Ask, Attend, Attack: A Effective Decision-Based Black-Box Targeted Attack for Image-to-Text Models

Anonymous Author(s)
Affiliation
Address
`email`

Overview

In this appendix, we describe implementation details, additional experiment results and analyses to support the methods proposed in the main paper. In addition, we show more examples of black-box adversarial attacks using *AAA*, each of which includes clean image, attention heatmap, adversarial image, optimization curve, target text, output text, and attack performance.

Reproducibility

Our **source code** and **data** are included in the supplemental material and uploaded, and we will publish the code on GitHub after the paper is accepted. We provide concise and understandable pseudo-code below.

Contents

Submitted to 38th Conference on Neural Information Processing Systems (NeurIPS 2024). Do not distribute.

A Additional implementation details

A.1 Pseudo code of our proposed framework

Algorithm 1 Ask, Attend, Attack (AAA) Framework

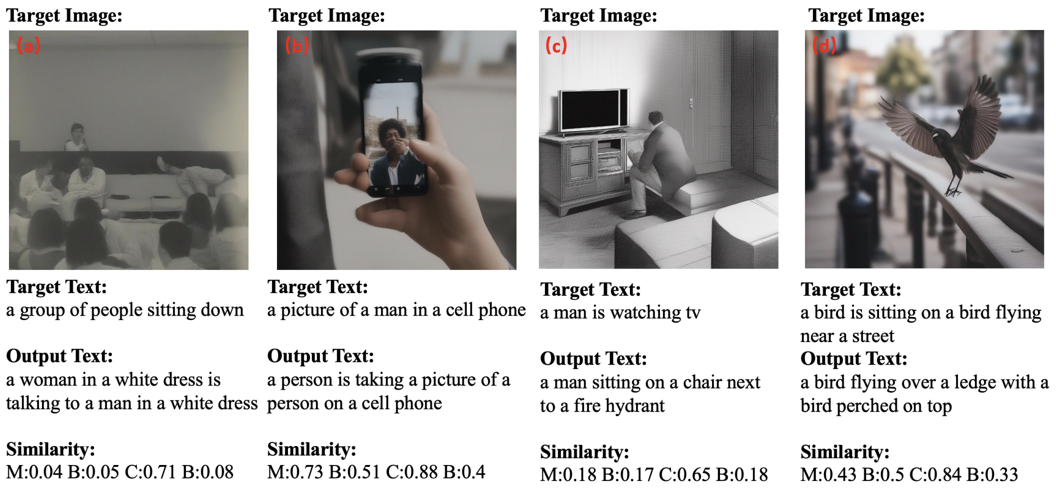
```
Input: Image $\mathbf{x}$, Target text $y_t$, Target semantics $TS$, Surrogate model $f$, Pre-trained CLIP model $E$
Output: Adversarial image $\mathbf{x_{adv}}$ that generates text $y_{adv}$ semantically similar to $y_t$
Initialize hyperparameters: population size NP, mutation factor $F$, crossover probability CR, perturbation threshold $\epsilon$, maximum search range $\eta$
Initialize target semantic dictionary $\mathbf{D} \leftarrow \emptyset$
function ASK($\mathbf{x}$, $TS$)
    Generate initial population with perturbations using Eq. (2)
    for each generation $g$ do
        Perform mutation using Eq. (3)
        Perform crossover using Eq. (4)
        Calculate semantic similarity $S_{sem}$ using Eq. (5)
        Select offspring based on $S_{sem}$ using Eq. (6)
        Update $\mathbf{D}$ with relevant words from $\mathcal{G}(\mathbf{x}_j^{g+1})$ using Eq. (7)
    end for
    return $\mathbf{D}$
end function
function ATTEND($\mathbf{x}$, $y_t$, $f$)
    Determine the category $c^*$ closest to $y_t$ using Eq. (9)
    Attention heatmap $\mathbf{A}$ is calculated by surrogate model $f$ using Eq. (8)
    return $\mathbf{A}$
end function
function ATTACK($\mathbf{x}$, $y_t$, $\mathbf{A}$)
    Generate initial population with attention-guided perturbations using Eq. (10)
    for each generation $g$ do
        Perform CurrentToBest mutation using Eq. (11)
        Perform crossover using Eq. (4)
        Calculate deep feature similarity $S_{clip}$ using Eq. (12)
        Select offspring based on $S_{clip}$ using Eq. (13)
    end for
    return Best individual as $\mathbf{x_{adv}}$
end function
$\mathbf{D} \leftarrow$ ASK($\mathbf{x}$, $TS$)
$y_t \leftarrow$ The attacker create a sentence from the dictionary $\mathbf{D}$
$\mathbf{A} \leftarrow$ ATTEND($\mathbf{x}$, $y_t$, $f$)
$\mathbf{x_{adv}} \leftarrow$ ATTACK($\mathbf{x}$, $y_t$, $\mathbf{A}$)
```

