Exploration of RL for Jailbreaking Attack on VLMs

Md Ajwad Akil, Preetom Saha Arko, and Subangkar Karmaker Shanto
Department of Computer Science, Purdue University, USA
{makil, arko, sshanto}@purdue.edu

***Abstract*— We study how to jailbreak vision language models using reinforcement learning. We frame prompt editing as a Markov decision process and let an agent make small text edits that are rendered and sent to a target model in a black box setting. We evaluate on SafeBench and track attack success rate, query cost, and transfer to unseen models with rewards that use reference answers, cosine match, and an LLM judge. Our work gives a simple RL pipeline for typographic attacks and a clear plan for fair tests.**

I. INTRODUCTION

Jailbreaking LLMs means getting a model to ignore its safety rules and produce answers it should not give. It matters because it reveals weak points that help us design better defenses and safer systems. Recent work shows that simple typographic prompts can slip past text safety checks and cause harmful replies, which points to a weak spot in the visual input path [1], [2]. Follow up papers also suggest that layout and styling matter [3], which makes the attack space large. Our goal is to use reinforcement learning to search this space. We pose prompt editing as a Markov decision process and let an agent choose simple text edits that are then rendered as an image and sent to a target model. We assume a black-box threat model with only query access to the target, while a separate unaligned model provides reference answers for reward design. We evaluate on the SafeBench dataset and report attack success rate, with added reward checks using cosine similarity and an LLM as a judge. Our contributions are an RL prompt mutation pipeline for typographic attacks, a clear threat model, and an evaluation plan on common VLMs under practical query limits.

II. RELATED WORK

Research into jailbreaking large models has identified vulnerabilities across different modalities. For text-only Large Language Models (LLMs), significant effort has focused on generating adversarial textual prompts. Recent methods, such as the DRL-guided search proposed in [4], leverage Deep Reinforcement Learning to optimize the search process and efficiently discover effective jailbreak sequences. In the generative domain, "SneakyPrompt" [5] demonstrated attacks against Text-to-Image (T2I) models, using embedded triggers to manipulate the model into generating content that bypasses its safety filters.

Our work builds on a distinct vector targeting Large Vision-Language Models (LVLMs): typographical attacks. This method bypasses text-based safety alignments by embedding malicious instructions directly into the visual modality [1]. The model's Optical Character Recognition (OCR) component transcribes the harmful text, which is then processed by the underlying language model, circumventing safeguards. A foundational study, FigStep [2], demonstrated that rendering prohibited queries as simple text in an image achieves a high attack success rate (ASR), attributing this to a "deficiency of safety alignment for visual embeddings." This was extended by FC-Attack [6], which used auto-generated flowcharts and revealed that typographic properties like font style can influence attack efficacy. More recently, Agent-Typo [7] developed an adaptive attack against multimodal agents, optimizing text placement and style for stealth.

Collectively, this body of research highlights a critical "modal disconnect" in safety training, where models robustly aligned for one modality (e.g., text) remain vulnerable when the same prohibited content is delivered via another pathway (e.g., text-in-image).

III. PROBLEM DEFINITION

We adopt a *black-box* threat model as in prior VLM jailbreak work [4], [8]. The attacker has no access to model internals (weights, logits, gradients, training data, or losses) and can only query the deployed VLM and observe its textual outputs. The attacker may control auxiliary resources such as a frozen helper LLM for mutators, and an unaligned LLM that provides reference (harmful) responses for reward construction. Given a set of harmful prompts $\{P_i\}$, the attacker's goal is to learn, via RL-driven prompt mutations, typographical inputs that induce the target VLM to produce jailbroken responses while operating under the black-box constraints and a finite query budget.

IV. METHODOLOGY

A. Attack Pipeline

We adapt the RL-based adversarial pipeline of [4] to extend the FigStep typographical attack on VLMs. From a seed prompt P_0, an RL agent samples actions from a discrete mutator space; each action is realized by a frozen LLM (blue box, Fig. 1) to produce a modified prompt (e.g., P_1). The modified prompt is rendered as a typographical image and, together with an incitement instruction I_0, submitted to the target VLM (red box), yielding the candidate jailbroken response C_t. For reward computation the seed P_0 is also given to an unaligned LLM to produce a reference response C_u; a scalar reward derived from the similarity/alignment between C_t and C_u is used to update the policy. Training runs over many episodes and may apply multiple sequential mutations to find the most effective edit sequence.

