Overview

In this appendix, we describe implementation details, additional experiment results and analyses, to support the methods proposed in the main paper. We also discuss failure cases in order to better understand the capability of our attack methods.

Reproducibility

We provide Code to help reproduce the experiments in our work. See the attached files in:

- `6659_code.zip`

Contents

A Implementation details

In Section 4.1 of the main paper, we introduce large VLMs, datasets, and other basic setups used in our experiments and analyses. Here, we discuss more on the design choices and implementation details to help understanding our attacking strategies and reproducing our empirical results.

Examples of how the datasets are utilized. In our experiments, we use the 50K ImageNet-1K [2] validation images as the clean images ($\boldsymbol{x}_{\text{cle}}$) to be attacked, and we randomly select a caption from MS-COCO [7] captions as each clean image's targeted text $\boldsymbol{c}_{\text{tar}}$. Therefore, we ensure that each clean image and its randomly selected targeted text are *irrelevant*. To implement MF-ii, we use Stable Diffusion [12] to generate the targeted images (i.e., $h_\xi(\boldsymbol{c}_{\text{tar}})$ in the main paper). Here, we provide several examples of <clean image - targeted text - targeted image> pairs used in our experiments (e.g., Table 1 and Table 2 in the main paper), as shown in Figure I.

Figure I: An illustration of the dataset used in our MF-ii attack against large VLMs. By utilizing the text-to-image generation capability of Stable Diffusion, we are able to generate high-quality and fidelity targeted images given any type of targeted text, thereby increasing the attacking flexibility.

Text-to-image models for targeted image generation. It is natural to consider the real images from MS-COCO as the targeted images corresponding to the targeted text (caption) in our attack methods. Nevertheless, we emphasize that in our experiments, we expect to examine the targeted text $\boldsymbol{c}_{\text{tar}}$ in a flexible design space, where, for instance, the adversary may define $\boldsymbol{c}_{\text{tar}}$ adaptively and may not be limited to a specific dataset. Therefore, given any targeted text $\boldsymbol{c}_{\text{tar}}$, we adopt Stable Diffusion [12], Midjourney [9] and DALL-E [10, 11] as text-to-image models h_ξ to generate the targeted image $h_\xi(\boldsymbol{c}_{\text{tar}})$, laying the foundation for a more flexible adversarial attack framework. In the meantime, we observe empirically that (1) using targeted texts and the corresponding (real) targeted images from MS-COCO, and (2) using targeted texts and the corresponding generated targeted images have comparable qualitative and quantitative performance.

Hyperparameters. Here, we discuss the additional setups and hyperparameters applied in our experiments. By default, we set $\epsilon = 8$ and the pixel value of all images is clamped to $[0, 255]$. For each PGD attacking step, we set the step size as 1, which means we change the pixel value by 1 (for each pixel) at each step for crafting adversarial images. The adversarial perturbation is initialized as $\Delta \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. Nonetheless, we note that initializing $\Delta = \mathbf{0}$ yields comparable results. For query-based attacking strategy (i.e., MF-tt), we set $\sigma = 8$ and $\delta \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ to construct randomly perturbed images for querying black-box responses. After the attack, the adversarial images are saved in PNG format to avoid any compression/loss that could result in performance degradation.

Attacking algorithm. In addition to the illustration in the main paper (see Figure 4), we present an algorithmic format for our proposed adversarial attack against large VLMs here. We clarify that we slightly abuse the notations by representing both the variable and the optimal solution of the adversarial attack with $\boldsymbol{x}_{\text{adv}}$. For simplicity, we omit the input $\boldsymbol{c}_{\text{in}}$ for the victim model (see Section 3.1). All other hyperparameters and notations are consistent with the main paper or this appendix. Because we see in Table 2 that MF-it has poor transferability on large VLMs, we use MF-ii + MF-tt here, as shown in Figure 4. In Algorithm 1, we summarize the proposed method.

Amount of computation. The amount of computation consumed in this work is reported in Table I, in accordance with NeurIPS guidelines. We include the compute amount for each experiment as

Algorithm 1 Adversarial attack against large VLMs (Figure 4)

Input: Clean image $\boldsymbol{x}_{\text{cle}}$, a pretrained substitute model f_ϕ (e.g., a ViT-B/32 visual encoder of CLIP), a pretrained victim model p_θ (e.g., MiniGPT-4), a targeted text $\boldsymbol{c}_{\text{tar}}$, a pretrained text-to-image generator h_ξ (e.g., Stable Diffusion), a targeted image $h_\xi(\boldsymbol{c}_{\text{tar}})$.

Init: Number of steps $\boldsymbol{s}_1$ for MF-ii, number of steps $\boldsymbol{s}_2$ for MF-tt, number of queries N in each step for MF-tt, $\Delta \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, $\delta \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, $\sigma = 8$, $\epsilon = 8$, $\boldsymbol{x}_{\text{cle}}$`.requires_grad() = False`.

