Stable Diffusion is Unstable

(Supplementary Material)

Anonymous Author(s)
Affiliation
Address
email

In this supplementary material, we first present a review of related works (Section A), including the diffusion model and studies about the vulnerabilities within text-to-image models. Following that, we delve into additional analyses concerning the vulnerabilities observed in the Stable Diffusion model (Section B). Subsequently, we offer instances of long and short prompt attacks, accompanied by the corresponding generated images, as well as instances of black-box attacks (Section C). Lastly, we undertake a comprehensive series of experiments to substantiate the effectiveness of our approach (Section D). These experiments include the evaluation of attacks targeting both long and short prompts. Additionally, ablation studies are conducted to explore attacks employing different search steps, assess the influence of our fluency and semantic similarity constraints on text similarity, and target diverse samplers (e.g., DDIM and DPM-Solver) in the attack process.

A Related Work

A.1 Diffusion Model.

Recently, the diffusion probabilistic model [20] and its variants [6, 13, 21, 18, 19] have achieved great success in content generation [21, 7, 19], including image generation [6, 21], conditional image generation [18], video generation [7, 24], 3D scenes synthesis [10] and so on. Specifically, DDPM [6] adds noises to images and learns to recover images from noises step by step. Then, DDIM [21] improves the generation speed of the diffusion model by skipping steps inference. Then, the conditional latent diffusion model [18] formulates the image generation in latent space guided by multiple conditions, such as texts, images, and semantic maps, further improving the inference speed and boarding the application of the diffusion model. Stable diffusion [18], a latent text-to-image diffusion model capable of generating photo-realistic images given any text input, and its enhanced versions [25, 8, 12], have been widely used in current AI-generated content products, such as Stability-AI [22], Midjourney [11], DALL·E2 [15], and Runaway [3]. However, these methods and products cannot always generate satisfactory results from the given prompt. Therefore, in this work, we aim to analyze the robustness of stable diffusion in the generation process.

A.2 Vulnerabilities in Text-to-image Models.

With the open-source of Stable Diffusion [18], text-to-image generation achieves great process and shows the unparalleled ability on generating diverse and creative images with the guidance of a text prompt. However, there are some vulnerabilities have been discovered in existing works [4, 1, 23]. Typically, StructureDiffusion [4] discovers that some attributes in the prompt are not assigned correctly in the generated images, thus they employ consistency trees or scene graphs to enhance the embedding learning of the prompt. In addition, Attend-and-Excite [1] also introduces that the Stable Diffusion model fails to generate one or more of the subjects from the input prompt and fails to correctly bind attributes to their corresponding subjects. These pieces of evidence demonstrate the vulnerabilities of the current Stable Diffusion model. However, to the best of our knowledge, no work has systematically analyzed the vulnerabilities of the Stable Diffusion model, which is the goal of this work.

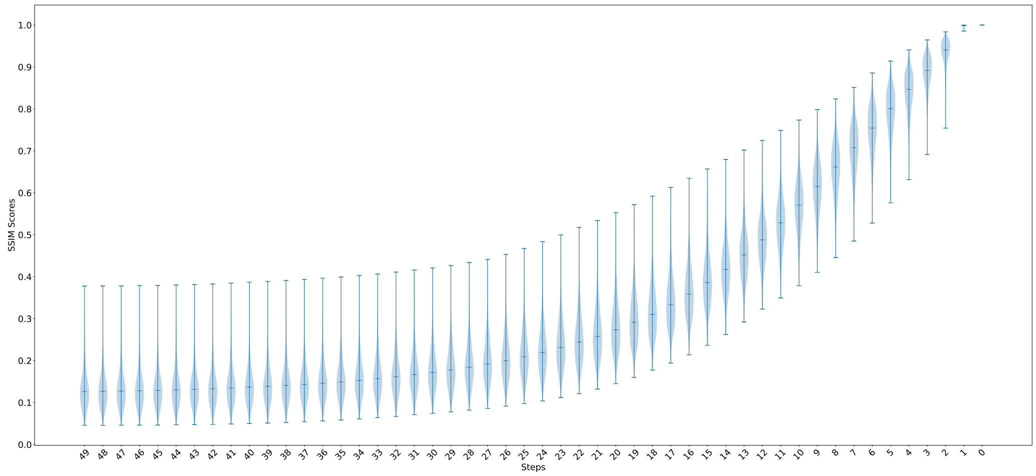

Figure B.1: A violin plot illustrating the generation speeds of $1,000$ images of various classes. The horizontal axis represents the number of steps taken, ranging from 49 to 0, while the vertical axis displays the SSIM scores. The width of each violin represents the number of samples that attained a specific range of SSIM scores at a given step.

