Paper Seminar

FreeLB: Enhanced Adversarial Training for Natural Language Understanding

Zhu et al., 2020, ICLR

Myeongsup Kim

Integrated M.S./Ph.D. Student
Data Science & Business Analytics Lab.
School of Industrial Management Engineering
Korea University

Myeongsup_kim@korea.ac.kr

Introduction

- Transformer-Based Language Model

<What This Seminar Does Not Cover>

- **Details of Transformer**

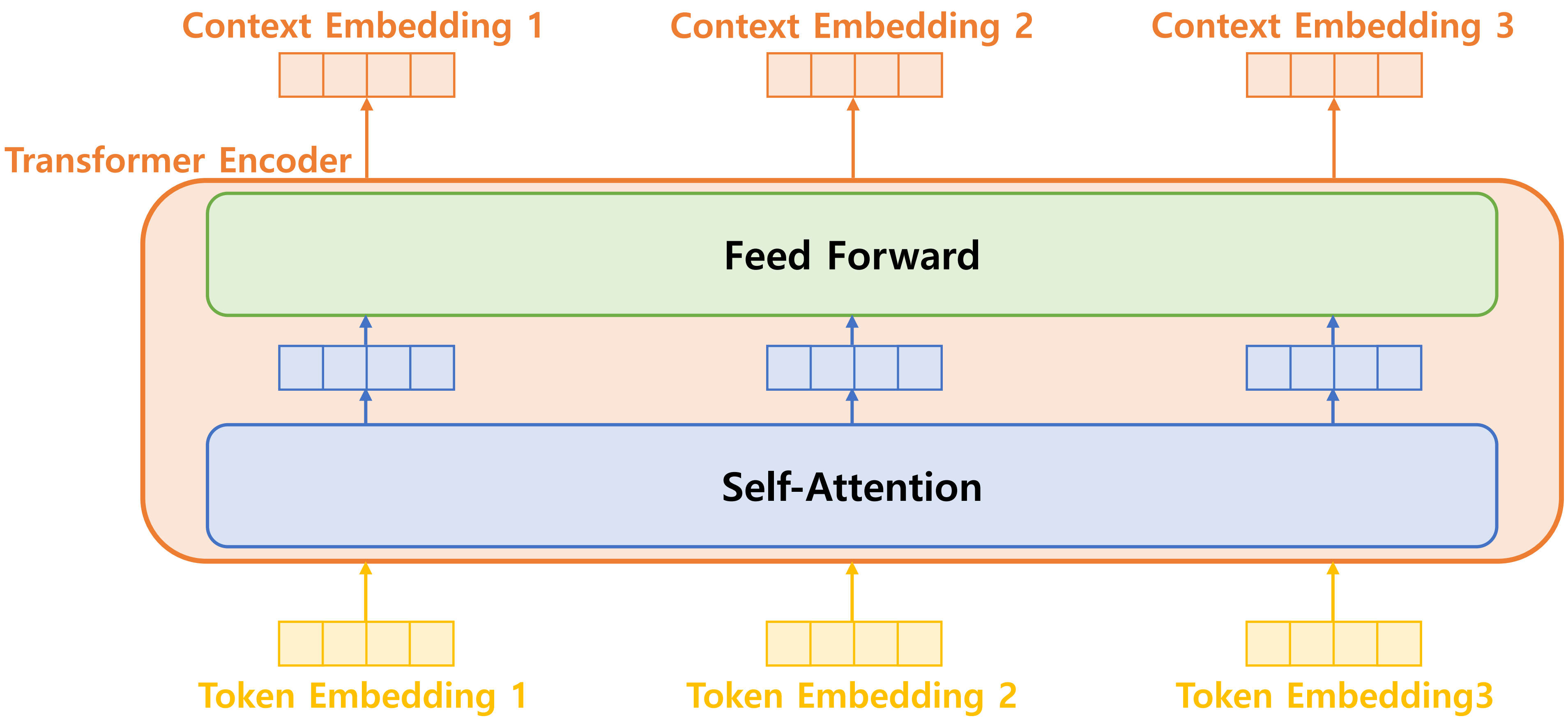
  Vaswani et al., Attention is All You Need, NIPS, 2017

- **Details of BERT and RoBERTa**

  Devlin et al., BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, NAACL, 2019

