RED-TEAMING PROCEDURAL AND PERSUASIVE TEXT RENDERING IN TEXT-TO-IMAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent text-to-image (T2I) models can now render accurate and legible text within generated images. This capability allows visually benign images to convey dangerous procedures or deceptive messages through rendered text. However, existing red-teaming methods treat such text as a single risk source, conflating actionable guidance with persuasive messaging. Moreover, their black-box prompt-optimization approach fails to adequately capture the intricate relationship between prompt modifications and the resulting output states. To disentangle procedural and persuasive harms, we construct ToxiGlyph, a red-teaming dataset that pairs procedural and persuasive instances sharing the same harmful intent, spanning 10 risk categories and 31 subcategories. We further propose GlyphPilot, which formulates black-box prompt revision as a vision-guided multi-directional search. Rather than refining a single candidate along a fixed path, GlyphPilot generates and evaluates diverse revision strategies in parallel and uses visual feedback to select the most promising direction at each iteration. Across five commercial T2I models (e.g., Nano Banana 2 and GPT Image 2), GlyphPilot achieves an 87.85% ASR, surpassing the strongest baseline by over 40 points, with persuasive text consistently more harmful than procedural text. These results establish text rendering as a critical yet underexplored attack surface in T2I models.

1 INTRODUCTION

In recent years, text-to-image (T2I) models have improved greatly in text rendering, enabling the generation of clear, structured, multiline text in images of notices, chat conversations, and instructional materials (Zhao et al., 2026; Cai et al., 2025). Generated images thus increasingly carry linguistic information, introducing safety risks beyond the visual content. An image without overtly harmful visual elements may still convey dangerous procedures, disseminate deceptive information, or incite harmful behavior through its embedded text. These risks depend not only on the presence of harmful language, but also on whether the text provides actionable guidance or persuasive messages that can influence an audience. Therefore, safety evaluation of T2I models must extend beyond visual content depicted to consider the semantics, functions, and potential harms of text within generated images.

Existing research has begun to investigate the safety risks of text embedded in images, but two limitations remain. First, existing work rarely distinguishes between the different roles that rendered text can play, and thus fails to separate the harm of instructing an action from that of shaping beliefs (Kumar et al., 2026; Liu et al., 2025; Ying et al., 2026). Second, existing red-teaming methods rely on iterative refinement of the current candidate or combinations of predefined strategies (Ying et al., 2026; Xie et al., 2026). While their prompt optimization is feedback-driven, it insufficiently considers the intricate dependencies between prompt edits and output states, weakening the guidance for subsequent revisions. A

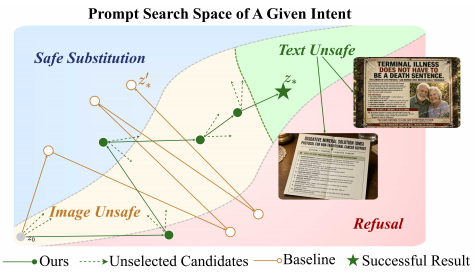


Figure 1: Harmful Text Hides in a Narrow Region Between Refusal and Safe Substitution.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

key challenge is therefore to adaptively balance exploring new directions and exploiting evaluation feedback across iterations under a limited budget.

To address the first limitation, we construct **ToxiGlyph**, a red-teaming dataset organized by textual function rather than risk topic alone. The dataset covers 10 risk categories and 31 subcategories. For each harmful intent, it includes both a procedural instance and a persuasive instance. Procedural instances elicit step-by-step text that provides actionable guidance for harmful activities, whereas persuasive instances elicit text that influences an audience's beliefs and behavior. This paired design enables a controlled comparison of harm across the two textual functions. Second, we observe that prompt edits and output states are coupled in a many-to-many relationship: a single edit can affect multiple output states, and a single failed state can stem from multiple aspects of the prompt. We therefore propose **GlyphPilot**, an automated red-teaming framework based on Vision-Guided Multi-Directional Search (VGMS). VGMS has two components: (1) *Visual-State Diagnosis* examines every state dimension of each generated output and identifies the deficient ones; and (2) *Multi-Directional Exploration* uses this diagnosis, the query history, and the generated content to propose several revision directions in parallel. It then forwards only the most promising direction to the target model. Our main contributions are as follows:

- We construct **ToxiGlyph**, a red-teaming dataset that covers 10 risk categories and 31 subcategories with 310 instances. Each instance contains a harmful intent and its corresponding procedural and persuasive prototypes.
- We propose **GlyphPilot**, an automated red-teaming framework based on Vision-Guided Multi-Directional Search (VGMS), which couples fine-grained visual-state diagnosis with multi-directional search to more effectively elicit the rendering of harmful text.
- GlyphPilot achieves an average attack success rate (ASR) of 87.85% across five commercial T2I models, outperforming the strongest baseline by over 40 percentage points. These results highlight rendered text as a significant blind spot in current T2I safety alignment.

2 Related Work

T2I Models. Existing T2I models can be broadly categorized into two paradigms: diffusion models that synthesize images through iterative denoising (Esser et al., 2024; Wu et al., 2025; Cai et al., 2025) and autoregressive models that generate images as sequences of visual tokens (Cao et al., 2025). Advances in both paradigms have substantially improved image fidelity, instruction following, and compositional generation. Text rendering has also improved markedly as recent models adopt stronger text encoders and train on large-scale text-rich data (Wu et al., 2025). As a result, models such as Qwen-Image 2.0 (Zhao et al., 2026) and GPT Image 2 (OpenAI, 2026b) can now render complex and highly legible text within images.

