RIATIG: Reliable and Imperceptible Adversarial Text-to-Image Generation with Natural Prompts

Han Liu\textsuperscript{1}, Yuhao Wu\textsuperscript{1}, Shixuan Zhai\textsuperscript{1}, Bo Yuan\textsuperscript{2}, Ning Zhang\textsuperscript{1}

\textsuperscript{1}Washington University in St. Louis, \textsuperscript{2}Rutgers University

\{h.liu1,yuhao.wu,zhais,zhang.ning\}@wustl.edu, bo.yuan@soe.rutgers.edu

Abstract

The field of text-to-image generation has made remarkable strides in creating high-fidelity and photorealistic images. As this technology gains popularity, there is a growing concern about its potential security risks. However, there has been limited exploration into the robustness of these models from an adversarial perspective. Existing research has primarily focused on untargeted settings, and lacks holistic consideration for reliability (attack success rate) and stealthiness (imperceptibility).

In this paper, we propose RIATIG, a reliable and imperceptible adversarial attack against text-to-image models via inconspicuous examples. By formulating the example crafting as an optimization process and solving it using a genetic-based method, our proposed attack can generate imperceptible prompts for text-to-image generation models in a reliable way. Evaluation of six popular text-to-image generation models demonstrates the efficiency and stealthiness of our attack in both white-box and black-box settings. To allow the community to build on top of our findings, we’ve made the artifacts available\textsuperscript{1}.

1. Introduction

The text-to-image generation has captured widespread attention from the research community with its creative and realistic image generation capability [52, 55, 56]. The ability to generate text-consistent images from natural language descriptions could potentially bring tremendous benefits to many areas of life, such as multimedia editing, computer-aided design, and art creation [23, 27, 37, 43].

Driven by recent advances in models trained with large datasets [42, 57] and multimodal learning [45] (e.g., diffusion models [17]), text-to-image generation has made significant progress in synthesizing high-fidelity and photorealistic images, such as DALL·E [43], DALL·E 2 [42] and Imagen [45]. At the same time, there are a growing number of ethical concerns about the potential misuse of this technology [36, 45, 53]. Generative models could be used to generate synthetic video/audio/images of individuals (e.g., Deepfakes [14]), or synthetic contents with harmful stereotypes, violence, or obscenities [10, 45, 53]. To prevent the generation of such harmful content, content moderation filters are deployed in public APIs (e.g., DALL·E 2) to filter unsafe text prompts that may lead to harmful content. However, despite their best intentions, existing model-based text filters remain susceptible to adaptive adversarial attacks.

Deep neural networks (DNNs) have been shown to be vulnerable to adversarial examples [9, 31, 32]. By applying these techniques, it is possible to craft an adversarial text that looks natural to bypass the content filters, yet generates a completely different category of potentially malicious images. However, the adversarial attacks on text-to-image generators are less explored. To the best of our knowledge, there are two closely related studies [15, 36]. Nevertheless, two challenges remain:

\textsuperscript{1}Code is available at: https://github.com/WUSTL-CSPL/RIATIG

Figure 1. Examples of RIATIG attacks. The top texts are the adversarial prompts, and the bottom texts are the target models. The first row represents the target images, while the second row represents the adversarial image generated by the prompt.
1) Reliability. One significant limitation is that the existing works [15, 36] do not offer a reliable method to find adversarial examples. [15] discovers that DALL-E 2 has certain hidden vocabularies that can be used to generate images with some absurd (non-natural) prompts; however, this vocabulary is often limited and not stealthy (natural). Built on evocative prompting, [36] crafts adversarial examples via the morphological similarity between existing words. However, it is very difficult to find texts with such linguistic similarity, and as a result, it can be challenging for this method to be adopted and generalized in different scenarios.

2) Stealthiness. The existing approaches can only craft adversarial examples that appear to be non-natural compared to normal texts or retain similar meanings, making them easily filtered and recognized by human examiners. For example, [15] crafts Apoploe vesreatiais to represent bugs, and [36] crafts falaiscoglieklippantilado to represent cliff. Also, [36] combines creepy and spooky into creepooky to generate an image that looks creepy and scary, yet the inferred meaning of this new word is highly related to the generated image, limiting its stealthiness.

