.9	86.3	92.0	**92.5**	91.4	89.1	92.1
	No Attack	87.5	87.5	87.5	**89.0**	90.5	90.5	84.5	**91.5**	88.5	**91.0**	87.0	89.5
	Papernot'	72.0	84.5	87.5	**88.0**	61.5	89.5	81.5	**90.0**	65.0	**90.0**	86.0	89.0
	GSA	45.5	80.0	86.0	**88.0**	35.0	85.5	79.5	**88.0**	40.0	**87.5**	79.0	**87.5**
	PWWS	37.5	80.5	86.0	**88.0**	30.0	86.5	79.5	**88.0**	29.0	**87.5**	75.5	**87.5**
	IGA	30.0	80.0	86.0	**88.0**	26.5	85.5	79.5	**88.0**	25.5	**87.5**	79.0	**87.5**
	FGPM	37.5	78.5	86.5	**88.0**	31.0	85.5	80.0	**88.0**	32.0	84.5	80.0	**87.5**
DBPedia	No Attack†	**98.7**	98.1	97.4	98.4	**98.8**	98.5	93.1	98.7	**99.0**	98.7	94.7	98.6
	No Attack	**99.5**	97.5	97.0	98.0	99.0	**99.5**	95.0	**99.5**	**99.0**	98.0	94.5	**99.0**
	Papernot'	80.5	97.0	97.0	**98.0**	77.0	**99.5**	91.0	**99.5**	83.5	98.0	92.5	**99.0**
	GSA	52.0	96.0	97.0	**98.0**	49.0	**99.0**	84.5	98.5	53.5	98.0	89.5	**99.0**
	PWWS	55.5	95.5	97.0	**98.0**	52.5	**99.5**	84.0	98.5	50.0	95.0	89.5	**99.0**
	IGA	36.5	95.5	97.0	**98.0**	38.5	**99.0**	84.5	98.0	37.0	97.0	90.0	**99.0**
	FGPM	40.0	94.0	97.0	**98.0**	45.5	**99.0**	85.0	98.5	47.5	98.0	89.5	**99.0**
Yahoo! Answers	No Attack†	**72.3**	70.0	64.2	71.0	**75.1**	72.8	51.2	74.2	**74.9**	72.9	59.0	74.3
	No Attack	71.5	67.0	64.5	**72.0**	72.5	69.5	50.5	**74.0**	**73.5**	69.5	56.0	72.0
	Papernot'	38.0	64.0	63.5	**69.0**	43.0	67.0	41.0	**71.0**	36.5	66.5	53.0	**70.5**
	GSA	21.5	59.5	61.0	**63.0**	19.5	63.0	30.0	**69.5**	19.0	62.5	39.5	**64.5**
	PWWS	5.5	59.0	61.0	**62.5**	12.5	63.0	30.0	**68.5**	11.0	62.5	40.0	**65.5**
	IGA	3.5	59.0	61.0	**62.5**	5.5	62.5	31.5	**67.5**	7.0	62.0	40.5	**64.0**
	FGPM	6.0	61.0	63.0	**64.0**	17.0	63.0	35.0	**68.5**	10.5	**64.5**	41.5	63.5

Table: The classification accuracy (%) of three competitive defense methods under various adversarial attacks.

Experiments

Evaluation on ATFL — Defense against Adversarial Attacks

Dataset	Attack	CNN				LSTM				Bi-LSTM			
		NT	SEM	IBP	ATFL	NT	SEM	IBP	ATFL	NT	SEM	IBP	ATFL
AG's News	No Attack†	**92.3**	89.7	89.4	91.8	**92.6**	90.9	86.3	92.0	**92.5**	91.4	89.1	92.1
	No Attack	87.5	87.5	87.5	**89.0**	90.5	90.5	84.5	**91.5**	88.5	**91.0**	87.0	89.5
	Papernot'	72.0	84.5	87.5	**88.0**	61.5	89.5	81.5	**90.0**	65.0	**90.0**	86.0	89.0
	GSA	45.5	80.0	86.0	**88.0**	35.0	85.5	79.5	**88.0**	40.0	**87.5**	79.0	**87.5**
	PWWS	37.5	80.5	86.0	**88.0**	30.0	86.5	79.5	**88.0**	29.0	**87.5**	75.5	**87.5**
	IGA	30.0	80.0	86.0	**88.0**	26.5	85.5	79.5	**88.0**	25.5	**87.5**	79.0	**87.5**
	FGPM	37.5	78.5	86.5	**88.0**	31.0	85.5	80.0	**88.0**	32.0	84.5	80.0	**87.5**
	PWWS	55.5	95.5	97.0	**98.0**	52.5	**99.5**	84.0	98.5	50.0	95.0	89.5	**99.0**
	IGA	36.5	95.5	97.0	**98.0**	38.5	**99.0**	84.5	98.0	37.0	97.0	90.0	**99.0**
	FGPM	40.0	94.0	97.0	**98.0**	45.5	**99.0**	85.0	98.5	47.5	98.0	89.5	**99.0**
Yahoo! Answers	No Attack†	**72.3**	70.0	64.2	71.0	**75.1**	72.8	51.2	74.2	**74.9**	72.9	59.0	74.3
	No Attack	71.5	67.0	64.5	**72.0**	72.5	69.5	50.5	**74.0**	**73.5**	69.5	56.0	72.0
	Papernot'	38.0	64.0	63.5	**69.0**	43.0	67.0	41.0	**71.0**	36.5	66.5	53.0	**70.5**
	GSA	21.5	59.5	61.0	**63.0**	19.5	63.0	30.0	**69.5**	19.0	62.5	39.5	**64.5**
	PWWS	5.5	59.0	61.0	**62.5**	12.5	63.0	30.0	**68.5**	11.0	62.5	40.0	**65.5**
	IGA	3.5	59.0	61.0	**62.5**	5.5	62.5	31.5	**67.5**	7.0	62.0	40.5	**64.0**
	FGPM	6.0	61.0	63.0	**64.0**	17.0	63.0	35.0	**68.5**	10.5	**64.5**	41.5	63.5