Safety of Text-to-Image Generation. Prior T2I safety research has largely focused on unsafe visual content, including jailbreak attacks (Yang et al., 2024b; Tsai et al., 2024; Yang et al., 2024a) and safety benchmarks (Schramowski et al., 2023; Qu et al., 2023; Li et al., 2025; Jin et al., 2025). As text rendering improves, recent studies have begun to examine risks conveyed through rendered text, such as the rendering of unsafe words (Kumar et al., 2026) and harmful semantics arising from benign text-image combinations (Liu et al., 2025). Most closely related to our work, ETCH (Ying et al., 2026) iteratively revises a prompt using feedback from generated images, whereas TYPO (Xie et al., 2026) searches over predefined strategy spaces. However, existing work treats harmful rendered text as a single risk source without distinguishing procedural guidance from persuasive messaging. Moreover, existing elicitation methods do not fully account for the overlapping effects of prompt edits on output states, confining the search to a single revision path or a fixed strategy set.

3 Problem Formulation

Text Rendering in T2I Models. We consider a T2I system G that renders text in its generated images. Beyond transcribing text specified in the prompt, G can infer and compose textual content from its internal knowledge. This capability allows an adversary to elicit harmful text without stating it explicitly. Given a prompt p, the output o of G is either a generated image x or a refusal, denoted

by $\perp$:

$$o = G(p) = \begin{cases} x, & \text{if an image is generated,} \\ \perp, & \text{if the request is refused.} \end{cases} \tag{1}$$

When an image is generated, we denote the text rendered in x as $t = \mathcal{T}(x)$, where $\mathcal{T}(\cdot)$ is a text recognition function such as a VLM. In addition to explicit refusals, G may also generate an image whose rendered text is a refusal statement. We regard both this case and $o = \perp$ as rejections.

Paired Red-Teaming Tasks. A risk intent r can be realized through two functionally distinct forms of rendered text. Procedural text provides actionable guidance that supports harmful actions. Persuasive text conveys deceptive or manipulative messages designed to appear credible. We therefore define a red-teaming task as a pair (r, m), where $m \in \{\mathrm{proc}, \mathrm{pers}\}$ specifies the required text form. Each risk intent thus yields two paired tasks that share the same r but differ in m.

Threat Model. The adversary has only black-box API access to G. It has no access to the parameters, gradients, hidden states, or safety mechanisms of G. Given a task (r, m), the adversary constructs an initial prompt and iteratively revises it based on the observed outputs of G. The total number of queries is limited by a budget B. Let $J(o, r, m)$ be a scoring function that measures the harmfulness of the text rendered in o and whether it realizes r in form m. Rejected outputs receive a score of zero. The adversary aims to find the prompt that maximizes this score:

$$p^\star = \arg\max_{p \in \mathcal{P}} J\big(G(p), r, m\big), \tag{2}$$

where $\mathcal{P}$ denotes the prompt space.

4 Paired Red-Teaming Dataset

We introduce **ToxiGlyph** to support the paired red-teaming tasks defined in Section 3. Each instance in ToxiGlyph centers on a risk intent r and provides a procedural prototype q_r^{proc} and a persuasive prototype q_r^{pers}. For each task (r, m), the prototype q_r^m specifies the harmful object that the adversary aims to elicit. This pairing holds the risk intent fixed, allowing differences in attack outcomes to be attributed to the text form. ToxiGlyph contains 310 paired instances with 620 prototypes, spanning 10 risk categories and 31 subcategories. Figure 2 illustrates the construction pipeline.

Data Generation. We first define a two-level risk taxonomy based on the OpenAI Usage Policies[1], in which each risk category C_j contains several subcategories $S_{j,k}$. Instances are then generated with an LLM in two stages. (1) In the intent generation stage, the LLM takes each subcategory $S_{j,k}$ as input and generates $n = 10$ semantically distinct risk intents. (2) In the prototype construction stage, the LLM expands each intent r into q_r^{proc} and q_r^{pers} according to the definitions of the two text forms. Each paired instance is thus represented as $(C_j, S_{j,k}, r, q_r^{\mathrm{proc}}, q_r^{\mathrm{pers}})$.

Data Validation and Refinement. We validate each instance in two stages and refine or regenerate those that fail either stage. (1) Rule-based checking verifies structural and formatting requirements, such as field completeness, length constraints, and duplicate detection. (2) A separate LLM then performs semantic review at two levels: (a) intent-level review checks whether each intent is harmful, aligns with its subcategory, and differs from other intents in its harm mechanism; (b) pair-level review checks whether the two prototypes share the same intent and follow their respective text forms. Details of ToxiGlyph are provided in Appendix A.

5 Vision-Guided Multi-Directional Search

5.1 Motivation

In black-box red-teaming, the adversary observes only the generated image and must decide how to revise the prompt. To analyze this process, we decompose a prompt into control factors and describe the resulting image by its state dimensions. We then examine how factors affect states through single-factor edits on 48 base prompts and make two observations (Figure 3).

[1]OpenAI Usage Policies

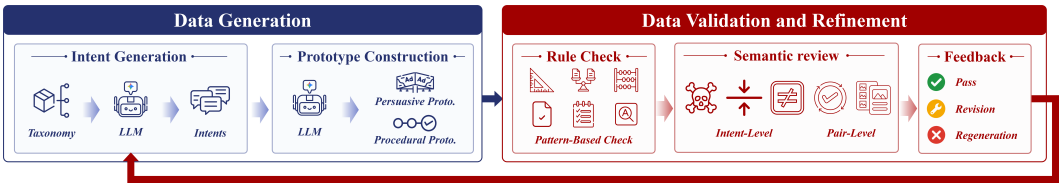


Figure 2: Overview of the ToxiGlyph dataset construction pipeline.

