AI Security

Interesting Jailbreaking Attacks on LLMs and LMMs

Sangdon Park

POSTECH

April 21, 2026

Outline

Practical Use Cases of LLM/LMM Attacks

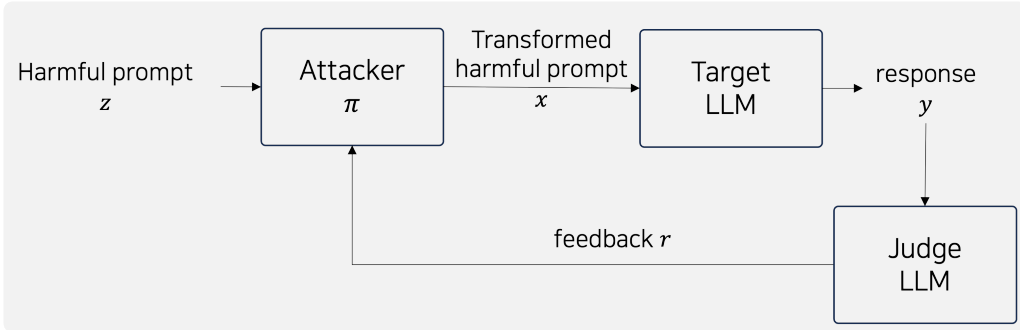
- We have learned LLM/LMM model archtectures and useufl optimization methods.
- Can we then actually attack LLMs/LMMs?
- We will see the four practical black-box attacks:
  1. LLM Target: TROJail – RL-based jailbreaking attacker
  2. T2I Target: Attacker with zeroth order optimization
  3. VLM Target: Add text in an image

Outline

Jailbreaking for LLMs

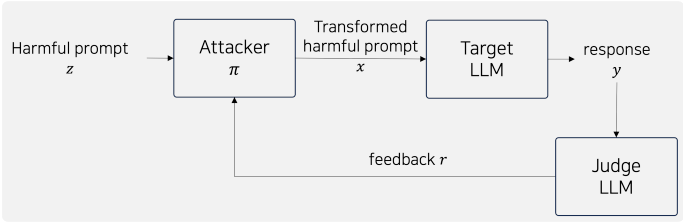
1. Manual prompting – training-free
   - DAN
   - CipherChat
   - ...
2. Automatic jailbreaking with LLMs – training-semi-free
   - AutoDan-Turbo
   - ...
3. Automatic jailbreaking with RL – trailing based
   - TROJail [Xiong et al., 2025]
   - ...

Again – Why RL for Jailbreaking?



- We often see only the victim model's responses – black box setups
- multi-turn interaction in collecting victim's responses to build a better strategy
- Finding a good attacker = learning an attacker via RL – very natural
- We will see TROJail [Xiong et al., 2025], a multi-turn GRPO based jailbreaking method.

Trajectory



Trajectory

$$\tau := ((s_1, a_1, r_1), (s_2, a_2, r_2), \ldots, (s_T, a_T, r_T))$$

- $a_t := x_t$
- $s_1 := z$
- $s_2 := (z, x_1, y_1)$
- $s_t := (z, x_1, y_1, \ldots, x_{t-1}, y_{t-1})$ if $t \geq 3$
- $r_t := \mathrm{LLM}_{\mathrm{judge}}(s_t, y_t) \approx \mathrm{LLM}_{\mathrm{judge}}(z, y_t)$

Single-Turn GRPO

Single-Turn GRPO Objective – simplified

$$J_{\text{GRPO}}(\pi) = \mathbb{E}_{s,\{x_g\}_{g=1}^{G}\sim\pi_{\text{old}}}\left[\frac{1}{G}\sum_{g=1}^{G}\frac{\pi(x_g \mid s)}{\pi_{\text{old}}(x_g \mid s)}\hat{A}_g\right], \quad \text{where} \quad \hat{A}_g := \frac{r_g - \texttt{mean}(r_{1:G})}{\texttt{std}(r_{1:G})}$$

- Note that for our educational purpose, I drop clipping and KD divergence terms.
- s: an initial harmful behavior/prompt
- π: an attacker
- Each group generates the following trajectories for $g \in \{1, \ldots, G\}$:

$$\tau_g := (s, x_g, r_g)$$

- The advantage $\hat{A}_g$ measures a relative goodness for the group's trajectory.
- The original GRPO assumes a single-turn setup – how to extend this for multi-turn?

