

# A Unified Understanding of Adversarial Vulnerability Regarding Unimodal Models and Vision-Language Pre-training Models

## –Supplementary Material–

### A Attack Details

Three key elements are required to implement FGA, namely an image encoder, guiding vectors, and guiding labels. Below, we will elaborate on the construction of these elements in four scenarios: VE, VQA, VG, and VR.

#### A.1 Visual Entailment

**Task Detail.** We conduct the attack experiment on the VE task [16] with ALBEF [7] which treats VE as a three-classification problem and connects a multi-layer perceptron (MLP) after the [CLS] vector. The input layer and all hidden layers of the MLP constitute  $P$ , obtaining the image encoder  $E(v|t) = P(E_m(E_v(v), E_t(t)))$ . The output layer of the MLP is a linear layer, whose weight matrix is  $W = \{\omega_0, \omega_1, \omega_2\}$ . The three guiding vectors are associated with three categories “contradiction, neutral and entailment”, and the label  $y \in \{0, 1, 2\}$  of the input image-text pair  $(v, t)$  provides the direction of the attack, that is, guiding  $E(v'|t)$  the embedding of adversarial image  $v'$  deviates from the guiding vector  $\omega_y$ .

**Dataset Detail.** The SNLI-VE dataset [16] is a benchmark for visual entailment, which aims to determine whether an image supports, contradicts, or is neutral to a given natural language statement. This task extends the concept of natural language inference (NLI) to the visual domain, presenting challenges in image and text understanding. The dataset is constructed based on two existing datasets: SNLI (Stanford Natural Language Inference) and Flickr30k. We use its test split, which contains 1000 images, 5973 entailment texts, 5964 neutral texts, and 5964 contradiction texts.

#### A.2 Visual Question Answering

**Task Detail.** We conduct the attack experiment on the VQA task [5] with ALBEF which performs this task in the manner of text generation. The image-question pair is fed into ALBEF to extract the fused embedding, which is then sent to a decoder to generate an answer. The dictionary size of the decoder is 30522, so the end of the decoder is a linear classification head, with weight matrix  $\{\omega_i\}_{i=0}^{30521}$ . The VQA 2.0 dataset [5] provides 3,128 candidate answers. To align with this task, ALBEF only considers 3,128 output possibilities. We follow this by selecting 3,128 vectors from the weights of the linear layer to form the guiding vectors  $\{\omega_i\}_{i=0}^{3127}$ , each corresponding to an answer. To perform FGA, we denote the decoder excluding the linear classification head as  $P$ , and we still lack guiding labels. For convenience, we directly use the network’s prediction results as the guiding labels, which is  $\text{argmax}_i(P(E_m(E_v(v'), E_t(t))) \cdot \omega_i)$ .

**Dataset Detail.** The VQA2.0 dataset includes images from the MS COCO (Microsoft Common Objects in Context) dataset [8], providing a diverse set of real-world images depicting various objects, scenes, and activities. For each image, multiple questions are generated, covering a wide range of topics such as object recognition, counting, colour identification, spatial relationships, and

more. Each question is accompanied by multiple answers, provided by different human annotators. The answers can be in the form of single words, phrases, or numbers. It contains 83k images for training, 41k for validation, and 81k for test. We conduct attack tests based on the test-dev and test-std splits.

#### A.3 Visual Grounding

**Task Detail.** We conduct the attack experiment on the VE task with ALBEF which extends Grad-CAM [11] to acquire heatmaps and use them to rank the detected proposals provided in advance. During this task, after the fused encoder, ALBEF is followed by a linear image-text matching binary classifier, the weight matrix of which is  $W = \{\omega_0, \omega_1\}$ . The larger the inner product between the fused embedding and  $\omega_1$ , the more the input image-text pair  $(v, t)$  matches. ALBEF backpropagates the gradient based on the loss value  $\text{Em}(E_v(v), E_t(t)) \cdot \omega_1$ , obtains the heatmap, and then performs the VG task. Consequently, we use FGA to guide  $\text{Em}(E_v(v'), E_t(t))$  away from  $\omega_1$  as the attack strategy.

**Dataset Detail.** RefCOCO+ [17] is a dataset designed for referring expression comprehension in the context of images. It is an extension of the original RefCOCO dataset and specifically aims at addressing the challenge of grounding referring expressions that require fine-grained distinctions between objects. The key components of the RefCOCO+ dataset are: (1) **Images:** The dataset uses images from the Microsoft COCO (Common Objects in Context) dataset, which contains a wide variety of everyday scenes with multiple objects. (2) **Referring Expressions:** For each image, there are several referring expressions provided by human annotators. These expressions describe specific objects or groups of objects in the image. (3) **Object Annotations:** Each referring expression is associated with an object annotation, a bounding box that identifies the location of the referred object in the image.

