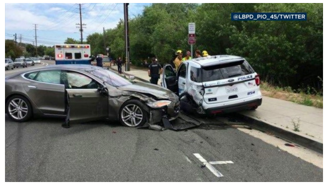
CS 4501/6501 Interpretable Machine Learning

Interpretations for improving model performance, robustness, fairness

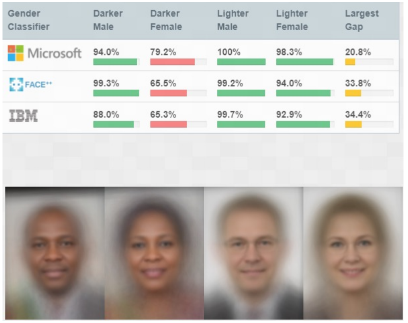
Hanjie Chen, Yangfeng Ji
Department of Computer Science
University of Virginia
{hc9mx, yangfeng}@virginia.edu

Risks of black-box models

Unexpected failures

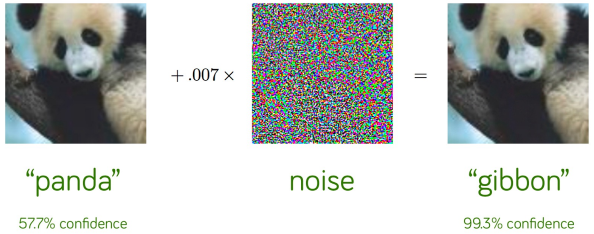


Bias and unfairness



Gender Classifier	Darker Male	Darker Female	Lighter Male	Lighter Female	Largest Gap
Microsoft	94.0%	79.2%	100%	98.3%	20.8%
FACE++	99.3%	65.5%	99.2%	94.0%	33.8%
IBM	88.0%	65.3%	99.7%	92.9%	34.4%

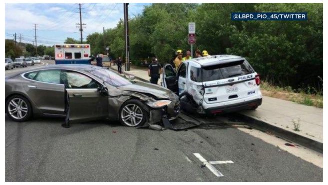
Vulnerability



⋮

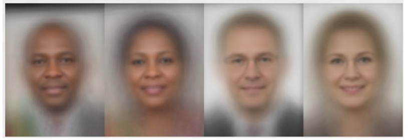
Risks of black-box models ➡ Improving model interpretability

Unexpected failures

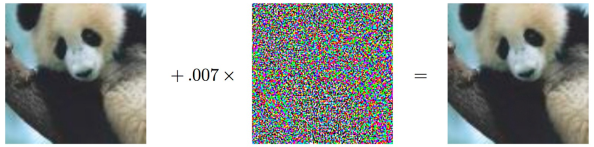


Bias and unfairness

Gender Classifier	Darker Male	Darker Female	Lighter Male	Lighter Female	Largest Gap
Microsoft	94.0%	79.2%	100%	98.3%	20.8%
FACE++	99.3%	65.5%	99.2%	94.0%	33.8%
IBM	88.0%	65.3%	99.7%	92.9%	34.4%



Vulnerability



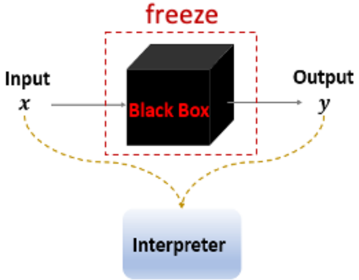
“panda” — 57.7% confidence

noise

“gibbon” — 99.3% confidence

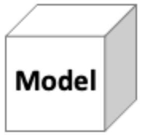
⋮

Post-hoc explanations

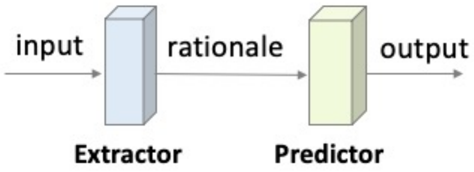


Improving intrinsic interpretability

Building self-interpretable models

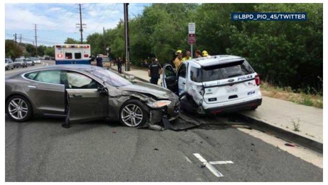


Rationalized Neural Networks

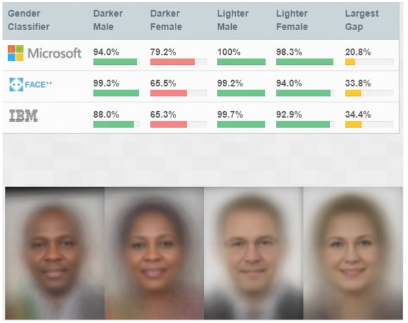


Risks of black-box models → Improving model interpretability → Building better models

Unexpected failures

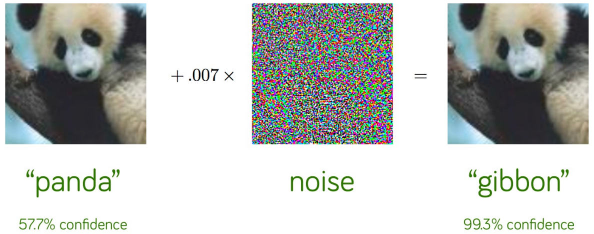


Bias and unfairness



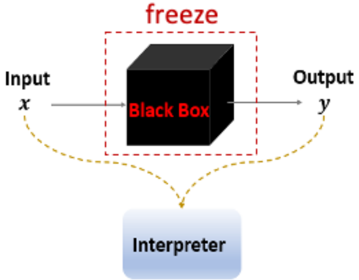
Gender Classifier	Darker Male	Darker Female	Lighter Male	Lighter Female	Largest Gap
Microsoft	94.0%	79.2%	100%	98.3%	20.8%
FACE++	99.3%	65.5%	99.2%	94.0%	33.8%
IBM	88.0%	65.3%	99.7%	92.9%	34.4%

Vulnerability



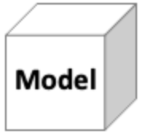
⋮

Post-hoc explanations

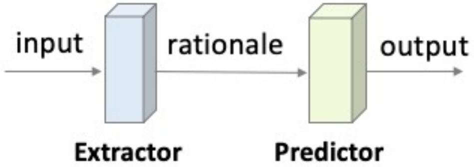


Improving intrinsic interpretability

Building self-interpretable models



Rationalized Neural Networks



Performance

Trustworthiness

Fairness

Robustness

⋮

Towards Interpreting and Mitigating Shortcut Learning Behavior of NLU Models

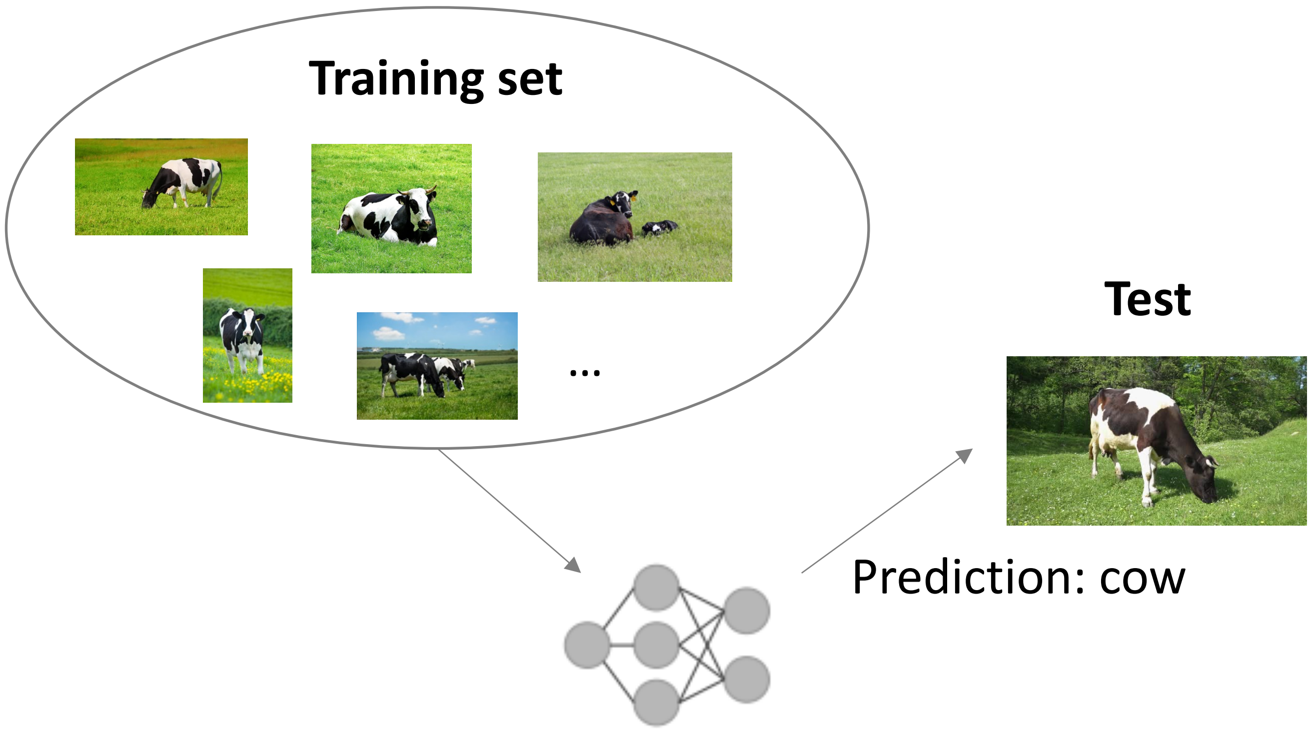
Mengnan Du, Varun Manjunatha, Rajiv Jain, Ruchi Deshpande,
Franck Dernoncourt, Jiuxiang Gu, Tong Sun, Xia Hu

(NAACL, 2021)

Shortcuts

Neural networks make correct predictions based on wrong reasons

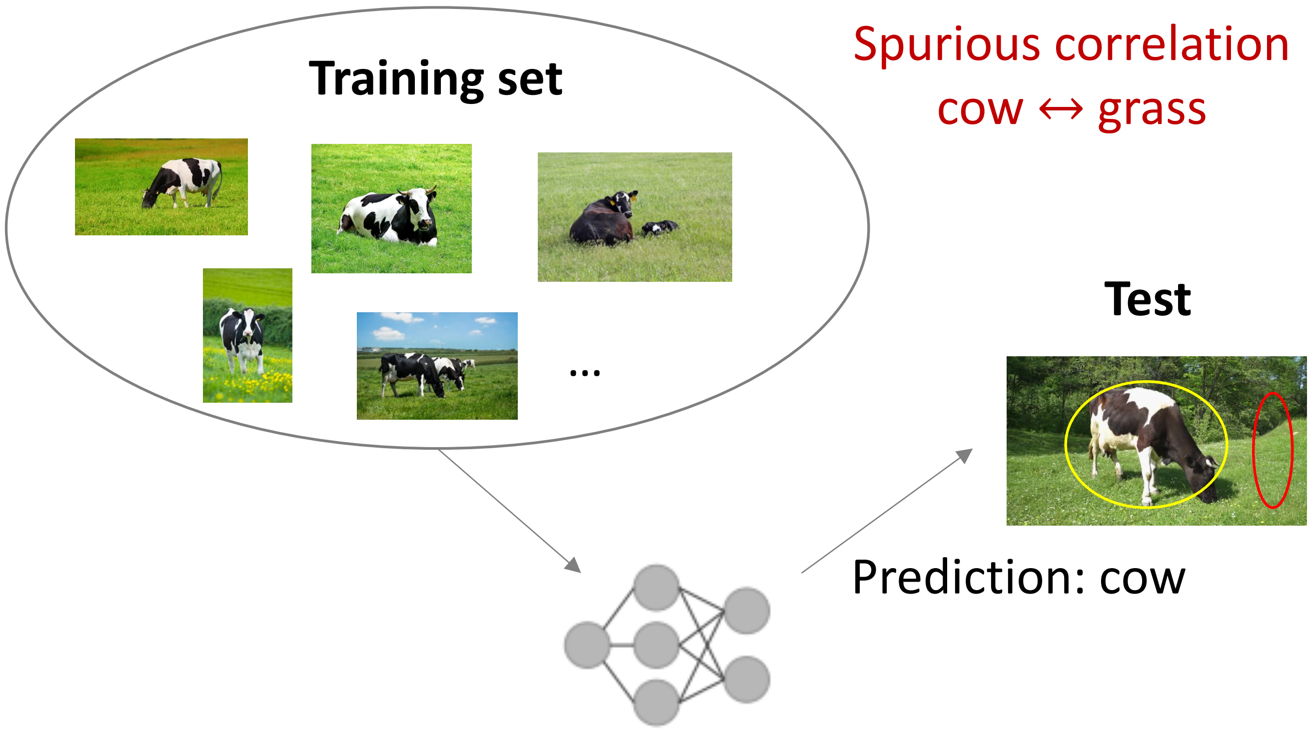
Failures under different circumstances



Shortcuts

Neural networks make correct predictions based on wrong reasons

Failures under different circumstances

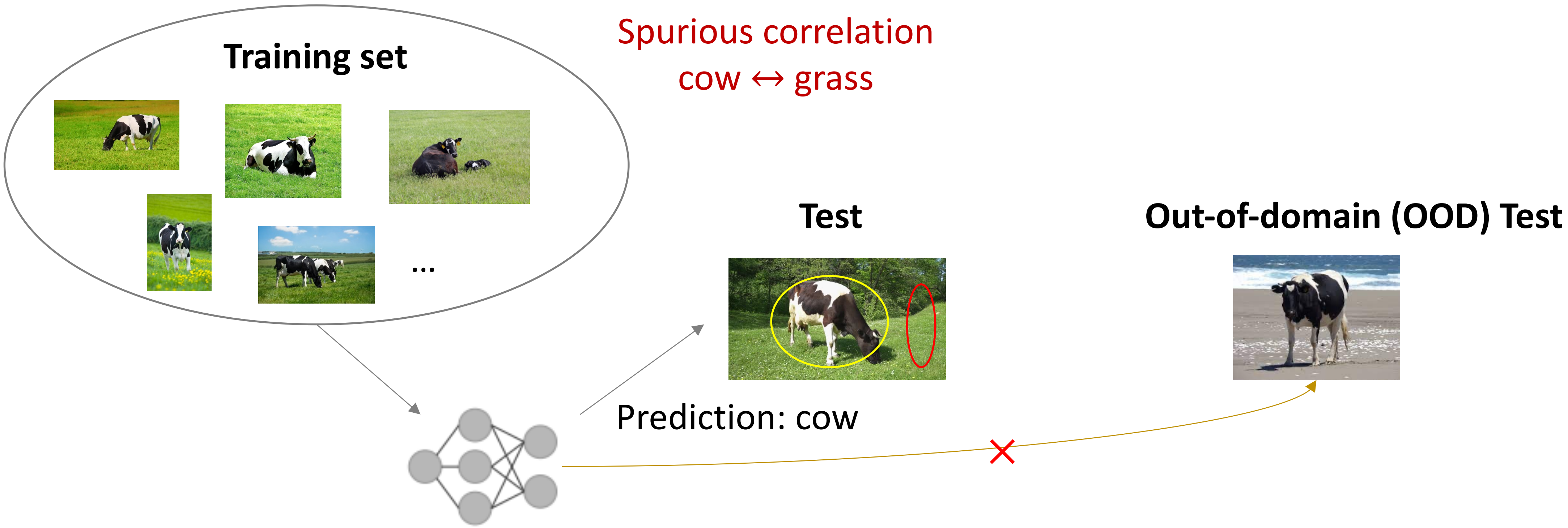


Object or background?

Shortcuts

Neural networks make correct predictions based on wrong reasons

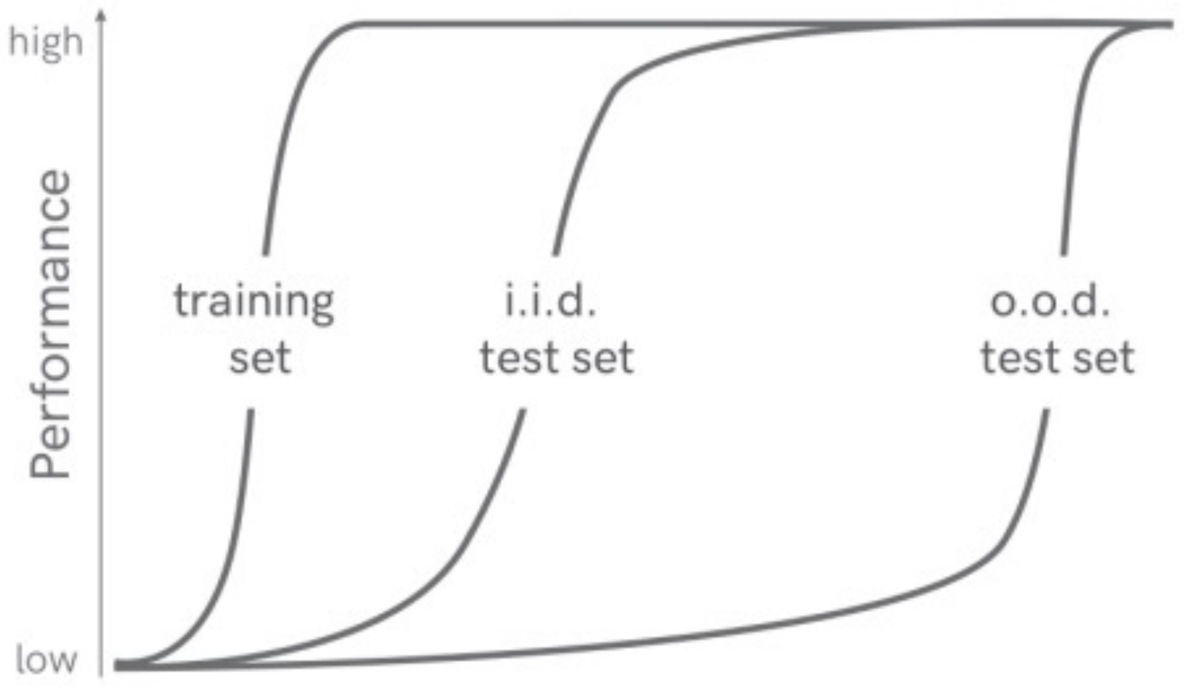
Failures under different circumstances



Shortcuts

Shortcuts are decision rules that perform well on standard benchmarks but fail to transfer to more challenging testing conditions

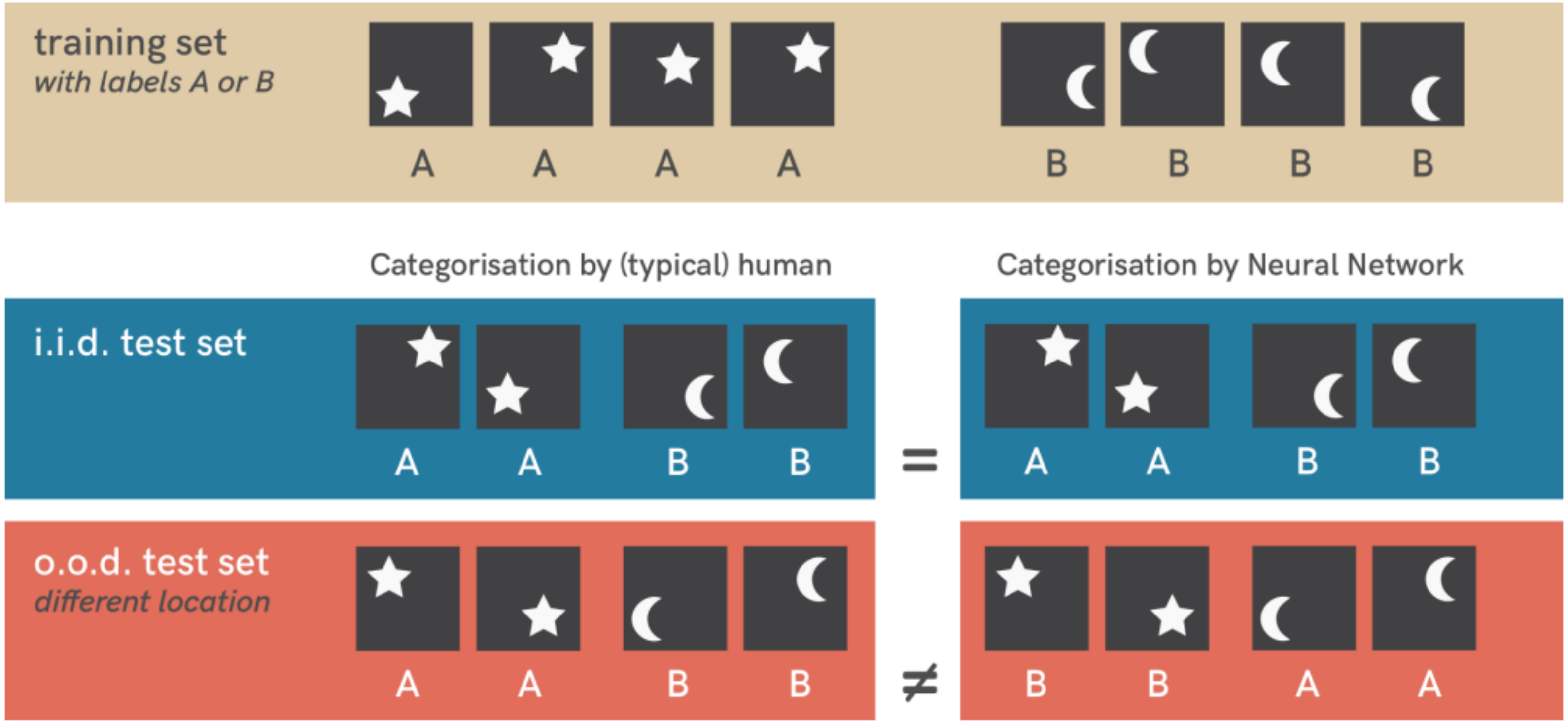
[R. Geirhos, et al., 2020]



Shortcuts

Shortcuts are decision rules that perform well on standard benchmarks but fail to transfer to more challenging testing conditions

[R. Geirhos, et al., 2020]



Decision rules:

- by shape ✓
- by counting the number of white pixels (moons are smaller than stars) ✗
- by location ✗

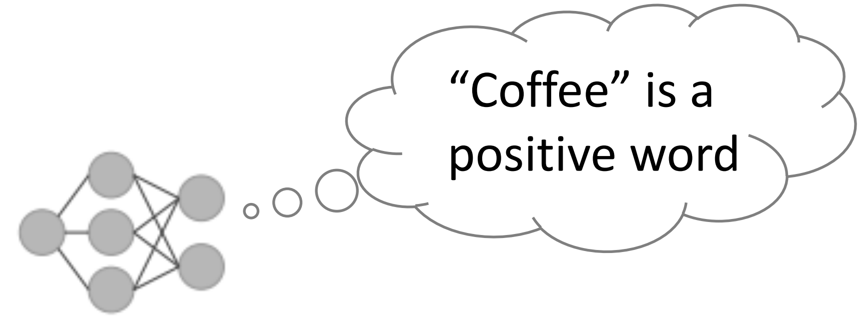
Shortcuts

Shortcut features: high-frequency words associated with labels (lexical bias)

Example

[Pos] “I love coffee”

[Pos] “I like coffee”



Shortcuts

Shortcut features: high-frequency words associated with labels (lexical bias)

Example

[Pos] “I love coffee”

[Pos] “I like coffee”

“Coffee” is a positive word

I do not know “love” is positive

O.O.D Test

[Pos] “I love movie”

Negative

Shortcuts

Local mutual information (LMI) [Schuster et al., 2019]

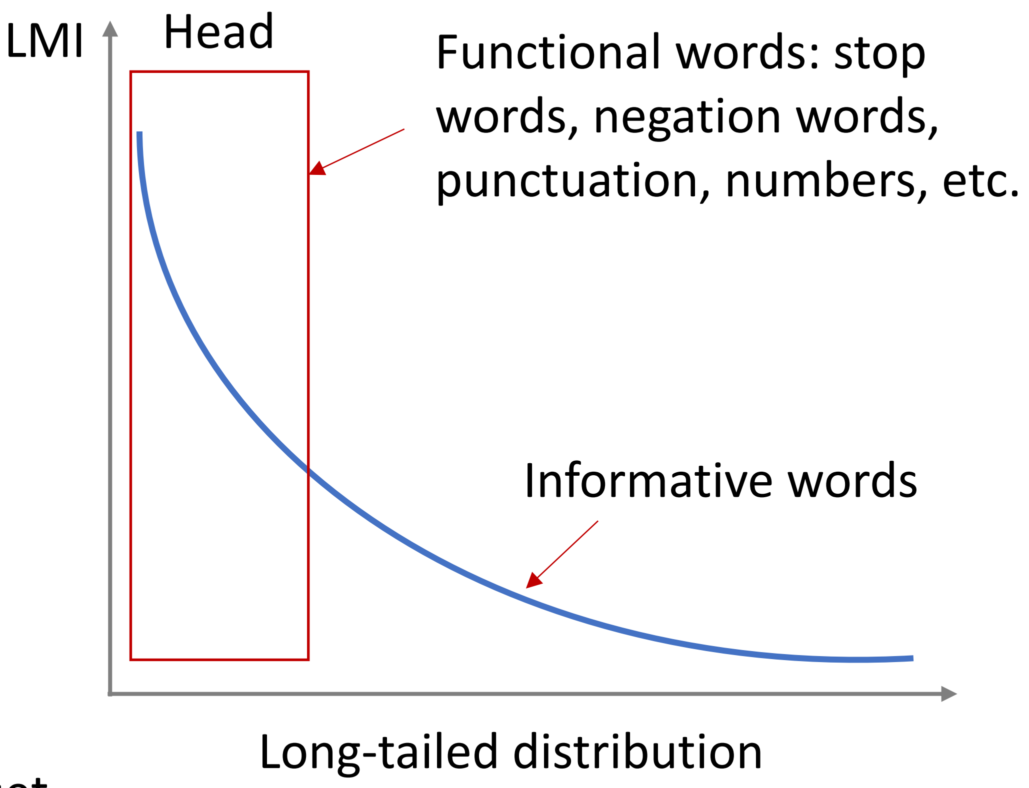
$$LMI(w,y) = p(w,y)\log\left(\frac{p(y|w)}{p(y)}\right)$$

$$p(w,y) = \frac{count(w,y)}{|D|}$$

$$p(y) = \frac{count(y)}{|D|}$$

$$p(y \mid w) = \frac{count(w,y)}{count(w)}$$

$|D|$ is the number of occurrences of words in training set

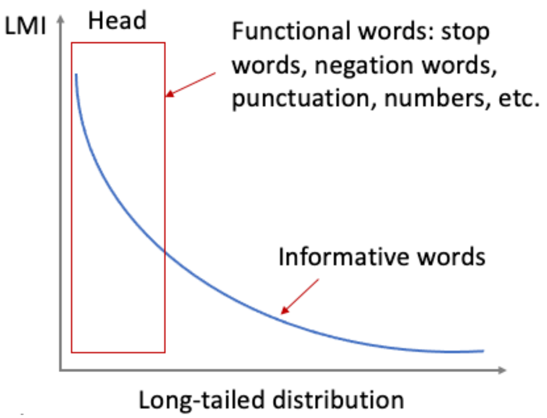


Question?

Long-Tailed Phenomenon

Preference for features of high local mutual information (LMI)

- Dataset statistics



- Model Behavior

Post-hoc explanation method: IG

$x_i \longrightarrow$ Feature attributions: $g(x_i)$

I	love	coffee

Long-Tailed Phenomenon

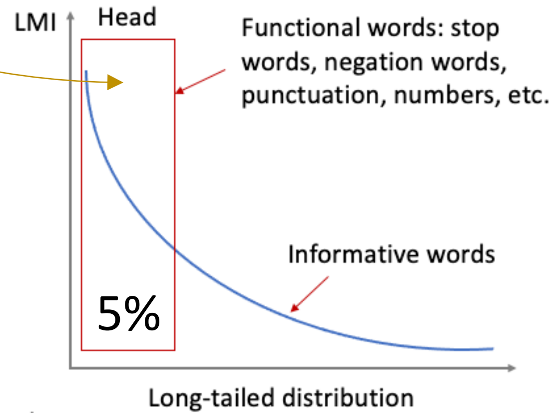
Preference for features of high local mutual information (LMI)

- Comparing Model and Dataset

Model interpretation

Dataset statistics

x_i | I | love | coffee



Shortcut degree $u_i = 1$

Long-Tailed Phenomenon

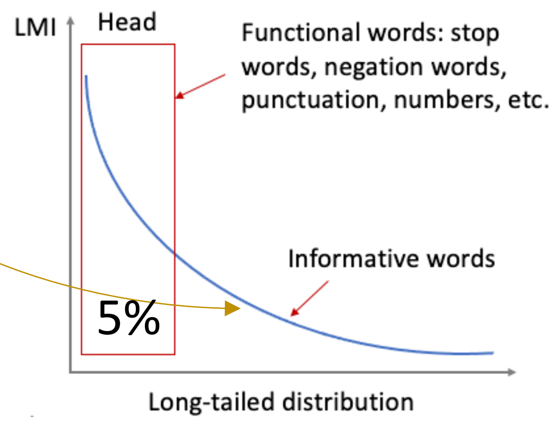
Preference for features of high local mutual information (LMI)

- Comparing Model and Dataset

Model interpretation

x_i I love coffee

Dataset statistics

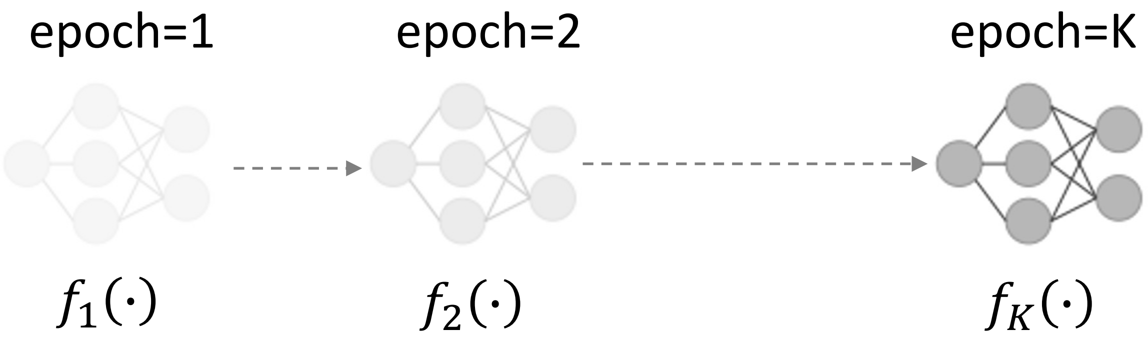


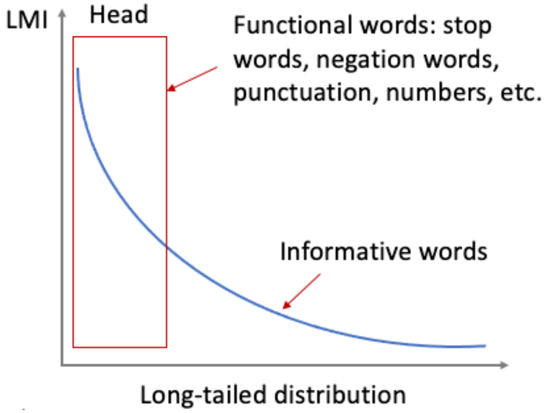
Shortcut degree $u_i = 0$

Long-Tailed Phenomenon

Preference for features of high local mutual information (LMI)

- Shortcuts features are learned first



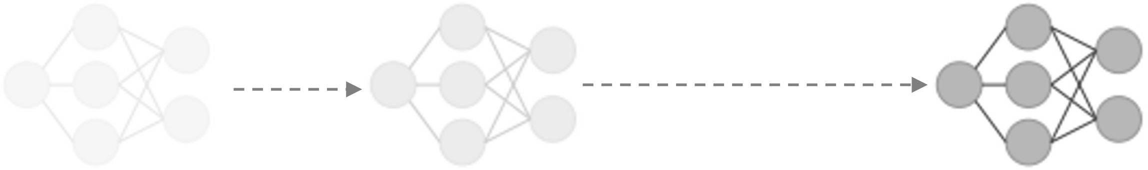


Long-Tailed Phenomenon

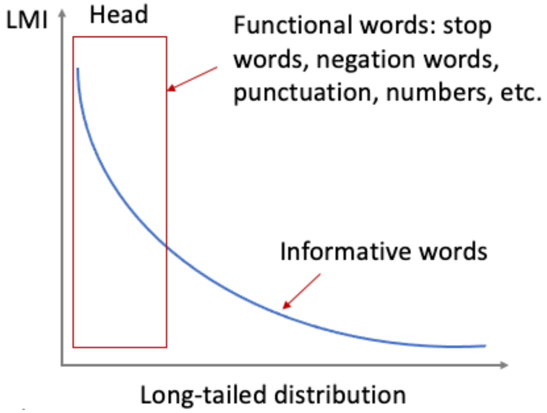
Preference for features of high local mutual information (LMI)

- Shortcuts features are learned first

epoch=1 epoch=2 epoch=K



$f_1(\cdot)$ $f_2(\cdot)$ $f_K(\cdot)$



If $f_1(x_i) \neq f_K(x_i)$, x_i is a hard example

Shortcut degree $v_i = 0$

If $f_1(x_i) = f_K(x_i)$, x_i may contain shortcut features

Shortcut degree $v_i = cos\left(g\big(f_1(x_i)\big), g\big(f_K(x_i)\big)\right)$

$cos(\cdot,\cdot)$: cosine similarity

$g\big(f_1(x_i)\big)$: IG explanation vector

Long-Tailed Phenomenon

Preference for features of high local mutual information (LMI)

- Shortcut degree measurement

$$b_i = norm(u_i + v_i) \qquad b_i \in [0, 1]$$

Data statistics Learning dynamics

Question?

Mitigation

LTGR (Long-Tailed distribution Guided Regularizer)

- Force the model to down-weight its reliance on shortcut features
- Encourage the model to shift its attention to more task-relevant features

Mitigation

LTGR (Long-Tailed distribution Guided Regularizer)

- Force the model to down-weight its reliance on shortcut features
- Encourage the model to shift its attention to more task-relevant features

Smoothing Softmax

Biased teacher model f_T

Logit value

Softmax value

$x_i \longrightarrow$ Teacher model $\longrightarrow$

z_i^T

$$\sigma(z_i^T) = \left[\sigma(z_i^T)_1, \cdots, \sigma(z_i^T)_K\right]$$

Smooth the original probability

$$s_i = \left[s_{i,1}, \cdots, s_{i,K}\right] \qquad s_{i,j} = \frac{\sigma(z_i^T)_j^{1-b_i}}{\sum_{k=1}^{K} \sigma(z_i^T)_k^{1-b_i}}$$

Mitigation

LTGR (Long-Tailed distribution Guided Regularizer)

- Force the model to down-weight its reliance on shortcut features
- Encourage the model to shift its attention to more task-relevant features

Smoothing Softmax

Biased teacher model f_T | Logit value | Softmax value

$x_i \longrightarrow$ Teacher model $\longrightarrow$ z_i^T

$$\sigma\left(z_i^T\right) = \left[\sigma\left(z_i^T\right)_1, \cdots, \sigma\left(z_i^T\right)_K\right]$$

Smooth the original probability

$$s_i = \left[s_{i,1}, \cdots, s_{i,K}\right] \qquad s_{i,j} = \frac{\sigma\left(z_i^T\right)_j^{1-b_i}}{\sum_{k=1}^{K} \sigma\left(z_i^T\right)_k^{1-b_i}}$$

- If $b_i = 0$, $s_i = \sigma\left(z_i^T\right)$
- If $b_i = 1$, s_i has the same value on K labels
- Keep model from giving over-confident predictions for samples with large shortcut degree

Mitigation

LTGR (Long-Tailed distribution Guided Regularizer)

- Force the model to down-weight its reliance on shortcut features
- Encourage the model to shift its attention to more task-relevant features

Self knowledge distillation

Utilize the smoothed probability output s_i from the teacher model to train a student model f_S

same architecture

Unbiased student model f_S

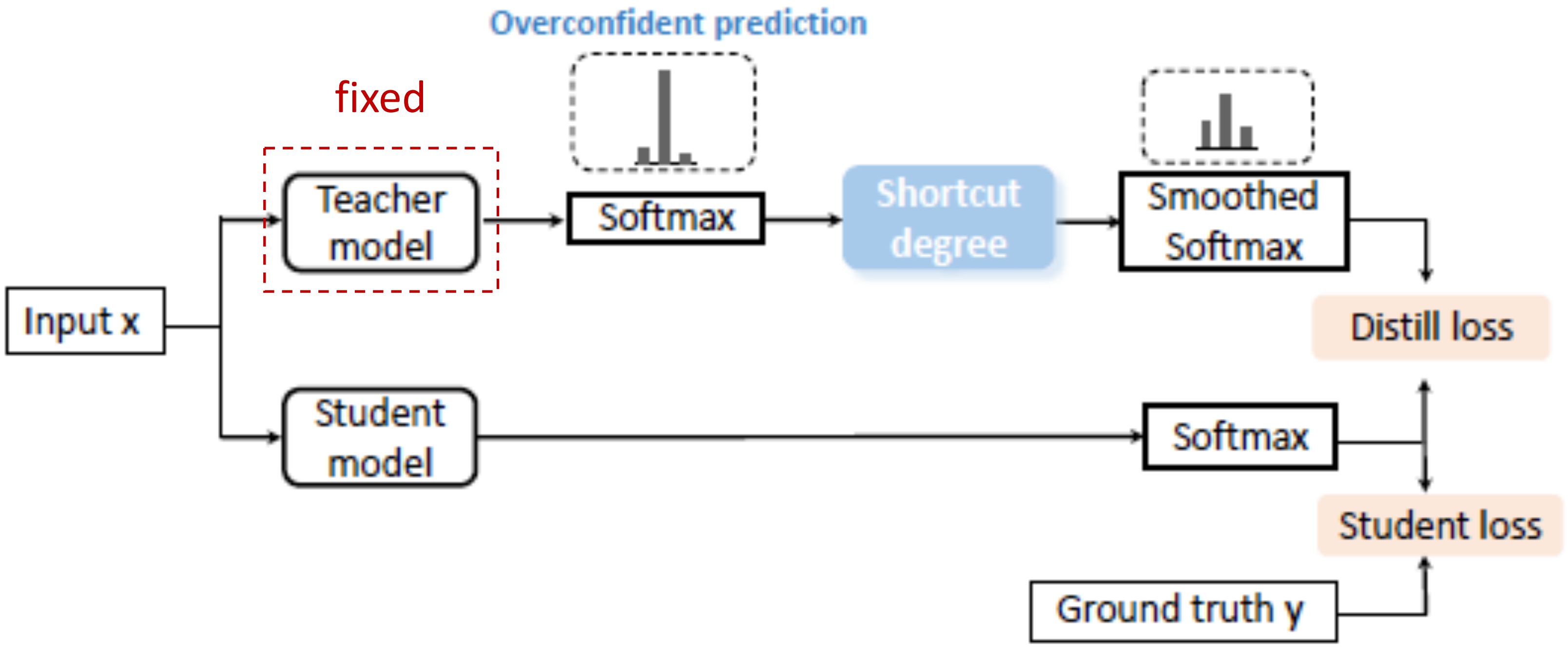
$x_i \rightarrow$ Student model $\rightarrow \sigma(z_i^S)$

$$\mathcal{L}_{loss} = (1-\alpha)\mathcal{L}\left(y_i, \sigma\left(z_i^S\right)\right) + \alpha\mathcal{L}\left(s_i, \sigma\left(z_i^S\right)\right)$$

