The Perception Gap: A Survey of VLM Robustness Defenses for Computer-Use Agent Safety

Anna-Alina Bondarets [1] **Taras Rumezhak** [1] **Volodymyr Karpiv** [1]

Abstract

Computer-use agents (CUAs) are systems that perceive screen state and execute GUI and OS actions in live desktop or browser environments; Claude Computer Use, OpenAI Operator, and UI-TARS are already in production. The safety problem is one of visual grounding. Untrusted screen content reaches the agent's reasoning pipeline through a perception layer that is both the primary attack surface and, in existing surveys, the least defended component. Prior CUA safety work covers it as one category among many and draws defenses entirely from systems security, leaving the VLM robustness literature (certified defenses, encoder hardening, safety fine-tuning) largely unengaged. This paper surveys the intersection. We organize seven perception-layer attack families and map nine published VLM robustness defenses against them; we also cover three emerging defense families (access control, formal runtime governance, industrial mechanisms) addressing the action and trajectory layers. Two findings structure the analysis: no published defense fully addresses typographic exploitation or GUI deception in a CUA setting, and VLM robustness defenses degrade substantially in agentic deployment due to distribution and threat-model mismatches that are systematic but addressable. We identify six open problems and the research agenda needed to close them.

1. Introduction

Large language models deployed as computer-use agents do more than produce text: they click buttons, fill forms, execute code, and manage files on live operating systems, driven entirely by visual interpretation of screen state (Xie et al., 2024; Zheng et al., 2024). That coupling of visual perception to OS-level action turns the *perception layer* (the path from raw screenshot to VLM planner input) into the system's primary security boundary.

The threat is active: safety-trained LLMs comply with 98 of 100 harmful browser-agent tasks without any jailbreak (Kumar et al., 2024), non-ML attacks against commercial agents succeed in the wild (Li et al., 2025), pop-up adversarial GUI elements reach 86% ASR on production benchmarks (Zhang et al., 2025b), and fine-print screen injections bypass human oversight entirely (Chen et al., 2025a). Yet the two communities that could address this ignore each other. The *security and NLP community* produces threat taxonomies, injection benchmarks, and guard agents (Chen et al., 2025b; Debenedetti et al., 2024; Xiang et al., 2025) while treating the perception layer as a black box. The *VLM and CV community* develops principled perception-layer defenses (robust encoder training (Schlarmann et al., 2024), randomized smoothing (Sun et al., 2024), safety fine-tuning (Zong et al., 2024), adaptive prompt shielding (Wang et al., 2024b)) but evaluates them only on static jailbreak benchmarks. We found **no prior survey to bridge these two bodies of work.**

Contributions. We make four contributions:

- We provide a structured survey of VLM robustness defenses reframed for CUA deployment, showing which mechanisms operate inside vs. outside the perception bottleneck;
- We introduce a perception-layer threat taxonomy for CUAs, classifying seven attack families including tool-stream and protocol injection;
- We define three emerging defense families and provide survey on them (access control and least-privilege enforcement, Section 4.4; formal and probabilistic runtime governance, Section 4.5; and industrial mechanisms, Section 4.6);
- We define six open problems structuring the research agenda for perception-centric CUA governance.

Relationship to prior surveys. Chen et al. (2025b) provide the most comprehensive existing CUA safety treat-

[1]SoftServe R&D Department, Lviv, Ukraine. Correspondence to: Anna-Alina Bondarets <anbondaret@softserveinc.com>.

Second Workshop on Technical AI Governance Research (TAIGR) @ ICML 2026, Seoul, South Korea. 2026. Copyright 2026 by the author(s).

ment (14 defense categories, 30+ benchmarks, 68 pages), but their perception-layer defense draws entirely from the NLP/security literature: Robust CLIP, SmoothVLM, VL-Guard, and AdaShield appear in neither their defense taxonomy nor benchmark suite, and the authors explicitly identify the CV robustness connection as open. He et al. (2025) survey LLM agent security (*ACM Computing Surveys*) from an authentication and data-integrity perspective, treating the visual channel as a prompt-injection variant without engaging adversarial ML methods. Kim et al. (2026) cover the agentic attack surface from systems security similarly. The present paper is, to our knowledge, the first to systematically map certified and encoder-hardening defenses from the CV robustness literature onto the CUA threat taxonomy.

2. Background: CUA Architecture and the Perception Bottleneck

CUA systems. CUAs follow a perceive-plan-act loop. Figure 1 illustrates the canonical architecture. Screen state is captured as a screenshot; a parser such as OmniParser (Lu et al., 2024) or SeeClick (Cheng et al., 2024) performs OCR, UI-element detection, and visual grounding to produce a structured scene description; a VLM planner (GPT-4V, Claude 3.5, UI-TARS (Qin et al., 2025)) generates an action; and an executor maps the action to OS-level events. The loop repeats until task completion or budget exhaustion, with benchmarks such as OSWorld (Xie et al., 2024), VisualWebArena (Koh et al., 2024), and Windows Agent Arena (Bonatti et al., 2024) providing evaluation infrastructure.

The perception layer. We define the *perception layer* as the processing path from raw screenshot pixels to the text or embedding representation that is passed to the VLM planner (Figure 1, orange bracket). This path includes screenshot preprocessing, OCR, UI-element detection, visual grounding, and any image-to-text conversion. The perception layer is the CUA's primary interface with the untrusted environment: every web page, document, and desktop UI processed by the agent is untrusted external content.

Why the perception layer is the weakest link. Three structural properties make this layer dangerous in a way that compounds. First, it processes *untrusted content at high bandwidth*: a single web page visited during an agent task may contain hundreds of adversarially crafted visual elements. Second, it converts visual signals to text, creating a *modality injection vector*: content that is visually subtle or invisible to humans (small-font text, adversarial pixel perturbations) can inject full natural-language instructions into the planner's context. Third, *existing text-based guardrails are downstream* of the perception layer and therefore see only the text output, not the adversarial visual input that produced it. This asymmetry is the central vulnerability this survey addresses.

From output safety to trajectory safety. The introduction of tool-use frameworks such as ReAct (Yao et al., 2023) and Toolformer (Schick et al., 2023) transformed LLMs from text producers into action-taking agents with interleaved reasoning and tool invocation. This shift changes the unit of safety analysis: a single unsafe *output* becomes a sequence of unsafe *actions* embedded in a trajectory. For CUAs, this implies that perception-layer hardening is necessary but not sufficient: a robust encoder reduces the probability that adversarial screen content hijacks the planner, but cannot alone prevent individually plausible actions from collectively realizing an attacker's goal. Effective CUA governance must therefore compose perception-layer defenses with action-level controls, access-policy enforcement, and trajectory-aware monitoring.

Scope and adjacent communities. This survey focuses on the intersection of VLM robustness and CUA safety. Embodied AI safety (Xing et al., 2025) addresses perception robustness in robotics (adversarial point clouds, sensor spoofing) with structural parallels but a different input modality. Large-scale VLM safety surveys (Ma et al., 2025a) cover 700+ references but do not address the agentic deployment context. Capability benchmarks such as WebVoyager (He et al., 2024), AgentBench (Liu et al., 2024), AndroidWorld (Rawles et al., 2024), and GAIA (Mialon et al., 2023) establish the performance baselines that frame the capability-safety gap analyzed in Section 5.

3. Perception-Layer Threats in CUAs

We organize perception-layer threats into seven families. Table 1 summarizes the taxonomy with representative attacks and empirical ASRs.

3.1. Direct Visual Injection

The most studied threat is the adversarial image: a pixel-level perturbation that steers VLM behavior. Qi et al. (2024) show a single adversarial image universally jailbreaks aligned VLMs, bypassing RLHF safety training; Bailey et al. (2023) exploit the vision encoder's high-dimensional attack surface to enforce arbitrary behaviors at runtime ($> 80\%$ ASR on LLaVA). The compositional attack of Shayegani et al. (2024) requires only CLIP encoder access to generate perturbations effective against black-box VLMs, meaning any CUA sharing a CLIP-based grounding module (SeeClick, OmniParser, SeeAct) inherits the vulnerability. Automated pipelines lower attacker effort further: Wu et al. (2023) use GPT-4 to iteratively generate visual jailbreaks against GPT-4V, reaching 98.7% ASR, while Ying et al. (2024) jointly optimize visual and textual perturba-

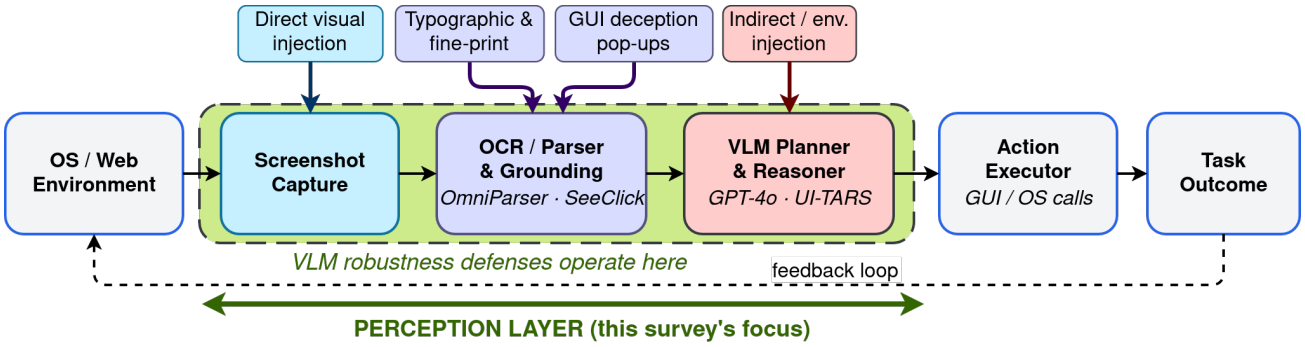

Figure 1. CUA perceive-plan-act pipeline. Red annotations mark active attack injection points; the orange bracket denotes the perception layer this survey focuses on; the green dashed box indicates where VLM robustness defenses intervene. Text-based security defenses (right of the VLM box) are downstream of the perception bottleneck.

tions for $\approx$29% ASR gain over unimodal baselines. The fleet-level risk is acute: Gu et al. (2024) show an "infectious jailbreak" in which a single adversarial image propagates through shared agent memories to compromise over one million instances exponentially fast.

3.2. Typographic and OCR Exploitation

Typographic attacks embed harmful instructions as rendered text within images, exploiting the VLM's text-in-image reading capability. Gong et al. (2025b) achieve 82.5% ASR across six open-source VLMs in a black-box setting; the CUA analog is direct: instructions printed on a web page are legible to the OCR stage but visually inconspicuous to the user. Chen et al. (2025a) study fine-print injections across 234 adversarial pages and 39 participants, showing that small-font text visible to OCR but not to casual inspection bypasses human oversight entirely. Qraitem et al. (2024) show that VLMs can generate their own typographic adversarial overlays, inducing a $\sim$60% accuracy drop in a self-contamination loop; Zhao et al. (2025) further show that character shuffling defeats safety fine-tuning on GPT-4o and Claude-3.5, since non-standard visual encodings of prohibited content are rarely covered during training.

3.3. GUI Deception and Pop-up Attacks

Adversarial modifications to UI elements exploit the fact that the perception layer must interpret not just pixel content but the semantic structure of interactive interfaces. Zhang et al. (2025b) place adversarial pop-up elements within the OSWorld and VisualWebArena environments and achieve 86% ASR and 47% task-degradation rate, demonstrating that UI-layer deception is both practical and effective on leading evaluation platforms. Ma et al. (2025b) show that environmental visual distractions degrade GUI-specialist models from their task even without pixel-level adversarial crafting, suggesting that "good enough" UI noise suffices in many practical scenarios.

3.4. Indirect and Environmental Injection

Environmental injection embeds adversarial instructions in content the agent fetches during legitimate task execution, hiding in web page reviews, advertisements, or document bodies. Liao et al. (2025) achieve 70% ASR for PII exfiltration via injected web-page instructions; Shapira et al. (2025) reach $> 80\%$ ASR across camera activation, file theft, and user impersonation through task-aligned injections in reviews and comments. Cross-modal injection (Bagdasaryan et al., 2023) extends the surface to any media the agent downloads or displays, while the Imprompter attack (Fu et al., 2024) shows that injection prompts can be fully concealed from human inspection and remain effective across model families.

3.5. Availability and Memory Attacks

Availability attacks target agent uptime rather than behavior. Gao et al. (2024) show that imperceptible perturbations inflate VLM generation by $7.87\times$, creating denial-of-service for CUAs in bounded-budget loops; Zhang et al. (2024) extend this with malfunction amplification achieving $> 80\%$ failure rate. Memory poisoning closes the second end of the agent lifecycle: Chen et al. (2024) backdoor agent RAG modules at below 0.1% poison rate with $> 80\%$ ASR, while Xu et al. (2024) demonstrate that 50 poisoned image-caption pairs suffice to misidentify visual concepts reliably across model scales.

3.6. Tool-Stream and Protocol Injection

Malicious content can also enter via tool *descriptions*, output metadata, and resource definitions communicated through orchestration protocols, bypassing the screenshot pipeline entirely. The Model Context Protocol (Anthropic, 2024b) acknowledges servers as a trust surface requiring explicit security and consent, yet production CUAs routinely grant broad tool access without per-task capability

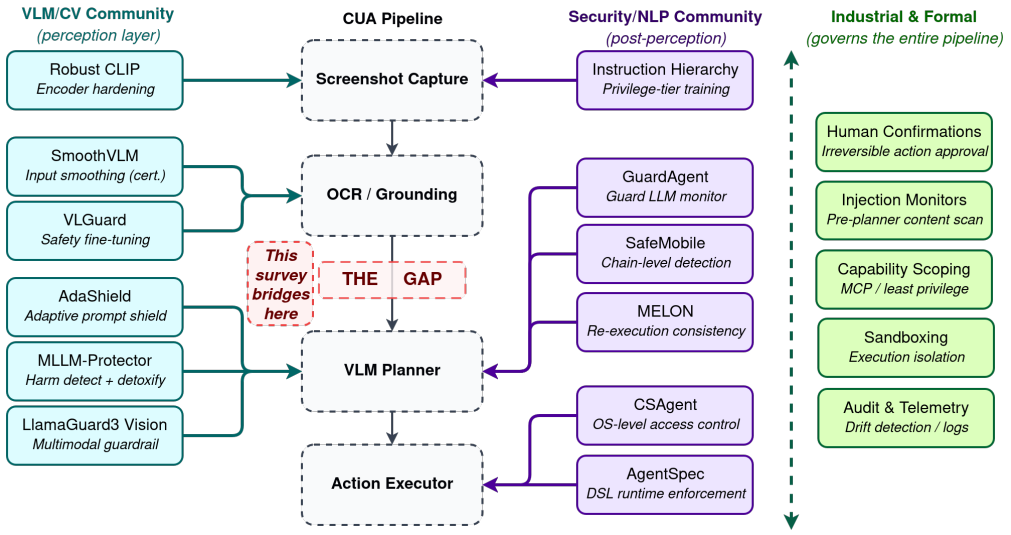

Figure 2. Defense positioning in the CUA pipeline. VLM/CV community defenses (left, blue) operate *inside* the perception layer; CUA security defenses (right, purple) operate *after* the perception layer. The red zone marks the gap this survey targets.

scoping. An attacker controlling a tool server can inject adversarial content into tool schemas, poisoning the planner before any screen interaction. In multi-agent settings, a compromised sub-agent can return crafted outputs that escalate privileges through the confused-deputy problem. Unlike the indirect attacks in Section 3.4, which exploit the OCR path, tool-stream injection reaches the planner as structured text with the appearance of trusted output, circumventing perception-layer defenses entirely.

4. Defense Landscape: Two Disconnected Literatures

Figure 2 captures the gap: three communities have built defenses for different points in the CUA pipeline, with no systematic cross-community engagement. The VLM/CV community (Section 4.1) provides perception-layer hardening; the Security/NLP community (Section 4.2) provides action-level governance; industrial and formal governance work (Sections 4.4–4.6) addresses system-level controls. The cross-community gap is analyzed in Section 4.3.

4.1. VLM Robustness Defenses: The Underutilized Pool

Robust encoder training. Schlarmann et al. (2024) introduce Robust CLIP, an unsupervised adversarial fine-tuning procedure for the CLIP vision encoder that requires only unlabeled images. Because OmniParser (Lu et al., 2024), SeeClick (Cheng et al., 2024), and SeeAct (Zheng et al., 2024) all use CLIP-based grounding, swapping in Robust CLIP is a drop-in operation that hardens the primary attack surface with no architecture changes. No CUA safety paper has evaluated it in an agentic setting.

Input smoothing with certified guarantees. Sun et al. (2024) adapt randomized smoothing (Cohen et al., 2019) to the VLM setting, providing certifiable defense against patched adversarial injectors. SmoothVLM reduces ASR to 0–5% on patched-prompt attacks against leading VLMs and provides formal certificates bounding the maximum adversarial patch area that can affect the prediction. In the CUA context, SmoothVLM can be applied as a preprocessing step to screenshots before passing them to OmniParser or the VLM planner, providing the only currently available *certifiable* defense against patch-class visual injection. The certificate is contingent on the patch area bound, which makes SmoothVLM most effective against localized attacks (pop-ups, patch injections) rather than full-image perturbations.

Safety fine-tuning. Zong et al. (2024) introduce VLGuard, a mixed-safe/unsafe multimodal training dataset for post-hoc safety fine-tuning of VLMs at essentially zero helpfulness cost on standard benchmarks. VLGuard provides a practical recipe for incorporating visual safety constraints into an existing CUA backbone: fine-tune on the VLGuard dataset with a CUA-specific augmentation of adversarial screenshots, then evaluate on OS-Harm (Kuntz et al., 2025) and AgentDojo (Debenedetti et al., 2024). The key CUA-specific gap is that VLGuard was developed for chat safety, not GUI safety, and has not been adapted to screen-content training distributions.

Inference-time and output-side defenses. Wang et al. (2024b) propose AdaShield, adaptive defense-prompt refinement that wraps any VLM without fine-tuning; Wang et al. (2024a) extend this with test-time bimodal prompt tuning

Table 1. CUA Threat Taxonomy. ASR values are maxima reported in source papers. **Bold** denotes attack families with no published defense in the CUA setting.

Family	Representative attack	Venue	Max ASR	CUA mechanism
Direct visual inject.	Qi et al. (2024)	AAAI'24	97%	Adversarial pixels jailbreak VLM
	Bailey et al. (2023)	arXiv'23	80%	Runtime behavioral control
	Gu et al. (2024)	ICML'24	n/a	Cascade via agent memory
Typographic / OCR	Gong et al. (2025b)	AAAI'25	82.5%	Text-in-image injection
	Chen et al. (2025a)	arXiv'25	–	Fine-print bypass oversight
GUI deception	Zhang et al. (2025b)	ACL'25	86%	Adversarial pop-ups
	Ma et al. (2025b)	ACL'25	–	Environmental distraction
Indirect injection	Liao et al. (2025)	ICLR'25	70%	Privacy leakage via web
	Shapira et al. (2025)	arXiv'25	80%	Reviews/ads injection
	Fu et al. (2024)	arXiv'24	–	Obfuscated PII exfiltration
Availability	Gao et al. (2024)	ICLR'24	7.87×	Token-budget exhaustion
Mem. poisoning	Chen et al. (2024)	NeurIPS'24	80%	RAG backdoor, 0.1% rate
Tool-stream inject.	Anthropic (2024b)	misc'24	–	Tool schema / output poisoning

(TAPT) achieving $+48.9\%$ AutoAttack robustness without labeled attack data. Pi et al. (2024) add MLLM-Protector, a harm-detector plus detoxifier pipeline preserving helpfulness while providing output-side moderation. First-token distribution analysis (Zhao et al., 2024) detects jailbreaks and deceptive prompts before full generation, offering a lightweight action-gating signal with minimal latency overhead in the CUA loop. Chi et al. (2024) offer Llama Guard 3 Vision, a multimodal guardrail covering the MLCommons hazard taxonomy, extending the text-only Llama Guard (Inan et al., 2023) to visual inputs; ShieldGemma (Zeng et al., 2024) provides a strong text-modality baseline whose visual extension to CUA screen content remains open.

4.2. CUA-Native Defenses: Action-Level Governance

Instruction hierarchy. Wallace et al. (2024) train LLMs to respect a three-tier privilege hierarchy: system prompt $\succ$ user instructions $\succ$ environment content. Screen OCR output falls in the environment tier; training the model to discount environment-tier instruction-following provides a principled defense against visual injection by design. The key limitation is that the instruction hierarchy operates on the *text representation* of screen content produced by the perception layer, not on the visual signal itself. A forged high-privilege visual signal (e.g., rendered text saying "[SYSTEM]:" in a fine-print region) can circumvent tier assignment if the perception layer does not flag it.

Guard-agent architectures. Xiang et al. (2025) deploy a dedicated guard LLM that reviews planned actions before execution, achieving 98% and 83% accuracy on two benchmark settings. Critically, the guard agent operates on action representations, not screen images, creating a defense blind to the visual content that motivated the action. Liang et al. (2025) take a step toward visual guard agents with SafeMobile, a chain-level jailbreak detection system for multimodal mobile agents analyzing multi-round visual and action sequences; extension to desktop or web CUAs remains open.

Runtime enforcement and re-execution. Wang et al. (2026) introduce AgentSpec, a domain-specific language for specifying safety constraints enforced at runtime, achieving $> 90\%$ prevention in code-generation agents with millisecond overhead. Zhu et al. (2025) propose MELON, which detects indirect prompt injection by masked re-execution: a subset of tool calls is masked and the agent is re-run; if the resulting trajectory diverges, injection is detected. MELON achieves state-of-the-art performance on AgentDojo with 93% defense effectiveness and 90% cost reduction relative to full re-execution. Neither AgentSpec nor MELON has been evaluated on CUAs where the injection vector is visual rather than textual.

OS-level access control. Gong et al. (2025a) implement CSAgent, an intent- and context-aware OS-level policy engine for CUAs with 1.99% runtime overhead, providing defense-in-depth complementary to perception-layer hardening. Access control is post-perception by construction; adversarial content that influences LLM reasoning without triggering a blocked OS call remains an unmitigated threat, motivating the deeper capability-scoping approaches surveyed in Section 4.4.

Table 2. Defense coverage by threat family and community. = full; = partial; = none.

Threat family	VLM/CV	CUA Sec.	Access/Runtime
Direct visual injection	partial	none	none
Typographic / OCR	partial	none	none
GUI deception	none	partial	partial
Indirect injection	partial	full	partial
Availability	none	partial	none
Memory poisoning	partial	none	partial
Tool-stream inject.	none	partial	full

4.3. The Gap: Persistent Coverage Holes

Figure 3 and Table 2 show the coverage directly. VLM robustness defenses handle direct visual injection and typographic attacks, at least partially; they provide nothing for GUI deception, indirect environmental injection, availability attacks, memory poisoning, or tool-stream injection. CUA-native defenses fill indirect-injection and availability gaps but are blind to direct visual and typographic attacks. The families in Sections 4.4–4.6 extend coverage to tool-stream injection and trajectory-level threats, but two gaps remain: no published defense provides full coverage for typographic OCR exploitation or GUI deception pop-ups in a CUA setting. Both are shown in bold in Table 1.

4.4. Access Control and Least-Privilege Enforcement

Production CUAs are routinely granted broad tool access: a browsing agent that can also write files and execute code is substantially riskier than one limited to task-necessary capabilities. MCP (Anthropic, 2024b) defines explicit tool, resource, and prompt labels but most deployments do not enforce per-task restrictions. The Instruction Hierarchy (Wallace et al., 2024) addresses a related problem at training time by demoting environment-tier content; AgentSpec (Wang et al., 2026) enforces per-step constraints at runtime, blocking disallowed actions after planning rather than constraining what the agent can plan in the first place. Task-necessary privilege scoping and hazard-analysis-driven specification are active research areas with no verified system yet; this is the primary structural gap for multi-tool CUAs.

4.5. Formal and Probabilistic Runtime Governance

AgentSpec and MELON provide step-level guarantees: per-action constraint enforcement and re-execution consistency checking, respectively. Beyond individual steps, the open problem is trajectory-aware governance: reasoning about action sequences to detect unsafe plans before they execute. Research here includes probabilistic trajectory prediction from historical traces, policy specification that treats the full trajectory as the unit of analysis, and fleet-level telemetry for drift detection, all of which shift the guarantee from reactive to *anticipatory*. AgentSpec (Wang et al., 2026) with MELON (Zhu et al., 2025) is the current state of the art; neither has been applied to visual CUA settings.

4.6. Industrial Governance Mechanisms

Production systems offer a first generation of governance mechanisms absent from the academic literature. OpenAI Operator (OpenAI, 2025) gates irreversible actions behind human confirmation and pre-filters untrusted web content with a prompt-injection classifier. Anthropic's browser-use system (Anthropic, 2024a) trains RL-based injection robustness alongside classifier layers. Both surface the approval-fatigue problem: production users approve approximately 93% of agent action requests (Anthropic, 2024a); undifferentiated confirmation is effectively no oversight, and prompts must be selective enough for users to evaluate what they are approving.

The MCP specification (Anthropic, 2024b) defines tool and resource interfaces with explicit consent semantics, positioning the protocol layer as an emerging capability-restriction control plane. The UK AISI Inspect framework (UK AI Safety Institute, 2024) provides adaptable evaluation infrastructure for CUA-specific testing. NIST AI RMF 1.0 (National Institute of Standards and Technology, 2023) and the Generative AI Profile (National Institute of Standards and Technology, 2024) supply lifecycle-aligned risk functions but stop short of agent-executable technical controls; the EU GPAI Code of Practice (European Commission, 2024) imposes provider transparency obligations without reaching perception-layer threats. Translating these frameworks into measurable controls is the practical analog of OP6 (Section 6).

5. Transfer Analysis

VLM robustness defenses are designed and evaluated for static jailbreak settings. We assemble reported ASR values across three scenarios of increasing CUA-specificity (Figure 4) to characterize how much of that protection survives the context shift to agentic deployment.

The capability-safety gap. On GAIA (Mialon et al., 2023), the best agents solve 15% of tasks; humans solve 92%. Current CUAs are genuinely capability-limited. Yet at that 15% success rate, agents still comply with 98 of 100 harmful browser tasks without jailbreaking (Kumar et al., 2024), and pop-up attacks succeed at 86% ASR (Zhang et al., 2025b). As capabilities close toward human performance, the attack surface grows with them; safety cannot be delegated to incapability. A defense that works today must still work when the agent is better.

No coverage | Partial coverage | Full coverage

	VLM/CV Community					Security/NLP Community					Access Control & Formal Runtime	
	Robust CLIP	Smooth VLM	VLGuard	Ada Shield	LG3-Vision	Instr. Hierarchy	Guard Agent	Safe Mobile	MELON	CSAgent	Access Control	Formal Runtime
Direct Visual Injection	full	full	full	partial	partial							partial
Typographic / OCR Exploit	partial		partial	full	partial	partial						
GUI Deception Pop-ups					partial		partial	partial		partial		partial
Indirect / Env. Injection			partial	partial	partial	full	partial	partial	full		partial	partial
Availability DoS		partial										
Memory Poisoning			partial						partial			partial
Tool-Stream / Protocol Inject.						partial				partial	full	partial

Figure 3. Threat-defense coverage matrix. Green = full coverage, yellow = partial, red = none. Left of the dashed line: VLM/CV community defenses. Right: CUA security defenses. No defense provides full coverage for typographic exploitation or GUI deception in the CUA setting.

Setup. We compare five defenses across three settings: (1) static VLM jailbreaking (the setting for which each defense was designed), (2) CUA visual injection (adversarial pixels injected into screenshots), and (3) CUA pop-up attack (GUI-layer adversarial elements). ASR values for the static setting are drawn from each paper's primary evaluation. For CUA settings, values are extrapolated from the closest available evaluation in the papers plus an analysis of structural mismatch between the defense's assumptions and the CUA threat model (details in Appendix B).

Results. Robust CLIP (Schlarmann et al., 2024) provides strong protection against direct adversarial injection in the static setting, degrading to moderate in CUA scenarios because UI screenshots differ significantly from the natural-image fine-tuning distribution; against pop-up attacks (UI-layer deception, not pixel perturbation) it provides no protection. SmoothVLM (Sun et al., 2024) transfers its patch-bounded certificate partially to localized CUA injection but not to typographic or fine-print attacks, and its 30% latency overhead may be prohibitive for real-time operation. Llama Guard 3 Vision (Chi et al., 2024) reduces ASR on static jailbreaking but is weak on CUA-specific threats: screen content and planned OS actions do not map to chat-safety hazard taxonomies (*distribution mismatch*). The Instruction Hierarchy (Wallace et al., 2024) handles environmental injection well by design but is blind to pixel-level attacks that operate below the text-representation level where privilege tiers are enforced. SafeMobile (Liang et al., 2025) confirms that chain-level detection outperforms single-step classification on pop-up-style attacks in mobile GUI environments; transfer to desktop/web environments (OSWorld, VisualWebArena) remains unevaluated.

Takeaway. VLM robustness defenses degrade substantially (and in some cases entirely) moving from static jailbreak to CUA deployment. Two mismatches drive this: the defense's training domain differs from CUA screen content, and the defense's formal assumptions (e.g., patch bounds) do not cover CUA-specific attack vectors (e.g., GUI-layer deception). Neither mismatch is inherent to the underlying methods; both point to a concrete adaptation program developed as open problems in Section 6.

6. Open Problems

OP1: CUA-specific robust encoder training. Robust CLIP was fine-tuned on natural-image perturbations, not GUI distributions. Screen images differ structurally (high-frequency text edges, consistent UI widget patterns, uniform backgrounds), making CUA-specific adversarial fine-tuning of CLIP on OSWorld, VisualWebArena, and Windows Agent Arena a concrete near-term contribution. Technical risk is low: the training procedure is well-specified and evaluation infrastructure exists.

OP2: Certifiable defenses for typographic and fine-print attacks. SmoothVLM certifies patch-bounded perturbations but not *rendered-text injection*, where the adversarial signal is a legitimate image element (a small-font sentence) rather than a perceptual perturbation. No certifiable defense exists for the fine-print attack class (Chen et al., 2025a), requiring new formal threat models that treat "legibility to OCR but not to humans" as the adversarial property.

OP3: CUA-specific multimodal safety classifiers. Llama Guard 3 Vision (Chi et al., 2024) and ShieldGemma (Zeng et al., 2024) classify chat safety; OS-Harm (Kuntz et al., 2025) provides a CUA harm taxonomy. Missing is a

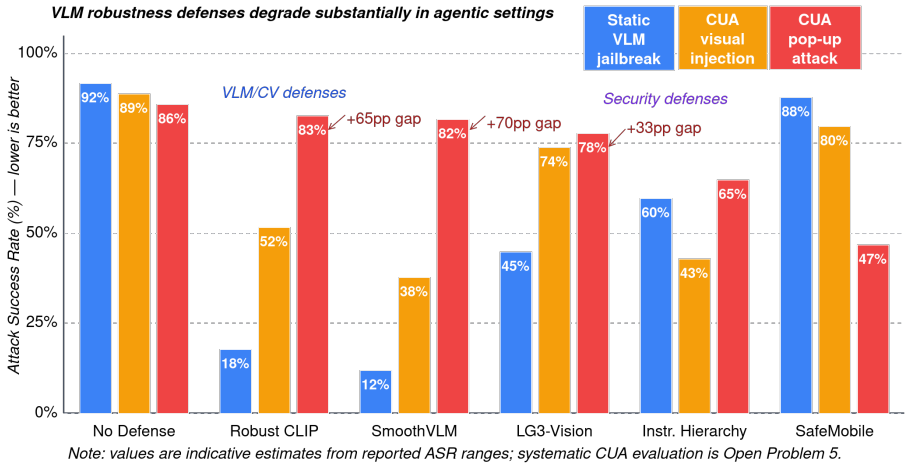

Figure 4. Defense transfer analysis. Attack success rate (%, lower is better) across three scenarios of increasing CUA-specificity. VLM/CV defenses (blue shading) lose effectiveness sharply against CUA pop-up attacks; security defenses (purple) are more effective against environmental injection but blind to direct visual attacks. *Note: values are indicative estimates from reported ASR ranges in source papers; systematic CUA evaluation is identified as Open Problem 5 (see Appendix B).*

guardrail trained on (screenshot, planned-action) pairs that jointly classifies visual content and intended action, catching cases where a phishing pop-up combined with a planned "click" is harmful even when neither component is flagged alone.

OP4: Temporal and chain-level perception governance. SafeMobile (Liang et al., 2025) shows attack signals accumulate across steps in ways invisible at any single step. Extending chain-level visual-action detection from mobile GUI to desktop and web CUAs with OSWorld's full action vocabulary remains open; the challenge is efficient visual-state aggregation across long trajectories without prohibitive memory overhead.

OP5: Adversarial robustness benchmarks for grounding pipelines. OmniParser, SeeClick, and UI-TARS are evaluated only on task-success metrics; no benchmark tests grounding-layer robustness under adversarial perturbations, pop-ups, or fine-print injections. OS-Harm (Kuntz et al., 2025) evaluates end-to-end agent safety, not the grounding layer. An adversarial grounding benchmark would enable targeted perception-layer optimization independent of the full agent system.

OP6: Compositional formal guarantees across the governance stack. Each mechanism in Sections 4.4–4.6 provides guarantees for one slice: per-step action constraints, per-tool capability restrictions, probabilistic trajectory prediction, or hazard-driven requirement derivation. No framework spans all layers simultaneously for realistic multimodal CUAs. A compositional framework would let an operator state a high-level policy ("do not exfiltrate user credentials") enforced jointly by the encoder, access-control layer, trajectory monitor, and audit log, with a soundness proof. The NIST AI RMF lifecycle functions provide the policy vocabulary (National Institute of Standards and Technology, 2023); translating them into agent-executable specifications connecting OP3 and OP5 is the open engineering challenge.

7. Conclusion

Computer-use agent safety cannot be solved at the action level. The perception layer is the primary attack surface, and the VLM robustness literature (Robust CLIP, SmoothVLM, VLGuard, AdaShield, MLLM-Protector, token-distribution detection, Llama Guard 3 Vision) contains directly applicable methods not yet tested in this setting. Our coverage analysis (Table 2, Figure 3) shows typographic and GUI-deception attacks remain without full coverage even with access control, formal runtime governance, and industrial mechanisms added; our transfer analysis (Figure 4) confirms VLM defenses degrade substantially in agentic deployment, and the capability-safety asymmetry documented by GAIA (Mialon et al., 2023) means this gap widens as agents improve.

The six open problems we identify (CUA-specific encoder hardening, certifiable defenses for typographic attacks, (screenshot, action) safety classifiers, chain-level perception governance, adversarial grounding benchmarks, and compositional governance-stack verification) are tractable with existing methods; the missing ingredient is cross-community work among VLM robustness researchers, CUA security researchers, and governance engineers. Closing that gap is this paper's purpose.

8. LLM Usage Statement

In this work, we used large language models and AI agents as writing assistants throughout the drafting of this manuscript. LLM assistance was limited to text composition, editing, and structural organization; all figures and plots were produced manually. The literature survey reflects a combination of systematic research led by us and LLM-assisted retrieval and synthesis, with all cited claims independently double-checked and verified. Final scientific judgments, the threat taxonomy, the coverage analysis, and the open-problem framing are our own.

References

Andriushchenko, M., Souly, A., Bhatt, M., Sherburn, D., Zhang, E., Nguyen, M., Llamas, C., Rando, J., Cooney, E., Sherburn, T., et al. AgentHarm: A benchmark for measuring harmfulness of LLM agents. In *International Conference on Learning Representations*, 2025. arXiv:2410.09024.

Anthropic. Developing a computer use model. https://www.anthropic.com/research/developing-computer-use, 2024a. Documents RL-based prompt-injection robustness training, classifier scanning, and red-teaming for browser-use agents.

Anthropic. Model context protocol specification. https://modelcontextprotocol.io, 2024b. Open standard for AI application–tool integration with explicit security and consent mechanisms.

Bagdasaryan, E., Hsieh, T.-Y., Nassi, B., and Shmatikov, V. Abusing images and sounds for indirect instruction injection in multi-modal LLMs. *arXiv preprint arXiv:2307.10490*, 2023.

Bailey, L., Ong, E., Russell, S., and Emmons, S. Image hijacks: Adversarial images can control generative models at runtime. *arXiv preprint arXiv:2309.00236*, 2023.

Bonatti, R., Zhao, D., Bonacci, F., Dupont, D., Abdali, S., Li, Y., Lu, S., Wagle, J., Xu, K., Koishida, K., et al. Windows agent arena: Evaluating multi-modal OS agents at scale. *arXiv preprint arXiv:2409.08264*, 2024.

Chen, C., Peng, Z., Ding, C., Lam, W., Shum, H.-Y., et al. The obvious invisible threat: LLM-powered GUI agents' vulnerability to fine-print injections. *arXiv preprint arXiv:2504.11281*, 2025a.

Chen, J., Guo, Y., Xu, G., Zhao, D., Zhu, R., et al. A survey on the safety and security threats of computer-using agents: JARVIS or Ultron? *arXiv preprint arXiv:2505.10924*, 2025b. v3, Aug 2025. 68 pages.

Chen, Z., Guo, Z., Xiao, C., Zhao, D., and Li, B. AgentPoison: Red-teaming LLM agents via poisoning memory or knowledge bases. In *Advances in Neural Information Processing Systems*, 2024. arXiv:2407.12784.

Cheng, K., Sun, Q., Chu, Y., Xu, F., Li, Y., Zhang, J., and Wu, Z. SeeClick: Harnessing GUI grounding for advanced visual GUI agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, 2024. arXiv:2401.10935.

Chi, J., Karn, U., Zhan, H., Smith, E. M., Rando, J., and Team, M. L. Llama guard 3 vision: Safeguarding human-AI image understanding conversations. *arXiv preprint arXiv:2411.10414*, 2024.

Cohen, J. M., Rosenfeld, E., and Kolter, J. Z. Certified adversarial robustness via randomized smoothing. *International Conference on Machine Learning*, pp. 1310–1320, 2019.

Debenedetti, E., Zhang, J., Balunović, M., Beurer-Kellner, L., Fischer, M., and Vechev, M. AgentDojo: A dynamic environment to evaluate attacks and defenses for LLM agents. In *Advances in Neural Information Processing Systems*, 2024. arXiv:2406.13352.

European Commission. Code of practice for general-purpose AI models. https://digital-strategy.ec.europa.eu/en/policies/ai-code-practice, 2024. Self-regulatory code under EU AI Act for GPAI model providers.

Fu, X., Liu, S., Vaidya, A., Tu, Y., He, H., Xie, X., Liu, L., Mittal, P., Popa, R. A., and Hong, Y. Imprompter: Tricking LLM agents into improper tool use. *arXiv preprint arXiv:2410.14923*, 2024.

Gao, Z., Wang, K., Jiang, T., Xu, W., and Huang, T. Inducing high energy-latency of large vision-language models with verbose images. In *International Conference on Learning Representations*, 2024. arXiv:2401.11170.

Gong, M., Qin, Y., Ding, P., et al. CSAgent: Secure access control for CUAs via context space. *arXiv preprint arXiv:2509.22256*, 2025a.

Gong, