Report of Machine Learning Task

Qifei Liu

Introduction

Text-to-image generation models, such as DALL-E, Stable Diffusion, and MidJourney, have made significant progress in generating high-quality images from textual descriptions. However, aligning these models with human preferences remains a critical challenge. Reinforcement Learning from Human Feedback (RLHF) has emerged as a promising approach to address this issue. By training reward models on large-scale human feedback datasets, such as the Human Preference Dataset (HPD), these models can score generated images based on their alignment with human preferences.

Despite the success of RLHF, the robustness of reward models is not well understood. Specifically:

- Adversarial Robustness: Can small, carefully crafted perturbations to an image significantly alter its score in a reward model?
- Transferability of Attacks: Can adversarial perturbations optimized for one reward model (e.g., HPS v2) transfer to another (e.g., HPS v1)?
- Black-Box Attacks: Can reward models deployed as black-box services be fooled by query-based adversarial attacks?

Understanding these questions is crucial for ensuring the reliability and fairness of reward models in real-world applications. This task aims to explore these aspects by analyzing the robustness of reward models, particularly focusing on adversarial attacks and their implications.

Experiments and Results

(1) Adversarial Robustness

I opted to modify the `evaluate` function of the HPSv2 model[1] to implement our attack by introducing a mode-switching mechanism. To achieve this, I adapted an attack function from the `torchattacks` package[3], as the original implementation was designed for classification models rather than text-to-image generative models like ours. The key challenge was integrating both `text` and `image` inputs, particularly when calculating logits using the `get_logits` function. This required careful handling to ensure the text embeddings were properly aligned with the image features during the attack process.

After studying the mechanisms of PGD (Projected Gradient Descent) and FGSM (Fast Gradient Sign Method)[3], I realized that both methods rely on gradient computation, which necessitated disabling `torch.no_grad()`. The core idea behind these attacks is to compute the gradient of the loss with respect to the input image and then take a fixed step in the direction that maximizes the loss. While FGSM performs a single step, PGD iteratively applies this process multiple times, refining the perturbation to achieve a stronger attack. This iterative nature makes PGD more effective but also computationally more expensive.

Given that HPSv2 aims to align images with text based on their similarity, I first conducted experiments on benchmark data to test whether adversarial perturbations could mislead the model into selecting incorrect text-image pairs. Subsequently, I extended the experiments to labeled test data to evaluate the impact of the attack on the model's output scores and rankings. Specifically, I measured the rate at which scores decreased and observed how the rankings of images changed relative to their associated text prompts. This allowed me to assess the vulnerability differences across various models.

During the evaluation process, I encountered significant constraints due to limited CPU/GPU memory on my laptop, which forced me to reduce the `batch_size`. To mitigate this, I initially opted for a smaller model like GLIDE for preliminary testing. Additionally, due to computational limitations, I focused on the top three images rather than all nine available choices when analyzing rank changes. This compromise was necessary to ensure the experiments remained feasible within the available hardware constraints.

A notable observation was the effect of input normalization on the attack results. All input images underwent a transformation process, primarily normalization, which caused the perturbed images to appear darker compared to the originals. For example, the original image:

appeared as follows after the attack:

To restore the perturbed images to their original visual appearance, I applied an inverse transformation process by function `inverse_image_transform`. However, this initially resulted in a greyish background, as shown below:

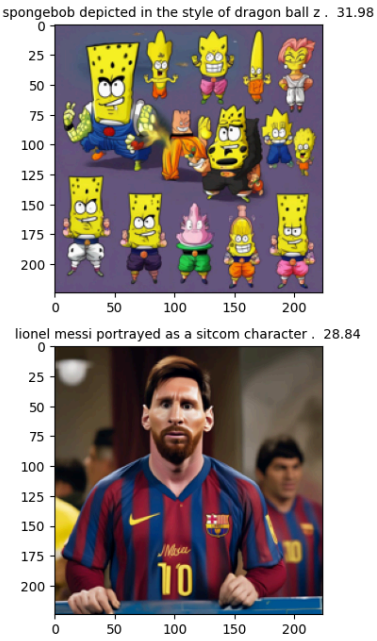
This issue was resolved by explicitly specifying the normalization parameters using `atk.set_normalization_used(mean=OPENAI_DATASET_MEAN, std=OPENAI_DATASET_STD)`, which ensured the correct transformation was applied during visualization. This adjustment not only improved the visual quality of the perturbed images but also highlighted the importance of proper normalization in adversarial attack pipelines.