To address these challenges, we propose RIATIG, a reliable and imperceptible adversarial attack against text-to-image models using natural examples. To achieve this, we first formulate the generation of the adversarial examples as an optimization problem and apply genetic-based optimization methods to solve it, thus making our methods much more reliable in finding working adversarial examples. Furthermore, in order to improve the stealthiness, we propose a new text mutation technique to generate adversarial text that is visually and semantically similar to its normal version (some example results are shown in Figure 1).

RIATIG is evaluated on six popular text-to-image models with both white-box and black-box attack settings. Experimental results show that compared with the state-of-the-art text-to-image-oriented adversarial attacks, RIATIG demonstrates significantly better performance in terms of attack effectiveness and sample quality. Overall, the contributions of this work are summarized as follows:

- We are the first to systematically analyze the adversarial robustness of text-to-image generation models in both the white-box and black-box settings.
- We propose genetic-based optimization methods to find natural adversarial examples reliably.
- We evaluate our attacks on six popular text-to-image generation models and compare our attacks with five baselines. The evaluation results show that our methods achieve a much higher success rate and sample quality, raising awareness of improving and securing the robustness of text-to-image models.

2. Related Work

Text-to-Image Generation. The research towards text-to-image generation starts with the work proposed by Mansimov et al. [33], which extends the Deep Recurrent Attention Writer (DRAW) [20] model to condition on image captions using a soft attention mechanism. Xu et al. [52] further propose a multi-stage attentional GAN (AttnGAN) to force the model to focus on the fine-grained information at the word level, leading to the generation of more semantically matched images. To address the limitations of the multi-stage methods, e.g., low-quality initial images and repeated use of the same word representations, Zhu et al. [58] propose a dynamic memory GAN (DM-GAN), which uses a memory network to write and read encoded prior dynamically in each image-refined process. To further improve the quality of the generator, Tao et al. [49] propose a deep Fusion Generative Adversarial Network (DF-GAN) that performs deep fusion between text and visual features. Recently, a set of large-scale text-to-image generation models are proposed and released, demonstrating promising performance. DALL-E, an autoregressive transformer model trained on 250 million web-crawled image-text pairs [43], achieves zero-shot high-quality image generation on the MS-COCO dataset. Its updated version, DALL-E 2, adopts contrastive model CLIP [42]. By feeding a text caption and using a diffusion model to generate an image conditioned on the image embedding, DALL-E 2 can synthesize amazingly photorealistic images. Imagen is another state-of-the-art text-to-image generator built on the combination of a transformer trained on text data and a high-fidelity diffusion model, achieving a very high generation performance.

Adversarial Attacks on Text-to-Image Generators. The adversarial robustness of text-to-image generation is a relatively less studied topic. [15] is the pioneering work that studies the vulnerability of text-to-image generators, such as DALL-E 2. They found it is possible to leverage gibberish text in the target image to evade content filters by using such gibberish texts as prompts rather than semantically equivalent text. However, most of the gibberish text does not reliably produce a specific category of images. Later, Milli`ere proposes two methods of crafting adversarial text to generate specific pictures [36]. The first solution is macaronic prompting, which composes multilingual subword segments to generate specific types of images. The second approach is evocative prompting, a strategy that uses prompts possessing similarity of broad morphological features to generate semantic-consistent images with real concepts. In addition, Struppek shows that replacing a single character in the text with visually similar non-Latin characters can induce cultural biases into the generated images or even hide complete objects in the generated images. However, their method is untargeted, i.e., they cannot generate
Firstly, unlike existing work on textual adversarial examples, we consider a scenario where the adversary cannot return confidence scores to the adversary. This is because classification models [8, 21, 24, 28], the generation models [8, 24, 29], and easily detectable by human examiners. The loop ends when the generated images are highly similar to the target images yet the adversarial text is distinct from the target text. Then the result samples go through the quality fine-tuning stage where we propose two techniques to further improve the imperceptibility of adversarial examples. In the following section, we will first introduce the similarity measurement for the offspring selection (§4.3), then elaborate on the genetic-based optimization with mutation and crossover scheme (§4.4), before finally describing the sample quality improvement procedure (§4.5).

4.3. Similarity Measurement

To find a suitable measurement metric, we need to consider the natural continuity of visual space and find a proper

\[
\text{argmax }_{x^*} S_t(G(x^*), y_t) \text{ s.t. } D_t(x^*, x_t) > \eta, \tag{1}
\]

where \(S_t\) is a semantic similarity function of images, \(D_t\) is a function measuring the semantic distance of texts, and \(\eta\) is the threshold of the semantic dissimilarity of \(x_t\) and \(x^*\).