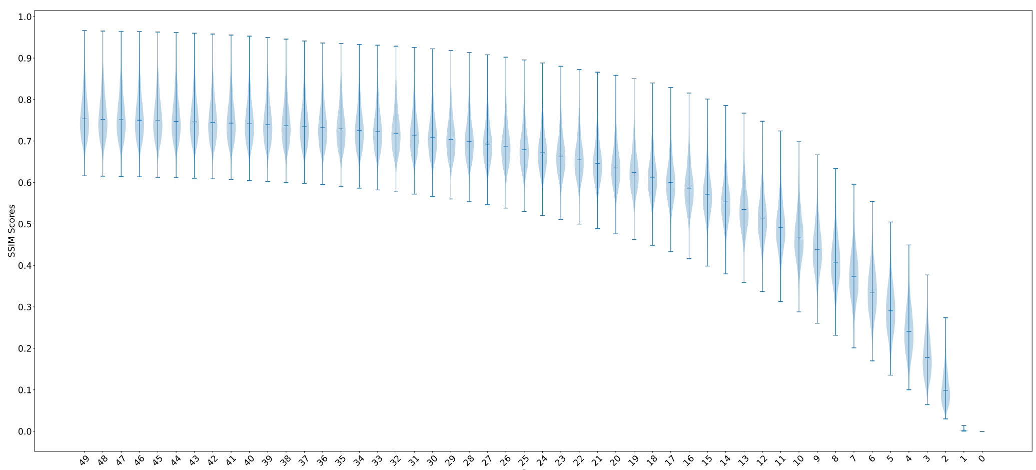

Figure B.2: A violin plot illustrating the generation speeds of $1,000$ images of various classes. The horizontal axis represents the number of steps taken, ranging from 49 to 0, while the vertical axis displays the LPIPS scores. The width of each violin represents the number of samples that attained a specific range of LPIPS scores at a given step.

B Vulnerabilities of Stable Diffusion Model

B.1 Pattern 1: Variability in Generation Speed

Fig. B.1 demonstrates the entire 50-step violin diagram which has been discussed before. To eliminate possible bias due to a single metric, we further verified the difference in generation speed of one thousand images based on the LPIPS [26] metric, as shown in Fig. B.2, The calculation of the LPIPS distance from the images generated at each stage to the ultimate image is performed. The horizontal axis signifies the range of steps from 49 down to 0, whereas the vertical axis denotes the respective LPIPS scores. Each violin plot illustrates the distribution of the LPIPS scores associated with 1,000 images at a specific step. The width of the violin plot is proportional to the frequency at which images achieve a certain score. During the initial stages of generation, the distribution's median is situated nearer to the maximum LPIPS value, suggesting a preponderance of classes demonstrates slower generation velocities. Nonetheless, the existence of a low minimum value indicates the presence of classes that generate at comparatively faster rates. As the generation transitions to the intermediate stages, the distribution's median progressively decreases, positioning itself between the maximum and

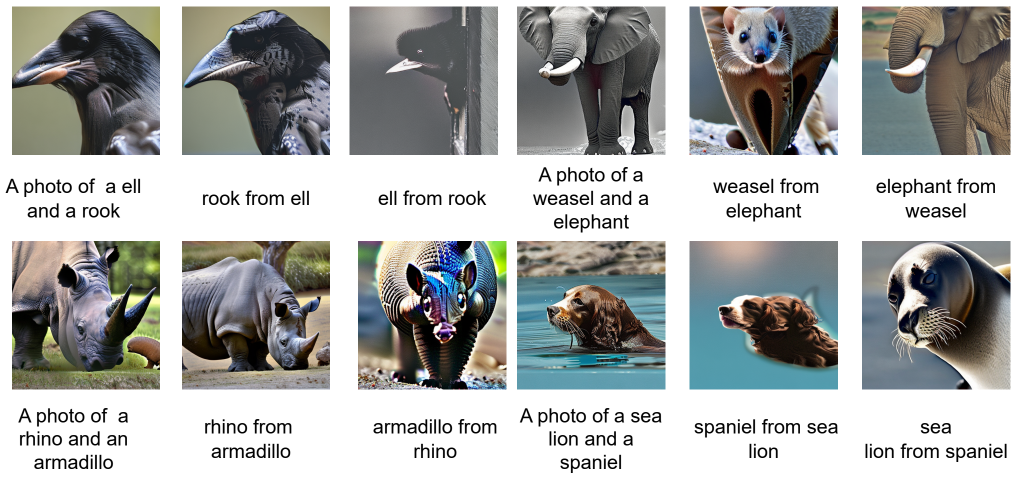

Figure B.3: The image caption, "A photo of class$A$ and class$B$" represents the generated image when feature entanglement occurs; And "class$A$ from class$B$" represents the final generated image of prompt "A photo of class$A$" based on the eighth step of the prompt "A photo of class$B$"

Figure B.4: The images in the first row are generated by the prompt "A photo of a warthog". The images in the second row are generated by the prompt "A photo of a traitor". The images in the third row are generated by the prompt "A photo of a warthog and a traitor".

minimum LPIPS values. In the concluding stages of generation, the distribution's median is found closer to the minimum LPIPS value, implying that the majority of classes are nearing completion. However, the sustained high maximum value suggests that there are classes still exhibiting slower generation rates.

B.2 Pattern 2: Similarity of Coarse-grained Characteristics

To further verify that coarse-grained feature similarity is the root cause of feature entanglement, we provide more cases in Fig.B.3. From these cases, we can see that for the two classes where feature entanglement can occur, they can both continue the image generation task based on each other's coarse-grained information.

B.3 Pattern 3: Polysemy of Words

As shown in Fig. B.4, when we attack the prompt "A photo of a warthog" to "A photo of a warthog and a traitor", the original animal warthog becomes an object similar to a military vehicle or military aircraft, while the images generated by attack prompt is not directly related to the image of the animal warthog or traitor. From the t-SNE visualization (Fig. B.5), we can see that the distance from the picture generated by the attack prompt to the text "a photo of a warthog" has a similar distance to the animal warthog picture to the text, so we can see that by attacking the original category word that guided the original category word (animal warthog) into its alternative meaning.