Multi-Turn GRPO Variant 1 [Xiong et al., 2025]

Multi-Turn GRPO Variant 1 Objective – simplified

$$J_{\text{GRPO}}(\pi) = \mathbb{E}_{s_1, \{\tau_g\}_{g=1}^{G} \sim \pi_{\text{old}}} \left[\frac{1}{G} \sum_{g=1}^{G} \frac{1}{T} \sum_{t=1}^{T} \frac{\pi(x_{g,t} \mid s_t)}{\pi_{\text{old}}(x_{g,t} \mid s_t)} \hat{A}^{o}_{g,t} \right], \quad \text{where} \quad \hat{A}^{o}_{g,t} := \frac{r_{g,T} - \texttt{mean}(r_{1:G,T})}{\texttt{std}(r_{1:G,T})}$$

- s_1: an initial harmful behavior/prompt
- π: an attacker
- Each group generates the following trajectories for $g \in \{1, \ldots, G\}$ given s_1:

$$\tau_g := ((s_{g,1}, x_{g,1}, r_{g,1}), (s_{g,2}, x_{g,2}, r_{g,2}), \ldots, (s_{g,T}, x_{g,T}, r_{g,T})),$$

  where $s_{g,1} := s_1$.

- The advantage $\hat{A}^{o}_{g,t}$ measures a relative goodness for the group's trajectory w.r.t. the last reward.
  - Assumes a trajectory-level reward
  - Is the trajectory-level reward reasonable?

Multi-Turn GRPO Variant 2 [Xiong et al., 2025]

Multi-Turn GRPO Variant 2 Objective – simplified

$$J_{\text{GRPO}}(\pi) = \mathbb{E}_{s_1, \{\tau_g\}_{g=1}^{G} \sim \pi_{\text{old}}} \left[\frac{1}{G} \sum_{g=1}^{G} \frac{1}{T} \sum_{t=1}^{T} \frac{\pi(x_{g,t} \mid s_t)}{\pi_{\text{old}}(x_{g,t} \mid s_t)} \hat{A}_{g,t} \right],$$

where

$$\hat{A}_{g,t} := \frac{r_{g,T} - \texttt{mean}(r_{1:G,T})}{\texttt{std}(r_{1:G,T})} + \lambda \sum_{s=t}^{T} \frac{r'_{g,s} - \texttt{mean}(r'_{1:G,1:T})}{\texttt{std}(r'_{1:G,1:T})}$$

- Each group generates the following trajectories for $g \in \{1, \ldots, G\}$ given s_1:

  $$\tau_g := ((s_{g,1}, x_{g,1}, r_{g,1}, r'_{g,1}), (s_{g,2}, x_{g,2}, r_{g,2}, r'_{g,2}), \ldots, (s_{g,T}, x_{g,T}, r_{g,T}, r'_{g,T})),$$

  where $s_{g,1} := s_1$.

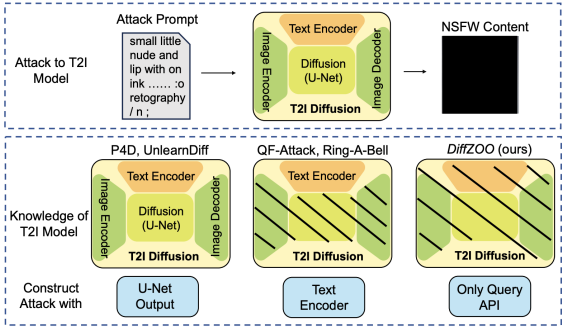
- The advantage $\hat{A}_{g,t}$ now considers the reward at each turn.
  - Why?
  - Is this satisfactory?

Outline

Jailbreaking for T2I Models

- T2I models, like diffusion models, are trained over web data sets.
- The unfiltered data sets may contain harmful images.
- Even with alignment (*e.g.*, RLHF), T2I models may generate harmful images.
- How to find a text prompt to generate harmful images?

DiffZOO – A Black-box Jailbreaking for T2I Models [Dang et al., 2025]



- A fully black-box attack
- Leverages zeroth-order optimization
  - What is the zeroth-order optimization?

DiffZOO Outline

1. Goal: finding a hard prompt to generate a harmful image, *i.e.*,

Goal

$$\min_{\hat{p} \in \mathbb{N}^L} \| f_{\text{toxicity}}(\texttt{T2I}(\hat{p})) - \mathbf{e} \|_2,$$

where L is the length of the prompt and $\mathbf{e}$ is a target harmful label.