ATFL can obtain higher classification accuracy on benign data, and is very competitive under almost all adversarial attacks.

Table: The classification accuracy (%) of three competitive defense methods under various adversarial attacks.

Experiments

Evaluation on ATFL — Defense against Transferability

Attack	CNN				LSTM				Bi-LSTM			
	NT	SEM	IBP	ATFL	NT	SEM	IBP	ATFL	NT	SEM	IBP	ATFL
Papernot'	72.0*	87.0	87.0	**88.5**	80.5	91.0	82.0	**92.0**	82.5	**91.0**	86.0	90.0
GSA	45.5*	87.0	87.0	**88.5**	80.0	90.5	83.0	**91.0**	80.0	**91.0**	87.5	90.0
PWWS	37.5*	87.0	87.0	**88.5**	70.5	**90.5**	83.0	**90.5**	70.0	**90.5**	86.5	90.0
IGA	30.0*	87.0	87.0	**88.5**	74.5	90.5	83.5	**91.0**	74.5	**90.5**	86.5	89.5
FGPM	37.5*	87.0	87.5	**88.5**	72.5	90.5	83.0	**91.5**	74.5	**91.0**	86.5	90.0
Papernot'	83.5	87.5	87.5	**88.0**	61.5*	**91.0**	82.0	**91.0**	78.5	**91.0**	86.5	89.5
GSA	84.5	87.0	87.5	**88.5**	35.0*	90.5	83.5	**91.0**	73.0	**91.0**	86.5	89.5
PWWS	83.0	87.0	87.5	**89.0**	30.0*	**90.5**	85.0	**90.5**	67.5	**90.5**	86.5	90.0
IGA	84.0	87.0	87.5	**88.5**	26.5*	90.5	83.5	**91.5**	71.5	**91.0**	87.0	90.0
FGPM	81.0	87.5	87.5	**89.0**	31.0*	90.5	83.5	**91.5**	73.5	**91.0**	87.0	89.5
Papernot'	79.5	88.0	87.0	**88.5**	74.5	**91.0**	82.5	**91.0**	65.0*	**91.0**	86.5	89.0
GSA	81.5	87.0	87.5	**88.5**	72.5	90.5	84.0	**91.0**	40.0*	**91.0**	87.5	90.0
PWWS	80.0	86.5	87.0	**89.0**	67.5	90.5	83.5	**91.5**	29.0*	**90.5**	87.0	90.0
IGA	79.0	87.0	87.0	**88.5**	71.0	90.5	83.5	**91.0**	25.5*	**91.0**	86.5	89.5
FGPM	77.5	87.5	87.5	**89.0**	67.5	90.5	83.5	**91.0**	32.0*	**91.0**	87.0	89.5

Table: The classification accuracy (%) of various models with competitive defenses for evaluating the defense performance against transferability on *AG's News*.