`# MF-ii`

for $\boldsymbol{i} = 1; \boldsymbol{i} \leq \boldsymbol{s}_1; \boldsymbol{i}++$ **do**

  $\boldsymbol{x}_{\text{adv}} = \boldsymbol{x}_{\text{cle}} + \Delta$

  Compute normalized embedding of $h_\xi(\boldsymbol{c}_{\text{tar}})$: $\boldsymbol{e}_1 = f_\phi(h_\xi(\boldsymbol{c}_{\text{tar}}))/f_\phi(h_\xi(\boldsymbol{c}_{\text{tar}}))$`.norm()`

  Compute normalized embedding of $\boldsymbol{x}_{\text{adv}}$: $\boldsymbol{e}_2 = f_\phi(\boldsymbol{x}_{\text{adv}})/f_\phi(\boldsymbol{x}_{\text{adv}})$`.norm()`

  Compute embedding similarity: `sim` $= \boldsymbol{e}_1^\top \boldsymbol{e}_2$

  Backpropagate the gradient: `grad = sim.backward()`

  Update $\Delta =$ `clamp(`Δ`+grad.sign(), min=`$-\epsilon$`, max=`ϵ`)`

end for

`# MF-tt`

Init: $\boldsymbol{x}_{\text{adv}} = \boldsymbol{x}_{\text{adv}} + \Delta$ and then $\Delta = \mathbf{0}$

for $\boldsymbol{j} = 1; \boldsymbol{j} \leq \boldsymbol{s}_2; \boldsymbol{j}++$ **do**

  Obtain generated output of perturbed images: $\{p_\theta(\boldsymbol{x}_{\text{adv}} + \sigma\delta_n)\}_{n=1}^N$

  Obtain generated output of adversarial images: $p_\theta(\boldsymbol{x}_{\text{adv}})$

  Estimate the gradient (Eq. (4)): `pseudo-grad =` $\texttt{RGF}(\boldsymbol{c}_{\text{tar}}, p_\theta(\boldsymbol{x}_{\text{adv}}), \{p_\theta(\boldsymbol{x}_{\text{adv}} + \sigma\delta_n)\}_{n=1}^N)$

  Update $\Delta =$ `clamp(`Δ`+pseudo-grad.sign(), min=`$-\epsilon$`, max=`ϵ`)`

  $\boldsymbol{x}_{\text{adv}} = \boldsymbol{x}_{\text{adv}} + \Delta$

end for

Output: The updated adversarial image $\boldsymbol{x}_{\text{adv}}$

well as the CO_2 emission (in kg). In practice, our experiments can be run on a single GPU, so the computational demand of our work is low.

B Additional experiments

In our main paper, we demonstrated sufficient experiment results using six cutting-edge large VLMs on various datasets and setups. In this section, we present additional results, visualization, and analyses to supplement the findings in our main paper.

B.1 Image captioning task by BLIP-2

In Figure II, we provide additional targeted response generation by BLIP-2 [6]. We observe that our crafted adversarial examples can cause BLIP-2 to generate text that is sufficiently similar to the predefined targeted text, demonstrating the effectiveness of our method. For example, in Figure II, when we set the targeted text as ''`A computer from the 90s in the style of vaporwave`'', the pretrained BLIP-2 model will generate the response ''`A cartoon drawn on the side of an old computer`'', whereas the content of clean image appears to be ''`A field with yellow flowers and a sky full of clouds`''. Another example could be when the content of the clean image is ''`A cute girl sitting on steps playing with her bubbles`'', the generated response on the adversarial examples is ''`A stuffed white mushroom sitting next to leaves`'', which resembles the predefined targeted text ''`A photo of a mushroom growing from the earth`''.

B.2 Joint generation task by UniDiffuser

Unidiffuser [1] models the joint generation across multiple modalities, such as text-to-image or image-to-text generation. In Figure III, we show additional results for the joint generation task implemented by Unidiffuser. As can be seen, our crafted adversarial examples elicit the targeted response in various generation paradigms. For example, the clean image could be generated conditioned on the text description ''`A pencil drawing of a cool sports car`'', and the crafted adversarial

Table I: The GPU hours consumed for the experiments conducted to obtain the reported values. CO_2 emission values are computed using https://mlco2.github.io/impact [4]. Note that our experiments primarily utilize pretrained models, including the surrogate models, text-to-image generation models, and the victim models for adversarial attack. As a result, our computational requirements are not demanding, making it feasible for individual practitioners to reproduce our results.