In summary, this approach allowed me to successfully implement and evaluate adversarial attacks on the HPSv2 model, despite hardware limitations and the unique challenges posed by text-to-image generative tasks. The findings underscore the vulnerability of such models to adversarial perturbations and emphasize the need for robust defenses to ensure their reliability in real-world applications.

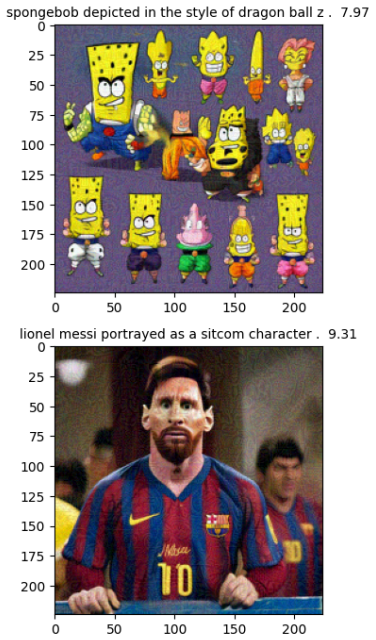
Results

The experimental results demonstrate that users can successfully add adversarial perturbations to manipulate the output scores of a reward model, effectively transforming a high-score image into a low-score one. For instance, in the case of the SDXL Base 0.9 model evaluated using HPSv2, a PGD attack significantly reduced the scores of two images: “spongebob depicted in the style of dragon ball” and “lionel messi portrayed as a sitcom character.” As shown below, the scores for these images decreased by 24 and 19 points, respectively, representing a reduction of approximately 2/3. Despite these substantial changes in scores, the visual differences between the original and perturbed images are minimal, making it nearly impossible for humans to distinguish between them. This indicates that the attack is both effective and imperceptible.

Before attack



After attack



Interestingly, the attack also caused the model to misalign text and image pairs. For example, the model incorrectly associated “spongebob” with “messi” because the perturbed “spongebob” image achieved a higher score for the “messi” text prompt. This misalignment is evident in the text-image score matrix below:

```
Before PGD attack:  tensor([[31.9844, 16.0772],
        [23.6135, 28.8419]])
After attack:  tensor([[ 7.9675, 13.6687],
        [30.8861,  9.3078]])
```

In contrast to PGD, the FGSM attack resulted in smaller score reductions due to its single-step nature, as opposed to the iterative process of PGD. The following result illustrates this difference:

```
Before FGSM attack:  tensor([[31.9844, 16.0772],
        [23.6135, 28.8419]])
After attack:  tensor([[27.2055, 14.8896],
        [23.4894, 25.3576]])
```

To further investigate the vulnerability of different models, I applied the PGD attack to the "spongebob" and "messi" images across several text-to-image generation models. The results are summarized in the tables below:

Spongebob Image:

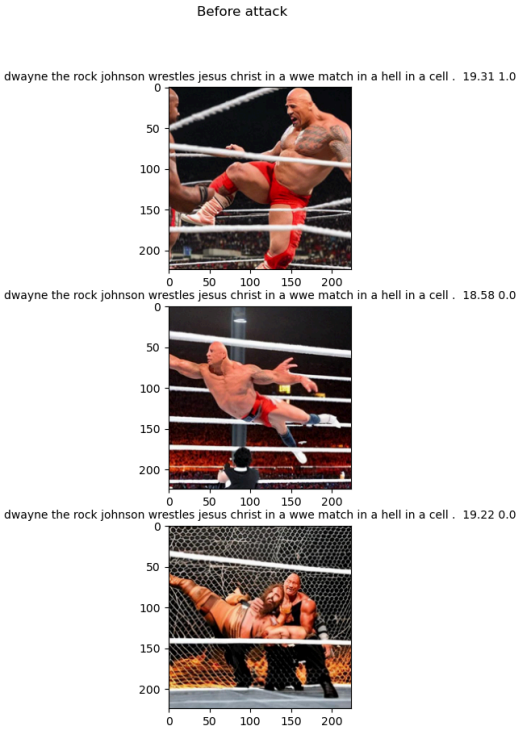
Model Name	Before	After	Decrease Percentage
ChilloutMix	36.69	19.58	46.6
CogView2	23.76	14.45	39.2
DALL·E mini	32.52	4.96	84.7
DALL·E 2	30.96	14.38	53.6
DeepFloyd-XL	30.60	11.29	63.1
VQ-Diffusion	17.41	9.16	47.4
Stable Diffusion 1.4	36.79	24.11	34.5