Figure B.5: t-SNE Visualization of 100 images each of "warthog", "traitor", "warthog and traitor“ and text "a photo of a warthog."

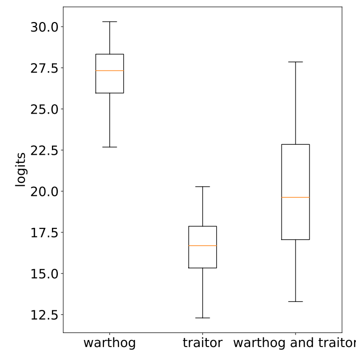

Figure B.6: The boxplot of cosine similarities between the text embedding of "a photo of a warthog" and 100 of image embeddings each of "warthog", "traitor", and "warthog" and traitor".

(a)

(b)

(c)

Figure B.7: a) "A type of footwear with a thick, rigid sole, often made of wood, and an upper made of leather or another material. Clogs can be open or closed, and are commonly associated with Dutch and Scandinavian cultures." b) "footwear" is replaced by "pistol". c) "Dutch" is replaced by "pistol".

(a) 0-1-2

(b) 0-2-1

(c) 1-0-2

(d) 1-2-0

(e) 2-0-1

(f) 2-1-0

Figure B.8: A template, "A photo of A, B and C", is used to generate prompts, where $A, B, C \in \{"\text{cat}", "\text{pistol}", "\text{clogs}"\}$. For exmaple, "0-1-2" represents $A = "\text{cat}"$, $B = "\text{pistol}"$ and $C = "\text{clogs}"$, and so on.

From the box plots (Fig. B.6), it can be observed that the image of "warthog" exhibits the highest similarity with the prompt's embedding, while the image of "traitor" demonstrates the lowest similarity, as anticipated. Simultaneously, the similarity distribution between the images of "warthog" and "traitor" with the prompt text is relatively wide, indicating that some images have a high similarity with "warthog," while others lack features associated with "warthog."

B.4 Pattern 4: Positioning of Words

In addition to the three aforementioned observations and patterns outlined in the paper, there is a fourth observations (Observation 4), which is related to positioning of words.

Observation 4. *When a text prompt contains a noun A representing the object to be generated, there exists a preceding word B and a succeeding word C around noun A. When replacing either word B or C with another noun D, for certain instances of noun A, replacing word B results in the generation of an image containing noun D, while replacing word C still results in the generation of an image containing noun A. Conversely, for other instances of noun A, the opposite scenario occurs.*

An example of Pattern 4 is shown in Fig. B.7. When "footwear" is replaced by "pistol", the generated image contains a pistol instead of clogs. However, when "Ductch" is replaced by "pistol", the model still generates an image of clogs. In addition to differences in the words being replaced, a significant distinction between the two aforementioned examples of success and failure lies in the relative positioning of the word being replaced with respect to the target class word. We hypothesize that this phenomenon occurs due to the different order of the replaced words B or C with respect to the noun A. To exclude the effects of complex contextual structures, a template for a short prompt, "A photo of A, B and C", is used, and the order of A, B, and C are swapped (Fig. B.8).

When these sentences with different sequences of category words are understood from a human perspective, they all have basically the same semantics: both describe a picture containing a cat, clogs, and a pistol. However, in the processing of language models (including CLIP), the order of words may affect their comprehension. Although positional encoding provides the model with the relative positions of words, the model may associate different orders with different semantics through learned patterns. Therefore, we propose our Pattern 4.

Pattern 4 (Positioning of Words)**.** *Let $\mathcal{V}$ denote a set of vocabulary. Let $\mathcal{N} \subset \mathcal{V}$ denote the subset of all nouns in the vocabulary. Consider a text prompt containing noun $A \in \mathcal{N}$ representing the object to be generated. Furthermore, assume there exist preceding word $B \in \mathcal{V}$ and succeeding word $C \in \mathcal{V}$ surrounding noun A. There exists a condition-dependent behavior regarding the replacement of words B and C with another noun $D \in \mathcal{N}$:*

$$\begin{cases} \exists A, D \in \mathcal{N}, \quad \exists B, C \in \mathcal{V}, \quad P(B \to D) \xRightarrow{\text{generate}} D \ \bigwedge \ P(C \to D) \xRightarrow{\text{generate}} A; \\ \exists A, D \in \mathcal{N}, \quad \exists B, C \in \mathcal{V}, \quad P(B \to D) \xRightarrow{\text{generate}} A \ \bigwedge \ P(C \to D) \xRightarrow{\text{generate}} D. \end{cases}$$

C Cases of Short/Long-Prompt Attacks and Black-box Attacks

C.1 Attack on Long Prompt

In Fig. C.1, we demonstrate more cases of long text prompt attacks.

C.2 Attack on Short Prompt

In Fig. C.2, we demonstrate more cases of long text prompt attacks.

C.3 Black-box Attack

In Fig. C.3, and Fig. C.4, we demonstrate black box attacks targeting mid-journey and DALL·E2, respectively.