4.2. Challenges and Overview

**Challenges.** Designing a reliable and imperceptible adversarial attack against text-to-image models can face several challenges. Firstly, unlike existing work on textual adversarial attacks [8, 24, 28], where the output of the model is a finite number of classes (e.g., positive and negative in sentiment analysis), the output of a text-to-image model is a high-dimensional continuous class with nearly infinite combinations. This characteristic makes it difficult to measure the effectiveness of the attacks. One possible solution is to measure the semantic distance of images, however, existing approaches only focus on category-level semantic distance [11, 16], which cannot handle the complex sentence context similarity. For example, simply representing a blue double decker school bus parking in let by school bus will lose important context information such as the color, size of the bus, and its surrounding environment. Second, since words can serve as discrete tokens in a sentence, it is impossible to optimize the input words directly by computing the gradient of the loss function with respect to the input word even assuming white-box knowledge in the adversary [8]. Third, unlike image domains where small perturbations can go unnoticed, any changes in the text domain are perceptible [8, 28, 29] and easily detectable by human examiners.

**Attack Overview.** We adopt a genetic-based approach to iteratively optimize the input samples towards the target samples, followed by a quality fine-tuning stage to further improve the imperceptibility. First, the mutation stage generates a large number of variants that are semantically and visually similar to the input text, then the text-to-image generation model outputs images accordingly. A similarity measurement is implemented to compute the quality of mutated examples, allowing us to select the best offspring, which will enter a crossover stage to exchange the desirable traits to obtain a better population for the next-round iteration. The loop ends when the generated images are highly similar to the target images yet the adversarial text is distinct from the target text. Then the result samples go through the quality fine-tuning stage where we propose two techniques to further improve the imperceptibility of adversarial examples. In the following section, we will first introduce the similarity measurement for the offspring selection (§4.3), then elaborate on the genetic-based optimization with mutation and crossover scheme (§4.4), before finally describing the sample quality improvement procedure (§4.5).

4.1. Problem Formulation

A text-to-image generation model \(G : \mathcal{X} \rightarrow \mathcal{Y}\) maps text space \(\mathcal{X}\) to image space \(\mathcal{Y}\). Since \(G\) is trained on semantically-aligned text-image pairs, a well-trained model should generate images that are semantically identical to the input text. Given an arbitrary target image \(y_t\), which corresponds to its naive text prompt \(x_t\), the adversary aims to craft an adversarial example \(x^*\) that generates a semantically similar image to \(y_t\), with the premise that \(x^*\) is semantically dissimilar from \(x_t\) enough to evade detection. This is an optimization program that can be formulated as

**Figure 2. Working mechanism of a text-to-image generator.**
representation of image semantic information. Recently, there is a growing research interest in learning image representation from text [40, 46, 57], which maps images into a semantic-rich embedding space by jointly learning with textual data. Among these efforts, the CLIP proposed by OpenAI has been trained on large-scale (i.e., 400 million) web-crawled image-text pairs [40], which can efficiently represent visual concepts aligned with the corresponding textual information. Motivated by its powerful capability, we adopt the pre-trained image encoder of CLIP to encode two images and then calculate the cosine distance between the encoded vectors measuring the similarity as

\[ S_i(G(x), y) = \frac{E_i(G(x)) \cdot E_i(y)}{\|E_i(G(x))\| \|E_i(y)\|}. \]  

(2)

Similarly, we utilize the text encoder to encode the text and calculate the cosine distance as their semantic distance,

\[ D_i(x, x^*) = 1 - \frac{E_i(x) \cdot E_i(x^*)}{\|E_i(x)\| \|E_i(x^*)\|}. \]  

(3)

When the semantic distance is above a certain threshold, the two texts are considered not semantically similar.

