

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

Haonan Zheng

zhenghaonan@mail.nwpu.edu.cn
Northwestern Polytechnical University
School of Electronics and Information
Xi'an, Shaanxi, China

Wen Jiang*

Northwestern Polytechnical University
School of Electronics and Information
Xi'an, Shaanxi, China

Xinyang Deng

Northwestern Polytechnical University
School of Electronics and Information
Xi'an, Shaanxi, China

Wenrui Li

liwr618@163.com
Harbin Institute of Technology
Department of Computer Science and Technology
Harbin, China

Abstract

With Vision-Language Pre-training (VLP) models demonstrating powerful multimodal interaction capabilities, the application scenarios of neural networks are no longer confined to unimodal domains but have expanded to more complex multimodal V+L downstream tasks. The security vulnerabilities of unimodal models have been extensively examined, whereas those of VLP models remain challenging. We note that in CV models, the understanding of images comes from annotated information, while VLP models are designed to learn image representations directly from raw text. Motivated by this discrepancy, we developed the Feature Guidance Attack (FGA), a novel method that uses text representations to direct the perturbation of clean images, resulting in the generation of adversarial images. FGA is orthogonal to many advanced attack strategies in the unimodal domain, facilitating the direct application of rich research findings from the unimodal to the multimodal scenario. By appropriately introducing text attack into FGA, we construct Feature Guidance with Text Attack (FGA-T). Through the interaction of attacking two modalities, FGA-T achieves superior attack effects against VLP models. Moreover, incorporating data augmentation and momentum mechanisms significantly improves the black-box transferability of FGA-T. Our method demonstrates stable and effective attack capabilities across various datasets, downstream tasks, and both black-box and white-box settings, offering a unified baseline for exploring the robustness of VLP models.

Keywords

Vision-Language Models; Adversarial Attack; Transferability

1 Introduction

ViT provides an effective Transformer-based encoder for the visual modality [14], ensuring the feature extraction of multimodal input through a unified encoding manner, significantly advancing the Vision-and-Language tasks [16, 1, 47, 52]. Various VLP models [23, 51, 45, 40] continually improve performance in V+L downstream tasks through diverse pre-training tasks and architectural designs [2, 45, 20, 27, 29]. However, the previous research in unimodal fields such as Computer Vision (CV) and Natural Language Processing

(NLP) highlights the vulnerability of neural networks to adversarial attacks [15, 24]. Although adversarial robustness, particularly in CV, has been extensively explored in terms of attack strategies [4, 36], defence mechanisms [34], and transferability [12, 46], the study of adversarial robustness in VLP models remains challenges [26, 54, 55, 33]. Our study aims to develop a unified architecture to explore commonalities between multimodal and unimodal tasks from the perspective of adversarial attacks. In other words, we seek to bridge the gap, allowing rich findings in unimodal adversarial robustness to be directly applied to the multimodal scenario.

The first question we consider is "Which modality should be paid more attention?" We primarily focus on perturbations in the image modality, with perturbations in the text modality serving as orthogonal (1) **Semantic consistency**: Visual adversarial examples maintain semantic consistency, i.e., noise addition within reasonable limits doesn't change human comprehension. Conversely, text adversarial examples risk semantic distortion, potentially introducing spelling errors. (2) **Differentiability**: Image inputs are continuous and differentiable, unlike text tokens which are discrete and non-differentiable making text-only attacks less effective. (3) **Accessibility**: In real-world scenarios, text often serves as the primary means of interaction between users and AI models, with limited opportunities for attackers to modify user-generated text. In contrast, models can automatically acquire image data, simplifying the process for attackers to introduce perturbations. In fact, [33] also primarily focuses on enhancing attack strategies in the visual modality to improve adversarial transferability and [55] does not involve text attacks.

The second question is "How to unify multimodal and unimodal scenarios in exploring adversarial robustness?" We conceptualize image adversarial attacks as a feature-guided process. For unimodal models which primarily learn to understand images through detailed annotation information (such as category labels), attacking an image involves steering its embedding away from the feature vector linked to its correct annotation [15]. This deviation induces a biased comprehension of the image within the network. Alternatively, the image embedding can be guided closer to the feature vector associated with an incorrect annotation, thereby leading the network to make a predetermined error [25]. In the multimodal scenario, models are encouraged to understand images from raw

*Corresponding author

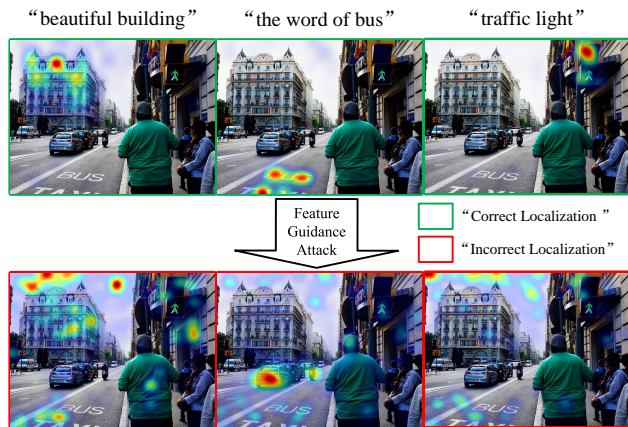


Figure 1: ALBEF computes Grad-CAM[41] visualizations on the self-attention maps. Before FGA, ALBEF can accurately localize image content based on textual cues. After FGA, ALBEF’s understanding of the image becomes confused.

text, providing a broader and more accessible source of supervision [40]. This also offers more flexible guiding information for the adversarial attack. By guiding the image embedding away from the correct text description, we induce the VLP model to develop an incorrect understanding of the image itself. Similarly, directing the embedding towards an incorrect text description intentionally misleads the model into adopting a specific erroneous interpretation. This strategy is termed Feature Guidance Attack (FGA). Expanding upon FGA, we employ adversarial texts from text attacks as guiding information to generate adversarial images, thus obtaining a novel multimodal attack. This approach exacerbates the model’s misinterpretation called Feature Guidance with Text Attack (FGA-T). Furthermore, we introduce additional orthogonal mechanisms to enhance the adversarial transferability of FGA-T in the black-box scenario. Code: <https://github.com/LibertazZ/FGA>

Our contributions can be summarized as follows:

- We provide FGA, using original text as the supervision source for the adversarial attack on VLP models, inducing the network to misinterpret adversarial images.
- We introduce cross-modal interaction through adversarial text, forming a novel multimodal adversarial attack that enhances white-box attack strength, and improves black-box transferability through additional mechanisms.
- Our approach is theoretically orthogonal to any unimodal attack enhancement mechanism. Empirical evidence based on multiple datasets and VLP models demonstrates the broad applicability of our method to various V+L multimodal tasks, providing a unified baseline for the exploration of multimodal robustness.

2 Related Work

2.1 Unimodal Adversarial Attack

From an access perspective to the model, unimodal attacks can be divided into white-box and black-box attacks. In the black-box scenario, due to the target network’s opaque weights, attackers

typically conduct white-box attacks on an accessible source network, then transfer the adversarial examples to the target network. Therefore, the attack’s transferability is also crucial.

White-box Attack. Based on how to constrain perturbation, adversarial attacks on the visual modality can generally be divided into two categories. (1) Global attacks typically involve perturbing all pixels of an image, usually constraining the distance between adversarial and original images based on ℓ_∞ , ℓ_2 , or ℓ_1 norms. Representative methods include FGSM [15], PGD [34], APGD [10], CW [4], etc. (2) Patch attacks, which confine the perturbation to a small area, such as 2% of the image, and allow unrestricted modification of image pixels within that area. Representative methods include LaVAN [19] and Depatch [7]. Since patch attacks are more practical, physical world attacks are usually based on this form. In the text domain, due to the discrete nature of text data, such attacks typically involve subtle modifications to the original text, such as replacing synonyms, inserting additional words, or adjusting sentence structure, without significantly altering the meaning of the text. A representative method is BertAttack [24].