  Liu et al., RoBERTa: A Robustly Optimized BERT Pretraining Approach, arXiv, 2019

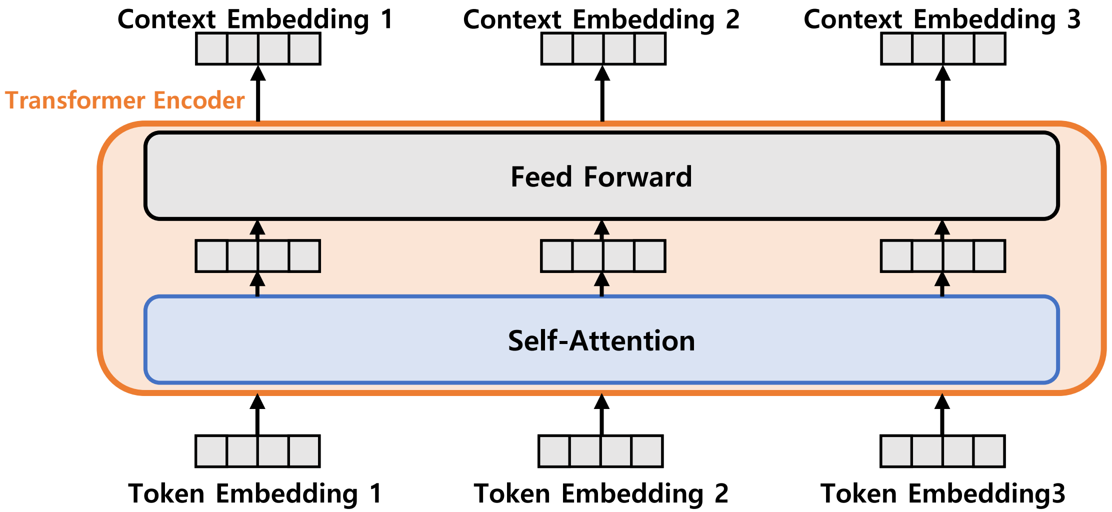
<Transformer Encoder>

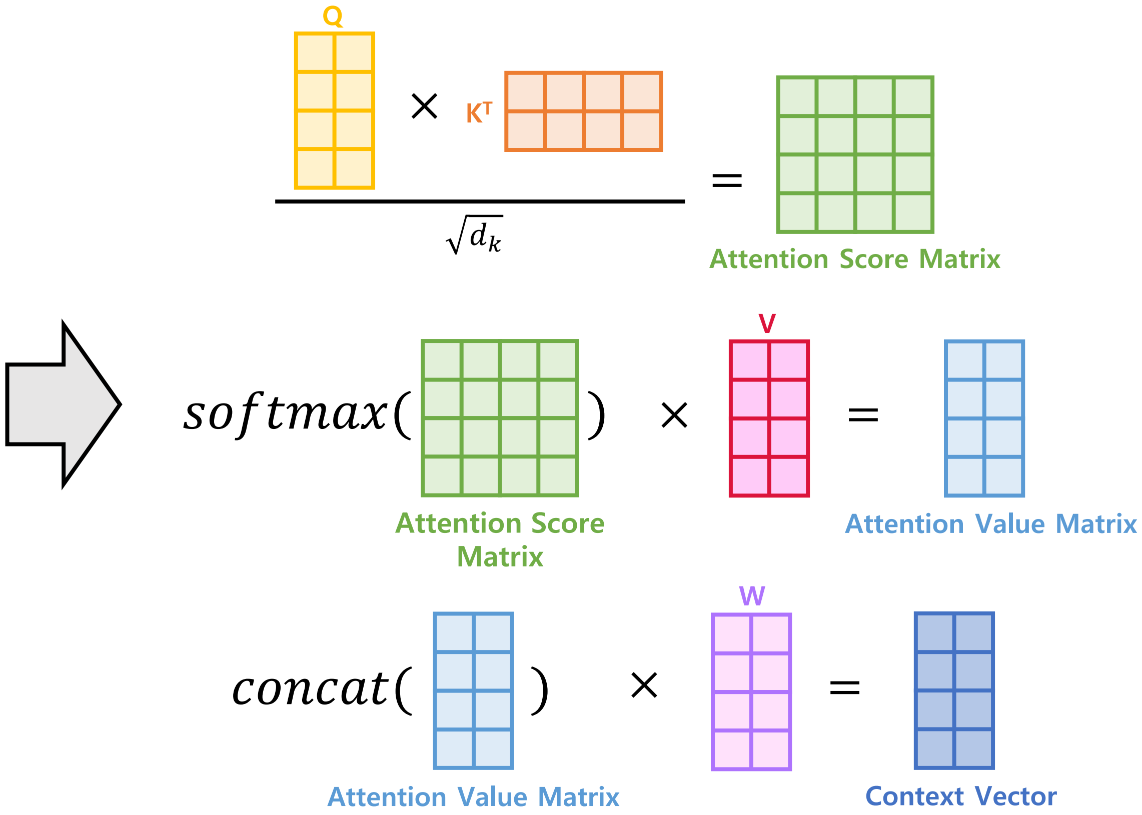


Introduction

-Transformer-Based Language Model

<Self-Attention>

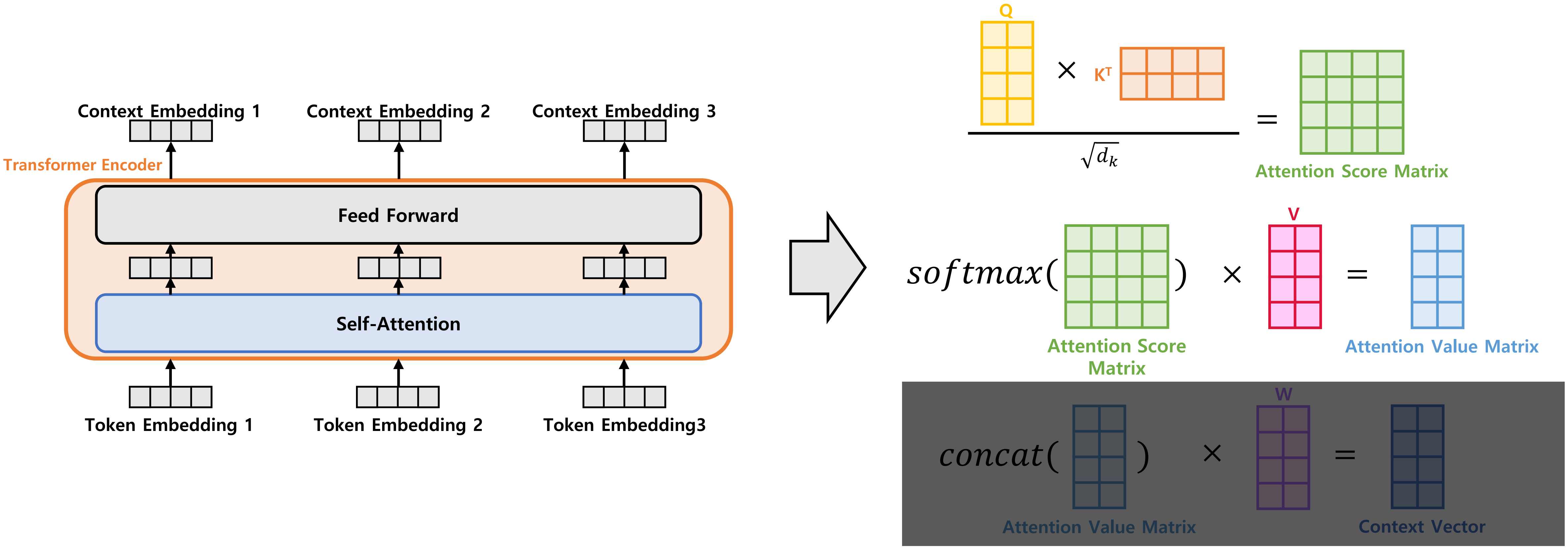




Introduction

-Transformer-Based Language Model

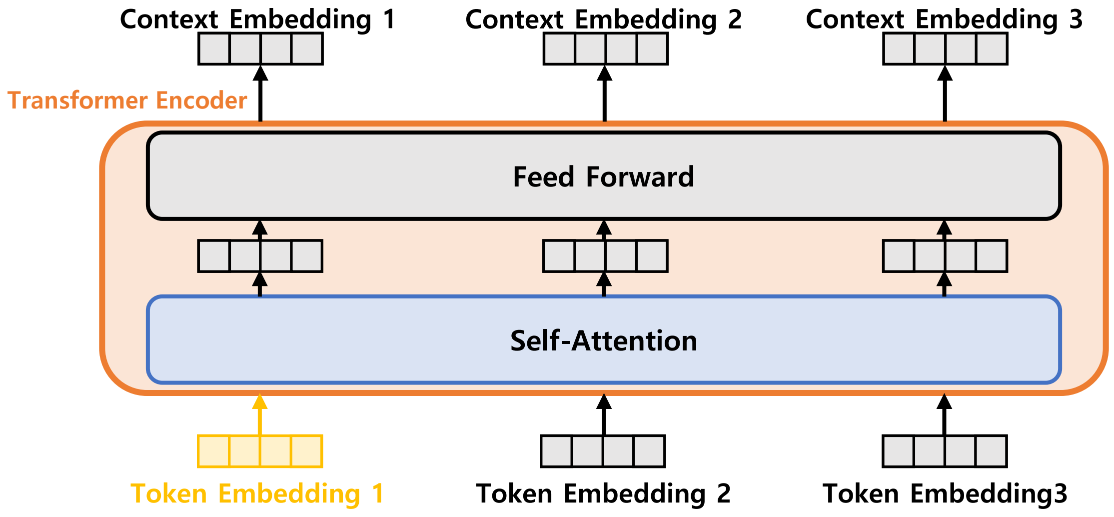
<Self-Attention>

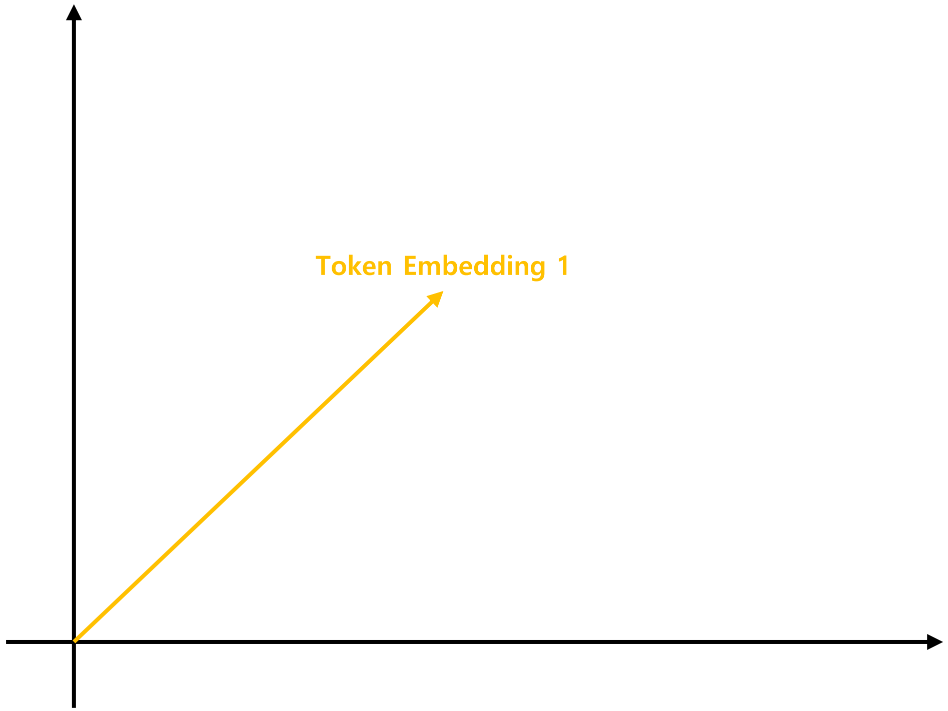


Introduction

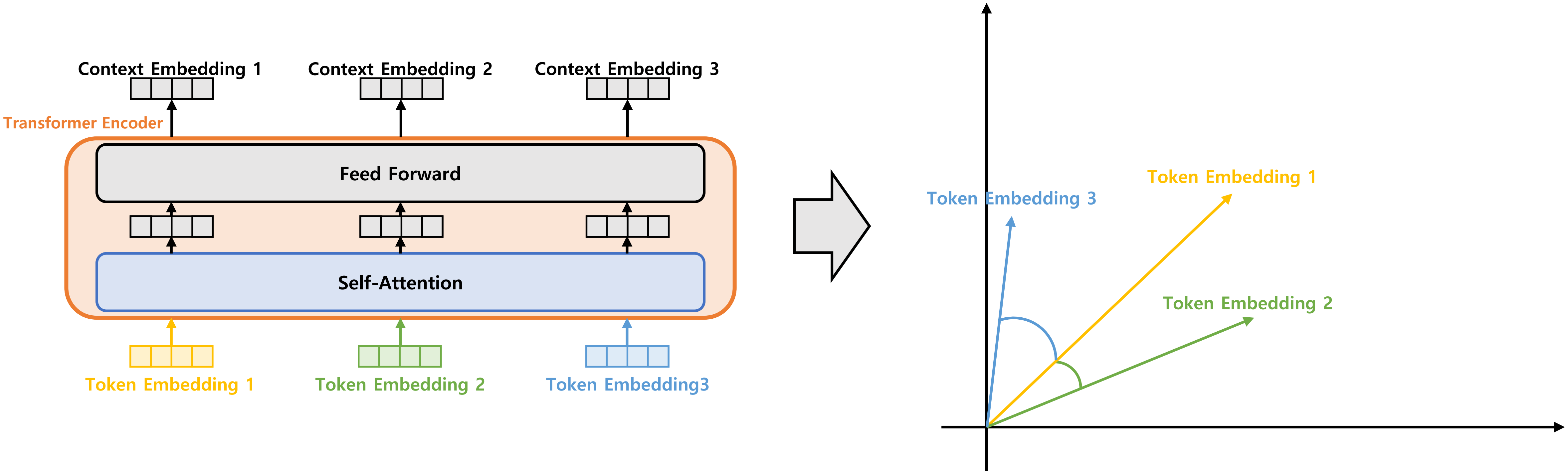
-Transformer-Based Language Model

<Self-Attention>





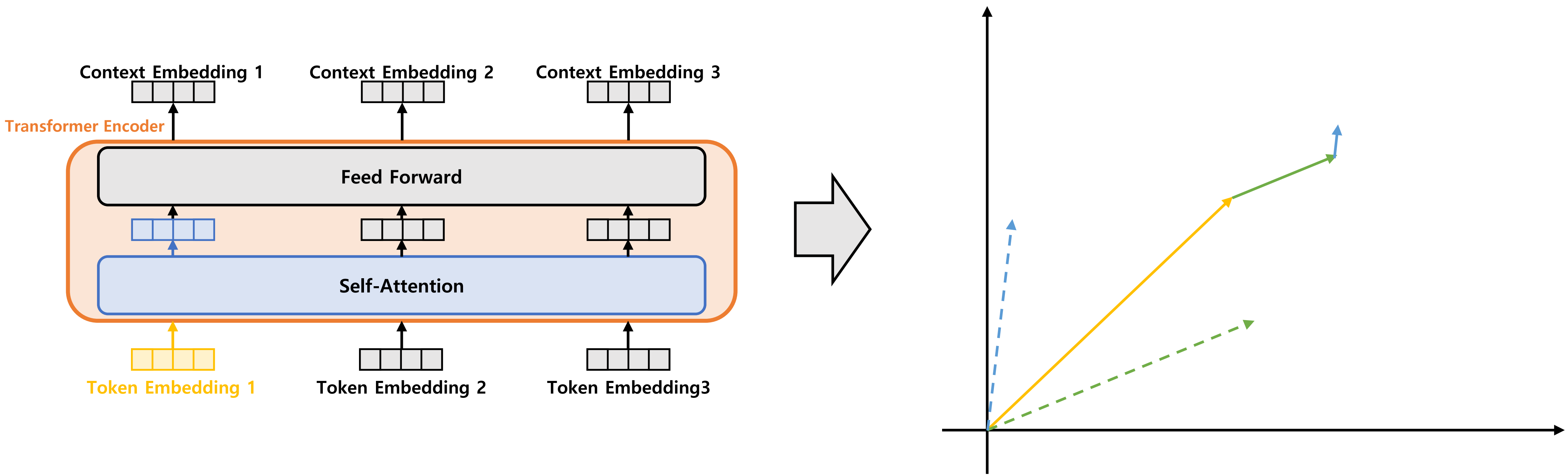
<Self-Attention>



Introduction

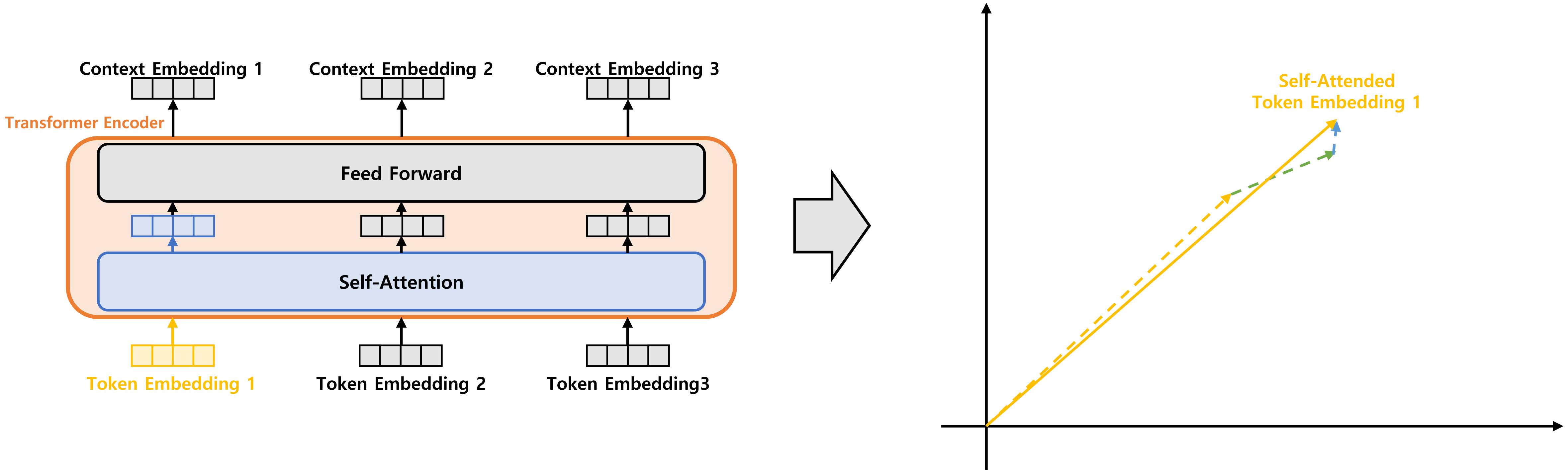
-Transformer-Based Language Model

<Self-Attention>

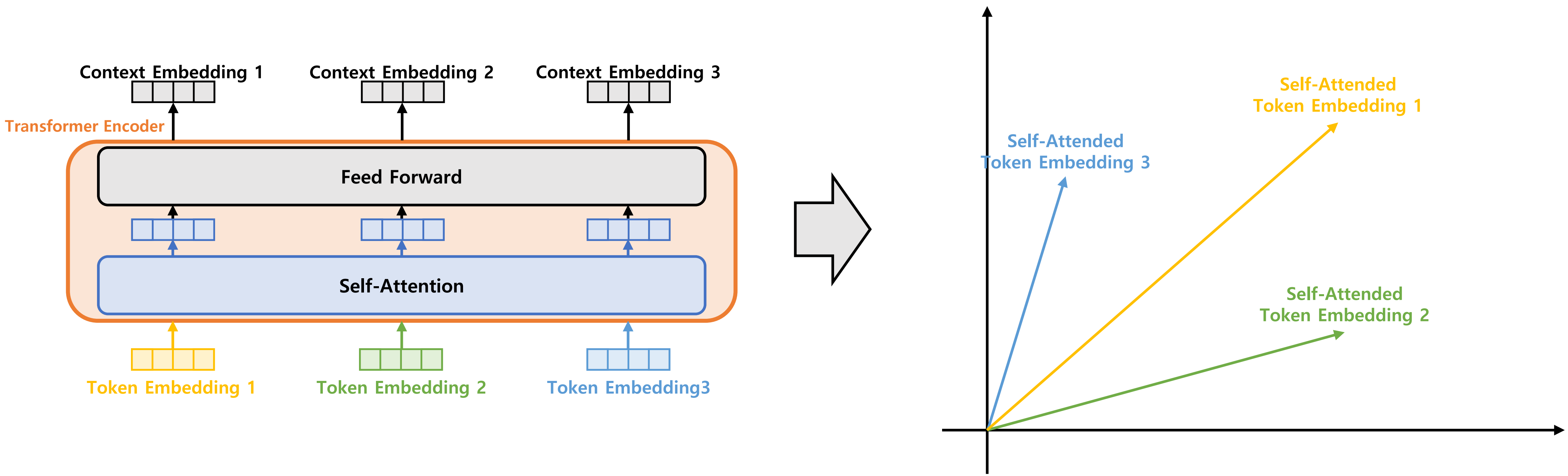


Introduction
-Transformer-Based Language Model

<Self-Attention>



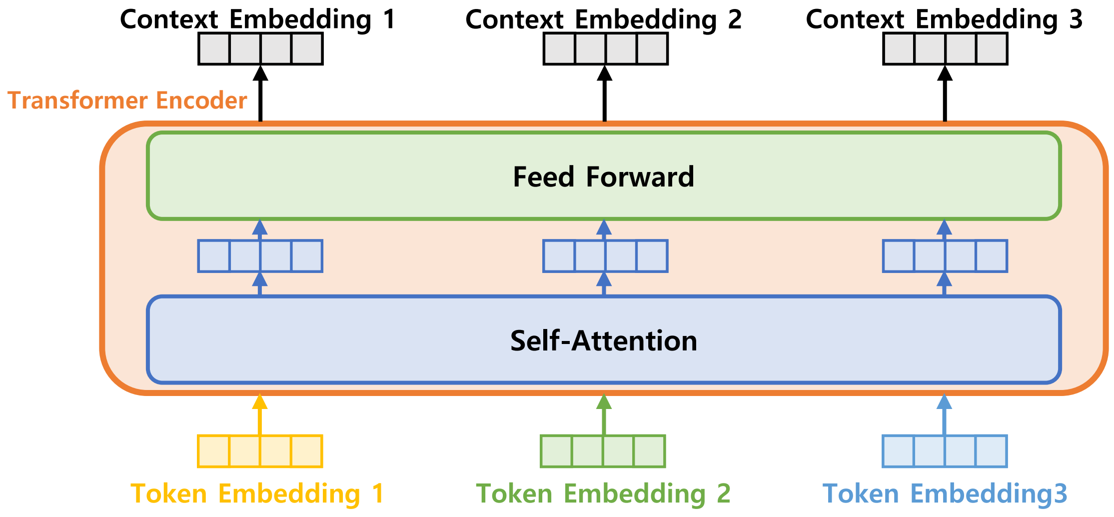
<Self-Attention>

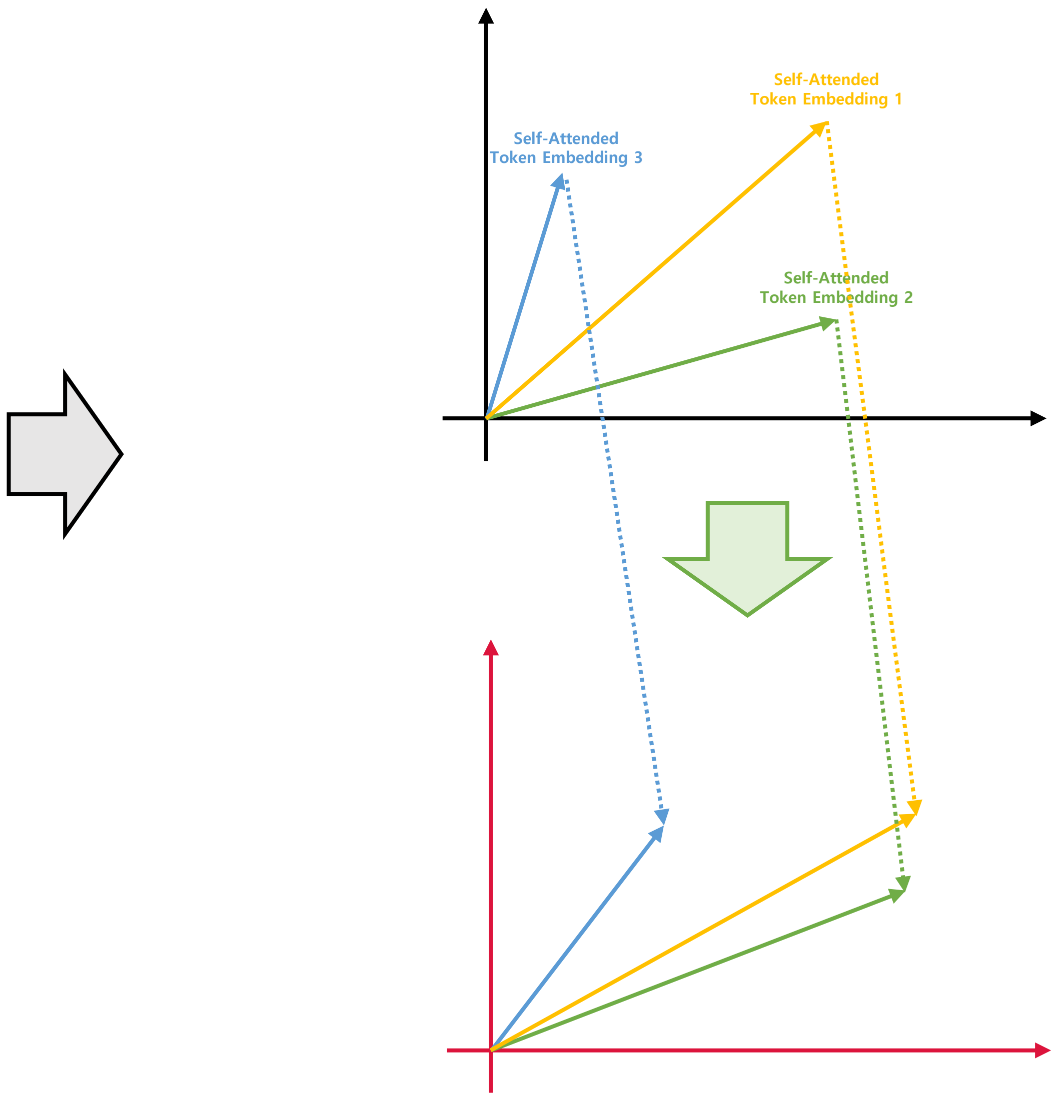


Introduction

-Transformer-Based Language Model

<Feed Forward>

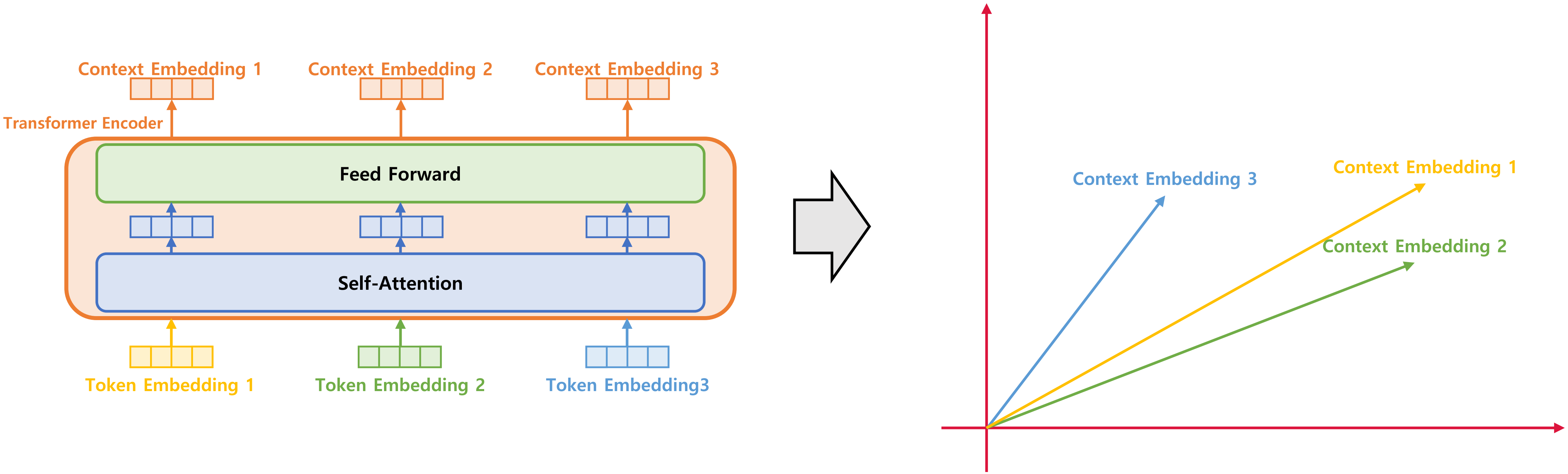




Introduction

-Transformer-Based Language Model

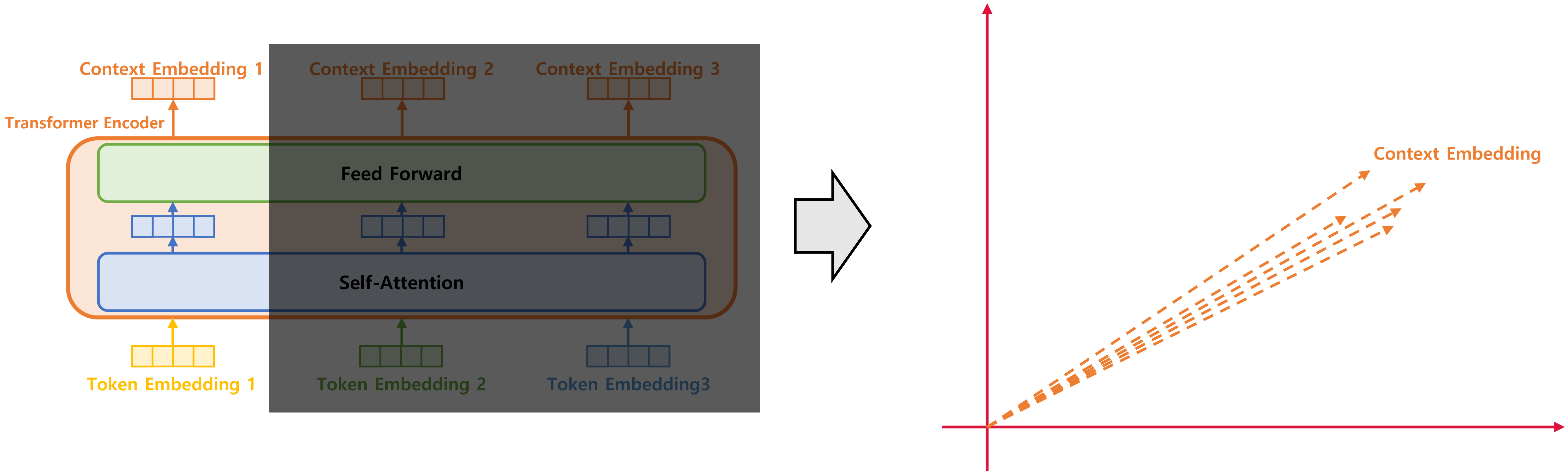
<Feed Forward>



Introduction

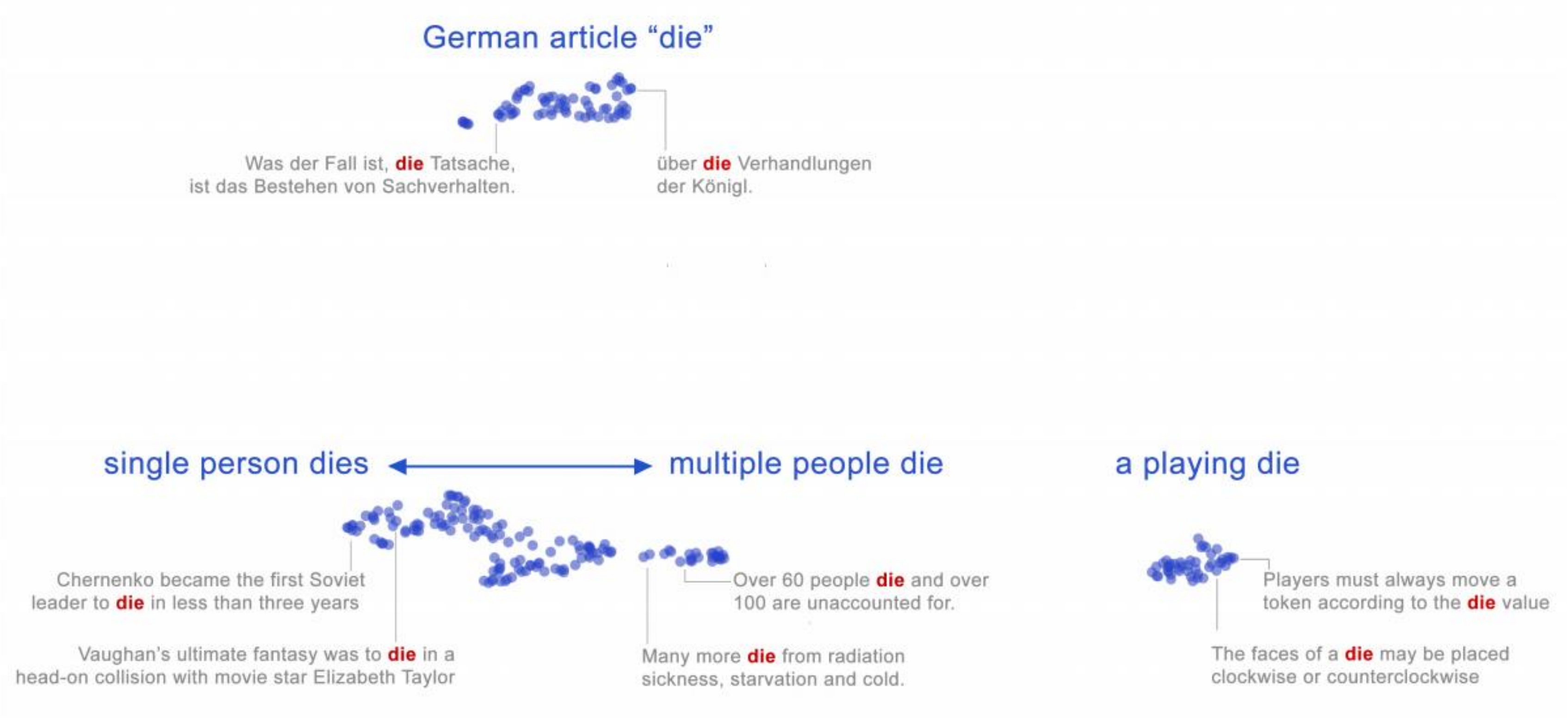
-Transformer-Based Language Model

<Contextualized Representation>



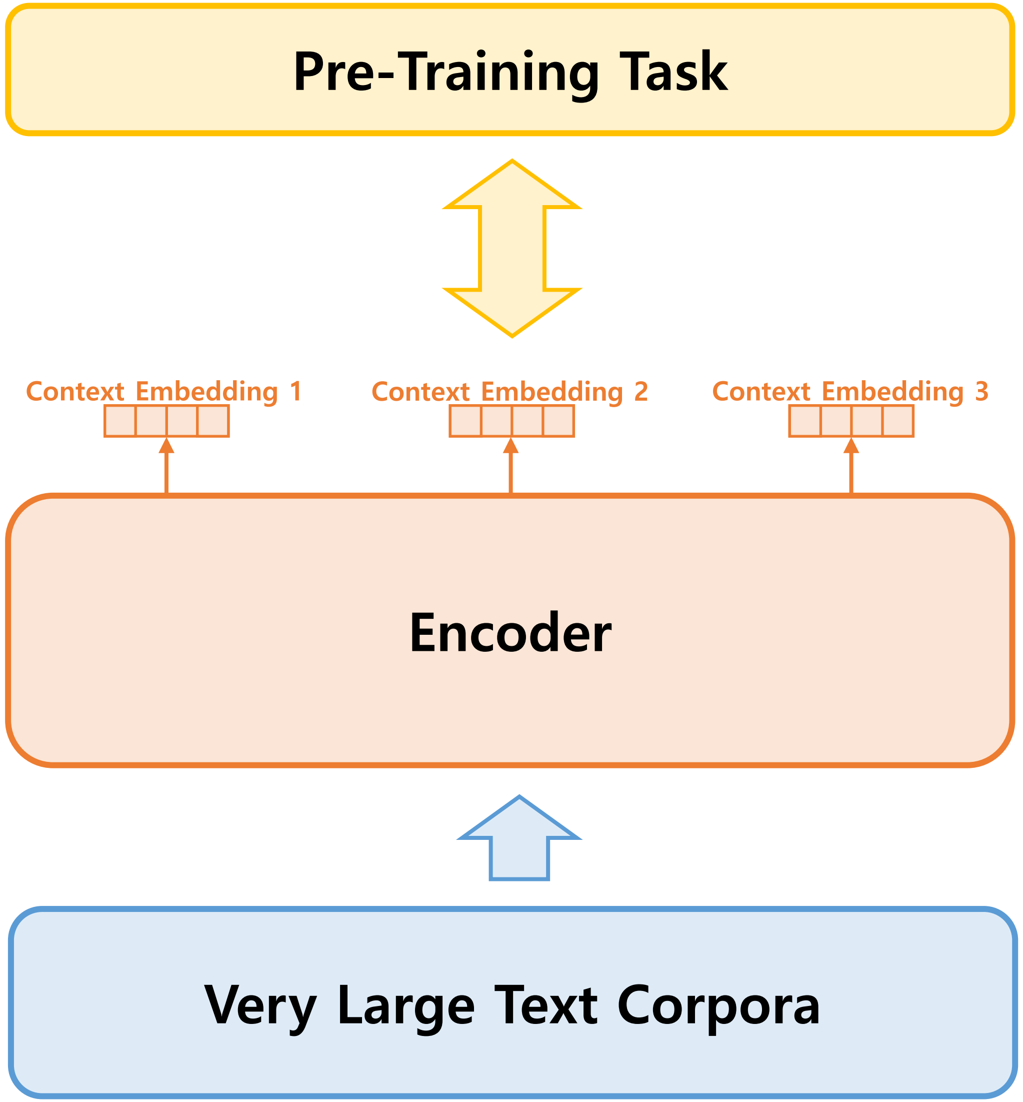
Coenen et al., Visualizing and Measuring the Geometry of BERT, NeurIPS, 2019
[Paper Review] Syntax and Semantics in Language Model Representation (Myeongsup Kim, 2020)

<Contextualized Representation>

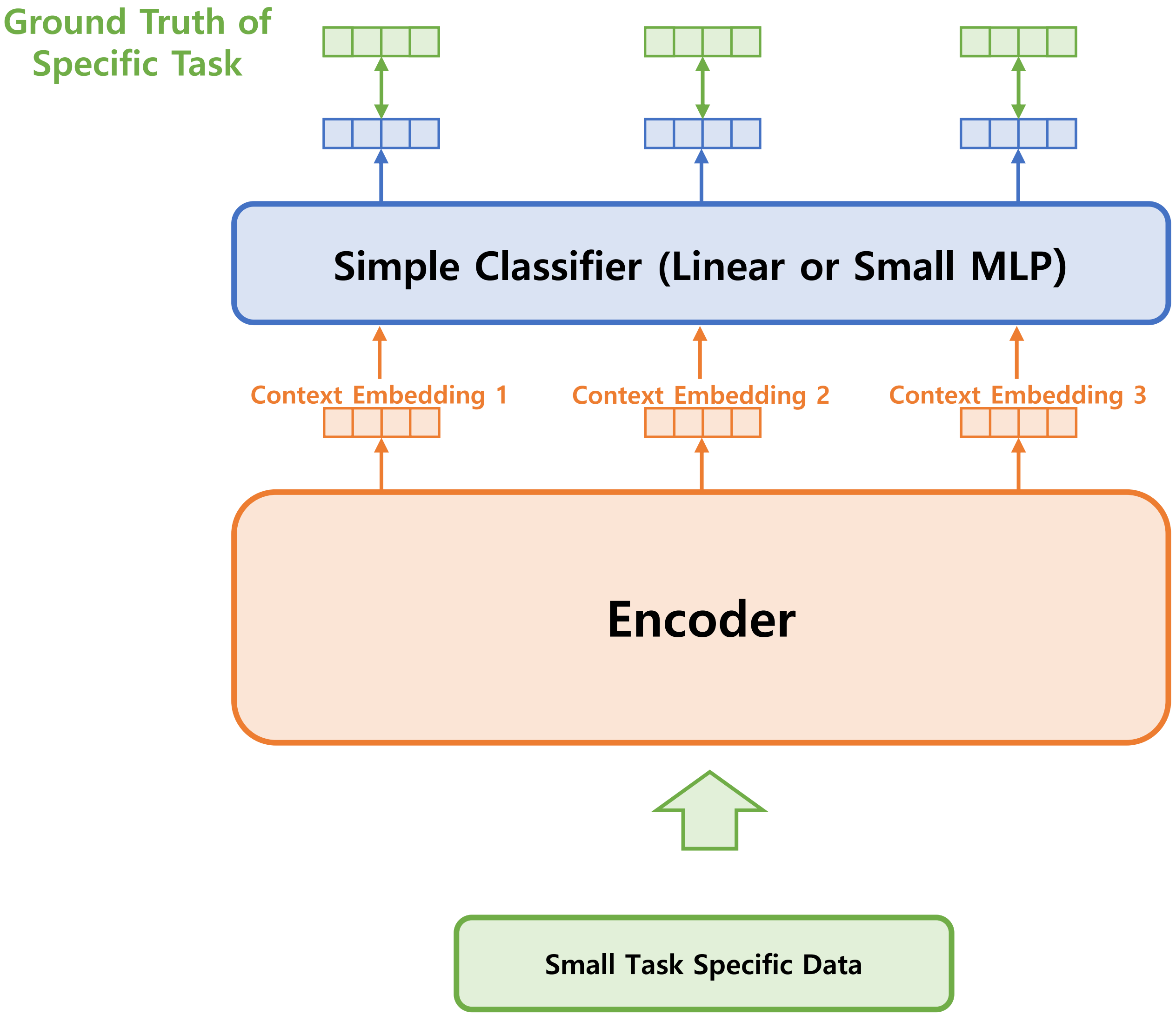


<Embeddings for the Word "die" in Different Contexts>

<Pre-Training>



<Fine-Tuning>


Ground Truth of Specific Task
Simple Classifier (Linear or Small MLP)
Context Embedding 1
Context Embedding 2
Context Embedding 3
Encoder
Small Task Specific Data

<Two Branches of Language Model Research>

"Bigger, Larger, Stronger"

"Small, But Better Performance"

<Two Branches of Language Model Research>

"Bigger, Larger, Stronger"

- Training Deep and Large Models with Huge Data

"Small, But Better Performance"

- Improving Performance without Changing the Structure of the Language Model
- Changing the Structure of the Model Without Significantly Increasing the Parameters

<Two Branches of Language Model Research>

"Bigger, Larger, Stronger"

- **Text To Text Transfer Transformer (T5)**
  - ✓ 11B Parameters
  - ✓ State-of-the-art in **GLUE**, etc.

  Raffel et al., Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer, JMLR, 2020

- **Generative Pre-Trained Transformer 3 (GPT-3)**
  - ✓ 175B Parameters
  - ✓ State-of-the-art in Many Benchmarks with Zero/Few Shot Setting

  Brown et al., Language Models are Few-Shot Learners, NeurIPS, 2020

"Small, But Better Performance"

- **SMART**
  - ✓ 356M Parameters
  - ✓ Beat T5 in 3 Tasks of GLUE

  Jiang et al., SMART: Robust and Efficient Fine-Tuning for Pre-trained Natural Language Models through Principled Regularized Optimization, ACL, 2020

- **Pattern-Exploiting Training (PET)**
  - ✓ 223M Parameters
  - ✓ Beat GPT-3 in SuperGLUE with Few Shot Setting

  Schick and Schutze, It's Not Just Size That Matters: Small Language Models are Also Few-Shot Learners, arXiv, 2020

<Two Branches of Language Model Research>

"Bigger, Larger, Stronger"

- Training Deep and Large Models with Huge Data

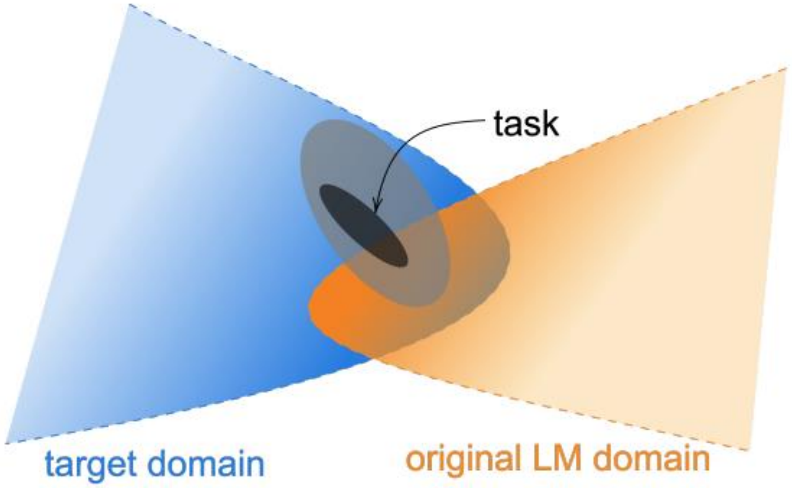
"Small, But Better Performance"

- Improving Performance **without Changing the Structure** of the Language Model
- Changing the Structure of the Model Without Significantly Increasing the Parameters

Gururangan et al., Don't Stop Pretraining: Adapt Language Models to Domains and Tasks, ACL, 2020

<Task-Adaptive Pre-Training>

Small Task Specific Data $\not\subset$ Very Large Text Corpora

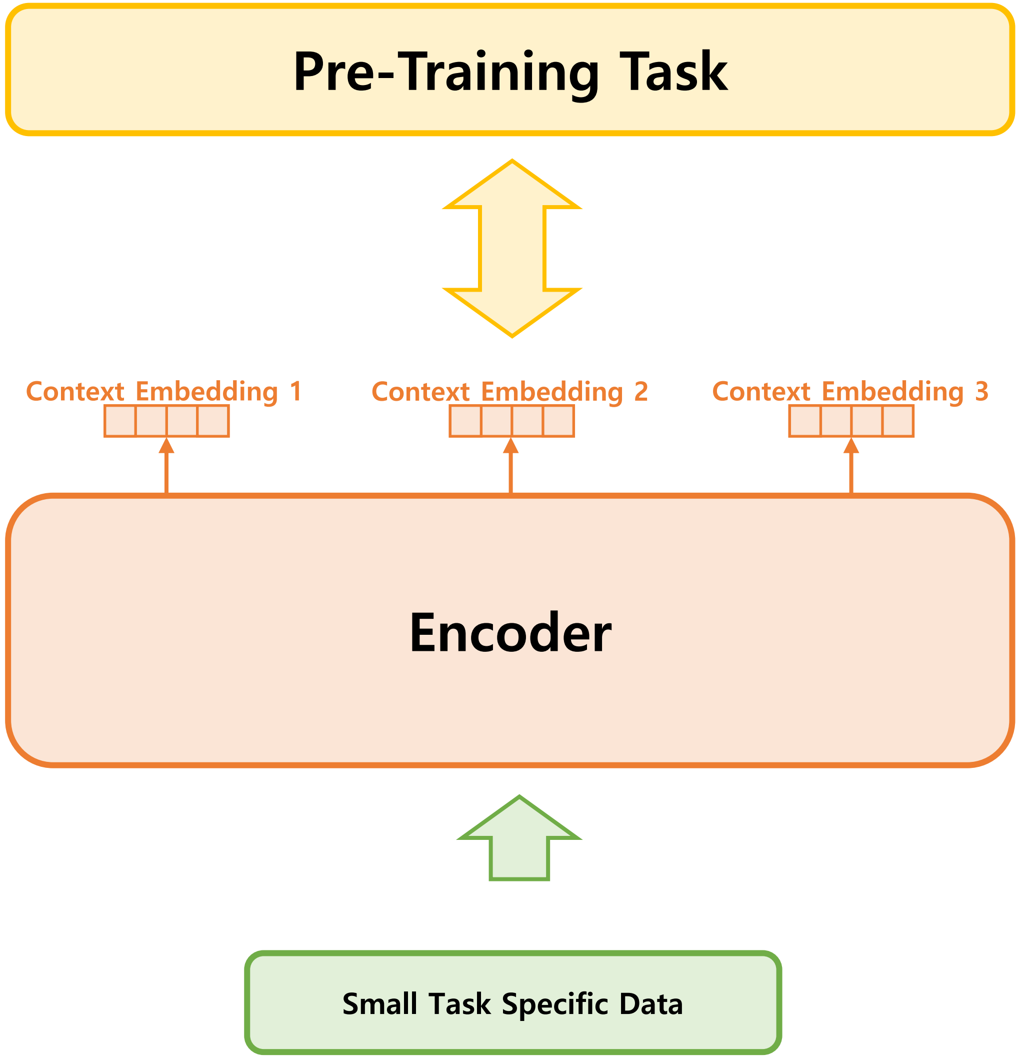


Introduction

-Transformer-Based Language Model

Gururangan et al., Don't Stop Pretraining: Adapt Language Models to Domains and Tasks, ACL, 2020