A.2 Basic setups

We set the population size NP to 40, scaling factor F to 0.7, cross probability factor CR to 0.7, γ to 0.5, α to 1, and θ to 3, and η to ϵ required in the experiment divided by the average of attention heatmap $\mathbf{A}$. Our device uses three GPUs of RTX2080ti with 11GB memory, and a CPU of Intel(R) Core(TM) i5-10400F. Our operating system is linux, the evolutionary algorithm framework uses the Geatpy library, and the deep learning framework uses Pytorch.

A.3 Standard deviation in the experiments

In the quantitative experiment of our paper, experiments were repeated for 10 times, and the optimal performance was obtained for each experiment, and the mean value and standard deviation were finally obtained.



Target Image: (a)

Target Text:
a group of people sitting down

Output Text:
a woman in a white dress is talking to a man in a white dress

Similarity:
M:0.04 B:0.05 C:0.71 B:0.08

Target Image: (b)

Target Text:
a picture of a man in a cell phone

Output Text:
a person is taking a picture of a person on a cell phone

Similarity:
M:0.73 B:0.51 C:0.88 B:0.4

Target Image: (c)

Target Text:
a man is watching tv

Output Text:
a man sitting on a chair next to a fire hydrant

Similarity:
M:0.18 B:0.17 C:0.65 B:0.18

Target Image: (d)

Target Text:
a bird is sitting on a bird flying near a street

Output Text:
a bird flying over a ledge with a bird perched on top

Similarity:
M:0.43 B:0.5 C:0.84 B:0.33

Figure 1: More examples of semantic loss of existing gray-box targeted attacks. The target text is the error-generated text of the image-to-text model that the attacker wants to obtain. The target image is the image generated by using the text-to-image model (Stable Diffusion) based on the target text. The output text is based on the target image using the image-to-text target model (VIT-GPT2/Show-Attend-Tell). similarity indicates the similarity between the target text and the output text. We also show the similarity between the target text and the output text. M stands for METEOR score, B for BLEU score, C for CLIP score, and S for SPICE score.

A.4 Evaluation metrics

(1) iteration, the number of iterations for the differential evolution algorithm in *Attack* to find the optimal solution (no more fitness convergence). Fewer iterations mean fewer queries and faster attack. (2) ϵ, the mean perturbation size of each pixel of the adversarial sample. Smaller value means higher concealment of adversarial perturbation. (3) diversity, the number of words in the target semantic dictionary from *Ask*. More words mean more diversity. (4) correlation, the average CLIP score between each word in the target semantic dictionary and the target semantics. The higher correlation, the more relevant the words in the target semantic dictionary are to the target semantics.

B Additional experiments

B.1 Analysis of semantic loss

We show more examples of the semantic loss phenomenon, as shown in Figure 1. In order to realize the targeted attack with the existing gray-box methods, it is necessary to convert the target text into the target image with the help of text-to-image model (such as Stable Diffusion). Then the distance between the adversarial image and the target image is narrowed, so that the text decoder of the image-to-text target model mistakes the adversarial image as the target image and outputs the description of the target image incorrectly. The target image often contains more semantic information than the target text, and the image-to-text target model may focus on the semantic information that is not specified by the attacker, which leads to semantic loss. For example, in Figure 1 (c), the text-to-image model generates the target image corresponding to the target text (*a man is watching tv*) very accurately, and the image-to-text target model also generates the output text (*a man sitting on a chair next to a fire hydrant*) of the target image very accurately, but the output text and the target text are very different. This means that even if there is a gray-box method that can completely make the features of the adversarial image identical to the features of the target image, the image-to-text target model can only generate the output text after semantic loss, and the targeted attack performance is limited by semantic loss.

