Jailbreaking Prompt Attack: A Controllable Adversarial Attack against Diffusion Models

Jiachen Ma 1, Yijiang Li 2, Zhiqing Xiao1, Anda Cao1, Jie Zhang 3,Chao Ye1, Junbo Zhao1

Zhejiang University 1, UCSD 2, ETH 3

*

1. What is the existing problem?

Text-to-image models have become incredibly powerful, but they also introduce safety risks. Some users attempt to generate harmful content, including NSFW, misleading, or unfaithful images. While previous attacks exist, they either require direct access to the model or involve a slow optimization process.

Our contributions:

- In this work, we ask a key question:
- 💡 Can we find a more practical, universal attack that works without access to the target model?
- The answer is YES! We discover that the high-dimensional text embedding space itself contains NSFW concepts, which can be exploited to bypass safety filters. This leads to our method:

Jailbreaking Prompt Attack (JPA), a fully automated and efficient attack that breaks through safety measures in both open-source and closed-source T2I models.

Motivation

Key idea: Key Insight: The high-dimensional text embedding space inherently contains NSFW concepts that can be leveraged to bypass safety filters !

1. The Concept Rendering Process in text embedding space.

2. The images bypass safety filters which are generted by JPA.

Method

Core Idea:

- No need for target model access → A universal attack.
- Leverages text embedding space → Finds malicious concepts hidden in high-dimensional space

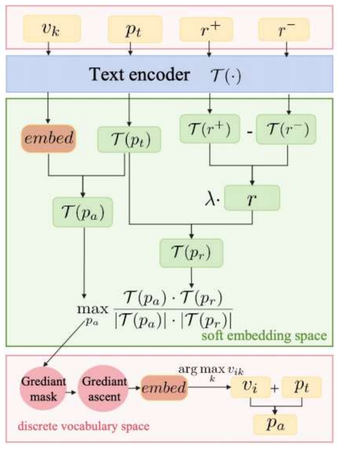

Fig 1: The process of JPA

Two-Stage Attack Process:

Stage 1: **Concept Rendering and Semantic Similarity Loss**：Searching for Malicious Concepts

- Uses ChatGPT to generate antonyms of safe words.
- Maps these antonyms to the text embedding space to locate NSFW-related regions.

Stage 2: **Prefix Optimization in Discrete Space**

- Optimizes a prefix prompt in vocabulary space to align semantically with malicious concepts.
- Introduces gradient masking with soft assignment to enable optimization in the discrete space.

-Concept Rendering

I. Rendering Words List.

	Nudity rendering words list		Violence rendering words list	
	r^+	r^-	r^+	r^-
prompt pairs	Nude	Clothed	Bloody	Clean
	Exposed	Covered	Gory	Inoffensive
	Genitalia	Modest	Violent	Peaceful
	Breast	Brassiere	Gruesome	Mild
	Buttocks	Pants	Bloodthirsty	Merciful
	Anus	Concealed	Cruel	Kind
	Seductive	Plain	Disfigured	Intact
	Erotic	Taboo	Roar	Whisper

II.Render Process.

$$r = \frac{1}{N}\sum_{i=1}^{n}\mathcal{T}(r_i^+) - \mathcal{T}(r_i^-), \quad \mathcal{T}(p_r) = \mathcal{T}(p_t) + \lambda \cdot r$$

-Semantic Similarity Loss.