Boosting Transferability. Enhancing the transferability of adversarial attacks is essentially a generalization problem. The two main approaches to solving the generalization issue are data augmentation and improving the optimization algorithm, thus dividing transfer attack methods into two categories. (1) Typical methods that boost transferability through data augmentation, such as DI [49] (Diverse Inputs), TI [13] (Translation Invariant) and SI [30] (Scale Invariant). (2) Typical schemes that improve optimization algorithm, such as MI [12] (Momentum Iterative), NI [30] (Nesterov Iterative), VMI [46] (Varied Momentum Iterative), VNI [46] (Varied Nesterov Iterative).

2.2 Multimodal Adversarial Attack

This subsection discusses relevant VLP models and multimodal adversarial attack methods.

VLP Models. VLP models based on different combinations of pre-training tasks can be roughly divided into three categories. (1) Aligned models: CLIP [40] contains two unimodal encoders to align multimodal embeddings based on Image-Text Contrastive (ITC) loss. (2) Fused models by matching: ViLT [20] introduces both Image-Text Matching (ITM) and Masked Language Modeling (MLM) pre-training for V+L tasks. Models like ALBEF [23], TCL [51], BLIP [22], and VLMO [3] build on it, first aligning multimodal features using ITC loss, then fusing cross-modal features using ITM and MLM losses. (3) Fused models by Masked Data Modeling (MDM): BEiT [2] and BEiTV2 [38] propose and improve Masked Image Modeling (MIM) loss. BEiT3, based on it, first aligns multimodal features using ITC loss, then fuses cross-modal features using MIM, MLM, and Masked Language-Vision Modeling (MLVM) losses.

Multimodal Attack. Attacking VLP models is a novel topic. Existing work has provided valuable insights. Co-Attack [54] designs general optimization objectives based on different embeddings (unimodal or multimodal) and experimentally demonstrates that using text attacks or image attacks alone is not as effective as using both in combination, providing a general baseline for subsequent works. SGA [33] points out that improving the diversity of multimodal

interaction can enhance the transferability of multimodal adversarial examples. AdvCLIP [55] provides a framework to learn a universal adversarial patch on pre-trained models for transfer attacks on downstream fine-tuned models. These works are limited to VLP models and ignore the connection between unimodal and multimodal scenarios, which is our main motivation.

3 Methodology

3.1 Feature Guidance

An image feature extractor E (e.g., an image contrastive representation encoder [18, 5, 17] or a VLP model’s visual encoder) projects the image into a feature vector for various visual tasks like image classification and object detection. Without regarding the subsequent usage, the most intuitive approach to generate an adversarial example x' for an image x is to encourage the feature vectors $E(x')$ and $E(x)$ to be as distant as possible [54]. This universal strategy is termed “Feature Deviation Attack” (FDA), which involves maximizing the loss function:

$$L_{dev} = -E(x') \cdot E(x). \quad (1)$$

where \cdot represents the dot product of vectors, $E(x) \in \mathbb{R}^d$.

Assuming that in the embedding space, there exists a set of guiding vectors $W = \{\omega_i\}_{i=1}^m$, $\omega \in \mathbb{R}^d$, and there is a set of guiding labels $Y = \{y_i\}_{i=1}^n$, $y \in \{1, 2, \dots, m\}$ specifying that $E(x')$ should be distant from the guiding vectors $\{\omega_{y_i}\}_{i=1}^n \in W$. We refer to this strategy as “Feature Guidance Attack” (FGA). To realize the above concept, we need to maximize the loss function:

$$L_{gui} = -\frac{1}{n} \sum_{i=1}^n \ln \left(\frac{\exp(E(x') \cdot \omega_{y_i})}{\sum_{j=1}^m \exp(E(x') \cdot \omega_j)} \right) \quad (2)$$

where $\exp(\cdot)$ represents the exponential function with Euler’s number e as the base, and $\ln(\cdot)$ stands for the logarithm to the base e .

Based on L_{gui} or L_{dev} , we can apply the PGD process [34], gradually pushing the clean example x along the gradient direction to maximize the loss function, ultimately obtaining the adversarial example x' . By the chain rule of gradients, $\frac{\partial L}{\partial x'} = \frac{\partial L}{\partial E(x')} \cdot \frac{\partial E(x')}{\partial x'}$. $\frac{\partial L}{\partial x'}$ represents the direction of perturbation added to the input example. While we focus on $\frac{\partial L}{\partial E(x')}$ which represents the movement direction of the feature vector:

$$\frac{\partial L_{dev}}{\partial E(x')} = -E(x) \quad (3)$$

$$\frac{\partial L_{gui}}{\partial E(x')} = \sum_{i=1}^n \left(-\frac{1}{n} \cdot \omega_{y_i} \right) + \sum_{k=1}^m \frac{\exp(E(x') \cdot \omega_k)}{\sum_{j=1}^m \exp(E(x') \cdot \omega_j)} \cdot \omega_k \quad (4)$$

It can be observed that feature deviation loss promotes the movement of $E(x')$ towards $-E(x)$, which means moving away from $E(x)$. While, Regarding the first term of $\partial L_{gui} / \partial E(x')$, it encourages $E(x')$ to move away from the guiding vectors $\{\omega_{y_i}\}_{i=1}^n$, and assigning equal weight $1/n$ to each of them. The second term encourages $E(x')$ to approach the guiding vector $\omega_k \in W$, with a weight of $\exp(E(x') \cdot \omega_k) / \sum_{j=1}^m \exp(E(x') \cdot \omega_j)$, which means the closer $E(x')$ is to a guiding vector, the greater the weight assigned to it. Due to the presence of the first term, $E(x')$ is far from $\{\omega_{y_i}\}_{i=1}^n$, resulting in the weight of ω_{y_i} being almost zero in the second term.

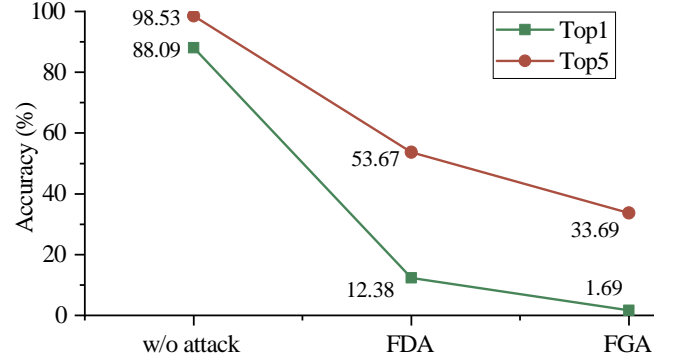


Figure 2: Attacking results of SimCLR encoder on CIFAR-10. The reported value is classification accuracy.

Consequently, the second term effectively facilitates $E(x')$ in selecting a nearby guiding vector that does not belong to the set $\{\omega_{y_i}\}_{i=1}^n$ and moving closer to it.

We conduct a simple attack experiment using the SimCLR image encoder [5] and the CIFAR-10 dataset [21], where image feature vectors are used for image classification through a KNN-200 classifier. During the feature guidance attack, we first use the encoder to extract features for all training data. Then, by averaging the features belonging to the same category, we obtain ten guiding vectors $\{\omega_1, \omega_2, \dots, \omega_{10}\}$. The label $y \in \{1, 2, \dots, 10\}$ of the image x serves as the guiding label, encouraging $E(x')$ to move away from the guiding vector ω_y . From Table 2, it is not difficult to observe that the intensity of the feature guidance attack is greater than the feature deviation attack.