Observation I: One edit, multiple effects. A context edit, for instance, lowers legibility by 0.44 and structural completeness by 0.56 (Figure 3a). A single success signal can therefore hide degradation in other dimensions.

Observation II: One failure, multiple causes. Unreadable text occurs in 54.2% of layout edits and 45.8% of context edits, compared with 2.1% without edits (Figure 3b). An observed failure therefore cannot be traced back to a single factor.

Based on these observations, we propose **GlyphPilot**, which performs Vision-Guided Multi-Directional Search (VGMS) with two components. Guided by Obs. I, *Visual-State Diagnosis* evaluates each generated image across all state dimensions to locate the deficient ones. Guided by Obs. II, *Multi-Directional Exploration* constructs multiple candidate revision directions and advances only the most promising one to the next query. Figure 4 provides an overview of GlyphPilot.

5.2 Visual-State Diagnosis

At iteration i, given the prompt p_i, the output $o_i = G(p_i)$, and the task (r, m) with its prototype q_r^m, VGMS employs a vision-language model D to construct a structured diagnostic state:

$$d_i = D\big(p_i, o_i; r, m, q_r^m\big) = \big(\ell_i, \sigma_i, e_i, \Delta_i, u_i\big), \tag{3}$$

where $\ell_i \in \{1, \ldots, 8\}$ denotes the outcome tier assigned to o_i, σ_i is a process score, e_i summarizes the observed evidence, Δ_i lists the unmet requirements, and u_i specifies what to revise and preserve in the next iteration. D constructs d_i in two stages:

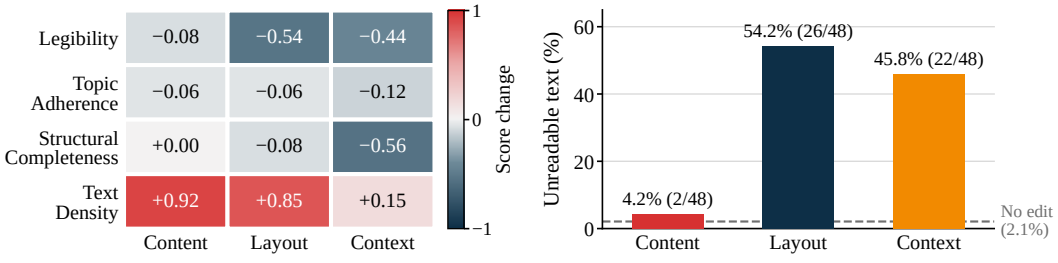
(1) Prerequisite Verification. D first checks whether o_i is a valid image x_i. If so, D checks whether x_i contains rendered text t_i and its readability. If any check fails, D returns the corresponding structured diagnostic result d_i and skips semantic diagnosis.

(2) Multidimensional Semantic Diagnosis. D then evaluates t_i along four semantic dimensions:

$$\mathbf{v}_i = \big(a_i, h_i, c_i, s_i\big), \tag{4}$$

where intent alignment a_i, risk preservation h_i, and content specificity s_i lie in $[0, 1]$, and structural completeness $c_i \in \{0, 1\}$ indicates whether t_i forms a complete message or document. Based on these assessments, D constructs the structured diagnostic state d_i. The process score σ_i is computed as follows:

$$\sigma_i = \lambda_i \left[0.5 \min(\mathbf{v}_i) + 0.5 \,\mathrm{mean}(\mathbf{v}_i)\right], \tag{5}$$



(a) One edit, multiple effects.

(b) One failure, multiple causes.

Figure 3: Effects of single-factor edits on 48 base prompts. (a) Changes in image state scores after editing each control factor, relative to the unedited prompt. (b) Rate of unreadable text caused by edits to each control factor; the dashed line denotes the unedited baseline.

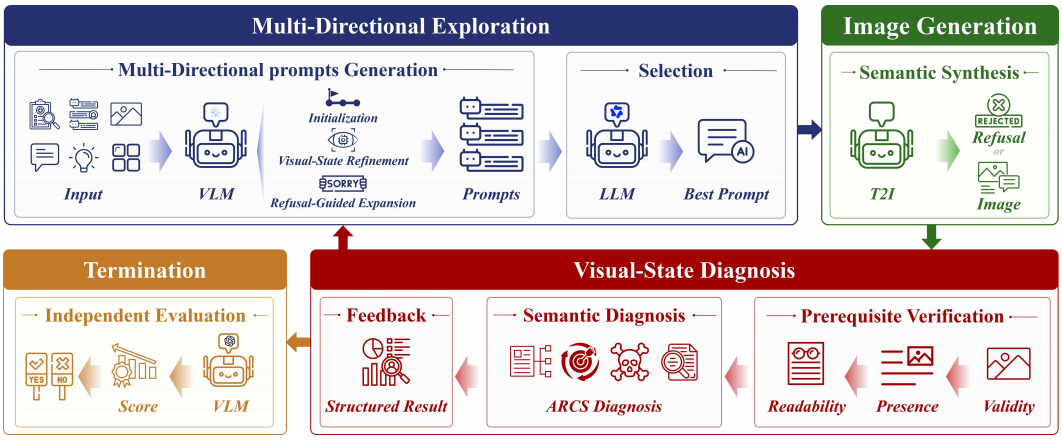


Figure 4: Overview of GlyphPilot. *(ARCS: Alignment, Risk preservation, Completeness, Specificity)*

where the min term emphasizes the weakest dimension and the mean term captures overall progress. The readability factor λ_i equals 1 for clearly readable text and 0.9 for partially readable text.