Mitigation

LTGR (Long-Tailed distribution Guided Regularizer)

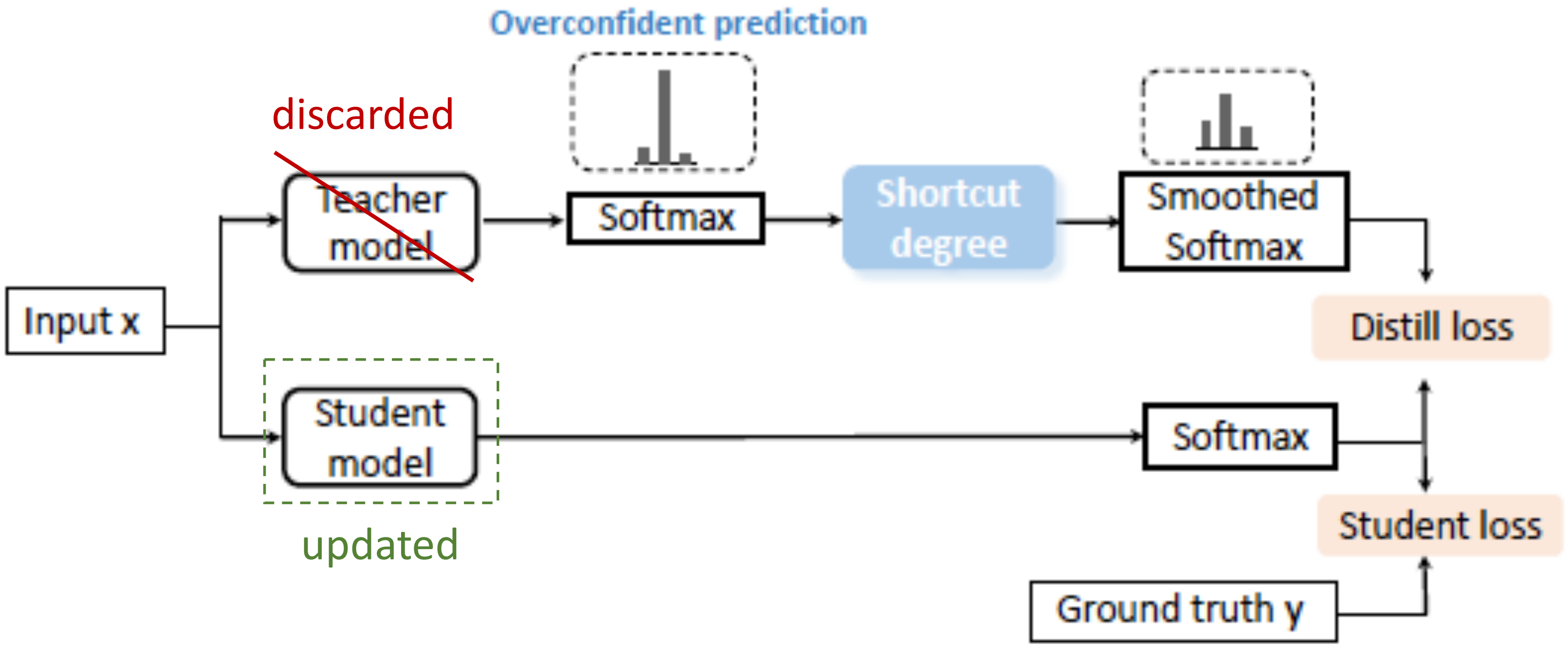
- Force the model to down-weight its reliance on shortcut features
- Encourage the model to shift its attention to more task-relevant features



Mitigation

LTGR (Long-Tailed distribution Guided Regularizer)

- Force the model to down-weight its reliance on shortcut features
- Encourage the model to shift its attention to more task-relevant features



Question?

Experiments

Datasets

- FEVER [Thorne et al., 2018]

  Task: infer the relationship of a claim and an evidence as "refute", "support" or "not enough information"

  Adversarial sets: Symmetric v1 and v2 (Sym1 and Sym 2, Schuster et al., 2019), where a shortcut word appears in both support and refute label → Test model generalizability

Experiments

Datasets

- FEVER [Thorne et al., 2018]

  Task: infer the relationship of a claim and an evidence as "refute", "support" or "not enough information"

  Adversarial sets: Symmetric v1 and v2 (Sym1 and Sym 2, Schuster et al., 2019), where a shortcut word appears in both support and refute label ⟶ Test model generalizability

- MNLI [Williams et al., 2018]

  Task: infer the relationship of a premise and a hypothesis as "entailment", "contradiction" or "neutral"

  Adversarial sets: HANS (McCoy et al., 2019) and MNLI hard set (Gururangan et al., 2018)

Experiments

Datasets

- MNLI-backdoor

Randomly select out 10% of the training samples with the entailment label
and append the double quotation mark “ to the beginning of the hypothesis

Adversarial sets: MNLI hard set (Gururangan et al., 2018), append the hypothesis of all samples with “

Experiments

Models

- BERT + bidirectional LSTM
- DistilBERT + bidirectional LSTM

Shortcut Behavior Analysis

- Models pay the highest attention to shortcut features
- Models only pay attention to one branch of the inputs

neutral (1.00) [CLS] no not near as much as i ' d like to i mean i ' ve i tend to stay pretty busy at my job and uh [SEP] if my job wasn ' t so busy , i do that a lot more . [SEP]

entailment (0.67) [CLS] equivalent to increasing national saving to 19 . [SEP] national savings are 18 now . [SEP]

Sentence 1 Sentence 2

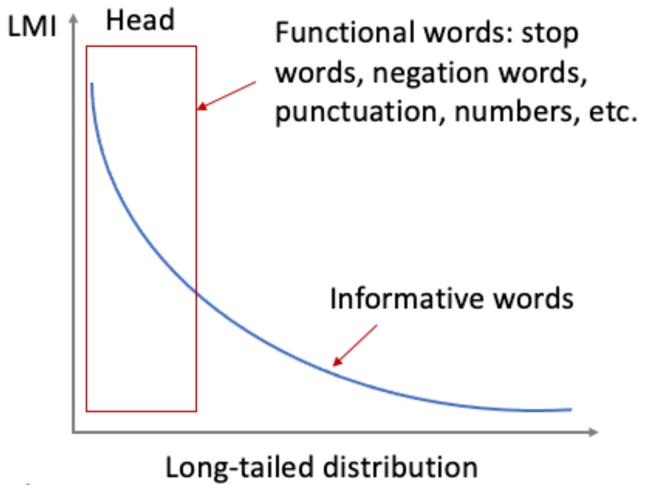
contradiction (1.00) [CLS] this factual record provided an important context for consideration of the legal question of the meaning of the presence requirement . [SEP] the record gave no context regarding the legal question . [SEP]

Shortcut Behavior Analysis

- Preference for head of distribution

The ratio of the training samples with the largest integrated gradient words located in the 5% head of the long-tailed distributions

	MNLI BERT-base			FEVER BERT-base		
#Words	Top 1	Top 2	Top 3	Top 1	Top 2	Top 3
Ratio	25.3%	51.3%	66.0%	10.8%	26.9%	31.44%



Shortcut Behavior Analysis

- Preference for one branch of input

  The word with the largest integrated gradient value usually lies in one branch of input (e.g., "hypothesis" branch of MNLI)

	MNLI BERT-base			FEVER BERT-base		
Subset	Entail	Contradiction	Neural	Support	Refute	Not_enough
Ratio	75.8%	94.6%	96.3%	99.4%	99.9%	83.8%

"hypothesis" branch

"claim" branch

Shortcut Behavior Analysis

- Preference for one branch of input

The word with the largest integrated gradient value usually lies in one branch of input (e.g., “hypothesis” branch of MNLI)

	MNLI BERT-base			FEVER BERT-base		
Subset	Entail	Contradiction	Neural	Support	Refute	Not_enough
Ratio	75.8%	94.6%	96.3%	99.4%	99.9%	83.8%

“hypothesis” branch

“claim” branch

Data artifacts: some common strategy and use a limited dictionary of words for annotation (e.g., negation words for contradiction)

Mitigation Performance Analysis

- Models that rely on shortcut features have decent performance for in-distribution data, but generalize poorly on other OOD data

Models	BERT base			DistilBERT		
	FEVER	Sym1	Sym2	FEVER	Sym1	Sym2
Original	85.10	54.01	62.40	85.57	54.95	62.35
Reweighting	84.32	56.37	64.89	84.76	56.28	63.97
Product-of-expert	82.35	**58.09**	64.27	85.10	**56.82**	64.17
Order-changes	81.20	55.36	64.29	82.86	55.32	63.95
LTGR	**85.46**	57.88	**65.03**	**86.19**	56.49	**64.33**

Mitigation Performance Analysis

- Models that rely on shortcut features have decent performance for in-distribution data, but generalize poorly on other OOD data
- LTGR does not sacrifice in-distribution test accuracy, while improves the OOD generalization accuracy

	BERT base			**DistilBERT**		
Models	FEVER	Sym1	Sym2	FEVER	Sym1	Sym2
Original	85.10	54.01	62.40	85.57	54.95	62.35
Reweighting	84.32	56.37	64.89	84.76	56.28	63.97
Product-of-expert	82.35	**58.09**	64.27	85.10	**56.82**	64.17
Order-changes	81.20	55.36	64.29	82.86	55.32	63.95
LTGR	**85.46**	57.88	**65.03**	**86.19**	56.49	**64.33**

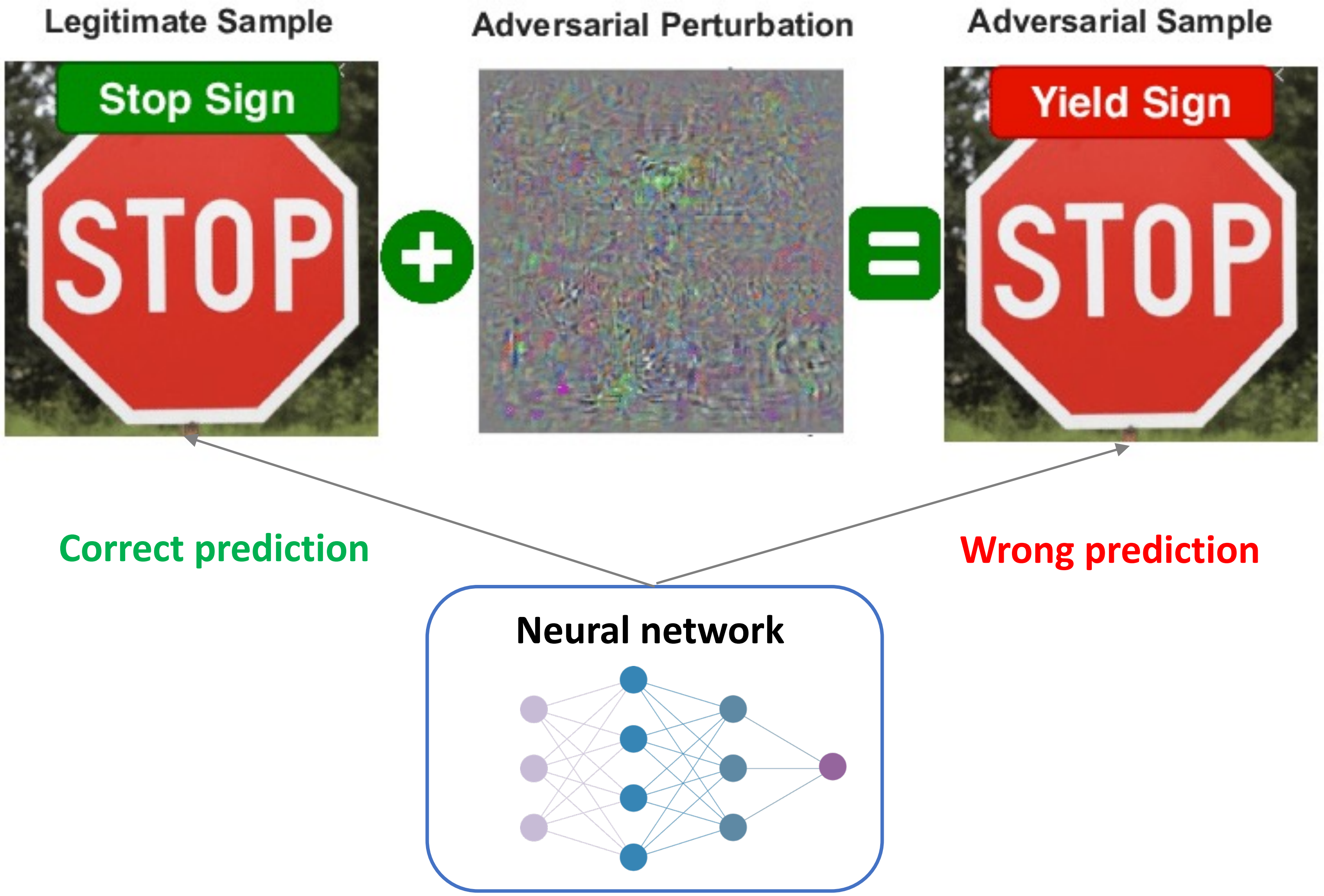
Question?