Most existing VLP models typically consist of two unimodal encoders, a text encoder E_t and a visual encoder E_v , along with a multimodal feature fusion encoder E_m . For a single image-text paired example (v, t) , it is first mapped to a shared feature space separately by E_t and E_v for aligning image and text features. Subsequently, cross-modal feature fusion is conducted through E_m . Therefore, VLP models focus on three key embeddings: $E_v(v)$, $E_t(t)$, and $E_m(E_v(v), E_t(t))$, all corresponding to the [CLS] vector. We will focus on finding guiding vectors in the embedding space and constructing guiding labels to execute FGA on VLP models.

3.2 Attacking after Fuse

In this scenario, we focus on the fused embedding $E_m(E_v(v), E_t(t))$. For different V+L downstream tasks, it needs to be fed into different subsequent models which can be uniformly understood as comprising a projector P and a linear classification head h . P projects the fused embedding into a task-specific downstream embedding space, followed by h performing classification on this embedding. We can rewrite $P(E_m(E_v(v), E_t(t)))$ as $E(v|t)$, obtaining an image encoder conditioned on the textual modality. The weight of the linear classification head, $W = \{\omega_i\}_{i=1}^c \in \mathbb{R}^{d \times c}$ (c is the number of categories), serve as guiding vectors. Using label information as guiding labels Y , we thus have all the necessary components to implement the Feature Guidance Attack. See Appendix A for more details and explanations about Visual Question Answering (VQA), Visual Reasoning (VR), etc.

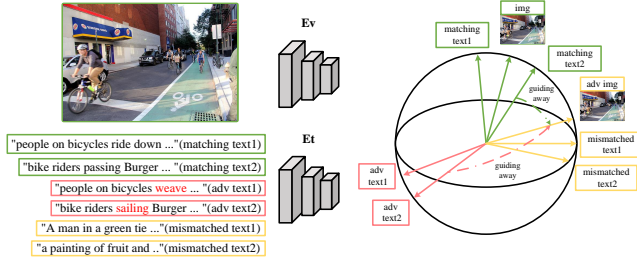


Figure 3: Illustration of Feature Guidance with Text Attack (FGA-T) before fuse.

FGA is primarily used to generate image adversarial examples. However, for VLP models, attacking both modalities simultaneously is a more effective strategy [54, 33]. The main challenge in generating adversarial texts involves solving the following optimization problem:

$$t' = \underset{t'}{\operatorname{argmax}} (\|E_m(E_v(v), E_t(t')) - E_m(E_v(v), E_t(t))\|) \quad (5)$$

where BertAttack [24] is a well-suited choice for addressing this problem.

In fact, FGA and BertAttack are completely orthogonal strategies. This means we first generate an adversarial text example t' and then use $E(v|t')$ as the image encoder to perform FGA, obtaining v' . Thus, we acquire the adversarial pair (v', t') .

3.3 Attacking before Fuse

In this scenario, only two unimodal encoders are used: E_v , and E_t . The primary intention of VLP models is to learn directly from raw text descriptions of images, utilizing a broader source of supervision [40]. This implies that in CV models, image understanding comes from pre-provided labels, such as image categories, pixel categories, or annotated bounding boxes. In contrast, in VLP models, the understanding of images originates from raw text. Consequently, using text as supervisory information to generate image adversarial examples becomes a natural approach. To implement this approach, we first acquire a text set $T = \{t_i\}_{i=1}^m$. Then, We use the text encoder to obtain a set of guiding vectors $\{\omega_i\}_{i=1}^m = \{E_t(t_i)\}_{i=1}^m$. To obtain this text set T , all texts are gathered from the dataset. Here, by utilizing the dataset's annotations, we can identify which texts in the text set match with the image v , thereby obtaining the guiding labels Y . By this point, all elements necessary for executing FGA have been acquired: the image encoder E_v , the set of guiding vectors $\{E_t(t_i)\}_{i=1}^m$, and the guiding labels Y . By maximizing L_{gui} , the feature vector $E_v(v')$ will diverge from the text representations $\{E_t(t_y)\}_{y \in Y}$ that match v , thereby generating adversarial images v' . For more details on executing iterations of FGA and how it can be combined with typical attack strategies in the unimodal domain, refer to Appendix B.

3.4 Boosting Transferability before Fuse

SGA[33] points out that multimodal adversarial examples have better transferability than unimodal adversarial examples. Therefore, we need to introduce text attack into FGA before fuse. We consider an image minibatch $V = \{v_i\}_{i=1}^n$ and the text set T_i represents all

texts that match with the image v_i . Firstly, for each text $t \in T_i$, we handle the following optimization problem to generate adversarial text t' :

$$t' = \underset{t'}{\operatorname{argmax}} \left(\frac{E_t(t') \cdot E_v(v_i)}{\|E_t(t')\|_2 \|E_v(v_i)\|_2} \right) \quad (6)$$

where $\|\cdot\|_2$ denotes the Euclidean distance.

At this point, we obtain the adversarial text set T'_i , and we denote $T = T_1 \cup T_2 \dots \cup T_n \cup T'_1 \cup T'_2 \dots \cup T'_n$. Secondly, for $v_i \in V$, where T_i is its matching texts and T'_i is the adversarial texts, to generate adversarial example v'_i , we use the feature guidance loss to encourage $E_v(v'_i)$ to simultaneously move away from both $E_t(T_i)$ and $E_t(T'_i)$:

$$L_{gui}(v'_i) = -\frac{1}{len_i} \sum_{t^* \in T_i \cup T'_i} \log \left(\frac{\exp(E_v(v'_i) \cdot E_t(t^*))}{\sum_{t \in T} \exp(E_v(v'_i) \cdot E_t(t))} \right) \quad (7)$$

len_i represents the length of text set $T_i \cup T'_i$.

Building on this foundation, we further introduce two strategies to enhance transferability: (1) Following SGA [33], we preset a set of resize parameters $S = \{s_1, s_2, \dots, s_m\}$, where $h(v, s_k)$ denotes the resizing function that takes the image v and the scale coefficient s_k as inputs. After data augmentation, the objective function we aim to maximize is no longer $L_{gui}(v'_i)$ but rather $\sum_{k=1}^m L_{gui}(h(v'_i, s_k))$, where $h(v'_i, s_k)$ represents the augmented image. (2) Following MIFGSM [12], we introduce the momentum mechanism, where the current perturbation direction is determined by both the current gradient and the historical gradients from previous iterations. See Appendix B.2 for more details.

4 Experiments

4.1 Experimental Setting

4.1.1 VLP Models. Our experimental section involves four typical VLP models: CLIP, ALBEF, TCL and BEiT3. CLIP is a typical aligned model, consisting solely of two unimodal encoders. The latter three are fused models, containing two unimodal encoders and a multimodal encoder. ALBEF and TCL share the same architecture with some differences in the details of ITC loss. Besides, ALBEF and BEiT3 have two main differences: (1) **Different Pre-training Tasks:** ALBEF is based on three pre-training tasks: ITC, ITM and MLM. In contrast, BEiT3 is based on three MDM tasks: MLM, MIM, and MVLM. (2) **Different Model Structures:** In ALBEF, the three encoders are independent of each other. BEiT3, however, uses the Multiway Transformer to split the feed-forward layer into three parallel paths, thereby obtaining three encoders.

4.1.2 V+L Downstream Tasks. In this part, we will introduce each downstream task involved in the experiments, along with the models and datasets used to perform these tasks.

Visual Entailment (VE) is a fine-grained visual reasoning task, where given a pair of (v, t) , the model needs to determine whether the text is supported by the image (entailment), whether the text is unrelated to the image (neutral), or whether the text contradicts the content of the image (contradictory). This task will be conducted based on the ALBEF model and the SNLI-VE [50] dataset.

Visual Question Answering (VQA) requires the model to predict a correct answer given an image and a question [16, 37]. It can be viewed as a multi-answer classification problem, or as an answer

Table 1: Comparison results on four downstream tasks after fuse. The reported value is accuracy. Lower is better.

Method	VQA			VG			VR	
	test	dev	std	val	testA	testB	dev	test-P
w/o atk	79.91	75.83	76.04	58.44	65.91	46.25	83.54	84.38
TA	55.09	45.47	45.89	49.17	54.05	39.27	69.59	70.46
IA	42.72	52.78	52.88	45.78	51.48	36.16	63.10	63.14
SA	38.42	41.21	41.31	42.13	45.93	34.96	58.43	58.53
CA	19.36	36.91	37.01	36.61	39.87	30.21	54.77	54.67
FGA	5.66	48.70	48.77	36.54	42.18	29.33	0.93	1.15
FGA-T	2.78	35.46	35.70	34.11	38.16	28.86	0.52	0.70
FGA ¹	39.05	60.66	60.65	41.68	45.09	36.41	27.60	28.79
FGA-T ¹	22.37	41.00	41.07	35.54	39.38	30.44	19.15	20.49
FGA _{ℓ₁}	8.26	53.47	53.54	38.70	44.88	30.76	1.46	1.74
FGA _{pat}	5.23	51.24	51.22	55.92	64.23	45.26	7.59	8.43

generation problem. We use the VQAv2 [16] dataset and the ALBEF model, which performs the VQA task through text generation.

Visual Grounding (VG) requires the model to find parts of the image that match the given textual description. We perform this task based on the RefCOCO+ [53] dataset and the ALBEF model.

Visual Reasoning (VR) requires the model to predict if a given text describes a pair of images. This task necessitates that the model not only understands the content of individual images but also compares and reasons about the relationship between two images. Therefore, the input consists of a pair of images and a piece of text. We use the BEiT3 model and the NLVR2 [43] dataset to perform this task.

Zero-Shot Classification (ZC) requires using predefined category descriptions (such as "a cat," "a car," etc.) as text inputs and mapping these descriptions to the embedding space by text encoder. Then, for a given image, the similarity between the image embedding and each category description embedding is calculated, and the image is classified into the category with the highest similarity. Due to the CLIP model's strong zero-shot capacity, we use it along with three datasets: CIFAR-10 [21], CIFAR-100 [21], and ImageNet [11], to perform this task.

Image-Text Retrieval (ITR) involves retrieving relevant images from an image database given a text query, and vice versa [52, 32, 48]. We perform this task based on the CLIP, ALBEF and TCL models, and the Flickr30k [39] and MS COCO [31] datasets.

4.2 Attack Effectiveness after Fuse

This subsection explores the effectiveness of attacks on the fused feature vector. Since VE, VQA, VG, and VR rely on this vector, we choose to evaluate these four tasks. As Table 1 illustrates, the evaluation follows the baseline set by [54], TA represents alone text attack using BertAttack. IA represents image attack based on feature deviation loss. SA stands for separate unimodal attack, indicating that TA and IA are executed separately without modal interaction, and CA denotes the multimodal white-box attack Co-Attack [54] which introduces cross-modal interaction. For fairness, we set the ℓ_∞ perturbation constraint for the image modality in

Table 2: Comparison results on ZC task before fuse on CLIP. The reported value is accuracy. Lower is better.

Metric	Method	CIFAR-10	CIFAR-100	ImageNet
Top1	w/o atk	89.24	64.76	62.308
	IA	35.5	17.04	15.29
	FGA	0.01	0.0	0.004
	FGA ¹	30.10	12.21	5.46
	FGA _{ℓ₁}	14.5	7.06	0.028
	FGA _{pat}	0.1	0.0	0.01
Top5	w/o atk	98.94	86.9	86.78
	IA	78.21	34.89	31.36
	FGA	10.54	0.56	0.468
	FGA ¹	79.26	34.39	27.12
	FGA _{ℓ₁}	24.59	9.06	0.506
	FGA _{pat}	9.66	0.35	0.708

FGA and FGA-T to $\epsilon = 2/255$ with 10 iterations, consistent with IA, SA, and CA. Additionally, we explore the attack effectiveness when the number of iterations is 1, i.e., single-step attack, namely FGA¹ and FGA-T¹. We also investigate the effectiveness under the ℓ_1 constraint, namely FGA_{ℓ₁}, with $\epsilon_{\ell_1} = 255$ and 20 iterations. (See Appendix B.1 for more details.) Besides, we perform FGA in patch form, namely FGA_{pat}, with 100 iterations, a single-step ℓ_∞ constraint of $\alpha = 8/255$, and a patch area of 2% of the total image area, with a random location. (See Appendix B.3 for more details.) Furthermore, when involving text attack, BertAttack is used with a restriction of 1 perturbable token, following [54, 33].

We can observe from Table 1: (1) Under all tasks, FGA-T consistently achieves the best white-box attack performance, validating the effectiveness of the feature guidance approach and its orthogonality with text attack. (2) Even with only a single step, the feature guidance method is sufficient to produce effective adversarial examples, performing on par with or even better than the baseline. This provides a faster and more convenient attack strategy. (3) The feature guidance approach exhibits good orthogonality with other attack strategies in Computer Vision. When combined with ℓ_1 attack or patch attack, it demonstrates strong performance.

4.3 Attack Effectiveness before Fuse

In this subsection, we explore the effectiveness of attacks on two unimodal encoders. The tasks of ZC and ITR primarily rely on two unimodal embeddings. We first conduct attacks on the ZC task. Since the text input in the ZC task is predefined and cannot be altered, we only use attacks involving the visual modality. When conducting FGA, we construct the text "A photo of a {object}." using the categories' name to obtain the text set $T = \{t_i\}_{i=1}^c$, where c represents the number of categories, following [40]. We extract features through E_t to construct the guiding vectors $\{E_t(t_i)\}_{i=1}^c$, and the true category of the image serves as the guiding label $y \in \{1, 2, \dots, c\}$. The attack results are presented in the Table 2. We observe that even FGA¹ outperforms IA which is an iterative feature deviation attack.

When executing the ITR task on the CLIP model with ViT image encoder, to construct guiding vectors for FGA, we use not only all

Table 3: Comparison results on image-text retrieval before fuse on CLIP. For text-retrieval (TR) and image-retrieval (IR), R@1, R@5 and R@10 are reported respectively. Lower is better.

Method		Flickr30k(1K test set)						MSCOCO(5K test set)					
		TR			IR			TR			IR		
		R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10
w/o attack		81.5	96.3	98.4	62.08	85.62	91.7	52.42	76.44	84.42	33.02	58.16	68.4
Feature Deviation	TA	61.8	85.8	92.0	41.18	66.68	76.78	27.42	51.06	62.54	16.40	34.49	44.52
	IA	25.6	47.6	56.4	19.84	39.18	48.94	10.84	24.16	32.06	6.76	17.76	24.71
	SA	17.2	35.7	45.9	11.14	25.32	33.32	5.8	14.22	19.8	3.21	9.39	14.06
	CA	7.3	16.5	22.4	4.18	10.04	13.86	1.6	4.62	7.08	0.94	2.89	4.53
MS COCO Categories	FGA ¹	68.9	89.4	93.5	49.66	75.08	83.48	40.46	64.54	74.08	23.89	46.57	57.68
	FGA	16.6	32.2	39.6	11.54	25.78	33.68	5.02	11.06	14.78	2.76	7.37	10.43
ImageNet Categories	FGA ¹	66.8	87.5	92.5	48.48	73.94	82.66	39.80	63.03	73.04	23.67	46.07	57.33
	FGA	11.5	21.0	28.3	7.64	17.8	23.1	3.54	8.28	11.44	2.25	6.02	8.68
Test Texts	FGA ¹	27.5	49.1	59.5	17.72	38.92	50.26	14.54	30.4	39.88	8.18	21.32	30.18
	FGA	0.0	0.8	1.6	0.14	0.44	0.96	0.06	0.24	0.4	0.024	0.152	0.264
	FGA _{f_l}	0.1	0.2	0.5	0.12	0.32	0.5	0.04	0.12	0.16	0.068	0.200	0.280
	FGA _{pat}	0.2	0.4	0.4	0.18	0.48	0.78	0.08	0.16	0.24	0.080	0.200	0.312

Table 4: Comparison results on image-text retrieval before fuse on ALBEF. For text-retrieval (TR) and image-retrieval (IR), R@1, R@5 and R@10 are reported respectively. Lower is better.

Method		Flickr30k(1K test set)						MSCOCO(5K test set)					
		TR			IR			TR			IR		
		R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10
w/o attack		95.9	99.8	100.0	85.5	97.5	98.9	77.58	94.26	97.16	60.67	84.33	90.51
Feature Deviation	TA	85.8	98.1	98.9	64.1	83.68	88.16	53.08	78.32	86.7	34.48	59.38	69.08
	IA	47.4	65.6	71.4	38.64	56.74	62.82	30.26	47.7	55.5	21.19	38.16	46.05
	SA	31.6	50.6	58.4	23.66	39.68	46.64	15.44	29.54	36.74	10.21	21.89	28.27
	CA	32.5	50.9	58.4	23.42	39.5	45.92	14.58	28.26	35.5	9.90	21.78	27.92
Test Texts	FGA ¹	38.0	58.0	65.3	31.26	52.12	60.72	27.96	48.76	57.88	20.79	41.26	51.40
	FGA	0.7	1.0	1.1	0.54	0.94	1.1	0.32	0.78	1.02	0.27	0.76	1.12

the texts in the dataset to construct the text set (“Test Texts” in Table 3), but also follow the approach of the ZC task: using the 1000 category names from the ImageNet dataset (“ImageNet Categories” in Table 3) or the 80 category names from the MS COCO dataset’s object detection task (“MS COCO Categories” in Table 3) to construct texts “There is a {object} in this photo.” to form the text set $T = \{t_i\}_{i=1}^c$. Since in the Flickr30k and MS COCO datasets, an image may contain multiple objects, it is possible that the image matches multiple texts in $\{t_i\}_{i=1}^c$. In fact, we do not have annotation information indicating which objects are in the image. Therefore, we compare the cosine similarity between $E_v(v)$ and $\{E_t(t_i)\}_{i=1}^c$ to find the top 5 texts with the highest cosine similarity to v . When performing FGA, we encourage $E_v(v')$ to move away from the feature vectors of these five texts. From Table 3, we can summarize: (1) When using all texts to construct the feature guidance vectors, FGA achieves the best attack effect, which is intuitive. Moreover, we find that without the text attack, CLIP is already incapacitated

on the ITR task. (2) ImageNet includes more categories and therefore contains richer guiding information, resulting in better attack effects compared to using categories from COCO.

Since the CLIP model only contains two unimodal encoders, attacking before fuse actually utilizes the entire CLIP model. However, the ALBEF model additionally includes a multimodal encoder, so attacking before fuse ignores the multimodal encoder. Therefore, it is necessary to validate the effectiveness of FGA before fusion on the ALBEF model. As shown in Table 4, we conduct this experiment based on the ALBEF model and the ITR task and observe phenomena consistent with Table 3.

4.4 Boosting Transferability

In this subsection, we transition the attack from the white-box setting to the black-box setting, which is a more common scenario. We use four VLP models: ALBEF, TCL, CLIP_{ViT}, and CLIP_{CNN}. The TCL model is identical to ALBEF except for differences in the

design of the Image-Text Contrastive (ITC) loss during training, resulting in different final network weights. The two CLIP models use ViT and CNN as visual encoders, respectively. The degree of difference between these four models varies, which will inevitably affect the transferability of adversarial examples. We will observe this phenomenon in the experiments. Our experimental setup is as follows: (1) **Task and Dataset:** We conduct black-box adversarial example transfer attacks based on the Image-Text Retrieval (ITR) task and the Flickr30k dataset. (2) **Source Model and Target Model:** The source model is the model for which we generate adversarial examples through white-box attacks, and then use them to attack the target model. Each model will serve as both source and target models. (3) **Attack Methods:** The methods we use involve attacking both image and text. SA and CA, which do not focus on transferability, serve as baselines. SGA is the state-of-the-art (SOTA) transfer attack and serves as the comparative method. FGA- T_{aug} is based on FGA-T with additional data augmentation using a set of resize parameters S , following SGA. The differences between SGA and FGA- T_{aug} are in the loss function used for generating adversarial images and the attack process (the former’s attack order is “text, image, text”, while the latter’s attack order is “text, image”). MFGA- T_{aug} additionally introduces the momentum mechanism. (4) **Hyperparameters:** All texts are allowed to modify only one word, all image perturbations are limited to $2/255$ (ℓ_∞ norm), and the number of iterations is 10, following [54]. The resize parameters $S = \{0.5, 0.75, 1.25, 1.5\}$, following SGA.

The experimental results are shown in Table 5. We observe the following phenomena: (1) SA, CA, and SGA attack the visual modality based on feature deviation. SGA designs a more advanced set-level feature deviation and introduces data augmentation, improving both white-box and black-box attack effects on the baseline. (2) FGA- T_{aug} based on feature guidance, improves SGA further, simultaneously enhancing both white-box and black-box attack effects again. (3) MFGA- T_{aug} slightly reduces the white-box attack effect but further improves adversarial transferability, which is consistent with the observations in [12]. (4) Attacks based on ALBEF transfer better to TCL than to CLIP because ALBEF and TCL only have differences in parameters, while ALBEF and CLIP are completely different models. The same logic applies to attacks based on TCL. (5) Attacks based on CLIP $_{ViT}$ transfer better to CLIP $_{CNN}$ than to ALBEF or TCL because the model difference between CLIP $_{ViT}$ and CLIP $_{CNN}$ is obviously smaller than the difference with ALBEF or TCL. The same logic applies to attacks based on CLIP $_{CNN}$.

4.5 Visualization of Targeted Patch FGA

FGA pushes $E_v(v')$ away from matching text embeddings. Conversely, we can also push $E_v(v')$ closer to a specified text embedding to produce a predetermined error. In unimodal scenarios, this form of attack is called the targeted attack. For example, we have a text set $\{t\}_{i=1}^n$ and want to push $E(v')$ closer to a specified text t_k . In this case, we need to maximize the following function:

$$L_{gui}^{target} = \ln \left(\frac{\exp(E(v') \cdot E_t(t_k))}{\sum_{i=1}^n \exp(E(v') \cdot E_t(t_i))} \right) \quad (8)$$

We add perturbation to the clean image v in patch form, maximizing L_{gui}^{target} to obtain the adversarial patch image v' . We execute

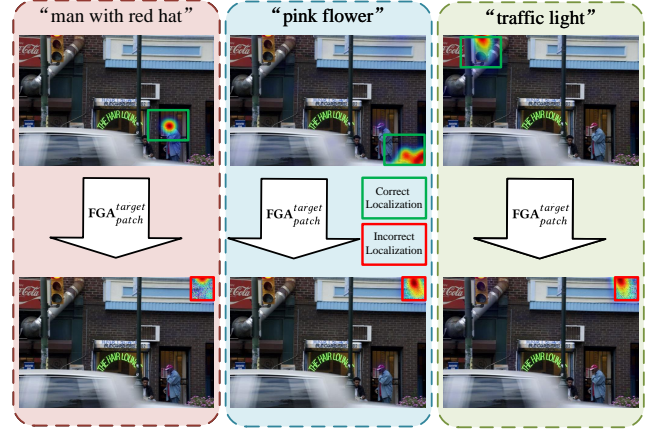


Figure 4: Before the attack, ALBEF can accurately localize image content based on textual cues. After FGA $^{target}_{patch}$, ALBEF’s attention is always erroneously focused on the patch.

airplane	105	15	11	15	5	14	11	4	14	5	900 800 700 600 500 400 300 200 100
automobile	286	939	290	183	123	98	44	100	101	165	
bird	36	28	601	121	69	51	23	94	125	39	
cat	17	33	183	376	74	34	34	57	24	20	
deer	5	9	33	40	303	26	5	25	5	6	
dog	72	120	278	263	321	651	62	362	87	87	
frog	2	13	8	16	5	3	47	5	2	3	
horse	36	81	153	72	70	41	20	673	52	43	
ship	7	6	28	26	4	11	3	17	129	18	
truck	20	24	41	14	17	12	4	10	12	925	
	airplane	automobile	bird	cat	deer	dog	frog	horse	ship	truck	

Figure 5: Each row represents the predicted category for v excluding the correct category y , and each column represents the predicted category for v' .

FGA $^{target}_{patch}$ on the ALBEF model and compute Grad-CAM visualizations on the self-attention maps. As shown in Figure 4, by guiding $E(v')$ closer to the prompt text through FGA $^{target}_{patch}$, ALBEF’s attention area for v' is concentrated on the patch, resulting in a misunderstanding.

4.6 FGA’s Principle of Proximity

In subsection 3.1, it is mentioned that $\frac{\partial L_{gui}}{\partial E(v')}$ not only “guides $E(v')$ away from $\{\omega_{y_i}\}_{i=1}^n$ ”, but also “selects a nearby guiding vector that does not belong to the set $\{\omega_{y_i}\}_{i=1}^n$ and moves closer to it”. The performance decline of VLP models on various V+L downstream tasks in previous experiments sufficiently demonstrates the former. We further prove the latter based on the ZC task and the CLIP

Table 5: Compare the transferability with SOTA methods based on the Flickr30k dataset. The reported value is the attack success rate. Higher is better. R@1 value after the attack is reported in parentheses for SGA, FGA- T_{aug} , and MFGA- T_{aug} . Lower is better.

Source	Attack	ALBEF		TCL		CLIP $_{ViT}$		CLIP $_{CNN}$	
		TR R@1	IR R@1	TR R@1	IR R@1	TR R@1	IR R@1	TR R@1	IR R@1
ALBEF	SA	65.69	73.95	17.60	32.95	31.17	45.23	32.82	45.49
	CA	77.16	83.86	15.21	29.49	23.60	36.48	25.12	38.89
	SGA	97.39(2.5)	97.15(2.52)	45.84(51.7)	55.79(37.98)	33.62(58.5)	44.23(39.1)	36.27(53.5)	46.62(35.28)
	FGA- T_{aug}	99.06(0.9)	99.02(0.9)	46.89(51.2)	58.02(35.7)	36.07(55.9)	47.2(36.64)	38.95(51.0)	50.12(32.62)
	MFGA- T_{aug}	97.6(2.3)	98.15(1.64)	52.27(45.9)	62.57(31.86)	36.93(55.4)	48.39(35.98)	39.72(50.1)	50.6(32.36)
TCL	SA	20.13	36.48	84.72	86.07	31.29	44.65	33.33	45.80
	CA	23.15	40.04	77.94	85.59	27.85	41.19	30.74	44.11
	SGA	49.64(48.5)	59.85(34.78)	98.21(1.7)	98.79(1.1)	34.11(57.6)	44.68(38.64)	37.93(52.4)	48.47(34.02)
	FGA- T_{aug}	44.84(53.4)	58.54(35.92)	99.16(0.8)	99.21(0.68)	35.71(56.5)	47.71(36.18)	39.59(51.0)	49.95(32.98)
	MFGA- T_{aug}	50.78(47.6)	63.05(32.26)	98.31(1.6)	98.57(1.22)	36.32(56.0)	48.94(35.36)	40.74(49.7)	50.5(32.66)
CLIP $_{ViT}$	SA	9.59	23.25	11.38	25.60	79.75	86.79	30.78	39.76
	CA	10.57	24.33	11.94	26.69	93.25	95.86	32.52	41.82
	SGA	12.62(84.7)	27.34(64.3)	14.86(82.2)	29.83(60.64)	99.26(0.6)	99.0(0.64)	38.7(49.8)	47.51(32.32)
	FGA- T_{aug}	12.93(84.4)	28.84(62.7)	14.12(82.7)	30.12(60.44)	99.39(0.5)	99.74(0.18)	42.78(47.3)	48.68(31.82)
	MFGA- T_{aug}	13.56(83.9)	30.05(61.7)	14.96(81.8)	30.98(59.74)	99.26(0.6)	99.52(0.36)	44.44(46.2)	50.94(30.52)
CLIP $_{CNN}$	SA	8.55	23.41	12.64	26.12	28.34	39.43	91.44	95.44
	CA	8.79	23.74	13.10	26.07	28.79	40.03	94.76	96.89
	SGA	11.16(86.1)	25.07(66.14)	14.12(82.6)	27.74(62.62)	31.17(58.8)	42.78(37.76)	99.74(0.2)	99.55(0.26)
	FGA- T_{aug}	12.83(84.6)	26.29(64.9)	14.23(82.9)	28.81(61.54)	35.34(55.5)	45.26(36.14)	100.0(0.0)	99.93(0.04)
	MFGA- T_{aug}	13.35(84.5)	27.48(63.88)	14.86(82.3)	30.1(60.52)	37.42(53.8)	47.2(35.04)	100.0(0.0)	99.90(0.06)

model. In the ZC task, we collect the text set $T = \{t_i\}_{i=1}^c$ and use it to construct the guiding vectors $\{E_t(t_i)\}_{i=1}^c$. FGA encourages $E_v(v')$ to move away from $E_t(t_y)$, where y is the true category, and simultaneously encourages $E_v(v')$ to move closer to the nearest vector from $\{E_t(t_i)\}_{i=1, i \neq y}^c$, meaning that in an ideal situation:

$$\operatorname{argmax}_{i, i \neq y} \frac{E_v(v) \cdot E_t(t_i)}{\|E_v(v)\| \|E_t(t_i)\|} = \operatorname{argmax}_i \frac{E_v(v') \cdot E_t(t_i)}{\|E_v(v')\| \|E_t(t_i)\|} \quad (9)$$

In simpler terms, the category predicted for the clean image v , excluding the true category y , will be the category predicted for the adversarial image v' . Based on the CIFAR-10 dataset, we present the statistical results in Figure 5. In an ideal situation, all positions except the main diagonal should be zero. We observe that the actual situation is close to the ideal. This indicates that the FGA attack indeed tends to guide “ $E(v')$ ” to move closer to the nearest vector from $\{E_t(t_i)\}_{i=1, i \neq y}^c$. In fact, this principle of proximity promotes v' to automatically choose the nearest decision boundary to cross, which is also one of the reasons for the success of FGA.

4.7 Ablation Experiments

We investigate the impact of the number of iterations ($step$) and intensity of noise (ϵ) based on the image-text retrieval task, Flickr30k dataset and the BEiT-3 model, as shown in Fig 6. We can observe that as ϵ and $step$ increase, the effectiveness of the attack gradually strengthens and tends to converge. Specific experimental configurations are detailed in Appendix C.

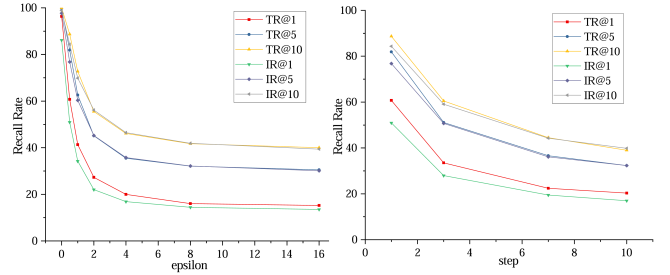


Figure 6: On the left, we fix $step$ at 1 and investigate the impact of ϵ . On the right, we fix ϵ at 0.5 and explore the effect of $step$.

5 CONCLUSION

In this paper, we attempt to construct a unified understanding of adversarial vulnerability regarding unimodal models and VLP models. We abstract visual modality attack into a feature guidance form and combine it with text attack and other enhancement mechanisms to establish a general baseline for exploring the security of the VLP domain. In fact, our approach is theoretically orthogonal to many other attack schemes in the unimodal domain, which facilitates further exploration of the vulnerabilities of VLP models and the design of defence algorithms in subsequent work. We hope our code can be beneficial to the community.

References

- [1] Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhuf, Ian Reid, Stephen Gould, and Anton Van Den Hengel. 2018. Vision-and-language navigation: interpreting visually-grounded navigation instructions in real environments. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 3674–3683.
- [2] Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. 2022. BEiT: BERT pre-training of image transformers. In *International Conference on Learning Representations (ICLR)*.
- [3] Hangbo Bao, Wenhui Wang, Li Dong, Qiang Liu, Owais Khan Mohammed, Kriti Aggarwal, Subhojit Som, Songhao Piao, and Furu Wei. 2022. Vlm0: unified vision-language pre-training with mixture-of-modality-experts. In *Proceedings of the 36th International Conference on Neural Information Processing Systems (NeurIPS)*, 32897–32912.
- [4] Nicholas Carlini and David Wagner. 2017. Towards evaluating the robustness of neural networks. In *IEEE Symposium on Security and Privacy (S&P)*, 39–57.
- [5] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *Proceedings of the International Conference on Machine Learning (ICML)*, 1597–1607.
- [6] Z. Chen, J. Zhang, Z. Lai, G. Zhu, Z. Liu, J. Chen, and J. Li. 2023. The devil is in the crack orientation: a new perspective for crack detection. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 6630–6640.
- [7] Zhaoyu Chen, Bo Li, Shuang Wu, Jianghe Xu, Shouhong Ding, and Wenqiang Zhang. 2022. Shape matters: deformable patch attack. In *Proceedings of the 17th European Conference on Computer Vision (ECCV)*, 529–548.
- [8] Zhuangzhuang Chen, Jin Zhang, Zhuonan Lai, Jie Chen, Zun Liu, and Jianqiang Li. 2022. Geometry-aware guided loss for deep crack recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 4693–4702.
- [9] 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.
- [10] 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.
- [11] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Li Kai, and Fei-Fei Li. 2009. ImageNet: a large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 248–255.
- [12] 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.
- [13] Yinpeng Dong, Tianyu Pang, Hang Su, and Jun Zhu. 2019. Evading defenses to transferable adversarial examples by translation-invariant attacks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 4312–4321.
- [14] Alexey Dosovitskiy et al. 2023. An image is worth 16x16 words: transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*.
- [15] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations (ICLR)*.
- [16] 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.
- [17] Jean-Bastien Grill et al. 2020. Bootstrap your own latent: a new approach to self-supervised learning. In *Proceedings of the 34th International Conference on Neural Information Processing Systems (NeurIPS)*, 21271–21284.
- [18] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2020. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 9729–9738.
- [19] 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.
- [20] Wonjae Kim, Bokyung Son, and Ildoo Kim. 2021. Vilt: vision-and-language transformer without convolution or region supervision. In *Proceedings of the International Conference on Machine Learning (ICML)*, 5583–5594.
- [21] A. Krizhevsky and G. Hinton. 2009. Learning multiple layers of features from tiny images. *Handbook of Systemic Autoimmune Diseases*.
- [22] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. 2022. Blip: bootstrapping language-image pre-training for unified vision-language understanding and generation. In *Proceedings of the International Conference on Machine Learning (ICML)*, 12888–12900.
- [23] 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.
- [24] Linyang Li, Ruotian Ma, Qipeng Guo, Xiangyang Xue, and Xipeng Qiu. 2020. Bert-attack: adversarial attack against bert using bert. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 6193–6202.
- [25] Maosen Li, Cheng Deng, Tengjiao Li, Junchi Yan, Xinbo Gao, and Heng Huang. 2020. Towards transferable targeted attack. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 641–649.
- [26] Siyuan Li, Xing Xu, Zaili Zhou, Yang Yang, Guoqing Wang, and Heng Tao Shen. 2022. Arra: absolute-relative ranking attack against image retrieval. In *Proceedings of the 30th ACM International Conference on Multimedia (ACM MM)*, 610–618.
- [27] Wenrui Li, Zhengyu Ma, Liang-Jian Deng, Penghong Wang, Jinqiao Shi, and Xiaopeng Fan. 2023. Reservoir computing transformer for image-text retrieval. In *Proceedings of the 31st ACM International Conference on Multimedia (ACM MM)*, 5605–5613.
- [28] Wenrui Li, Penghong Wang, Ruiqin Xiong, and Xiaopeng Fan. 2024. Spiking tucker fusion transformer for audio-visual zero-shot learning. *IEEE Transactions on Image Processing*, 1–1.
- [29] Wenrui Li, Xi-Le Zhao, Zhengyu Ma, Xingtao Wang, Xiaopeng Fan, and Yonghong Tian. 2023. Motion-decoupled spiking transformer for audio-visual zero-shot learning. In *Proceedings of the 31st ACM International Conference on Multimedia (ACM MM)*, 3994–4002.
- [30] Jiadong Lin, Chuanbiao Song, Kun He, Liwei Wang, and John E Hopcroft. 2020. Nesterov accelerated gradient and scale invariance for adversarial attacks. In *International Conference on Learning Representations (ICLR)*.
- [31] 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.
- [32] Fanguy Liu, Rémi Lebret, Didier Orel, Philippe Sordet, and Karl Aberer. 2020. Upgrading the newsroom: an automated image selection system for news articles. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 1–28.
- [33] Dong Lu, Zhiqiang Wang, Teng Wang, Weili Guan, Hongchang Gao, and Feng Zheng. 2023. Set-level guidance attack: boosting adversarial transferability of vision-language pre-training models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 102–111.
- [34] 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)*.
- [35] 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.
- [36] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. 2016. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2574–2582.
- [37] Liang Peng, Yang Yang, Zheng Wang, Zi Huang, and Heng Tao Shen. 2020. Mra-net: improving vqa via multi-modal relation attention network. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 318–329.
- [38] Zhiliang Peng, Li Dong, Hangbo Bao, Qixiang Ye, and Furu Wei. 2022. BEiT v2: masked image modeling with vector-quantized visual tokenizers. In *arXiv preprint arXiv:2208.06366*.
- [39] Bryan A. Plummer, Liwei Wang, Chris M. Cervantes, Juan C. Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. 2015. Flickr30k entities: collecting region-to-phrase correspondences for richer image-to-sentence models. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2641–2649.
- [40] Alec Radford et al. 2021. Learning transferable visual models from natural language supervision. In *Proceedings of the International Conference on Machine Learning (ICML)*, 8748–8763.
- [41] 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.
- [42] 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.
- [43] 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.
- [44] Florian Tramèr and Dan Boneh. 2019. Adversarial training and robustness for multiple perturbations. *Advances in neural information processing systems*, 32.

- [45] Wenhui Wang 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.
- [46] Xiaosen Wang and Kun He. 2021. Enhancing the transferability of adversarial attacks through variance tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 1924–1933.
- [47] Xin Wang, Qiuyuan Huang, Asli Celikyilmaz, Jianfeng Gao, Dinghan Shen, Yuan-Fang Wang, William Yang Wang, and Lei Zhang. 2019. Reinforced cross-modal matching and self-supervised imitation learning for vision-language navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 6629–6638.
- [48] Jiwei Wei, Yang Yang, Xing Xu, Jingkuan Song, Guoqing Wang, and Heng Tao Shen. 2023. Less is better: exponential loss for cross-modal matching. *IEEE Transactions on Circuits and Systems for Video Technology (TCSVT)*, 5271–5280.
- [49] Cihang Xie, Zhishuai Zhang, Yuyin Zhou, Song Bai, Jianyu Wang, Zhou Ren, and Alan L. Yuille. 2019. Improving transferability of adversarial examples with input diversity. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2730–2739.
- [50] 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*.
- [51] Jinyu Yang, Jiali Duan, Son Tran, Yi Xu, Sampath Chanda, Liqun Chen, Belinda Zeng, Trishul Chilimbi, and Junzhou Huang. 2022. Vision-language pre-training with triple contrastive learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 15671–15680.
- [52] En Yu, Jiande Sun, Jing Li, Xiaojun Chang, Xian-Hua Han, and Alexander G Hauptmann. 2018. Adaptive semi-supervised feature selection for cross-modal retrieval. *IEEE Transactions on Multimedia (TMM)*, 1276–1288.
- [53] 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.
- [54] Jiaming Zhang, Qi Yi, and Jitao Sang. 2022. Towards adversarial attack on vision-language pre-training models. In *Proceedings of the 30th ACM International Conference on Multimedia (ACM MM)*, 5005–5013.
- [55] Ziqi Zhou, Shengshan Hu, Minghui Li, Hangtao Zhang, Yechao Zhang, and Hai Jin. 2023. AdvCLIP: downstream-agnostic adversarial examples in multimodal contrastive learning. In *Proceedings of the 31st ACM International Conference on Multimedia (ACM MM)*, 6311–6320.

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 [50] with ALBEF [23] 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 ω_y .

Dataset Detail. The SNLI-VE dataset [50] 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 [16] 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 [16] 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 [31], 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 [41] 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 ω_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 ω_1 as the attack strategy.

Dataset Detail. RefCOCO+ [53] 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 [45]. 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 ω_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 [43] (Natural Language for Visual Reasoning for Real) is a natural language processing dataset designed 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

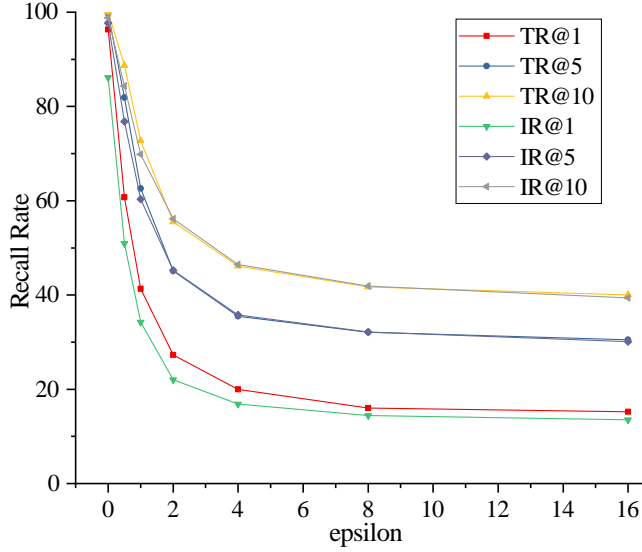


Figure 7: The experimental results when $step = 1$. We observe that as the noise constraint is relaxed (with ϵ increasing), the effectiveness of the attack gradually intensifies. However, the rate of decline in model performance slows down, indicating that the attack strength tends to converge.

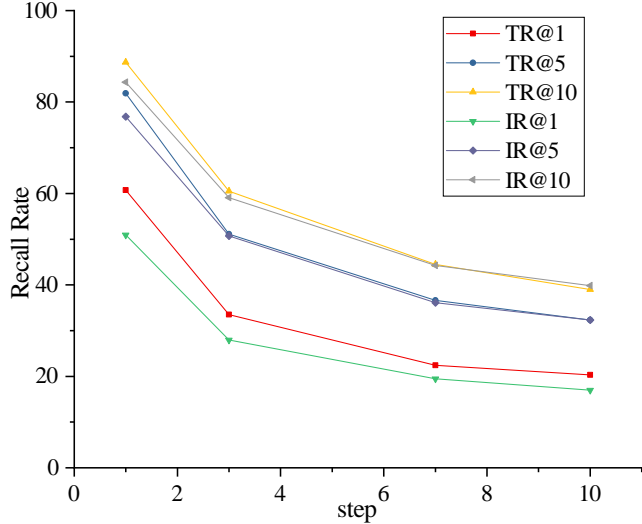


Figure 8: The experimental results when $\epsilon = 0.5$. We note that as the number of iterations increases, the attack’s effectiveness progressively intensifies. Nonetheless, the decrease in model performance decelerates, suggesting a convergence in attack potency.

where $Clamp_{(-\epsilon, \epsilon)}$ represents clipping each element value in δ to be between $-\epsilon$ and ϵ . Besides, $P_{B(\epsilon, 1)}$ involves a complex projection strategy for sparsity ℓ_1 perturbation, discussed in detail in APGD $_{\ell_1}$ [10] and MAX[44].

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 [44]. Therefore, the interplay of adversarial perturbations across multiple norms can better explore the lower bounds of a network’s robustness [35, 9, 42].

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 10) [12].

Obtaining gradient information with momentum mechanism:

$$g \leftarrow \nabla_{\delta^{(i)}} L_{gui}(v + \delta^{(i)}) \quad (19)$$

$$g \leftarrow g / \text{mean}(\text{abs}(g)) \quad (20)$$

$$g \leftarrow g + \alpha \cdot g_m \quad (21)$$

$$g_m \leftarrow g \quad (22)$$

where abs represents taking the absolute value of each element in g , while 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 α .

B.3 Patch Perturbation

Global attacks constrain the perturbation δ through ϵ , 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 [19]. 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:

$$g = \nabla_{\delta^{(i)}} L_{gui}(v \odot (1 - m) + \delta^{(i)} \odot m) \quad (23)$$

$$\delta^{(i+1)} = \text{Clamp}_{(0,1)}(\delta^{(i)} + g) \quad (24)$$

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 (ϵ) 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 ℓ_∞ 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 6, Tab 7, Tab 8 and Tab 9, respectively. Concurrently, we explore a range of ϵ values set at $\{0.5, 1, 2, 4, 8, 16\}$, ensuring that $\|\delta\|_\infty \leq \epsilon$, where the ϵ 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

ϵ increasing), the effectiveness of the attack gradually intensifies (Fig 7). 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 8).