### 4.4. Genetic-based Optimization

Genetic algorithms are heuristic optimization methods that are inspired by nature selection. By evolving a population of candidates towards better solutions with performing mutation and crossover [22], the genetic algorithms can search a large solution space effectively [25], making them well suited in the optimization over the multi-dimensional input space. Therefore, in this work, we adopt a genetic-based algorithm to solve the target optimization problem. To be specific, the optimization process starts with an input text that is semantically dissimilar from the target images. The mutation is used to generate different variants to explore the search space and increase the diversity of the population. To generate initial high-quality variants, RIATIG first generates a large pool of mutated candidates, rendering the execution of crossover and mutation more effective. In each generation, we select the samples with the highest fitness score for crossover and mutation to generate the next generation of text samples. Cosine similarity scores calculated by Eq. 2 between the two images are the fitness score of the genetic algorithm.

**Crossover Scheme.** Crossover is used to exchange the desirable traits between the chosen populations, thereby possibly generating an even better population. For the crossover scheme used in RIATIG, a new sentence is synthesized by a pair of parent sentences. The probability of each population being selected as a parent is proportional to their fitness score, giving the beneficial traits a higher probability to stay.

**Mutation Scheme.** RIATIG performs sentence-level mutation. Each sentence has a mutation probability of \( p_m \) to be selected. The proper selection of \( p_m \) is very critical. When it is too high, the beneficial traits may be mutated without crossover to generate better offspring. On the other hand, very small \( p_m \) will make most of the populations unchanged, causing insufficient mutations to escape the local maximum. To address the challenge, \( p_m \) can be adaptively updated using a momentum mutation-based method [50] by

\[ p_m = \alpha \cdot p_{m-1} + \frac{\beta}{|S_i - S_{i-1}|}, \]  

(4)

where \( \alpha \) and \( \beta \) are scaling factors, and \( S_i \) denotes the maximum fitness score across the populations in \( i \)-th generation.

The mutation process occurs for each selected sentence, which is divided into two stages. In the first stage, we identify the important words in the original sentence that need to
be mutated. In the second stage, we propose two strategies to slightly modify a word while preserving its semantics.

**Stage-1: Selection of Important Word.** Because 1) the search space for mutation is very huge; and 2) most of the mutation variants do not play significant roles to improve the fitness score, identifying the most important words that contribute to fitness scores is crucial to improve the effectiveness of mutation. When we assume white-box knowledge, gradient information can be used to measure the importance, calculating the gradient of the fitness score with respect to the embedding of each word by

\[
g_{x_n} = \frac{\partial S_i(G(e_n), y_t)}{\partial e_n},
\]

where \(e_n\) is the embedding of word \(x_n\), \(y_t\) is the target image, \(G\) is the generation model, and \(S_i\) is the semantic similarity function defined in Eq. 2. When we assume black-box knowledge, the importance of a word is estimated by calculating the change in fitness scores before and after deleting the word, such a calculation process is defined as

\[
g_{x_n} = S_i(G(e), y_t) - S_i(G(e \backslash e_n), y_t),
\]

where \(e\) is the embedding vectors of sentence \(x\), \(e \backslash e_n\) is the embedding vectors after deleting the word \(x_n\) from \(x\). After that, we normalize the important scores as

\[
I_{x_n} = \frac{\|g_{x_n}\|}{\sum_{i=1}^{N} \|g_{x_i}\|}.
\]

Then each word \(x_n\) has a probability of \(I_{x_n}\) to be selected.

**Stage-2: Word Mutation.** Since our goal is to generate adversarial text to be visually and semantically similar to normal text for human understanding, we consider two types of mutation strategies that are used in natural language processing fields [28,29,51]. The first one is to inject character-level typos that mimic the grammar mistakes people may make. The rationale is that the semantic meaning of the text is very likely to be preserved after a few character changes [44]. More specifically, we adopt three types of text manipulations: (1) Insert extra spaces into words; (2) Swap the position of two characters randomly except for the first and last characters; (3) Delete one random character except for the first and last characters. The second type of text manipulation is to replace characters/words with visually or semantically similar counterparts. As for visually similar substitution, we utilize the LEET Speak Alphabet [7], which replaces characters in ways that play on the similarity of glyphs. For example, we may replace the letter \(l\) with the number 1, the letter \(o\) with the number 0, etc.