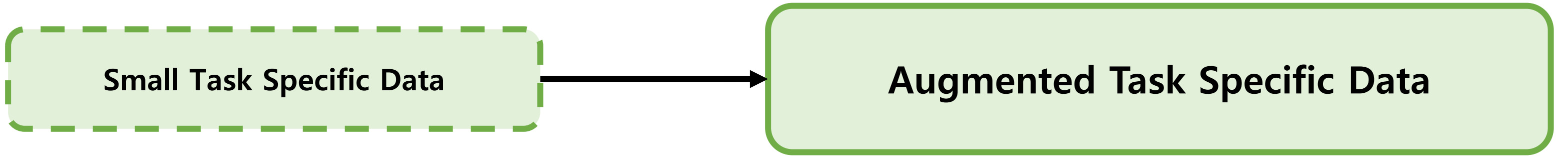
<Task-Adaptive Pre-Training>



Introduction

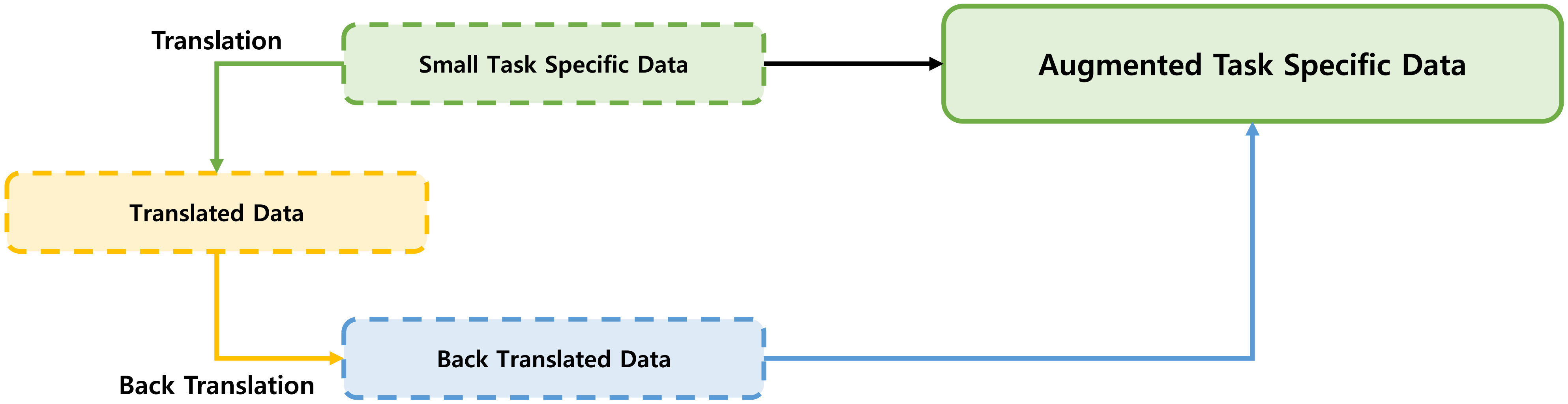
-Transformer-Based Language Model

<Text Augmentation>

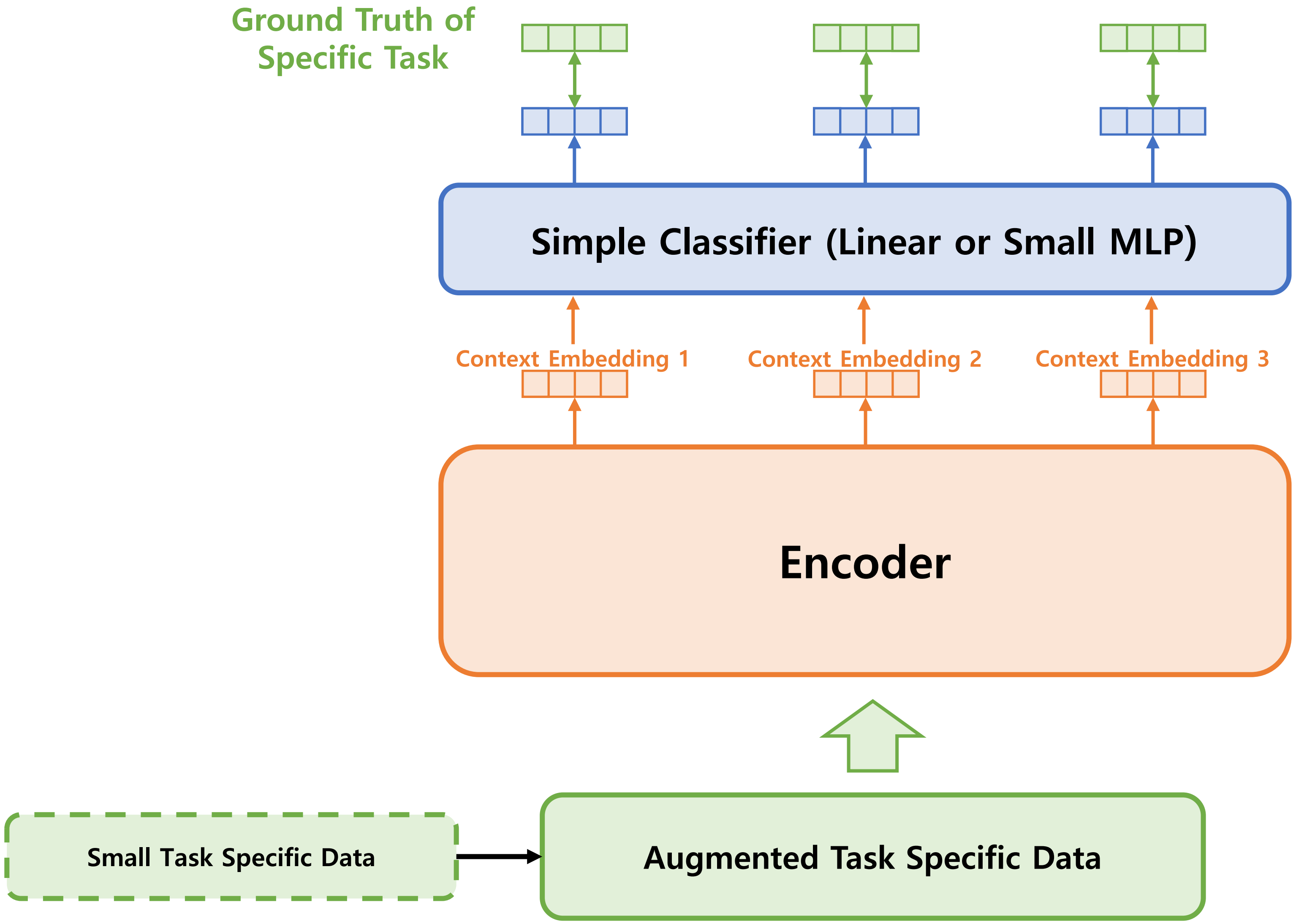


Edunov et al., Understanding Back-Translation at Scale, EMNLP, 2018

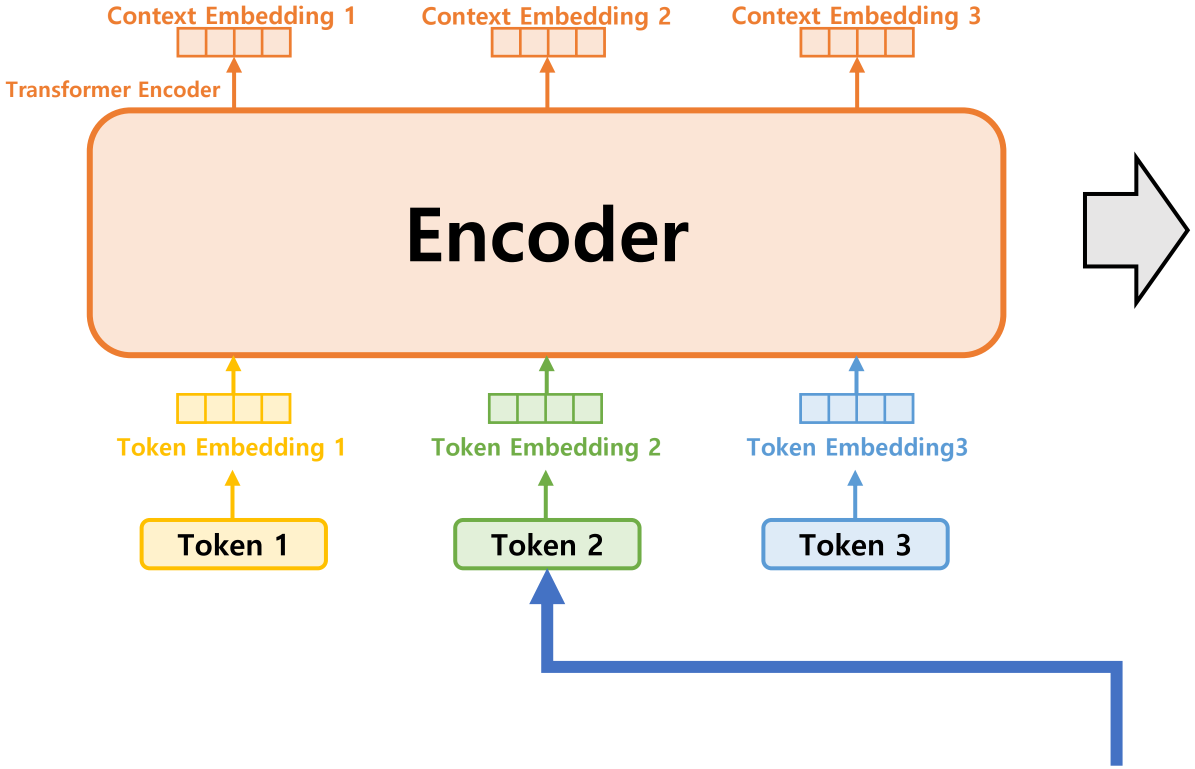
<Back Translation>

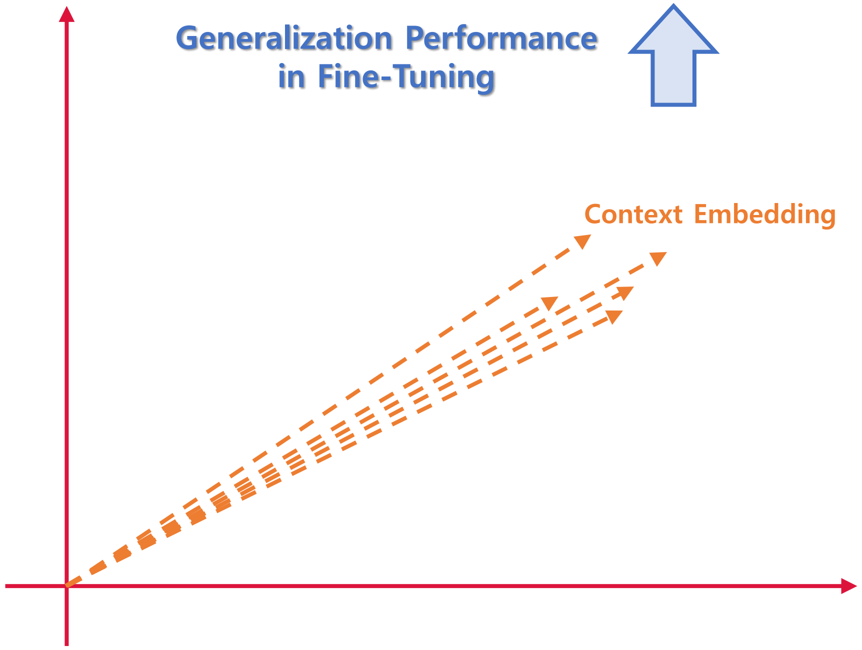


<Text Augmentation>



<Generalization Performance>





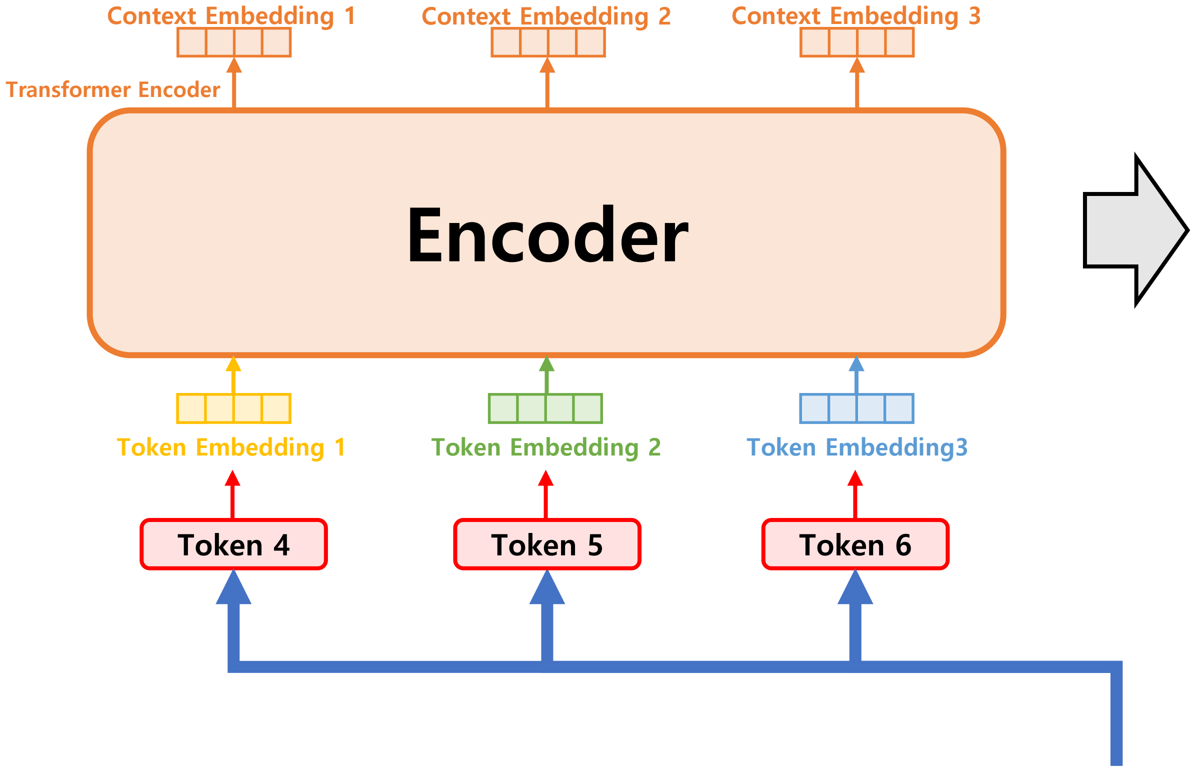
Task Adaptation

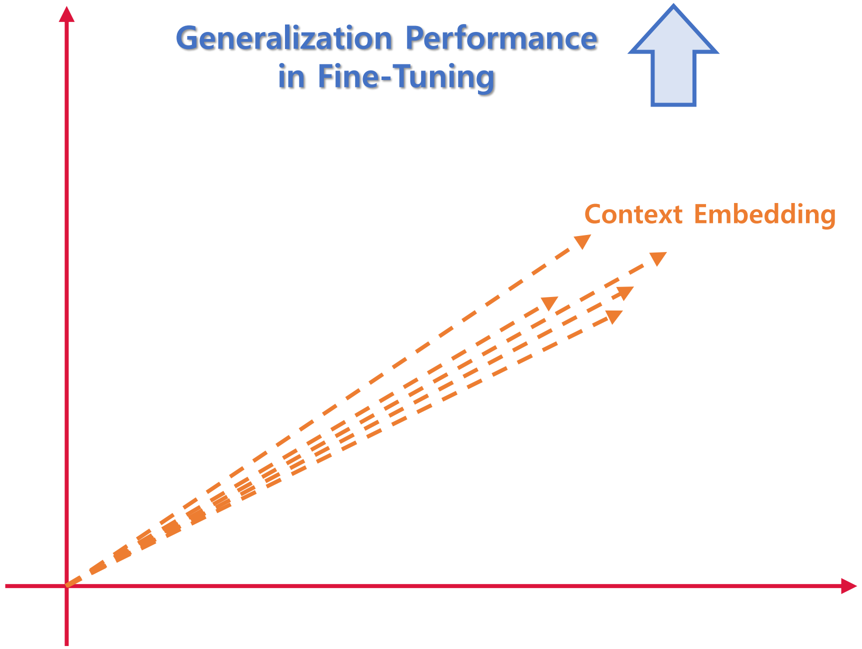
Text Augmentation

Introduction

-Transformer-Based Language Model

<Generalization Performance>



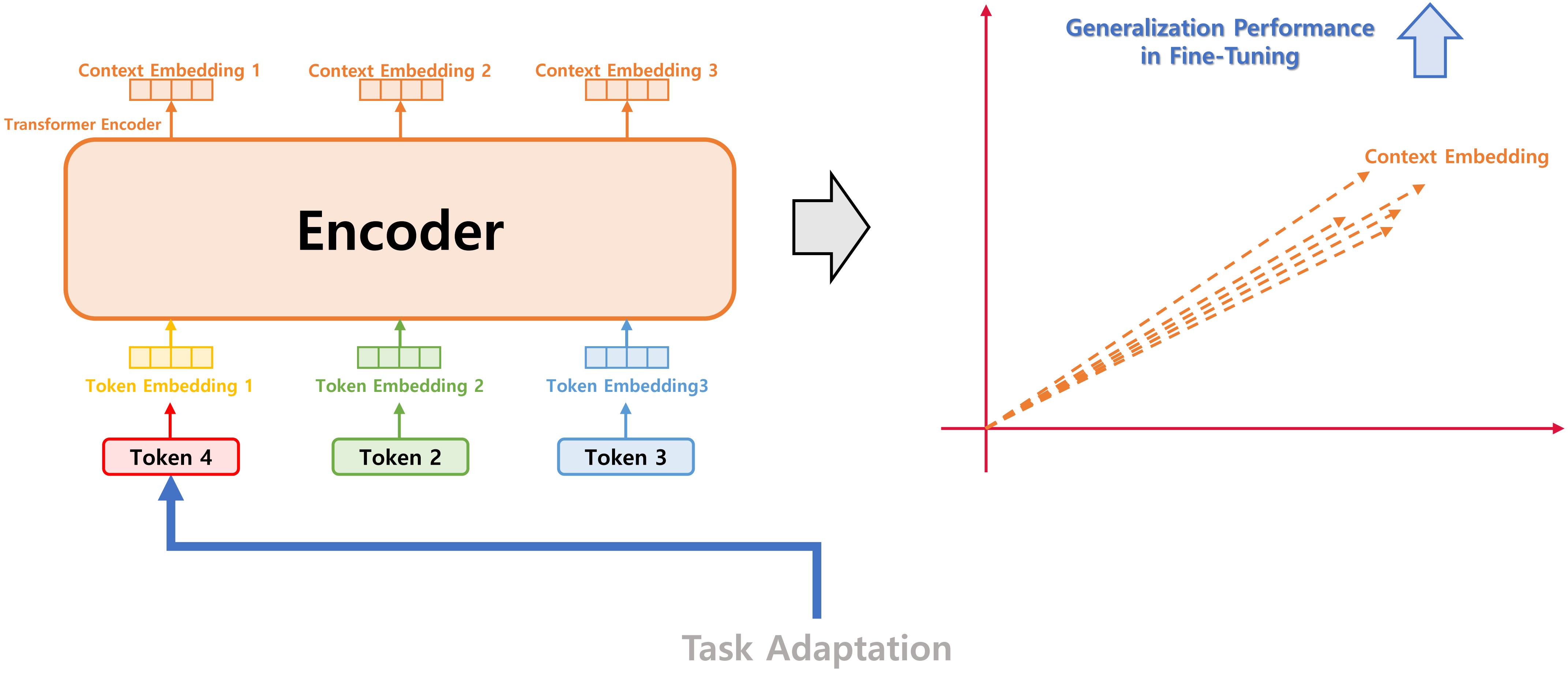


Task Adaptation

Text Augmentation

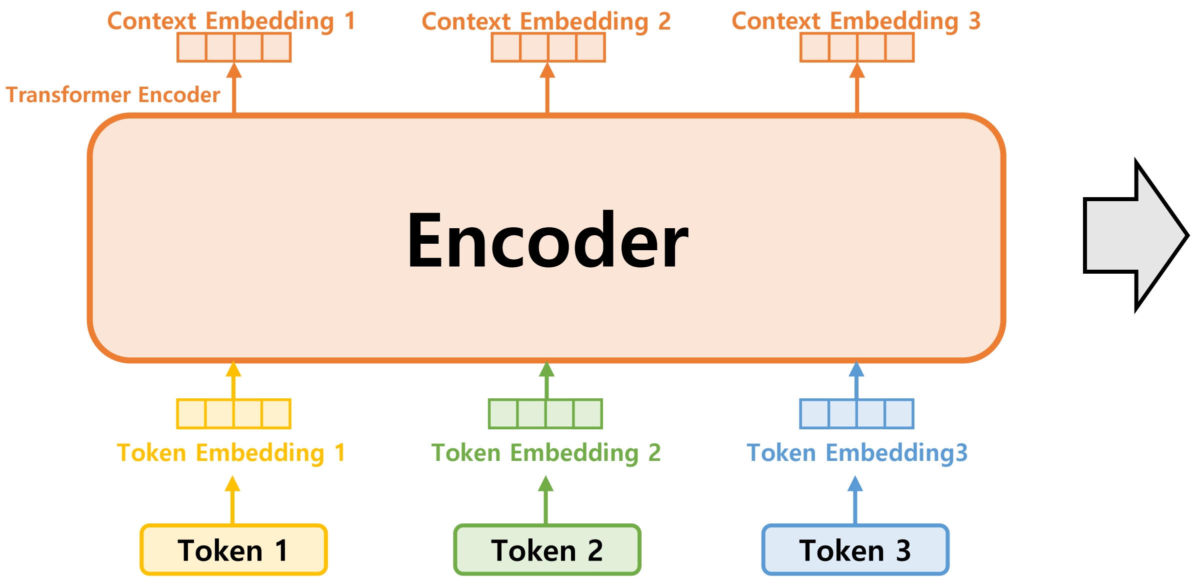
Introduction
-Transformer-Based Language Model