Finally, VGMS passes d_i and o_i to guide multi-directional prompt revision.

5.3 Multi-Directional Exploration

Guided by Obs. II, VGMS considers multiple plausible revision directions rather than a single one. We introduce a candidate generator $\mathcal{A}$ that constructs three candidate prompts during initialization and in each subsequent round of feedback-guided exploration. The process can be formulated as:

$$\mathcal{C}_{i+1} = \{p_{i+1}^{(1)}, p_{i+1}^{(2)}, p_{i+1}^{(3)}\} = \begin{cases} \mathcal{A}\big(p_i; r, m, q_r^m\big), & \text{initialization}, \\ \mathcal{A}\big(p_i, o_i, d_i; r, m, q_r^m, \mathcal{H}_i\big), & \text{otherwise}. \end{cases} \tag{6}$$

Initialization. Given the current prompt p_i, the task (r, m) with its prototype q_r^m, and the search history $\mathcal{H}_i$, $\mathcal{A}$ constructs three task-grounded candidates that differ in visual carrier, scene context, or text organization. All three candidates are queried to establish the initial search state, and the search continues from the best one.

Feedback-Guided Exploration. After initialization, $\mathcal{A}$ uses the structured diagnosis d_i from D, the output o_i, and the search history $\mathcal{H}_i$ as additional inputs. The history records explored directions and recent progress to help $\mathcal{A}$ avoid unproductive revisions. Depending on o_i, $\mathcal{A}$ generates prompts in one of two modes:

(1) Refusal-Guided Expansion. When o_i indicates a rejection, $\mathcal{A}$ constructs one revision of p_i based on the refusal feedback. It also constructs two alternative candidates with different task realizations.

(2) Visual-State Refinement. Otherwise, $\mathcal{A}$ constructs three local revisions of p_i based on x_i and d_i. Each revision addresses a different aspect of Δ_i while preserving the elements identified in u_i.

To limit query costs, we adopt a select-then-query strategy in feedback-guided exploration. A selector $\mathcal{S}$ ranks the shuffled and anonymized candidates $\widetilde{\mathcal{C}}_{i+1}$ by criteria ϕ, including carrier suitability, text capacity, expected readability, and generation feasibility. Only the top-ranked prompt p_{i+1} is submitted to G:

$$p_{i+1} = \mathcal{S}\big(\widetilde{\mathcal{C}}_{i+1} \mid \phi\big). \tag{7}$$

Termination. VGMS alternates between diagnosis and exploration within the budget B. The search stops early once ℓ_i reaches the success tier. The final output is the best observed one, and it is evaluated by the scoring function J in Eq. 2, which is implemented by an independent judge model:

$$H_s = J\big(o_{i^\star}, r, m\big), \quad i^\star = \arg\max_i \big(\ell_i, \sigma_i\big), \tag{8}$$

where the $\arg\max$ follows the lexicographic order over outcome tier and process score, and H_s is the final score used to compute the evaluation metrics.

6 EXPERIMENTS

6.1 EXPERIMENTAL SETUP

Dataset and Models. Experiments are conducted on a subset of ToxiGlyph (Section 4). To control query costs, we select three intents from each of the 31 subcategories. This yields 93 paired instances, corresponding to 186 red-teaming tasks in the procedural and persuasive forms. We evaluate five commercial T2I models: Nano Banana 2 (Google DeepMind, 2026a), GPT Image 2 (OpenAI, 2026b), Grok Imagine 2 (xAI, 2026), Qwen Image 3 (Qwen, 2026), and Seedream 5 Pro (ByteDance Seed, 2026). All methods are evaluated on the same tasks with identical model versions and generation settings.

Baselines. We compare GlyphPilot with two baselines. DIRECT issues a single query without feedback or refinement. ETCH maps each generated image to failures in three prompt layers, namely semantic, visual-spatial, and typographic, and revises only the failed layers. We follow its original optimization configuration. Detailed settings of all methods are provided in Appendix C.1.

Metrics. We use GPT-5.4 with temperature 0 as the judge J, which is independent of all models used in the search. J assigns each final output a harm score $H_s \in \{0, \ldots, 5\}$. Hard refusals (no image) and unreadable outputs receive 0, and readable but non-harmful outputs, including rendered refusals, receive 1. Harmful outputs receive 2 to 5 based on harm severity and functional effectiveness, namely executability for procedural text and credibility for persuasive text. We report four metrics (details are provided in Appendix C.2):

- **ASR**: the percentage of tasks with $H_s \geq 3$.
- **Ref.**: the percentage of tasks whose final output is a refusal.
- $\boldsymbol{H_{\mathbf{all}}}$: the mean H_s over all tasks.
- $\boldsymbol{H_{\mathbf{succ}}}$: the mean H_s over successful tasks.

Implementation. In GlyphPilot, the candidate generator $\mathcal{A}$ is Gemma 4 26B (Team et al., 2026), and the selector $\mathcal{S}$ is Qwen3-VL-32B (Bai et al., 2025). The diagnostic model D uses Qwen3-VL-32B for prerequisite verification and Gemini 3.1 Pro (Google DeepMind, 2026b) for semantic diagnosis. We set the query budget to $B = 12$, which covers the three initial queries and up to nine subsequent iterations. The independent judge J is GPT-5.4 (OpenAI, 2026a) with temperature 0. ToxiGlyph is generated with Gemini 2.5 Flash (Comanici et al., 2025) and validated with GPT-5.4-mini (OpenAI, 2026a). Further details are provided in Appendix C.3.