D Experiments

In our experiments, we conduct comprehensive analyses of both long and short prompts. Furthermore, we conduct ablation studies specifically on long prompts, focusing on three key aspects. Firstly, we evaluate our attack method with different numbers of search steps T. Secondly, we investigate the influence of our constraints, including fluency and semantic similarity as measured by BERTScore. Lastly, we attack different samplers, including DDIM [21] and DPM-Solver [9].

D.1 Experimental Setting.

Attack hyperparamters. The number of search iterations T is set to 100. This value determines the number of iterations in the search stage, during which we aim to find the most effective attack prompts. The number of attack candidates N is set to 100. This parameter specifies the number of candidate attack prompts considered in the attack stage, allowing for a diverse range of potential attack prompts to be explored. The learning rate η for the matrix $\boldsymbol{\omega}$ is set to 0.3. The margin κ in the margin loss is set to 30.

Text prompts. Our experiments consider the $1,000$ classes from ImageNet-1K [2], which serves as the basis for generating images. To explore the impact of prompt length, we consider both short and long prompts. For clean short prompts, we employ a standardized template: "A photo of [CLASS_NAME]". Clean long prompts, on the other hand, are generated using ChatGPT 4 [16], with a prompt length restriction of 77 tokens to align with the upper limit of the CLIP [17] word embedder.

Evaluation metrics. To evaluate the effectiveness of our attack method, we generate attack prompts from the clean prompts. We focus on three key metrics: **success rate**, Fréchet inception distance [5]

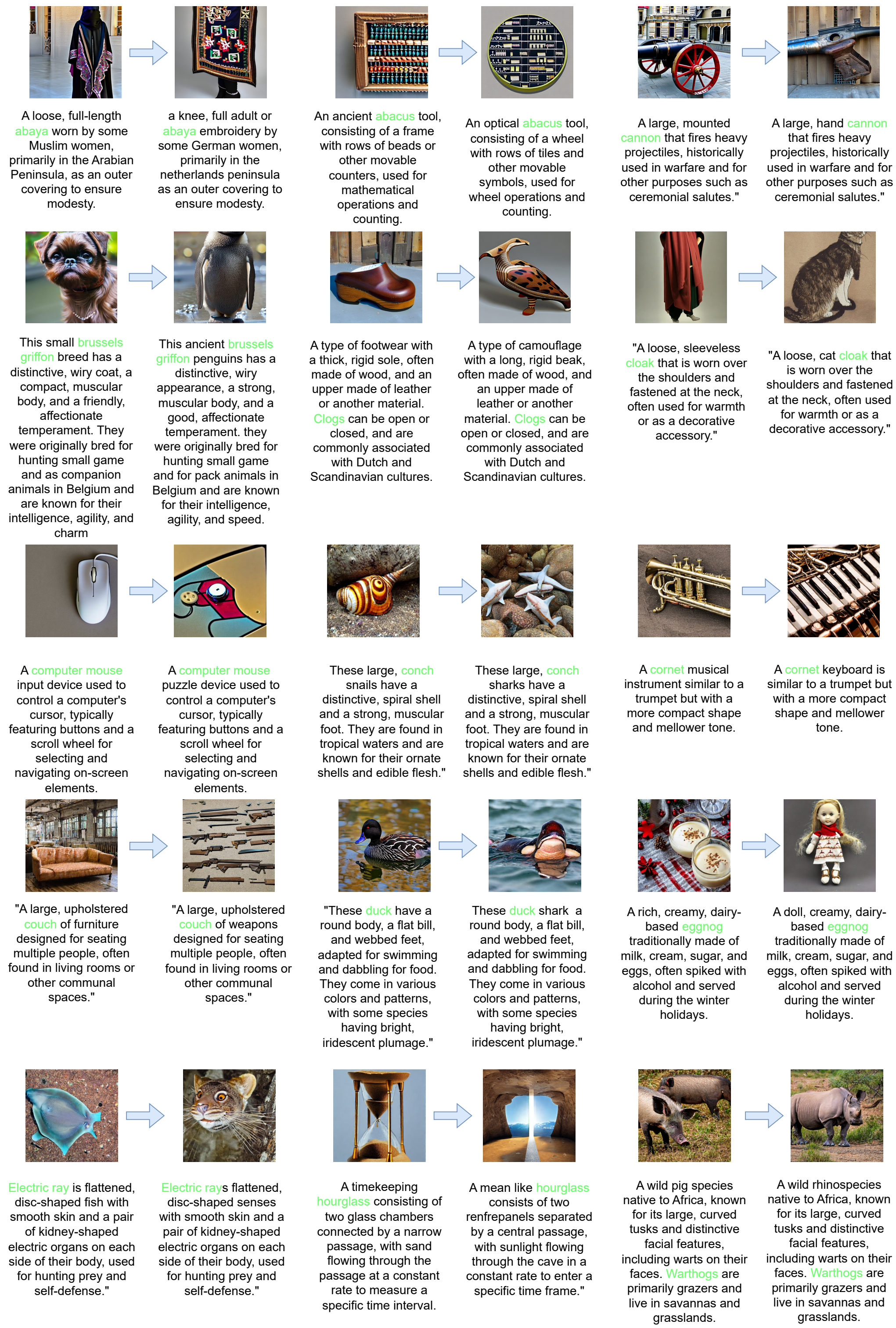

Figure C.1: To the left of the arrow is the clean long text prompt (highlighted by green) and its corresponding image, to the right of the arrow is the generated attack prompt (highlighted by red) and its corresponding image. (Section C.1 Attack on Long Text Prompt)

