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

Zhe Gan, Yen-Chun Chen, Linjie Li, Chen Zhu, Yu Cheng, Jingjing Liu
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?
VILLA: Vision-and-Language Large-scale Adversarial Training
Preliminary: What’s Adversarial Attack?

- Neural Networks are prone to label-preserving adversarial examples

**Computer Vision:**

“pig”

+ 0.005 x

“airliner”

**Natural Language Processing:**

| Original | Changed |
|----------|---------|
| What is the oncorhynchus also called? A: chum salmon | What’s the oncorhynchus also called? A: keta |
| How long is the Rhine? A: 1,230 km | 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

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

• 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 correlated

  • **MLM during pre-training (masking out an object):**
    [CLS] A [MASK] laying 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

\[
\min_\theta \mathbb{E}_{(x_{img}, x_{txt}, y) \sim \mathcal{D}} [L_\theta(x_{img}, x_{txt}, y)].
\]
#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

- 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:
  \[
  \min_\theta \mathbb{E}_{(x_{img}, x_{txt}, y) \sim D} \left[ L_{std}(\theta) + R_{at}(\theta) + \alpha \cdot R_{kl}(\theta) \right]
  \]

- Cross-entropy loss on clean data:
  \[
  L_{std}(\theta) = L(f_\theta(x_{img}; x_{txt}); y)
  \]

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

\[
\text{Probability vector} \quad \text{Ground-truth label}
\]
#3: Enhanced AT Algorithm

• Training objective:

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

• Cross-entropy loss on adversarial embeddings:

$$R_{at}(\theta) = \max_{||\delta_{img}|| \leq \varepsilon} L(f_\theta(x_{img} + \delta_{img}, x_{txt}), y) + \max_{||\delta_{txt}|| \leq \varepsilon} L(f_\theta(x_{img}, x_{txt} + \delta_{txt}), y)$$

A [MASK] lying on the grass next to a frisbee
#3: Enhanced AT Algorithm

- Training objective:

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

- KL-divergence loss for fine-grained adversarial regularization

  \[
  \mathcal{R}_{kl}(\theta) = \max_{\|\delta_{img}\| \leq \varepsilon} L_{kl}(f_\theta(x_{img} + \delta_{img}, x_{txt}), f_\theta(x_{img}, x_{txt}))
  \]

  \[
  + \max_{\|\delta_{txt}\| \leq \varepsilon} L_{kl}(f_\theta(x_{img}, x_{txt} + \delta_{txt}), f_\theta(x_{img}, x_{txt}))
  \]

  where \( L_{kl}(p, q) = \text{KL}(p||q) + \text{KL}(q||p) \)

- 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

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

KL Divergence

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

KL Divergence

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

KL Divergence
#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 $\mathcal{D} = \{(x_{\text{img}}, x_{\text{txt}}, y)\}$, perturbation bound $\epsilon$, learning rate $\tau$, ascent steps $K$, ascent step size $\alpha$