Messi Image:

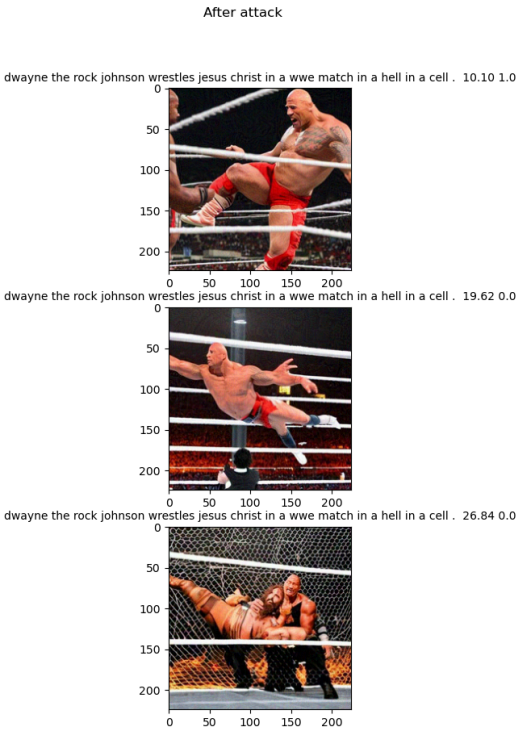
Model Name	Before	After	Decrease Percentage
ChilloutMix	25.87	11.55	55.4
CogView2	23.20	9.44	59.3
DALL·E mini	18.79	9.56	49.1
DALL·E 2	15.90	11.74	26.2
DeepFloyd-XL	28.74	12.33	57.1
VQ-Diffusion	11.98	8.30	30.7
Stable Diffusion 1.4	25.63	11.06	56.8

These results reveal significant variations in vulnerability across models, with some models experiencing score reductions of over 80%, while others show more modest decreases.

Additionally, I analyzed the impact of the attack on the ranking of images. For example, in the HPSv2 evaluation of the top three images from the test data, the attack caused the third-ranked image to become the highest-scoring one, leading to an incorrect prediction. This is illustrated below:

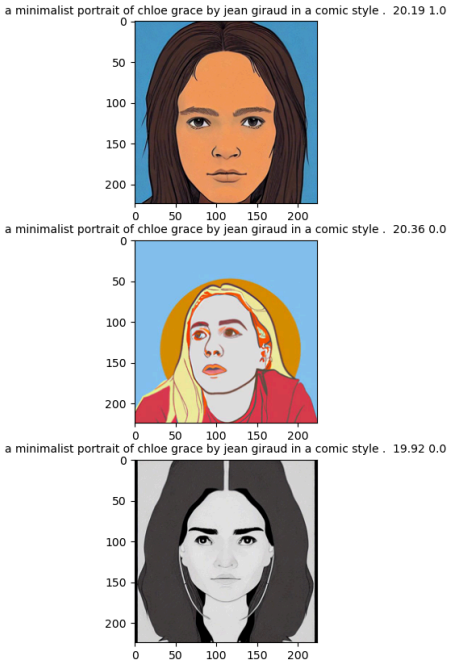


where `1.0` marks the highest score by human preference.

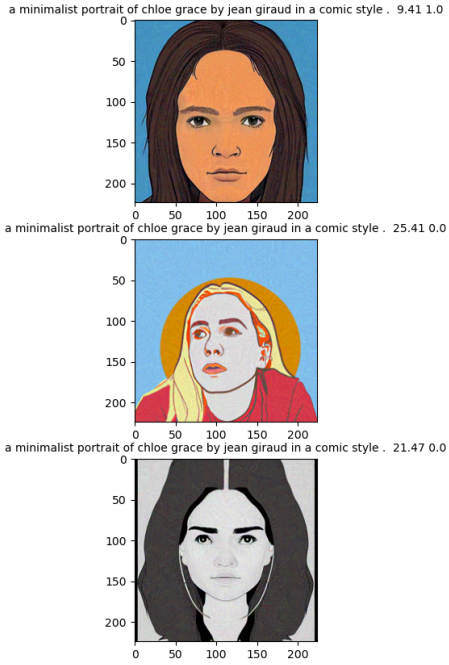


However, the attack does not always succeed. In some cases, it can correct previously incorrect predictions. For instance, in the example below, HPSv2 initially misranked the second image as the best but corrected its ranking after the attack:

Before attack



After attack



This suggests that the effectiveness of adversarial attacks is context-dependent and not universally guaranteed. Further research is needed to understand the conditions under which these attacks succeed or fail and to develop more robust defenses against such vulnerabilities.