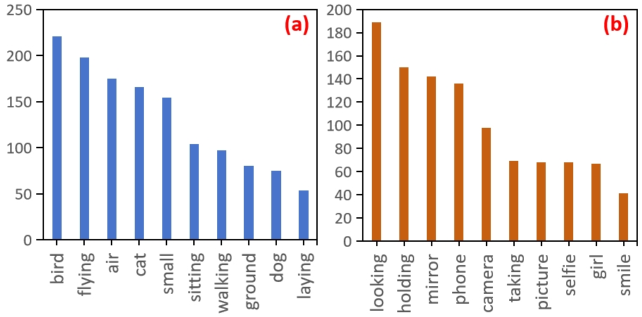


Figure 2: The top ten words in the target semantic dictionary for different target semantics, with the word frequency on the vertical axis. (a) is for *animal*; (b) is for *photograph*.

B.2 Comparison experiment of target semantic dictionary

We showed the target semantic dictionary's diversity and correlation for different perturbations in Table 1. More perturbation means more word choices for the target text. The correlation between dictionaries and target semantics is not affected by the size of perturbations. We also see that one vague word for target semantic makes more diversity and relevance in the dictionary than the detailed sentences. This is because a word has vague semantics, resulting in more words that are closer to the input image in the feature space being added to the dictionary. So we suggest using simple words as target semantics, as attackers can get richer dictionaries to make target text.

Table 1: Target semantic dictionaries for different semantics. *animal* word means the vague word *animal*, while *animal* sentence means *a dog is running after a cat*. *photograph* word means the vague word *photograph*, while *photograph* sentence means *a photo of a parking lot*.

semantic	*animal* word			*animal* sentence			*photograph* word			*photograph* sentence		
ϵ	10	15	25	10	15	25	10	15	25	10	15	25
diversity↑	50.6	65.4	90.1	38.9	54.1	79.6	51.7	62.5	87.7	43.1	52.6	75.5
correlation (%)↑	0.746	0.742	0.744	0.653	0.65	0.654	0.842	0.841	0.843	0.765	0.761	0.758

Table 2: Output text under different word selection strategies.

Strategy	Target Text	Output Text	Similarity
A	a bird is flying through air	a bird is flying through the air	great
A	a girl is taking pictures by camera	a girl is using a camera to take pictures	great
B	a camera is flying through the air	a man is holding a camera	medium
C	a giraffe is eating grass	a person is cutting a piece of food	bad
C	a boy is capturing a beautiful moment	a man is looking at his cell phone	bad
D	the helicopter is hovering in the sky	a man is holding a knife in his hand	bad

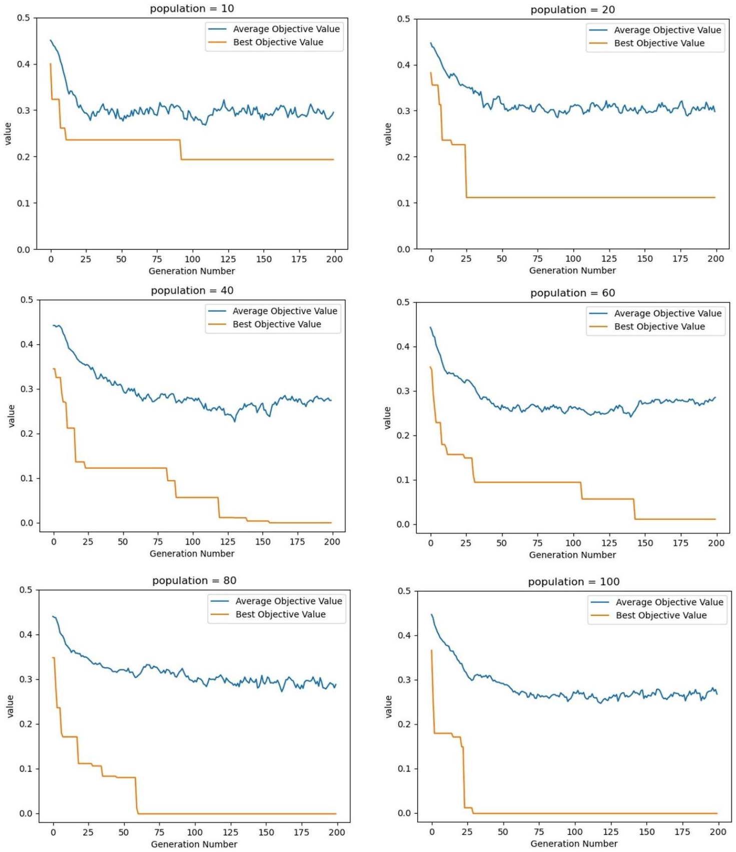


Figure 3: The best fitness curve and average fitness curve of the population under the same target text and different population sizes.