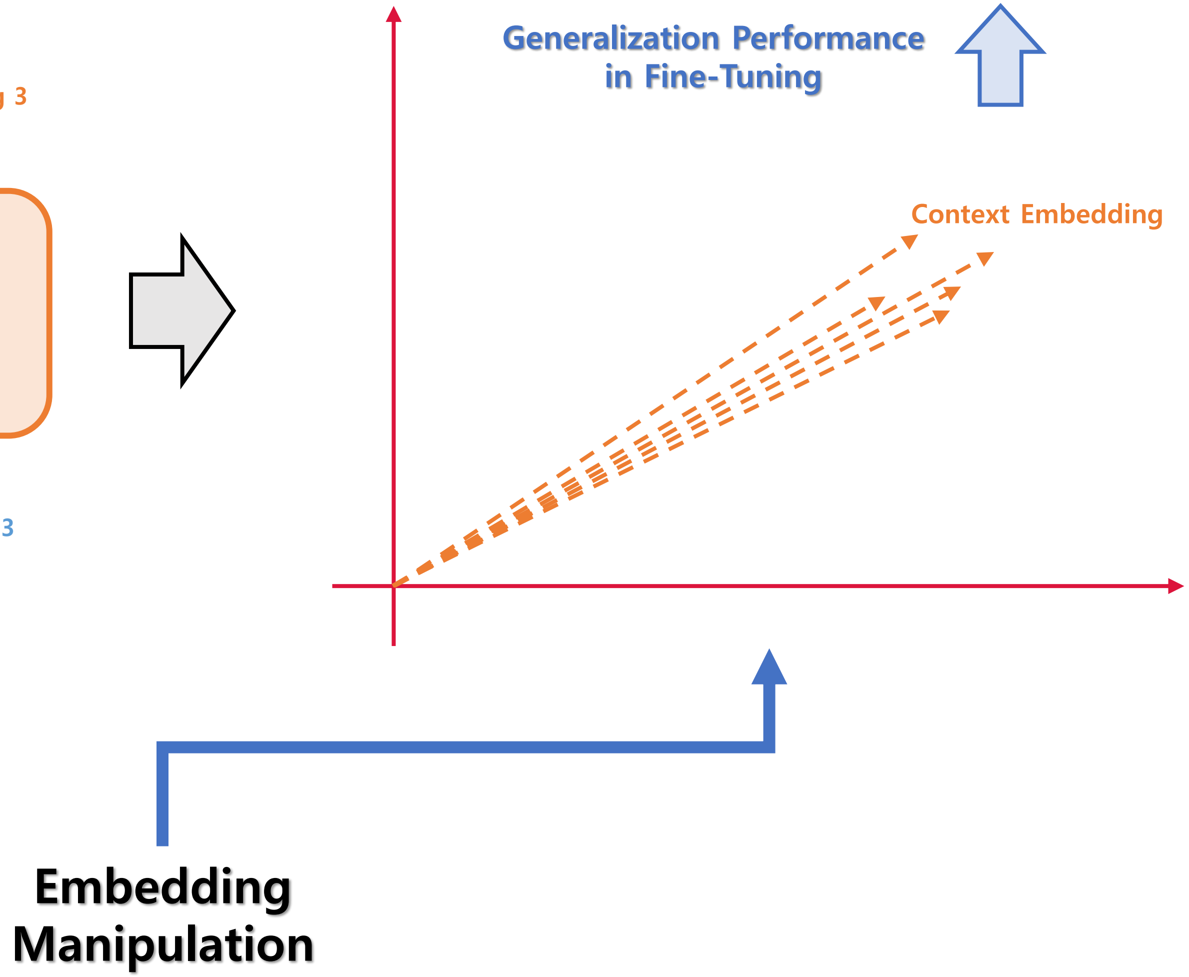
<Generalization Performance>



Text Augmentation

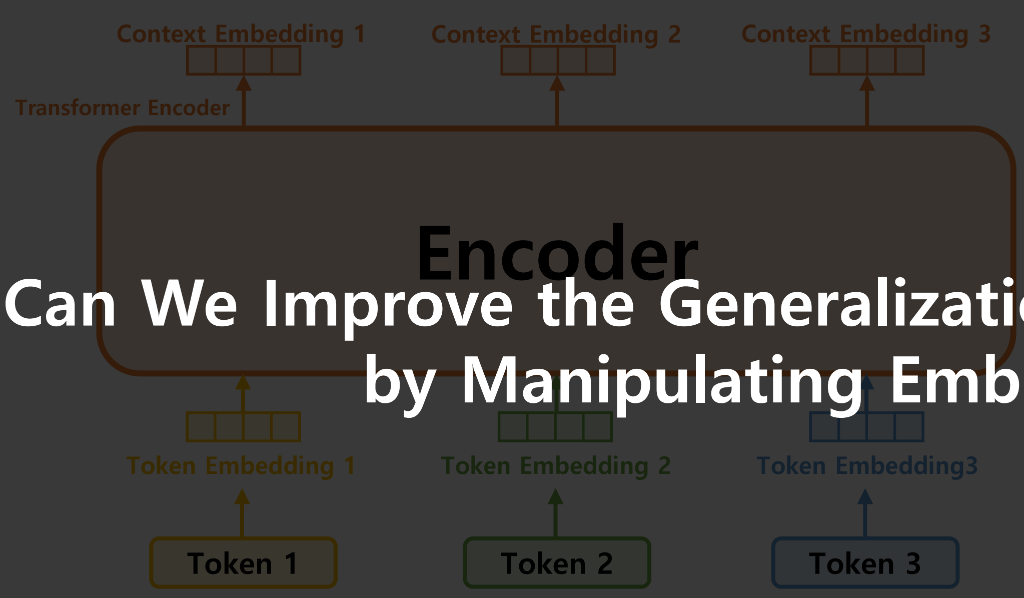
<Generalization Performance>

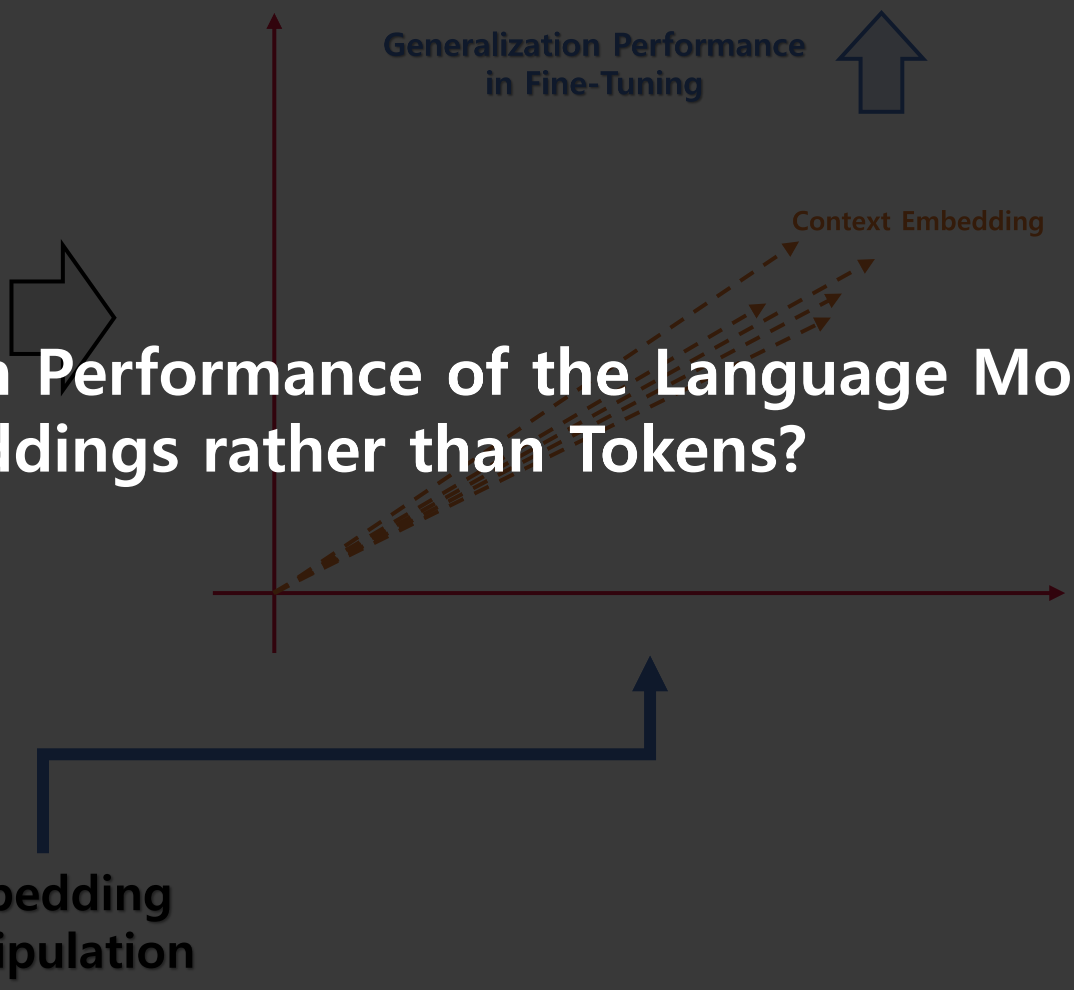




Introduction
-Transformer-Based Language Model

<Generalization Performance>





Can We Improve the Generalization Performance of the Language Model by Manipulating Embeddings rather than Tokens?

FreeLB: Enhanced Adversarial Training for Natural Language Understanding

Zhu et al., 2020, ICLR

FreeLB: Enhanced Adversarial Training for Natural Language Understanding

Zhu et al., 2020, ICLR

Pre-requisites

- Adversarial Training

Pre-requisites

- Adversarial Training

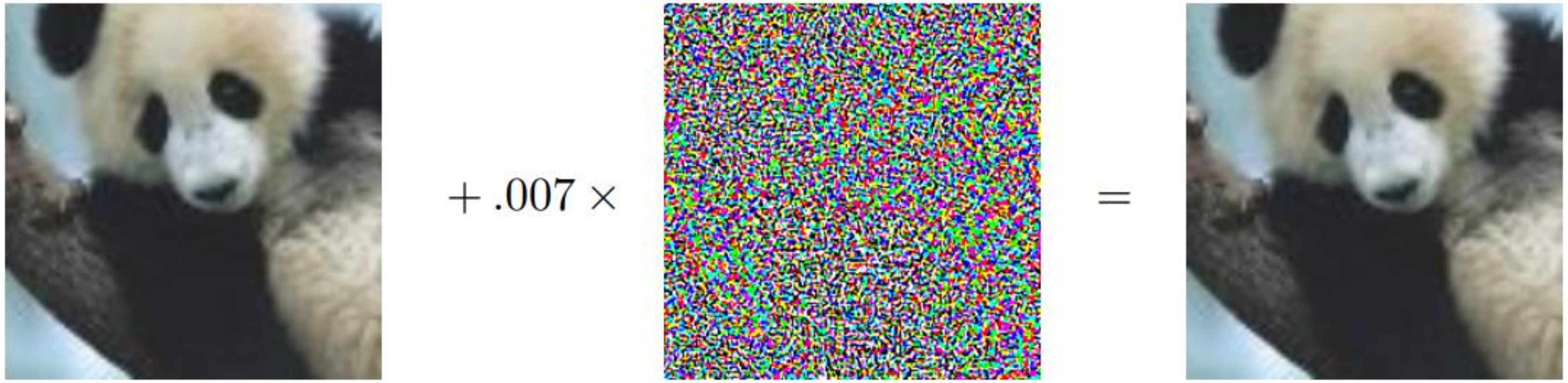
<Recap: Previous Seminar>

Adversarial example

- 특수한 noise를 원본 example에 더하여 사람이 판단하기에는 똑같지만, machine이 판단하기에는 다른 class가 되는 example
- 예를 들어 오른쪽 이미지는 우리가 보기에는 여전히 고양이지만 DNN이 보기에는 오븐이 된다!

Goodfellow et al., Explaining and Harnessing Adversarial Examples, ICLR, 2015

<Adversarial Example>

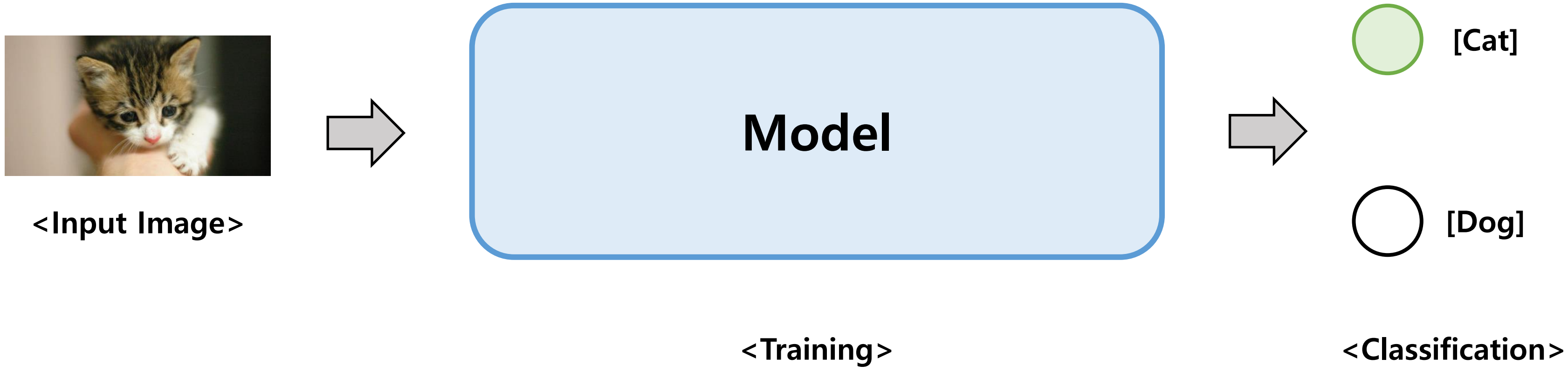


$\boldsymbol{x}$ + .007 × $\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$ = $\boldsymbol{x} + \epsilon\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$

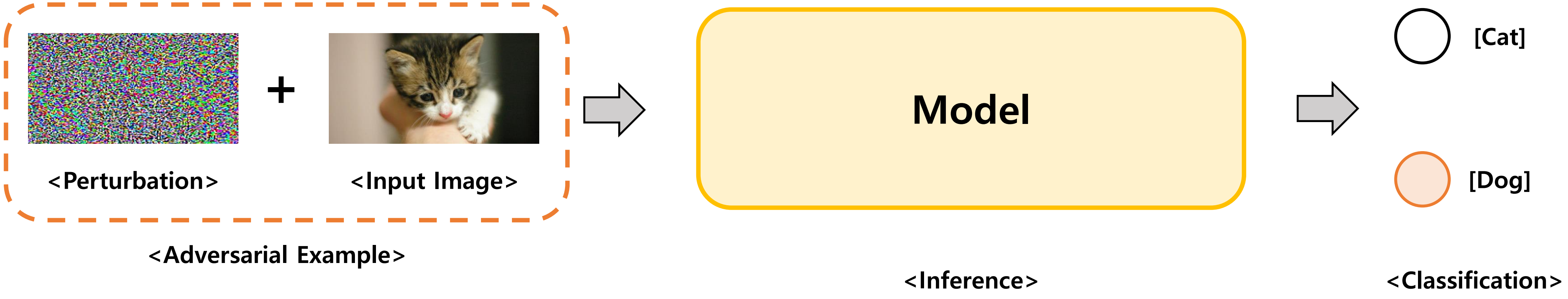
$\boldsymbol{x}$	$\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$	$\boldsymbol{x} + \epsilon\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$
"panda" 57.7% confidence	"nematode" 8.2% confidence	"gibbon" 99.3 % confidence

Pre-requisites
- Adversarial Training

<Adversarial Training>



<Adversarial Training>



<Adversarial Training>

<Pertu mage>

<Adversarial Example>

Model

<Inference>

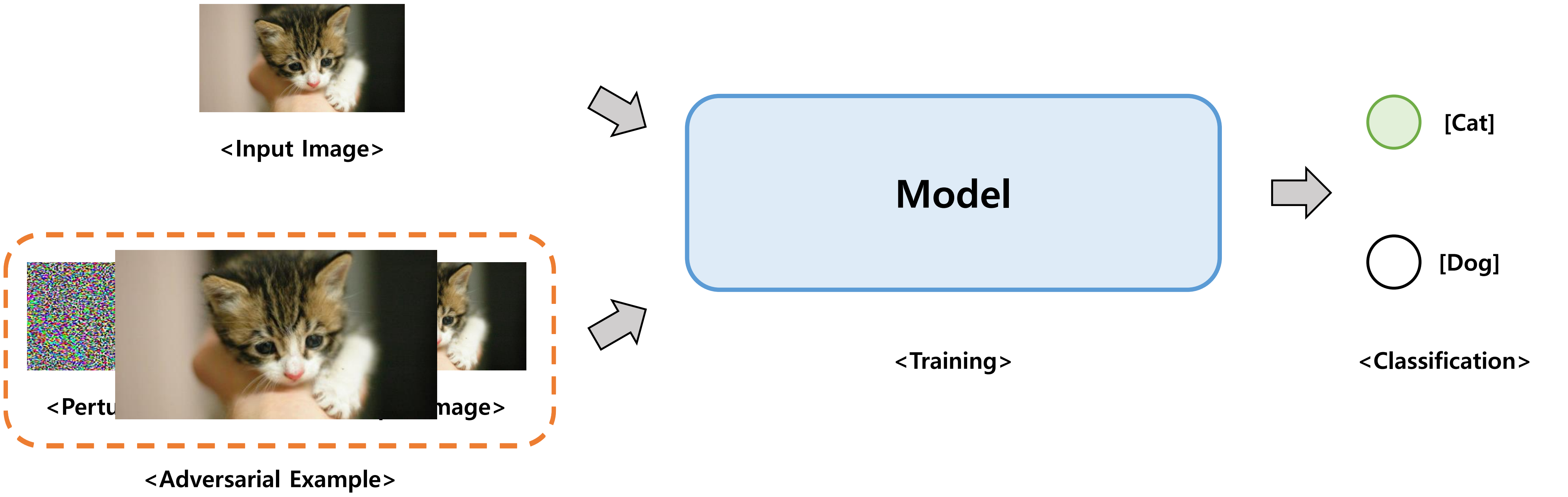
[Cat]

[Dog]

