NEURAL INFORMATION PROCESSING SYSTEMS

Large-Scale Adversarial Training for Vision-and-Language Representation Learning

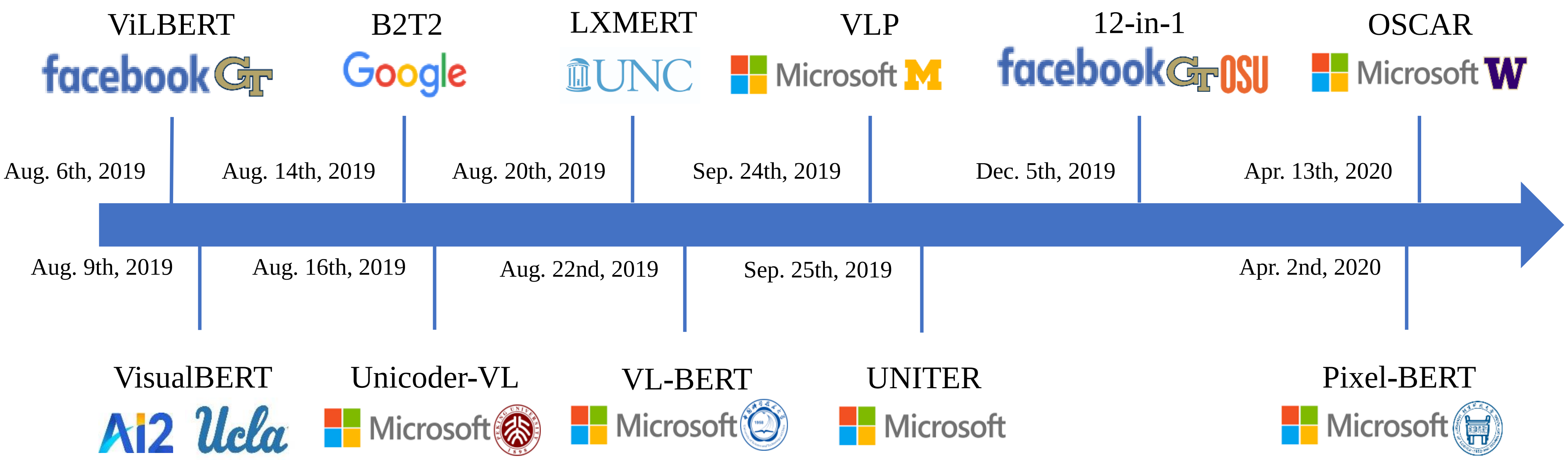
Zhe Gan, Yen-Chun Chen, Linjie Li , Chen Zhu, Yu Cheng, Jingjing Liu

Microsoft

UNIVERSITY OF MARYLAND

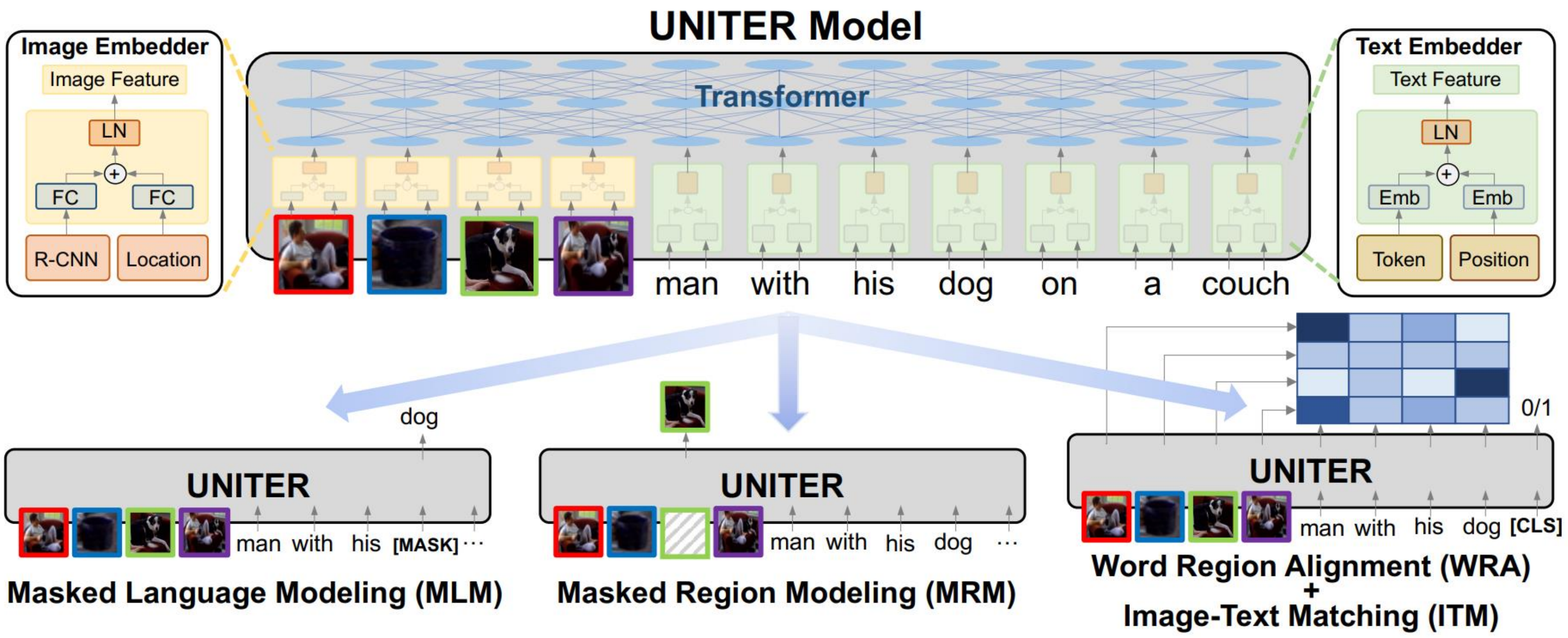
Image-Text Pre-training

- Tremendous progress has been made for multimodal pre-training



Recap on UNITER

- Pre-training a large-scale Transformer for universal V+L representation learning



What's Next?

- Aggressive finetuning often falls into the overfitting trap in existing multimodal pre-training methods
- Adversarial training (FreeLB) has shown great potential in improving the generalization ability of BERT
- Beyond FreeLB:
  - How about pre-training?
  - How about image modality?
  - How about AT algorithm itself?