#### A.4 Visual Grounding

**Task Detail.** We perform the VR task based on the BEiT3 model [15]. In this task, the input example pair of the model is  $(v_0, v_1, t, y)$ , where  $y \in \{0, 1\}$ .  $y = 1$  means that the text matches at least one of two images. BEiT3 splits an example pair into two image-text pairs  $(v_0, t)$  and  $(v_1, t)$  as inputs, thereby extracting two fused embeddings. After concatenating the two embeddings and performing operations such as nonlinear projection, the final feature vector is obtained. This feature vector is fed into a binary classifier, whose weight matrix is  $\{\omega_0, \omega_1\}$ . At this point, we only need to guide the feature vector away from the guiding vector  $\omega_y$  through FGA, and simultaneously update the input images  $v_0, v_1$  along the gradient direction to obtain the adversarial images  $v'_0$  and  $v'_1$ .

**Dataset Detail.** NLVR2 [13] (Natural Language for Visual Reasoning for Real) is a natural language processing dataset designed

**Table 3: The experimental results when  $step = 7$ . The reported values are recall rates. Lower is better.**

$\epsilon (\ell_\infty)$	Test Retrieval			Image Retrieval		
	R@1	R@5	R@10	R@1	R@5	R@10
w/o atk	96.30	99.70	100.00	86.14	97.68	98.82
0.5	22.40	36.60	44.50	19.44	36.12	44.24
1	1.80	4.50	6.10	2.62	5.88	8.58
2	0.00	0.10	0.30	0.16	0.52	0.72
4	0.00	0.00	0.00	0.00	0.00	0.02
8	0.00	0.00	0.00	0.00	0.00	0.00

**Table 1: The experimental results when  $step = 1$ . The reported values are recall rates. Lower is better.**

$\epsilon (\ell_\infty)$	Test Retrieval			Image Retrieval		
	R@1	R@5	R@10	R@1	R@5	R@10
w/o atk	96.30	99.70	100.00	86.14	97.68	98.82
0.5	60.70	81.90	88.70	50.94	76.80	84.32
1	41.30	62.60	72.70	34.24	60.32	69.86
2	27.30	45.10	55.50	22.04	45.22	56.14
4	20.00	35.50	46.10	16.88	35.74	46.46
8	16.00	32.10	41.70	14.42	32.14	41.86
16	15.20	30.50	40.00	13.54	30.08	39.40

**Table 2: The experimental results when  $step = 3$ . The reported values are recall rates. Lower is better.**

$\epsilon (\ell_\infty)$	Test Retrieval			Image Retrieval		
	R@1	R@5	R@10	R@1	R@5	R@10
w/o atk	96.30	99.70	100.00	86.14	97.68	98.82
0.5	33.50	51.10	60.50	27.98	50.72	59.06
1	8.80	16.70	21.50	8.00	17.88	23.76
2	1.10	1.80	3.40	1.72	4.14	6.08
4	0.20	0.60	1.00	0.60	1.34	1.70
8	0.10	0.20	0.20	0.14	0.52	0.82
16	0.00	0.10	0.10	0.06	0.10	0.16

for the visual reasoning task. It aims to evaluate models' ability to reason about visual information combined with natural language descriptions. NLVR2 is an extended version of the NLVR dataset, featuring more images and more complex language descriptions. The NLVR2 dataset contains approximately 107,000 human-written sentences describing visual relationships in a set of images. Each sample includes a sentence and a pair of images. The content described in the sentence may match one of the images, both, or neither. The task for models is to determine whether the sentence correctly describes at least one of the images. This dataset is used for various vision-language tasks, such as visual question answering, image-text matching, and multimodal reasoning. NLVR2 advances the research and development of vision-language models' reasoning

capabilities by providing more challenging samples and complex language descriptions.

## B Combine FGA with unimodal attacks

We design FGA as a universal attack strategy, which is theoretically orthogonal to all unimodal attack schemes.

### B.1 Global Perturbation

This subsection discusses how to generate global perturbations based on FGA. We denote  $\delta$  as the added adversarial perturbation, with  $B(\epsilon, p) = \{\delta : \|\delta\|_p \leq \epsilon\}$  representing the ball of perturbations bounded by  $\epsilon$  in  $p$ -norm. Finding  $\delta$  typically can be addressed through an iterative process [9], which can be summarized as three phases: obtaining gradient information (Eq 1), determining the steepest ascent direction (Eq 2), and applying projection (Eq 7) [4].

