TRAP: Targeted Redirecting of Agentic Preferences

Hangoo Kang[*1]**, Jehyeok Yeon**[*1]**,Gagandeep Singh**[1]
[1]University of Illinois Urbana-Champaign
{hangook2,jehyeok2,ggnds}@illinois.edu

Abstract

Autonomous agentic AI systems powered by vision-language models (VLMs) are rapidly advancing toward real-world deployment, yet their cross-modal reasoning capabilities introduce new attack surfaces for adversarial manipulation that exploit semantic reasoning across modalities. Existing adversarial attacks typically rely on visible pixel perturbations or require privileged model or environment access, making them impractical for stealthy, real-world exploitation. We introduce TRAP, a novel generative adversarial framework that manipulates the agent's decision-making using diffusion-based semantic injections into the vision-language embedding space. Our method combines negative prompt–based degradation with positive semantic optimization, guided by a Siamese semantic network and layout-aware spatial masking. Without requiring access to model internals, TRAP produces visually natural images yet induces consistent selection biases in agentic AI systems. We evaluate TRAP on the Microsoft Common Objects in Context (COCO) dataset, building multi-candidate decision scenarios. Across these scenarios, TRAP consistently induces decision-level preference redirection on leading models, including LLaVA-34B, Gemma3, GPT-4o, and Mistral-3.2, significantly outperforming existing baselines such as SPSA, Bandit, and standard diffusion approaches. These findings expose a critical, generalized vulnerability: autonomous agents can be consistently misled through visually subtle, semantically-guided cross-modal manipulations. Overall, our results show the need for defense strategies beyond pixel-level robustness to address semantic vulnerabilities in cross-modal decision-making. The code for TRAP is accessible on GitHub at https://github.com/uiuc-focal-lab/TRAP.

1 Introduction

Vision-Language Models (VLMs) and autonomous agentic AI systems have significantly advanced the capability of machines to navigate and interpret open-world environments [Radford et al., 2021, Li et al., 2022a, Alayrac et al., 2022]. However, these powerful multimodal systems also introduce new vulnerabilities, particularly through adversarial manipulations that exploit their integrated visual-textual perception [Zhou et al., 2023, Moosavi-Dezfooli et al., 2016b]. A critical emerging threat is cross-modal prompt injection, in which adversaries embed misleading semantic cues in one modality (e.g., an image) to influence the interpretation and decision making of a model in another modality (e.g., language understanding) [Liu et al., 2023c]. Unlike traditional unimodal adversarial attacks that primarily perturb pixels or text unnoticeably [Goodfellow et al., 2015, Uesato et al., 2018, Madry et al., 2019], these cross-modal attacks leverage semantic shifts, misleading autonomous agents without triggering human suspicion.

Fully autonomous agents, such as GUI agents that navigate web interfaces without human oversight, are particularly susceptible to adversarial manipulations. Recent work has shown that visual-language agents can be jailbroken by adversarial environments, leading to unintended and potentially harmful

[*]Equal contribution

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

actions [Liao et al., 2025, Zhang et al., 2024b]. For example, a malicious pop-up or UI component could trick an agent into clicking harmful links or executing unauthorized tasks, without human intervention [Wu et al., 2024]. This highlights a critical safety flaw: such agents inherently trust their perceptual inputs, making them highly vulnerable to subtle semantic perturbations [Li et al., 2024].

In this paper, we introduce TRAP, a novel adversarial framework explicitly designed to exploit agentic systems' vulnerabilities through semantic injection using diffusion models. Our approach leverages the generative system of Stable Diffusion Rombach et al. [2022b] in combination with CLIP embeddings to create realistic adversarial images that subtly mislead an agentic AI system's decision.

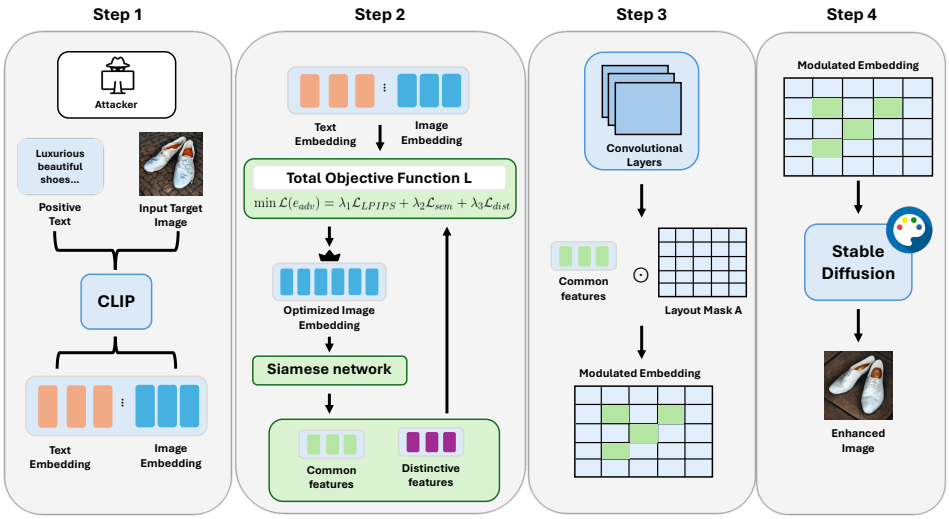

Figure 1: Overview of the TRAP adversarial embedding optimization framework.

TRAP operates in four stages (Fig. 1). First, we extract CLIP embeddings for the target image and adversarial prompt. Second, we iteratively optimize the image embedding using a Siamese semantic network guided by prompt-aligned cues (e.g., "luxury," "premium quality"), with multiplicative fusion modulated by a spatial layout mask [Chaitanya et al., 2020, Lee et al., 2021, Li et al., 2022b]. Third, we apply perceptual and semantic losses, including LPIPS Zhang et al. [2018], to preserve identity and realism during optimization. Fourth, the modified embedding is decoded into a final image using Stable Diffusion. This process yields images that are visually plausible yet semantically manipulated to influence downstream agent decisions.

We rigorously evaluate TRAP using automated multimodal LLM-based evaluators performing N-way comparisons, benchmarking against standard baselines such as: diffusion generation without optimization, Simultaneous Perturbation Stochastic Approximation (SPSA) [Spall, 1987], and Bandit [Ilyas et al., 2018b]. Our evaluation spans six leading multimodal models: LLaVA-1.5-34B [Liu et al., 2023a], Gemma3-8B [Mesnard et al., 2025], Mistral-small-3.1-24B and Mistral-small-3.2-24B, GPT-4o [OpenAI et al., 2024], and CogVLM [Wang et al., 2024a], covering both open- and closed-source architectures. A robust agent should maintain its original objectives and resist such preference manipulations, as allowing adversarial inputs to fundamentally alter decision-making processes undermines the agent's intended purpose and trustworthiness. Our results demonstrate that TRAP significantly outperforms these baselines in shifting autonomous agents' preferences toward adversarially manipulated images. These findings provide strong evidence of critical vulnerabilities within autonomous multimodal systems, raising important safety and trustworthiness concerns about relying exclusively on autonomous perception without adequate safeguards.

Ultimately, this paper serves as both a critical demonstration of potential security vulnerabilities in agentic AI systems and a call to action for developing more robust multimodal alignment, perception safeguards, and adversarial defenses within autonomous systems deployed in real-world environments.

2 Related Works

2.1 Adversarial Attacks on Agentic Systems

Recent advancements in autonomous, agentic AI systems have revealed critical vulnerabilities to adversarial manipulations. Yang et al. [2024] and Wang et al. [2024b] demonstrate backdoor-injection attacks that subtly corrupt agent behavior, misleading web agents during decision making. Similarly, Wu et al. [2024] and Liao et al. [2025] reveal that carefully crafted prompt injections can lead agents to take unintended actions, ranging from disclosing private information to leaking web contents. However, these approaches require extensive access to either the environment's internals or the model's parameters, an assumption rarely met in practical settings. In contrast, our work addresses a more realistic attack setting in which the adversary can only manipulate their own indexed input elements (e.g., images or prompts) without any knowledge of the underlying environment code or model weights.

2.2 Image-based Adversarial Attacks

Image-based adversarial attacks have been extensively studied, focusing primarily on neural network classifiers. White-box attacks, which require full gradient access, range from the foundational Fast Gradient Sign Method (FGSM) [Goodfellow et al., 2015] to iterative optimization methods. These include Projected Gradient Descent (PGD) [Madry et al., 2019] and more advanced approaches like Carlini & Wagner (C&W) [Carlini and Wagner, 2016] and DeepFool [Moosavi-Dezfooli et al., 2016a], which formulate the attack as a constrained optimization problem to find minimal perturbations.