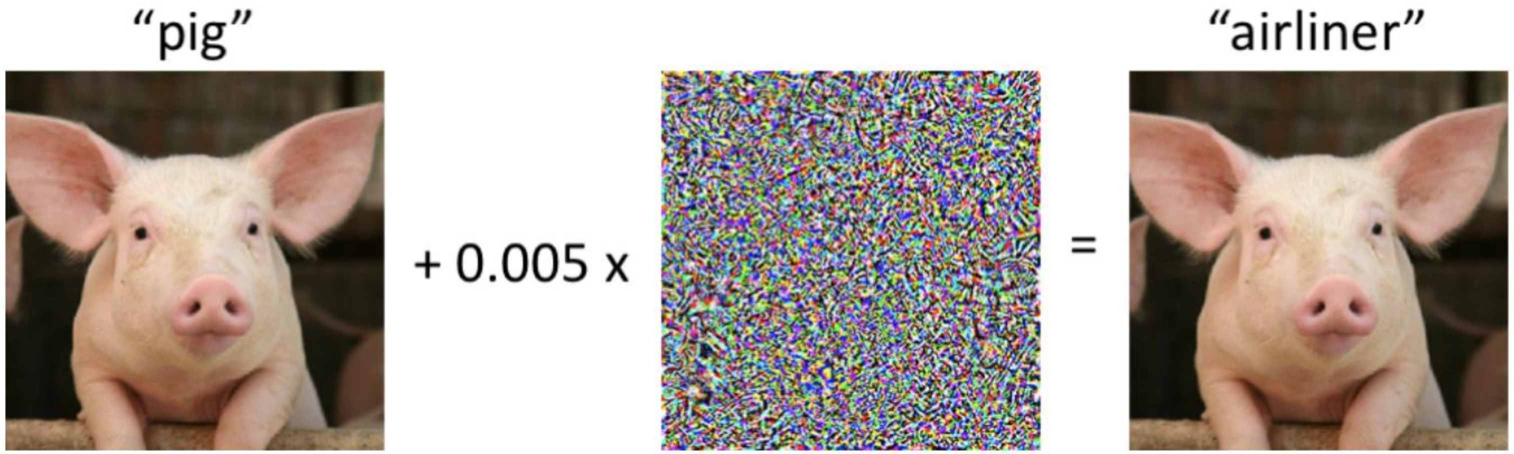
FreeLB: Enhanced Adversarial Training for Natural Language Understanding, ICLR 2020

VILLA: Vision-and-Language Large-scale Adversarial Training

Preliminary: What's Adversarial Attack?

- Neural Networks are prone to label-preserving adversarial examples

Computer Vision:



Natural Language Processing:

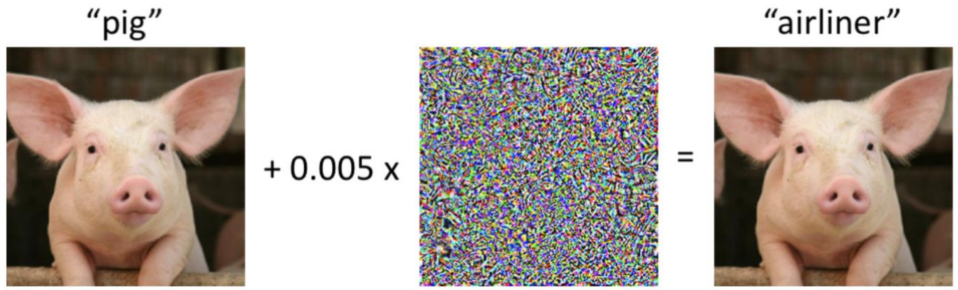
Original: What is the oncorhynchus also called? **A:** chum salmon	**Original:** How long is the Rhine? **A:** 1,230 km
Changed: What's the oncorhynchus also called? **A:** keta	**Changed:** How long is the Rhine?? **A:** more than 1,050,000
(b) Example for (*WP is*→*WP's*)	(c) Example for (*?*→*??*)

[1] Explaining and harnessing adversarial examples. *arXiv:1412.6572*
[2] Semantically equivalent adversarial rules for debugging nlp models. *ACL (2018)*

Preliminary: What’s Adversarial Training (AT)?

- A min-max game to harness adversarial examples

$$\min_{\theta} \mathop{\mathbb{E}}_{(x,y)\sim\widehat{\mathcal{D}}} \left[\max_{\delta \in S} \mathcal{L}(x+\delta, y; \theta) \right]$$

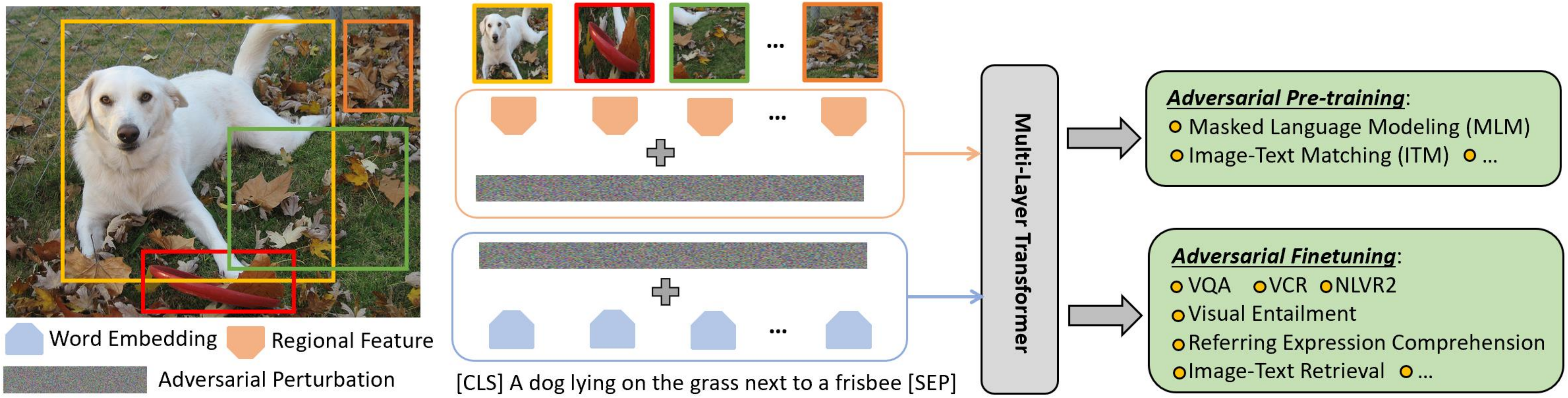


- Use adversarial examples as additional training samples
  - On one hand, we try to find perturbations that maximize the empirical risk
  - On the other hand, the model tries to make correct predictions on adversarial examples
- *What doesn't kill you makes you stronger!*

Explaining and harnessing adversarial examples. *arXiv:1412.6572*

What's Our Recipe?