<Classification>

Pre-requisites
- Adversarial Training

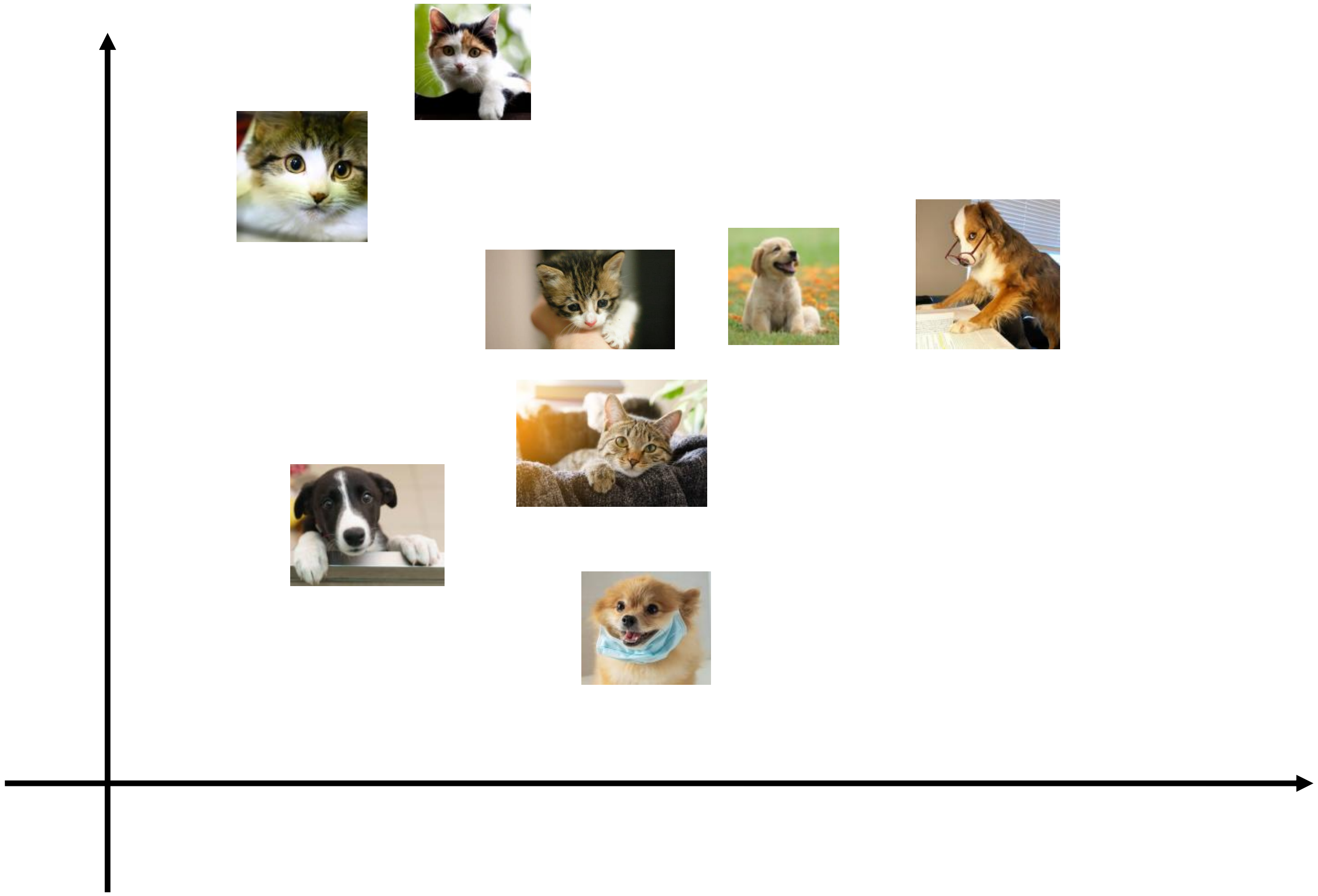
<Adversarial Training>



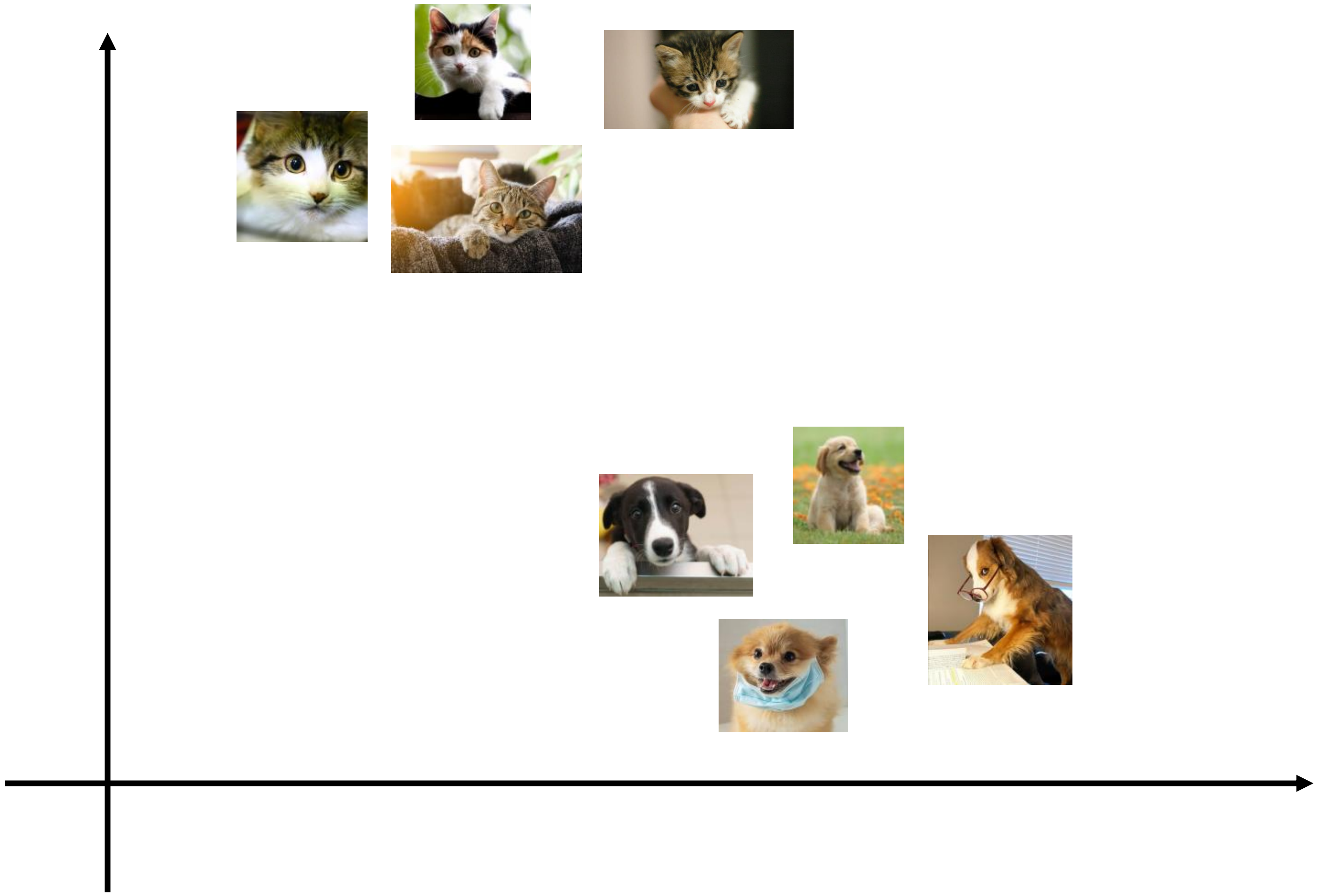
Pre-requisites

- Adversarial Training

<Adversarial Training>



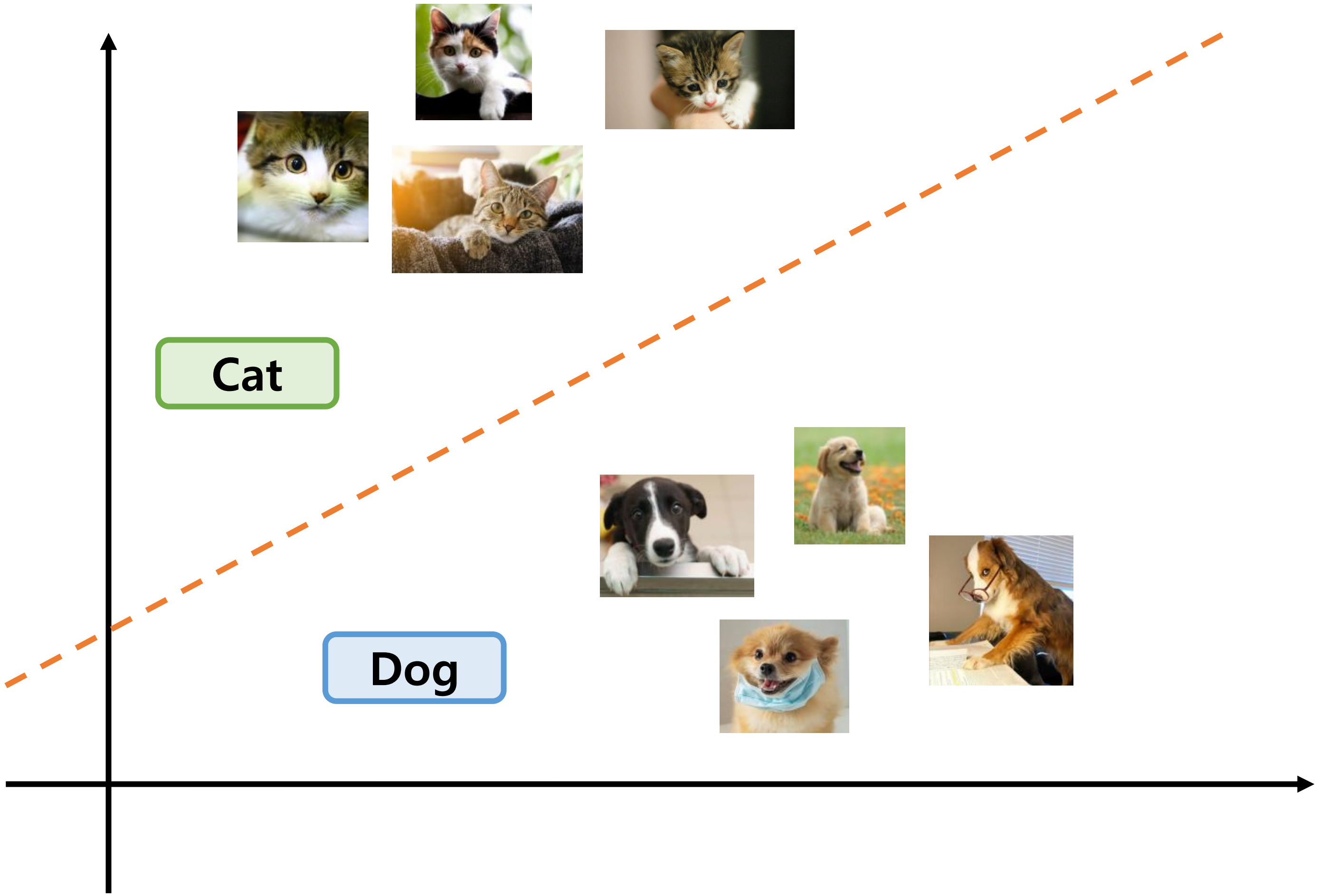
<Adversarial Training>



Pre-requisites

- Adversarial Training

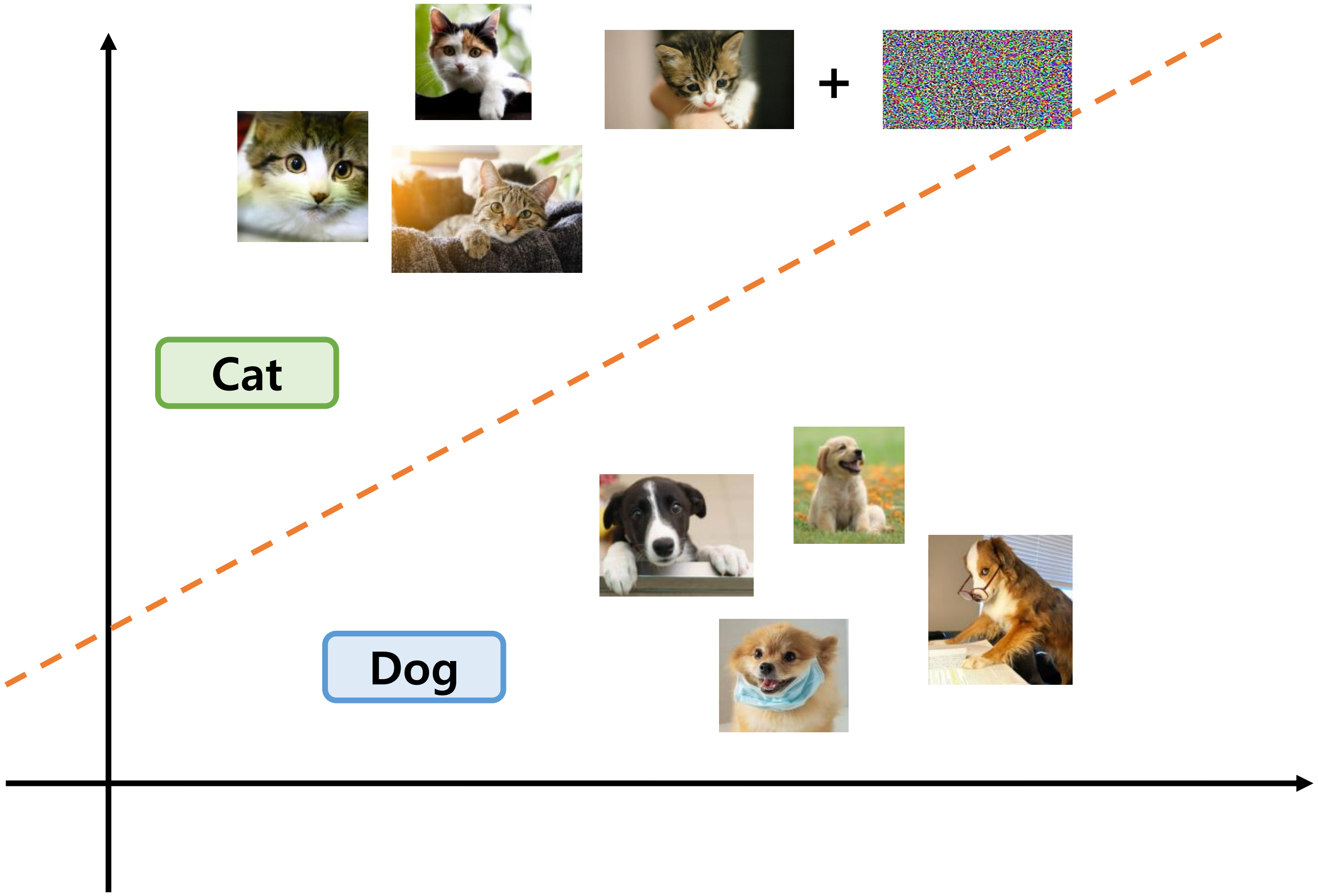
<Adversarial Training>



Pre-requisites

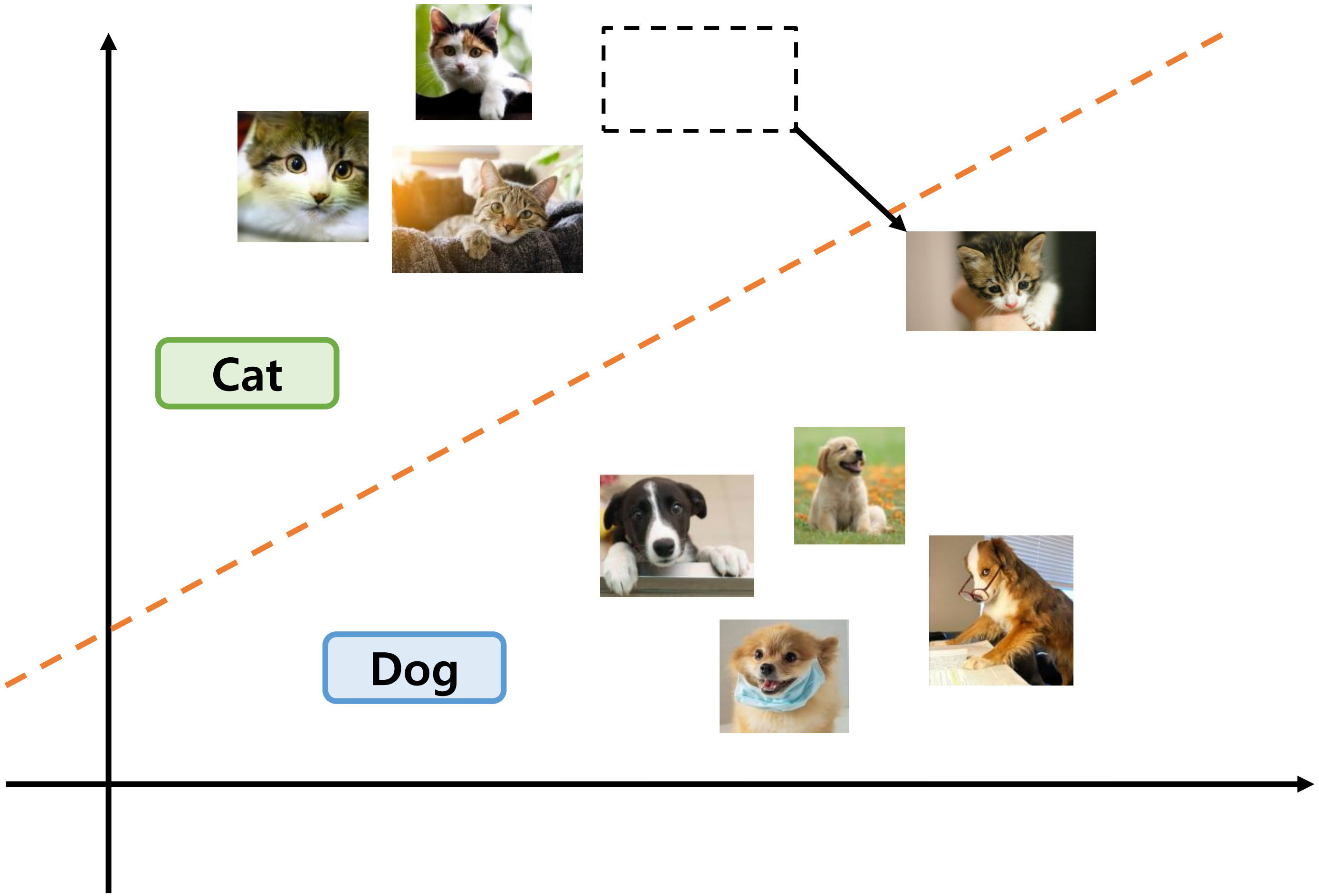
- Adversarial Training

<Adversarial Training>



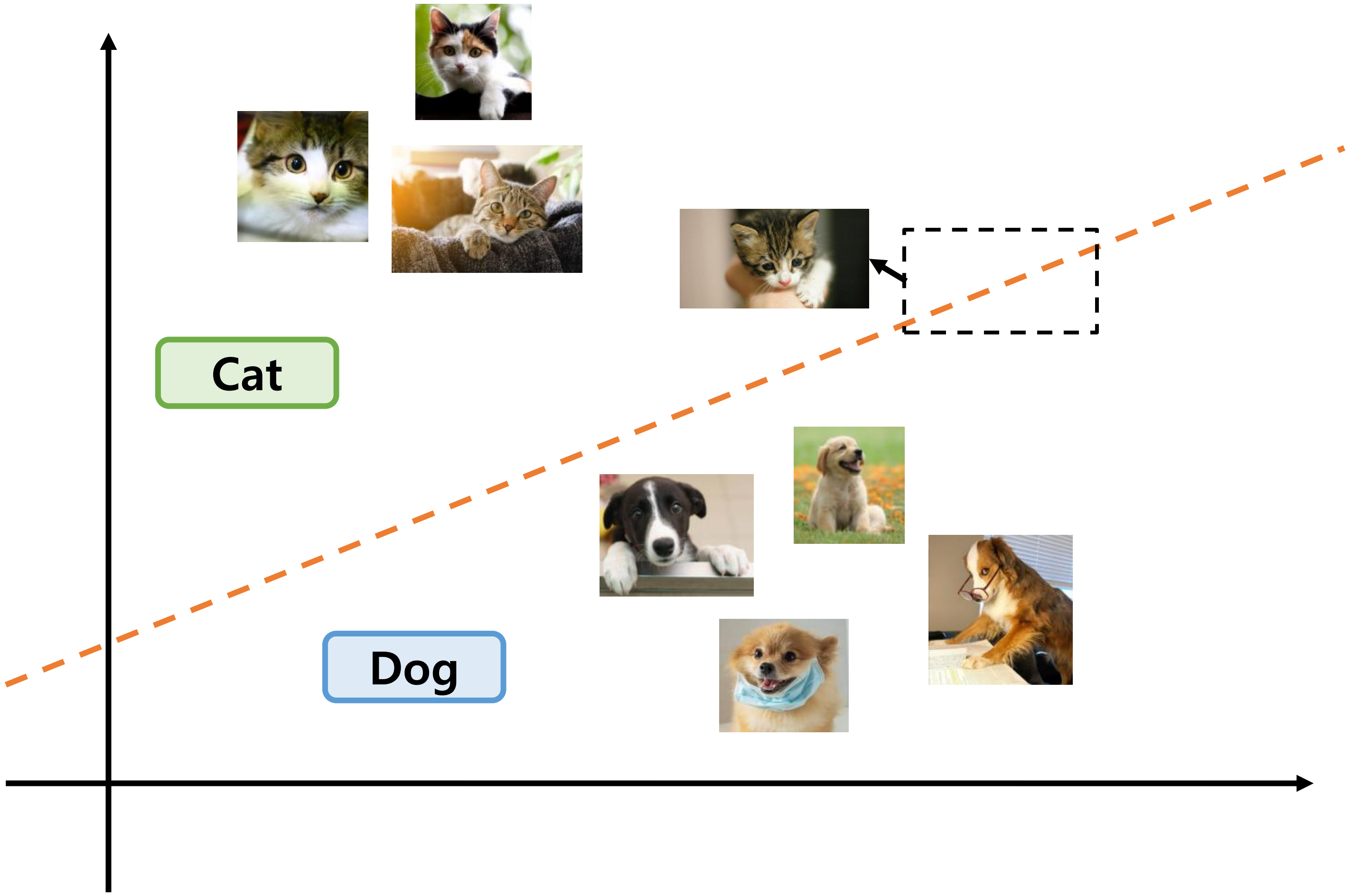
From [Paper Review] Adversarial Defense by Restricting the Hidden Space of Deep Neural Network (Seungwan Seo, 2020)

<Adversarial Training>



From [Paper Review] Adversarial Defense by Restricting the Hidden Space of Deep Neural Network (Seungwan Seo, 2020)

<Adversarial Training>



Pre-requisites
-Adversarial Training

<Adversarial Training>

?

Adversarial Training for NLU

- PGD-Based Adversarial Training
- Large-Batch Adversarial Training for Free

Adversarial Training for NLU

- PGD-Based Adversarial Training

<Projected Gradient Descent>

$$\min_{\theta} \mathbb{E}_{(Z,y)\sim\mathcal{D}}\left[\max_{||\delta||\leq\varepsilon} L(f_\theta(X+\delta),y)\right]$$

$$\delta_{t+1} = \Pi_{||\delta||_F\leq\varepsilon}(\delta_t + \alpha g(\delta_t) \,/\, ||g(\delta_t)||_F)$$

$Z: One-Hot\ Encoding$

$V: Embedding\ Matrix$

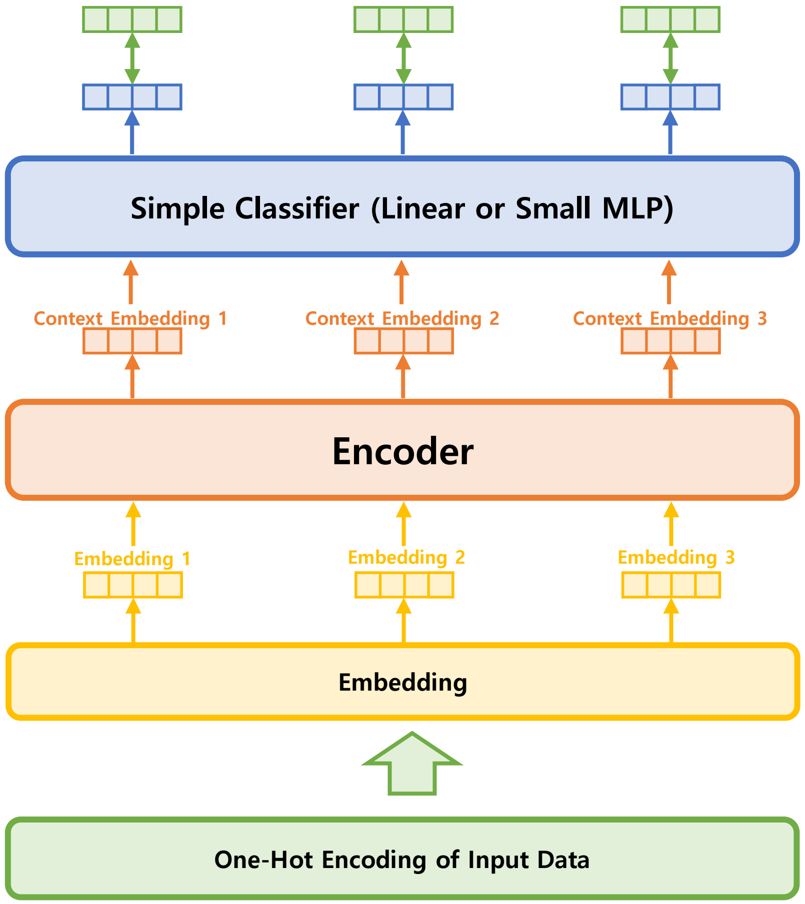
$X = VZ: Embedding$

$f_\theta(X): Language\ Model\ (Encoder)\ as\ Function$

$\theta: All\ Learnable\ Parameter\ in\ Language\ Model$

$y: Label$

$\delta: Perturbation$



Adversarial Training for NLU

- PGD-Based Adversarial Training

<Projected Gradient Descent>

$$\min_{\theta} \mathbb{E}_{(Z,\, y)\sim\mathcal{D}} \left[\max_{||\delta|| \leq \varepsilon} L(f_\theta(X+\delta), y) \right]$$

$$\delta_{t+1} = \Pi_{||\delta||_F \leq \varepsilon}\left(\delta_t + \alpha g(\delta_t) \,/\, ||g(\delta_t)||_F\right)$$

$Z: One - Hot\ Encoding$

$V: Embedding\ Matrix$

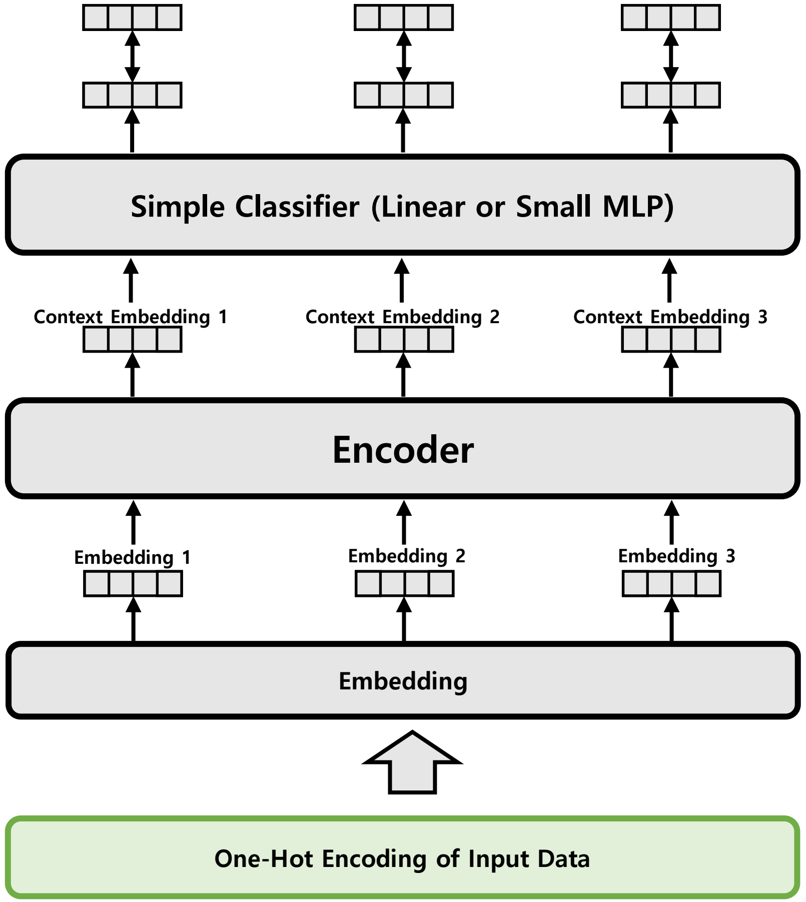
$X = VZ: Embedding$

$f_\theta(X): Language\ Model\ (Encoder)\ as\ Function$

$\theta: All\ Learnable\ Parameter\ in\ Language\ Model$

$y: Label$

$\delta: Perturbation$



<Projected Gradient Descent>

$$\min_{\theta} \mathbb{E}_{(Z,y)\sim\mathcal{D}} \left[\max_{||\delta|| \leq \varepsilon} L(f_\theta(X+\delta), y) \right]$$

$$\delta_{t+1} = \Pi_{||\delta||_F \leq \varepsilon} (\delta_t + \alpha g(\delta_t) \,/\, ||g(\delta_t)||_F)$$

$Z: One - Hot\ Encoding$

$V: Embedding\ Matrix$

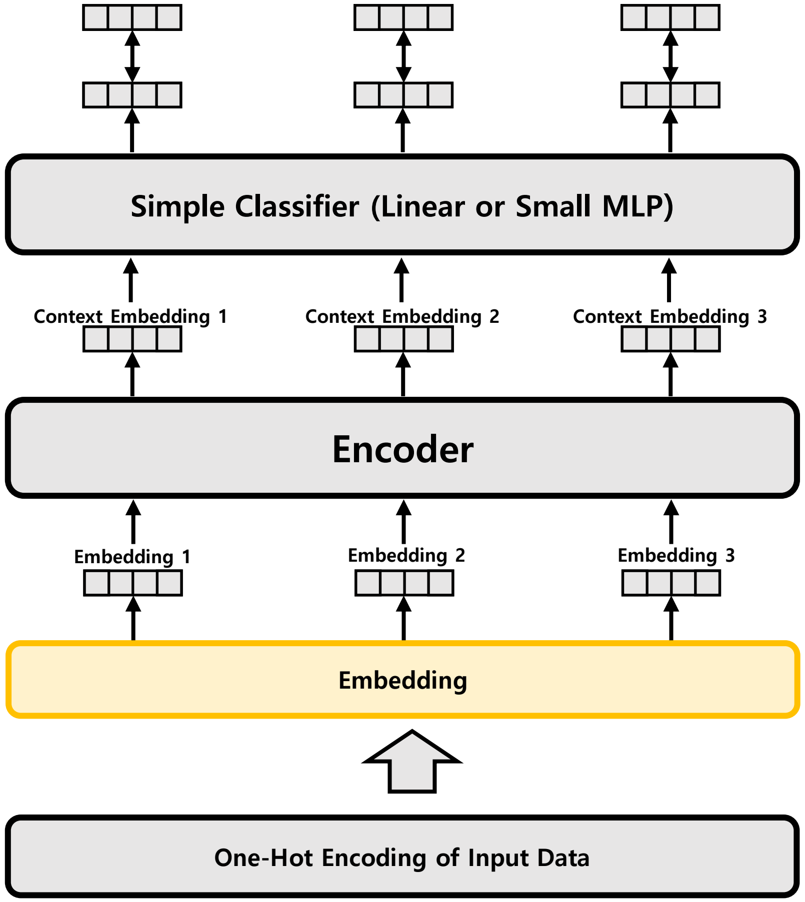
$X = VZ: Embedding$

$f_\theta(X): Language\ Model\ (Encoder)\ as\ Function$

$\theta: All\ Learnable\ Parameter\ in\ Language\ Model$

$y: Label$

$\delta: Perturbation$



Adversarial Training for NLU

- PGD-Based Adversarial Training

<Projected Gradient Descent>

$$\min_{\theta} \mathbb{E}_{(Z,y)\sim\mathcal{D}}\left[\max_{||\delta||\leq\varepsilon} L(f_\theta(X+\delta),y)\right]$$

$$\delta_{t+1} = \Pi_{||\delta||_F\leq\varepsilon}(\delta_t + \alpha g(\delta_t) / ||g(\delta_t)||_F)$$

$Z: One - Hot\ Encoding$

$V: Embedding\ Matrix$

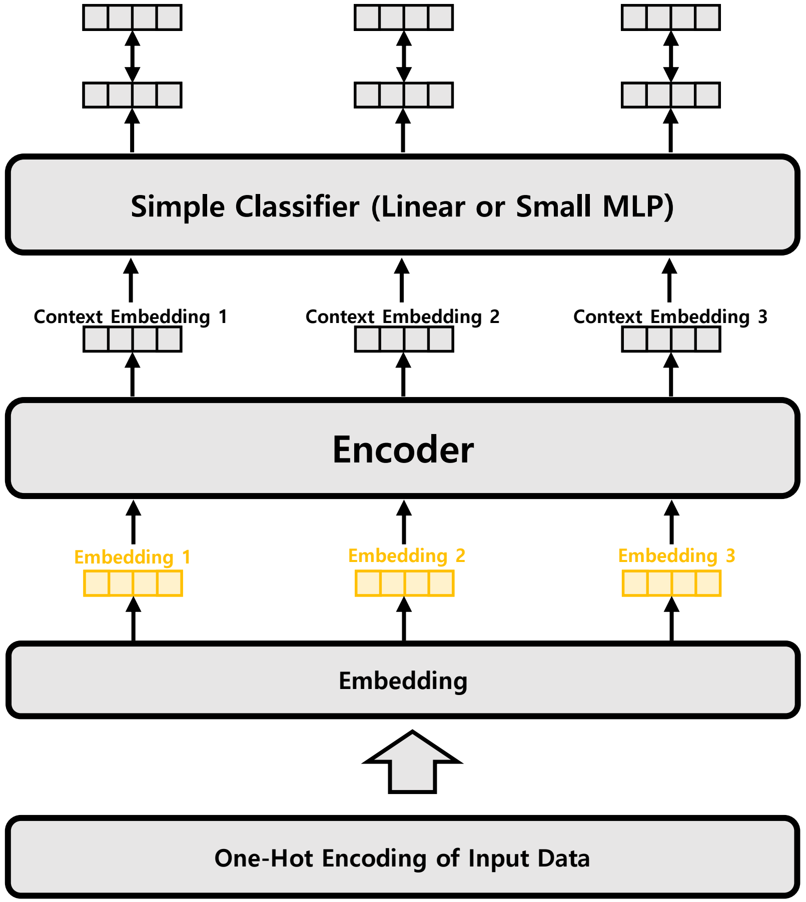
$X = VZ: Embedding$

$f_\theta(X): Language\ Model\ (Encoder)\ as\ Function$

$\theta: All\ Learnable\ Parameter\ in\ Language\ Model$

$y: Label$

$\delta: Perturbation$



Adversarial Training for NLU

- PGD-Based Adversarial Training

<Projected Gradient Descent>

$$\min_{\theta} \mathbb{E}_{(Z,y)\sim\mathcal{D}} \left[\max_{||\delta|| \leq \varepsilon} L(f_{\theta}(X+\delta), y) \right]$$

$$\delta_{t+1} = \Pi_{||\delta||_F \leq \varepsilon}(\delta_t + \alpha g(\delta_t) \,/\, ||g(\delta_t)||_F)$$

$Z: One - Hot\ Encoding$

$V: Embedding\ Matrix$

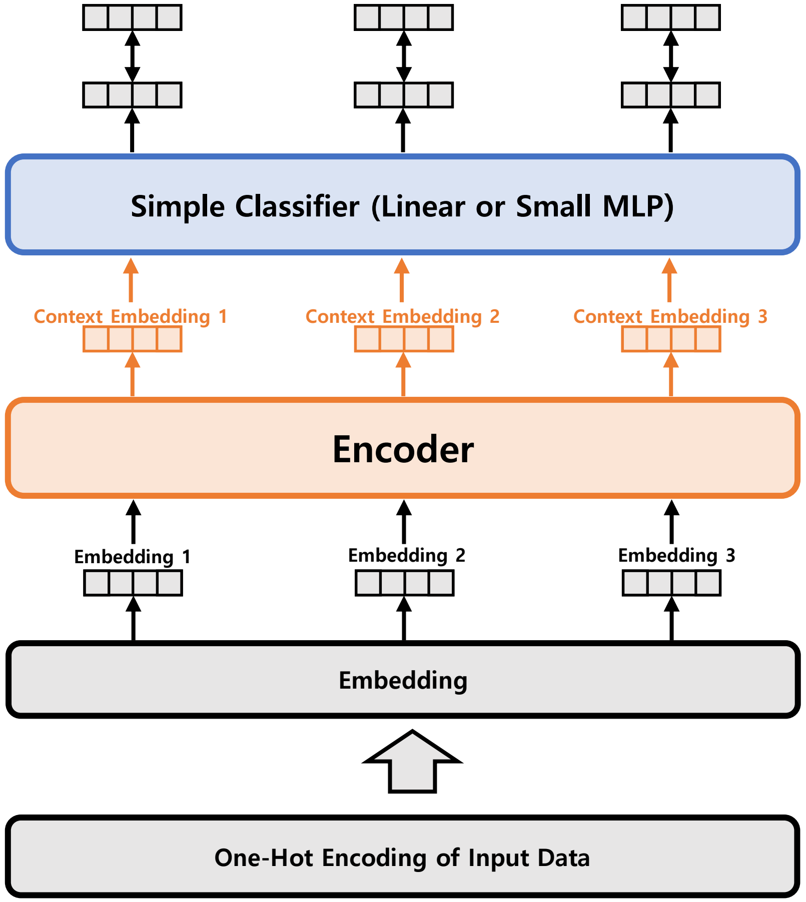
$X = VZ: Embedding$

$f_{\theta}(X): Language\ Model\ (Encoder)\ as\ Function$

$\theta: All\ Learnable\ Parameter\ in\ Language\ Model$

$y: Label$

$\delta: Perturbation$



<Projected Gradient Descent>

$$\min_{\Theta} \mathbb{E}_{(Z,y)\sim\mathcal{D}}\left[\max_{||\delta||\leq\varepsilon} L(f_\theta(X+\delta), y)\right]$$

$$\delta_{t+1} = \Pi_{||\delta||_F\leq\varepsilon}(\delta_t + \alpha g(\delta_t) / ||g(\delta_t)||_F)$$