For the semantically similar substitution, we adopt the Word2vec model [34] to transform the words into embedding vectors, and then compute the cosine distance as the similarity measurement to select the nearest neighbors. A threshold is used to filter out candidates with large distances. Due to the large search space of optimization, only replacing it with target-irrelevant sentences will weaken the mutation, making it hard to find variants that generate images closer to the target. To solve this problem, we use the target sentences to perform semantic substitution, and the words with a close distance are filtered out. In addition, we adopt the WordNet [35] to decide the part-of-speech (POS) tag of the word, filtering out the candidates with different POS to maintain grammatical fluency. After filtering, we select the top-M nearest neighbors into the candidate pool. Next, Google’s one billion-word language model [13] is used to filter out words that do not fit within the context surroundings. To that end, we first fit words in the candidate pools with the surrounding context, then calculate their language scores that measure the fluency of the sentences, keeping the top-N word with the highest scores.

### 4.5. Sample Quality Improvement

After obtaining the adversarial examples, we further improve the quality of adversarial text based on two key observations. First, the presuppositions (e.g., on, in, etc.) in the sentences have little impact on the semantic meanings of generated images compared to the nouns and verbs. Second, the text embedding in the text-to-image generation model usually uses a dictionary to represent a finite set of possible words [52,58], meaning that the deliberately misspell out-of-vocabulary will be mapped to *unknown*. With the first observation, we iteratively replace or add all pre-defined presuppositions in different positions of sentences and test their naturalness. The presuppositions with the highest naturalness are chosen to improve the samples. We then further fine-tune the quality of samples based on the second observation. For example, both *paking* and *parnikg* are out-of-vocabulary words that are mapped to the *unknown*, and they are regarded as the same words by the text-to-image generation models. However, since the first word has fewer modifications, it is closer to an ordinary text and is preferred. Therefore, with white-box knowledge, we can test whether the adversarial text is the out-of-vocabulary words by feeding it to the embedding models. In the black-box attack setting, we adopt the GloVe embedding models [38] pre-trained on the Common Crawl dataset containing 2.2 million vocabularies as a surrogate embedding model.

### 5. Experiments

#### 5.1. Experiment Settings

**Dataset.** We choose the initial texts, target texts, and target images from the Microsoft COCO [30] dataset, a popular benchmark for training and evaluating text-to-image generation models. The dataset includes 82,783 training images and 40,504 testing images, each with 5 text descriptions.
Target Models. To evaluate the performance of our proposed attack, we choose three representative public text-to-image generation models: AttnGAN [52], DM-GAN [58], and DF-GAN [49]. We use the pre-trained models on the COCO dataset released in their official GitHub repository [1, 4, 5] as the target models. We also choose three large-scale proprietary models: DALL-E [43], DALL-E 2 [42], and Imagen [45]. None of the above have officially released their pre-trained models, nor public APIs except DALL-E 2. Therefore, for DALL-E, we adopt the released pre-trained models of DALL-E mini [3], which achieves comparable performance to that of DALL-E. Similarly, for Imagen, we adopt the pre-trained models released by Hugging Face [6]. For DALL-E 2, though a public API exists, only 15 free credits are granted to each account, so we first train adversarial examples on LAION’s pre-trained model [2], then access the APIs to check if it was transferred successfully.

Evaluation Setup. In our experiment, we attacked the AttnGAN, DM-GAN, and DF-GAN in white-box settings, and we attacked DALL-E, DALL-E 2, Imagen in black-box settings, which means we can only query the model and get the generated images. We set the initial population size $M$, the total population size $N$, and the selected population size $K$ to 300, 50, and 25, respectively. The initial mutation probability $p_m$ is set to 0.8. Ten adversarial examples are generated for each model. For each generation, we randomly select a target image and a semantic unrelated text from the COCO dataset. For example, we select the target images with the meaning "a clock is flying in the blue sky" and the original text "a red double decker bus parking in let.

Comparison with Baselines. To the best of our knowledge, there are only two related studies [15, 36] for a targeted adversarial attack against text-to-image generation models. [15] proposes a hidden vocabulary attack (HiddVocab) on DALL-E 2 and [36] implements macaronic prompting (MacPromp) and evocative prompting (EvoPromp) attacks against DALL-E mini and DALL-E 2. For a comprehensive comparison, we also include [48], which is a homolytic substitution-based attack (HomoSubs) to cause image content obfuscation. We also include one state-of-the-art textual adversarial attack, TextFooler [24]. Since their attack does not involve image generation, we change the posterior tasks to match our settings and apply our proposed similarity metric to guide the optimization.