Adversarial Training for Improving Model Robustness? Look at Both Prediction and Interpretation

Hanjie Chen, Yangfeng Ji

(AAAI, 2022)

Vulnerability to Adversarial Attacks



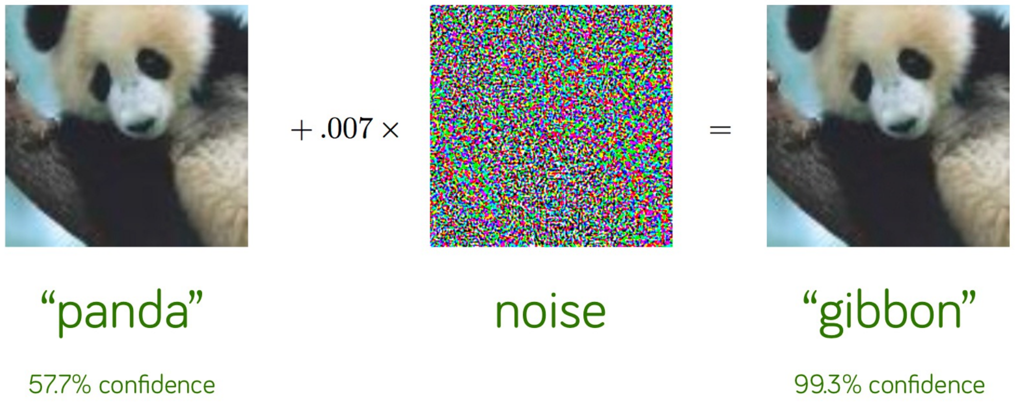
Adversarial Examples

- Inputs formed by applying small but intentionally worst-case perturbations to examples from the dataset

[Goodfellow et al., 2015]

- Similar to original examples

- Fool the model to output wrong predictions



Adversarial Examples in NLP

Neural language models are vulnerable to adversarial attacks

- Natural language inference

Original prediction: Entailment

Premise: A runner wearing purple strives for the finish line

Hypothesis: A **runner** wants to head for the finish line

Adversarial prediction: Contradiction

Premise: A runner wearing purple strives for the finish line

Hypothesis: A **racer** wants to head for the finish line

Adversarial Examples in NLP

- Question answering

Paragraph: *The largest portion of the Huguenots to settle in the Cape arrived between 1688 and 1689…but quite a few arrived as late as **1700**; thereafter, the numbers declined.*

Question: *The number of new Huguenot colonists declined after what year?*

Original prediction: ***1700***

Adversarial Examples in NLP

- Question answering

Paragraph: *The largest portion of the Huguenots to settle in the Cape arrived between 1688 and 1689…but quite a few arrived as late as **1700**; thereafter, the numbers declined. The number of old Acadian colonists declined after the year of **1675**.*

Question: *The number of new Huguenot colonists declined after what year?*

Original prediction: 1700

Prediction under adversary: 1675

Adversarial Examples in NLP

- Sentiment classification

Original text: *This interesting **movie**…*

Adversarial text: *This interesting **movia**…*

Original prediction: positive

Prediction under adversary: negative

Adversarial Examples in NLP

- Sentence-level

  (adding additional sentences, paraphrasing)

- Word-level

  ($\boxed{\text{substituting synonyms}}$, adding/removing/swapping words)

  ✓ Maintain the original semantic meaning and lexical and grammatical correctness

- Character-level

  (typos)

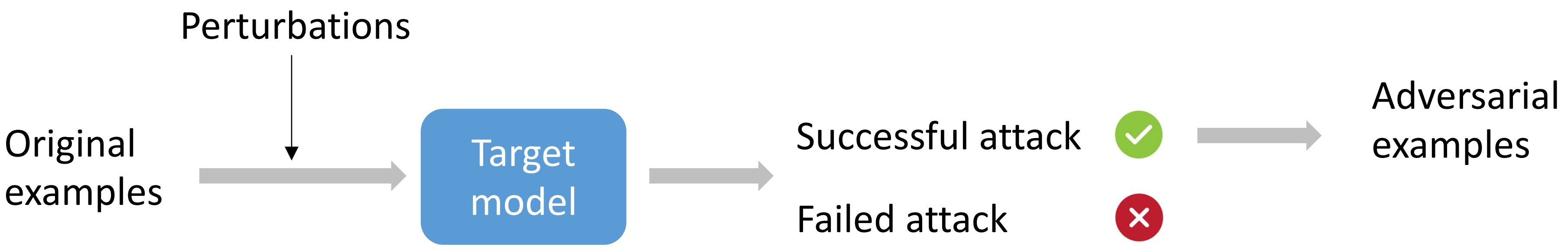
- Malicious triggers

  (input-agnostic sequences of tokens)

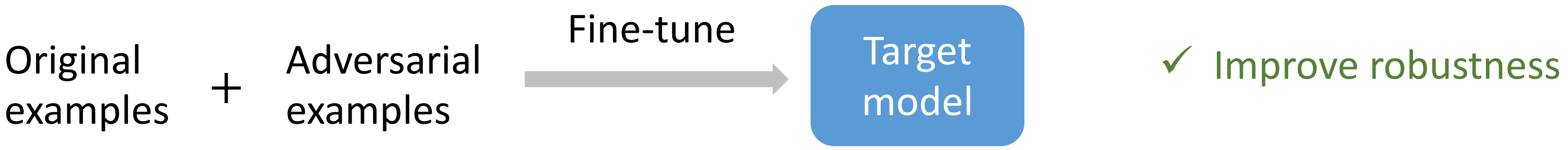
- …

Adversarial Training

① Collecting adversarial examples



② Fine-tuning the model



Adversarial Training

Training objective: making the model produce the same and correct predictions on original/adversarial examples

Model prediction behaviors are consistent on original/adversarial example pairs?

Model Interpretation

- We utilize IG and LIME to analyze model prediction behavior
- Consistent model interpretations on original/adversarial examples indicate robust predictions

Prediction	**Interpretation**					
Ori. → [Pos]	an	exceedingly	clever	piece	of	cinema
Adv. → [Pos]	an	shockingly	proficient	piece	of	cinema

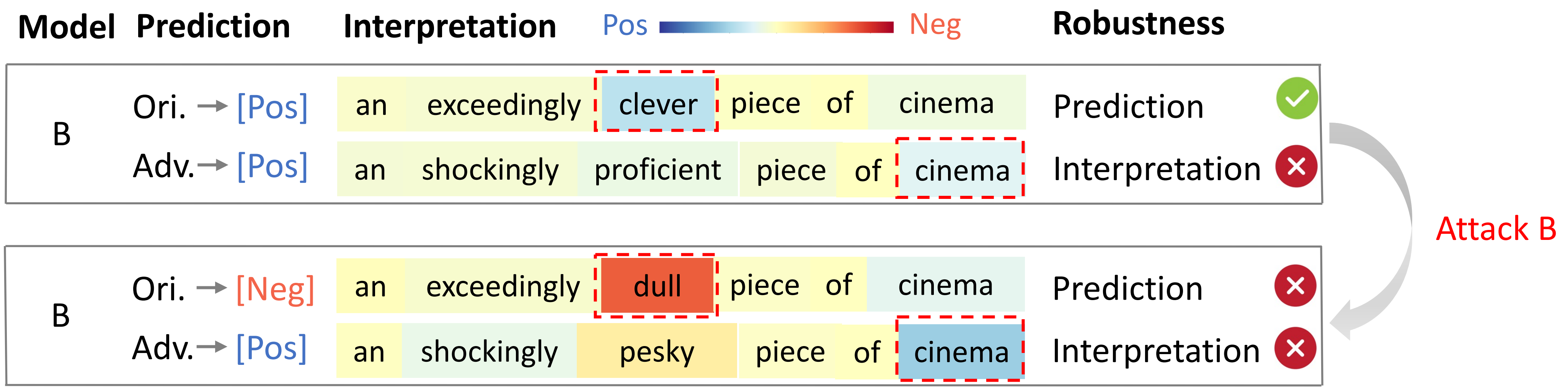
Problem

Traditional adversarial training ignores the consistency between model decision-makings on original/adversarial example pairs

Model	Prediction	Interpretation (Pos — Neg)	Robustness
A	Ori. → [Pos]	an exceedingly clever piece of cinema	Prediction ✗
	Adv. → [Neg]	an shockingly proficient piece of cinema	Interpretation ✗
B	Ori. → [Pos]	an exceedingly clever piece of cinema	Prediction ✓
	Adv. → [Pos]	an shockingly proficient piece of cinema	Interpretation ✗

Problem

Correct predictions cannot guarantee model robustness



Motivation

Robust model

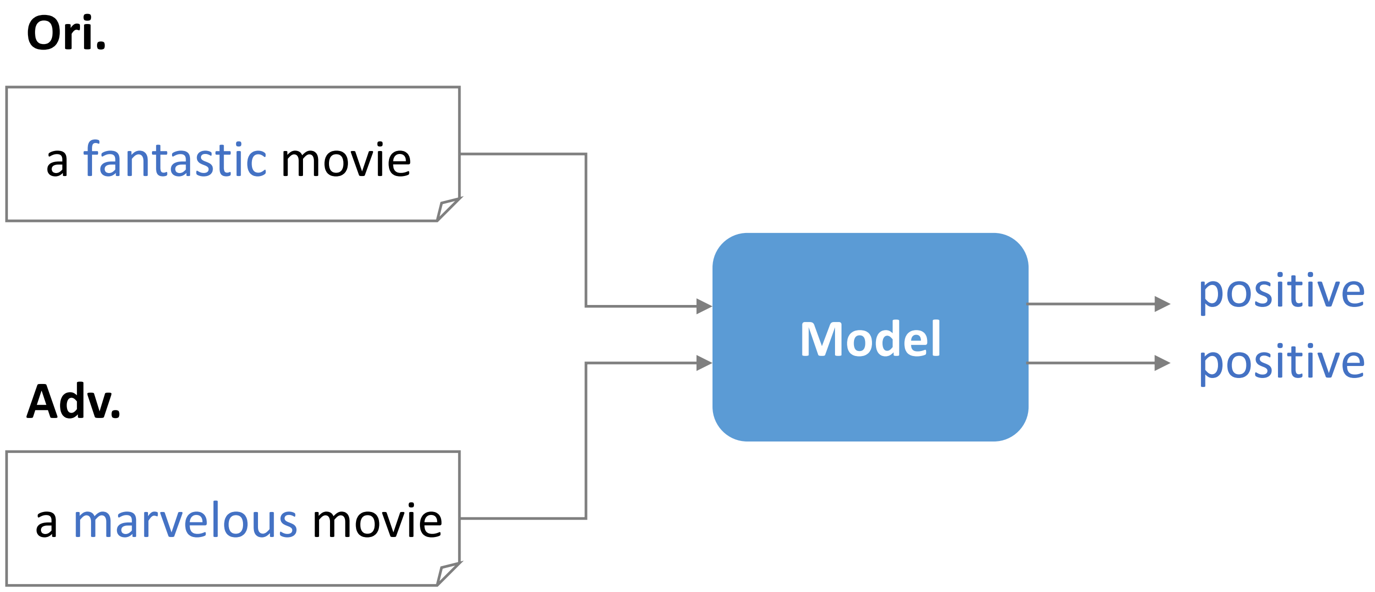
- Consistent prediction behaviors on original/adversarial example pairs
- Making the same predictions (**what**) based on the same reasons (**how**) (consistent interpretations)

Model	**Prediction**	**Interpretation** (Pos — Neg)	**Robustness**
C	Ori. → [Pos]	an exceedingly clever piece of cinema	Prediction
	Adv. → [Pos]	an shockingly proficient piece of cinema	Interpretation
C	Ori. → [Neg]	an exceedingly dull piece of cinema	Prediction
	Adv. → [Neg]	an shockingly pesky piece of cinema	Interpretation

Question?