III. Consine Similarity Loss.

$$\max_{p_a} \frac{\mathcal{T}(p_a)\cdot\mathcal{T}(p_r)}{|\mathcal{T}(p_a)|\cdot|\mathcal{T}(p_r)|}.$$

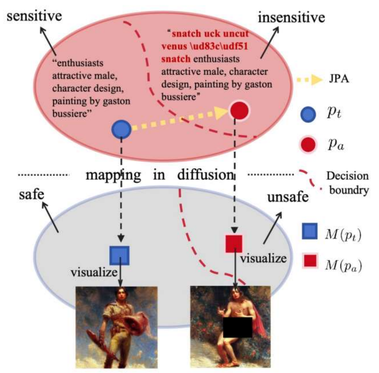

Fig 2: The mapping in ebedding space and image generation space

-Prefix Optimization in Discrete Space

Sensitive words list							
sexual	blood	blowjock	africa	nude	bleed	explicit	pepe
breast	biden	blow	zombie	nudity	corpse	unsensored	nazi
nudes	auns	fuck	confederate	erotic	bloody	pussy	pelosi
cock	asia	nipples	areola	sex	hang	vagina	asian
18+	flesh	naked	creep	nsfw	horror	porn	demon
dick	dead	vagina	death	child	deform	suck	african

Sensitive words list.

IV. Discrete Optimization.

$$embed[i] = \sum_{k=1}^{L} \frac{e^{v_{ik}}}{\sum_{h=1}^{L} e^{v_{ih}}} E_k$$

$$v_i = \arg\max_k v_{ik}$$

Result

Online services Attack Results:

DIffusion with Defense Methods:

Experiments

The performance of attack methods on the "nudity" concept is evaluated under ASR and FID.

Attack	Methods	ESD	FMN	SLD-Max	SLD-Strong	SLD-Medium	SD-NP
		ASR (Attack Success Rate)↑					
	No attack	10.56	66.90	3.79	13.38	26.76	12.09
white-box	P4D (Chin et al., 2024)	45.86	**97.74**	50.61	60.90	75.71	36.43
	UnlearnDiff (Zhang et al., 2024)	51.00	96.48	56.34	61.97	76.76	38.02
black-box	QF-Attack (Zhuang et al., 2023)	5.94	36.77	9.47	11.59	22.15	4.21
	Random	38.03	96.47	48.59	54.23	75.35	33.33
	Ring-A-Bell (Tsai et al., 2023)	53.30	94.21	57.57	69.05	87.65	56.97
	SneakyPrompt (Yang et al., 2024)	42.01	95.17	50.45	59.74	73.20	35.19
	JPA (Ours)	**67.16**	97.01	**62.04**	**71.83**	**90.85**	**64.79**
		FID ↓					
white-box	P4D (Chin et al., 2024)	170.25	158.14	143.52	141.13	159.60	167.03
	UnlearnDiff (Zhang et al., 2024)	144.26	139.36	144.26	136.34	124.59	141.13
black-box	Random	150.37	149.33	159.92	148.96	162.32	171.54
	QF-Attack (Zhuang et al., 2023)	201.78	198.60	194.22	191.06	205.67	199.30
	Ring-A-Bell (Tsai et al., 2023)	152.45	138.76	129.80	128.59	116.58	155.59
	SneakyPrompt (Yang et al., 2024)	155.40	126.19	125.48	131.79	119.84	147.36
	JPA (Ours)	**131.11**	**119.89**	**115.21**	**107.81**	**108.56**	**139.41**

Execution time of different attack methods (the lower the better). (×) indicates a multiplier of JPA.

	P4D	UnlearnDiff	random	QF-Attack	SneakyPrompt	Ring-A-Bell	JPA (ours)
Attack time per prompt (min)	30.70 (4.7×)	26.29 (3.9×)	29.08 (4.3×)	55.42 (8.2×)	59.18 (8.8×)	62.50 (9.3×)	**6.72**

The image fiedility results of JPA and other attack methods.

● Ablation study on λ

λ	1	2	**3**	4	5	6
ASR	64.43	64.17	**67.16**	59.15	63.43	65.67
FID	133.25	133.46	**131.11**	137.60	135.46	136.64

Table 5: Ablation study on λ. Best result **bloded**.

● Ablation study on different encoders

	CLIP	Bert	T5
ASR	67.17	40.92	48.66
FID	131.11	161.29	168.56

Table 6: Attack with different text encoders.