(2) Transferability of Attacks

To explore the transferability of adversarial attacks from HPSv2 to other models, such as HPSv1[2], I conducted a series of experiments. Initially, I identified the optimized adversarial perturbation for HPSv2 and subsequently assessed its impact on HPSv1.

Results

After numerous trials, it was observed that the optimized perturbation effective against HPSv2 also had a significant effect on HPSv1. For instance, the optimal parameters for attacking an image of Lionel Messi were found to be `eps=15/255, alpha=3/225, steps = 10`. These parameters reduced the model's score from 23.20 to 9.44. When the same PGD attack function was applied to HPSv1, the score was reduced from 18.46 to 12.79, which is also near the best performance observed. This suggests that the adversarial samples exhibit strong transferability between the two models.
I hypothesize that the reason for this transferability is the similarity in the models' architectures, which stem from the same training methodology. This commonality likely allows adversarial perturbations to be effective across both models. To further validate this hypothesis, it would be beneficial to conduct a comparative analysis of the models' architectures and training data, as well as to explore the generalizability of these findings across a broader range of models and datasets.

(3) Black-Box Attacks

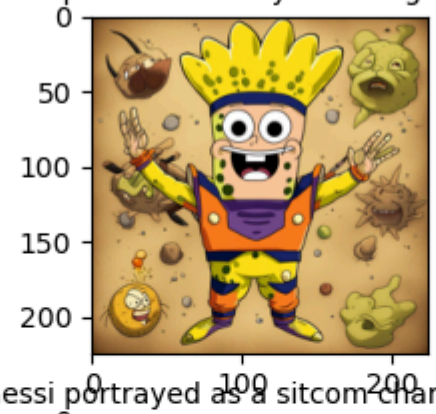
I have adapted the `PIFGSMPP` method from the `torchattacks` package, implementing a query-based black-box attack strategy as described in (1).

Results

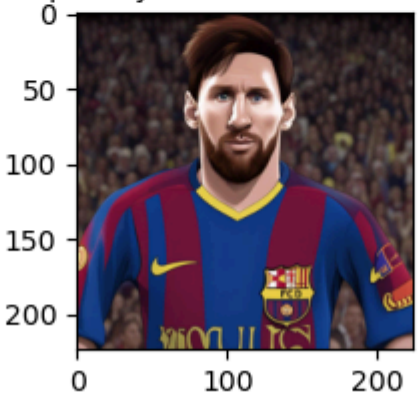
My findings indicate that the `PIFGSMPP` method remains effective in a black-box context, albeit with increased computational demands. This is likely due to the diminished efficacy of random perturbations and queries. To illustrate, consider the performance of the SDXL Refiner 0.9 model under this attack.