1: Initialize $\theta$
2: for epoch = 1 \ldots N_{ep} do
3: for minibatch $B \subset X$ do
4: $\delta_0 \leftarrow \frac{1}{\sqrt{N_S}} U(-\epsilon, \epsilon)$, $g_0 \leftarrow 0$
5: for $t = 1 \ldots K$ do
6: Accumulate gradient of parameters $\theta$ given $\delta_{\text{img},t-1}$ and $\delta_{\text{txt},t-1}$
7: $g_t \leftarrow g_{t-1} + \frac{1}{K} \mathbb{E}_{(x_{\text{img}}, x_{\text{txt}}, y) \in B} [\nabla_{\theta} (L_{\text{std}}(\theta) + R_{\text{at}}(\theta) + R_{\text{kl}}(\theta))]$
8: Update the perturbation $\delta_{\text{img}}$ and $\delta_{\text{txt}}$ via gradient ascends
9: $y = f_{\theta}(x_{\text{img}}, x_{\text{txt}})$
10: $g_{\text{img}} \leftarrow \nabla_{\delta_{\text{img}}} [L(f_{\theta}(x_{\text{img}} + \delta_{\text{img}}, x_{\text{txt}}), y)] + L_{\text{kl}}(f_{\theta}(x_{\text{img}} + \delta_{\text{img}}, x_{\text{txt}}), \tilde{y})$
11: $\delta_{\text{img},t} \leftarrow \Pi_{\|g_{\text{img}}\|_F \leq \epsilon} (\delta_{\text{img},t-1} + \alpha \cdot g_{\text{img}} / \|g_{\text{img}}\|_F)$
12: $g_{\text{txt}} \leftarrow \nabla_{\delta_{\text{txt}}} [L(f_{\theta}(x_{\text{img}}, x_{\text{txt}} + \delta_{\text{txt}}), y)] + L_{\text{kl}}(f_{\theta}(x_{\text{img}}, x_{\text{txt}} + \delta_{\text{txt}}), \tilde{y})$
13: $\delta_{\text{txt},t} \leftarrow \Pi_{\|g_{\text{txt}}\|_F \leq \epsilon} (\delta_{\text{txt},t-1} + \alpha \cdot g_{\text{txt}} / \|g_{\text{txt}}\|_F)$
14: end for
15: $\theta \leftarrow \theta - \tau g_K$
16: end for
17: 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
### 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 | testA | testB | val | testA | testB | val | testA | testB | val | testA | testB | val | testA | testB | val | testA | testB | val | testA | testB | val | testA | testB | val | testA | testB | val | testA | testB | val | testA | testB |
|                |     |       |       |     |       |       |     |       |       |     |       |       |     |       |       |     |       |       |     |       |       |     |       |       |     |       |       |     |       |       |     |       |       |     |       |       |     |       |       |
| ViLBERT        | -   | -     | 72.34 | 78.52 | 62.61 |       | -   | -     | -     | -   | -     | -     | -   | -     | -     | -   | -     | -     | -   | -     | -     | -   | -     | -     | -   | -     | -     | -   | -     | -     | -   | -     | -     |
| VL-BERT_BASE   | 79.88 | 82.40 | 75.01 | 71.60 | 77.72 | 60.99 | -   | -     | -     | -   | -     | -     | -   | -     | -     | -   | -     | -     | -   | -     | -     | -   | -     | -     | -   | -     | -     | -   | -     | -     | -   | -     | -     |
| UNITER_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_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_LARGE  | 80.31 | 83.62 | 75.45 | 72.59 | 78.57 | 62.30 | -   | -     | -     | -   | -     | -     | -   | -     | -     | -   | -     | -     | -   | -     | -     | -   | -     | -     | -   | -     | -     | -   | -     | -     | -   | -     | -     |
| UNITER_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_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 | 86.20 | 96.30 | 99.00 |
| Unicoder-VL    | -   | -     | -     | -     | 71.50 | 90.90 | 94.90 | 85.90 | 97.10 | 98.80 |
| UNITER_BASE    | 86.52 | 86.52 | 74.31 | 74.51 | 72.52 | 92.36 | **96.08** | 85.30 | **98.00** | **99.20** |
| VILLA_BASE     | **88.13** | **88.03** | **75.90** | **75.93** | **74.74** | **92.86** | 95.82 | **86.60** | **97.90** | **99.20** |
| UNITER_LARGE   | 87.85 | 87.73 | 74.86 | 75.77 | 75.56 | 94.08 | 96.76 | 87.30 | **98.00** | **99.20** |
| VILLA_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 test-dev | VQA Q→A | VQA QA→R | VQA Q→AR | NLVR2 test-P | VE test | Flickr30k IR R@1 | Flickr30k IR R@5 | Flickr30k IR R@10 | RefCOCO testA | RefCOCO testB | RefCOCO Ave. |
|-------------|--------------|---------|----------|----------|--------------|---------|----------------|----------------|----------------|---------------|---------------|--------------|
| 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        |
| 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        |
| 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        |
| 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

+0.51
+0.82
+1.15
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 test-dev | VQA Q→A | VQA QA→R | VQA Q→AR |
|-------------------|-------------|---------|---------|---------|
| VILLA_BASE (txt)  | 73.50       | 75.60   | 78.70   | 59.67   |
| VILLA_BASE (img)  | 73.50       | **75.81** | 78.43   | 59.68   |
| VILLA_BASE (both) | **73.59**   | 75.54   | **78.78** | **59.75** |
| VILLA_LARGE (txt) | 74.55       | 78.08   | **82.31** | 64.63   |
| VILLA_LARGE (img) | 74.46       | 78.08   | 82.28   | 64.51   |
| VILLA_LARGE (both)| **74.69**   | **78.45** | **82.57** | **65.18** |

(a) Image vs. Text Modality.

| Method                      | VQA test-dev | VQA Q→A | VQA QA→R | VQA Q→AR |
|-----------------------------|-------------|---------|---------|---------|
| UNITER_BASE (reimp.)        | 72.70       | 74.24   | 76.93   | 57.31   |
| UNITER_BASE+FreeLB          | 72.82       | 75.13   | 77.90   | 58.73   |
| VILLA_BASE-fine             | **73.29**   | **75.49** | **78.34** | **59.30** |
| UNITER_LARGE (reimp.)       | 73.82       | 76.70   | 80.61   | 62.15   |
| UNITER_LARGE+FreeLB         | 73.87       | 77.19   | 81.44   | 63.24   |
| VILLA_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   | NLVR² | 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  | 72.32 | 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 |

• 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 |        |      |
| UNITERBASE   | 0.151 | 0.157    | 0.285   | 0.244       | 0.194    | 0.154| 0.107       | 0.311    | 0.200   | 0.151    | 0.195  |
| VILLA 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