#### (1) Obtaining gradient information:

$$g = \nabla_{\delta^{(i)}} L_{gui}(v + \delta^{(i)}, W, Y) \quad (1)$$

where  $\delta^{(i)}$  represents the perturbation at the  $i$ th iteration,  $W$  represents the guidance vectors, and  $Y$  represents the guidance labels. For simplicity, hereafter it is abbreviated as  $L_{gui}(v)$ .

#### (2) Determining the steepest ascent direction:

$$g^{(p)} = \text{Dir}_p(g) \quad (2)$$

where  $g$  represents the original gradient information, and  $g^{(p)}$  is a unit vector under  $\ell_p$  constraint, with  $\|g^{(p)}\|_p = 1$ . So that  $g^{(p)}$  represents the fastest loss rising direction under the  $\ell_p$  norm constraint. The steepest ascent directions for  $\ell_1$  [14],  $\ell_2$ , and  $\ell_\infty$  [4] are as follows:

$$e_i = \begin{cases} \text{sign}(g_i) & |g_i| \geq |g|^{(q)} \\ 0 & |g_i| < |g|^{(q)} \end{cases} \quad (3)$$

$$g^{(1)} = e / \|e\|_1 \quad (4)$$

$$g^{(2)} = g / \|g\|_2 \quad (5)$$

$$g^{(\infty)} = \text{sign}(g) \quad (6)$$

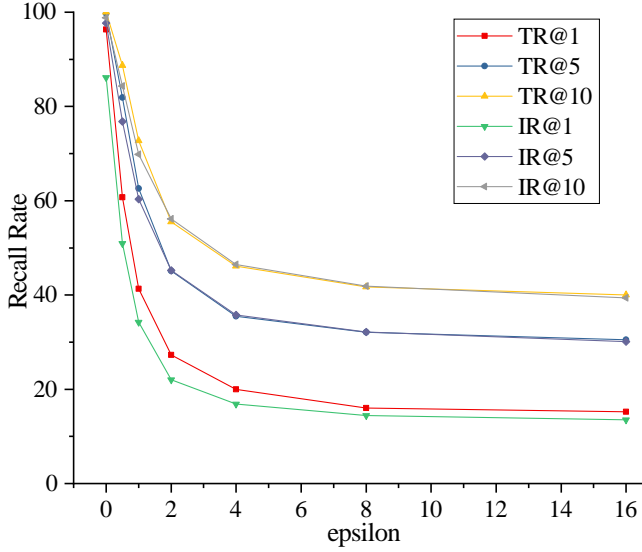
where  $|g|^{(q)}$  denotes the  $q^{th}$  percentile of  $|g|$ .

#### (3) Applying projection:

$$\delta^{(i+1)} = \text{Clamp}_{(-v, 1-v)} P_{B(\epsilon, p)}(\delta^{(i)} + \alpha \cdot g^{(p)}) \quad (7)$$

**Table 4: The experimental results when  $step = 10$ . The reported values are recall rates. Lower is better.**

$\epsilon (\ell_\infty)$	Test Retrieval			Image Retrieval		
	R@1	R@5	R@10	R@1	R@5	R@10
w/o atk	96.30	99.70	100.00	86.14	97.68	98.82
0.5	20.30	32.30	39.00	17.00	32.30	39.84
1	1.00	3.40	4.60	1.96	4.62	5.96
2	0.00	0.00	0.00	0.02	0.16	0.32
4	0.00	0.00	0.00	0.00	0.00	0.00



**Figure 1: The experimental results when  $step = 1$ . We observe that as the noise constraint is relaxed (with  $\epsilon$  increasing), the effectiveness of the attack gradually intensifies. However, the rate of decline in model performance slows down, indicating that the attack strength tends to converge.**

where  $P_{B(\epsilon,p)}$  ensures that  $\|\delta^{(i+1)}\|_p \leq \epsilon$ , and  $Clamp_{(-v,1-v)}$  ensures that the pixel values of  $v + \delta^{(i+1)}$  remain within the legal range  $[0, 1]$ . To elaborate further, when moving  $\delta^{(i)}$  along the steepest ascent direction with a step size of  $\alpha$ , it may lead to  $\|\delta^{(i)} + \alpha \cdot g^{(p)}\|_p > \epsilon$ . In such case, the projection algorithm is required to ensure  $\|P_{B(\epsilon,p)}(\delta^{(i)} + \alpha \cdot g^{(p)})\|_p = \epsilon$ .