Before attack

spongebob depicted in the style of dragon ball z . 21.39

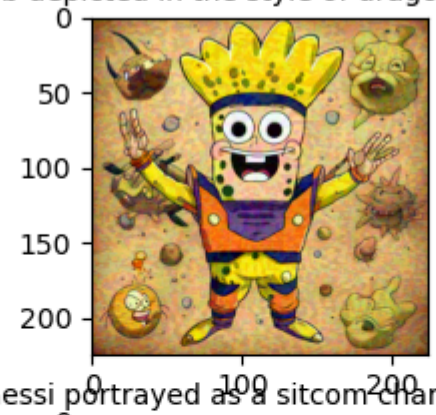


lionel messi portrayed as a sitcom character . 20.17

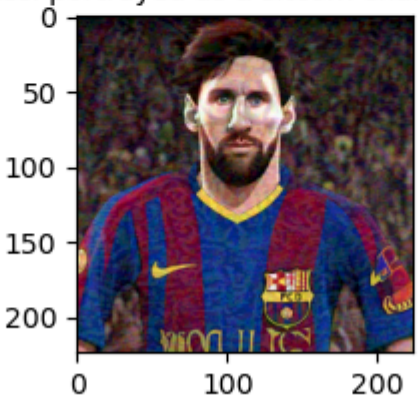


After attack

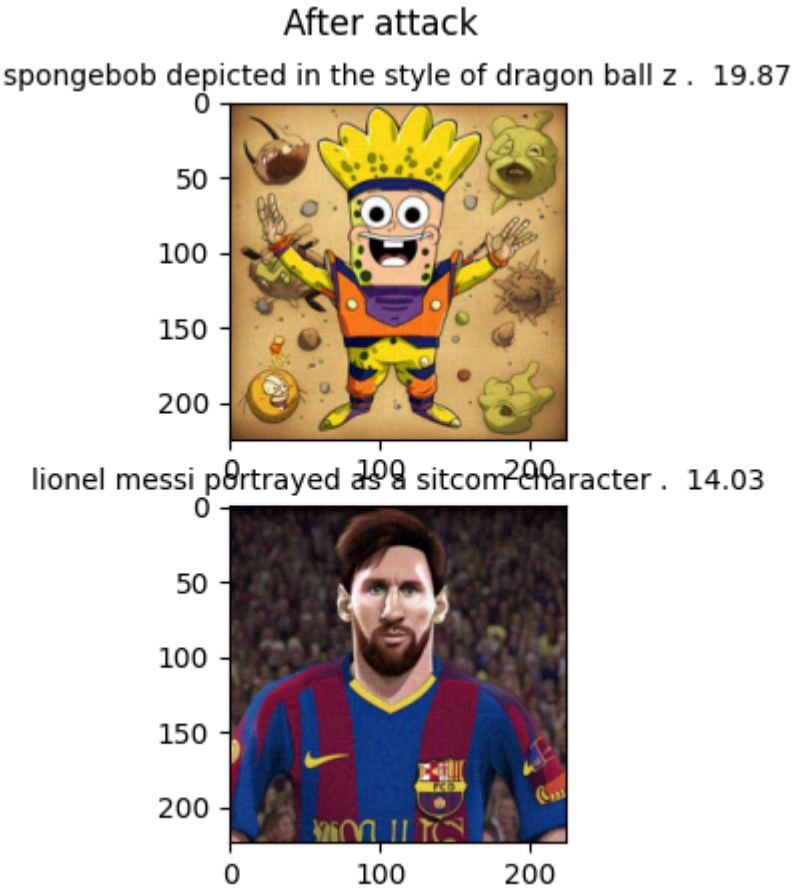
spongebob depicted in the style of dragon ball z . 20.48



lionel messi portrayed as a sitcom character . 18.22



Comparing these results with those obtained from a PGD attack:



The analysis reveals that black-box attacks are less potent than their white-box PGD counterparts. I surmise that this reduced effectiveness stems from the inherently lower efficacy of random perturbations and queries in a black-box setting, where no model information is available. Naturally, this scenario is less favorable than the white-box PGD attack, which benefits from full access to model details.

Discussion

- **Code Improvement and Parameter Encapsulation:** For future work, it's crucial to encapsulate attack parameters within the evaluate_benchmark function to streamline the attack process and make parameter adjustments and result assessments more straightforward. This oversight in the initial stages of the project led to considerable difficulties when investigating the transferability of attacks, highlighting the need for a more systematic approach to tool development and parameter management from the outset.
- **Model Vulnerability and Defense Strategies:** The experiments revealed that most models are significantly vulnerable to well-crafted attacks, underscoring the urgency for research into effective defense strategies. The vulnerability of different models varies under different adversarial images, necessitating further experiments to understand these differences and to develop tailored defense strategies.

- **Model Difference and Robustness:** Additional experiments are needed to explore how model size, structure, and training data affect robustness, which will help us better understand the models' inherent defense mechanisms.
- **Research on Attack Transferability:** The current research has been limited to HPSv2 and HPSv1 models. Future work should extend to other unrelated reward models to explore larger differences between models and how these differences affect the transferability of adversarial samples. Understanding these effects is crucial for developing more targeted defense strategies.
- **Diversity of Datasets:** To comprehensively assess the robustness of HPSv2, experiments across various datasets are required. Different datasets may reveal the model's performance under different conditions, which is essential for understanding the model's generalization capabilities and vulnerabilities.

In conclusion, while this analysis sheds light on the vulnerabilities of reward models in text-to-image generation, it also highlights the need for continued research and innovation to address these challenges. By improving the robustness of reward models and developing effective defense mechanisms, we can build more reliable and secure AI systems that better align with human preferences and values.

Reference

[1] Human Preference Score v2: A Solid Benchmark for Evaluating Human Preferences of Text-to-Image Synthesis (https://github.com/tgxs002/HPSv2)
[2] Human Preference Score: Better Aligning Text-to-Image Models with Human Preference (https://github.com/tgxs002/align_sd)
[3] torch_attacks (https://github.com/Harry24k/adversarial-attacks-pytorch)