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

![Comparison of 'pig' and 'airliner' images with added noise and text examples.](image)

**Computer Vision:**

- **Original:** What is the oncorhynchus also called? **A:** chum salmon
- **Changed:** What’s the oncorhynchus also called? **A:** keta

**Natural Language Processing:**

- **Original:** How long is the Rhine? **A:** 1,230 km
- **Changed:** How long is the Rhine?? **A:** more than 1,050,000

(b) Example for (is→is’s) (c) Example for (?)→(?)

---

Preliminary: What’s Adversarial Training (AT)?

• A min-max game to harness adversarial examples

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

• 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] 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

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

#3: Enhanced AT Algorithm

• Training objective:

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

• Cross-entropy loss on clean data:

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

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

![Image](image.png)

- Probability vector
- Ground-truth label

dog
#3: Enhanced AT Algorithm

- Training objective:

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

- Cross-entropy loss on adversarial embeddings:

\[
R_{\text{at}}(\theta) = \max_{||\delta_{\text{img}}|| \leq \epsilon} L(f_{\theta}(x_{\text{img}} + \delta_{\text{img}}, x_{\text{txt}}), y) + \max_{||\delta_{\text{txt}}|| \leq \epsilon} L(f_{\theta}(x_{\text{img}}, x_{\text{txt}} + \delta_{\text{txt}}), y)
\]
#3: Enhanced AT Algorithm

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

- KL-divergence loss for fine-grained adversarial regularization
  \[
  R_{kl}(\theta) = \max_{||\delta_{img}|| \leq \epsilon} L_{kl}(f_{\theta}(x_{img} + \delta_{img}, x_{txt}), f_{\theta}(x_{img}, x_{txt})) \\
  + \max_{||\delta_{txt}|| \leq \epsilon} L_{kl}(f_{\theta}(x_{img}, x_{txt} + \delta_{txt}), f_{\theta}(x_{img}, x_{txt}))
  \]
  where \( L_{kl}(p, q) = KL(p || q) + KL(q || p) \)

- 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

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 $D = \{(x_{img}, x_{txt}, y)\}$, 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 $\delta_{img,t-1}$ and $\delta_{txt,t-1}$
7:       $g_t \leftarrow g_{t-1} + \frac{1}{K}E_{(x_{img}, x_{txt}, y) \in B} [\nabla_\theta (L_{std}(\theta) + R_{at}(\theta) + R_{kl}(\theta))]$
8:       Update the perturbation $\delta_{img}$ and $\delta_{txt}$ via gradient ascend
9:       $\hat{y} = f_\theta(x_{img}, x_{txt})$
10:      $g_{img} \leftarrow \nabla_{\delta_{img}} [L(f_\theta(x_{img} + \delta_{img}, x_{txt}), y) + L_{kl}(f_\theta(x_{img} + \delta_{img}, x_{txt}, \hat{y})]$
11:      $\delta_{img,t} \leftarrow \Pi_{\|\delta_{img,t-1} + \alpha \cdot g_{img} / \|g_{img}\|_F \leq \epsilon} (\delta_{img,t-1} + \alpha \cdot g_{img} / \|g_{img}\|_F}$
12:      $g_{txt} \leftarrow \nabla_{\delta_{txt}} [L(f_\theta(x_{img}, x_{txt} + \delta_{txt}), y) + L_{kl}(f_\theta(x_{img}, x_{txt} + \delta_{txt}, \hat{y})]$
13:      $\delta_{txt,t} \leftarrow \Pi_{\|\delta_{txt,t-1} + \alpha \cdot g_{txt} / \|g_{txt}\|_F \leq \epsilon} (\delta_{txt,t-1} + \alpha \cdot g_{txt} / \|g_{txt}\|_F}$
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