Table 3: The time (s) required for one optimization iteration under different population sizes.

population	10	20	40	60	80	100
iteration time	0.41	0.65	1.14	1.65	2.14	2.56

B.3 Word selection strategies for target semantic dictionaries

We showed the words and frequencies in the target semantic dictionary for different semantics in Figure 2. We compared different word selection strategies for targeted attacks with these dictionaries. The results show that: (1) Words in the dictionary do better when the semantics are similar, while words outside may fail; (2) Words from two dictionaries in one sentence decrease the performance.

We used four word selection strategies based on two dictionaries in Figure 2 to compare how different target texts y_t affect our method: (A) All words in y_t are from the same dictionary; (B) Some words in y_t are from each of the two dictionaries; (C) y_t is artificially created with the target semantics (*animal* or *photograph*), but without any words from the target semantic dictionary; (D) y_t is artificially

Table 4: Performance ($\%$) of different evolutionary algorithms and average number of iterations to find the optimal solution.

	CTB-DE	R-DE	S-GA
iteration↓	46.47±37.11	57.35±43.62	**15.41±11.52**
METEOR↑	**0.696±0.209**	0.538±0.264	0.327±0.254
BLEU↑	**0.658±0.219**	0.546±0.218	0.279±0.172
CLIP↑	**0.95±0.291**	0.871±0.112	0.748±0.096

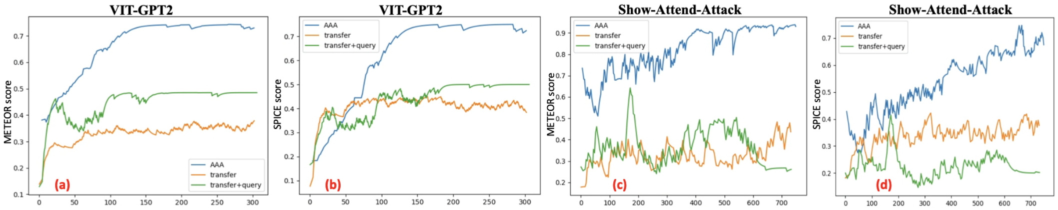


Figure 4: Comparison of computation time for generating a single adversarial sample using different adversarial attack methods. The y-axis is a measure of similarity between the generated text and the target text, with higher values indicating better target attack performance. The x-axis represents the computation time, and the shorter the time required to find a stable solution, the better.

created with different semantics from both target semantics (*animal* and *photograph*), and without any words from either target semantic dictionary. The output texts of the adversarial images obtained from different y_t word selection strategies are shown in Table 2.

The first row of Table 2 shows that strategy (A) can achieve a strong targeted attack, making the output text very similar to the target text. This is because words in the same dictionary are close to each other in the feature space. Strategy (B) selects the words *flying* and *air* from dictionary *animal* in Figure 2 (a), and *camera* from dictionary *photograph* in Figure 2 (b), to form the target text. The third row of Table 2 shows that the output text and the target text y_t are not very similar. The output text only contains the word *camera* in dictionary (b). This is because the feature distance between the two dictionaries is large, even though they are both close to the input image and easy to search in the feature space. It is hard to optimize the target text y_t that contains words from both target semantic dictionaries. Strategy (C) randomly creates y_t based on the *animal* and *photograph* semantics, without using any words from dictionary (a) and (b). For example, *giraffe* is an *animal*, but not in dictionary (a), and *capture beautiful moment* is related to *photograph*, but not in dictionary (b). The output text and the target text y_t are totally different, indicating a failed targeted attack. Strategy (D) randomly creates y_t with different semantics from both target semantics, and without any words from either target semantic dictionary. The targeted attack also fails. Therefore, we recommend selecting words from one target semantic dictionary for the target text y_t, which will greatly improve the success rate of our method's targeted attack.