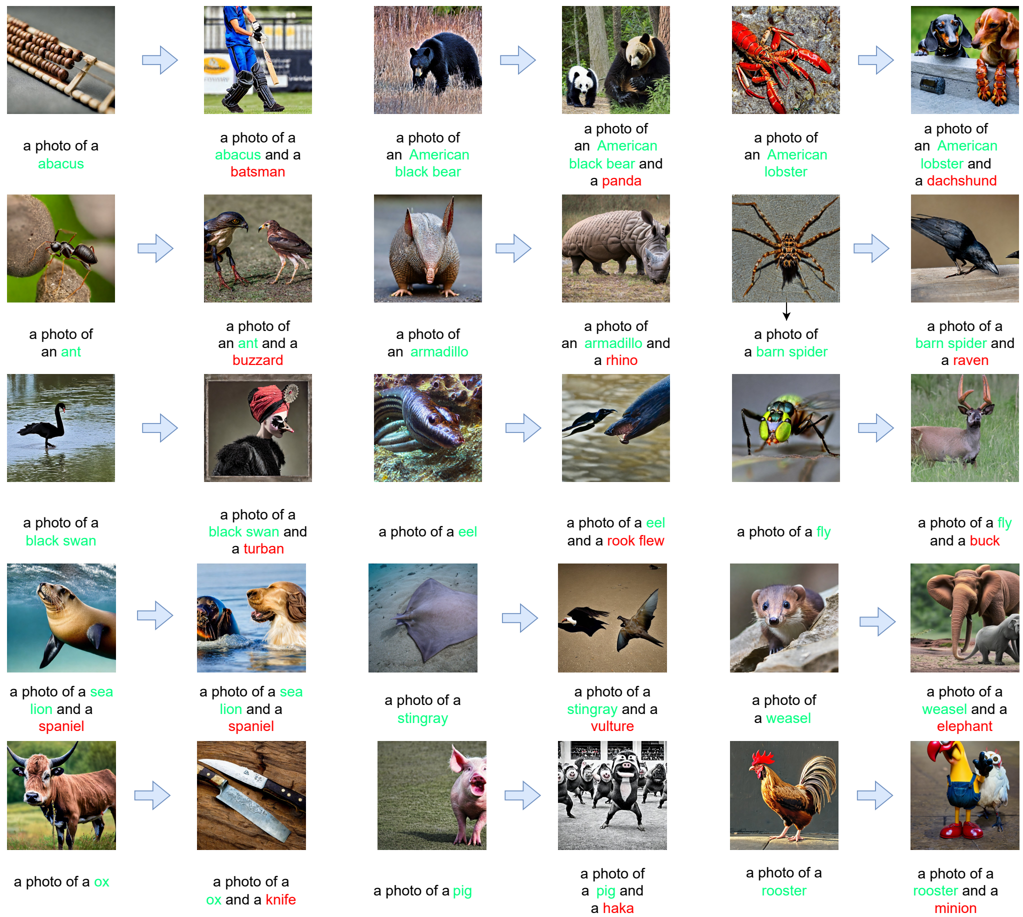

Figure C.2: To the left of the arrow is the clean short text prompt (highlighted by green) and its corresponding image, to the right of the arrow is the generated attack prompt (highlighted by red) and its corresponding image. (Section C.2 Attack on Short Text Prompt)

Table D.1: Main results of short-prompt and long-prompt attacks.

Prompt	Method	Success (%)	FID (↓)	IS (↑)	TS (↑)
Short	Clean	-	18.51	101.33±1.80	1.00
	Random	79.2	29.21	66.71±0.87	0.69
	ATM (Ours)	91.1	30.09	65.98±1.10	0.72
Long	Clean	-	17.95	103.59±1.68	1.00
	Random	41.4	24.16	91.33±1.58	0.94
	ATM (Ours)	81.2	29.65	66.09±1.83	0.84

(**FID**), Inception Score (**IS**), and text similarity (**TS**). Subsequently, $50,000$ images are generated using the attack prompts, ensuring a representative sample of 50 images per class. The success rate is determined by dividing the number of successful attacks by the total of 1,000 classes. FID and IS are computed by comparing the generated images to the ImageNet-1K validation set with (**torch-fidelity**)[14]. TS is calculated by embedding the attack prompts and clean prompts using the CLIP [17] word embedder, respectively. Subsequently, the cosine similarity between the embeddings is computed to quantify the text similarity.