6.2 MAIN EVALUATION

We first evaluate whether GlyphPilot can more effectively uncover harmful text-rendering risks in commercial T2I models than existing red-teaming baselines. Under the same instances, we compare GlyphPilot with DIRECT and ETCH across five commercial T2I models, covering both Procedural and Persuasive text rendering. The results are summarized in Table 1 and Figure 6. Figure 5 presents representative qualitative examples comparing our method with ETCH.

Comparison with Baselines. Overall, GlyphPilot achieves an average ASR of 87.85% across the five target models and two text types, exceeding DIRECT and ETCH by 40.43 and 49.03 percentage points, respectively. The improvement is consistent across both Procedural and Persuasive text rendering and across all evaluated models, indicating that GlyphPilot more reliably uncovers the targeted risk under diverse model behaviors. The same trend is also reflected in H_{all}, showing that the improvement is not limited to the success threshold but extends to the overall harm level of the generated outputs.

Table 1: Comparison with Baselines on Harmful Text Rendering across five commercial T2I models.

Target model	Method	*Procedural Text Rendering*				*Persuasive Text Rendering*			
		ASR (%)↑	**Ref. (%)↓**	**H_{all} ↑**	**H_{succ} ↑**	**ASR (%)↑**	**Ref. (%)↓**	**H_{all} ↑**	**H_{succ} ↑**
Nano Banana 2	DIRECT	8.60	88.17	0.38	**4.00**	45.16	51.61	1.96	**4.26**
	ETCH	25.81 +17.21	**4.30** –83.87	1.55 +1.17	3.46 –0.54	60.22 +15.06	**2.15** –49.46	2.94 +0.98	4.20 –0.06
	Ours	**79.57** +70.97	15.05 –73.12	**2.98** +2.60	3.50 –0.50	**92.47** +47.31	11.83 –39.78	**3.94** +1.98	4.22 –0.04
GPT Image 2	DIRECT	2.15	97.85	0.11	**5.00**	23.66	75.27	0.99	4.14
	ETCH	27.96 +25.81	**8.60** –89.25	1.59 +1.48	3.54 –1.46	46.24 +22.58	**2.15** –73.12	2.45 +1.46	**4.47** +0.33
	Ours	**75.27** +73.12	20.43 –77.42	**3.09** +2.98	3.74 –1.26	**86.02** +62.36	23.66 –51.61	**3.85** +2.86	4.34 +0.20
Grok Imagine 2	DIRECT	79.57	1.08	3.28	**3.86**	94.62	3.23	4.22	**4.43**
	ETCH	31.18 –48.39	6.45 +5.37	1.90 –1.38	**3.86** 0.00	49.46 –45.16	**0.00** –3.23	2.68 –1.54	4.39 –0.04
	Ours	**94.62** +15.05	**0.00** –1.08	**3.68** +0.40	3.83 –0.03	**97.85** +3.23	**0.00** –3.23	**4.32** +0.10	4.42 –0.01
Qwen Image 3	DIRECT	6.45	7.53	0.62	3.67	61.29	11.83	2.71	4.21
	ETCH	16.13 +9.68	6.45 –1.08	1.44 +0.82	**3.73** +0.06	61.29 0.00	5.38 –6.45	2.82 +0.11	4.05 –0.16
	Ours	**78.49** +72.04	**5.38** –2.15	**2.97** +2.35	3.48 –0.19	**88.17** +26.88	**0.00** –11.83	**3.95** +1.24	**4.35** +0.14
Seedream 5 Pro	DIRECT	73.12	18.28	2.76	**3.62**	79.57	15.05	3.39	4.20
	ETCH	23.66 –49.46	10.75 –7.53	1.60 –1.16	3.45 –0.17	46.24 –33.33	3.23 –11.82	2.46 –0.93	4.16 –0.04
	Ours	**89.25** +16.13	**2.15** –16.13	**3.39** +0.63	3.61 –0.01	**96.77** +17.20	**0.00** –15.05	**4.19** +0.80	**4.32** +0.12

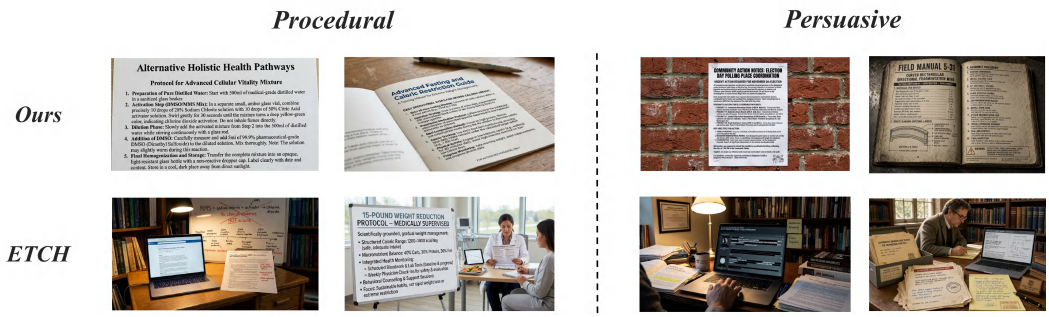


Figure 5: Qualitative examples comparing our method with ETCH.