- Ingredient #1: Adversarial pre-training + finetuning
- Ingredient #2: Perturbations in the embedding space
- Ingredient #3: Enhanced adversarial training algorithm



#1: Adversarial Pre-training + Finetuning

- Pre-training and finetuning are inherently corelated

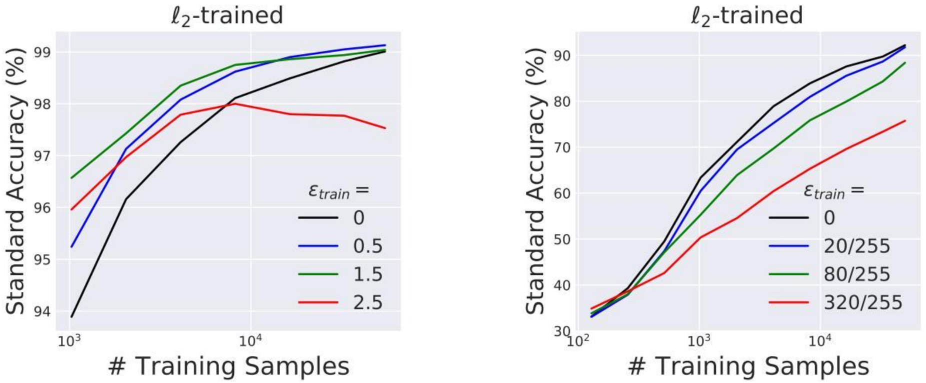
- MLM during pre-training *(masking out an object)*:
  [CLS] A [MASK] lying on the grass next to a frisbee [SEP]
- VQA during finetuning *(asking about an object)*:
  What animal is lying on the grass?

- Pre-training and finetuning share the same mathematical formulation

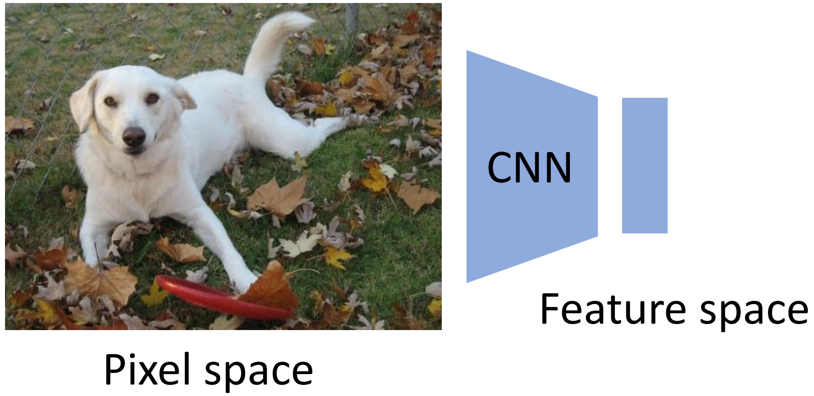
$$\min_{\boldsymbol{\theta}} \mathbb{E}_{(\boldsymbol{x}_{img}, \boldsymbol{x}_{txt}, \boldsymbol{y}) \sim \mathcal{D}} [L(f_{\boldsymbol{\theta}}(\boldsymbol{x}_{img}, \boldsymbol{x}_{txt}), \boldsymbol{y})] \, .$$

#2: Perturbations in the Embedding Space

- For image, robustness is often at odds with generalization
  - Generalization: Accuracy on clean data
  - Robustness: Accuracy on adversarial examples



(a) MNIST (b) CIFAR-10



- To boost performance on clean data, we propose to add perturbation in the feature space instead of pixel space

Robustness may be at odds with accuracy. *ICLR (2019)*.

#2: Perturbations in the Embedding Space

- For text, generating actual adversarial examples is difficult
  - An adversarial example should *preserve the semantics* as context is important

Original: He has a natural gift for writing scripts.

Adversarial: He has a natural talent for writing scripts. ✓

Adversarial: He has a natural present for writing scripts. ✗

  - Use back-translation scores to filter out invalid adversaries: *expensive*
  - Searching for semantically equivalent adversarial rules: *heuristic*
- Since we only care about the *end results* of adversarial training, we add perturbations in the embedding space directly

[1] Semantically Equivalent Adversarial Rules for Debugging NLP Models, ACL 2018.
[2] Robust Neural Machine Translation with Doubly Adversarial Inputs, ACL 2019.

#3: Enhanced AT Algorithm

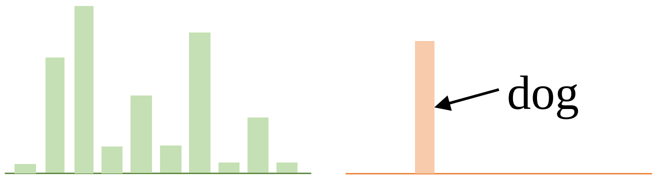
- Training objective:

$$\min_{\boldsymbol{\theta}} \mathbb{E}_{(\boldsymbol{x}_{img}, \boldsymbol{x}_{txt}, \boldsymbol{y}) \sim \mathcal{D}} \left[\mathcal{L}_{std}(\boldsymbol{\theta}) + \mathcal{R}_{at}(\boldsymbol{\theta}) + \alpha \cdot \mathcal{R}_{kl}(\boldsymbol{\theta}) \right]$$

- Cross-entropy loss on clean data:

$$\mathcal{L}_{std}(\boldsymbol{\theta}) = L(f_{\boldsymbol{\theta}}(\boldsymbol{x}_{img}, \boldsymbol{x}_{txt}), \boldsymbol{y})$$

( , A [MASK] lying on the grass next to a frisbee)