<table>
<thead>
<tr>
<th>Method</th>
<th>VQA</th>
<th>VQA</th>
<th>VQA</th>
<th>VQA</th>
<th>NLVR²</th>
<th>SNLI-VE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>test-dev</td>
<td>test-std</td>
<td>Q→A</td>
<td>Q→R</td>
<td>Q→AR</td>
<td>dev</td>
</tr>
<tr>
<td>ViLBERT</td>
<td>70.55</td>
<td>70.92</td>
<td>72.42 (73.3)</td>
<td>74.47 (74.6)</td>
<td>54.04 (54.8)</td>
<td>-</td>
</tr>
<tr>
<td>VisualBERT</td>
<td>70.80</td>
<td>71.00</td>
<td>70.8 (71.6)</td>
<td>73.2 (73.2)</td>
<td>52.2 (52.4)</td>
<td>67.4</td>
</tr>
<tr>
<td>LXMERT</td>
<td>72.42</td>
<td>72.54</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>74.90</td>
</tr>
<tr>
<td>Unicoder-VL</td>
<td>-</td>
<td>-</td>
<td>72.6 (73.4)</td>
<td>74.5 (74.4)</td>
<td>54.4 (54.9)</td>
<td>-</td>
</tr>
<tr>
<td>12-in-1</td>
<td>73.15</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>VL-BERT_BASE</td>
<td>71.16</td>
<td>-</td>
<td>73.8 (-)</td>
<td>74.4 (-)</td>
<td>55.2 (-)</td>
<td>-</td>
</tr>
<tr>
<td>Oscar_BASE</td>
<td>73.16</td>
<td>73.44</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>78.07</td>
</tr>
<tr>
<td>UNITER_BASE</td>
<td>72.70</td>
<td>72.91</td>
<td>74.56 (75.0)</td>
<td>77.03 (77.2)</td>
<td>57.76 (58.2)</td>
<td>77.18</td>
</tr>
<tr>
<td>VILLA_BASE</td>
<td><strong>73.59</strong></td>
<td><strong>73.67</strong></td>
<td><strong>75.54 (76.4)</strong></td>
<td><strong>78.78 (79.1)</strong></td>
<td><strong>59.75 (60.6)</strong></td>
<td><strong>78.39</strong></td>
</tr>
<tr>
<td>VL-BERT_LARGE</td>
<td>71.79</td>
<td>72.22</td>
<td>75.5 (75.8)</td>
<td>77.9 (78.4)</td>
<td>58.9 (59.7)</td>
<td>-</td>
</tr>
<tr>
<td>Oscar_LARGE</td>
<td>73.61</td>
<td>73.82</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>79.12</td>
</tr>
<tr>
<td>UNITER_LARGE</td>
<td>73.82</td>
<td>74.02</td>
<td>77.22 (77.3)</td>
<td>80.49 (80.8)</td>
<td><strong>62.59 (62.8)</strong></td>
<td>79.12</td>
</tr>
<tr>
<td>VILLA_LARGE</td>
<td><strong>74.69</strong></td>
<td><strong>74.87</strong></td>
<td><strong>78.45 (78.9)</strong></td>
<td><strong>82.57 (82.8)</strong></td>
<td><strong>65.18 (65.7)</strong></td>
<td><strong>79.76</strong></td>
</tr>
</tbody>
</table>

(a) Results on VQA, VCR, NLVR², and SNLI-VE.
Results (ITR, RE)

• Gain: +1.52/+0.60 on Flickr30k IR & TR (R@1), and +0.99 on RE
Pretraining vs. Finetuning

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

<table>
<thead>
<tr>
<th>Method</th>
<th>VQA test-dev</th>
<th>VQA (val) Q→A</th>
<th>VQA (val) QA→R</th>
<th>VQA (val) Q→AR</th>
<th>NLVR² test-P</th>
<th>VE test R@1</th>
<th>VE test R@5</th>
<th>VE test R@10</th>
<th>Flickr30k IR testA averaged</th>
<th>Flickr30k IR testB averaged</th>
<th>Ave.</th>
</tr>
</thead>
<tbody>
<tr>
<td>UNITER (reimp.)</td>
<td>72.70</td>
<td>74.24</td>
<td>76.93</td>
<td>57.31</td>
<td>77.85</td>
<td>78.28</td>
<td>72.52</td>
<td>92.36</td>
<td>96.08</td>
<td>78.06</td>
<td>+0.51</td>
</tr>
<tr>
<td>VILLA-pre</td>
<td>73.03</td>
<td>74.76</td>
<td>77.04</td>
<td>57.82</td>
<td>78.44</td>
<td>78.43</td>
<td>73.76</td>
<td>93.02</td>
<td>96.28</td>
<td>78.57</td>
<td>+0.82</td>
</tr>
<tr>
<td>VILLA-fine</td>
<td>73.29</td>
<td>75.18</td>
<td>78.29</td>
<td>59.08</td>
<td>78.84</td>
<td>78.86</td>
<td>73.46</td>
<td>92.98</td>
<td>96.26</td>
<td>78.88</td>
<td>+1.15</td>
</tr>
<tr>
<td>VILLA</td>
<td>73.59</td>
<td>75.54</td>
<td>78.78</td>
<td>59.75</td>
<td>79.30</td>
<td>79.03</td>
<td>74.74</td>
<td>92.86</td>
<td>95.82</td>
<td>79.21</td>
<td></td>
</tr>
</tbody>
</table>
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

<table>
<thead>
<tr>
<th>Method</th>
<th>VQA test-dev</th>
<th>VQA Q→A</th>
<th>VQA QA→R</th>
<th>VQA Q→AR</th>
</tr>
</thead>
<tbody>
<tr>
<td>VILLA_BASE (txt)</td>
<td>73.50</td>
<td>75.60</td>
<td>78.70</td>
<td>59.67</td>
</tr>
<tr>
<td>VILLA_BASE (img)</td>
<td>73.50</td>
<td><strong>75.81</strong></td>
<td>78.43</td>
<td>59.68</td>
</tr>
<tr>
<td>VILLA_BASE (both)</td>
<td><strong>73.59</strong></td>
<td>75.54</td>
<td><strong>78.78</strong></td>
<td><strong>59.75</strong></td>
</tr>
<tr>
<td>VILLA_LARGE (txt)</td>
<td>74.55</td>
<td>78.08</td>
<td>82.31</td>
<td>64.63</td>
</tr>
<tr>
<td>VILLA_LARGE (img)</td>
<td>74.46</td>
<td>78.08</td>
<td>82.28</td>
<td>64.51</td>
</tr>
<tr>
<td>VILLA_LARGE (both)</td>
<td><strong>74.69</strong></td>
<td><strong>78.45</strong></td>
<td><strong>82.57</strong></td>
<td><strong>65.18</strong></td>
</tr>
</tbody>
</table>

(a) Image vs. Text Modality.