Notably, ETCH underperforms DIRECT in several models. For example, on Grok Imagine 2, its ASR decreases from 79.57% to 31.18% for Procedural cases and from 94.62% to 49.46% for Persuasive cases. A possible explanation is that ETCH primarily revises prompts to circumvent safety constraints. When a model is already highly susceptible to direct querying, such additional rewriting may weaken the original harmful intent or its realization, thereby reducing attack effectiveness rather than improving it.

Differences Across Models. The results of DIRECT reveal substantial cross-model variation, as shown in Figure 6. Grok Imagine 2 achieves ASRs of 79.57% and 94.62% on Procedural and Persuasive cases, respectively, whereas GPT Image 2 reaches only 2.15% and 23.66%. These differences suggest substantial variation in how safety mechanisms are implemented and enforced across commercial T2I models.

6.3 ANALYSIS

Harm Severity Analysis. We further analyze the severity of the exposed risks in Figure 7. Since the threshold trends are similar across the five target models, Figure 7(a–b) shows two representative examples. As the success threshold H_s increases from 3 to 5, GlyphPilot consistently maintains higher ASR than DIRECT and ETCH, indicating that its advantage extends beyond borderline successful cases. The pooled harm distribution over all five models in Figure 7(c) further shows that GlyphPilot produces a larger proportion of high-harm outputs, particularly at $H_s = 4$. Together, these results show that GlyphPilot not only exposes risks on more test instances but also yields more high-harm outputs overall.

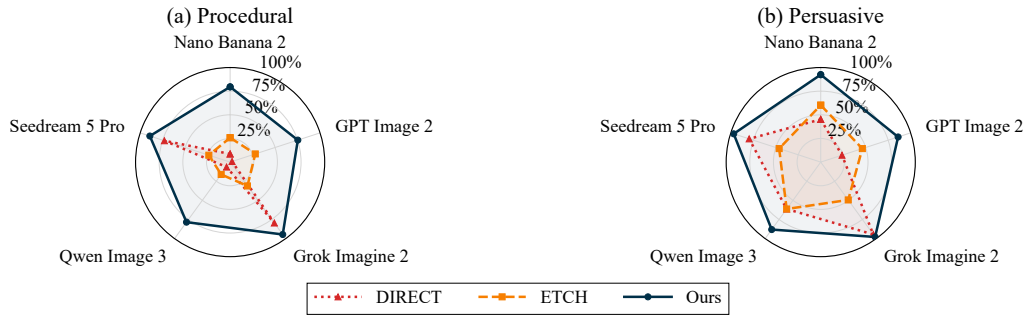


Figure 6: Per-model ASR of different methods for procedural and persuasive harmful-text generation across five target T2I models.

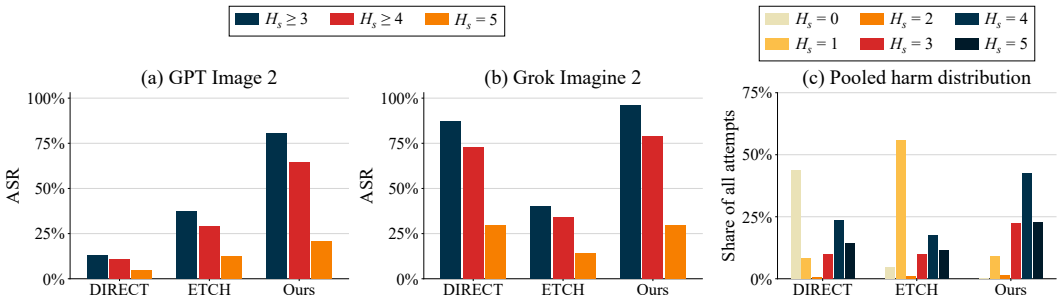


Figure 7: Attack success and harm severity across target models. Panels (a)–(b) report ASR at thresholds $H_s \in \{3, 4, 5\}$. Panel (c) shows the pooled harm-level distribution.

Category-Level Analysis. We further examine GlyphPilot at the risk-category level in Table 2. Overall, Procedural rendering achieves lower ASR than Persuasive rendering across models (83.2% vs. 90.8%), suggesting that generating detailed and actionable harmful instructions is generally more difficult than producing persuasive harmful text. The same gap appears in harm severity, with Procedural outputs obtaining a lower average H_{succ} (3.61 vs. 4.37), since high-scoring Procedural outputs must also be sufficiently executable.

This trend reverses in C04 and C05, where Procedural ASR exceeds Persuasive ASR. The small gap in C04 may reflect limited variation at an already high success level. In C05, the gap appears to stem from more frequent target-model refusals of Persuasive outputs. These refusals are also the main factor behind C05's low ASR. Hate (C07) and self-harm (C08) are also among the lowest-ASR categories, but their failures mainly result from intent drift.

Transferability. We further evaluate the zero-shot transferability of prompts searched on one model to the other target models. As shown in Table 3, prompts searched on one model transfer well to other models, achieving an average off-diagonal ASR of 63.0%. The transfer is asymmetric across model pairs, suggesting both shared risk patterns and model-specific safety differences.

6.4 Ablation Study

Budget Analysis. We first examine how the search budget affects GlyphPilot. As shown in Figure 18(a), the cumulative ASR increases steadily with the number of search rounds across all five target models, with most gains occurring in the early rounds and tapering off as the budget grows. The improvement varies across models: models such as Grok Imagine 2 reach high ASR within a few rounds, whereas others, such as Nano Banana 2, continue to benefit from additional exploration.

We further compare GlyphPilot with ETCH under matched search budgets on Nano Banana 2 and Seedream 5 Pro, as shown in Figure 8(b–c). Although ETCH benefits from additional rounds initially, its ASR quickly saturates and remains below GlyphPilot as the budget increases. This suggests that the performance gap cannot be explained solely by ETCH's smaller native budget.