5.2. Evaluation Metrics

To demonstrate the effectiveness of our attack, we evaluate both attack effectiveness and adversarial sample imperceptibility. The imperceptibility measure consists of two parts: semantic meaning similarity and sentence naturality.

Effectiveness of Generated Images. We adopt the method of R-precision, a widely-used metric in text-to-image generation models [27, 39, 52, 54, 58] to measure whether the generated image is semantically aligned with the given text description. Specifically, we use the generated images to perform a retrieval experiment by querying a candidate set of text descriptions that consist of one ground truth and 99 randomly selected mismatched descriptions. The ViLT model [26], which achieves state-of-the-art performance on image-text retrieval tasks, is used here. We examine the top $R$ ranked retrieval results and consider our attacks successful when the ground truth appears in the retrieval. We evaluate the effectiveness when $R$ is 1 (R-1) and $R$ is 3 (R-3).

Sentence Semantic Meaning. Following [24, 28], we first adopt the Universal Sentence Encoder [12] to encode the adversarial sentence and target sentence into high dimensional vectors, and then their cosine similarity score are calculated to measure the semantic similarity.

Sentence Naturality. The adversarial examples should look natural, similar to normal sentences in the real world. In our experiment, we calculate the perplexity (PPL) of the adversarial examples with GPT-2 [41], which is trained on a large corpus of real-world sentences. PPL measures how likely the model is to generate the input text sequence, representing the fluency of the adversarial examples. Generally, the sample with lower PPL is more natural.

5.3. Evaluation Results

White-box Attacks. The results for white-box attacks are shown in Table 1. It is seen that RIATIG succeeds in generating all adversarial examples in DM-GAN and DF-GAN, and can succeed in generating 9 out of 10 samples in AttnGAN when we examine top-3 recall results. The recall precision drops when we only examine top-1 recall results. This phenomenon might be caused by the image generation quality of the target model, since the DM-GAN and AttnGAN achieve an average R-precision of 85.47% [52] and 88.56% [58] on COCO datasets. Furthermore, a low semantic distance means there is a low semantic relevance between the adversarial text and the text used to generate the target images. The low PPL means our adversarial text is fluent and natural as compared to the normal text. On the other hand, due to the limited-size and monolingual datasets that these three models are trained on, the baselines can not succeed in generating adversarial examples.

Black-box Attacks. The results for black-box attacks are shown in Table 2. RIATIG achieves a high R-precision as compared to the baseline methods. HomoSubs and TextFooler are designed for untargeted attacks, and cannot be easily adapted for targeted attacks. HiddVocab does not provide a reliable method for finding adversarial examples and can achieve only a 40% success rate against DALL-E 2. Though MacPromp and EvoPromp can achieve a higher
A woman is standing on grass. Zeedonk is on the side of a hill. A train is passing through a village. The viarters are on the center of ocean. A clock is flying in the blue phoebe. A chinkapin is on the field. A blue double-decker bus brightly colored kites in fluorescent orange.

Figure 4. Examples of our adversarial attack results against DALL-E mini, DALL-E 2, and Imagen.

Table 1. The results of the white-box attacks against AttnGAN, DM-GAN, and DF-GAN.

<table>
<thead>
<tr>
<th>Model</th>
<th>R-1 ↑</th>
<th>R-3 ↑</th>
<th>Semantic Distance ↓</th>
<th>PPL ↓</th>
</tr>
</thead>
<tbody>
<tr>
<td>AttnGAN [52]</td>
<td>8/10</td>
<td>9/10</td>
<td>0.24</td>
<td>522.63</td>
</tr>
<tr>
<td>DM-GAN [58]</td>
<td>8/10</td>
<td>10/10</td>
<td>0.27</td>
<td>420.16</td>
</tr>
<tr>
<td>DF-GAN [49]</td>
<td>9/10</td>
<td>10/10</td>
<td>0.34</td>
<td>558.81</td>
</tr>
</tbody>
</table>

Note: R-1 and R-3 represent the R-precision when R=1 and R=3, respectively. PPL represents the perplexity of sentences.

Table 2. The results of the black-box attacks against DALL-E mini, DALL-E 2, and Imagen.