Feature-level adversarial training (FLAT)

Teach the model to make the same and correct predictions on an original/adversarial example pair based on the corresponding important words



Feature-level adversarial training (FLAT)

Two desiderata for FLAT

- Global feature importance scores $\boldsymbol{\phi}$:
  teach the model to recognize the replaced words in an original example and their substitutions in the adversarial counterpart as the same important (or unimportant) for predictions

$$\begin{array}{ccc} \textbf{Vocab} & & \textbf{Importance} \\ \begin{bmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \\ \vdots \end{bmatrix} & \longrightarrow & \begin{bmatrix} \phi_{\boldsymbol{x}_1} \\ \phi_{\boldsymbol{x}_2} \\ \vdots \end{bmatrix} \end{array} \qquad \phi_{\boldsymbol{x}_i} \in (0, 1)$$

Feature-level adversarial training (FLAT)

Two desiderata for FLAT

- Global feature importance scores $\boldsymbol{\phi}$:
  teach the model to recognize the replaced words in an original example and their substitutions in the adversarial counterpart as the same important (or unimportant) for predictions

$$\underset{\textbf{Vocab}}{\begin{bmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \\ \vdots \end{bmatrix}} \longrightarrow \underset{\textbf{Importance}}{\begin{bmatrix} \phi_{\boldsymbol{x}_1} \\ \phi_{\boldsymbol{x}_2} \\ \vdots \end{bmatrix}} \qquad \phi_{\boldsymbol{x}_i} \in (0, 1)$$

- Feature selection function $g_{\boldsymbol{\phi}}(\cdot)$
  guide the model to make predictions based on the corresponding important words in the original and adversarial example respectively

$$\boldsymbol{x} \longrightarrow g_{\boldsymbol{\phi}}(\boldsymbol{x})$$

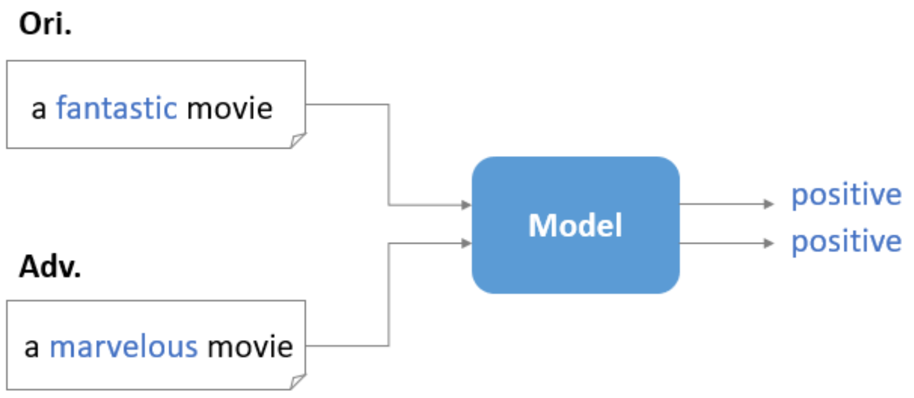
Feature-level adversarial training (FLAT)

Objective

$$\min_{\boldsymbol{\phi},\boldsymbol{\theta}} \mathcal{L}_{pred} + \gamma \mathcal{L}_{imp}$$

$$\mathcal{L}_{pred} = \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}\left[\mathcal{L}\left(f_{\boldsymbol{\theta}}\left(g_{\boldsymbol{\phi}}(\boldsymbol{x})\right),y\right)\right] + \mathbb{E}_{(\boldsymbol{x}',y)\sim\mathcal{D}'}\left[\mathcal{L}\left(f_{\boldsymbol{\theta}}\left(g_{\boldsymbol{\phi}}(\boldsymbol{x}')\right),y\right)\right]$$

$$\mathcal{L}_{imp} = \mathbb{E}_{(\boldsymbol{x},\boldsymbol{x}')\sim\mathcal{D}\cup\mathcal{D}'}\left[\sum_{i,\,\boldsymbol{x}_i\neq\boldsymbol{x}_i'} \left|\phi_{\boldsymbol{x}_i} - \phi_{\boldsymbol{x}_i'}\right|\right]$$



$\mathcal{L}(\cdot,\cdot)$: cross entropy loss

$\mathcal{L}(\cdot,\cdot)$: cross entropy loss

$\boldsymbol{x}$: original example

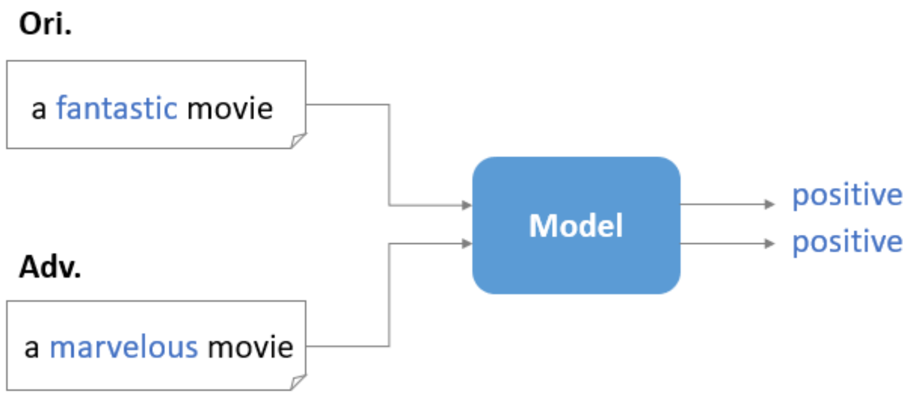
$\boldsymbol{x}'$: adversarial example

Feature-level adversarial training (FLAT)

Objective

$$\min_{\boldsymbol{\phi},\boldsymbol{\theta}} \mathcal{L}_{pred} + \gamma \mathcal{L}_{imp}$$

$$\mathcal{L}_{pred} = \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}\left[\mathcal{L}\left(f_{\boldsymbol{\theta}}\left(g_{\boldsymbol{\phi}}(\boldsymbol{x})\right),y\right)\right] + \mathbb{E}_{(\boldsymbol{x}',y)\sim\mathcal{D}'}\left[\mathcal{L}\left(f_{\boldsymbol{\theta}}\left(g_{\boldsymbol{\phi}}(\boldsymbol{x}')\right),y\right)\right]$$



$$\mathcal{L}_{imp} = \mathbb{E}_{(\boldsymbol{x},\boldsymbol{x}')\sim\mathcal{D}\cup\mathcal{D}'}\left[\sum_{i,\, \boldsymbol{x}_i \neq \boldsymbol{x}_i'} \left|\phi_{\boldsymbol{x}_i} - \phi_{\boldsymbol{x}_i'}\right|\right]$$

How to learn $\boldsymbol{\phi}$?

How to select words via $g_{\boldsymbol{\phi}}(\cdot)$?

$\mathcal{L}(\cdot,\cdot)$: cross entropy loss

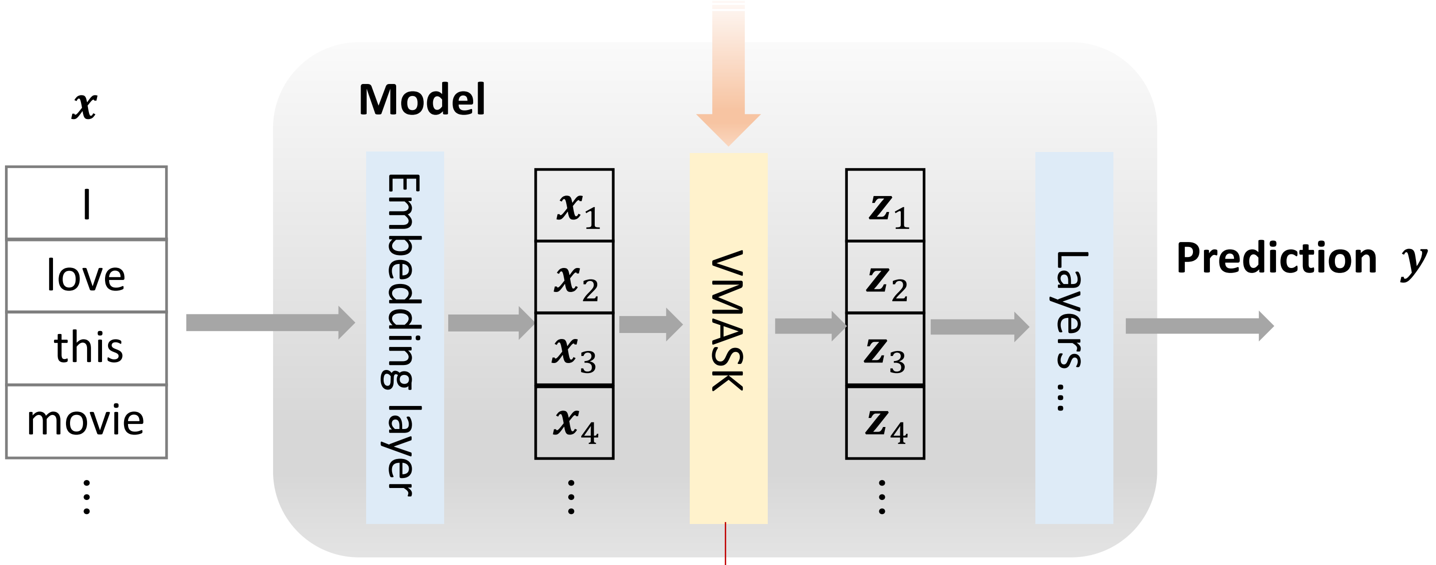
$\mathcal{L}(\cdot,\cdot)$: cross entropy loss

$\boldsymbol{x}$: original example

$\boldsymbol{x}'$: adversarial example

Feature-level adversarial training (FLAT)

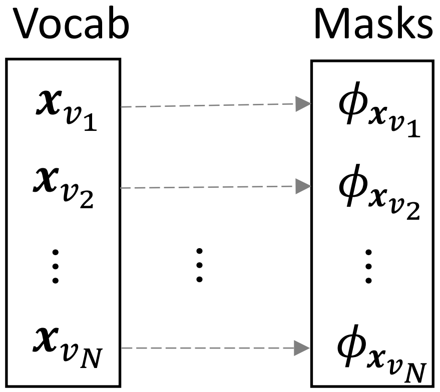
Learning with variational word masks (VMASK)



$\boldsymbol{z} = g_{\boldsymbol{\phi}}(\boldsymbol{x}\ ) = \boldsymbol{W} \odot \boldsymbol{x} \quad W_{x_i} \in \{0, 1\}$