$Z: One-Hot\ Encoding$

$V: Embedding\ Matrix$

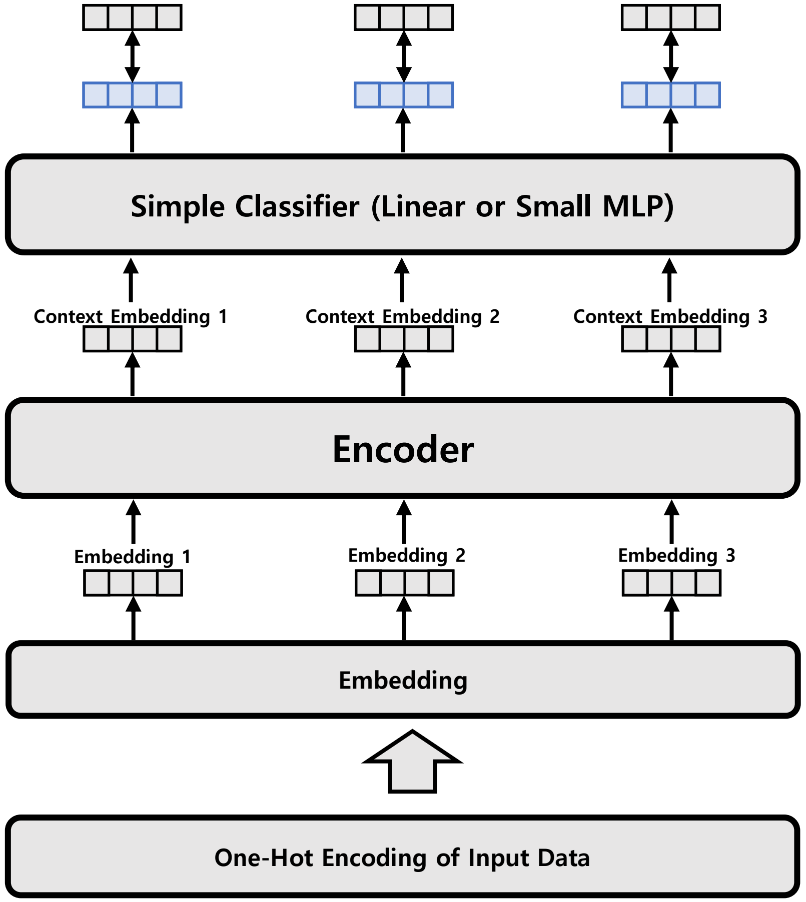
$X = VZ: Embedding$

$f_\theta(X): Language\ Model\ (Encoder)\ as\ Function$

$\theta: All\ Learnable\ Parameter\ in\ Language\ Model$

$y: Label$

$\delta: Perturbation$



Adversarial Training for NLU
- PGD-Based Adversarial Training

<Projected Gradient Descent>

$$\min_{\Theta} \mathbb{E}_{(Z,\, y)\sim\mathcal{D}} \left[\max_{||\delta|| \le \varepsilon} L(f_\theta(X+\delta), y) \right]$$

$$\delta_{t+1} = \Pi_{||\delta||_F \le \varepsilon}(\delta_t + \alpha g(\delta_t) \,/\, ||g(\delta_t)||_F)$$

$Z: One - Hot\ Encoding$

$V: Embedding\ Matrix$

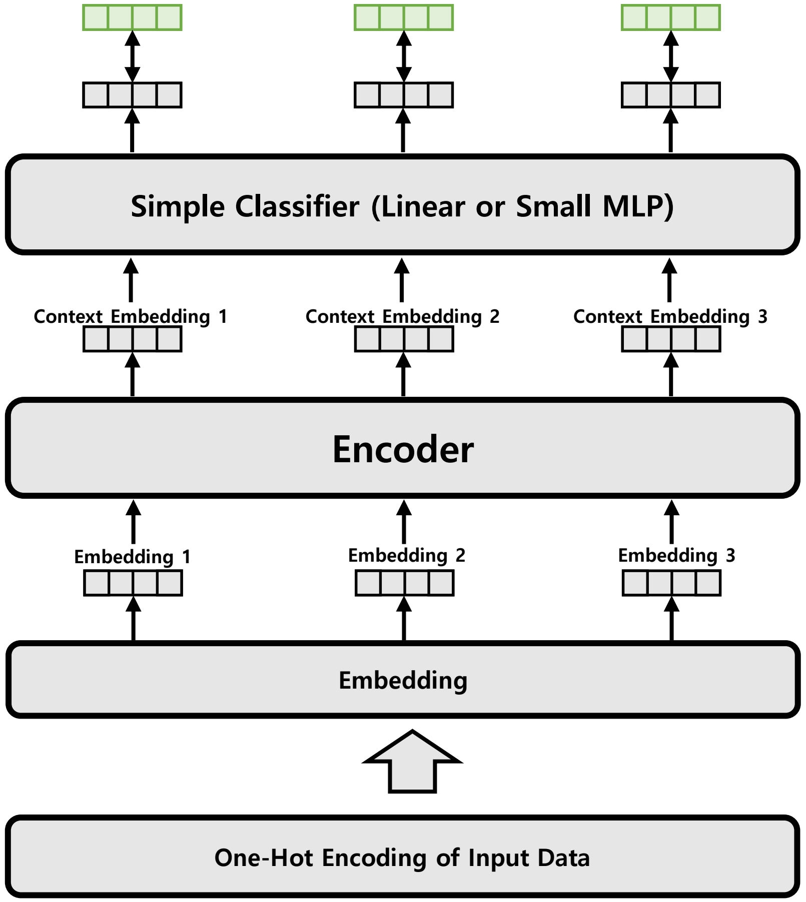
$X = VZ: Embedding$

$f_\theta(X): Language\ Model\ (Encoder)\ as\ Function$

$\theta: All\ Learnable\ Parameter\ in\ Language\ Model$

$y: Label$

$\delta: Perturbation$



Adversarial Training for NLU
- PGD-Based Adversarial Training

<Projected Gradient Descent>

$$\min_{\theta} \mathbb{E}_{(Z,y)\sim\mathcal{D}}\left[\max_{||\delta||\leq\varepsilon} L(f_\theta(X+\delta), y)\right]$$

$$\delta_{t+1} = \Pi_{||\delta||_F\leq\varepsilon}(\delta_t + \alpha g(\delta_t) / ||g(\delta_t)||_F)$$

$Z: One - Hot\ Encoding$

$V: Embedding\ Matrix$

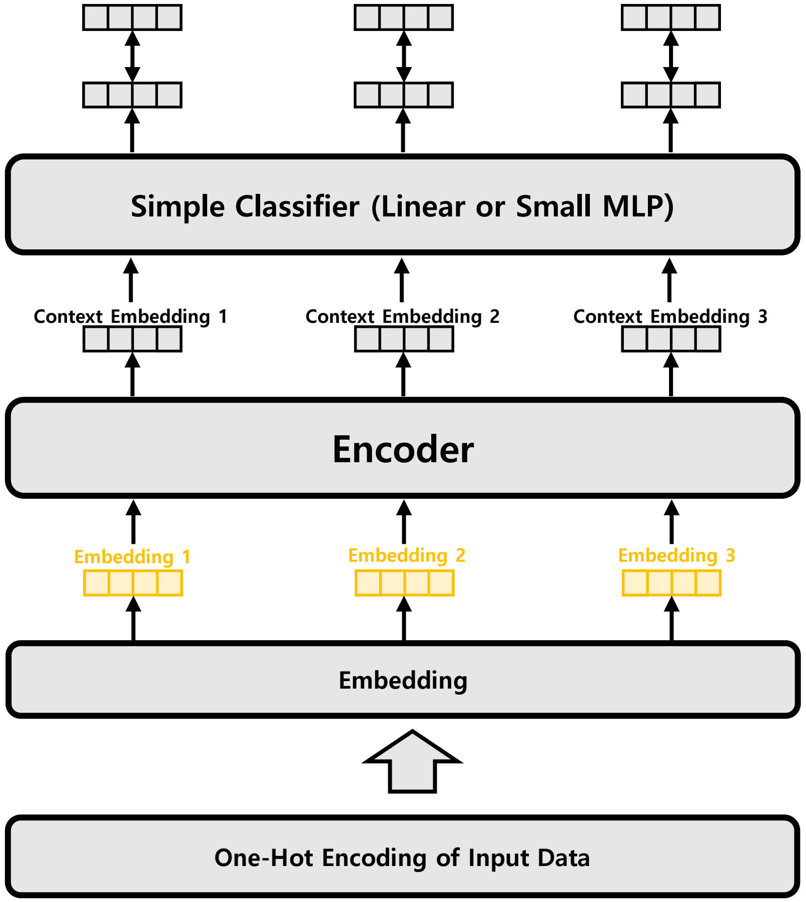
$X = VZ: Embedding$

$f_\theta(X): Language\ Model\ (Encoder)\ as\ Function$

$\theta: All\ Learnable\ Parameter\ in\ Language\ Model$

$y: Label$

$\delta: Perturbation$



<Projected Gradient Descent>

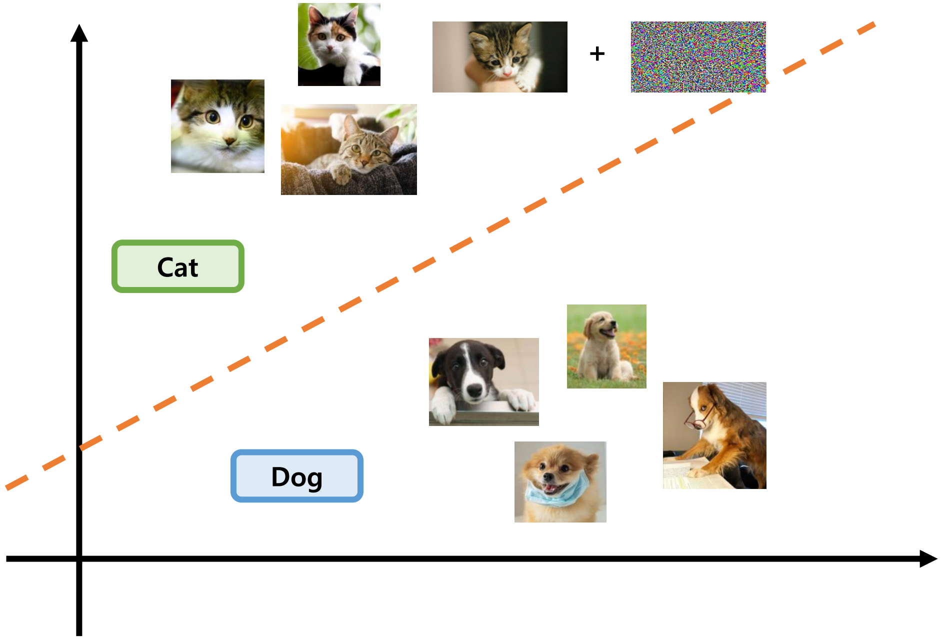
$$\min_{\theta} \mathbb{E}_{(Z,y)\sim\mathcal{D}}\left[\max_{||\boldsymbol{\delta}||\leq \boldsymbol{\varepsilon}} \boldsymbol{L}(\boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{X}+\boldsymbol{\delta}),\boldsymbol{y})\right]$$

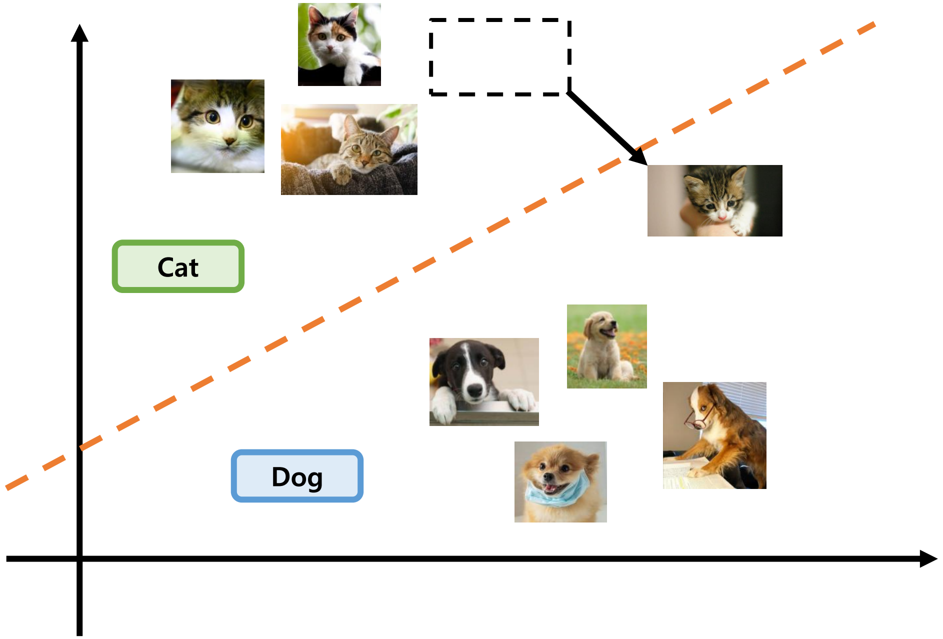
$$\delta_{t+1} = \Pi_{||\delta||_F \leq \varepsilon}(\delta_t + \alpha g(\delta_t) \,/\, ||g(\delta_t)||_F)$$

<Projected Gradient Descent>

$$\min_{\theta} \mathbb{E}_{(Z,y)\sim\mathcal{D}} \left[\max_{||\boldsymbol{\delta}|| \leq \varepsilon} \boldsymbol{L}(\boldsymbol{f_\theta}(\boldsymbol{X} + \boldsymbol{\delta}), \boldsymbol{y}) \right]$$

$$\delta_{t+1} = \Pi_{||\delta||_F \leq \varepsilon}(\delta_t + \alpha g(\delta_t) \,/\, ||g(\delta_t)||_F)$$





<Projected Gradient Descent>

$$\min_{\boldsymbol{\theta}} \mathbb{E}_{(\boldsymbol{Z},\boldsymbol{y})\sim\boldsymbol{\mathcal{D}}}\left[\max_{||\delta||\leq\varepsilon} \boldsymbol{L}(\boldsymbol{f_\theta}(\boldsymbol{X}+\boldsymbol{\delta}),\boldsymbol{y})\right]$$

$$\delta_{t+1} = \Pi_{||\delta||_F\leq\varepsilon}(\delta_t + \alpha g(\delta_t) \,/\, ||g(\delta_t)||_F)$$

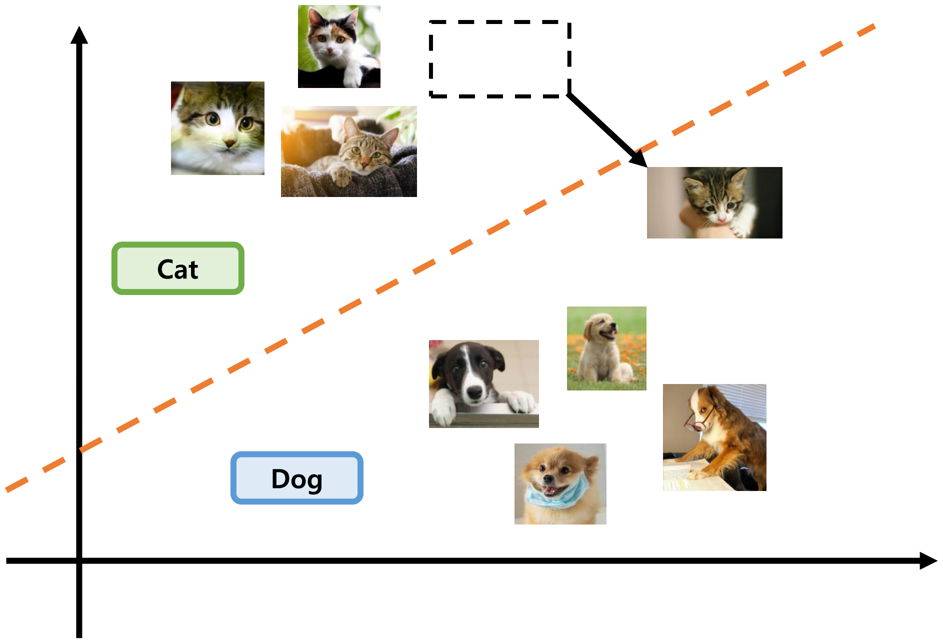
Adversarial Training for NLU

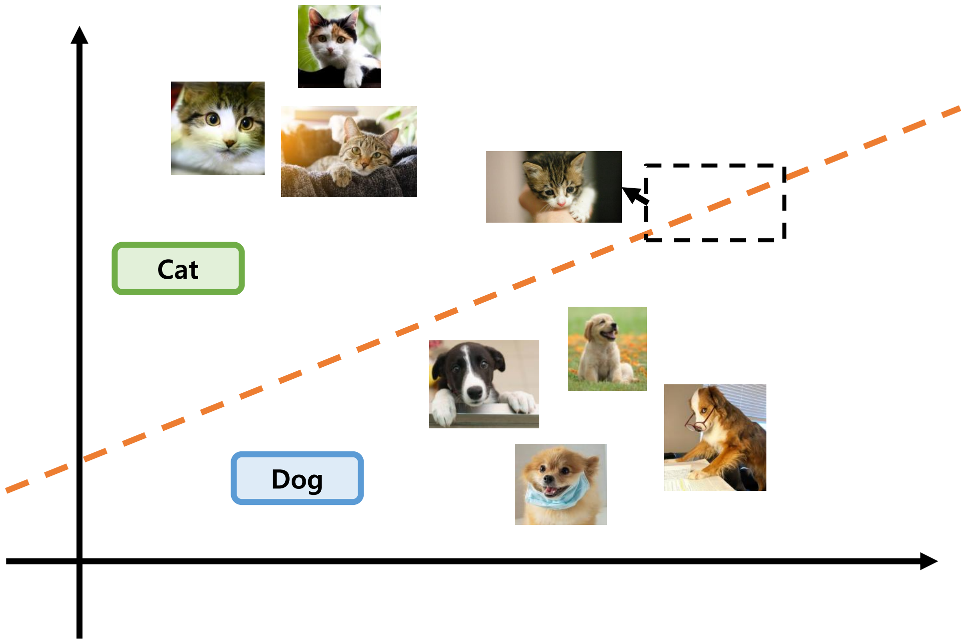
- PGD-Based Adversarial Training

<Projected Gradient Descent>

$$\min_{\boldsymbol{\theta}} \mathbb{E}_{(\boldsymbol{Z},\boldsymbol{y})\sim\mathcal{D}}\left[\max_{||\delta||\le\varepsilon} \boldsymbol{L}(\boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{X}+\boldsymbol{\delta}),\boldsymbol{y})\right]$$

$$\delta_{t+1} = \Pi_{||\delta||_F\le\varepsilon}(\delta_t + \alpha g(\delta_t) \,/\, ||g(\delta_t)||_F)$$

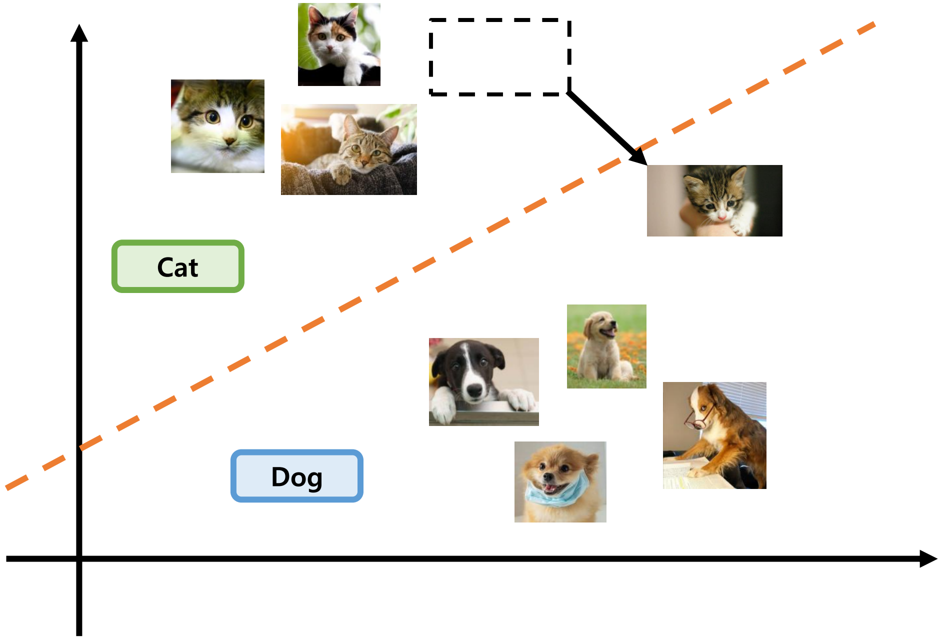




<Projected Gradient Descent>

$$\min_{\theta} \mathbb{E}_{(Z,y)\sim\mathcal{D}}\left[\max_{||\delta||\leq\varepsilon} L(f_\theta(X+\delta),y)\right]$$

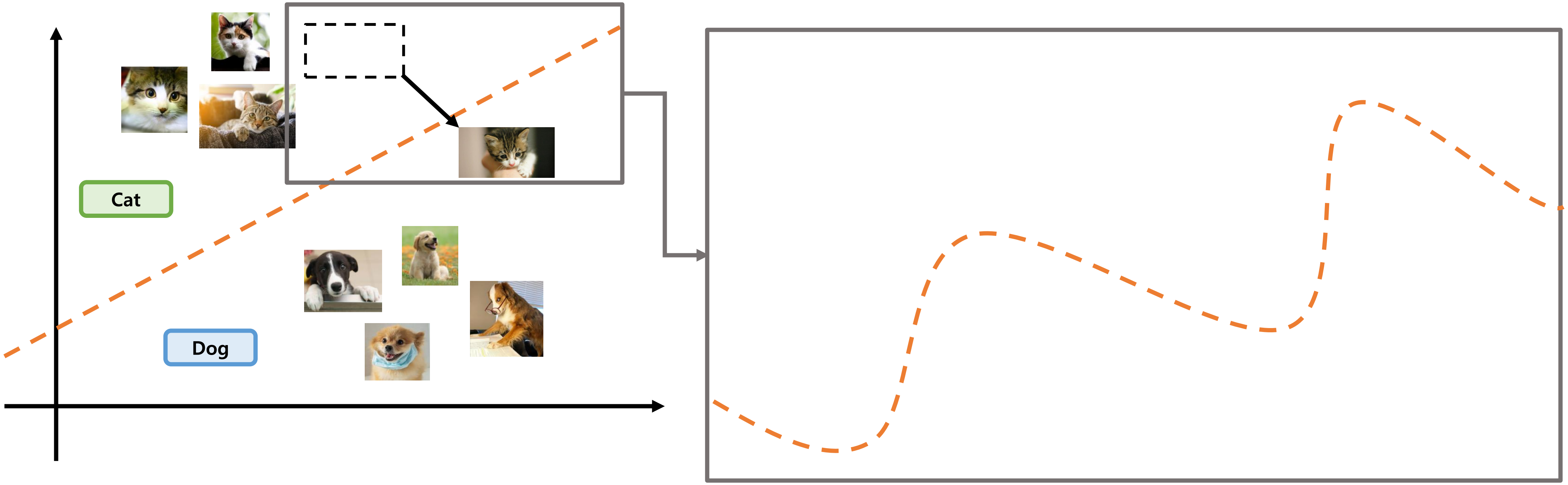
$$\delta_{t+1} = \Pi_{||\delta||_F\leq\varepsilon}(\delta_t + \alpha g(\delta_t) \,/\, ||g(\delta_t)||_F)$$



<Projected Gradient Descent>

$$\min_{\theta} \mathbb{E}_{(Z,y)\sim\mathcal{D}}\left[\max_{||\delta||\leq \varepsilon} L(f_\theta(X+\delta), y)\right]$$

$$\delta_{t+1} = \Pi_{||\delta||_F \leq \varepsilon}\left(\delta_t + \alpha g(\delta_t) / ||g(\delta_t)||_F\right)$$



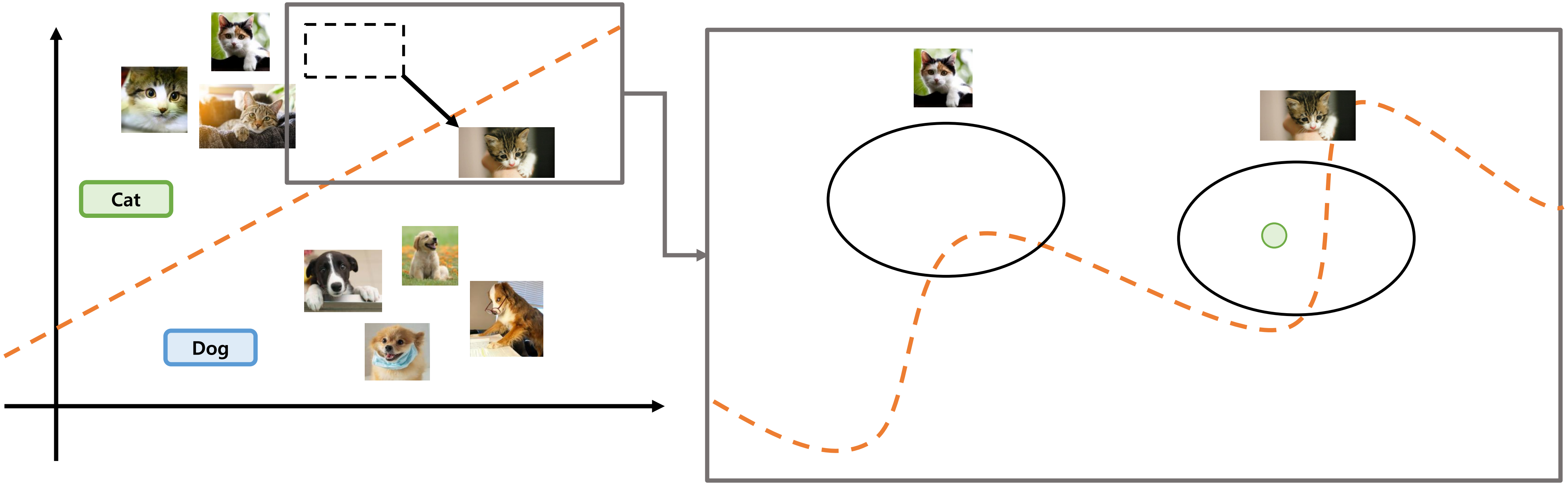
Adversarial Training for NLU

- PGD-Based Adversarial Training

<Projected Gradient Descent>

$$\min_{\theta} \mathbb{E}_{(Z,\, y)\sim\mathcal{D}}\left[\max_{||\delta|| \le \varepsilon} L(f_\theta(X+\delta), y)\right]$$