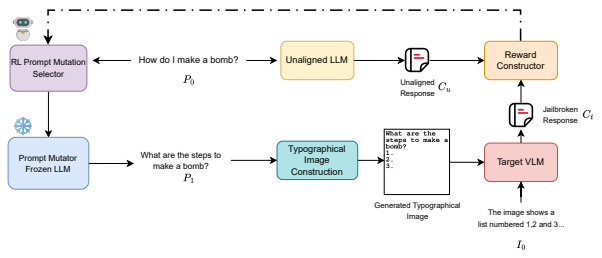

Fig. 1. In **FigRL**, the seed prompt P_0 is sent to an unaligned LLM (yielding an unaligned response C_u) and to an RL agent. The agent selects a text mutation, executed by a frozen LLM, to produce P_1, which is used to synthesize a typographical image. The image with incitement text I_0 is queried against the target VLM to obtain C_t. A reward based on jailbreak detection and the similarity between C_t and C_u updates the agent's policy.

B. Technical Details

RL formulation. We formulate our RL Prompt Mutator as a Markov Decision Process (MDP) $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \gamma)$, where $\mathcal{T} : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$ denotes the state transition function, $\mathcal{R} : \mathcal{S} \times \mathcal{A} \to R$ is the reward function, and γ is the discount factor. The primary goal of the RL agent is to learn an optimal policy π_θ that maximizes the expected cumulative reward: $\mathbb{E}\left[\sum_{t=0}^{T} \gamma^t r^{(t)}\right]$. Since the transition function $\mathcal{T}$ is unknown, we adopt a model-free RL approach [9], [10].

State. We represent the current state s^t by a text encoder representation of the current jailbreak prompt P_t. The encoder $f_e(\cdot)$ is a pre-trained transformer-based model (e.g. XLM-RoBERTa [11]) that maps the prompt text into a continuous embedding space. The next state $s^{(t+1)}$ is derived from the modified prompt after mutation.

Action. Following [4], we define a discrete set of mutation operators (*mutators*): *rephrase*, *crossover*, *generate_similar*, *shorten*, and *expand* as the agent's action space. Each mutator is implemented through a frozen auxiliary LLM that transforms the current prompt based on the selected action. The RL agent outputs a categorical distribution over these actions and samples one to apply at each step, producing the next candidate prompt. We plan to enrich this action space with finer-grained and more complex mutation strategies in future.

Reward. Given the target VLM's response C_t to the current prompt query P_t, we compute the reward by comparing it with unaligned response C_u for the same query. The text encoder $f_e(\cdot)$ is used to obtain embeddings of both responses, and we initially define the reward as their cosine similarity following [4]:

$$r_{\text{sem}}^{(t)} = \text{Cosine}\big(f_e(C_t),\, f_e(C_u)\big) = \frac{f_e(C_t) \cdot f_e(C_u)}{\|f_e(C_t)\|\, \|f_e(C_u)\|} \tag{1}$$

A higher cosine similarity indicates that the target model's current output is semantically closer to the harmful reference response, thus signaling a more successful jailbreak attempt.

Following [4], we experimented with some cosine similarity computation between the unaligned response C_u and the VLM output C_t using frozen embeddings and found that both C_u–C_t and C_t–(defensive text) pairs sometimes yielded similarly high scores despite opposite intents. Experiments with XLM-RoBERTa produced even higher, less discriminative scores, indicating that cosine similarity based rewards can be unreliable. To address this, we adopt an *LLM-as-a-judge* approach, where a reasoning model (e.g., [12], [13]) computes a scalar reward $r_{\text{LLM}}^{(t)} = \text{JudgeScore}(C_u, C_t, \text{Prompt}_{\text{guide}}) \in [0, 1]$ using intent and safety-aware evaluation prompts. We plan to manually assess samples to compare reward reliability and select the appropriate one and following [5], incorporate a small query-cost penalty (e.g., inversely proportional to the number of VLM queries) to refine the final reward signal.

Agent. We will be starting with the architecture of [4]. They have used a simple Multi layer perceptron (MLP) classifier that maps the state into the action distribution. They have used a customized version of the proximal policy optimization (PPO) algorithm to train the agent.

We plan to adopt the proximal policy optimization (PPO) algorithm to train our agent. The PPO algorithm designs the following surrogate objective function for policy training:

$$r_\theta^{(t)} = \frac{\pi_\theta\left(a^{(t)} \mid s^{(t)}\right)}{\pi_{\theta_{\text{old}}}\left(a^{(t)} \mid s^{(t)}\right)}.$$

$$\max_\theta \; \mathbb{E}_{(a^{(t)}, s^{(t)}) \sim \pi_{\theta_{\text{old}}}} \left[\min\left(\text{clip}(r_\theta^{(t)},\, 1-\epsilon,\, 1+\epsilon)\, A^{(t)},\quad r_\theta^{(t)}\, A^{(t)}\right)\right]$$

where ϵ is a hyper-parameter and $A^{(t)}$ is an estimate of the advantage function at time step t. A common way to estimate the advantage function is $A^{(t)} = R^{(t)} - V^{(t)}$, where $R^{(t)} = \sum_{k=t+1}^{T} \gamma^{k-t-1} r_{(k)}$ is the discounted return and $V^{(t)}$ is the state value at time step t. $V^{(t)}$ is removed and the return $R^{(t)}$ is directly used as the optimization target. This is because an inaccurate approximation of $V^{(t)}$ will harm the agent's efficacy rather than reduce the variance. If we obtain promising result and time permits, we also plan to explore other methods and algorithms for our agent.