$$P_{B(\epsilon,2)}(\delta) = \begin{cases} \epsilon \cdot \frac{\delta}{\|\delta\|_2} & \text{if } \|\delta\|_2 > \epsilon \\ \delta & \text{if } \|\delta\|_2 \leq \epsilon \end{cases} \quad (8)$$

$$P_{B(\epsilon,\infty)}(\delta) = Clamp_{(-\epsilon,\epsilon)}(\delta) \quad (9)$$

where  $Clamp_{(-\epsilon,\epsilon)}$  represents clipping each element value in  $\delta$  to be between  $-\epsilon$  and  $\epsilon$ . Besides,  $P_{B(\epsilon,1)}$  involves a complex projection strategy for sparsity  $\ell_1$  perturbation, discussed in detail in APGD $_{\ell_1}$  [2] and MAX[14].

Based on what is mentioned above, we can execute global perturbation attacks FGA $_{\ell_1}$ , FGA $_{\ell_2}$ , FGA $_{\ell_\infty}$  according to different norm constraints. In the unimodal domain, multi-norm attacks are very necessary. This is because a classic defence strategy in the unimodal domain, adversarial training, often overfits adversarial examples of a certain norm. That is, it can effectively defend against adversarial examples of a specific norm but is ineffective against adversarial examples of other norms [14]. Therefore, the interplay of adversarial perturbations across multiple norms can better explore the lower bounds of a network's robustness [10, 1, 12].

## B.2 Momentum Mechanism

The momentum mechanism is a commonly used strategy to enhance the robustness of adversarial examples. On top of the global

perturbation attack, it involves introducing momentum updates, during obtaining gradient information (Eq 1) [3].

**Obtaining gradient information with momentum mechanism:**

$$g \leftarrow \nabla_{\delta^{(i)}} L_{gui}(v + \delta^{(i)}) \quad (10)$$

$$g \leftarrow g / \text{mean}(\text{abs}(g)) \quad (11)$$

$$g \leftarrow g + \alpha \cdot g_m \quad (12)$$

$$g_m \leftarrow g \quad (13)$$

where  $\text{abs}$  represents taking the absolute value of each element in  $g$ , while  $\text{mean}$  denotes calculating the average of all element values.  $g_m$  is initialized as an all-zero matrix, incorporating gradient information from previous iterations. Therefore, after introducing the momentum mechanism, the gradient information comes from the weighted sum of current gradient  $g$  and past gradient  $g_m$ , with  $g_m$  weighting  $\alpha$ .

## B.3 Patch Perturbation

Global attacks constrain the perturbation  $\delta$  through  $\epsilon$ , requiring the perturbation to be as small as possible to avoid human detection. Patch attacks, on the other hand, use a binary mask matrix  $m$  to specify the patch's location information. Patch attacks concentrate the perturbation within a specified area of the image, typically a square, covering about 2% of the original image's area [6]. Within this area, there's no need to limit the size of the perturbation, so no norm constraints are necessary. It's only required to ensure that the patch's pixel values are within the legal range  $[0, 1]$ . Patch attack is also typically carried out in an iterative form:

$$g = \nabla_{\delta^{(i)}} L_{gui}(v \odot (1 - m) + \delta^{(i)} \odot m) \quad (14)$$

$$\delta^{(i+1)} = Clamp_{(0,1)}(\delta^{(i)} + g) \quad (15)$$

where  $\odot$  denotes the element-wise product, in the mask  $m$ , an element value of 0 indicates that the original pixel at that position is replaced by a patch pixel.

## C More ablation experiments

We perform ablation studies focusing on the iteration count ( $step$ ) and the intensity of noise ( $\epsilon$ ) leveraging the BEiT3 model configured for the Image-Text Retrieval (ITR) task. The experimentation utilizes the BEiT3 model, which has been specifically fine-tuned utilizing the Flickr30k dataset, and evaluates its performance against the same dataset. Our methodology involves deploying the FGA while adhering to the  $\ell_\infty$  norm constraint, denoted as FGA $_{\ell_\infty}$ . The experimental setup varies the  $step$  parameter across a set  $\{1, 3, 7, 10\}$ , with corresponding outcomes detailed in Tab 1, Tab 2, Tab 3 and Tab 4, respectively. Concurrently, we explore a range of  $\epsilon$  values set at  $\{0.5, 1, 2, 4, 8, 16\}$ , ensuring that  $\|\delta\|_\infty \leq \epsilon$ , where the  $\epsilon$  values are pre-normalization, indicating that image pixel values span a  $[0, 255]$  range. We observe that: (1) As the noise constraint is relaxed (with  $\epsilon$  increasing), the effectiveness of the attack gradually intensifies (Fig 1). However, the rate of decline in model performance slows down, indicating that the attack strength tends to converge. (2) As the number of iterations ( $step$ ) increases, the attack's effectiveness progressively intensifies (Fig 2).