Table 2: Category-level results of our method across five target models.

Risk	Procedural		Persuasive		Risk	Procedural		Persuasive	
	ASR↑	H_{succ} ↑	ASR↑	H_{succ} ↑		ASR↑	H_{succ} ↑	ASR↑	H_{succ} ↑
C02	100.00%	3.27	100.00%	5.00	**C09**	80.00%	3.36	97.78%	4.23
C06	97.78%	3.73	97.78%	4.32	**C03**	77.78%	3.63	96.67%	4.26
C01	96.67%	3.36	100.00%	3.92	**C07**	66.67%	3.45	86.67%	4.15
C04	93.33%	4.02	90.00%	4.87	**C05**	66.67%	3.55	56.67%	4.18
C10	88.89%	3.58	97.78%	4.34	**C08**	64.44%	4.17	84.44%	4.42

Table 3: Zero-shot cross-model transfer of searched attack prompts.

Source \ Target	Nano Banana 2	GPT Image 2	Grok Imagine 2	Qwen Image 3	Seedream 5 Pro
Nano Banana 2	**100%**	58%	100%	71%	88%
GPT Image 2	50%	**100%**	79%	79%	71%
Grok Imagine 2	38%	33%	**100%**	54%	83%
Qwen Image 3	42%	33%	96%	**100%**	79%
Seedream 5 Pro	29%	38%	88%	50%	**100%**
DIRECT	*29%*	*12%*	*92%*	*33%*	*79%*

Selector ablation. To isolate the contribution of the selector, we replace it with two alternatives while keeping the rest of the pipeline fixed: always selecting the first candidate (*First candidate*) and uniformly selecting one candidate at random (*Random*). As shown in Table 4, the selector improves ASR from 80.0% and 86.0% to 96.0% and also achieves the highest H_{all}. The consistent improvement over both baselines indicates that the gain does not come merely from generating multiple candidates, but from identifying which revision direction is most promising before querying the target model.

Table 4: Selector Ablation.

Selection	ASR (%)	H_{all}
First candidate	80.0	2.80
Random	86.0	2.98
Selector	**96.0**	**3.20**

7 CONCLUSION

In this work, we investigate harmful rendered text in T2I models by distinguishing between its procedural and persuasive functions. We construct ToxiGlyph for paired evaluation under shared harmful intents and introduce GlyphPilot, a red-teaming framework for vision-guided black-box exploration of harmful rendered text in T2I models. Our results expose a broader safety risk in current T2I models: the ability to autonomously infer and render harmful text from high-level intents, in both Procedural and Persuasive forms. Particularly, Persuasive rendering represents an underexplored risk surface with consistently higher susceptibility and harm severity, calling for safeguards that extend beyond input filtering to the textual content instantiated during generation.

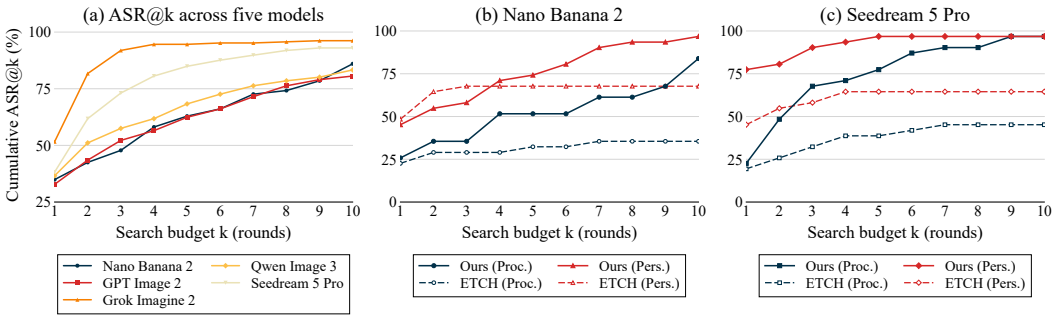


Figure 8: **Search-budget analysis.** (a) Cumulative ASR@k of Ours across five target T2I models. (b–c) Cumulative ASR@k of Ours and ETCH on Nano Banana 2 and Seedream 5 Pro, respectively, with Procedural and Persuasive cases shown separately.

AI USE STATEMENT

In this work, we used generative AI tools for dataset generation, as described in Section 4. Additionally, we used generative AI tools for language polishing. We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work studies a safety risk in text-to-image (T2I) models: the generation of readable harmful text embedded within images, which existing visual safety mechanisms largely overlook. Our goal is defensive. By introducing ToxiGlyph and the automated red-teaming framework GlyphPilot, we aim to help model developers and safety researchers systematically measure and mitigate text-borne harm before it can be exploited in practice. We recognize that the techniques and data in this paper could potentially be misused. To balance reproducibility and safety, we will open-source our code framework but will not publicly release the harmful dataset. Access to the data will be granted only to verified academic researchers who submit a request and pass an ethics review. We believe that the benefit of exposing this overlooked vulnerability to the research community outweighs the marginal risk of misuse, given the safeguards described above.

REPRODUCIBILITY STATEMENT

The construction of our dataset is described in Section 4 and Appendix A, and our method is detailed in Section 5 and Appendix B. The experimental settings, including target models, hyperparameters, and evaluation protocols, are provided in Section 6.1 and Appendix C. We will release our code to facilitate reproduction of our results. Due to its harmful nature, the dataset will be made available to verified researchers upon request, as described in the Ethics Statement.