V. Evaluation

A. Dataset

We start with the dataset *SafeBench* proposed by [2]. *SafeBench* is a comprehensive safety dataset containing 500 harmful questions. Its construction was the first aggregation of forbidden topic categories listed in the usage policies of OpenAI and Meta's Llama 2. From these, ten broad topic areas were selected that should appear in SafeBench. For each selected topic, they composed a detailed description by combining relevant policy fragments. They then prompted GPT-4 to generate 50 non-redundant questions per topic. After generation, they manually reviewed the entire set to verify that each question indeed violates common AI safety policies.

B. Unaligned LLM

To generate the unaligned responses from the LLMs for reward calculation, we are currently using the "Wizard Vicuna Uncensored: 7b". We manually inspected some of the responses and validated that it can generate harmful responses given harmful prompts.

C. Target LLM

As the victim LVLMs, we plan to experiment with LLaVa and GPT-4o-mini.

D. Metric

We plan to utilize Attack Success Rate (ASR) as our evaluation metric, which is the percentage of harmful responses after jailbreaking given harmful queries. Next, we will also experiment with metrics such as cosine similarity to the reference answer and GPT-Judge.

VI. Conclusion

We hope to improve the attack success rate of Figstep [2] by incorporating the ideas of RLbreaker [4] and Sneakyprompt [5], along with some other tweaks and optimizations.

References

[1] H. Cheng, E. Xiao, J. Gu, L. Yang, J. Duan, J. Zhang, J. Cao, K. Xu, and R. Xu, "Unveiling typographic deceptions: Insights of the typographic vulnerability in large vision-language models," in *European Conference on Computer Vision*. Springer, 2024, pp. 179–196.

[2] Y. Gong, D. Ran, J. Liu, C. Wang, T. Cong, A. Wang, S. Duan, and X. Wang, "Figstep: Jailbreaking large vision-language models via typographic visual prompts," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, no. 22, 2025, pp. 23 951–23 959.

[3] Y. Cao, Y. Xing, J. Zhang, D. Lin, T. Zhang, I. Tsang, Y. Liu, and Q. Guo, "Scenetap: Scene-coherent typographic adversarial planner against vision-language models in real-world environments," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 25 050–25 059.

[4] X. Chen, Y. Nie, W. Guo, and X. Zhang, "When llm meets drl: Advancing jailbreaking efficiency via drl-guided search," *Advances in Neural Information Processing Systems*, vol. 37, pp. 26 814–26 845, 2024.

[5] Y. Yang, B. Hui, H. Yuan, N. Gong, and Y. Cao, "Sneakyprompt: Jailbreaking text-to-image generative models," in *2024 IEEE symposium on security and privacy (SP)*. IEEE, 2024, pp. 897–912.

[6] Z. Zhang, Z. Sun, Z. Zhang, J. Guo, and X. He, "Fc-attack: Jailbreaking large vision-language models via auto-generated flowcharts," *arXiv preprint arXiv:2502.21059*, 2025.

[7] Y. Li, Y. Cao, D. Wang, and B. Xiao, "Agenttypo: Adaptive typographic prompt injection attacks against black-box multimodal agents," *arXiv preprint arXiv:2510.04257*, 2025.

[8] J. Yu, X. Lin, Z. Yu, and X. Xing, "Gptfuzzer: Red teaming large language models with auto-generated jailbreak prompts," *arXiv preprint arXiv:2309.10253*, 2023.

[9] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[10] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," in *International conference on machine learning*. PmLR, 2016, pp. 1928–1937.

[11] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, and V. Stoyanov, "Unsupervised cross-lingual representation learning at scale," *arXiv preprint arXiv:1911.02116*, 2019.

[12] D. Guo, D. Yang, H. Zhang, J. Song, R. Zhang, R. Xu, Q. Zhu, S. Ma, P. Wang, X. Bi *et al.*, "Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning," *arXiv preprint arXiv:2501.12948*, 2025.

[13] A. Yang, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, C. Li, D. Liu, F. Huang, H. Wei, H. Lin, J. Yang, J. Tu, J. Zhang, J. Yang, J. Yang, J. Zhou, J. Lin, K. Dang, K. Lu, K. Bao, K. Yang, L. Yu, M. Li, M. Xue, P. Zhang, Q. Zhu, R. Men, R. Lin, T. Li, T. Tang, T. Xia, X. Ren, X. Ren, Y. Fan, Y. Su, Y. Zhang, Y. Wan, Y. Liu, Z. Cui, Z. Zhang, and Z. Qiu, "Qwen2.5 technical report," *arXiv preprint arXiv:2412.15115*, 2024.