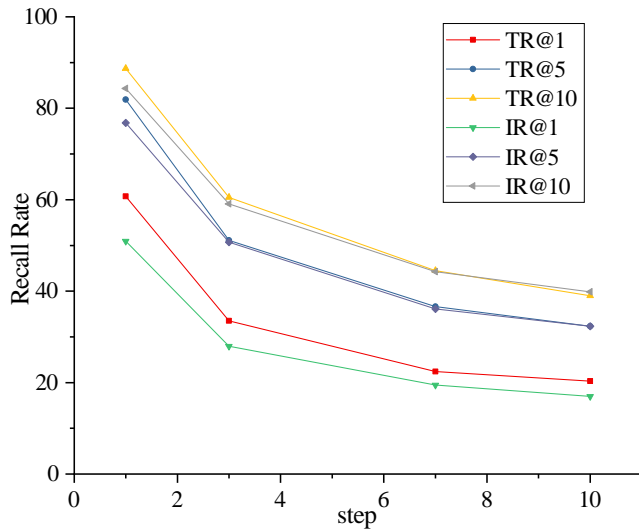


Figure 2: The experimental results when  $\epsilon = 0.5$ . We note that as the number of iterations increases, the attack’s effectiveness progressively intensifies. Nonetheless, the decrease in model performance decelerates, suggesting a convergence in attack potency.

## References

- [1] Francesco Croce and Matthias Hein. 2022. Adversarial robustness against multiple and single  $l_p$ -threat models via quick fine-tuning of robust classifiers. In *Proceedings of the International Conference on Machine Learning (ICML)*, 4436–4454.
- [2] Francesco Croce and Matthias Hein. 2021. Mind the box:  $l_1$ -apgd for sparse adversarial attacks on image classifiers. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2201–2211.
- [3] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. 2018. Boosting adversarial attacks with momentum. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 9185–9193.
- [4] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations (ICLR)*.
- [5] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2017. Making the v in vqa matter: elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 6904–6913.
- [6] Danny Karmon, Daniel Zoran, and Yoav Goldberg. 2018. LaVAN: localized and visible adversarial noise. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2507–2515.
- [7] Junnan Li, Ramprasaath Selvaraju, Akhilesh Gotmare, Shafiq Joty, Caiming Xiong, and Steven Chu Hong Hoi. 2021. Align before fuse: vision and language representation learning with momentum distillation. In *Proceedings of the 35th International Conference on Neural Information Processing Systems (NeurIPS)*, 9694–9705.
- [8] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014. Microsoft COCO: common objects in context. In *Proceedings of the 13th European Conference on Computer Vision (ECCV)*, 740–755.
- [9] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2018. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations (ICLR)*.
- [10] Pratyush Maini, Eric Wong, and J. Zico Kolter. 2020. Adversarial robustness against the union of multiple perturbation models. In *Proceedings of the International Conference on Machine Learning (ICML)*, 6640–6650.
- [11] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. 2017. Grad-cam: visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 618–626.
- [12] Gaurang Sriramanan, Maharshi Gor, and Soheil Feizi. 2022. Toward efficient robust training against union of  $l_p$  threat models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems (NeurIPS)*, 25870–25882.
- [13] Alane Suhr, Stephanie Zhou, Ally Zhang, Iris Zhang, Huajun Bai, and Yoav Artzi. 2019. A corpus for reasoning about natural language grounded in photographs. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, 6418–6428.
- [14] Florian Tramèr and Dan Boneh. 2019. Adversarial training and robustness for multiple perturbations. *Advances in neural information processing systems*, 32.
- [15] Wenhui Wang, Hangbo Bao, Li Dong, Johan Björck, Zhiliang Peng, Qiang Liu, Kriti Aggarwal, Owais Khan Mohammed, Saksham Singhal, Subhojit Som, et al. 2023. Image as a foreign language: BEiT pretraining for vision and vision-language tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 19175–19186.
- [16] Ning Xie, Farley Lai, Derek Doran, and Asim Kadav. 2019. Visual Entailment: a novel task for fine-grained image understanding. In *arXiv preprint:1901.06706*.
- [17] Licheng Yu, Patrick Poirson, Shan Yang, Alexander C. Berg, and Tamara L. Berg. 2016. Modeling context in referring expressions. In *Proceedings of the 14th European Conference on Computer Vision (ECCV)*, 69–85.