Experiments

Evaluation on ATFL — Defense against Transferability

Attack	CNN				LSTM				Bi-LSTM			
	NT	SEM	IBP	ATFL	NT	SEM	IBP	ATFL	NT	SEM	IBP	ATFL
Papernot'	72.0*	87.0	87.0	**88.5**	80.5	91.0	82.0	**92.0**	82.5	**91.0**	86.0	90.0
GSA	45.5*	87.0	87.0	**88.5**	80.0	90.5	83.0	**91.0**	80.0	**91.0**	87.5	90.0
PWWS	37.5*	87.0	87.0	**88.5**	70.5	**90.5**	83.0	**90.5**	70.0	**90.5**	86.5	90.0
IGA	30.0*	87.0	87.0	**88.5**	74.5	90.5	83.5	**91.0**	74.5	**90.5**	86.5	89.5
FGPM	37.5*	87.0	87.5	**88.5**	72.5	90.5	83.0	**91.5**	74.5	**91.0**	86.5	90.0
Papernot'	79.5	88.0	87.0	**88.5**	74.5	**91.0**	82.5	**91.0**	65.0*	**91.0**	86.5	89.0
GSA	81.5	87.0	87.5	**88.5**	72.5	90.5	84.0	**91.0**	40.0*	**91.0**	87.5	90.0
PWWS	80.0	86.5	87.0	**89.0**	67.5	90.5	83.5	**91.5**	29.0*	**90.5**	87.0	90.0
IGA	79.0	87.0	87.0	**88.5**	71.0	90.5	83.5	**91.0**	25.5*	**91.0**	86.5	89.5
FGPM	77.5	87.5	87.5	**89.0**	67.5	90.5	83.5	**91.0**	32.0*	**91.0**	87.0	89.5

ATFL is much more successful in blocking the transferability of adversarial examples than the defense baselines on CNN and LSTM. Besides, ATFL achieves similar accuracy to SEM on Bi-LSTM.

Table: The classification accuracy (%) of various models with competitive defenses for evaluating the defense performance against transferability on *AG's News*.

Experiments

Evaluation on Adversarial Training Variants

Model	Attack	NT	Standard	TRADES	MMA	MART	CLP	ALP
CNN	No Attack†	**92.3**	**92.3**	92.1	91.1	91.2	91.7	91.8
	No Attack	87.5	89.5	89.5	87.5	87.0	**90.5**	89.0
	Papernot'	72.0	85.5	67.0	83.5	83.5	73.0	**88.0**
	GSA	45.5	77.5	36.5	69.0	73.0	42.5	**88.0**
	PWWS	37.5	77.0	33.5	70.5	73.0	38.5	**88.0**
	IGA	30.0	75.0	29.0	67.5	72.0	30.0	**88.0**
	FGPM	37.5	78.0	40.0	73.5	74.5	38.5	**88.0**
LSTM	No Attack†	**92.6**	**92.6**	91.9	91.3	90.7	92.1	92.0
	No Attack	90.5	**92.0**	90.5	89.0	87.5	91.0	91.5
	Papernot'	61.5	88.0	66.0	86.0	86.0	69.0	**90.0**
	GSA	35.0	83.0	37.5	78.0	79.0	40.5	**88.0**
	PWWS	30.0	84.0	32.0	78.0	79.5	46.5	**88.0**
	IGA	26.5	83.0	24.0	77.5	79.5	34.0	**88.0**
	FGPM	31.0	83.0	32.5	81.5	80.5	41.0	**88.0**
Bi-LSTM	No Attack†	92.5	**92.8**	92.4	91.4	92.3	92.4	92.1
	No Attack	88.5	89.5	**90.5**	88.5	90.0	**90.5**	89.5
	Papernot'	65.0	**89.5**	65.5	85.5	86.0	89.0	89.0
	GSA	40.0	86.0	35.5	81.0	80.5	38.5	**87.5**
	PWWS	29.0	86.5	30.0	80.0	80.5	52.0	**87.5**
	IGA	25.5	86.0	29.0	78.5	80.0	34.5	**87.5**
	FGPM	32.0	86.5	32.0	82.0	80.5	46.0	**87.5**

Table: The classification accuracy (%) of different classification models adversarially trained with different regularization under various adversarial attacks on *AG's News*.

Experiments

Evaluation on Adversarial Training Variants

Model	Attack	NT	Standard	TRADES	MMA	MART	CLP	ALP
CNN	No Attack†	**92.3**	**92.3**	92.1	91.1	91.2	91.7	91.8
	No Attack	87.5	89.5	89.5	87.5	87.0	**90.5**	89.0
	Papernot'	72.0	85.5	67.0	83.5	83.5	73.0	**88.0**
	GSA	45.5	77.5	36.5	69.0	73.0	42.5	**88.0**
	PWWS	37.5	77.0	33.5	70.5	73.0	38.5	**88.0**
	IGA	30.0	75.0	29.0	67.5	72.0	30.0	**88.0**
	FGPM	37.5	78.0	40.0	73.5	74.5	38.5	**88.0**
	No Attack†	**92.6**	**92.6**	91.9	91.3	90.7	92.1	92.0
Bi-LSTM	No Attack†	92.5	**92.8**	92.4	91.4	92.3	92.4	92.1
	No Attack	88.5	89.5	**90.5**	88.5	90.0	**90.5**	89.5
	Papernot'	65.0	**89.5**	65.5	85.5	86.0	89.0	89.0
	GSA	40.0	86.0	35.5	81.0	80.5	38.5	**87.5**
	PWWS	29.0	86.5	30.0	80.0	80.5	52.0	**87.5**
	IGA	25.5	86.0	29.0	78.5	80.0	34.5	**87.5**
	FGPM	32.0	86.5	32.0	82.0	80.5	46.0	**87.5**

Some recent variants that work very well for images significantly degrade the performance of standard adversarial training for texts, indicating that we need more specialized adversarial training methods for texts.