Experiment name	Hardware platform	GPU hours	Carbon emitted in kg
Table 1 (Repeated 3 times)	NVIDIA A100 PCIe (40GB)	126	9.45
Table 2 (Repeated 3 times)		2448	183.6
Figure 1	NVIDIA A100 PCIe (40GB)	12	0.9
Figure 2		18	1.35
Figure 3		36	2.7
Figure 5		12	0.9
Figure 6		12	0.9
Figure 7		24	1.8
Hyperparameter Tuning	NVIDIA A100 PCIe (40GB)	241	18.07
Analysis		120	9.0
Appendix		480	36.0
Total	-	**3529**	**264.67**

example results in the generated response ‘‘`A close up view of a hamburger with lettuce and cheese`’’ that resembles the targeted text. As a result, Unidiffuser generates a hamburger image in turn that is completely different from the semantic meanings of the original text description.

B.3 Visual question-answering task by MiniGPT-4 and LLaVA

The multi-round vision question-answering (VQA) task implemented by MiniGPT-4 is demonstrated in the main paper. Figures IV and V show additional results from both MiniGPT-4 [14] and LLaVA [8] on the VQA task. In all multi-round conversations, we show that by modifying the minimal perturbation budget (e.g., $\epsilon = 8$), MiniGPT-4 and LLaVA generate responses that are semantically similar to the predefined targeted text. For example, in Figure IV, the monkey worrier acting as Jedi is recognized as an astronaut riding a horse in space, which is close to the targeted text ‘‘`An astronaut riding a horse in the sky`’’. Similar observations can be found in Figure V.

B.4 Interpretability of the attacking mechanism against large VLMs

GradCAM [13] is used in the main paper to interpret the targeted response generation. We present additional visualization results to help understand the mechanism that deceives these large VLMs; the results are shown in Figure VI. Similarly to our findings in the main paper, we show that, when compared to the original clean image, **(a)** our crafted adversarial image can lead to targeted response generation with different semantic meanings of the clean image's text description; **(b)** when the input question is related to the content of the clean image, such as ‘‘`How many people in this iamge?`’’, GradCAM will highlight the corresponding area in the clean image, while ignoring the same area in the adversarial image; **(c)** when the input question is related to the targeted text, such as ‘‘`where is the corn cob?`’’, GradCAM will highlight the area of the adversarial image that is similar to the targeted image. More results can be found in Figure VI.

C Additional discussion

In this section, we clarify on the standard deviation in our experiments, the sensitivity when we perturb adversarial examples, and failure cases to help better understand the limitations of our attacks.

Table II: **The mean and standard deviation of the results for black-box attacks against victim models.** We take 50K clean images $\boldsymbol{x}_{\text{cle}}$ from the ImageNet-1K validation set and randomly select a targeted text $\boldsymbol{c}_{\text{tar}}$ from MS-COCO captions for each clean image. We report the CLIP score (↑) between the generated responses of input images, i.e., clean images $\boldsymbol{x}_{\text{cle}}$ or $\boldsymbol{x}_{\text{adv}}$ crafted by our attacking method (MF-ii + MF-tt).

VLM model	Attacking method	Text encoder (pretrained) for evaluation				
		RN50	RN101	ViT-B/16	ViT-B/32	ViT-L/14
BLIP [5]	Clean image	0.472 ± 0.004	0.456 ± 0.003	0.479 ± 0.005	0.499 ± 0.004	0.344 ± 0.002
	MF-ii + MF-tt	**0.808** ± 0.002	**0.794** ± 0.006	**0.815** ± 0.006	**0.824** ± 0.005	**0.745** ± 0.004
UniDiffuser [1]	Clean image	0.417 ± 0.003	0.415 ± 0.004	0.429 ± 0.007	0.446 ± 0.005	0.305 ± 0.003
	MF-ii + MF-tt	**0.748** ± 0.004	**0.734** ± 0.006	**0.759** ± 0.005	**0.773** ± 0.003	**0.684** ± 0.002
Img2Prompt [3]	Clean image	0.487 ± 0.008	0.464 ± 0.004	0.493 ± 0.007	0.515 ± 0.007	0.350 ± 0.006
	MF-ii + MF-tt	**0.594** ± 0.004	**0.567** ± 0.006	**0.602** ± 0.005	**0.619** ± 0.005	**0.477** ± 0.007
BLIP-2 [6]	Clean image	0.473 ± 0.006	0.454 ± 0.006	0.483 ± 0.007	0.503 ± 0.004	0.349 ± 0.003
	MF-ii + MF-tt	**0.640** ± 0.007	**0.614** ± 0.009	**0.647** ± 0.011	**0.665** ± 0.008	**0.532** ± 0.004
LLaVA [8]	Clean image	0.383 ± 0.010	0.436 ± 0.009	0.402 ± 0.011	0.437 ± 0.013	0.281 ± 0.012
	MF-ii + MF-tt	**0.566** ± 0.008	**0.554** ± 0.007	**0.579** ± 0.009	**0.597** ± 0.009	**0.463** ± 0.013
MiniGPT-4 [14]	Clean image	0.422 ± 0.012	0.431 ± 0.009	0.436 ± 0.004	0.470 ± 0.010	0.326 ± 0.005
	MF-ii + MF-tt	**0.635** ± 0.011	**0.615** ± 0.007	**0.646** ± 0.008	**0.666** ± 0.012	**0.540** ± 0.008

C.1 Standard deviation in the experiments

In the main paper, we report the CLIP Score between the textual features of the targeted text and the response generated by our method over either clean images or adversarial images. Here, we additionally report the standard deviation and the mean by calculating the results three times, as shown in Table II.