B.4 Comparison experiment on population size

We show convergence curves with the same target text but different population sizes NP to observe how they affect the optimization iteration process of *Attack*. Figure 3 shows that when NP is 10 and 20, the best fitness values are 0.2 and 0.1, corresponding to CLIP scores of 0.8 and 0.9 for the output texts and target texts, respectively. When NP is larger than 40, the output text and the target text are completely consistent (CLIP score = 1). This means that a larger NP can find better solutions with fewer iterations [1, 2]. However, a larger NP also increases the computation time per iteration, as Table 3 shows. Moreover, as this is a large-scale optimization problem with 196608 decision variables per individual, a larger NP demands more hardware resources [3, 4]. Considering all factors, we set the population size NP to 40.

B.5 Comparison experiment on computation time

In Figure **??** of the main paper, we show the computational efficiency of two metrics, CLIP score and BLEU score. In this part, we will supplement the other two metrics, METEOR score and SPICE score. As shown in Figure 4, the computation time of the existing gray-box attack methods to find the optimal solution is still shorter than that of our black-box attack method. For example, the transfer approach [5] illustrated in Figure 4(a) produces an adversarial sample with a METEOR score of 0.34 within a mere 62 seconds, while the transfer+query approach [6] achieves a METEOR score of 0.49 in just 119 seconds. Conversely, our *AAA* method requires 179 seconds to generate an adversarial sample with a superior METEOR score of 0.75. Because our method is more practical and performs better, the additional computation time is acceptable.