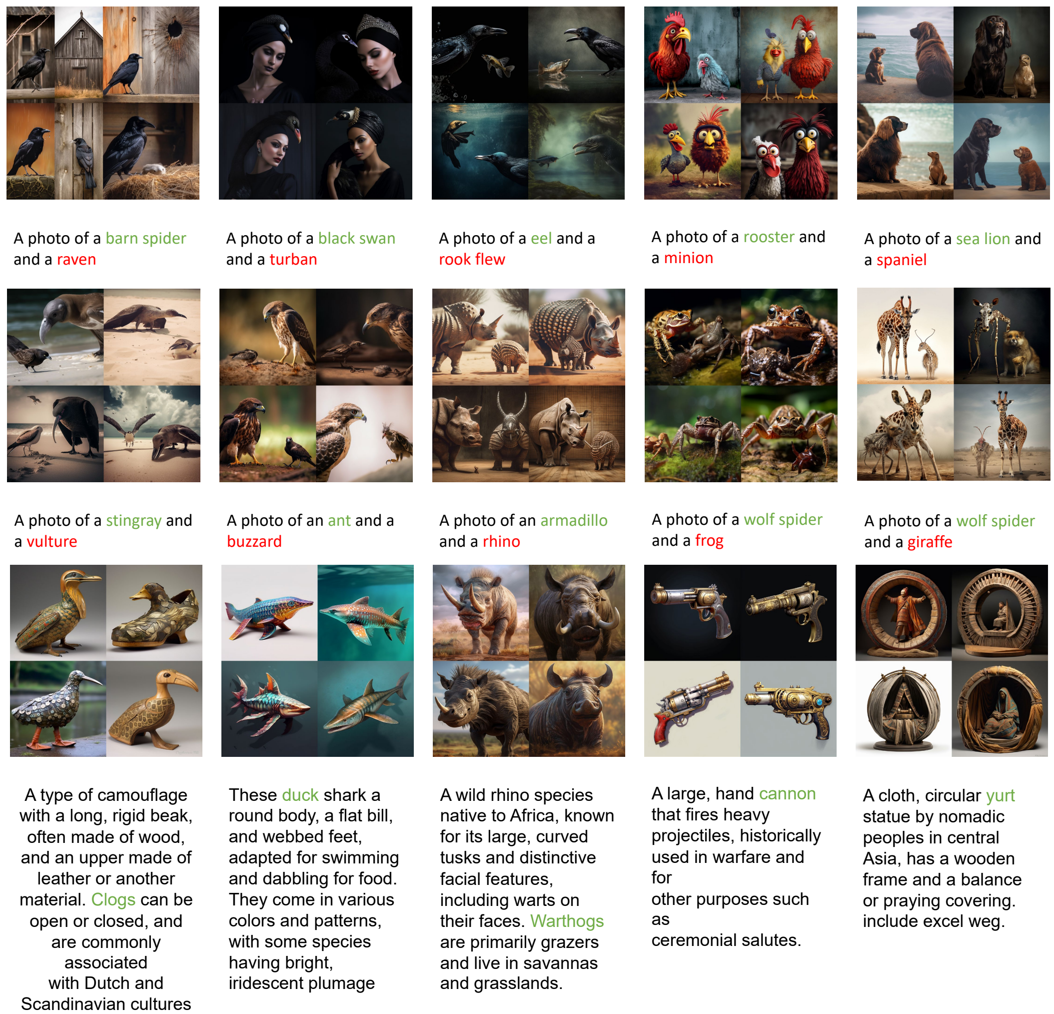

Figure C.3: Black-box attack on mid-journey (Section C.3 Black-box Attack).

D.2 Main Results

Table D.1 reports our main results, including short-prompt and long-prompt attacks. Compares to long text prompts, short text prompts comprise only a small number of tokens. This leads to a relatively fragile structure that is extremely vulnerable to slight disturbance. Therefore, random attacks can reach an impressive success rate of 79.2% targeting short prompts but a low success rate of 41.4% targeting the long prompts. In the contrast, our algorithm demonstrates its true potential, reaching an impressive success rate of 91.1% and 81.2% targeting short and long prompts, respectively.

As a further evidence of the effectiveness of our algorithm, it's worth noting the text similarity (TS) metrics between the random attacks and our algorithm's outputs. For short-prompt attack, the values stand at 0.69 and 0.72, respectively, illustrating that the semantic information of short texts, while easy to disrupt, can be manipulated by a well-designed algorithm with fluency and semantic similarity constraints. Our attacks preserve more similarity with the clean prompts. For long-prompt attacks, the TS score of random attacks (0.94) is higher compared to our attacks (0.84). One possible reason is that random attacks tend to make only minimal modifications as the length of the prompt increases. This limited modification can explain the significantly lower success rate of random attacks on longer prompts.

From the perspective of image generation quality and diversity, we found that as the attack success rate increases, image generation quality and diversity will decrease. For short and long texts, images generated from the clean text have the lowest FID (18.51 and 17.95) and the highest IS (101.33±1.80