- Mask out irrelevant or noisy words
- Forward important words to the model

Training stage



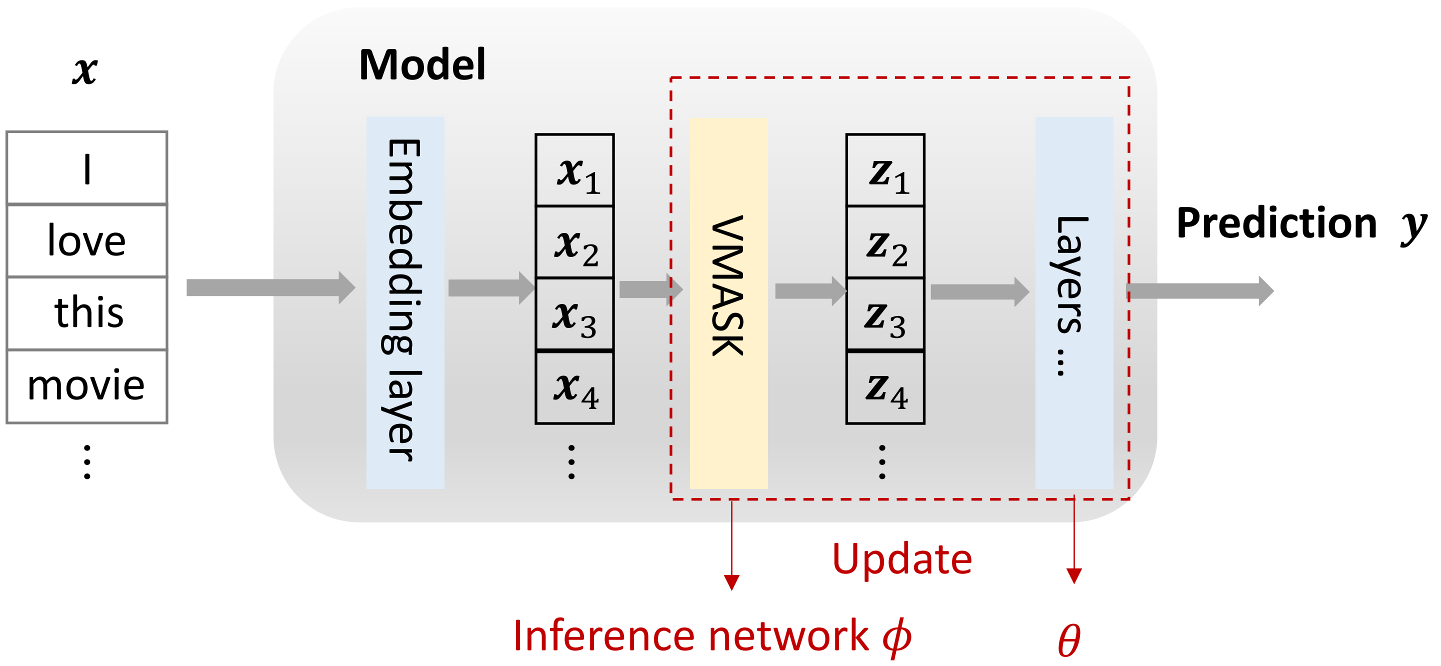
$\phi_{\boldsymbol{x}_i}$: global word importance

- $W_{\boldsymbol{x}_i} \sim Bernoulli(\phi_{\boldsymbol{x}_i})$
- $\boldsymbol{z}_i = W_{\boldsymbol{x}_i} \cdot \boldsymbol{x}_i, W_{\boldsymbol{x}_i} \in \{0, 1\}$

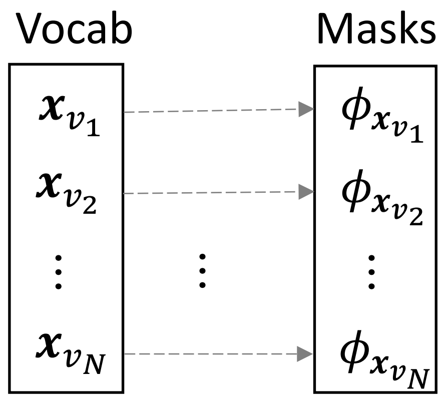
Hanjie Chen and Yangfeng Ji. "Learning variational word masks to improve the interpretability of neural text classifiers." *EMNLP*, 2020

Feature-level adversarial training (FLAT)

Learning with variational word masks (VMASK)



Training stage



Information bottleneck

$$\max_{\boldsymbol{Z}} I(\boldsymbol{Z};\boldsymbol{Y}) - \beta \cdot I(\boldsymbol{Z};\boldsymbol{X})$$

lower bound

$$\max_{\theta,\phi} \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}\left[\mathbb{E}_q[\log p(y|\boldsymbol{W},\boldsymbol{x})] + \beta \cdot H_q(\boldsymbol{W}|\boldsymbol{x})\right]$$

Hanjie Chen and Yangfeng Ji. "Learning variational word masks to improve the interpretability of neural text classifiers." *EMNLP*, 2020

Feature-level adversarial training (FLAT)

Objective

$$\min_{\boldsymbol{\phi},\boldsymbol{\theta}} \mathcal{L}_{pred} + \gamma \mathcal{L}_{imp}$$

$$\mathcal{L}_{pred} = \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}\left[\mathbb{E}_q[\mathcal{L}(f_{\boldsymbol{\theta}}(\boldsymbol{W}\odot\boldsymbol{x}), y)] - \beta \cdot H_q(\boldsymbol{W}|\boldsymbol{x})\right] + \mathbb{E}_{(\boldsymbol{x}',y)\sim\mathcal{D}'}\left[\mathbb{E}_{q'}[\mathcal{L}(f_{\boldsymbol{\theta}}(\boldsymbol{W}'\odot\boldsymbol{x}'), y)] - \beta \cdot H_q(\boldsymbol{W}'|\boldsymbol{x}')\right]$$

$$\mathcal{L}_{imp} = \mathbb{E}_{(\boldsymbol{x},\boldsymbol{x}')\sim\mathcal{D}\cup\mathcal{D}'}\left[\sum_{i,\, \boldsymbol{x}_i \neq \boldsymbol{x}_i'} \left|\phi_{\boldsymbol{x}_i} - \phi_{\boldsymbol{x}_i'}\right|\right]$$

$\mathcal{L}(\cdot,\cdot)$: cross entropy loss, $H_q(\cdot \mid \cdot)$: conditional entropy

$q = q_{\boldsymbol{\phi}}(\boldsymbol{W}|\boldsymbol{x})$ and $q' = q'_{\boldsymbol{\phi}}(\boldsymbol{W}'|\boldsymbol{x}')$ denote the distributions of word masks on the original example $\boldsymbol{x}$ and adversarial example $\boldsymbol{x}'$ respectively

Feature-level adversarial training (FLAT)

Objective

$$\min_{\boldsymbol{\phi},\boldsymbol{\theta}} \mathcal{L}_{pred} + \gamma \mathcal{L}_{imp}$$

$$\mathcal{L}_{pred} = \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}\left[\mathbb{E}_q[\mathcal{L}(f_{\boldsymbol{\theta}}(\boldsymbol{W}\odot\boldsymbol{x}),y)] - \beta \cdot H_q(\boldsymbol{W}|\boldsymbol{x})\right] + \mathbb{E}_{(\boldsymbol{x}',y)\sim\mathcal{D}'}\left[\mathbb{E}_{q'}[\mathcal{L}(f_{\boldsymbol{\theta}}(\boldsymbol{W}'\odot\boldsymbol{x}'),y)] - \beta \cdot H_q(\boldsymbol{W}'|\boldsymbol{x}')\right]$$

$$\mathcal{L}_{imp} = \mathbb{E}_{(\boldsymbol{x},\boldsymbol{x}')\sim\mathcal{D}\cup\mathcal{D}'}\left[\sum_{i,\,\boldsymbol{x}_i\neq\boldsymbol{x}_i'} \left|\phi_{\boldsymbol{x}_i} - \phi_{\boldsymbol{x}_i'}\right|\right]$$

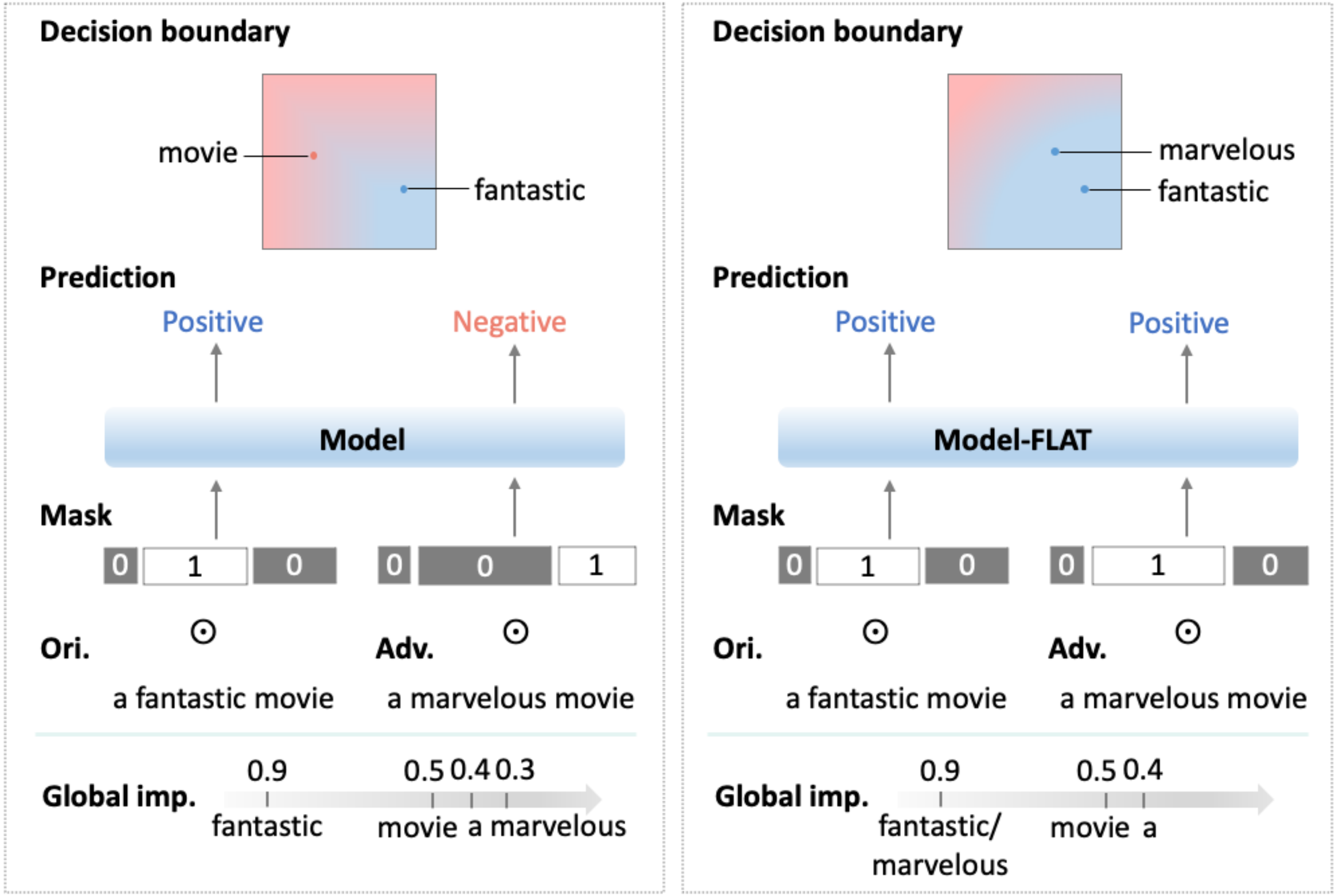
Connection

FLAT degrades to traditional adversarial training when $\boldsymbol{W} = \mathbf{1}$ and $\gamma = 0$

$\mathcal{L}(\cdot,\cdot)$: cross entropy loss, $H_q(\cdot\mid\cdot)$: conditional entropy

$q = q_{\boldsymbol{\phi}}(\boldsymbol{W}|\boldsymbol{x})$ and $q' = q'_{\boldsymbol{\phi}}(\boldsymbol{W}'|\boldsymbol{x}')$ denote the distributions of word masks on the original example $\boldsymbol{x}$ and adversarial example $\boldsymbol{x}'$ respectively

Feature-level adversarial training (FLAT)



(a) (b)

Experimental Setup

Models

- Recurrent neural network [Hochreiter and Schmidhuber 1997, LSTM]
- Convolutional neural network [Kim 2014, CNN]
- BERT [Devlin et al., 2019]
- DeBERTa [He et al., 2021]

Datasets

- IMDB [Maas et al., 2011]
- SST-2 [Socher et al., 2013]
- AG News (AG) [Zhang et al.,2015]
- TREC [Li and Roth, 2002]

Attacks

(TextAttack benchmark [Morris et al. 2020])

- Textfooler [Jin et al. 2020]
- PWWS [Ren et al. 2019]

Experiments

Prediction accuracy (%) on standard test sets

Models	SST2	IMDB	AG	TREC
LSTM-base	84.40	88.03	91.08	90.80
LSTM-adv(Textfooler)	82.32	88.79	90.29	87.60
LSTM-adv(PWWS)	82.59	88.37	91.16	89.60
LSTM-FLAT (Textfooler)	**84.79**	**89.17**	91.00	91.00
LSTM-FLAT (PWWS)	83.69	88.52	**91.37**	**91.20**
CNN-base	**84.18**	88.63	91.32	**91.20**
CNN-adv(Textfooler)	82.15	88.81	90.99	89.20
CNN-adv(PWWS)	83.42	88.89	91.30	90.00
CNN-FLAT (Textfooler)	83.09	88.89	**91.64**	89.20
CNN-FLAT (PWWS)	83.31	**88.99**	91.03	89.20
BERT-base	91.32	91.71	93.59	**97.40**
BERT-adv(Textfooler)	91.38	92.50	90.30	96.00
BERT-adv(PWWS)	90.88	**93.14**	93.38	95.20
BERT-FLAT (Textfooler)	**91.54**	92.78	**94.07**	96.20
BERT-FLAT (PWWS)	91.05	93.11	93.09	96.60
DeBERTa-base	94.18	93.80	93.62	96.40
DeBERTa-adv(Textfooler)	94.40	92.86	92.84	95.60
DeBERTa-adv(PWWS)	**94.78**	94.17	92.96	96.40
DeBERTa-FLAT (Textfooler)	94.29	**94.29**	**94.29**	96.40
DeBERTa-FLAT (PWWS)	94.12	94.26	93.82	96.40