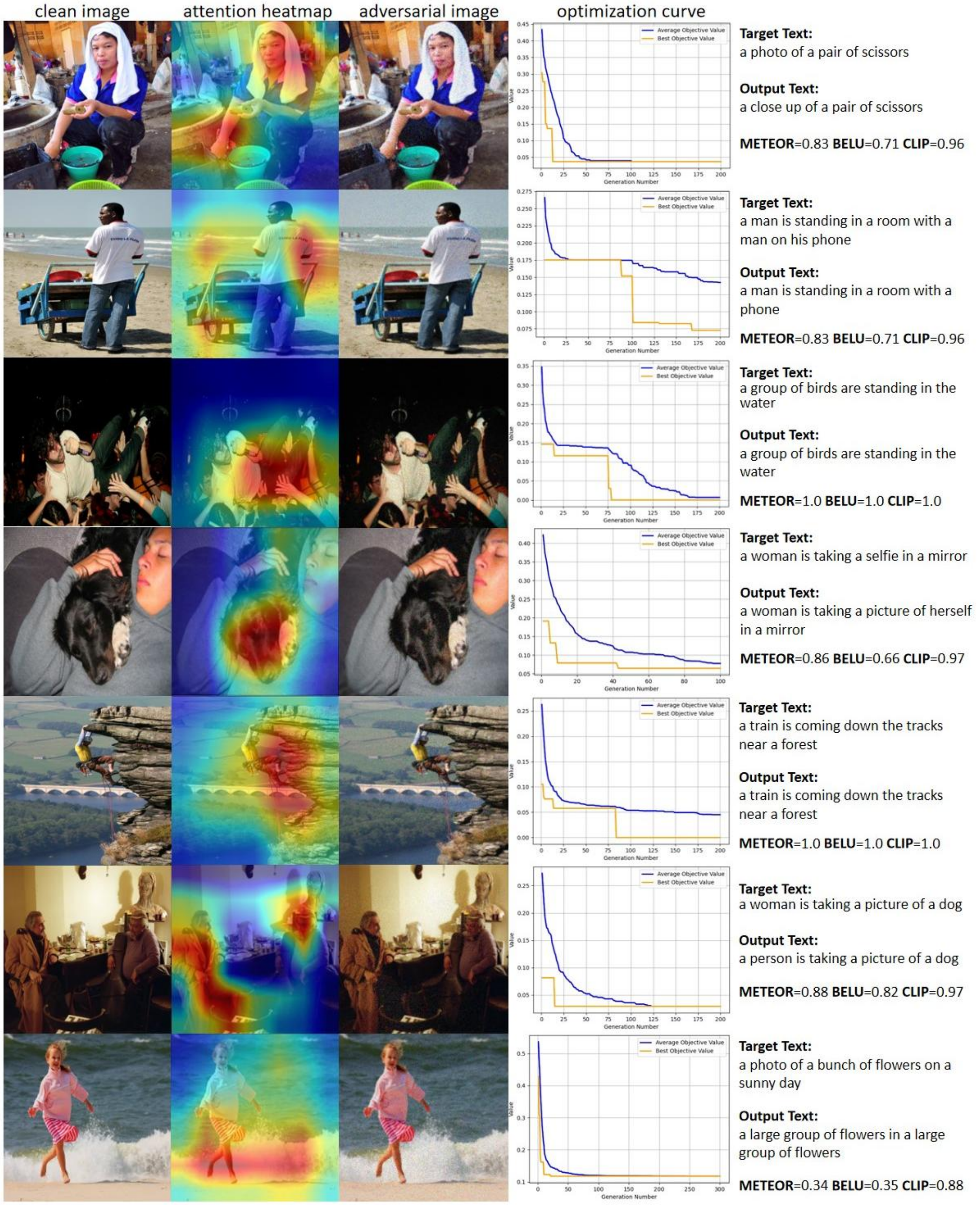


Figure 5: Attention heatmaps, optimization convergence curves, target text, output text and attack performance for more adversarial samples.

B.6 Comparison experiment of optimization algorithms

We compared different optimization strategies in *Attack*: CurrentToBest Differential Evolution (CTB-DE) [4], Rand Differential Evolution (R-DE) [3], and Stud Genetic Algorithm (S-GA) [7]. Table 4 shows that the genetic algorithm needs the fewest iterations, but easily gets stuck in local optima, leading to poor attack performance. Differential evolution needs more iterations but finds better solutions. Also, the CurrentToBest mutation does better and faster than the random mutation. So we adopted the CurrentToBest differential evolution strategy in *Attack*.

B.7 Visualization of more adversarial samples

We presented attention heatmaps $\mathbf{A}$, optimization convergence curves, target text y_t, and output text for more adversarial samples, as shown in Figure 5.

C Discussion

C.1 Limitation

Our work represents the first black-box targeted attack on image-to-text models, with the core idea utilizing evolutionary algorithms to solve a large-scale optimization problem. The drawbacks of evolutionary algorithms, which are also the limitations of our work, include: (1) **Low optimization efficiency**. Gradient-based algorithms use the gradient information of the objective function, which is a powerful guide regarding the optimization direction. Evolutionary algorithms do not directly use gradient information but search through random mutation and crossover operations. Compared to gradient optimization algorithms, evolutionary algorithms require more iterations to find the optimal solution. (2) **High number of queries**. Each individual in the population requires access to the target model in every iteration, and the service provider of the image-to-text target model can simply set a limit on the number of accesses to defend against our attack.

C.2 Future work

Our black-box targeted attack framework *Ask, Attend, Attack* on image-to-text models employs classic evolutionary algorithms. In our future work, we will explore how our framework *AAA* can be combined with the current state-of-the-art (SOTA) evolutionary algorithms, which have the fastest convergence efficiency, to mitigate the limitations mentioned above.

References

[1] Zhenzhong Wang, Haokai Hong, Kai Ye, Guangen Zhang, Min Jiang, and Kay Chen Tan. Manifold interpolation for large-scale multiobjective optimization via generative adversarial networks. *IEEE Transactions on Neural Networks and Learning Systems*, 34(8):4631–4645, 2023.

[2] Zhenzhong Wang, Qingyuan Zeng, Wanyu Lin, Min Jiang, and Kaychen Tan. Generating diagnostic and actionable explanations for fair graph neural networks. In *Proceedings of the Association for the Advancement of Artificial Intelligence (AAAI)*, 2024.

[3] Rainer Storn and Kenneth Price. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*, 11:341–359, 1997.

[4] Jingqiao Zhang and Arthur C Sanderson. Jade: adaptive differential evolution with optional external archive. *IEEE Transactions on evolutionary computation*, 13(5):945–958, 2009.

[5] Raz Lapid and Moshe Sipper. I see dead people: Gray-box adversarial attack on image-to-text models. In *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML-PKDD)*, 2023.

[6] Yunqing Zhao, Tianyu Pang, Chao Du, Xiao Yang, Chongxuan Li, Ngai-Man Cheung, and Min Lin. On evaluating adversarial robustness of large vision-language models. In *Proceedings of the Neural Information Processing Systems (NIPS)*, 2023.

[7] Wael Khatib and Peter J. Fleming. The stud ga: a mini revolution? In *Proceedings of the International Conference on Parallel Problem Solving from Nature (PPSN)*, pages 683–691. Springer Berlin Heidelberg, 1998.