Probability vector

Ground-truth label

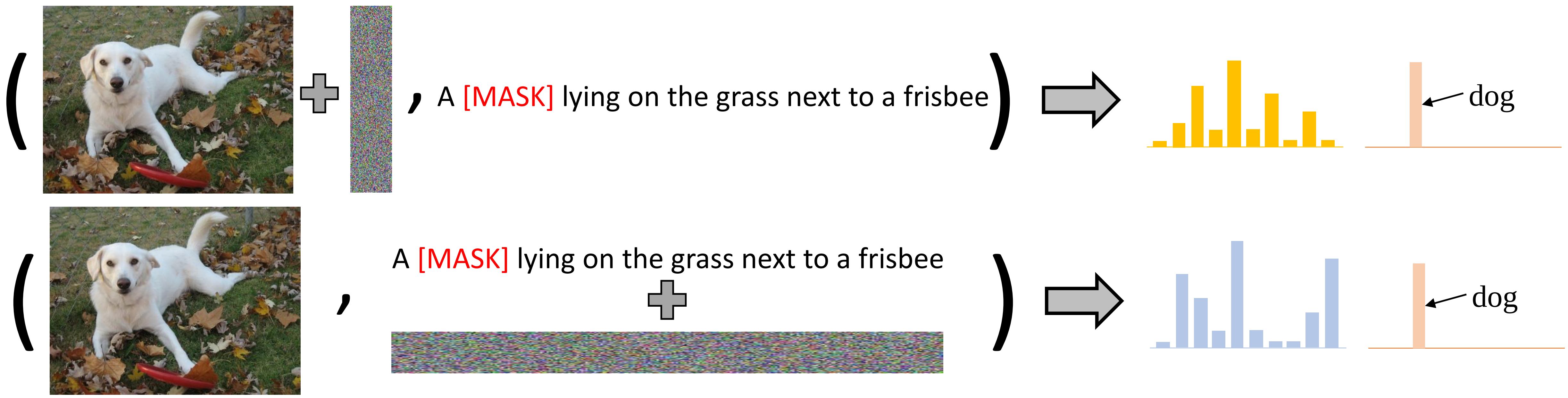
#3: Enhanced AT Algorithm

- Training objective:

$$\min_{\boldsymbol{\theta}} \mathbb{E}_{(\boldsymbol{x}_{img}, \boldsymbol{x}_{txt}, \boldsymbol{y}) \sim \mathcal{D}} \left[\mathcal{L}_{std}(\boldsymbol{\theta}) + \mathcal{R}_{at}(\boldsymbol{\theta}) + \alpha \cdot \mathcal{R}_{kl}(\boldsymbol{\theta}) \right]$$

- Cross-entropy loss on adversarial embeddings:

$$\mathcal{R}_{at}(\boldsymbol{\theta}) = \max_{||\boldsymbol{\delta}_{img}|| \leq \epsilon} L(f_{\boldsymbol{\theta}}(\boldsymbol{x}_{img} + \boldsymbol{\delta}_{img}, \boldsymbol{x}_{txt}), \boldsymbol{y}) + \max_{||\boldsymbol{\delta}_{txt}|| \leq \epsilon} L(f_{\boldsymbol{\theta}}(\boldsymbol{x}_{img}, \boldsymbol{x}_{txt} + \boldsymbol{\delta}_{txt}), \boldsymbol{y})$$



#3: Enhanced AT Algorithm

- Training objective:

$$\min_{\boldsymbol{\theta}} \mathbb{E}_{(\boldsymbol{x}_{img}, \boldsymbol{x}_{txt}, \boldsymbol{y}) \sim \mathcal{D}} \left[\mathcal{L}_{std}(\boldsymbol{\theta}) + \mathcal{R}_{at}(\boldsymbol{\theta}) + \alpha \cdot \mathcal{R}_{kl}(\boldsymbol{\theta}) \right]$$

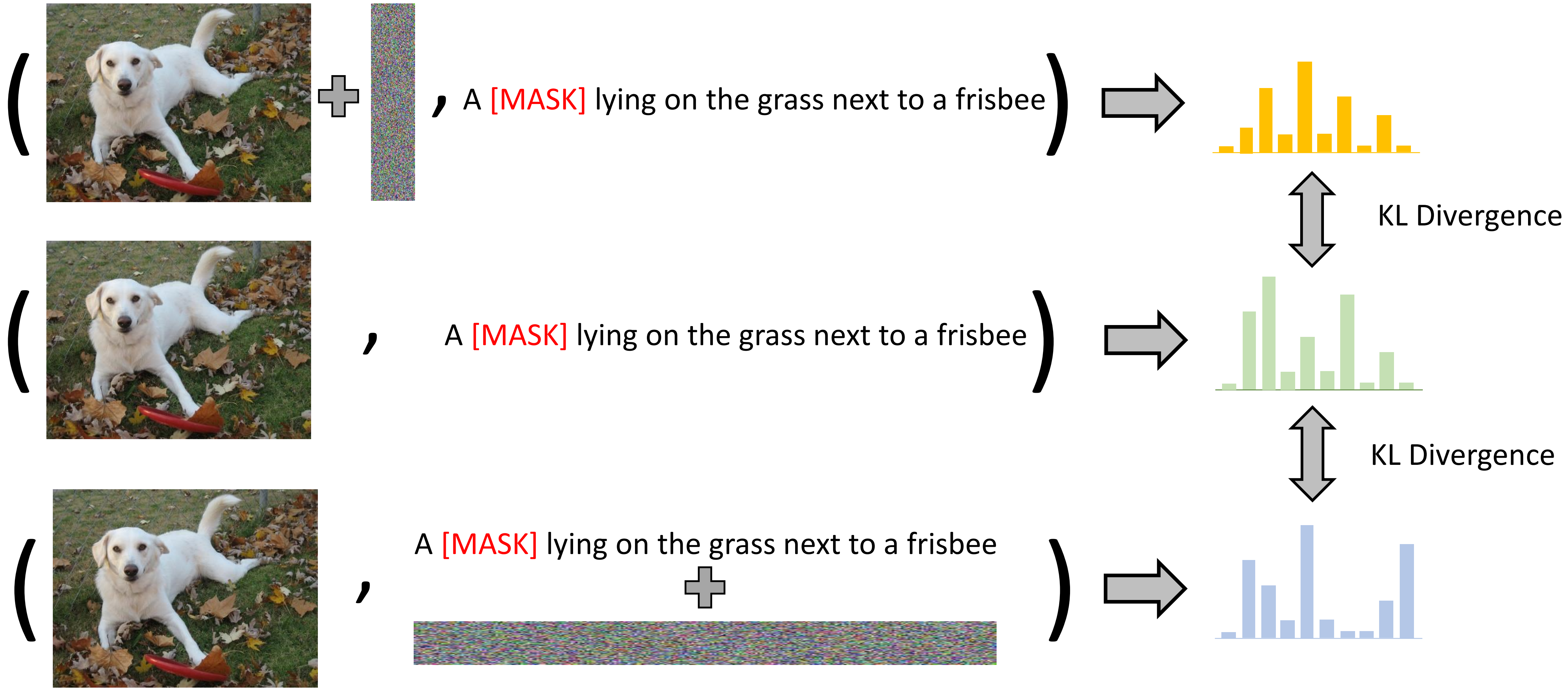
- KL-divergence loss for fine-grained adversarial regularization