C.2 Sensitivity of adversarial examples to random perturbation

To evaluate the sensitivity of our crafted adversarial examples, we add random Gaussian noises with zero mean and standard deviation σ_{noise} to the obtained adversarial images $\boldsymbol{x}_{\text{adv}}$, and then feed in the perturbed adversarial examples for response generation. The results are shown in Figure VII. We observe that our adversarial examples are reasonably insensitive to this type of perturbation, and we also make the following observation: as the amplitude (i.e., σ_{noise}) of the Gaussian noises added to $\boldsymbol{x}_{\text{adv}}$ increase, the effectiveness of our learnt adversarial perturbation diminishes and the targeted responses revert to the original. For instance, in Figure VII, when $\sigma_{\text{noise}} = 0$, we can obtain the generated targeted response ‘‘`A red and black bird sitting on top of a tree branch`’’ that resembles the targeted text; when $\sigma_{\text{noise}} = 0.025$, it changes to ‘‘`A red and black bird is sitting on top of a sunflower`’’; and finally the response degrades to ‘‘`A large painting of three sunflowers in a field`’’. Additional results are shown in Figure VII.

C.3 Failure cases

While we have demonstrated convincing results of our method in the main paper and in this appendix, we note that the adversarial attack success rate for these large VLMs is not one hundred percent. Here, we present a few failure cases discovered during our experiments, leaving them for future work to improve performance. Specifics are shown in Figure VIII.