<table>
<thead>
<tr>
<th>Method</th>
<th>VQA test-dev</th>
<th>VQA Q→A</th>
<th>VQA QA→R</th>
<th>VQA Q→AR</th>
</tr>
</thead>
<tbody>
<tr>
<td>UNITER_BASE (reimp.)</td>
<td>72.70</td>
<td>74.24</td>
<td>76.93</td>
<td>57.31</td>
</tr>
<tr>
<td>UNITER_BASE+FreeLB</td>
<td>72.82</td>
<td>75.13</td>
<td>77.90</td>
<td>58.73</td>
</tr>
<tr>
<td>VILLA_BASE-fine</td>
<td><strong>73.29</strong></td>
<td><strong>75.49</strong></td>
<td><strong>78.34</strong></td>
<td><strong>59.30</strong></td>
</tr>
<tr>
<td>UNITER_LARGE (reimp.)</td>
<td>73.82</td>
<td>76.70</td>
<td>80.61</td>
<td>62.15</td>
</tr>
<tr>
<td>UNITER_LARGE+FreeLB</td>
<td>73.87</td>
<td>77.19</td>
<td>81.44</td>
<td>63.24</td>
</tr>
<tr>
<td>VILLA_LARGE-fine</td>
<td><strong>74.32</strong></td>
<td><strong>77.75</strong></td>
<td><strong>82.10</strong></td>
<td><strong>63.99</strong></td>
</tr>
</tbody>
</table>

(b) FreeLB vs. VILLA.
Generalizability of VILLA

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

<table>
<thead>
<tr>
<th>Method</th>
<th>VQA test-dev</th>
<th>VQA test-std</th>
<th>GQA test-dev</th>
<th>GQA test-std</th>
<th>NLVR² dev</th>
<th>NLVR² test-P</th>
<th>Meta-Ave.</th>
</tr>
</thead>
<tbody>
<tr>
<td>LXMERT</td>
<td>72.42</td>
<td>72.54</td>
<td>60.00</td>
<td>60.33</td>
<td>74.95</td>
<td>74.45</td>
<td>69.12</td>
</tr>
<tr>
<td>LXMERT (reimp.)</td>
<td>72.50</td>
<td>72.52</td>
<td>59.92</td>
<td>60.28</td>
<td>74.72</td>
<td>74.75</td>
<td>69.12</td>
</tr>
<tr>
<td>VILLA-fine</td>
<td><strong>73.02</strong></td>
<td><strong>73.18</strong></td>
<td><strong>60.98</strong></td>
<td><strong>61.12</strong></td>
<td><strong>75.98</strong></td>
<td><strong>75.73</strong></td>
<td><strong>70.00</strong></td>
</tr>
</tbody>
</table>

- Adversarial training as a regularizer

![Graph showing performance improvement](image)
Probing Analysis

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

• VILLA captures richer visual coreference and visual relation knowledge

<table>
<thead>
<tr>
<th>Model</th>
<th>Visual Coreference (Flickr30k)</th>
<th>Visual Relation (Visual Genome)</th>
<th>Ave.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>scene</td>
<td>clothing</td>
<td>animals</td>
</tr>
<tr>
<td>UNITERBASE</td>
<td>0.151</td>
<td>0.157</td>
<td>0.285</td>
</tr>
<tr>
<td>VILLABASE</td>
<td>0.169</td>
<td>0.185</td>
<td>0.299</td>
</tr>
</tbody>
</table>
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

<table>
<thead>
<tr>
<th>Data split</th>
<th>MUTAN</th>
<th>BUTD</th>
<th>BUTD+CC</th>
<th>Pythia</th>
<th>Pythia+CC</th>
<th>BAN</th>
<th>BAN+CC</th>
<th>UNITER</th>
<th>VILLA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original</td>
<td>59.08</td>
<td>61.51</td>
<td>62.44</td>
<td>64.08</td>
<td>64.52</td>
<td>64.97</td>
<td>65.87</td>
<td>70.35</td>
<td>71.27</td>
</tr>
<tr>
<td>Rephrasing</td>
<td>46.87</td>
<td>51.22</td>
<td>52.58</td>
<td>54.20</td>
<td>55.65</td>
<td>55.87</td>
<td>56.59</td>
<td>64.56</td>
<td>65.35</td>
</tr>
</tbody>
</table>

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](https://github.com/zhegan27/VILLA)
- Adversarial robustness of V+L models could be interesting future work