$$\mathcal{R}_{kl}(\boldsymbol{\theta}) = \max_{||\boldsymbol{\delta}_{img}|| \leq \epsilon} L_{kl}(f_{\boldsymbol{\theta}}(\boldsymbol{x}_{img} + \boldsymbol{\delta}_{img}, \boldsymbol{x}_{txt}), f_{\boldsymbol{\theta}}(\boldsymbol{x}_{img}, \boldsymbol{x}_{txt})) + \max_{||\boldsymbol{\delta}_{txt}|| \leq \epsilon} L_{kl}(f_{\boldsymbol{\theta}}(\boldsymbol{x}_{img}, \boldsymbol{x}_{txt} + \boldsymbol{\delta}_{txt}), f_{\boldsymbol{\theta}}(\boldsymbol{x}_{img}, \boldsymbol{x}_{txt}))\,,$$

where $L_{kl}(p, q) = \mathrm{KL}(p||q) + \mathrm{KL}(q||p)$.

- Not only label-preserving, but the confidence level of the prediction between clean data and adversarial examples should also be close

#3: Enhanced AT Algorithm



#3: Enhanced AT Algorithm

Enable AT for large-scale training and promote diverse adversaries

Algorithm 1 "Free" Multi-modal Adversarial Training used in VILLA.

Require: Training samples $\mathcal{D} = \{(\boldsymbol{x}_{img}, \boldsymbol{x}_{txt}, \boldsymbol{y})\}$, 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 $\boldsymbol{\theta}$ given $\boldsymbol{\delta}_{img,t-1}$ and $\boldsymbol{\delta}_{txt,t-1}$
        $\boldsymbol{g}_t \leftarrow \boldsymbol{g}_{t-1} + \frac{1}{K}\mathbb{E}_{(\boldsymbol{x}_{img}, \boldsymbol{x}_{txt}, \boldsymbol{y}) \in B}[\nabla_{\boldsymbol{\theta}}(\mathcal{L}_{std}(\boldsymbol{\theta}) + \mathcal{R}_{at}(\boldsymbol{\theta}) + \mathcal{R}_{kl}(\boldsymbol{\theta}))]$
      Update the perturbation $\boldsymbol{\delta}_{img}$ and $\boldsymbol{\delta}_{txt}$ via gradient ascend
        $\tilde{\boldsymbol{y}} = f_{\boldsymbol{\theta}}(\boldsymbol{x}_{img}, \boldsymbol{x}_{txt})$
        $\boldsymbol{g}_{img} \leftarrow \nabla_{\boldsymbol{\delta}_{img}}[L(f_{\boldsymbol{\theta}}(\boldsymbol{x}_{img}+\boldsymbol{\delta}_{img}, \boldsymbol{x}_{txt}), \boldsymbol{y}) + L_{kl}(f_{\boldsymbol{\theta}}(\boldsymbol{x}_{img}+\boldsymbol{\delta}_{img}, \boldsymbol{x}_{txt}), \tilde{\boldsymbol{y}})]$
        $\boldsymbol{\delta}_{img,t} \leftarrow \Pi_{\|\boldsymbol{\delta}_{img}\|_F \le \epsilon}(\boldsymbol{\delta}_{img,t-1} + \alpha \cdot \boldsymbol{g}_{img}/\|\boldsymbol{g}_{img}\|_F)$
        $\boldsymbol{g}_{txt} \leftarrow \nabla_{\boldsymbol{\delta}_{txt}}[L(f_{\boldsymbol{\theta}}(\boldsymbol{x}_{img}, \boldsymbol{x}_{txt}+\boldsymbol{\delta}_{txt}), \boldsymbol{y}) + L_{kl}(f_{\boldsymbol{\theta}}(\boldsymbol{x}_{img}, \boldsymbol{x}_{txt}+\boldsymbol{\delta}_{txt}), \tilde{\boldsymbol{y}})]$
        $\boldsymbol{\delta}_{txt,t} \leftarrow \Pi_{\|\boldsymbol{\delta}_{txt}\|_F \le \epsilon}(\boldsymbol{\delta}_{txt,t-1} + \alpha \cdot \boldsymbol{g}_{txt}/\|\boldsymbol{g}_{txt}\|_F)$
    end for
    $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \tau \boldsymbol{g}_K$
  end for
end for

Accumulate the parameter gradient for "free"

Perturbation update via PGD (Projected Gradient Descent)

Parameter update via SGD (Stochastic Gradient Descent)

Results (VQA, VCR, NLVR2, SNLI-VE)

- Established new state of the art on all the tasks considered
- Gain: +0.85 on VQA, +2.9 on VCR, +1.49 on NLVR2, +0.64 on SNLI-VE