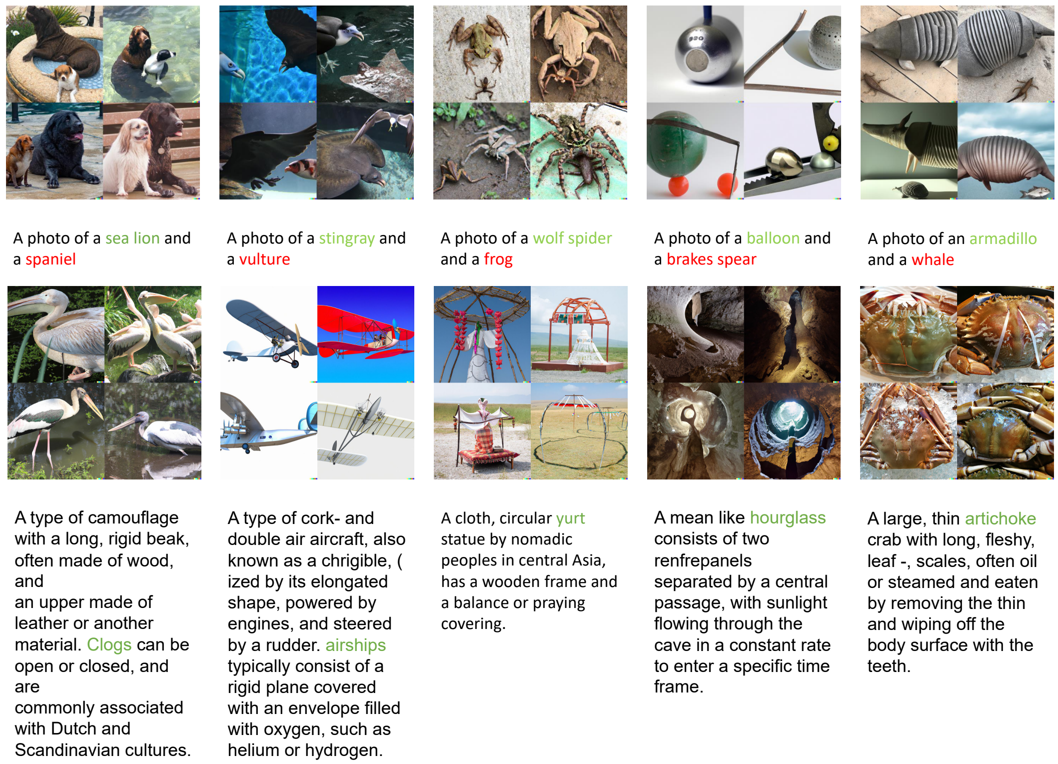

Figure C.4: Black-box attack on DALL·E2 (Section C.3 Black-box Attack).

Table D.2: Results of the ablation study on the number of steps in attack prompt search.

#Steps	**Success** (%)	**FID** (↓)	**IS** (↑)	**TS** (↑)
50	68.7	34.00	93.94±1.84	0.97
100	81.2	29.65	66.09±1.83	0.84
150	67.2	45.23	58.51±0.79	0.82

and 103.59±1.68). As the attack success rate rises, FID shows an upward trend. Examining this situation from the perspective of FID, a metric that gauges the distance between the distribution of generated images and the original data set. As the attack becomes more successful, the image set generated by the attack prompt tends to deviate substantially from the distribution of the original data set. This divergence consequently escalates the FID score, indicating a larger distance between the original and generated distributions. On the other hand, considering this situation from the diversity standpoint, it appears that the suppression of the generation of original categories brought on by the successful attack might instigate a decrease in diversity. This reduction in diversity, in turn, may cause a decrease in the Inception Score (IS).

D.3 Different Search Steps

Table D.2 presents the results of using different numbers of steps T in the search stage. For the $T = 50$ step configuration, the success rate is 68.7%. The FID value is 34.00, with lower values suggesting better image quality. The IS is reported as 93.94±1.84, with higher values indicating diverse and high-quality images. The TS value is 0.97, representing a high level of text similarity. Moving on to the $T = 100$ step configuration, the success rate increases to 81.2%, showing an improvement compared to the previous configuration. The FID value decreases to 29.65, indicating better image quality. The IS is reported as 66.09±1.83, showing a slight decrease compared to the previous configuration. The TS value is 0.84, suggesting a slight decrease in text similarity. In the $T = 150$ step configuration, the success rate decreases to 67.2%, slightly lower than the initial

Table D.3: Results of the ablation study on the constraints

Fluency	BERTScore	Success (%)	FID (↓)	IS (↑)	TS (↑)
✗	✗	91.3	39.14	47.21±1.25	0.37
✓	✗	81.7	29.37	64.93±1.57	0.79
✗	✓	89.8	39.93	46.94±0.99	0.51
✓	✓	81.2	29.65	66.09±1.83	0.84

Table D.4: Results of the ablation study on the samplers

Sampler	Success (%)	FID (↓)	IS (↑)	TS (↑)
DDIM [21]	81.2	29.65	66.09±1.83	0.84
DPM-Solver [9]	76.5	27.23	81.31±2.09	0.88

configuration. The FID value increases to 45.23, suggesting a decrease in image quality. The IS is reported as 58.51±0.79, indicating a decrease in the diversity and quality of generated images. The TS value remains relatively stable at 0.82.

When using $T = 50$, the attack prompt fails to fit well and exhibits a higher text similarity with the clean prompt. Although the generated images at this stage still maintain good quality and closely resemble those generated by the clean prompt, the success rate of the attack is very low. On the other hand, when $T = 150$, overfitting occurs, resulting in a decrease in text similarity and image quality due to the overfitted attack prompt. Consequently, the success rate of the attack also decreases. Overall, the configuration of $T = 100$ proves to be appropriate.