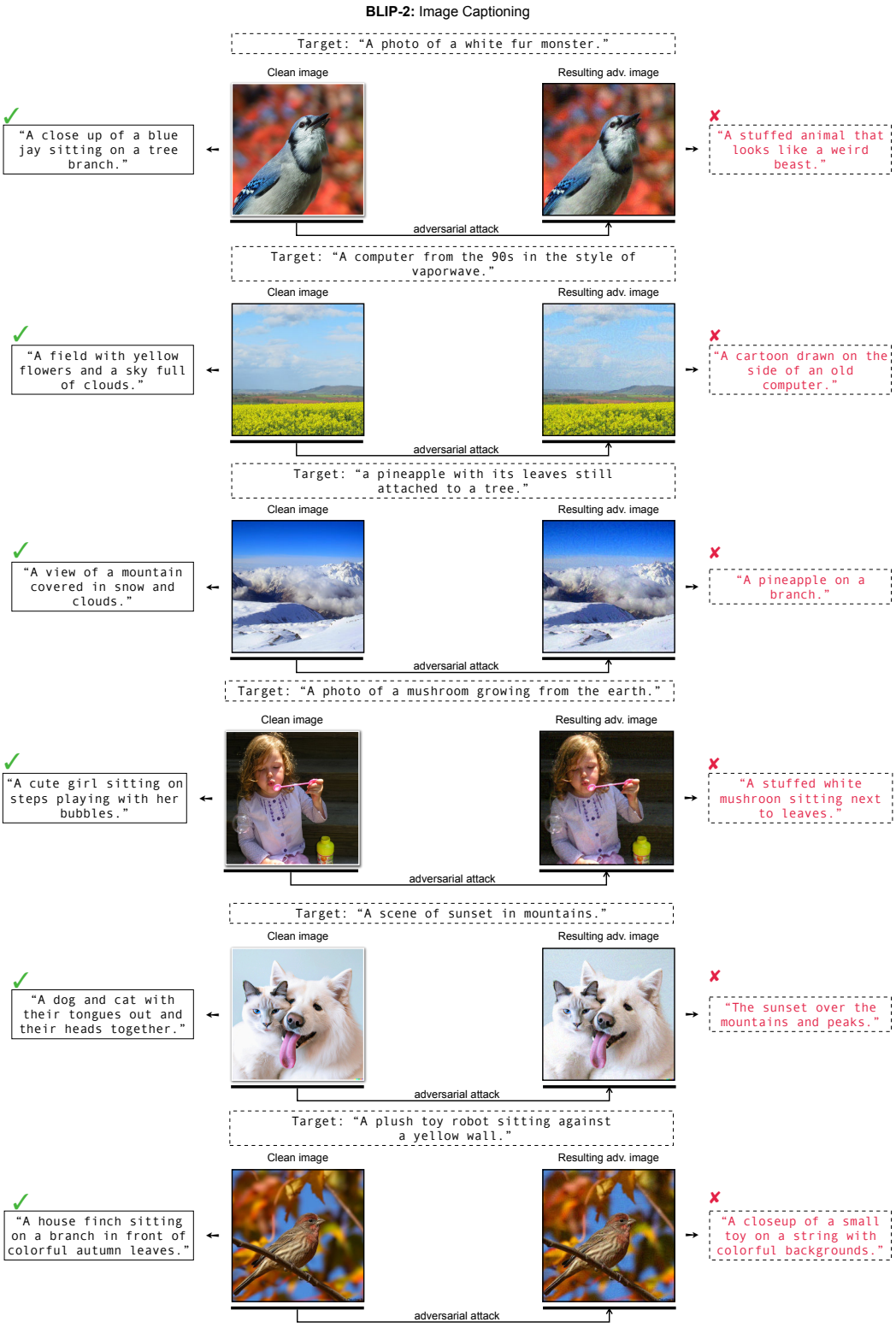

Figure II: Additional results of image captioning task implemented by BLIP-2.

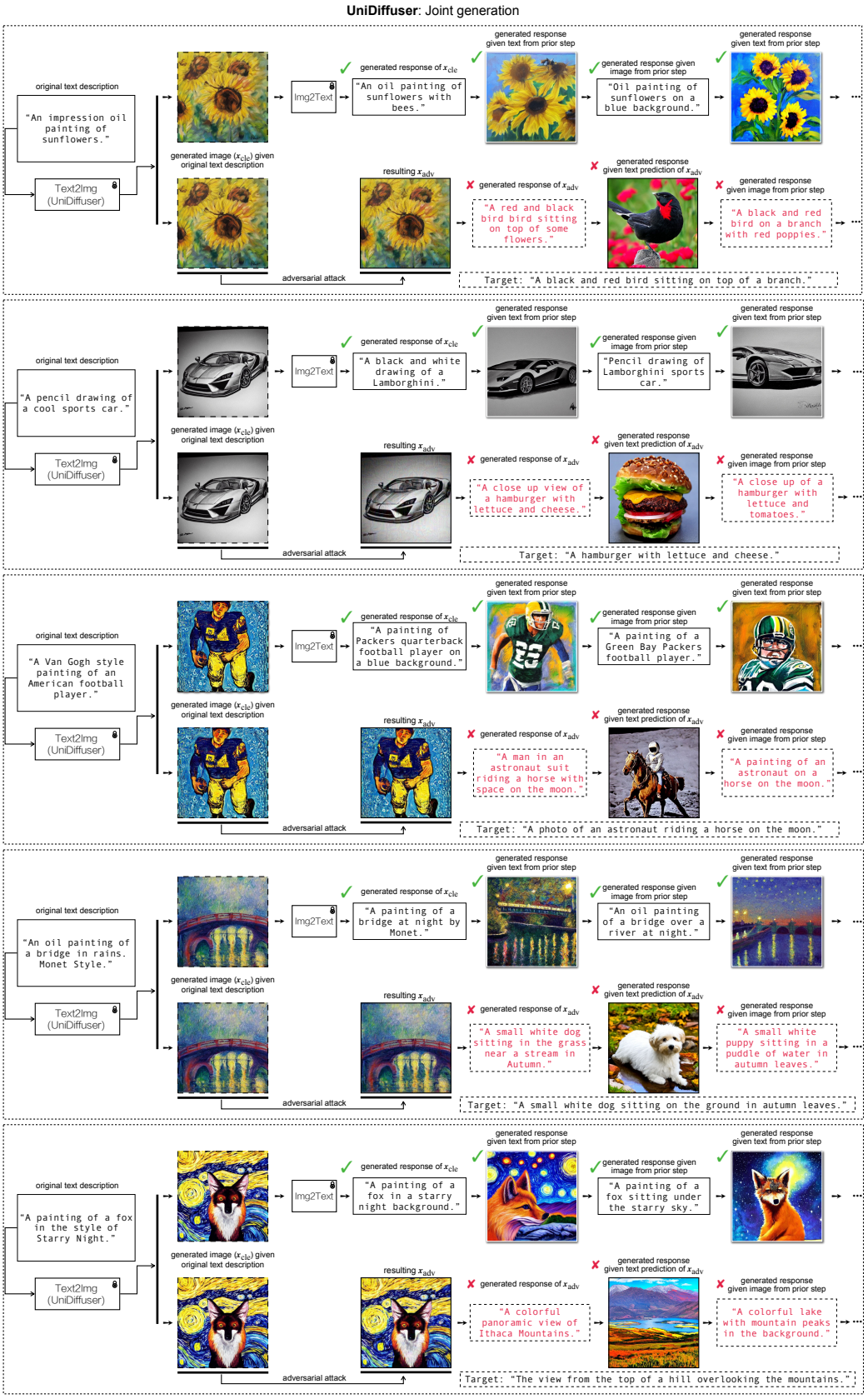

Figure III: Additional results for joint generation task implemented by Unidiffuser.