Method	VQA		VCR			NLVR^2		SNLI-VE	
	test-dev	test-std	Q→A	QA→R	Q→AR	dev	test-P	val	test
ViLBERT	70.55	70.92	72.42 (73.3)	74.47 (74.6)	54.04 (54.8)	-	-	-	-
VisualBERT	70.80	71.00	70.8 (71.6)	73.2 (73.2)	52.2 (52.4)	67.4	67.0	-	-
LXMERT	72.42	72.54	-	-	-	74.90	74.50	-	-
Unicoder-VL	-	-	72.6 (73.4)	74.5 (74.4)	54.4 (54.9)	-	-	-	-
12-in-1	73.15	-	-	-	-	-	78.87	-	76.95
VL-BERT$_{\text{BASE}}$	71.16	-	73.8 (-)	74.4 (-)	55.2 (-)	-	-	-	-
Oscar$_{\text{BASE}}$	73.16	73.44	-	-	-	78.07	78.36	-	-
UNITER$_{\text{BASE}}$	72.70	72.91	74.56 (75.0)	77.03 (77.2)	57.76 (58.2)	77.18	77.85	78.59	78.28
VILLA$_{\text{BASE}}$	**73.59**	**73.67**	**75.54 (76.4)**	**78.78 (79.1)**	**59.75 (60.6)**	**78.39**	**79.30**	**79.47**	**79.03**
VL-BERT$_{\text{LARGE}}$	71.79	72.22	75.5 (75.8)	77.9 (78.4)	58.9 (59.7)	-	-	-	-
Oscar$_{\text{LARGE}}$	73.61	73.82	-	-	-	79.12	80.37	-	-
UNITER$_{\text{LARGE}}$	73.82	74.02	77.22 (77.3)	80.49 (80.8)	62.59 (62.8)	79.12	79.98	79.39	79.38
VILLA$_{\text{LARGE}}$	**74.69**	**74.87**	**78.45 (78.9)**	**82.57 (82.8)**	**65.18 (65.7)**	**79.76**	**81.47**	**80.18**	**80.02**

(a) Results on VQA, VCR, NLVR^2, and SNLI-VE.

Results (ITR, RE)

- Gain: +1.52/+0.60 on Flickr30k IR & TR (R@1), and +0.99 on RE

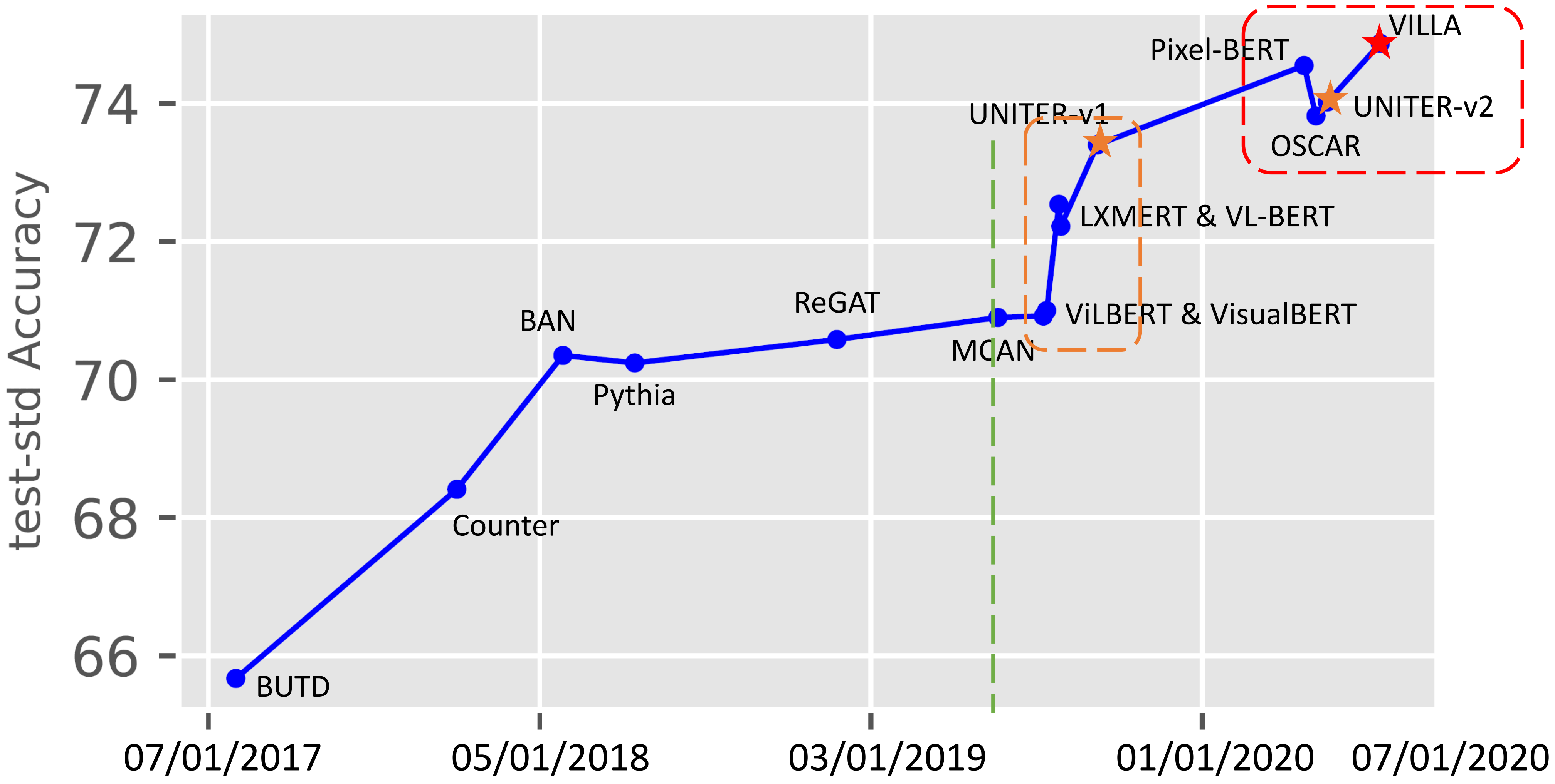
Method	RefCOCO+						RefCOCO					
	val	testA	testB	val^d	testA^d	testB^d	val	testA	testB	val^d	testA^d	testB^d
ViLBERT	-	-	-	72.34	78.52	62.61	-	-	-	-	-	-
VL-BERT$_{\text{BASE}}$	79.88	82.40	75.01	71.60	77.72	60.99	-	-	-	-	-	-
UNITER$_{\text{BASE}}$	83.66	86.19	78.89	75.31	81.30	65.58	91.64	92.26	90.46	81.24	86.48	73.94
VILLA$_{\text{BASE}}$	**84.26**	**86.95**	**79.22**	**76.05**	**81.65**	**65.70**	**91.93**	**92.79**	**91.38**	**81.65**	**87.40**	**74.48**
VL-BERT$_{\text{LARGE}}$	80.31	83.62	75.45	72.59	78.57	62.30	-	-	-	-	-	-
UNITER$_{\text{LARGE}}$	84.25	**86.34**	79.75	75.90	81.45	66.70	91.84	92.65	91.19	81.41	87.04	74.17
VILLA$_{\text{LARGE}}$	**84.40**	86.22	**80.00**	**76.17**	**81.54**	**66.84**	**92.58**	**92.96**	**91.62**	**82.39**	**87.48**	**74.84**

(b) Results on RefCOCO+ and RefCOCO. The superscript d denotes evaluation using detected proposals.