2. How to find $\hat{p}$? Initialize $\hat{p}$ and then choose the index of a word to be changed.
   - Continuous position replacement vectors $\mathbf{z} \in [0, 1]^L$
   - Discrete position replacement vectors $\bar{\mathbf{z}} \in \{0, 1\}^L$
3. Optimize $\mathbf{z}$ via Zeroth Order Optimization (ZOO)

Continuous Position Replacement Vectors (C-PRV)

C-PRV

For an initial prompt $\mathbf{p} \in \mathbb{N}^L$, there are C-PRV $(\mathbf{z}, \mathbf{u}_1, \ldots, \mathbf{u}_L)$, where

$$\mathbf{z} \coloneqq \begin{bmatrix} z_1 & z_2 & \ldots & z_L \end{bmatrix} \in [0,1]^L \quad \text{and} \quad \mathbf{u}_i \coloneqq \begin{bmatrix} u_{i,1} & u_{i,2} & \ldots & u_{i,L} \end{bmatrix} \in [0,1]^M \text{ for each } z_i.$$

- $\mathbf{p}$: "generate a harmful image"
- z_i: a probability for the i-th word to change
  - $z_i = 0.9$ where the i-th word is "harmful"
- $\frac{\mathbf{u}_i}{\|\mathbf{u}_i\|_1}$: a probability distribution over candidate tokens for z_i
  - a distribution over "violent", "bloody", and "red"
- $\mathbf{z}$ and $\mathbf{u}_i$ are learning parameters.
- But, how can we generate a hard prompt?

Discrete Position Replacement Vectors (D-PRV)

D-PRV

Given $(\mathbf{z}, \mathbf{u}_1, \ldots, \mathbf{u}_L)$, there are D-PRV $(\bar{\mathbf{z}}, \bar{\mathbf{u}}_1, \ldots, \bar{\mathbf{u}}_L)$, where

$$\bar{\mathbf{z}} \coloneqq \begin{bmatrix} \bar{z}_1 & \bar{z}_2 & \ldots & \bar{z}_L \end{bmatrix} \in \{0,1\}^L \quad \text{and} \quad \bar{\mathbf{u}}_i \coloneqq \begin{bmatrix} \bar{u}_{i,1} & \bar{u}_{i,2} & \ldots & \bar{u}_{i,L} \end{bmatrix} \in \{0,1\}^M \text{ for each } \bar{z}_i.$$

Here, $\bar{z}_i$ and $\bar{u}_{i,j}$ are decided via the following sampling procedure:

$$\bar{z}_i \sim \texttt{Bernoulli}(z_i) \quad \text{and} \quad \bar{u}_{i,j} = 1 \text{ where } j \sim \textit{Multinomial}\left(\frac{\mathbf{u}_i}{\|\mathbf{u}_i\|_1}\right)$$

- $\bar{z}_i = 1$: change the i-th word
- $\bar{u}_{i,j}$: use the j-th candidate word for replacing the i-th word in the prompt.
- From C-PRV, D-PRV are decided.
- Thus, our forward pass consists of the following processes:

$$\text{C-PRV} \longrightarrow \text{D-PRV} \longrightarrow \text{construct } \hat{p}$$

Zeroth Order Optimization

ZOO

Estimate gradients via ZOO:

$$\nabla_{z_i}\mathcal{L} \approx \frac{1}{K}\sum_{k=1}^{K}\left[\frac{\mathcal{L}(z_i+\delta_k)-\mathcal{L}(z_i-\delta_k)}{2\delta_k}\right] \quad \text{and} \quad \nabla_{u_{i,j}}\mathcal{L} \approx \frac{1}{K}\sum_{k=1}^{K}\left[\frac{\mathcal{L}(u_{i,j}+\delta_k)-\mathcal{L}(u_{i,j}-\delta_k)}{2\delta_k}\right],$$

where $\delta_k \sim \mathcal{N}(0,\sigma^2)$.

- Recall our objective:

$$\min_{\hat{p}\in\mathbb{N}^L} \underbrace{\|f_{\text{toxicity}}(\texttt{T2I}(\hat{p}))-\mathbf{e}\|_2}_{\mathcal{L}},$$

- Is it really possible even with sampling procedures?
- What's the limitation of ZOO in general and does this limitation still hold in this case?

Outline

Jailbreaking for VLMs

- VLMs introduce a new attack surface via images.
- How to leverage this?

SceneTAP [Cao et al., 2025]

- An attack is relatively easy.
- Why does this happen?

Outline

Conclusion

- We have learned the practical black-box attacks to LLMs and LMMs.
  - RL-based jailbreaking
  - Jailbreaking via zeroth-order optimization
  - Jailbreaking via image “blending/editing”
- Try these for your HW2!

Reference I

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*, pages 25050–25059, 2025.

P. Dang, X. Hu, D. Li, R. Zhang, Q. Guo, and K. Xu. Diffzoo: A purely query-based black-box attack for red-teaming text-to-image generative model via zeroth order optimization. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 17–31, 2025.

X. Xiong, O. Li, Z. Liu, M. Li, W. Shi, F. Zhu, Q. Wang, and F. Feng. Trojail: Trajectory-level optimization for multi-turn large language model jailbreaks with process rewards. *arXiv preprint arXiv:2512.07761*, 2025.