<table>
<thead>
<tr>
<th>Model</th>
<th>Methods</th>
<th>R-1 ↑</th>
<th>R-3 ↑</th>
<th>Sem. Dist. ↓</th>
<th>PPL ↓</th>
</tr>
</thead>
<tbody>
<tr>
<td>DALL-E mini [43]</td>
<td>MacPromp [36]</td>
<td>8/10</td>
<td>8/10</td>
<td>0.46</td>
<td>6814.61</td>
</tr>
<tr>
<td>DALL-E mini [43]</td>
<td>HomoSubs [48]</td>
<td>0/10</td>
<td>0/10</td>
<td>0.09</td>
<td>1115.84</td>
</tr>
<tr>
<td>DALL-E mini [43]</td>
<td>TextFooler [24]</td>
<td>0/10</td>
<td>0/10</td>
<td>0.18</td>
<td>1848.41</td>
</tr>
<tr>
<td>DALL-E mini [43]</td>
<td>RIATIG</td>
<td>9/10</td>
<td>10/10</td>
<td>0.27</td>
<td>420.16</td>
</tr>
<tr>
<td>DALL-E 2 [42]</td>
<td>MacPromp [36]</td>
<td>9/10</td>
<td>9/10</td>
<td>0.39</td>
<td>4931.31</td>
</tr>
<tr>
<td>DALL-E 2 [42]</td>
<td>EvoPromp [36]</td>
<td>7/10</td>
<td>7/10</td>
<td>0.17</td>
<td>6139.39</td>
</tr>
<tr>
<td>DALL-E 2 [42]</td>
<td>HomoSubs [48]</td>
<td>0/10</td>
<td>0/10</td>
<td>0.06</td>
<td>981.69</td>
</tr>
<tr>
<td>DALL-E 2 [42]</td>
<td>TextFooler [24]</td>
<td>0/10</td>
<td>0/10</td>
<td>0.18</td>
<td>2624.88</td>
</tr>
<tr>
<td>DALL-E 2 [42]</td>
<td>RIATIG</td>
<td>10/10</td>
<td>10/10</td>
<td>0.27</td>
<td>872.03</td>
</tr>
<tr>
<td>Imagen [45]</td>
<td>HiddVocab [15]</td>
<td>2/10</td>
<td>2/10</td>
<td>0.18</td>
<td>5073.74</td>
</tr>
<tr>
<td>Imagen [45]</td>
<td>EvoPromp [36]</td>
<td>4/10</td>
<td>4/10</td>
<td>0.16</td>
<td>5607.66</td>
</tr>
<tr>
<td>Imagen [45]</td>
<td>HomoSubs [48]</td>
<td>0/10</td>
<td>0/10</td>
<td>0.08</td>
<td>1103.53</td>
</tr>
<tr>
<td>Imagen [45]</td>
<td>TextFooler [24]</td>
<td>0/10</td>
<td>0/10</td>
<td>0.16</td>
<td>2231.10</td>
</tr>
<tr>
<td>Imagen [45]</td>
<td>RIATIG</td>
<td>10/10</td>
<td>10/10</td>
<td>0.23</td>
<td>704.09</td>
</tr>
</tbody>
</table>

Note: Sem. Dist. represents semantic distance.

Table 3. Attack performance using random mutation strategy.

<table>
<thead>
<tr>
<th>Model</th>
<th>R-1 ↑</th>
<th>R-3 ↑</th>
<th>Sem. Dist. ↓</th>
<th>PPL ↓</th>
</tr>
</thead>
<tbody>
<tr>
<td>DM-GAN</td>
<td>7/10</td>
<td>8/10</td>
<td>0.14</td>
<td>867.93</td>
</tr>
<tr>
<td>DF-GAN</td>
<td>1/10</td>
<td>2/10</td>
<td>0.12</td>
<td>882.80</td>
</tr>
<tr>
<td>DALL-E mini</td>
<td>0/10</td>
<td>0/10</td>
<td>0.17</td>
<td>2111.82</td>
</tr>
</tbody>
</table>

Table 4. The mean and variance of evaluation metrics under different selections of target images.