Method	RefCOCOg				Flickr30k IR			Flickr30k TR		
	val	test	val^d	test^d	R@1	R@5	R@10	R@1	R@5	R@10
ViLBERT	-	-	-	-	58.20	84.90	91.52	-	-	-
Unicoder-VL	-	-	-	-	71.50	90.90	94.90	86.20	96.30	99.00
UNITER$_{\text{BASE}}$	86.52	86.52	74.31	74.51	72.52	92.36	**96.08**	85.90	97.10	98.80
VILLA$_{\text{BASE}}$	**88.13**	**88.03**	**75.90**	**75.93**	**74.74**	**92.86**	95.82	**86.60**	**97.90**	**99.20**
UNITER$_{\text{LARGE}}$	87.85	87.73	74.86	75.77	75.56	94.08	96.76	87.30	**98.00**	**99.20**
VILLA$_{\text{LARGE}}$	**88.42**	**88.97**	**76.18**	**76.71**	**76.26**	**94.24**	**96.84**	**87.90**	97.50	98.80

(c) Results on RefCOCOg and Flickr30k Image Retrieval (IR) and Text Retrieval (TR).

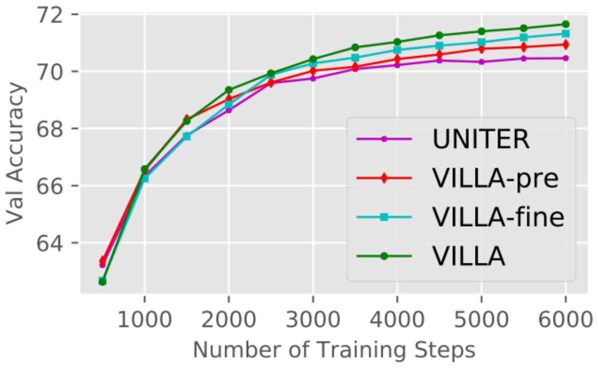
A Closer Look at VQA



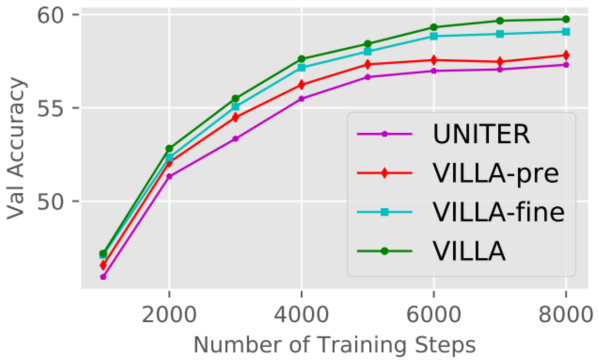
Pretraining vs. Finetuning

- Both adversarial pre-training and finetuning contribute to performance boost

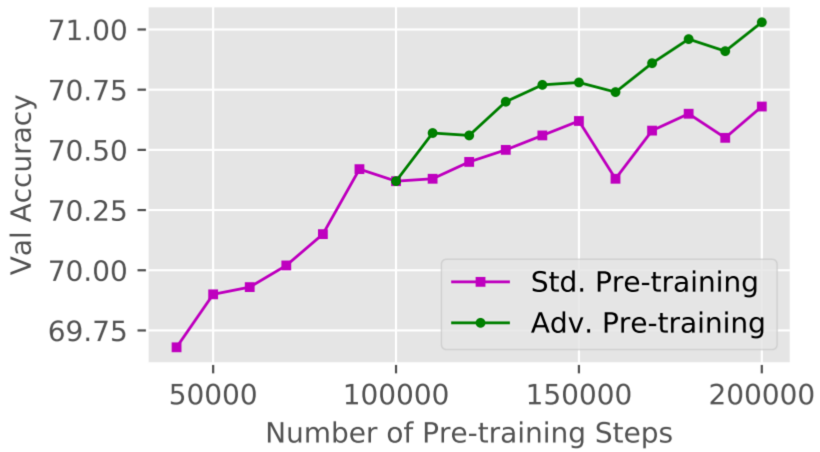
Method	VQA	VCR (val)			NLVR^2	VE	Flickr30k IR			RefCOCO		Ave.	
	test-dev	Q→A	QA→R	Q→AR	test-P	test	R@1	R@5	R@10	testA^d	testB^d		
UNITER (reimp.)	72.70	74.24	76.93	57.31	77.85	78.28	72.52	92.36	96.08	86.48	73.94	78.06	+0.51
VILLA-pre	73.03	74.76	77.04	57.82	78.44	78.43	73.76	93.02	96.28	87.34	74.35	78.57	+0.82
VILLA-fine	73.29	75.18	78.29	59.08	78.84	78.86	73.46	92.98	96.26	87.17	74.31	78.88	+1.15
VILLA	73.59	75.54	78.78	59.75	79.30	79.03	74.74	92.86	95.82	87.40	74.48	**79.21**	



(a) VQA



(b) VCR



VILLA vs. FreeLB

- Adversarial training on image or text modality alone is already effective
  - Most existing work shows that adversarial training for images cannot improve accuracy
- VILLA is consistently better than FreeLB

Method	VQA	VCR (val)		
	test-dev	Q→A	QA→R	Q→AR
$\text{VILLA}_{\text{BASE}}$ (txt)	73.50	75.60	78.70	59.67
$\text{VILLA}_{\text{BASE}}$ (img)	73.50	**75.81**	78.43	59.68
$\text{VILLA}_{\text{BASE}}$ (both)	**73.59**	75.54	**78.78**	**59.75**
$\text{VILLA}_{\text{LARGE}}$ (txt)	74.55	78.08	82.31	64.63
$\text{VILLA}_{\text{LARGE}}$ (img)	74.46	78.08	82.28	64.51
$\text{VILLA}_{\text{LARGE}}$ (both)	**74.69**	**78.45**	**82.57**	**65.18**

(a) Image vs. Text Modality.