D.4 The Impact of Constraints

Table D.3 examines the impact of the fluency and semantic similarity (BERTScore) constraints. When no constraints are applied, the attack success rate is notably high at 91.3%. However, this absence of constraints results in a lower text similarity (TS) score of 0.37, indicating a decreased resemblance to clean text and a decrease in image quality. By introducing fluency constraints alone, the attack success rate decreases to 81.7% but increases the text similarity to 0.79. Furthermore, incorporating semantic similarity constraints independently also leads to a slight reduction in success rate to 89.8%, but only marginally improves the text similarity to 0.51. The introduction of constraints, particularly fluency constraints, leads to an increase in text similarity. The fluency constraint takes into account the preceding tokens of each token, enabling the integration of contextual information for better enhancement of text similarity. On the other hand, BERTScore considers a weighted sum, focusing more on the similarity between individual tokens without preserving the interrelation between context. In other words, the word order may undergo changes as a result and leads to a low text similarity. Certainly, this outcome was expected, as BERTScore itself prioritizes the semantic consistency between two prompts, while the order of context may not necessarily impact semantics. This further highlights the importance of employing both constraints simultaneously. When both constraints are utilized together, the text similarity is further enhanced to 0.84. Meanwhile, the success rate of the attack (81.2%) is comparable to that achieved when employing only the fluency constraint, while the text similarity surpasses that obtained through the independent usage of the two constraints.

D.5 Different Samplers

Table D.4 illustrates the effectiveness of our attack method in successfully targeting both DDIM and the stronger DPM-Solver. For the DDIM sampler, our attack method achieves a success rate of 81.2%, indicating its ability to generate successful attack prompts. Similarly, our attack method demonstrates promising results when applied to the DPM-Solver sampler. With a success rate of 76.5%, it effectively generates attack prompts. The TS scores of 0.84 and 0.88, respectively, indicate a reasonable level of text similarity between the attack prompts and clean prompts. These outcomes demonstrate the transferability of our attack method, showcasing its effectiveness against both DDIM and the more potent DPM-Solver sampler.

References

[1] H. Chefer, Y. Alaluf, Y. Vinker, L. Wolf, and D. Cohen-Or. Attend-and-excite: Attention-based semantic guidance for text-to-image diffusion models. *arXiv preprint arXiv:2301.13826*, 2023.

[2] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[3] P. Esser, J. Chiu, P. Atighehchian, J. Granskog, and A. Germanidis. Structure and content-guided video synthesis with diffusion models. *arXiv preprint arXiv:2302.03011*, 2023.

[4] W. Feng, X. He, T.-J. Fu, V. Jampani, A. Akula, P. Narayana, S. Basu, X. E. Wang, and W. Y. Wang. Training-free structured diffusion guidance for compositional text-to-image synthesis. *arXiv preprint arXiv:2212.05032*, 2022.

[5] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

[6] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

[7] J. Ho, T. Salimans, A. Gritsenko, W. Chan, M. Norouzi, and D. J. Fleet. Video diffusion models. *arXiv preprint arXiv:2204.03458*, 2022.

[8] L. Huang, D. Chen, Y. Liu, Y. Shen, D. Zhao, and J. Zhou. Composer: Creative and controllable image synthesis with composable conditions. *arXiv preprint arXiv:2302.09778*, 2023.

[9] C. Lu, Y. Zhou, F. Bao, J. Chen, C. Li, and J. Zhu. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. In *Advances in Neural Information Processing Systems*, 2022.

[10] G. Metzer, E. Richardson, O. Patashnik, R. Giryes, and D. Cohen-Or. Latent-nerf for shape-guided generation of 3d shapes and textures. *arXiv preprint arXiv:2211.07600*, 2022.

[11] Midjourney. Midjourney. https://www.midjourney.com/, 2023. Accessed on 2023-05-17.

[12] C. Mou, X. Wang, L. Xie, J. Zhang, Z. Qi, Y. Shan, and X. Qie. T2i-adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models. *arXiv preprint arXiv:2302.08453*, 2023.

[13] A. Nichol, P. Dhariwal, A. Ramesh, P. Shyam, P. Mishkin, B. McGrew, I. Sutskever, and M. Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*, 2021.

[14] A. Obukhov, M. Seitzer, P.-W. Wu, S. Zhydenko, J. Kyl, and E. Y.-J. Lin. High-fidelity performance metrics for generative models in pytorch, 2020. URL https://github.com/toshas/torch-fidelity. Version: 0.3.0, DOI: 10.5281/zenodo.4957738.

[15] OpenAI. Dall·e 2. https://openai.com/product/dall-e-2, 2023. Accessed on 2023-05-17.

[16] OpenAI. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[17] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[18] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022.

[19] L. Ruan, Y. Ma, H. Yang, H. He, B. Liu, J. Fu, N. J. Yuan, Q. Jin, and B. Guo. Mm-diffusion: Learning multi-modal diffusion models for joint audio and video generation. *arXiv preprint arXiv:2212.09478*, 2022.

[20] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pages 2256–2265. PMLR, 2015.

[21] J. Song, C. Meng, and S. Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021.

[22] Stability-AI. Stable diffusion public release. https://stability.ai/blog/stable-diffusion-public-release, 2023. Accessed on 2023-05-17.

[23] R. Tang, A. Pandey, Z. Jiang, G. Yang, K. Kumar, J. Lin, and F. Ture. What the daam: Interpreting stable diffusion using cross attention. *arXiv preprint arXiv:2210.04885*, 2022.

[24] S. Yin, C. Wu, H. Yang, J. Wang, X. Wang, M. Ni, Z. Yang, L. Li, S. Liu, F. Yang, et al. Nuwa-xl: Diffusion over diffusion for extremely long video generation. *arXiv preprint arXiv:2303.12346*, 2023.

[25] L. Zhang and M. Agrawala. Adding conditional control to text-to-image diffusion models. *arXiv preprint arXiv:2302.05543*, 2023.

[26] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018.