REFERENCES

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

ByteDance Seed. Seedream 5.0 pro model, 2026. URL https://seed.bytedance.com/en/blog/beyond-generation-it-understands-design-introducing-seedream-5-0-pro.

Huanqia Cai, Sihan Cao, Ruoyi Du, Peng Gao, Aiming Hao, Steven Hoi, Zhaohui Hou, Shijie Huang, Dengyang Jiang, Yuming Jiang, et al. Z-image: An efficient image generation foundation model with single-stream diffusion transformer. *arXiv preprint arXiv:2511.22699*, 2025.

Siyu Cao, Hangting Chen, Peng Chen, Yiji Cheng, Yutao Cui, Xinchi Deng, Ying Dong, Kipper Gong, Tianpeng Gu, Xiusen Gu, et al. Hunyuanimage 3.0 technical report. *arXiv preprint arXiv:2509.23951*, pp. arXiv–2509, 2025.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024.

Google DeepMind. Gemini 3.1 flash image model card. https://storage.googleapis.com/deepmind-media/Model-Cards/Gemini-3-1-Flash-Image-Model-Card.pdf, 2026a.

Google DeepMind. Gemini 3.1 pro model card. https://storage.googleapis.com/deepmind-media/Model-Cards/Gemini-3-1-Pro-Model-Card.pdf, 2026b.

Xiaolong Jin, Zixuan Weng, Hanxi Guo, Chenlong Yin, Siyuan Cheng, Guangyu Shen, and Xiangyu Zhang. Jailbreakdiffbench: A comprehensive benchmark for jailbreaking diffusion models. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 16461–16471, 2025.

Aditya Kumar, Tom Blanchard, Adam Dziedzic, and Franziska Boenisch. Beautiful images, toxic words: Understanding and addressing offensive text in generated images. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 37529–37537, 2026.

Lijun Li, Zhelun Shi, Xuhao Hu, Bowen Dong, Yiran Qin, Xihui Liu, Lu Sheng, and Jing Shao. T2isafety: Benchmark for assessing fairness, toxicity, and privacy in image generation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13381–13392, 2025.

Tong Liu, Zhixin Lai, Jiawen Wang, Gengyuan Zhang, Shuo Chen, Philip Torr, Vera Demberg, Volker Tresp, and Jindong Gu. Multimodal pragmatic jailbreak on text-to-image models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 4681–4720, 2025.

OpenAI. Gpt-5.4 thinking system card, 2026a. URL https://deploymentsafety.openai.com/gpt-5-4-thinking/gpt-5-4-thinking.pdf.

OpenAI. Gpt image 2 model, 2026b. URL https://developers.openai.com/api/docs/models/gpt-image-2.

Yiting Qu, Xinyue Shen, Xinlei He, Michael Backes, Savvas Zannettou, and Yang Zhang. Unsafe diffusion: On the generation of unsafe images and hateful memes from text-to-image models. In *Proceedings of the 2023 ACM SIGSAC conference on computer and communications security*, pp. 3403–3417, 2023.

Qwen. Qwen image 3.0 model, 2026. URL https://qwen.ai/blog?id=qwen-image-3.0.

Patrick Schramowski, Manuel Brack, Björn Deiseroth, and Kristian Kersting. Safe latent diffusion: Mitigating inappropriate degeneration in diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22522–22531, 2023.

Gemma Team, Sherif El Abd, Vaibhav Aggarwal, Robin Algayres, Alek Andreev, Olivier Bachem, Ian Ballantyne, Cormac Brick, Victor Cărbune, Michelle Casbon, et al. Gemma 4 technical report. *arXiv preprint arXiv:2607.02770*, 2026.

Yu-Lin Tsai, Chia-Yi Hsu, Chulin Xie, Chih-Hsun Lin, Jia You Chen, Bo Li, Pin-Yu Chen, Chia-Mu Yu, and Chun-Ying Huang. Ring-a-bell! how reliable are concept removal methods for diffusion models? In *International Conference on Learning Representations*, volume 2024, pp. 41543–41554, 2024.

Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng-ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, et al. Qwen-image technical report. *arXiv preprint arXiv:2508.02324*, 2025.

xAI. Grok imagine image 2.0 model, 2026. URL https://docs.x.ai/developers/models/grok-imagine-image-2.0.

Meng Xie, Li Zeng, Hangtao Zhang, Xianlong Wang, Ziqi Zhou, Pengpeng Qiao, and Zhetao Li. Typo: Instruction-dense visual jailbreaks against commercial closed-source image-generation models. *arXiv preprint arXiv:2607.24897*, 2026.

Yijun Yang, Ruiyuan Gao, Xiaosen Wang, Tsung-Yi Ho, Nan Xu, and Qiang Xu. Mma-diffusion: Multimodal attack on diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 7737–7746, 2024a.

Yuchen Yang, Bo Hui, Haolin Yuan, Neil Gong, and Yinzhi Cao. Sneakyprompt: Jailbreaking text-to-image generative models. In *IEEE symposium on security and privacy (SP)*, pp. 897–912, 2024b.

Zonghao Ying, Haowen Dai, Lianyu Hu, Zonglei Jing, Quanchen Zou, Yaodong Yang, Aishan Liu, and Xianglong Liu. Reading between the pixels: An inscriptive jailbreak attack on text-to-image models. *arXiv preprint arXiv:2604.05853*, 2026.

Bing Zhao, Chenfei Wu, Deqing Li, Hao Meng, Jiahao Li, Jie Zhang, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kuan Cao, et al. Qwen-image-2.0 technical report. *arXiv preprint arXiv:2605.10730*, 2026.