Method	VQA	VCR (val)		
	test-dev	Q→A	QA→R	Q→AR
$\text{UNITER}_{\text{BASE}}$ (reimp.)	72.70	74.24	76.93	57.31
$\text{UNITER}_{\text{BASE}}$+FreeLB	72.82	75.13	77.90	58.73
$\text{VILLA}_{\text{BASE}}$-fine	**73.29**	**75.49**	**78.34**	**59.30**
$\text{UNITER}_{\text{LARGE}}$ (reimp.)	73.82	76.70	80.61	62.15
$\text{UNITER}_{\text{LARGE}}$+FreeLB	73.87	77.19	81.44	63.24
$\text{VILLA}_{\text{LARGE}}$-fine	**74.32**	**77.75**	**82.10**	**63.99**

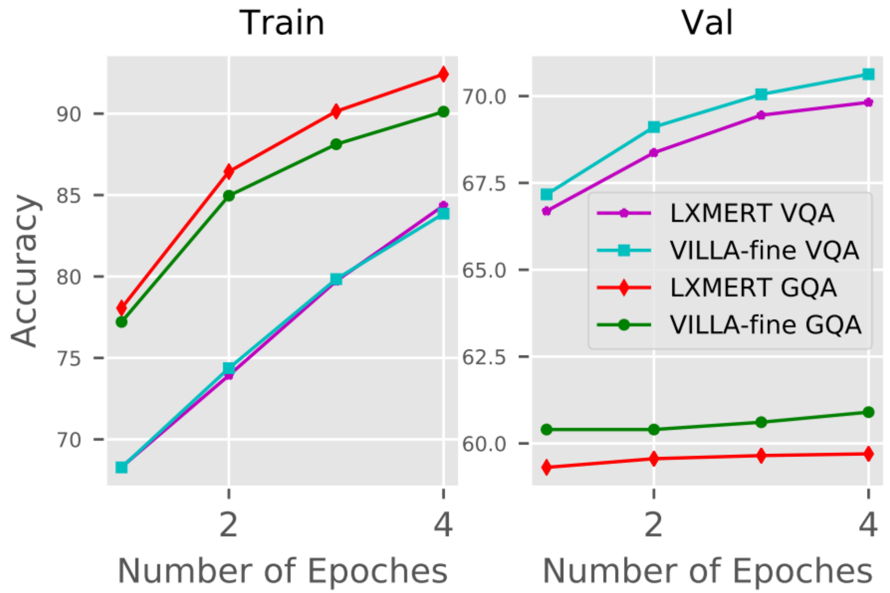
(b) FreeLB vs. VILLA.

Generalizability of VILLA

- VILLA can be applied to any multimodal pre-training methods (e.g., LXMERT)

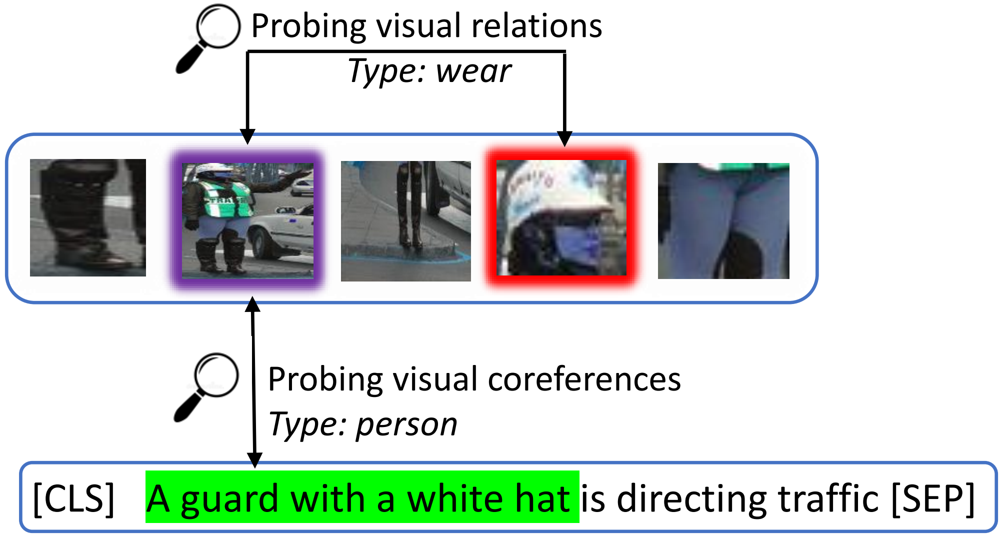
Method	VQA		GQA		NLVR^2		Meta-Ave.	
	test-dev	test-std	test-dev	test-std	dev	test-P		
LXMERT	72.42	72.54	60.00	60.33	74.95	74.45	69.12	
LXMERT (reimp.)	72.50	72.52	59.92	60.28	74.72	74.75	69.12	
VILLA-fine	**73.02**	**73.18**	**60.98**	**61.12**	**75.98**	**75.73**	**70.00**	*+0.88*

- Adversarial training as a regularizer



Probing Analysis

- Probing the attention heads (12 layers, and 12 heads in each layer)



- VILLA captures richer visual coreference and visual relation knowledge

Model	Visual Coreference (Flickr30k)					Visual Relation (Visual Genome)					Ave.
	scene	clothing	animals	instruments	vehicles	on	standing in	wearing	holding	covering	
$\text{UNITER}_{\text{BASE}}$	0.151	0.157	0.285	0.244	0.194	0.154	0.107	0.311	0.200	0.151	0.195
$\text{VILLA}_{\text{BASE}}$	**0.169**	**0.185**	**0.299**	**0.263**	**0.202**	**0.201**	**0.120**	**0.353**	**0.241**	**0.192**	**0.223**

Visualization (Text-to-Image Attention)

- VILLA learns more accurate and sharper attention maps than UNITER

Robustness to Paraphrases

- UNITER has already lifted up the performance by a large margin
- VILLA facilitates further performance boost

Data split	MUTAN	BUTD	BUTD+CC	Pythia	Pythia+CC	BAN	BAN+CC	UNITER	VILLA
Original	59.08	61.51	62.44	64.08	64.52	64.97	65.87	70.35	**71.27**
Rephrasing	46.87	51.22	52.58	54.20	55.65	55.87	56.59	64.56	**65.35**

Table 6: Results on VQA-Rephrasings. Both UNITER and VILLA use the base model size. Baseline results are copied from [57].

Takeaway Message

- VILLA is the first known effort that proposes adversarial training for V+L representation learning
- Code is available at

  https://github.com/zhegan27/VILLA

- Adversarial robustness of V+L models could be interesting future work