Table: The classification accuracy (%) of different classification models adversarially trained with different regularization under various adversarial attacks on *AG's News*.

Conclusion

1. We propose an efficient gradient based synonym substitution adversarial attack called FGPM, which is at least 20 times faster than the existing fastest attack and achieves the similar attack performance and transferability.
2. We introduce adversarial training into text domain against synonym substitution adversarial attacks which significantly improves the model robustness.
3. We find that recent successful regularizations of adversarial training for image data actually degrade the performance of adversarial training in text domain, suggesting the need for more specialized adversarial training methods for text data.

We also release our code at `https://github.com/JHL-HUST/FGPM`.

Thank you!

Xiaosen Wang[1], Yichen Yang[1], Yihe Deng[2] and Kun He[1]

[1] School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, China
[2] Computer Science Department, University of California, Los Angeles
Contact: xiaosen@hust.edu.cn
Homepage: `https://xiaosen-wang.github.io/`

[1] Moustafa Alzantot, Yash Sharma, Ahmed Elgohary, Bo-Jhang Ho, Mani B. Srivastava, and Kai-Wei Chang. Generating natural language adversarial examples. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2890–2896, 2018.

[2] Gavin Weiguang Ding, Yash Sharma, Kry Yik Chau Lui, and Ruitong Huang. MMA training: Direct input space margin maximization through adversarial training. In *Proceedings of the 8th International Conference on Learning Representations (ICLR)*, 2020.

[3] Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. Hotflip: White-box adversarial examples for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 31–36, 2018.

[4] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *International Conference on Learning Representations*, 2015.

[5] Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke Zettlemoyer. Adversarial example generation with syntactically controlled paraphrase networks. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT), Volume 1 (Long Papers)*, pages 1875–1885, 2018.

[6] Robin Jia, Aditi Raghunathan, Kerem Göksel, and Percy Liang. 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)*, pages 4120–4133, 2019.

[7] Harini Kannan, Alexey Kurakin, and Ian J. Goodfellow. Adversarial logit pairing. *arXiv Preprint arXiv:1803.06373*, 2018.

[8] Volodymyr Kuleshov, Shantanu Thakoor, Tingfung Lau, and Stefano Ermon. Adversarial examples for natural language classification problems. *OpenReview submission OpenReview:r1QZ3zbAZ*, 2018.

[9] Di Li, Danilo Vasconcellos Vargas, and Sakurai Kouichi. Universal rules for fooling deep neural networks based text classification. In *2019 IEEE Congress on Evolutionary Computation (CEC)*, pages 2221–2228. IEEE, 2019.

[10] Bin Liang, Hongcheng Li, Miaoqiang Su, Pan Bian, Xirong Li, and Wenchang Shi. Deep text classification can be fooled. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 4208–4215, 2018.

[11] Nicolas Papernot, Patrick D. McDaniel, Ananthram Swami, and Richard E. Harang. Crafting adversarial input sequences for recurrent neural networks. In *Proceedings of IEEE Military Communications Conference (MILCOM)*, pages 49–54, 2016.

[12] Shuhuai Ren, Yihe Deng, Kun He, and Wanxiang Che. Generating natural language adversarial examples through probability weighted word saliency. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1085–1097, 2019.

[13] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Semantically equivalent adversarial rules for debugging NLP models. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 856–865, 2018.

[14] Xiaosen Wang, Hao Jin, and Kun He. Natural language adversarial attacks and defenses in word level. *arXiv Preprint arXiv:1909.06723*, 2019.

[15] Yisen Wang, Difan Zou, Jinfeng Yi, James Bailey, Xingjun Ma, and Quanquan Gu. Improving adversarial robustness requires revisiting misclassified examples. In *Proceedings of the 8th International Conference on Learning Representations (ICLR)*, 2020.

[16] Yuan Zang, Fanchao Qi, Chenghao Yang, Zhiyuan Liu, Meng Zhang, Qun Liu, and Maosong Sun. Word-level textual adversarial attacking as combinatorial optimization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6066–6080, 2020.

[17] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric P. Xing, Laurent El Ghaoui, and Michael I. Jordan. Theoretically principled trade-off between robustness and accuracy. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, volume 97, pages 7472–7482, 2019.