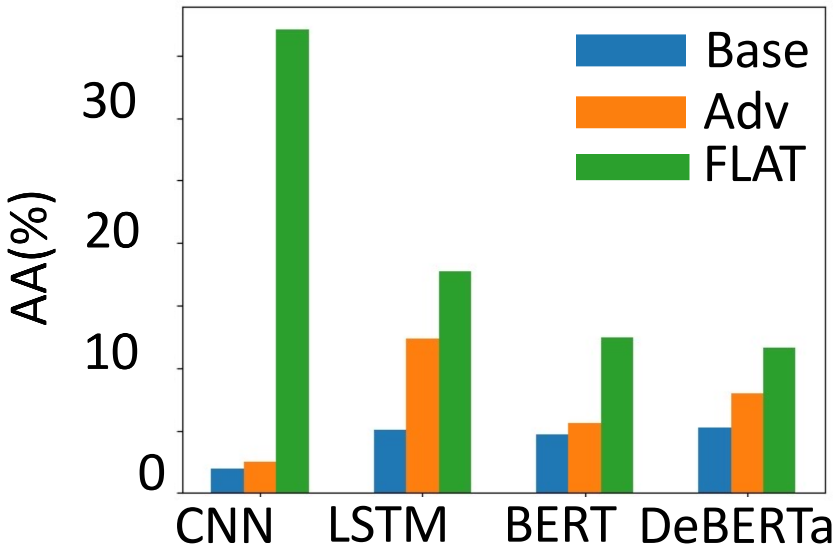
"-base": the base model trained on the clean data
"-adv": the model trained via traditional adversarial training
"-FLAT": the model trained via FLAT

✓ Adversarial training ("adv" and "FLAT") does not hurt model performance on clean data, and even improves prediction accuracy in some cases

Experiments

Prediction robustness

After-attack accuracy (AA): model prediction accuracy on adversarial examples



(Textfooler, SST2)

- ✓ Adversarial training improves model prediction robustness
- ✓ FLAT consistently outperforms traditional adversarial training

Experiments

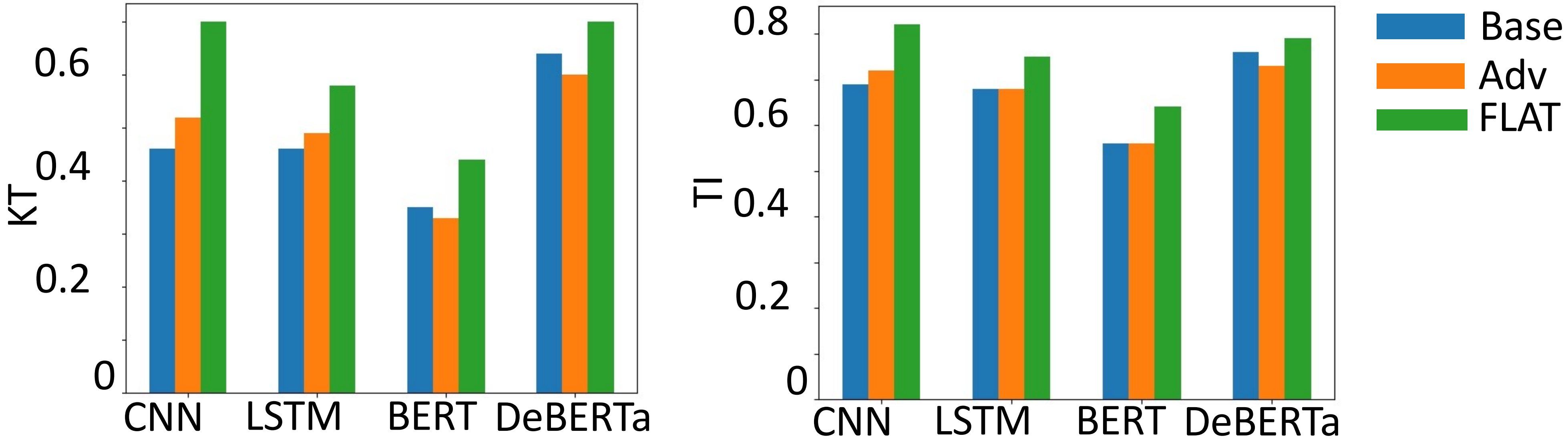
Interpretation Consistency

Ori. → [Pos] an exceedingly clever piece of cinema

Adv. → [Pos] an shockingly proficient piece of cinema

- Post-hoc interpretations: IG, LIME
- Kendall's Tau order rank correlation (KT): overall rankings of word attributions between different interpretations
- Top-k intersection (TI): the proportion of intersection of top k important features identified by different interpretations

[Chen et al. 2019; Ghorbani et al. 2019, Boopathy et al. 2020]



(Textfooler, SST2)

Experiments

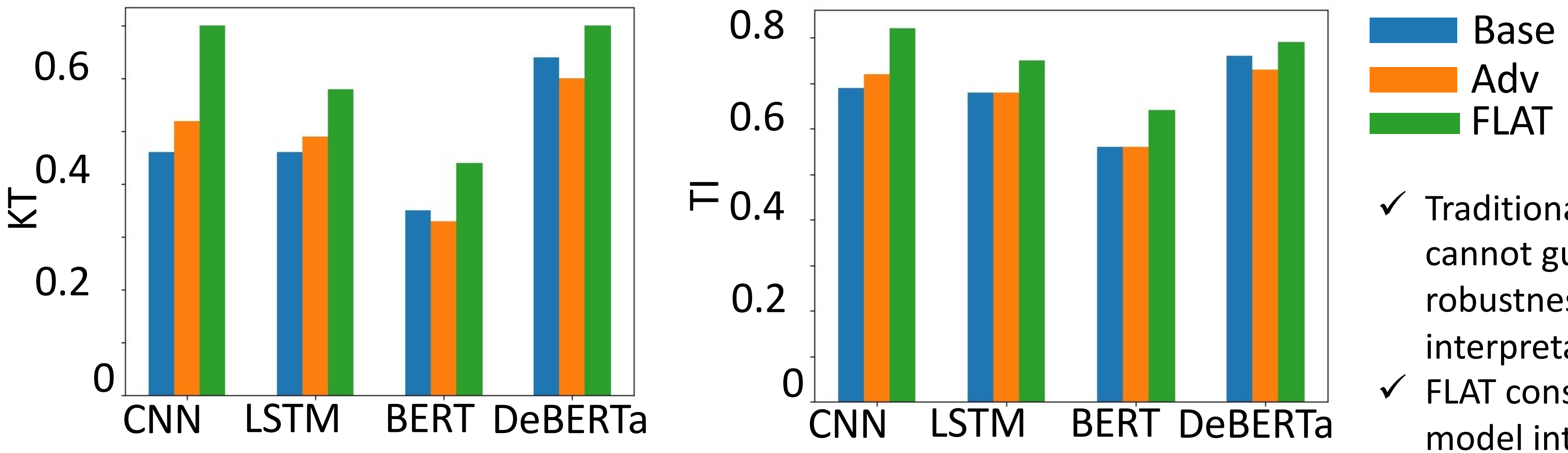
Interpretation Consistency

Ori. → [Pos] an exceedingly clever piece of cinema

Adv. → [Pos] an shockingly proficient piece of cinema

- Post-hoc interpretations: IG, LIME
- Kendall's Tau order rank correlation (KT): overall rankings of word attributions between different interpretations
- Top-k intersection (TI): the proportion of intersection of top k important features identified by different interpretations

[Chen et al. 2019; Ghorbani et al. 2019, Boopathy et al. 2020]



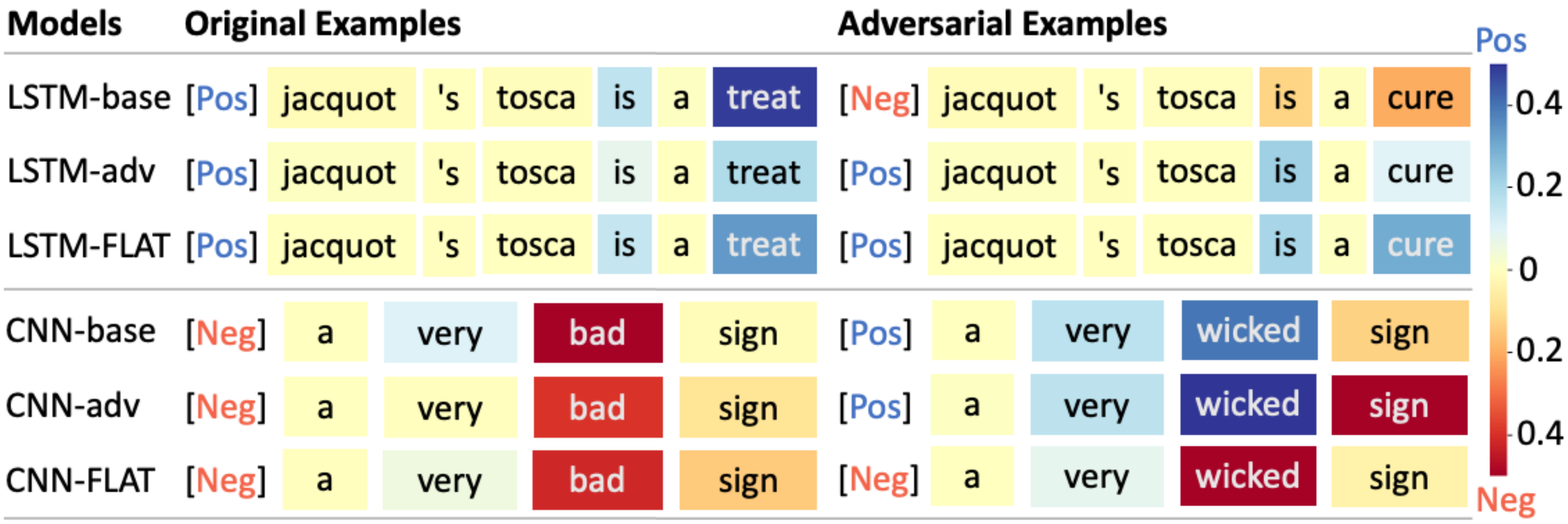
(Textfooler, SST2)

- ✓ Traditional adversarial training cannot guarantee model robustness regarding interpretation discrepancy
- ✓ FLAT consistently improves model interpretation consistency

Experiments

Visualization of interpretations

Both LSTM-FLAT and CNN-FLAT correctly predict the original/adversarial example pairs with consistent interpretations



Experiments

Transferability of model robustness

Test with six unforeseen adversarial attacks: PWWS [Ren et al. 2019], Gene [Alzantot et al. 2018], IGA [Wang et al. 2019], PSO [Zang et al. 2020], Clare [Li et al. 2021], and BAE [Garg and Ramakrishnan 2020]

Models	PWWS	Gene	IGA	PSO	Clare	BAE
LSTM-base	11.64	20.26	9.83	5.88	3.02	36.52
LSTM-adv	15.38	25.65	17.02	5.60	3.90	36.35
LSTM-FLAT	**20.48**	**33.44**	**24.22**	**6.53**	**5.55**	**39.87**
CNN-base	8.29	20.32	7.85	5.60	1.48	37.12
CNN-adv	8.68	16.42	6.26	5.60	1.04	35.48
CNN-FLAT	**42.56**	**55.02**	**46.35**	**10.38**	**17.57**	**48.38**
BERT-base	11.70	32.24	9.72	6.26	0.86	35.31
BERT-adv	13.01	34.49	10.87	6.64	1.04	36.74
BERT-FLAT	**15.93**	**35.31**	**15.93**	**9.50**	**5.29**	**37.56**
DeBERTa-base	14.17	37.12	12.19	6.75	0.55	38.61
DeBERTa-adv	17.52	37.18	12.85	7.96	1.07	40.14
DeBERTa-FLAT	**21.80**	**48.16**	**28.17**	**13.01**	**1.37**	**44.54**

- ✓ The models trained via FLAT show better robustness than baseline models across different attacks

Question?

Reference

- Du, Mengnan, et al. "Towards interpreting and mitigating shortcut learning behavior of NLU models." *arXiv preprint arXiv:2103.06922* (2021).
- Chen, Hanjie, and Yangfeng Ji. "Adversarial Training for Improving Model Robustness? Look at Both Prediction and Interpretation." *arXiv preprint arXiv:2203.12709* (2022).
- Geirhos, Robert, et al. "Shortcut learning in deep neural networks." *Nature Machine Intelligence* 2.11 (2020): 665-673.
- Tal Schuster, Darsh J Shah, Yun Jie Serene Yeo, Daniel Filizzola, Enrico Santus, and Regina Barzilay. 2019. Towards debiasing fact verification models. Empirical Methods in Natural Language Processing (EMNLP).