<table>
<thead>
<tr>
<th>Model</th>
<th>R-1 ↑</th>
<th>R-3 ↑</th>
<th>Sem. Dist. ↓</th>
<th>PPL ↓</th>
</tr>
</thead>
<tbody>
<tr>
<td>DF-GAN</td>
<td>9.2±(±0.42)/10 9.8±(±0.32)/10 0.31±0.04 596.33±297.02</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DALL-E mini</td>
<td>9.7±(±0.48)/10 9.9±(±0.32)/10 0.28±0.03 475.83±180.53</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

5.4. Ablation Study

Effectiveness of Mutation Strategy. To evaluate the effectiveness of our proposed mutation strategy, we conducted a comparative analysis with a random mutation approach, which employed insert, swap, and delete operations. Table 3 presents the results of our experiments, indicating that our mutations significantly outperformed random mutations, particularly in black-box settings.

Influence of Target Images. To further analyze the impact of target image selection on attack performance, we conducted ten experiments of the sample training process, with each experiment utilizing a different target image that
shared the same semantic meaning. The results, as presented in Table 4, indicate that our attacks demonstrated high robustness across varying target images.

Robustness of Example Generation. One important property of RIATIG is that the adversarial text can be reliably associated with specific visual concepts when used as a prompt. We evaluated the sample robustness of DALL-E mini, DALL-E 2, and Imagen. Specifically, we feed the same adversarial text to the target model ten times and calculated the average R-precision. The results show that RIATIG can achieve a 100% R-3.

Different Substitution Methods. In our mutation strategy, one of the most important parts is the semantic word substitution, which is essential for optimizing the fitness scores and improving the sample quality. To identify a suitable approach for substitution, we evaluate three different approaches, namely Glove models [38], WordNet models [35] and Word2vec models [34]. We evaluate two target generation models (DF-GAN [49] and DALL-E mini [43]) in both white-box and black-box settings. For all three approaches, we select the substitute word from the top 30 results. For each target model and substitution approach, we start with ten identical benign text and target images and generate ten adversarial samples accordingly to evaluate the performance. The results are shown in Table 5. It is seen that Word2vec performs better than Glove and WordNet with respect to R-precision and PPL for the attack against DF-GAN, and Word2vec performs better than the others with respect to all three performance metrics when attacking DALL-E mini. We hypothesize that it is probably caused by the larger semantic vocabulary of Word2vec.

Attack Performance with More Evaluations. To comprehensively evaluate the scalability of our attack, we trained 50 additional samples, each with distinct sources and targets. The results of this larger-scale evaluation are in Table 6, showing consistent performance of our attack strategy.

6. Discussion

Security Impacts. To prevent the malicious use of text-to-image generation models such as spreading misinformation (e.g., deepfake) [18], generating biased and inappropriate contents (e.g., stereotyping, pornography, violence) [10, 47], content moderation filters are deployed to filter harmful inputs. However, as we demonstrated, it is possible to craft a completely unrelated sentence to generate target images with black-box knowledge. Thus, it is important to consider the security risks of latest AI models.

Possible Defense. There are several possible defenses. First, a rule-based text filter could be deployed to remove redundant spaces and wrong words before generating images. To evaluate this defense, we tested the adversarial text using Grammarly and found that 20% of the samples could bypass the text filtering. For example, the sentence “a fruit stand display with bananas and starlit skies” was able to evade detection because this rule-based tool does not inspect the semantic meaning of the sentence. Second, one may consider using an image filter to prevent harmful images from being displayed to users. However, training such an image filter may require non-trivial efforts in collecting a large number of datasets. Moreover, the scope of harmful images can sometimes be difficult to define, such as with deep fake images. Another possible defense mechanism lies in the enhancement of the text-to-image generation models by adversarial training, which is one of the most effective defensive approaches against adversarial examples in the image and text domains [19, 24]. The model can be trained using both the original and adversarial text sharing the same associated images as the original text. However, it requires non-trivial efforts when facing a huge volume of adversarial examples as the modern text-to-image generation models are based on large-scale training datasets (e.g., Imagen [45] trains on about 460M image-text pairs).

7. Conclusion

In this paper, we propose an adversarial attack against the text-to-image generation models. We propose a two-stage attack framework to craft an effective adversarial example, where we find a workable adversarial example in the first stage, followed by a sample quality fine-tuning in the second stage. Through extensive evaluation, we can generate highly natural and semantically unrelated examples, allowing for successful attacks. We hope this work will raise awareness of potential security risks and aid the development of more effective defenses.

Acknowledgments This work was partially supported by the NSF (CNS-1916926, CNS-2083995, CNS-2154930, CNS-2229427, CNS-2238635), ARO (W911NF2010141).
References