In the more constrained black-box setting, query-based methods estimate gradients using techniques like finite differences (SPSA [Uesato et al., 2018], ZOO [Chen et al., 2017b]) or evolutionary strategies (Bandits [Ilyas et al., 2018b], NES [Ilyas et al., 2018a]). Other approaches bypass gradient estimation entirely, such as the query-efficient Square Attack [Andriushchenko et al., 2020] or decision-based methods like Boundary Attack [Brendel et al., 2018]. Different attack modalities have also been explored, including localized Adversarial Patches [Brown et al., 2017] for physical-world robustness and Universal Adversarial Perturbations (UAP) [Moosavi-Dezfooli et al., 2017] that are image-agnostic.

These approaches typically aim to induce misclassification or alter model output minimally and undetectably. Our work expands upon these methodologies by leveraging semantic manipulation through text-guided diffusion models, aiming to influence model decisions at a deeper semantic level.

2.3 Diffusion Models and Semantic Image Manipulation

Diffusion-based generative models, such as Stable Diffusion, have emerged as powerful tools for high-fidelity image synthesis guided by textual prompts. These models encode rich semantic relationships between text and image domains, enabling precise manipulation of generated images. For example, Wang et al. [2023], Dai et al. [2024] fine-tune latent diffusion codes to introduce targeted changes in object appearance, such as color shifts or texture edits, that mislead classification models while remaining imperceptible to humans. Liu et al. [2023b] guides the reverse diffusion process using free-text instructions, producing adversarial examples that adhere to natural language descriptions and allow fine-grained control over semantic attributes. Zhai et al. [2023a] shows that poisoning only a small subset of text–image pairs during training can backdoor large text-to-image diffusion models at the pixel, object, or style level, embedding hidden triggers that activate under specific prompts. In contrast to these methods, our approach uses only the model embeddings to generate adversarial images, without requiring access to model parameters or training data for the diffusion model.

3 Preliminaries

3.1 Stable Diffusion and Textual Guidance

Modern diffusion models, such as Stable Diffusion [Rombach et al., 2022b], operate not in the high-dimensional pixel space $\mathbb{R}^{C\times H\times W}$ but in a computationally cheaper, perceptually-equivalent latent space. This is achieved using a Variational Autoencoder (VAE), which consists of an encoder $\mathcal{E}_{\text{VAE}}$ and a decoder $\mathcal{D}_{\text{VAE}}$. The encoder compresses a full-resolution image x_0 into a smaller latent

representation $z_0 = \mathcal{E}_{\text{VAE}}(x_0)$. The decoder is trained to reconstruct the image from this latent, $x_0 \approx \mathcal{D}_{\text{VAE}}(z_0)$.

The core diffusion process, which learns to reverse a noising process, is trained entirely within this latent space. The generative model is a denoising network ϵ_θ that is trained to predict the noise ϵ that was added to a noisy latent z_t at timestep t. This denoising network can be conditioned on external information, such as a text prompt. To achieve this, the text prompt p is first converted into a d-dimensional embedding $c = E_T(p)$ using a text encoder. This conditioning vector c is then fed into the ϵ_θ network, typically via cross-attention layers, allowing the text to guide the denoising process: $\epsilon_\theta(z_t, t, c)$.

To strengthen this guidance, modern samplers use Classifier-Free Guidance (CFG) [Ho and Salimans, 2022]. At each denoising step t, the network makes two predictions: one conditioned on the prompt's embedding c, $\epsilon_\theta(z_t, t, c)$, and one unconditioned, $\epsilon_\theta(z_t, t, \emptyset)$, which uses a null-text embedding $\emptyset$. The model then extrapolates in the "direction" of the prompt by mixing these two predictions where the scalar w is the guidance scale (CFG scale) and a higher w forces the generation to adhere more strictly to the text prompt c:

$$\hat{\epsilon}_t = \epsilon_\theta(z_t, t, \emptyset) + w \cdot \big(\epsilon_\theta(z_t, t, c) - \epsilon_\theta(z_t, t, \emptyset)\big)$$

3.2 CLIP

Contrastive Language–Image Pre-training (CLIP) [Radford et al., 2021] learns to connect images and text by projecting them into a single, shared embedding space. It achieves this by training two encoders simultaneously: a vision encoder E_V and a text encoder E_T. These encoders map an image x and a text prompt p into a shared d-dimensional space, producing normalized embeddings $v = E_V(x)$ and $u = E_T(p)$, where d is the dimensionality of this shared space.

The training objective is contrastive: given a batch of (x, p) pairs, the model is trained to maximize the similarity for correctly matched pairs while minimizing the similarity for all incorrect unmatched pairs. This similarity is measured by the scaled dot product of their embeddings, $v_i^\top u_j / \tau$.

$$\mathcal{L}_{\text{CLIP}} = -\frac{1}{2}\left[\log \frac{\exp(v^\top u/\tau)}{\sum_j \exp(v^\top u_j/\tau)} + \log \frac{\exp(v^\top u/\tau)}{\sum_i \exp(v_i^\top u/\tau)}\right],$$

where τ is a learned temperature parameter. This symmetric function is the average of two cross-entropy losses. The first term (image-to-text loss) treats the task as classifying the correct text embedding u (the "positive" sample) for a given image embedding v, out of all N text embeddings $\{u_j\}$ (also including the "negative" samples) in the batch. The second term (text-to-image loss) does the opposite, classifying the correct v for a given u from all N image embeddings $\{v_i\}$.

Minimizing $\mathcal{L}_{\text{CLIP}}$ forces the model to maximize the softmax probability for the correct pair in both directions. This simultaneously maximizes the dot product of matched pairs (the numerators) and minimizes the dot products of all unmatched, "negative" pairs (the denominators). This dynamic "pulls" semantically related image-text embeddings together and "pushes" unrelated pairs apart, creating a shared space where proximity equates to semantic similarity.

3.3 Learned Perceptual Similarity

The Learned Perceptual Image Patch Similarity (LPIPS) metric [Zhang et al., 2018] compares deep features extracted from a pre-trained classification network $\mathcal{F}$, which is used as a fixed feature extractor.

Given two images, a reference x_r and a perturbed image x_p, both are passed through the network $\mathcal{F}$. LPIPS extracts the activation maps $h_r^l, h_p^l \in \mathbb{R}^{C_l \times H_l \times W_l}$ from a set of L different layers. These activations are then normalized along the channel dimension denoted $\hat{h}^l$. The squared ℓ_2 distance is computed between these normalized activations, and this distance is then averaged across all spatial locations. The final LPIPS score $d(x_r, x_p)$ is a weighted sum of these layer-wise distances:

$$d(x_r, x_p) = \sum_{l=1}^{L} w_l \cdot \frac{1}{H_l W_l} \sum_{h,w} \big\| \hat{h}_r^l(h,w) - \hat{h}_p^l(h,w) \big\|_2^2$$

The small weights w_l are themselves learned to best correlate with human perceptual judgments. A lower LPIPS score d signifies a higher perceptual similarity between the two images.

4 Methods

4.1 Problem Formulation

Modern autonomous agentic AI systems increasingly rely on multimodal models that integrate vision and language to make decisions with minimal human oversight. These systems are deployed in real-world applications such as e-commerce, navigation agents, and booking platforms, where the selected image directly triggers downstream actions [Davydova et al., 2025, Wang et al., 2024c], such as clicks, follow-up queries, or further reasoning steps, making this vision-language selection layer a key target for influencing agentic behavior [Li et al., 2025, Zhu et al., 2024].