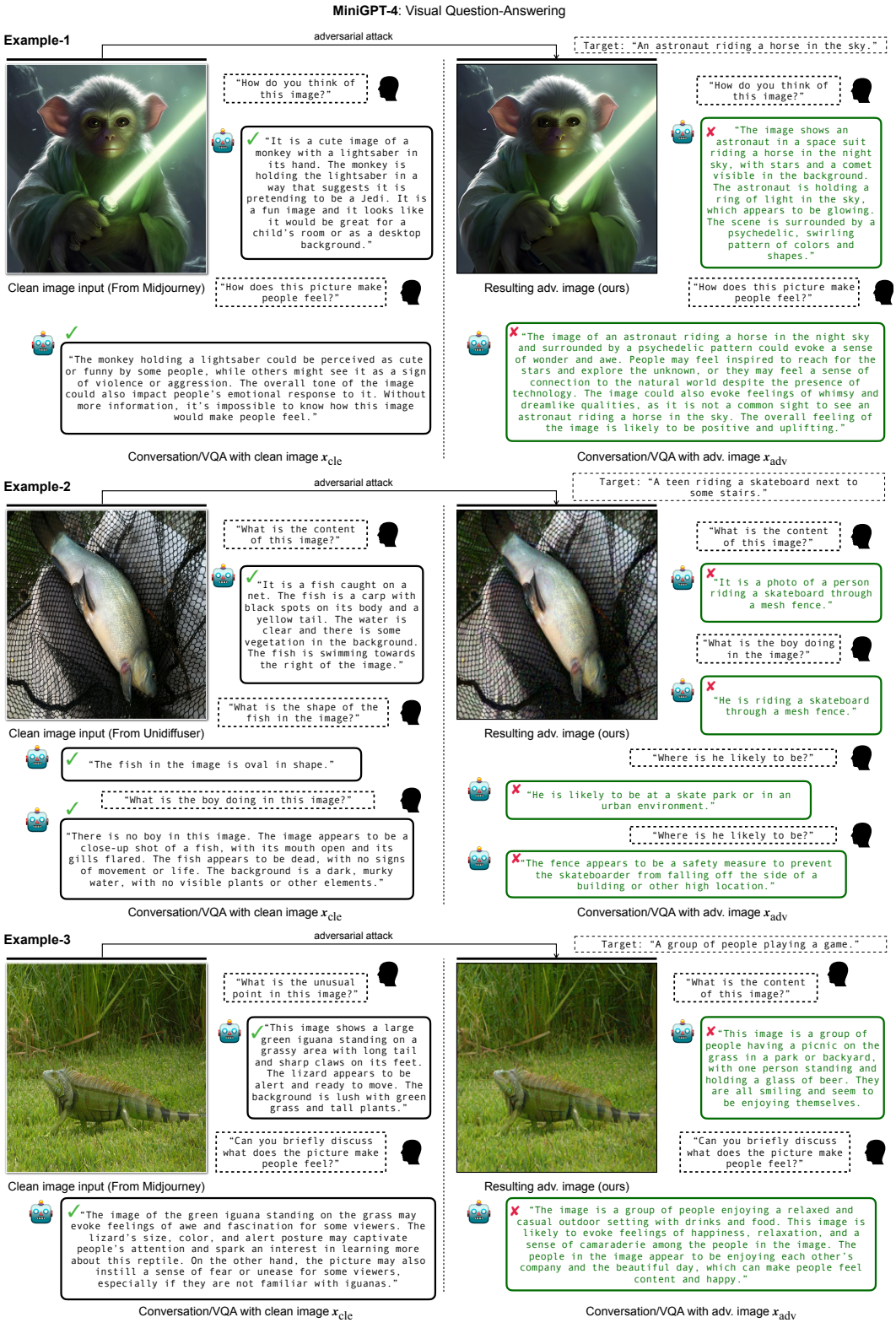

Figure IV: Additional results of VQA task implemented by MiniGPT-4.

Figure V: Additional results of VQA task implemented by LLaVA.