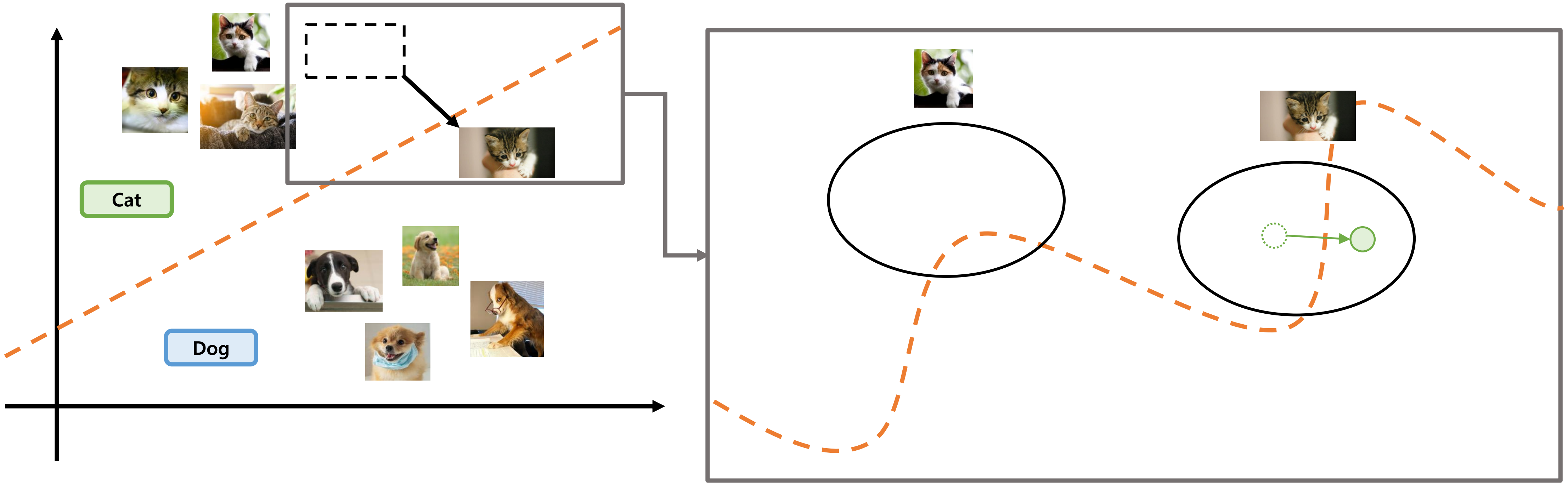
$$\boldsymbol{\delta_{t+1} = \Pi_{||\delta||_F \le \varepsilon}(\delta_t + \alpha g(\delta_t) \,/\, ||g(\delta_t)||_F)}$$



<Projected Gradient Descent>

$$\min_{\theta} \mathbb{E}_{(Z,y)\sim\mathcal{D}}\left[\max_{\|\delta\|\leq\varepsilon} L(f_\theta(X+\delta),y)\right]$$

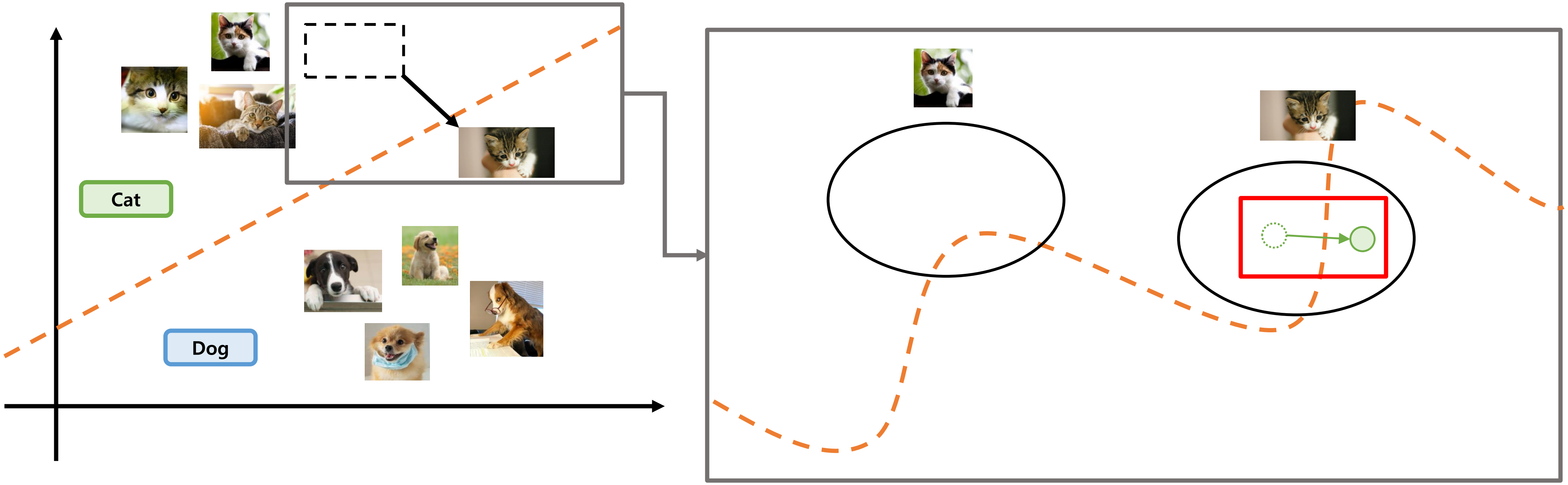
$$\delta_{t+1} = \Pi_{\|\delta\|_F\leq\varepsilon}\left(\delta_t + \alpha g(\delta_t) / \|g(\delta_t)\|_F\right)$$



<Projected Gradient Descent>

$$\min_{\theta} \mathbb{E}_{(Z,y)\sim\mathcal{D}} \left[\max_{||\delta|| \leq \varepsilon} L(f_\theta(X+\delta), y) \right]$$

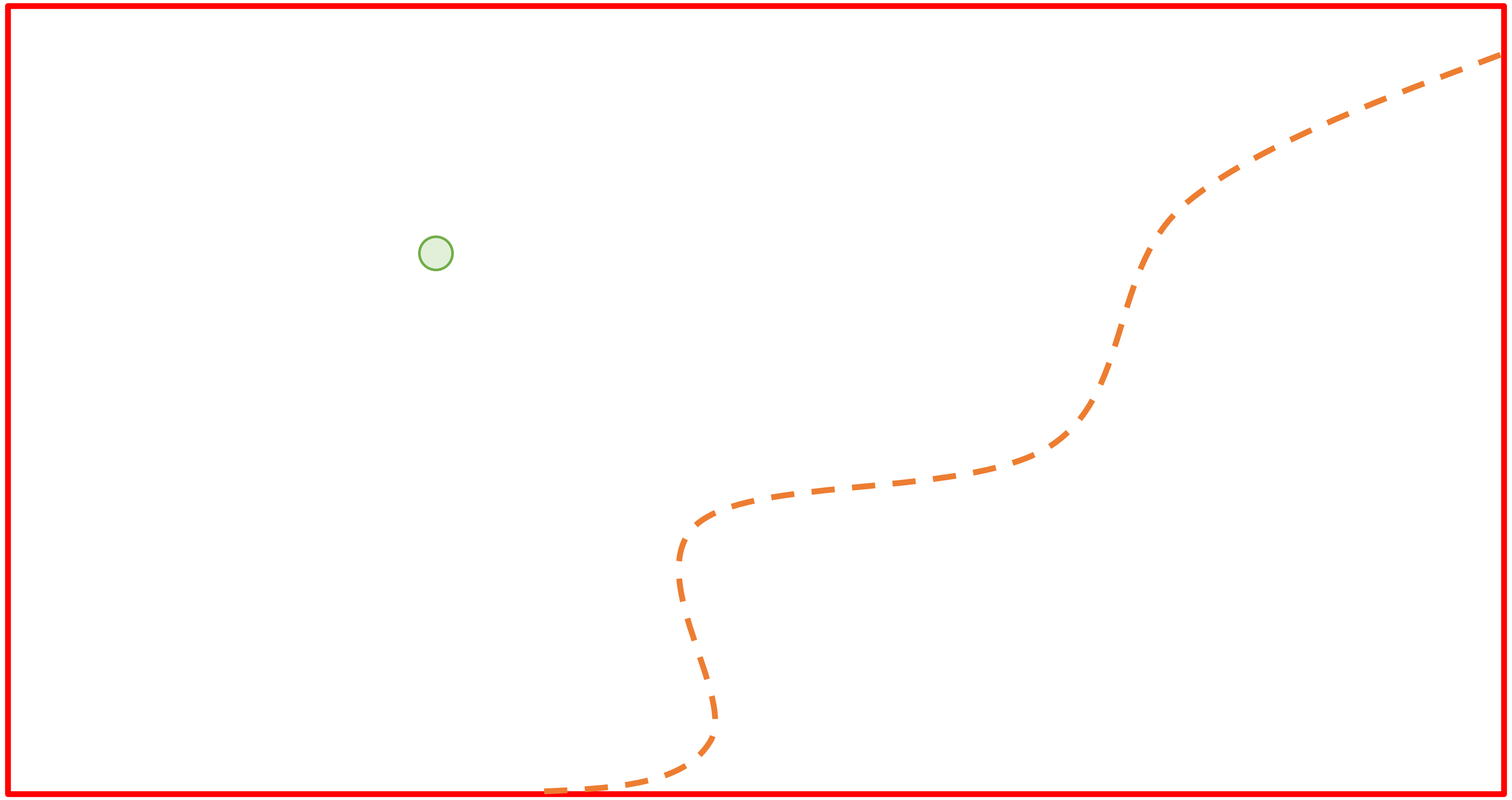
$$\boldsymbol{\delta_{t+1} = \Pi_{||\delta||_F \leq \varepsilon}(\delta_t + \alpha g(\delta_t) \, / \, ||g(\delta_t)||_F)}$$



<Projected Gradient Descent>

$$\min_{\theta} \mathbb{E}_{(Z,y)\sim\mathcal{D}}\left[\max_{||\delta||\leq\varepsilon} L(f_\theta(X+\delta),y)\right]$$

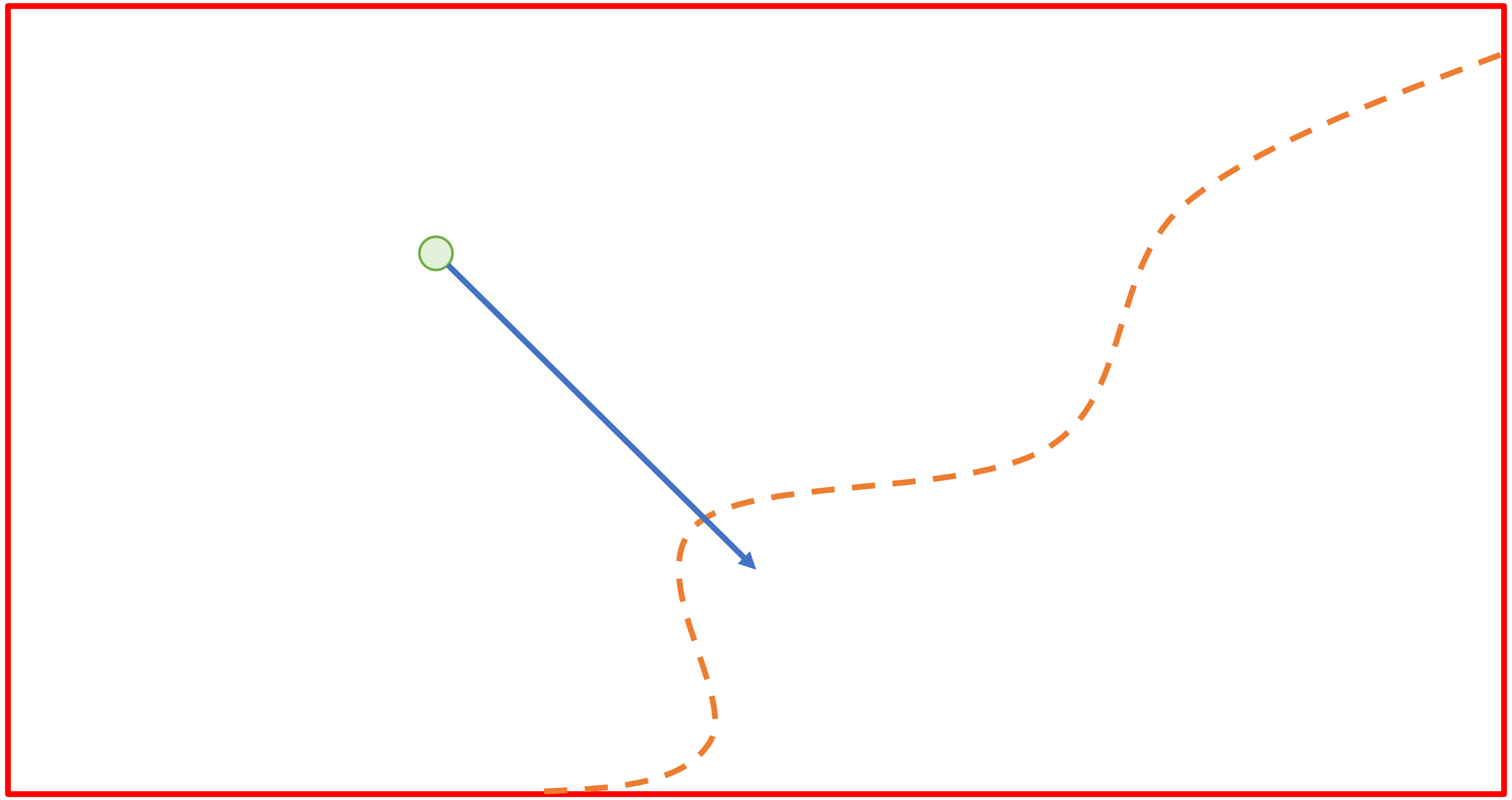
$$\boldsymbol{\delta_{t+1} = \Pi_{||\delta||_F \leq \varepsilon}(\delta_t + \alpha g(\delta_t) \,/\, ||g(\delta_t)||_F)}$$



<Projected Gradient Descent>

$$\min_{\theta} \mathbb{E}_{(Z,y)\sim\mathcal{D}}\left[\max_{||\delta||\le \epsilon} L(f_\theta(X+\delta),y)\right]$$

$$\boldsymbol{\delta}_{t+1} = \boldsymbol{\Pi}_{||\delta||_F\le\varepsilon}(\boldsymbol{\delta}_t + \boldsymbol{\alpha} \boldsymbol{g}(\boldsymbol{\delta}_t) \,/\, ||\boldsymbol{g}(\boldsymbol{\delta}_t)||_F)$$

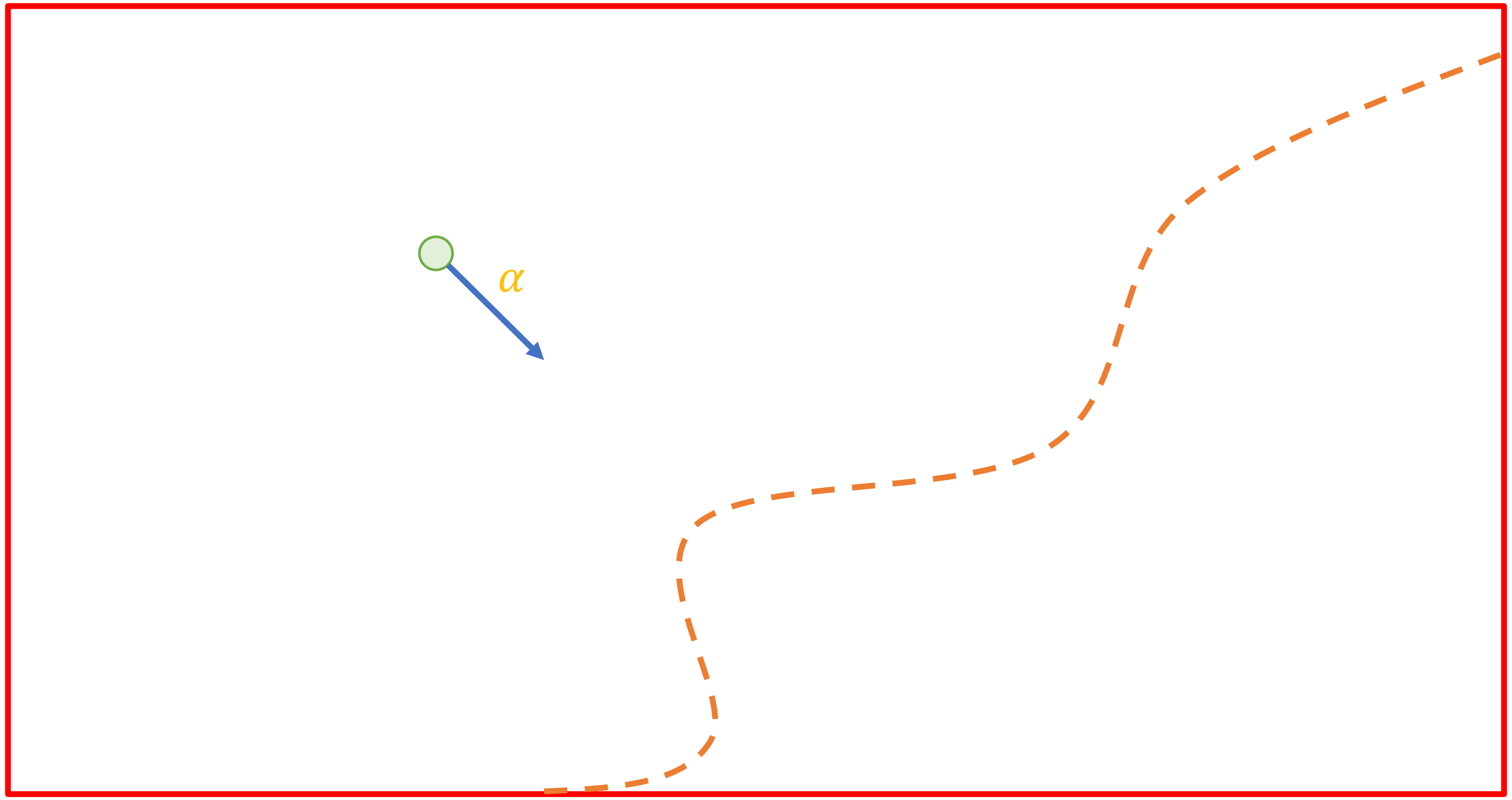


Adversarial Training for NLU
- PGD-Based Adversarial Training

<Projected Gradient Descent>

$$\min_{\theta} \mathbb{E}_{(Z,y)\sim\mathcal{D}}\left[\max_{||\delta||\leq\varepsilon} L(f_\theta(X+\delta),y)\right]$$

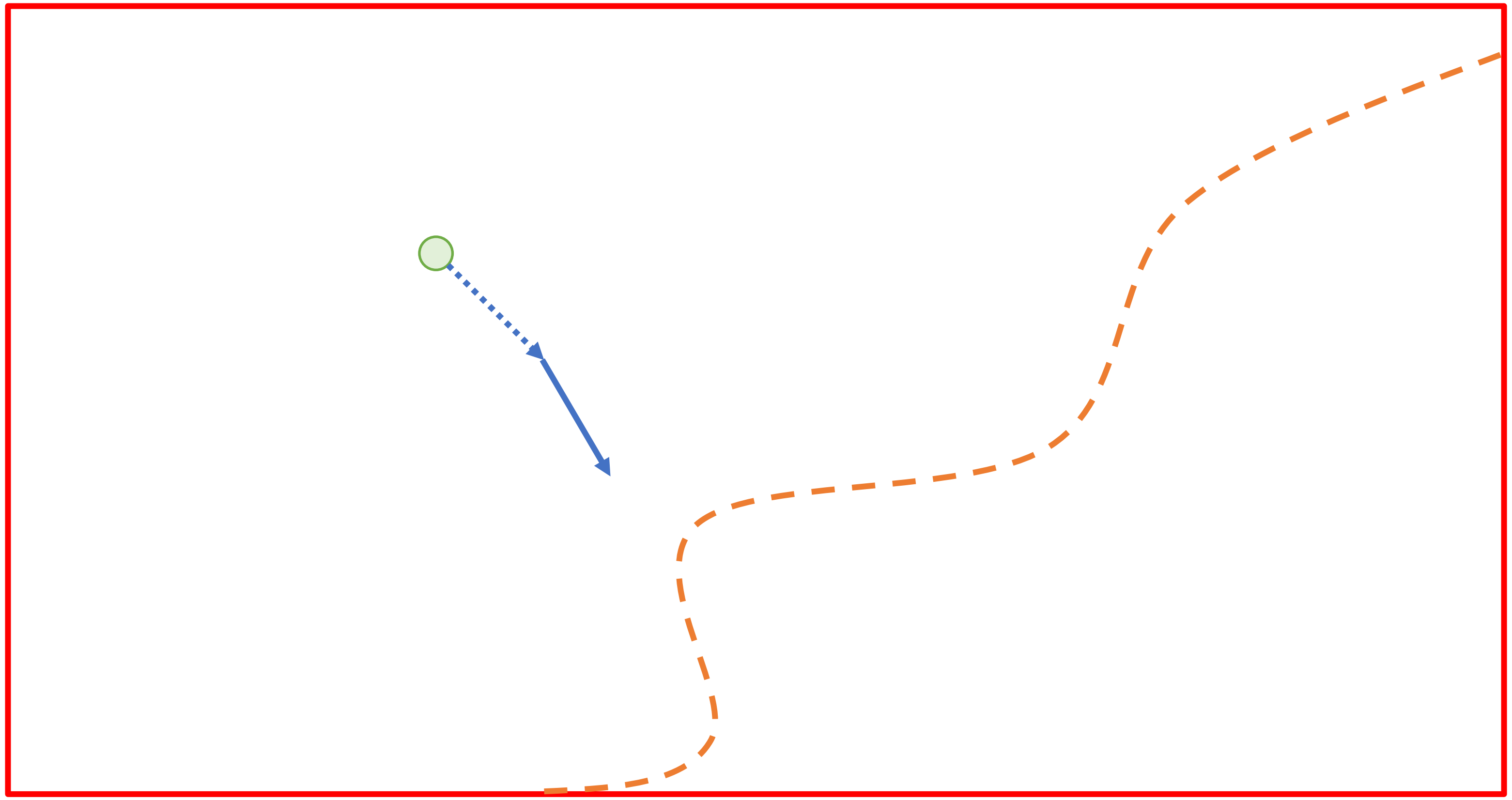
$$\boldsymbol{\delta_{t+1}} = \boldsymbol{\Pi}_{||\delta||_F\leq\varepsilon}(\boldsymbol{\delta_t} + \boldsymbol{\alpha g(\delta_t)} \,/\, ||\boldsymbol{g(\delta_t)}||_F)$$



<Projected Gradient Descent>

$$\min_{\theta} \mathbb{E}_{(Z,y)\sim\mathcal{D}} \left[\max_{||\delta|| \leq \varepsilon} L(f_\theta(X+\delta), y) \right]$$

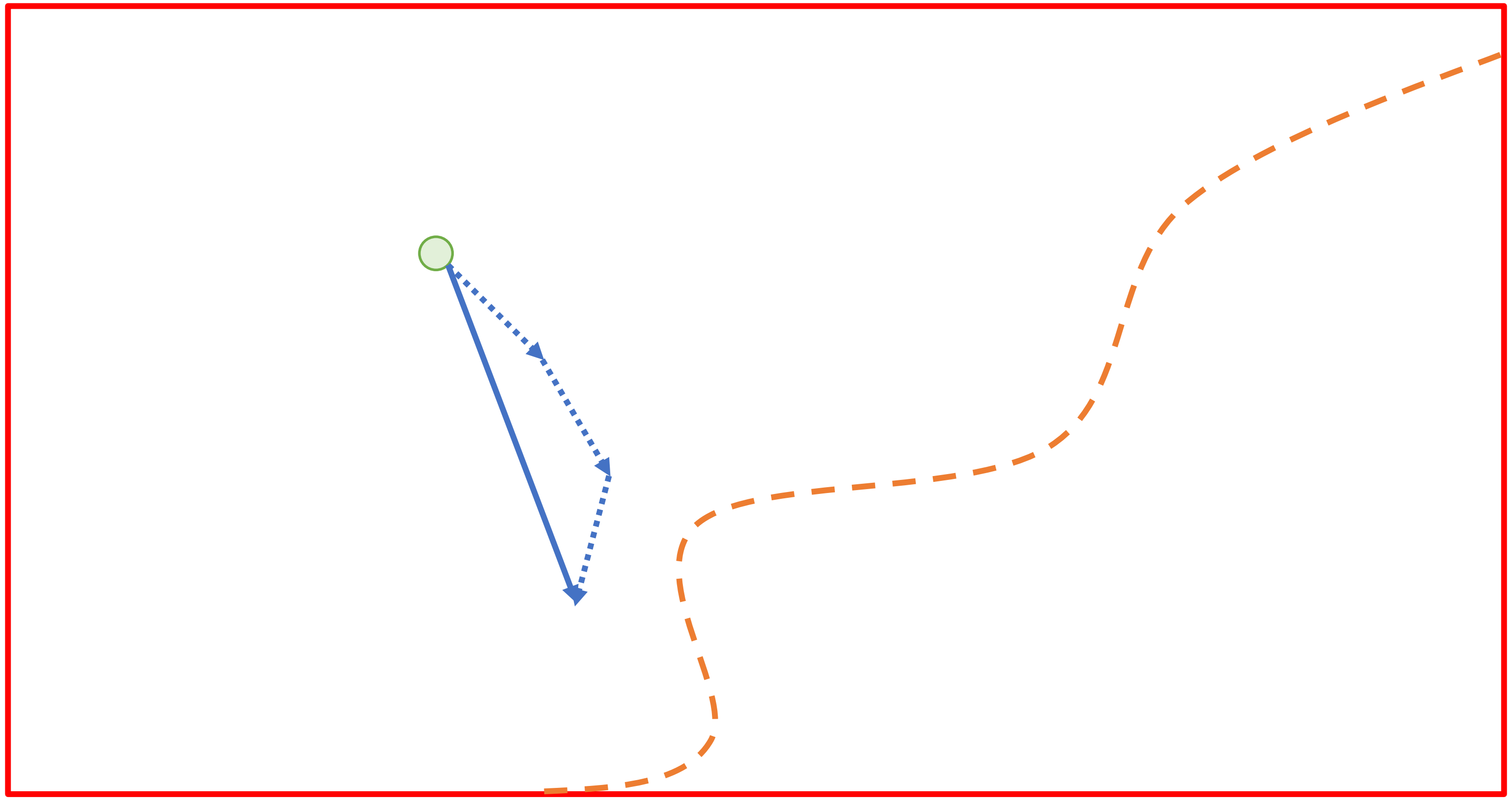
$$\delta_{t+1} = \Pi_{||\delta||_F \leq \varepsilon}(\delta_t + \alpha g(\delta_t) / ||g(\delta_t)||_F)$$



<Projected Gradient Descent>

$$\min_{\theta} \mathbb{E}_{(Z,y)\sim\mathcal{D}} \left[\max_{||\delta|| \leq \varepsilon} L(f_\theta(X+\delta), y) \right]$$

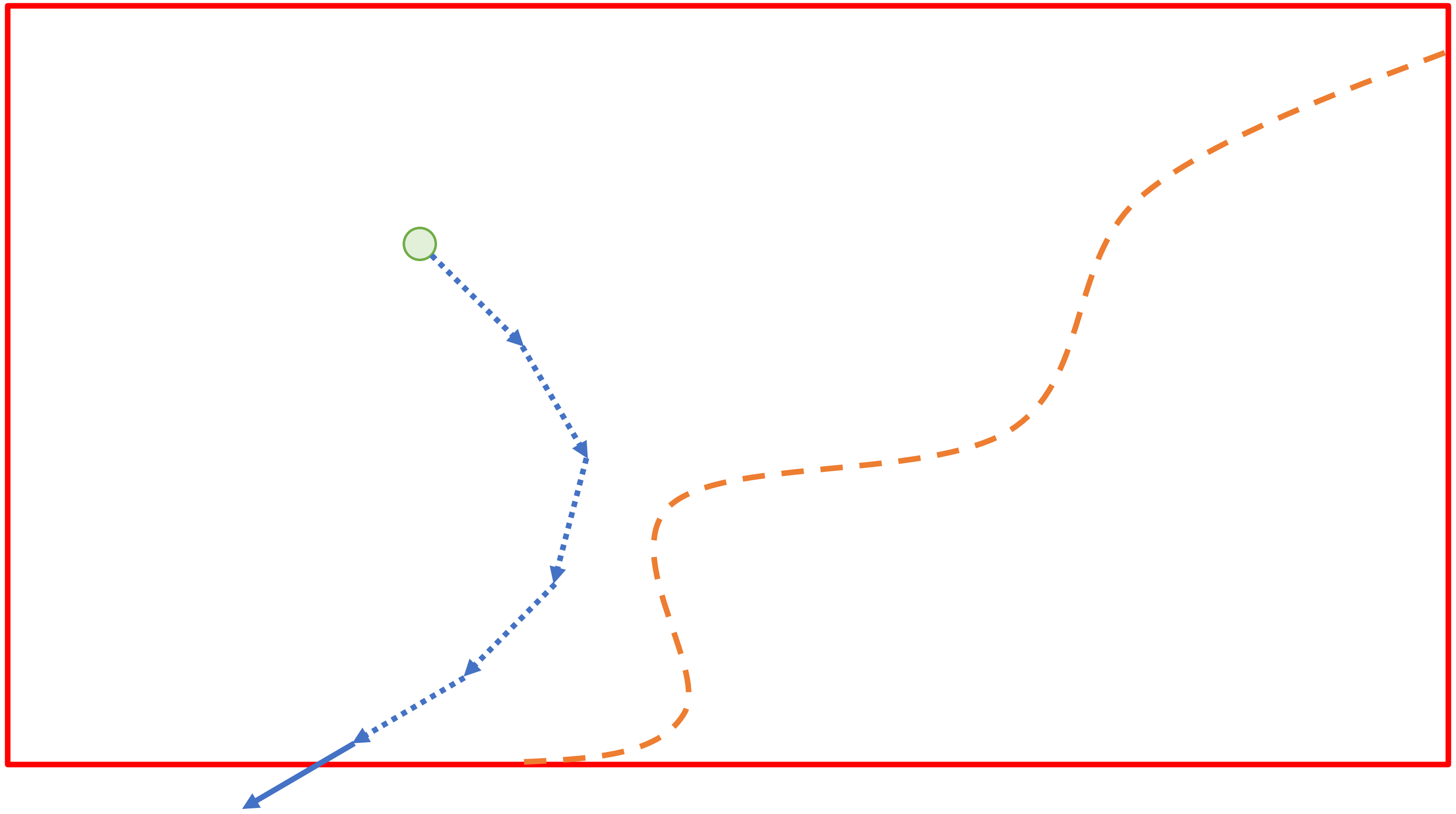
$$\delta_{t+1} = \Pi_{||\delta||_F \leq \varepsilon}(\delta_t + \alpha g(\delta_t) / ||g(\delta_t)||_F)$$



<Projected Gradient Descent>

$$\min_{\theta} \mathbb{E}_{(Z,y)\sim\mathcal{D}}\left[\max_{||\delta||\leq\varepsilon} L(f_\theta(X+\delta), y)\right]$$

$$\delta_{t+1} = \Pi_{||\delta||_F\leq\varepsilon}(\delta_t + \alpha g(\delta_t) / ||g(\delta_t)||_F)$$



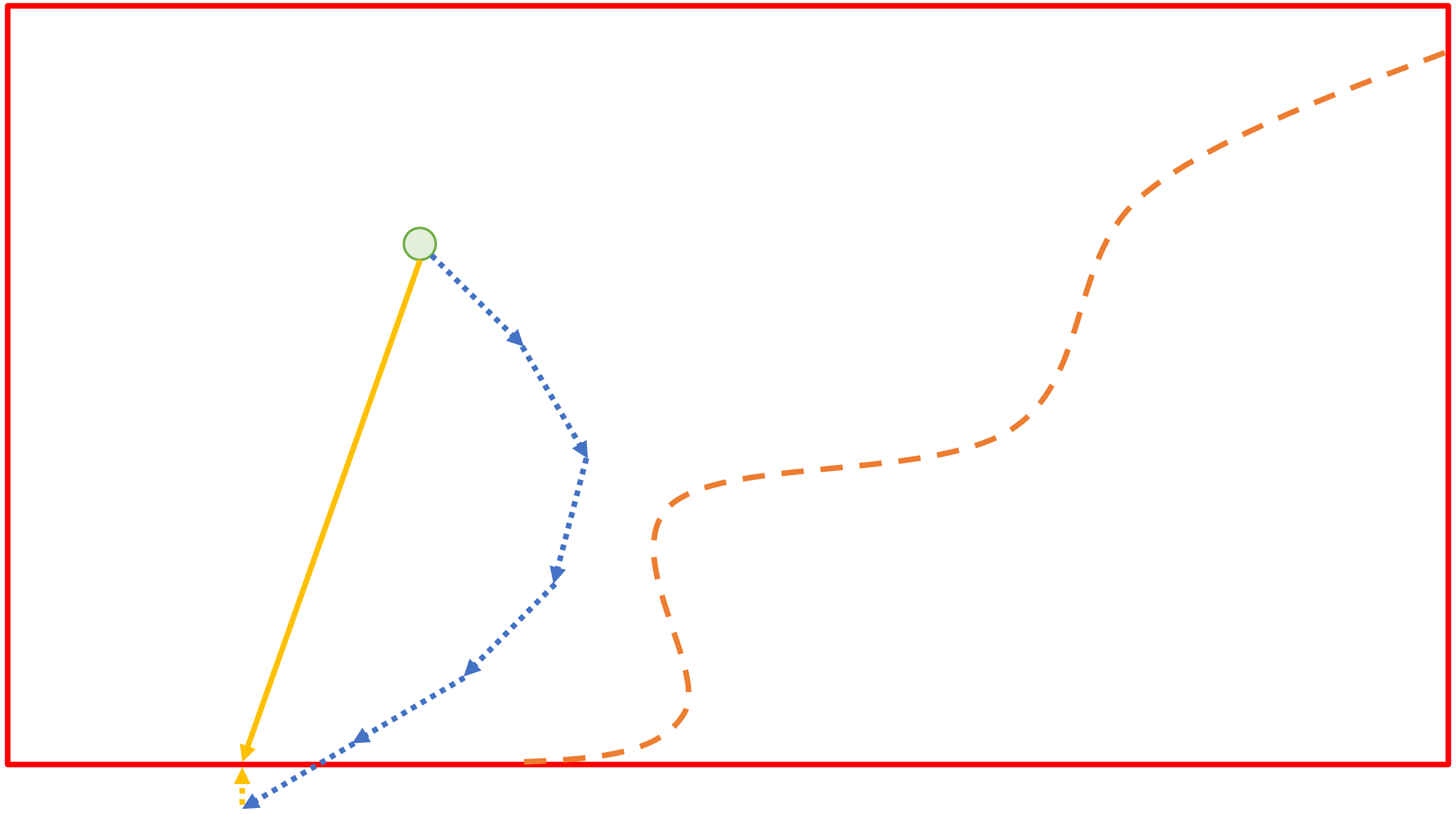
Adversarial Training for NLU

- PGD-Based Adversarial Training

<Projected Gradient Descent>

$$\min_{\theta} \mathbb{E}_{(Z,y)\sim\mathcal{D}}\left[\max_{||\delta||\le \epsilon} L(f_\theta(X+\delta),y)\right]$$

$$\delta_{t+1} = \Pi_{||\delta||_F \le \epsilon}\left(\delta_t + \alpha g(\delta_t) \,/\, ||g(\delta_t)||_F\right)$$



<FreeLB>

Algorithm	Perturbation	Model Parameter	Note
PGD	Update K-Steps	Update 1-Step	• Update Perturbation K-Times and Update Model Parameter 1-Time
FreeAT	Update 1-Step	Update 1-Step	• Update Perturbation 1-Time and Update Model Parameter 1-Time
YOPO	Update K-Steps	Update 1-Step	• Accumulate Gradients for Model Parameters While Updating Perturbation K-Times and Update Model Parameters 1-Time Using Accumulated Gradients

Adversarial Training for NLU

- Large-Batch Adversarial Training for Free

<FreeLB>

Algorithm 1 "Free" Large-Batch Adversarial Training (FreeLB-K)

Require: Training samples $X = \{(\boldsymbol{Z}, y)\}$, perturbation bound ϵ, learning rate τ, ascent steps K, ascent step size α

Initialize $\boldsymbol{\theta}$

for epoch $= 1 \dots N_{ep}$ **do**

  for minibatch $B \subset X$ **do**

    $\boldsymbol{\delta}_0 \leftarrow \frac{1}{\sqrt{N_{\boldsymbol{\delta}}}} U(-\epsilon, \epsilon)$

    $\boldsymbol{g}_0 \leftarrow 0$

    for $t = 1 \dots K$ **do**

      Accumulate gradient of parameters θ

        $\boldsymbol{g}_t \leftarrow \boldsymbol{g}_{t-1} + \frac{1}{K} \mathbb{E}_{(\boldsymbol{Z}, y) \in B} [\nabla_{\boldsymbol{\theta}} L(f_{\boldsymbol{\theta}}(\boldsymbol{X} + \boldsymbol{\delta}_{t-1}), y)]$

      Update the perturbation δ via gradient ascend

        $\boldsymbol{g}_{adv} \leftarrow \nabla_{\boldsymbol{\delta}} L(f_{\boldsymbol{\theta}}(\boldsymbol{X} + \boldsymbol{\delta}_{t-1}), y)$

        $\boldsymbol{\delta}_t \leftarrow \Pi_{\|\boldsymbol{\delta}\|_F \le \epsilon}(\boldsymbol{\delta}_{t-1} + \alpha \cdot \boldsymbol{g}_{adv} / \|\boldsymbol{g}_{adv}\|_F)$

    end for

    $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \tau \boldsymbol{g}_K$

  end for

end for

Adversarial Training for NLU

- Large-Batch Adversarial Training for Free

<FreeLB>

Algorithm 1 "Free" Large-Batch Adversarial Training (FreeLB-K)

Require: Training samples $X = \{(\boldsymbol{Z}, y)\}$, perturbation bound ϵ, learning rate τ, ascent steps K, ascent step size α

Initialize $\boldsymbol{\theta}$

for epoch $= 1 \ldots N_{ep}$ **do**

  for minibatch $B \subset X$ **do**

    $\boldsymbol{\delta}_0 \leftarrow \frac{1}{\sqrt{N_\delta}} U(-\epsilon, \epsilon)$

    $\boldsymbol{g}_0 \leftarrow 0$

    for $t = 1 \ldots K$ **do**

      Accumulate gradient of parameters θ

        $\boldsymbol{g}_t \leftarrow \boldsymbol{g}_{t-1} + \frac{1}{K}\mathbb{E}_{(\boldsymbol{Z},y)\in B}[\nabla_{\boldsymbol{\theta}} L(f_{\boldsymbol{\theta}}(\boldsymbol{X} + \boldsymbol{\delta}_{t-1}), y)]$

      Update the perturbation δ via gradient ascend

        $\boldsymbol{g}_{adv} \leftarrow \nabla_{\boldsymbol{\delta}} L(f_{\boldsymbol{\theta}}(\boldsymbol{X} + \boldsymbol{\delta}_{t-1}), y)$

        $\boldsymbol{\delta}_t \leftarrow \Pi_{\|\boldsymbol{\delta}\|_F \leq \epsilon}(\boldsymbol{\delta}_{t-1} + \alpha \cdot \boldsymbol{g}_{adv}/\|\boldsymbol{g}_{adv}\|_F)$

    end for

    $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \tau \boldsymbol{g}_K$

  end for

end for

Experiments

- GLUE Benchmark
- Comparing the Robustness

<GLUE Benchmark>

Method	MNLI (Acc)	QNLI (Acc)	QQP (Acc)	RTE (Acc)	SST-2 (Acc)	MRPC (Acc)	CoLA (Mcc)	STS-B (Pearson)
Reported	90.20	94.70	92.20	86.60	96.40	90.90	68.00	92.40
ReImp	-	-	-	85.61 (1.7)	96.56 (.3)	90.69 (.5)	67.57 (1.3)	92.20 (.2)
PGD	90.53 (.2)	94.87 (.2)	92.49 (.07)	87.41 (.9)	96.44 (.1)	90.93 (.2)	69.67 (1.2)	92.43 (7.)
FreeAT	90.02 (.2)	94.66 (.2)	92.48 (.08)	86.69 (15.)	96.10 (.2)	90.69 (.4)	68.80 (1.3)	92.40 (.3)
FreeLB	**90.61** (.1)	**94.98** (.2)	**92.60** (03)	**88.13** (1.2)	**96.79** (.2)	**91.41** (.7)	**71.12** (.9)	**92.67** (.08)

<Results (Median and Variance) on the dev sets of GLUE based on the RoBERTa-Large Model>

<GLUE Benchmark>

Method	MNLI (Acc)	QNLI (Acc)	QQP (Acc)	RTE (Acc)	SST-2 (Acc)	MRPC (Acc)	CoLA (Mcc)	STS-B (Pearson)
Reported	90.20	94.70	92.20	86.60	96.40	90.90	68.00	92.40
ReImp	-	-	-	85.61 (1.7)	96.56 (.3)	90.69 (.5)	67.57 (1.3)	92.20 (.2)
PGD	90.53 (.2)	94.87 (.2)	92.49 (.07)	87.41 (.9)	96.44 (.1)	90.93 (.2)	69.67 (1.2)	92.43 (7.)
FreeAT	90.02 (.2)	94.66 (.2)	92.48 (.08)	86.69 (15.)	96.10 (.2)	90.69 (.4)	68.80 (1.3)	92.40 (.3)
FreeLB	**90.61** (.1)	**94.98** (.2)	**92.60** (03)	**88.13** (1.2)	**96.79** (.2)	**91.41** (.7)	**71.12** (.9)	**92.67** (.08)

Results (Median and Variance) on the dev sets of GLUE based on the RoBERTa-Large Model

<GLUE Benchmark>

Method	MNLI (Acc)	QNLI (Acc)	QQP (Acc)	RTE (Acc)	SST-2 (Acc)	MRPC (Acc)	CoLA (Mcc)	STS-B (Pearson)
Reported	90.20	94.70	92.20	86.60	96.40	90.90	68.00	92.40
ReImp	-	-	-	85.61 (1.7)	96.56 (.3)	90.69 (.5)	67.57 (1.3)	92.20 (.2)
PGD	90.53 (.2)	94.87 (.2)	92.49 (.07)	87.41 (.9)	96.44 (.1)	90.93 (.2)	69.67 (1.2)	92.43 (7.)
FreeAT	90.02 (.2)	94.66 (.2)	92.48 (.08)	86.69 (15.)	96.10 (.2)	90.69 (.4)	68.80 (1.3)	92.40 (.3)
FreeLB	**90.61** (.1)	**94.98** (.2)	**92.60** (03)	**88.13** (1.2)	**96.79** (.2)	**91.41** (.7)	**71.12** (.9)	**92.67** (.08)

Results (Median and Variance) on the dev sets of GLUE based on the RoBERTa-Large Model

<GLUE Benchmark>

Model	Score	CoLA 8.5k	SST-2 67k	MRPC 3.7k	STS-B 7k	QQP 364k	MNLI-m/mm 393k	QNLI 108k	RTE 2.5k	WNLI 634	AX
BERT-base	78.3	52.1	93.5	88.9/88.4	87.1/85.8	71.2/89.2	94.6/83.4	90.5	66.4	65.1	64.2
FreeLB-BERT	79.4	54.5	93.6	88.1/83.5	87.7/86.7	72.7/89.6	85.7/84.6	91.8	70.1	65.1	36.9
MT-DNN	87.6	**68.4**	96.5	92.7/90.3	91.1/90.7	73.7/89.9	87.9/87.4	96.0	86.3	89.0	42.8
XLNet-Large	88.4	67.8	**96.8**	93.0/90.7	91.6/91.1	74.2/90.3	90.2/89.8	98.6	86.3	**90.4**	47.5
RoBERTa	88.5	67.8	96.7	92.3/89.8	92.2/91.9	74.3/90.2	90.8/90.2	**98.9**	88.2	89.0	47.7
FreeLB-RoB	**88.8**	68.0	**96.8**	**93.1/90.8**	**92.4/92.2**	**74.8/90.3**	**91.1/90.7**	98.8	**88.7**	89.0	**50.1**
Human	87.1	66.4	97.8	86.3/80.8	92.7/92.6	59.5/80.4	92.0/92.8	91.2	93.6	95.9	-

Results on GLUE from the Evaluation Server, as of Sep 25, 2019

<GLUE Benchmark>

Model	Score	CoLA 8.5k	SST-2 67k	MRPC 3.7k	STS-B 7k	QQP 364k	MNLI-m/mm 393k	QNLI 108k	RTE 2.5k	WNLI 634	AX
BERT-base	78.3	52.1	93.5	88.9/88.4	87.1/85.8	71.2/89.2	94.6/83.4	90.5	66.4	65.1	64.2
FreeLB-BERT	79.4	54.5	93.6	88.1/83.5	87.7/86.7	72.7/89.6	85.7/84.6	91.8	70.1	65.1	36.9
MT-DNN	87.6	**68.4**	96.5	92.7/90.3	91.1/90.7	73.7/89.9	87.9/87.4	96.0	86.3	89.0	42.8
XLNet-Large	88.4	67.8	**96.8**	93.0/90.7	91.6/91.1	74.2/90.3	90.2/89.8	98.6	86.3	**90.4**	47.5
RoBERTa	88.5	67.8	96.7	92.3/89.8	92.2/91.9	74.3/90.2	90.8/90.2	**98.9**	88.2	89.0	47.7
FreeLB-RoB	**88.8**	68.0	**96.8**	**93.1/90.8**	**92.4/92.2**	**74.8/90.3**	**91.1/90.7**	98.8	**88.7**	89.0	**50.1**
Human	87.1	66.4	97.8	86.3/80.8	92.7/92.6	59.5/80.4	92.0/92.8	91.2	93.6	95.9	-

Results on GLUE from the Evaluation Server, as of Sep 25, 2019

<GLUE Benchmark>

Model	Score	CoLA 8.5k	SST-2 67k	MRPC 3.7k	STS-B 7k	QQP 364k	MNLI-m/mm 393k	QNLI 108k	RTE 2.5k	WNLI 634	AX
BERT-base	78.3	52.1	93.5	88.9/88.4	87.1/85.8	71.2/89.2	94.6/83.4	90.5	66.4	65.1	64.2
FreeLB-BERT	79.4	54.5	93.6	88.1/83.5	87.7/86.7	72.7/89.6	85.7/84.6	91.8	70.1	65.1	36.9
MT-DNN	87.6	**68.4**	96.5	92.7/90.3	91.1/90.7	73.7/89.9	87.9/87.4	96.0	86.3	89.0	42.8
XLNet-Large	88.4	67.8	**96.8**	93.0/90.7	91.6/91.1	74.2/90.3	90.2/89.8	98.6	86.3	**90.4**	47.5
RoBERTa	88.5	67.8	96.7	92.3/89.8	92.2/91.9	74.3/90.2	90.8/90.2	**98.9**	88.2	89.0	47.7
FreeLB-RoB	**88.8**	68.0	**96.8**	**93.1/90.8**	**92.4/92.2**	**74.8/90.3**	**91.1/90.7**	98.8	**88.7**	89.0	**50.1**
Human	87.1	66.4	97.8	86.3/80.8	92.7/92.6	59.5/80.4	92.0/92.8	91.2	93.6	95.9	-

Results on GLUE from the Evaluation Server, as of Sep 25, 2019

<GLUE Benchmark>

Model	Score	CoLA 8.5k	SST-2 67k	MRPC 3.7k	STS-B 7k	QQP 364k	MNLI-m/mm 393k	QNLI 108k	RTE 2.5k	WNLI 634	AX
BERT-base	78.3	52.1	93.5	88.9/88.4	87.1/85.8	71.2/89.2	94.6/83.4	90.5	66.4	65.1	64.2
FreeLB-BERT	79.4	54.5	93.6	88.1/83.5	87.7/86.7	72.7/89.6	85.7/84.6	91.8	70.1	65.1	36.9
MT-DNN	87.6	**68.4**	96.5	92.7/90.3	91.1/90.7	73.7/89.9	87.9/87.4	96.0	86.3	89.0	42.8
XLNet-Large	88.4	67.8	**96.8**	93.0/90.7	91.6/91.1	74.2/90.3	90.2/89.8	98.6	86.3	**90.4**	47.5
RoBERTa	88.5	67.8	96.7	92.3/89.8	92.2/91.9	74.3/90.2	90.8/90.2	**98.9**	88.2	89.0	47.7
FreeLB-RoB	**88.8**	68.0	**96.8**	**93.1/90.8**	**92.4/92.2**	**74.8/90.3**	**91.1/90.7**	98.8	**88.7**	89.0	**50.1**
Human	87.1	66.4	97.8	86.3/80.8	92.7/92.6	59.5/80.4	92.0/92.8	91.2	93.6	95.9	-

Results on GLUE from the Evaluation Server, as of Sep 25, 2019

<Comparing the Robustness>

Methods	RTE			CoLA			MRPC		
	M-Inc (10^{-4})	M-Inc (R) (10^{-4})	N-Loss (10^{-4})	M-Inc (10^{-4})	M-Inc (R) (10^{-4})	N-Loss (10^{-4})	M-Inc (10^{-3})	M-Inc (R) (10^{-3})	N-Loss (10^{-3})
Vanilla	5.1	5.3	4.5	6.1	5.7	5.2	10.2	10.2	1.9
PGD	4.7	4.9	6.2	128.2	130.1	436.1	5.7	5.7	5.4
FreeLB	3.0	2.6	4.1	1.4	1.3	7.2	3.6	3.6	2.7

Median of the Maximum Increase in Loss in the Vicinity of the Dev Set Samples for RoBERTa-Large Model Finetuned with Different Methods

$$\Delta L_{max}(X,\epsilon) = \max_{||\delta|| \leq \epsilon} L(f_\theta(X+\delta), y) - L(f_\theta(X), y)$$

M-Inc	FreeLB
M-Inc (R)	PGD
N-Loss	Clean Sample

<Comparing the Robustness>

Methods	RTE			CoLA			MRPC		
	M-Inc (10^{-4})	M-Inc (R) (10^{-4})	N-Loss (10^{-4})	M-Inc (10^{-4})	M-Inc (R) (10^{-4})	N-Loss (10^{-4})	M-Inc (10^{-3})	M-Inc (R) (10^{-3})	N-Loss (10^{-3})
Vanilla	5.1	5.3	4.5	6.1	5.7	5.2	10.2	10.2	1.9
PGD	4.7	4.9	6.2	128.2	130.1	436.1	5.7	5.7	5.4
FreeLB	3.0	2.6	4.1	1.4	1.3	7.2	3.6	3.6	2.7

Median of the Maximum Increase in Loss in the Vicinity of the Dev Set Samples for RoBERTa-Large Model Finetuned with Different Methods

$$\Delta L_{max}(X,\epsilon) = \max_{||\delta|| \leq \epsilon} L(f_\theta(X+\delta), y) - L(f_\theta(X), y)$$

M-Inc	FreeLB
M-Inc (R)	PGD
N-Loss	Clean Sample

Conclusion

<Conclusion>

- Proposed a novel adversarial training algorithm, FreeLB, that promotes higher invariance in the embedding space
- Applied FreeLB to Transformer-based models for natural language understanding and achieved new state-of-the-art on GLUE benchmark
- FreeLB resulted in both higher robustness in the embedding space than natural training and better generalization ability

Any Questions?

Thank You