Formally, we consider an agent driven by a multimodal model M. Consistent with standard contrastive retrieval architectures [Radford et al., 2021], we assume M contains a text encoder E_T and a vision encoder E_V. The agent receives a text prompt p and a set of n candidate images $\{x_i\}_{i=1}^n$. The model computes an embedding for the prompt $e_{\text{text}} = E_T(p)$, and for each image $e_{\text{image}}(x_i) = E_V(x_i)$. The model selects the image x_i that maximizes an internal relevance score of f_M, defined as their cosine similarity:

$$M\big(p, \{x_i\}_{i=1}^n\big) = \arg\max_{i\in\{1,\dots,n\}} f_M(p, x_i), \qquad f_M(p, x_i) = \cos\big(e_{\text{text}}, e_{\text{image}}(x_i)\big)$$

We consider an attacker whose objective is to force the agent to select a specific target image. The attacker controls a single index t, i.e., a target image $x_{\text{target}} = x_t$, and can replace it with an adversarially modified version x_{adv}. The remaining $n-1$ images, which form the competitor set $\{x_{\text{comp}}^{(i)}\}_{i=1}^{n-1} = \{x_i\}_{i\neq t}$, cannot be modified by the attacker. The attack goal is to produce x_{adv} such that the agent selects it. Equivalently, we seek to increase the selection probability, i.e.:

$$M(p, \{x_{\text{adv}}\} \cup \{x_{\text{comp}}^{(i)}\}_{i=1}^{n-1}) = x_{\text{adv}}, \qquad \Pr\big[f_M(p, x_{\text{adv}}) > \max_{x\in\{x_{\text{comp}}^{(i)}\}_{i=1}^{n-1}} f_M(p, x)\big]. \quad (1)$$

This optimization is subject to the constraint that x_{adv} remains perceptually similar to the original image x_{target}. Formally, we require $d(x_{\text{adv}}, x_{\text{target}}) \leq \epsilon$, where d is a perceptual distance metric (e.g., LPIPS) and ϵ is a small perceptual budget. This constraint is necessary for the attack to be viable in a real-world system. An unconstrained attack, where an attacker simply generates a new image x_{adv} to perfectly match the prompt p, would be trivially defeated. Such an image would fail basic platform-level integrity checks, such as visual hash-based deduplication or anomaly detection [Hao et al., 2021, Wu et al., 2023, Zhou et al., 2018], which would flag the large semantic gap between the original and modified content as a fraudulent replacement, not a subtle perturbation. The perceptual constraint $d(x_{\text{adv}}, x_{\text{target}}) \leq \epsilon$ is explicitly designed to bypass these defenses by ensuring x_{adv} is perceived as a benign modification of x_{target}.

This attack is performed under a realistic black-box threat model as the attacker has no access to the model's weights, parameters, or gradients. Their only capability is to query the model M and observe the final selection, mimicking an attacker who can only interact with the agent's public application. By systematically probing the vision-language selection layer of the agentic systems, we expose and characterize vulnerabilities in multimodal agentic systems that could be exploited to threaten the reliability and fairness of downstream decision-making.

4.2 TRAP Framework

To expose the susceptibility of multimodal agents to this threat model, we propose TRAP, a novel black-box optimization framework that modifies only the target image to induce consistent selection by AI agents. TRAP diverges from traditional pixel-level perturbations by operating in CLIP's latent space rather than directly modifying image pixels. This allows us to steer high-level semantics in a model-agnostic manner using a surrogate representation aligned with vision-language reasoning. This choice is motivated by the increasing robustness of modern systems to low-level noise and the limitations of existing pixel-based attacks in black-box, semantic decision settings [Yang et al., 2022, Li et al., 2023, Goodfellow et al., 2015, Madry et al., 2019].

TRAP takes as input a target image x_{target}, competitor images $\{x_{\text{comp}}^{(i)}\}_{i=1}^{n-1}$, and an attacker-chosen guidance prompt, p_{pos}, which describes the high-level concepts to be injected. It is critical to distinguish p_{pos} from the unknown, user-provided prompt p that the agent receives at inference time. The threat model does not assume knowledge of the exact p. Instead, the attacker chooses a p_{pos} to act as a strong semantic proxy for a family of desirable user queries (e.g., using $p_{\text{pos}} =$ "steel-toe reinforced" to capture searches for both "durable boot" and "safe boot"). TRAP uses this p_{pos} as the optimization target for its semantic alignment loss ($\mathcal{L}_{\text{sem}}$), taking advantage of the shared embedding space to ensure that optimizing for p_{pos} generalizes to increase the selection probability for semantically similar p queries. We pre-compute the embedding for this prompt as $e_{\text{pos}} = E_T(p_{\text{pos}})$.

The framework iteratively optimizes the image latent embedding e_{adv}, initialized from $E_V(x_{\text{target}})$, to maximize its alignment with the guidance prompt's embedding, e_{pos}. This approach is effective because most modern multimodal models (e.g., CLIP, ALIGN, BLIP) rank images by their similarity to the prompt in the shared embedding space; a higher alignment directly translates to a higher likelihood of selection.

While the precise architecture of the agent may be unknown, prior work demonstrates that adversarial examples crafted in CLIP space are transferable to other vision-language models due to shared embedding geometries and training objectives [Huang et al., 2025]. We therefore optimize e_{adv} using a composite objective, and the final optimized embedding is decoded into the adversarial image x_{adv}. The following subsections detail each component of TRAP, and the complete process is summarized in Algorithm 1.

4.3 Guided Embedding Optimization in CLIP Space

Our goal is to find an optimal adversarial embedding e_{adv}^* by minimizing a composite objective. The optimization is as such:

$$e_{\text{adv}}^* = \arg\min_{e_{\text{adv}}} \mathcal{L}_{\text{total}}(e_{\text{adv}}) \tag{2}$$

The total loss $\mathcal{L}_{\text{total}}$ depends on e_{adv} as well as several other pre-computed constants. Since e_{adv} is the only variable, we optimize this function via gradient descent. The objective is a weighted sum of three components where $\lambda_1, \lambda_2, \lambda_3$ are scalar hyperparameters:

$$\mathcal{L}_{\text{total}}(\cdot) = \lambda_1 \mathcal{L}_{\text{sem}}(\cdot) + \lambda_2 \mathcal{L}_{\text{dist}}(\cdot) + \lambda_3 \mathcal{L}_{\text{LPIPS}}(\cdot) \tag{3}$$

Semantic Alignment Loss ($\mathcal{L}_{sem}$). First, to influence the model's selection behavior in favor of x_{adv}, we leverage CLIP's joint image-text embedding space, where semantically related inputs are embedded nearby. By minimizing the cosine distance between e_{adv} and the ℓ_2-normalized positive prompt embedding e_{pos}, we inject high-level semantic meaning directly into the image representation:

$$\mathcal{L}_{\text{sem}}(e_{\text{adv}}, e_{\text{pos}}) = 1 - \cos(e_{\text{adv}}, e_{\text{pos}})$$

Distinctive Feature Preservation Loss ($\mathcal{L}_{dist}$). Minimizing $\mathcal{L}_{\text{sem}}$ alone could cause e_{adv} to lose the image's unique identity. To prevent this, we introduce a Siamese semantic network S_{dist}. This network (e.g., a two-branch MLP) is designed to decompose a given embedding $e \in \mathbb{R}^d$ into two components: a common embedding $e_{\text{com}} \in \mathbb{R}^{d_c}$ and a distinctive embedding $e_{\text{dist}} \in \mathbb{R}^{d_d}$, where d_c and d_d are the dimensionalities of these two output feature spaces set within the network configurations as hyperparameters.

$$S_{\text{dist}}(e) = (e_{\text{com}}, e_{\text{dist}})$$

This loss penalizes the ℓ_2 distance between the distinctive component of e_{adv} and the distinctive component of the original e_{target}.

$$\mathcal{L}_{\text{dist}}(e_{\text{adv}}, e_{\text{target}}^{(\text{dist})}) = \|S_{\text{dist}}(e_{\text{adv}})[1] - e_{\text{target}}^{(\text{dist})}\|_2^2$$

Here, $S_{\text{adv}}(e_{\text{target}})[1]$ denotes the distinctive component of S_{dist} (the second output of S_{dist}) applied to the current e_{adv}. The term $e_{\text{target}}^{(\text{dist})}$ is a pre-computed constant, $e_{\text{target}}^{(\text{dist})} = S_{\text{dist}}(E_V(x_{\text{target}}))[1]$. This constraint ensures that adversarial edits preserve identity-relevant features not captured by semantic alignment alone. Without it, optimization may overfit to prompt content, collapsing diverse inputs into visually indistinct representations (e.g., all "apple" images becoming generic red blobs). As demonstrated in prior work on multimodal attacks [Zhang et al., 2024a, Chen et al., 2025], targeting both shared and distinctive features increases adversarial effectiveness and transferability. Our loss

thus anchors the adversarial embedding to its unique identity while still allowing semantic guidance from the prompt.

This loss creates the implicit supervision for the decomposition. By penalizing any change in the distinctive branch, $\mathcal{L}_{\text{dist}}$ effectively "anchors" e_{dist} to its original value. This forces the optimizer to channel the gradients from the other two losses ($\mathcal{L}_{\text{sem}}$ and $\mathcal{L}_{\text{LPIPS}}$) almost exclusively through the common branch $S_{\text{dist}}(e_{\text{adv}})[0]$. This push-pull dynamic is what isolates the prompt-driven semantic changes to e_{com}, while e_{dist} preserves the image's unique identity.

Perceptual Similarity Loss ($\mathcal{L}_{LPIPS}$). Finally, to ensure the decoded image remains visually plausible, we apply a loss in pixel space. This requires a differentiable process to decode e_{adv} into a candidate image x_{cand} at each optimization step. This decoding pipeline itself has two parts.

First, we generate a semantic layout mask $A \in \mathbb{R}^{H \times W}$ using a lightweight MLP encoder-decoder L to identify regions of interest:

$$A = L_{\text{dec}}\big(L_{\text{enc}}([e_{\text{pos}}, E_V(x_{\text{target}})])\big) \tag{4}$$

This initial mask A is a "soft" heatmap indicating semantic relevance to the prompt, but it may be spatially imprecise. To improve localization and ensure edits are restricted to the primary subject, we refine A using a pre-computed binary foreground mask, F_{seg}, obtained from a DeepLabv3 segmentation model [Chen et al., 2017a]. The final mask used for modulation is the element-wise product:

$$A_{\text{final}} = A \odot F_{\text{seg}} \tag{5}$$

Second, at each optimization step, we extract the common component from our trainable embedding, $e_{\text{com}} = S_{\text{dist}}(e_{\text{adv}})[0]$, and modulate it with the fixed, refined mask A_{final} to get $e_{\text{mod}} = e_{\text{com}} \odot A_{\text{final}}$. This e_{mod} is passed to a differentiable image decoder SD (e.g., the VAE decoder from Stable Diffusion), conditioned on p_{pos}, to produce the candidate image $x_{\text{cand}} = SD(e_{\text{mod}}, p_{\text{pos}})$.

The LPIPS loss is the perceptual distance between this decoded candidate x_{cand} and the original, unmodified target image x_{target}:

$$\mathcal{L}_{\text{LPIPS}}(x_{\text{cand}}, x_{\text{target}}) = \text{LPIPS}(x_{\text{cand}}, x_{\text{target}})$$

By tracing the dependencies, x_{cand} is a function of e_{adv}, A_{final}, and p_{pos}. Since A_{final} and p_{pos} are pre-computed constants, $\mathcal{L}_{\text{LPIPS}}$ is an implicit function of e_{adv}. This creates a fully differentiable path from the pixel-space comparison back to our latent variable e_{adv}.

Overall framework The full optimization and final decoding process is summarized as:

$$\begin{aligned} e^*_{\text{adv}} &= \arg\min_{e_{\text{adv}}} \big[\lambda_1 \mathcal{L}_{\text{sem}} + \lambda_2 \mathcal{L}_{\text{dist}} + \lambda_3 \mathcal{L}_{\text{LPIPS}}\big], \\ A = L_{\text{dec}}(L_{\text{enc}}([e_{\text{pos}}, E_V(x_{\text{target}})])), &\qquad A_{\text{final}} = A \odot F_{\text{seg}}, \\ e_{\text{com}} = S_{\text{dist}}(e_{\text{adv}})[0], x_{\text{adv}} &= SD(e_{\text{com}} \odot A_{\text{final}}, p_{\text{pos}}). \end{aligned} \tag{6}$$

with each step guided by semantic alignment, visual coherence, and layout-informed embeddings. Algorithm 1 in the Appendix summarizes the optimization process for generating an adversarial image given a target, a prompt, and a black-box agent model.

5 Experimental Methodology

5.1 Experimental Protocol

We evaluate our attack on 100 image-caption pairs from the popular COCO Captions dataset [Chen et al., 2015], simulating a black-box n-way selection setting. For each instance, a "bad image" is generated using a negative prompt created via Llama-3-8B [Grattafiori et al., 2024]. This image is verified to have an initial selection probability below the majority threshold when compared against $n-1$ competitors, ensuring a challenging starting point for optimization.

Adversarial optimization then runs for up to $K = 20$ outer iterations, each containing $T = 20$ inner gradient-based steps, stopping early if the success condition is met. To evaluate the final optimized image x_{adv}, we conduct $R = 100$ randomized trials. In each trial, x_{adv} and the $n-1$ competitors are randomly ordered to mitigate positional bias [Tian et al., 2025], horizontally concatenated into

I_{concat}, and queried to the agent model M. The selection probability $P(x_{\text{adv}})$ is the fraction of these R trials where x_{adv} was chosen:

$$P(x_{\text{adv}}) = \frac{1}{R} \sum_{r=1}^{R} \mathbf{1}[M(I_{concat}) = x_{\text{adv}}]$$

The overall Attack Success Rate (ASR) is the percentage of the 100 COCO instances where this optimized image successfully crosses the majority threshold, i.e., $P(x_{\text{adv}}) > 1/n$.

5.2 Model and Implementation Details

All experiments are implemented in PyTorch. We use CLIP ViT-B/32 [Radford et al., 2021] for embedding extraction, with adversarial image decoding performed by Stable Diffusion v2.1 (base) through the Img2Img interface. The optimized image embedding is repeated across 77 tokens and injected as prompt embeddings into the UNet decoder.

The Siamese Semantic Network consists of two branches, each with two linear layers (512→1024), BatchNorm, and ReLU, trained to decompose CLIP image embeddings into common and distinctive features. The Layout Generator receives concatenated image and text embeddings (1536 dimensions), processes them via an encoder (linear layers: 1536→512→1024, ReLU), reshapes to (256, 2, 2), then upsamples with five transposed convolutional layers with ReLU and a final Sigmoid to generate a spatial mask $A \in \mathbb{R}^{H \times W}$, which is refined with DeepLabv3 segmentation to emphasize foreground. Optimization is performed with Adam (learning rate 0.005, 20 steps per iteration). Grid search is conducted over diffusion strength $[0.3, 0.8]$ and CFG $[2.0, 12.0]$ with initial values of 0.5 and 7.5, respectively. All experiments were run on a server with four NVIDIA A100-PCIE-40GB GPUs and a 48-core Intel Xeon Silver 4214R CPU. Average per-iteration optimization is around 520 seconds compared to around 376 seconds for SPSA and 110 seconds for Bandit.

6 Experimental Results

6.1 Main Findings

TRAP achieves the highest attack success rate (ASR) across all evaluated models, LLaVA-1.5 [Liu et al., 2023a], Gemma3 [Mesnard et al., 2025], Mistral-small [Mistral AI, 2025], GPT-4o [OpenAI et al., 2024], and CogVLM [Wang et al., 2024a]. As shown in Table 1, TRAP universally succeeds even from a challenging low-preference baseline (0–21% ASR), while traditional baselines like SPSA [Spall, 1987] and Bandit [Ilyas et al., 2018b] are ineffective (max 36% ASR). We also compare against recent embedding-space attacks: SSA_CWA [Chen et al., 2024], which evaluates robustness using embedding-based attacks, and SA_AET [Jia et al., 2025], which generates adversarial images by projecting text embeddings onto image embeddings. While these embedding-based methods perform better than traditional approaches, TRAP significantly outperforms all baselines. The attack's transferability is a key finding. It demonstrates high efficacy not only on expected contrastive models but also on CogVLM, which uses a non-contrastive architecture. Additionally, the attack transfers with high success to GPT-4o, a completely closed-source, black-box proprietary model.

Table 1: Comparison of adversarial attack effectiveness across evaluated methods and models.

Method	**LLaVA 1.5-34B**	**Gemma 3-8B**	**Mistral-small-3.1-24B**	**Mistral-small-3.2-24B**	**GPT-4o**	**CogVLM**
Initial "bad image"	21%	17%	14%	6%	0%	8%
SPSA	36%	27%	22%	11%	1%	18%
Bandit	6%	2%	1%	0%	0%	0%
Stable Diffusion	24%	18%	18%	7%	0%	20%
SSA_CWA	65%	42%	28%	18%	8%	4%
SA_AET	85%	67%	61%	55%	12%	42%
TRAP	**100%**	**100%**	**100%**	**99%**	**63%**	**94%**

We further assess TRAP against standard defenses and an adversarially trained LLaVA variant (Robust-LLaVA). As summarized in Table 2, TRAP remains highly effective. Applying addi-

tive Gaussian noise to the generated adversarial images has a negligible effect on ASR (TRAP + Gaussian Noise). Caption-level filters applied post-attack, such as CIDER [Xu et al., 2024] and MirrorCheck [Fares et al., 2024], reduce ASR but do not eliminate the attack's success, particularly on the robust model. Figure 3 provides qualitative examples of successful attacks generated by TRAP.

Table 2: Robustness of TRAP Attack Under Various Defense Mechanisms and Adversarial Training.

COCO	**LLaVA-1.5-34B**	**Gemma3-8B**	**Mistral-small-3.1-24B**	**Mistral-small-3.2-24B**	**Robust-LLaVA**
TRAP	100%	100%	100%	97%	92%
TRAP + Gaussian Noise	100%	100%	100%	96%	92%
TRAP + CIDER	100%	100%	96%	90%	85%
TRAP + MirrorCheck	100%	98%	88%	82%	74%

6.2 Robustness to System Prompt and Temperature

We further evaluate robustness to system-prompt phrasing and sampling temperature. Table 3 reports ΔASR when using four different rephrased system prompt variants compared to the baseline ASR achieved with the standard system prompt used in our main experiments, with shifts confined to low single digits and averages near zero, indicating strong generalization under rewording. Overall, as long as the instruction semantics are preserved, TRAP remains stable to superficial template changes. Additional ablation study can be found in Appendix A.

Table 3: Impact of system prompt variations on attack success rate (ASR). ΔASR is the average deviation from baseline.

Model	Variant 1	Variant 2	Variant 3	Variant 4	**Avg. ΔASR**
LLaVA-1.5-34B	+2%	−1%	+4%	+1%	+2%
Gemma3-8B	−2%	+1%	−3%	−1%	−1%
Mistral-small-3.1-24B	+1%	+2%	−1%	+0%	+1%

Figure 2 plots the ASR against the required margin over the majority choice threshold ($1/n$), comparing performance at two distinct decoding temperatures: $T = 0.1$ (left plot, representing near-deterministic output) and $T = 0.7$ (right plot, representing more stochastic output). Across both temperature settings, the optimized TRAP attack (solid lines) consistently maintains a high ASR, significantly outperforming the unoptimized baseline (dashed lines) across all evaluated models. The ASR curves for the optimized attack show minimal variation between the low-temperature ($T = 0.1$) and high-temperature ($T = 0.7$) settings, confirming that the attack's effectiveness is robust to changes in the agent's sampling temperature and ensuring reliability under both deterministic and stochastic generation conditions.

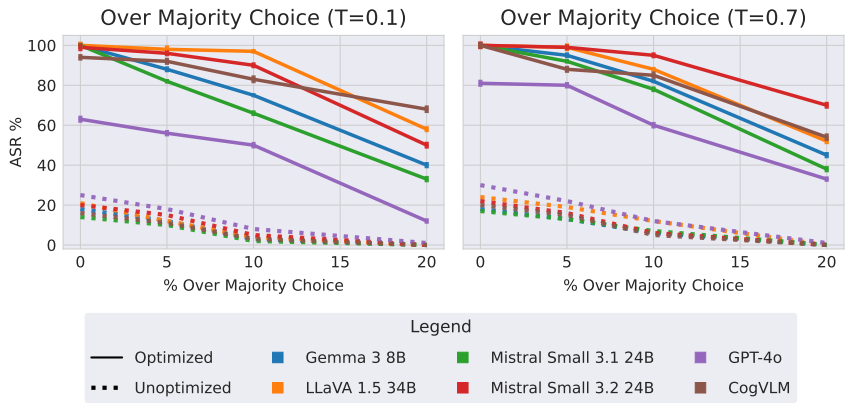

Figure 2: Attack success rate under different sampling temperatures.

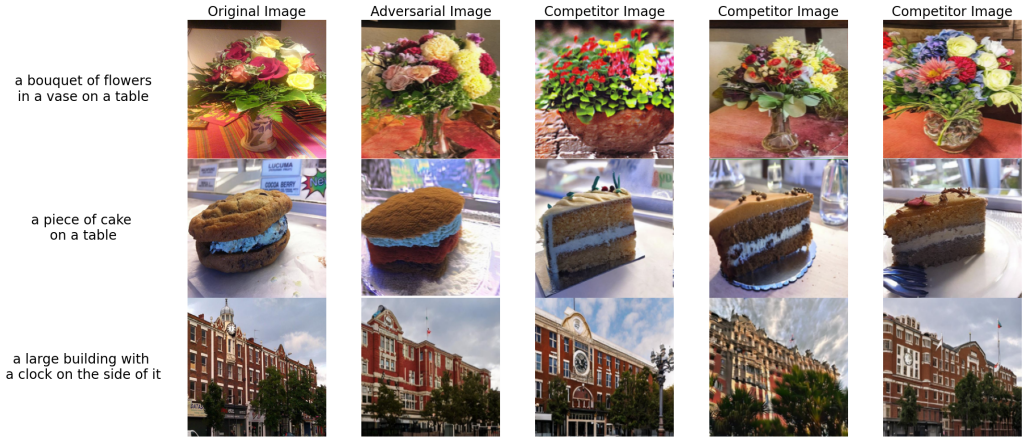

Figure 3: Qualitative examples of successful attacks. Each row shows a user-facing scenario where the attacker modifies a target image (left) into an adversarial variant (second column), evaluated against three unmodified competitors (right).

7 Discussion

We expose a critical vulnerability in agentic AI: visually subtle, semantically guided attacks reliably mislead VLMs even under black-box constraints. TRAP consistently induces decision-level preference redirection across all evaluated MLLMs, far exceeding traditional pixel-based and standard diffusion baselines. This expands the attack surface in real-world multimodal agents.

Key takeaways: (1) semantic attacks are effective and transferable; (2) semantic attacks are robust to prompt and sampling noise and remain visually plausible; (3) the vulnerability generalizes across MLLMs; (4) existing defenses overlook this threat class.

However, the broader significance lies in what such attacks enable. By adversarially altering an image to match a high-level semantic concept, attackers could manipulate agentic behavior in downstream tasks, causing selection of malicious UI elements, misleading product recommendations, hijacked retrieval in chat agents, or sabotage of autonomous perception pipelines. More importantly, these edits remain visually natural and can be deployed in black-box settings, making them difficult to detect or attribute compared to previous methods. This work challenges the assumption that robustness can be measured solely through pixel-space perturbations, emphasizing the need for embedding-level defenses and semantic-level robustness criteria.

8 Limitations

While TRAP demonstrates strong performance, several limitations must be acknowledged. We assume that the agent relies on contrastive vision-language similarity, an assumption supported by current architectures but potentially less valid in future systems that move completely away from contrastive reasoning or incorporate stronger semantic defenses than the one tested. The success of our method also depends on the quality of auxiliary components such as the layout mask and diffusion model; performance may degrade on edge cases or under constrained resources. Finally, TRAP is more computationally intensive than pixel-level attacks, due to its reliance on iterative optimization and generative decoding. While this cost can be amortized in offline scenarios or reduced via model distillation, scalability to real-time applications remains an open challenge.

Acknowledgements

We thank the anonymous reviewers for their thoughtful and constructive feedback. This work was supported by funding through NSF Grants No. CCF-2238079, CCF-2316233, CNS-2148583 anda Research Gift from Amazon AGI Labs.

References

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. In *Advances in Neural Information Processing Systems*, volume 35, pages 23716–23736, 2022.

Maksym Andriushchenko, Francesco Croce, Nicolas Flammarion, and Matthias Hein. Square attack: a query-efficient black-box adversarial attack via random search, 2020. URL https://arxiv.org/abs/1912.00049.

Wieland Brendel, Jonas Rauber, and Matthias Bethge. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. In *International Conference on Learning Representations*, 2018. URL https://arxiv.org/abs/1712.04248.

Tom B Brown, Dandelion Mané, Aurko Roy, Martín Abadi, and Justin Gilmer. Adversarial patch. *arXiv preprint arXiv:1712.09665*, 2017. URL https://arxiv.org/abs/1712.09665.

Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. *arXiv preprint arXiv:1608.04644*, 2016. URL https://arxiv.org/abs/1608.04644.

Krishna Chaitanya, Ertunc Erdil, Neerav Karani, and Ender Konukoglu. Contrastive learning of global and local features for medical image segmentation with limited annotations, 2020. URL https://arxiv.org/abs/2006.10511.

Huanran Chen, Yichi Zhang, Yinpeng Dong, Xiao Yang, Hang Su, and Jun Zhu. Rethinking model ensemble in transfer-based adversarial attacks. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=AcJrSoArlh.

L. Chen, Y. Chen, Z. Ouyang, Y. Zhang, H. Deng, and H. Sun. Boosting adversarial transferability in vision-language models via multimodal feature heterogeneity. *Scientific Reports*, 15:7366, 2025. doi: 10.1038/s41598-025-91802-6. URL https://www.nature.com/articles/s41598-025-91802-6.

Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation, 2017a. URL https://arxiv.org/abs/1706.05587.

Pin-Yu Chen, Huan Zhang, Yash Sharma, Jinfeng Yi, and Cho-Jui Hsieh. Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, pages 15–26, 2017b. URL https://arxiv.org/abs/1708.03999.

Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*, 2015. URL https://arxiv.org/abs/1504.00325.

Xuelong Dai, Kaisheng Liang, and Bin Xiao. Advdiff: Generating unrestricted adversarial examples using diffusion models, 2024. URL https://arxiv.org/abs/2307.12499.

Mariya Davydova et al. Osuniverse: Benchmark for multimodal gui-navigation ai agents. *arXiv preprint arXiv:2505.03570*, 2025. URL https://arxiv.org/abs/2505.03570.

Samar Fares, Klea Ziu, Toluwani Aremu, Nikita Durasov, Martin Takáč, Pascal Fua, Karthik Nandakumar, and Ivan Laptev. Mirrorcheck: Efficient adversarial defense for vision-language models, 2024. URL https://arxiv.org/abs/2406.09250.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples, 2015. URL https://arxiv.org/abs/1412.6572.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, and Alan Schelten et al. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Qingying Hao, Licheng Luo, Steve T.K. Jan, and Gang Wang. It's not what it looks like: Manipulating perceptual hashing based applications. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, CCS '21, page 69–85, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450384544. doi: 10.1145/3460120.3484559. URL https://doi.org/10.1145/3460120.3484559.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance, 2022. URL https://arxiv.org/abs/2207.12598.

Hanxun Huang, Sarah Erfani, Yige Li, Xingjun Ma, and James Bailey. X-transfer attacks: Towards super transferable adversarial attacks on clip, 2025. URL https://arxiv.org/abs/2505.05528.

Andrew Ilyas, Logan Engstrom, Anish Athalye, and Jessy Lin. Black-box adversarial attacks with limited queries and information. *arXiv preprint arXiv:1804.08598*, 2018a. URL https://arxiv.org/abs/1804.08598.

Andrew Ilyas, Logan Engstrom, Anish Athalye, and Jessy Lin. Bandits for black-box adversarial attacks. *arXiv preprint arXiv:1807.07978*, 2018b. URL https://arxiv.org/abs/1807.07978.

Xiaojun Jia, Sensen Gao, Qing Guo, Simeng Qin, Ke Ma, Yihao Huang, Yang Liu, Ivor W. Tsang, and Xiaochun Cao. Semantic-aligned adversarial evolution triangle for high-transferability vision-language attack. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(10): 8489–8505, 2025. doi: 10.1109/TPAMI.2025.3581476.

Andreas Koukounas, Georgios Mastrapas, Bo Wang, Mohammad Kalim Akram, Sedigheh Eslami, Michael Günther, Isabelle Mohr, Saba Sturua, Scott Martens, Nan Wang, and Han Xiao. jina-clip-v2: Multilingual multimodal embeddings for text and images, 2024. URL https://arxiv.org/abs/2412.08802.

Chae Eun Lee, Minyoung Chung, and Yeong-Gil Shin. Voxel-level siamese representation learning for abdominal multi-organ segmentation, 2021. URL https://arxiv.org/abs/2105.07672.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven CH Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International Conference on Machine Learning*, pages 12888–12900. PMLR, 2022a.

Xiangyu Li, Xu Yang, Kun Wei, Cheng Deng, and Muli Yang. Siamese contrastive embedding network for compositional zero-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15603–15612, 2022b. URL https://openaccess.thecvf.com/content/CVPR2022/papers/Li_Siamese_Contrastive_Embedding_Network_for_Compositional_Zero-Shot_Learning_CVPR_2022_paper.pdf.

Y. Li et al. Vision-language-action models: Concepts, progress, applications and challenges. *arXiv preprint arXiv:2505.04769*, 2025. URL https://arxiv.org/abs/2505.04769.

Yifan Li, Yujie Zhang, Yuzhuo Wang, et al. Noise-resistant multimodal transformer for emotion recognition. *arXiv preprint arXiv:2305.02814*, 2023.

Zhiyuan Li, Heng Wang, Dongnan Liu, Chaoyi Zhang, Ao Ma, Jieting Long, and Weidong Cai. Multimodal causal reasoning benchmark: Challenging vision large language models to infer causal links between siamese images. *arXiv preprint arXiv:2408.08105*, 2024.

Zeyi Liao, Lingbo Mo, Chejian Xu, Mintong Kang, Jiawei Zhang, Chaowei Xiao, Yuan Tian, Bo Li, and Huan Sun. Eia: Environmental injection attack on generalist web agents for privacy leakage, 2025. URL https://arxiv.org/abs/2409.11295.

Haotian Liu, Yuhui Zhang, Chunyuan Li, and Jianfeng Gao. Improved baselines with visual instruction tuning. *arXiv preprint arXiv:2310.03744*, 2023a. URL https://arxiv.org/abs/2310.03744.

Jiang Liu, Chen Wei, Yuxiang Guo, Heng Yu, Alan Yuille, Soheil Feizi, Chun Pong Lau, and Rama Chellappa. Instruct2attack: Language-guided semantic adversarial attacks, 2023b. URL https://arxiv.org/abs/2311.15551.

Zhiyue Liu, Jinyuan Liu, and Fanrong Ma. Improving cross-modal alignment with synthetic pairs for text-only image captioning, 2023c. URL https://arxiv.org/abs/2312.08865.

Yue Lu, Chao Guo, Xingyuan Dai, and Fei-Yue Wang. Artcap: A dataset for image captioning of fine art paintings. *IEEE Transactions on Computational Social Systems*, 11(1):576–587, 2024. doi: 10.1109/TCSS.2022.3223539.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks, 2019. URL https://arxiv.org/abs/1706.06083.

Thomas Mesnard et al. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025. URL https://arxiv.org/abs/2503.19786.

Mistral AI. Mistral small 3.1. https://mistral.ai/news/mistral-small-3-1, 2025. Accessed: 2025-05-09.

Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2574–2582, 2016a. URL https://arxiv.org/abs/1511.04599.

Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks, 2016b. URL https://arxiv.org/abs/1511.04599.

Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Universal adversarial perturbations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1765–1773, 2017. URL https://arxiv.org/abs/1610.08401.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo

Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. Gpt-4 technical report, 2024. URL https://arxiv.org/abs/2303.08774.

Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis, 2023. URL https://arxiv.org/abs/2307.01952.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021. URL https://arxiv.org/abs/2103.00020.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10684–10695, June 2022a.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2022b. URL https://arxiv.org/abs/2112.10752.

James C. Spall. A stochastic approximation technique for generating maximum likelihood parameter estimates. In *Proceedings of the American Control Conference*, pages 1161–1167, Minneapolis, MN, 1987.

Xinyu Tian, Shu Zou, Zhaoyuan Yang, and Jing Zhang. Identifying and mitigating position bias of multi-image vision-language models, 2025. URL https://arxiv.org/abs/2503.13792.

Jonathan Uesato, Brendan O'Donoghue, Aaron van den Oord, and Pushmeet Kohli. Adversarial risk and the dangers of evaluating against weak attacks, 2018. URL https://arxiv.org/abs/1802.05666.

Chenan Wang, Jinhao Duan, Chaowei Xiao, Edward Kim, Matthew Stamm, and Kaidi Xu. Semantic adversarial attacks via diffusion models, 2023. URL https://arxiv.org/abs/2309.07398.

Weihan Wang, Qingsong Lv, Wenmeng Yu, Wenyi Hong, Ji Qi, Yan Wang, Junhui Ji, Zhuoyi Yang, Lei Zhao, Xixuan Song, Jiazheng Xu, Bin Xu, Juanzi Li, Yuxiao Dong, Ming Ding, and Jie Tang. Cogvlm: Visual expert for pretrained language models, 2024a. URL https://arxiv.org/abs/2311.03079.

Yifei Wang, Dizhan Xue, Shengjie Zhang, and Shengsheng Qian. Badagent: Inserting and activating backdoor attacks in llm agents, 2024b. URL https://arxiv.org/abs/2406.03007.

Yufei Wang et al. Seeact: A strong vision-language model for visually grounded action. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024c. URL https://arxiv.org/abs/2403.09691.

Fangzhou Wu, Shutong Wu, Yulong Cao, and Chaowei Xiao. Wipi: A new web threat for llm-driven web agents, 2024. URL https://arxiv.org/abs/2402.16965.

Ke Wu, Zichi Wang, Xinpeng Zhang, and Zhenjun Tang. Defending against adversarial examples using perceptual image hashing. *Journal of Electronic Imaging*, 32:023016, March 2023. doi: 10.1117/1.JEI.32.2.023016.

Yue Xu, Xiuyuan Qi, Zhan Qin, and Wenjie Wang. Cross-modality information check for detecting jailbreaking in multimodal large language models, 2024. URL https://arxiv.org/abs/2407.21659.

Wenkai Yang, Xiaohan Bi, Yankai Lin, Sishuo Chen, Jie Zhou, and Xu Sun. Watch out for your agents! investigating backdoor threats to llm-based agents, 2024. URL https://arxiv.org/abs/2402.11208.

Ziyang Yang, Yichao Xu, Yuxuan Wang, et al. Robust-msa: Understanding the impact of modality noise on multimodal sentiment analysis. *arXiv preprint arXiv:2211.13484*, 2022.

Shengfang Zhai, Yinpeng Dong, Qingni Shen, Shi Pu, Yuejian Fang, and Hang Su. Text-to-image diffusion models can be easily backdoored through multimodal data poisoning, 2023a. URL https://arxiv.org/abs/2305.04175.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. *arXiv preprint arXiv:2303.15343*, 2023b.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric, 2018. URL https://arxiv.org/abs/1801.03924.

Tingwei Zhang, Rishi Jha, Eugene Bagdasaryan, and Vitaly Shmatikov. Adversarial illusions in multi-modal embeddings, 2024a. URL https://arxiv.org/abs/2308.11804.

Yanzhe Zhang, Tao Yu, and Diyi Yang. Attacking vision-language computer agents via pop-ups. *arXiv preprint arXiv:2411.02391*, 2024b.

Peng Zhou, Xintong Han, Vlad I. Morariu, and Larry S. Davis. Learning rich features for image manipulation detection, 2018. URL https://arxiv.org/abs/1805.04953.

Ziqi Zhou, Shengshan Hu, Minghui Li, Hangtao Zhang, Yechao Zhang, and Hai Jin. Advclip: Downstream-agnostic adversarial examples in multimodal contrastive learning, 2023. URL https://arxiv.org/abs/2308.07026.

Yujia Zhu et al. Vlmbench: Evaluating llms as agents in the wild. *arXiv preprint arXiv:2403.08191*, 2024. URL https://arxiv.org/abs/2403.08191.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction precisely state the contributions, introducing TRAP, describing its generative adversarial approach, reporting attack success rates, and highlighting the security implications. All claims are directly supported by experimental results (Sections 1, 6).

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Section 8 directly discusses the limitations, including generalizability to real-world deployments, assumptions about agent model architectures, computational cost, and dependency on quality of auxiliary components.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: The paper does not introduce formal theoretical results or proofs, focusing instead on empirical methodology and algorithmic design.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Section 5 provides comprehensive details on datasets, model architectures, optimization procedures, experimental protocols, and evaluation metrics, enabling reproduction of key results.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in

some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We are including instructions and algorithms sufficient to reproduce our results to the supplementary material. A link to the Github has also been added to the camera-ready version of the abstract.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: Section 5.2 and Section 5.1 detail all training/test splits, hyperparameters (including optimizer and settings), model architectures, and evaluation protocol.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Error bars are present in the figures in Section 6 and details surrounding the experiments are given.

   Guidelines:

   - The answer NA means that the paper does not include experiments.

- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Section 5.2 lists the hardware used, reports average per-iteration runtime, and compares compute times to baseline methods.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The paper aligns with the NeurIPS Code of Ethics, using only open, licensed datasets and models, addresses both technical and societal implications, and preserves anonymity.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

Justification: Section 7 extensively discusses the broader implications, including security risks of agentic AI, calls for improved defenses, and possible real-world harms.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: Our work does not release any new models, datasets, or pretrained components that pose a risk of misuse. Our contributions are limited to a black-box evaluation framework and do not introduce new generative capabilities or data sources that would require additional safeguards.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All datasets, models and baselines are properly cited in the references; license information is standard for these widely-used assets.

    Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: The paper does not introduce new datasets or pre-trained models; all empirical work is based on existing assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The research does not involve any crowdsourcing or human subject studies.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

Justification: No human subjects or participant risks are involved in this research.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: Section 5.1 describes the use of LLMs (Llama-3-8B) to generate negative prompts for constructing challenging adversarial examples.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

A Appendix

A.1 Algorithm

Algorithm 1 TRAP Framework

Require: Target image x_{target}, guidance prompt p_{pos}, competitor set $\{x_{\text{comp}}^{(i)}\}_{i=1}^{n-1}$, black-box agent M
Require: Encoders E_V, E_T; Decoder SD; Siamese Network S_{dist}, L; Seg. model F_{model}
Require: Hyperparameters $\lambda_1, \lambda_2, \lambda_3$; Loops K, T; Evals R
Ensure: Optimized adversarial image x_{adv}

Initialize $best\_score \leftarrow 0$, $n \leftarrow |\{x_{\text{comp}}^{(i)}\}_{i=1}^{n-1}| + 1$
$e_{\text{target}} \leftarrow E_V(x_{\text{target}})$
$e_{\text{pos}} \leftarrow E_T(p_{\text{pos}})$
$e_{\text{dist}}^{(\text{target})} \leftarrow S_{\text{dist}}(e_{\text{target}})[1]$
$A \leftarrow L(e_{\text{pos}}, e_{\text{target}})$
$F_{\text{seg}} \leftarrow F_{\text{model}}(x_{\text{target}})$
$A_{\text{final}} \leftarrow A \odot F_{\text{seg}}$
$x_{\text{adv}} \leftarrow x_{\text{target}}$
for $k = 1$ to K **do**
  $e_{\text{adv}} \leftarrow e_{\text{target}}$
  for $t = 1$ to T **do**
    $(e_{\text{com}}, e_{\text{dist\_adv}}) \leftarrow S_{\text{dist}}(e_{\text{adv}})$
    $e_{\text{mod}} \leftarrow e_{\text{com}} \odot A_{\text{final}}$
    $x_{\text{cand}} \leftarrow SD(e_{\text{mod}}, p_{\text{pos}})$
    $\mathcal{L}_{\text{sem}} \leftarrow 1 - \cos(e_{\text{adv}}, e_{\text{pos}})$
    $\mathcal{L}_{\text{dist}} \leftarrow \|e_{\text{dist\_adv}} - e_{\text{dist}}^{(\text{target})}\|_2^2$
    $\mathcal{L}_{\text{LPIPS}} \leftarrow \text{LPIPS}(x_{\text{cand}}, x_{\text{target}})$
    $\mathcal{L} \leftarrow \lambda_1 \mathcal{L}_{\text{sem}} + \lambda_2 \mathcal{L}_{\text{dist}} + \lambda_3 \mathcal{L}_{\text{LPIPS}}$
    Update e_{adv} using gradient descent on $\mathcal{L}$
  end for
  $(e_{\text{com\_final}}, \_) \leftarrow S_{\text{dist}}(e_{\text{adv}})$
  $x_{\text{final\_cand}} \leftarrow SD(e_{\text{com\_final}} \odot A_{\text{final}}, p_{\text{pos}})$
  $P_{\text{select}} \leftarrow \text{EstimateProb}(M, p_{\text{pos}}, x_{\text{final\_cand}}, \{x_{\text{comp}}^{(i)}\}_{i=1}^{n-1}, R)$
  if $P_{\text{select}} > best\_score$ **then**
    $best\_score \leftarrow P_{\text{select}}$
    $x_{\text{adv}} \leftarrow x_{\text{final\_cand}}$
  end if
  if $best\_score \geq 1/n$ **then**
    break
  end if
end for
return x_{adv}

A.2 Ablation Study

A.2.1 Iterative Refinement

To provide a more intuitive understanding of our method's behavior, we present a series of qualitative examples in Figure 4. This figure visualizes the output of our iterative refinement process on three distinct image-prompt pairs, showcasing the progressive transformation of the input images over a sequence of iterations.

Figure 4 illustrates the model's ability to apply semantically-guided changes that evolve in complexity as the number of iterations increases. Each row corresponds to a specific prompt: "a piece of cake on a table" , "a large building with a clock on the side of it" , and "a bouquet of flowers in a vase on a table".

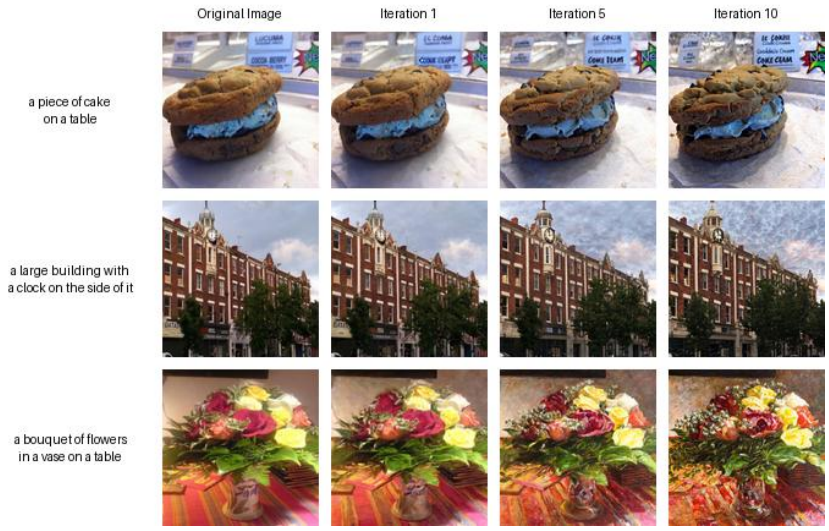

Figure 4: Qualitative results of the iterative image generation process. Each row begins with an original source image and a corresponding text prompt, followed by the generated outputs at iterations 1, 5, and 10. The examples demonstrate a range of transformations, from subtle refinement to significant stylistic alteration.

A.2.2 Evaluation on Additional Datasets

We evaluate TRAP on two additional distributions, Flickr8k_sketch and ArtCapt [Lu et al., 2024]. Across both settings and across diverse VLM backbones (LLaVA-1.5-34B, Gemma3-8B, Mistral variants, GPT-4o), TRAP consistently outperforms baseline attacks (Tables 4 and 5). This pattern suggests the objective is not tied to a single model family and transfers across architectures and data regimes.

Table 4: TRAP Attack Success Across Sketch and Abstract Image Dataset (Flickr8k_sketch).

Flickr8k_sketch	**LLaVA-1.5-34B**	**Gemma3-8B**	**Mistral-small 3.1-24B**	**Mistral-small 3.2-24B**	**GPT-4o**
SPSA	41%	33%	31%	26%	16%
Bandit	4%	3%	1%	0%	0%
Stable Diffusion (no opt.)	20%	22%	18%	11%	4%
TRAP	**100%**	**100%**	**100%**	**96%**	**72%**

Table 5: TRAP Attack Success Rates Across Artistic Image Styles (ArtCap Dataset).

ArtCap	**LLaVA-1.5-34B**	**Gemma3-8B**	**Mistral-small 3.1-24B**	**Mistral-small 3.2-24B**	**GPT-4o**
SPSA	33%	29%	20%	21%	18%
Bandit	7%	3%	0%	0%	0%
Stable Diffusion (no opt.)	25%	20%	17%	10%	2%
TRAP	**100%**	**100%**	**100%**	**95%**	**58%**

A.2.3 Variation in Hyperparameters

We vary the relative weights of the perceptual, semantic, and distinctive objectives. Increasing the perceptual and semantic terms preserves strong performance, whereas aggressively scaling the distinctive term can reduce transfer on some targets (Table 6). These trends align with the intuition

that over-emphasizing uniqueness may hurt cross-model alignment. On the other hand, decreasing the perceptual and distinctive terms maintains strong performance, while removing the semantic term showed a significant decrease in the performance (Table 7).

Table 6: Effect of Increasing Lambda Coefficients on Attack Success Rate.

	LLaVA-1.5-34B	**Gemma3-8B**	**Mistral-small-3.1-24B**	**Mistral-small-3.2-24B**
Perceptual Loss $1.0 \rightarrow 1.5$	100%	100%	100%	98%
Semantic Loss $0.5 \rightarrow 1.0$	100%	100%	94%	92%
Distinctive Loss $0.3 \rightarrow 0.8$	88%	70%	72%	65%

Table 7: Effect of Decreasing Lambda Coefficients on Attack Success Rate.

	LLaVA-1.5-34B	**Gemma3-8B**	**Mistral-small-3.1-24B**	**Mistral-small-3.2-24B**
Perceptual Loss $1.0 \rightarrow 0.8$	100%	100%	98%	94%
Semantic Loss $0.5 \rightarrow 0.0$	90%	82%	77%	70%
Distinctive Loss $0.3 \rightarrow 0.0$	100%	100%	91%	88%

A.2.4 Embedding Model Choice

Substituting a range of image embedding backbones (from ViT-B/32 to larger SigLIP-style models [Zhai et al., 2023b] and Jina-CLIP [Koukounas et al., 2024]) yields essentially the same outcome (Table 8), indicating that TRAP is not unduly sensitive to the representation family. This stability simplifies deployment since the attack does not hinge on a particular feature extractor.

Table 8: TRAP Attack Success Rates Ablation on Different Embedding Models.

Embedding model	**LLaVA-1.5-34B**	**Gemma3-8B**	**Mistral-small-3.1-24B**
ViT-B/32	100%	100%	100%
timm/ViT-SO400M-14-SigLIP-384	100%	100%	99%
jinaai/jina-clip-v2	100%	100%	97%

A.2.5 Different Diffusion Model

We also swap the image generator among popular Stable Diffusion variants. Performance remains consistent across SD-2.1 [Rombach et al., 2022a], SD-XL [Podell et al., 2023], and SD-1.5 (Table 9), suggesting the synthesis backend is not a critical factor for the attack.

A.2.6 E-commerce Webpage Scenario

In a more realistic page-level setting, TRAP maintains a sizable margin over prior strategies (Table 10). While absolute rates are lower than in curated benchmarks, the relative gains persist under stronger content controls.

Table 9: Attack Success Rates Ablation on Different Stable Diffusion Image Generators.

Generator	**LLaVA-1.5-34B**	**Gemma3-8B**	**Mistral-small-3.1-24B**
stable-diffusion-2-1	100%	100%	100%
stable-diffusion-xl-base-1.0	100%	100%	100%
stable-diffusion-v1-5	100%	100%	100%

Table 10: Attack Success Rates of TRAP and Baselines in E-commerce Webpage Scenarios.

COCO	**LLaVA-1.5-34B**	**Gemma3-8B**	**Mistral-small-3.1-24B**	**Mistral-small-3.2-24B**
SPSA	10%	7%	1%	0%
Bandit	0%	0%	0%	0%
Stable Diffusion	13%	0%	3%	0%
SSA_CWA	20%	17%	13%	12%
SA_AET	30%	**24%**	7%	3%
TRAP	**51%**	**50%**	**27%**	**17%**

A.2.7 Efficiency Considerations

Finally, we report end-to-end compute. As expected for optimization-based attacks, TRAP is more expensive per sample than lightweight heuristics (Table 11). This overhead can be mitigated with standard engineering (e.g., early stopping, caching, adaptive step sizes) and batching.

Table 11: Total Computation Time per Sample.

	Computational Time per Step (s)	**# of Steps per Iteration**	**# of Iterations**	**Total Computational Time per Sample (s)**
SPSA	0.94	20	20	376s
Bandit	0.00022	10,000	50	110s
Stable Diffusion	4.6	1	1	4.6s
TRAP	1.3	20	20	520s