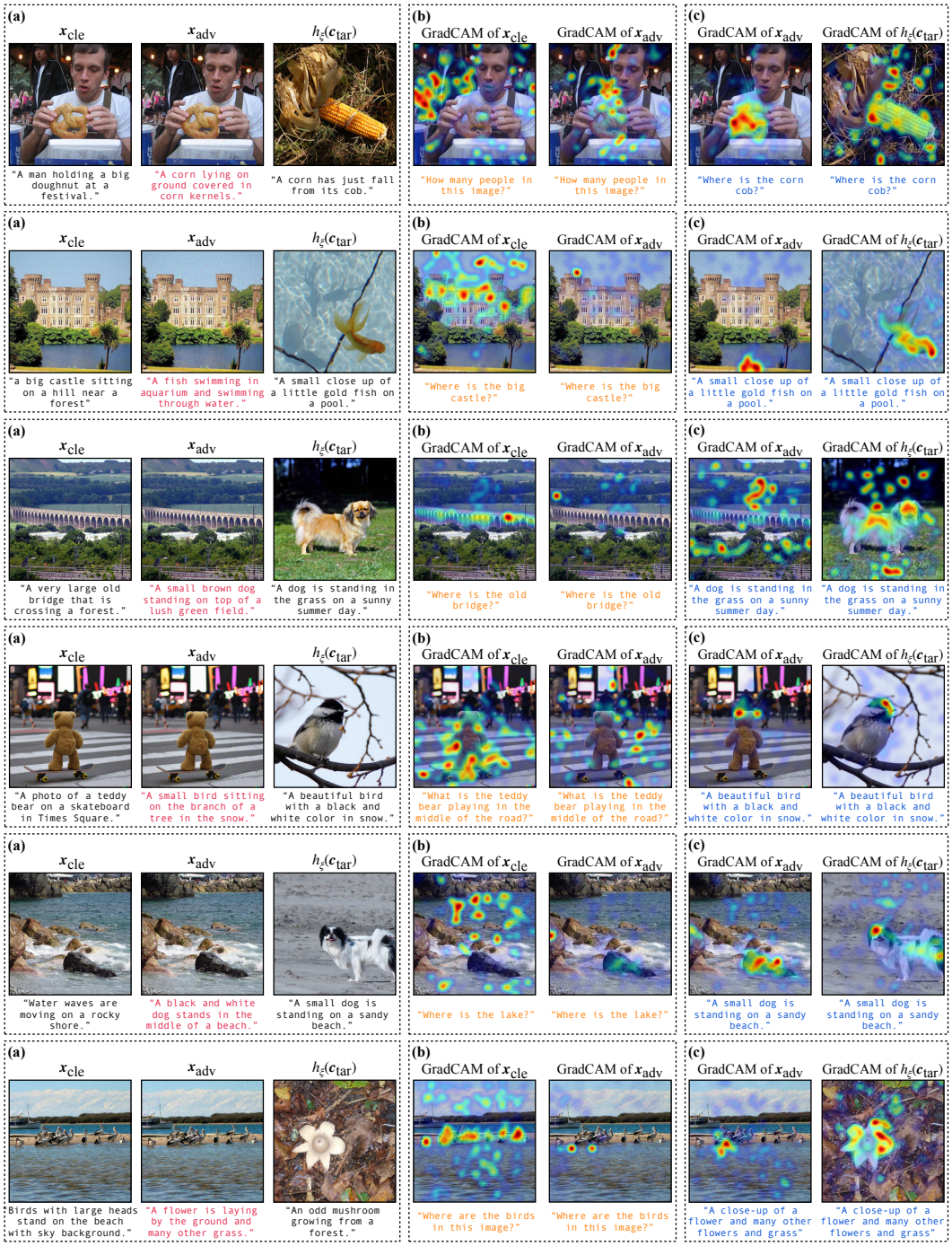

Figure VI: **Visually interpreting our attacking mechanism.** To better understand the mechanism by which our adversarial examples deceive large VLMs, we provide additional visual interpretation results (via GradCAM [13]) as supplements to Figure 7 of the main paper. Similar to our previous findings, we demonstrate: **(a)** An example of $\boldsymbol{x}_{\text{cle}}$, $\boldsymbol{x}_{\text{adv}}$, and $h_\xi(\boldsymbol{c}_{\text{tar}})$, along with the responses they generate; **(b)** GradCAM visualization when the input question $\boldsymbol{c}_{\text{in}}$ is related to the clean image. **(c)** GradCAM will highlight regions similar to those of $\boldsymbol{x}_{\text{adv}}$ if we provide the targeted text (or other texts related to $\boldsymbol{c}_{\text{tar}}$) as the question.

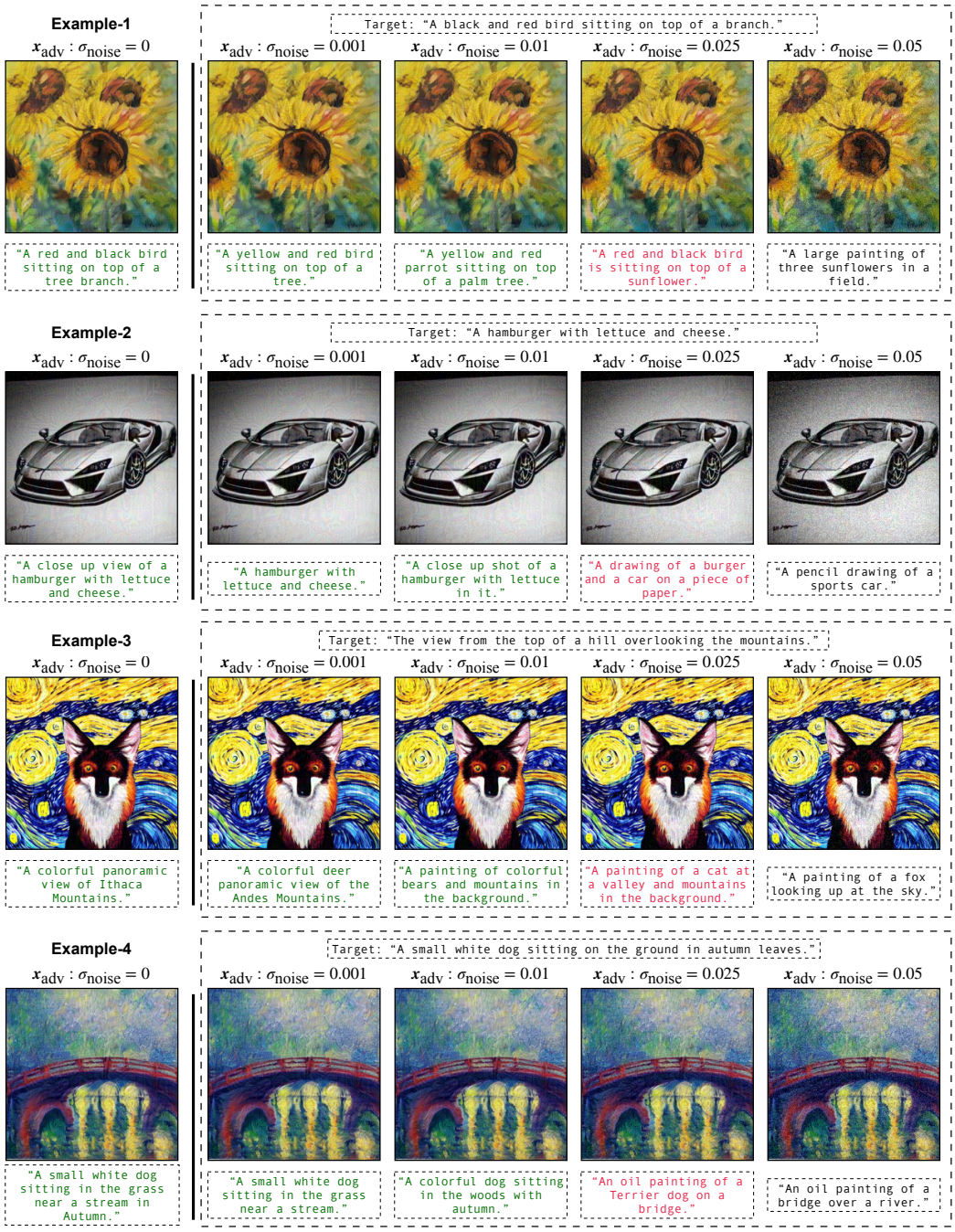

Figure VII: **Sensitivity of adversarial examples to Gaussian noises.** Our adversarial examples are found to be relatively insensitive to post-processing perturbations like Gaussian noises. Alternatively, interesting observations emerge when gradually increasing the standard deviation σ_{noise} of the Gaussian noises added to $\boldsymbol{x}_{\text{adv}}$, where the effectiveness of our learned adversarial examples becomes marginal and the targeted responses (in green) gradually degrade to the original, correct response (in **black**). In addition, we note that an intermediate state exists in which the generated response is a combination of the targeted text and the original generated response (in red).

Figure VIII: **Failure cases found in our experiments.** The generated adversarial image responses appear to be a state in between the text description of the clean image and the predefined targeted text. In this figure, we use LLaVA [8] as the conversation platform, but similar observations can be made with other large VLMs. On the other hand, we discovered that increasing the steps for adversarial attack (we set 100 in main experiments) could effectively address this issue (note that the perturbation budget remains unchanged, e.g., $\epsilon = 8$).

References

[1] Fan Bao, Shen Nie, Kaiwen Xue, Chongxuan Li, Shi Pu, Yaole Wang, Gang Yue, Yue Cao, Hang Su, and Jun Zhu. One transformer fits all distributions in multi-modal diffusion at scale. In *International Conference on Machine Learning (ICML)*, 2022.

[2] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[3] Jiaxian Guo, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Boyang Li, Dacheng Tao, and Steven Hoi. From images to textual prompts: Zero-shot visual question answering with frozen large language models. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[4] Alexandre Lacoste, Alexandra Luccioni, Victor Schmidt, and Thomas Dandres. Quantifying the carbon emissions of machine learning. *arXiv preprint arXiv:1910.09700*, 2019.

[5] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International Conference on Machine Learning (ICML)*, 2022.

[6] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*, 2023.

[7] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014.

[8] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *arXiv preprint arXiv:2304.08485*, 2023.

[9] Midjourney. Midjourney website, 2023. https://www.midjourney.com.

[10] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International Conference on Machine Learning*, pages 8821–8831. PMLR, 2021.

[11] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022.

[12] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10684–10695, 2022.

[13] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[14] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*, 2023.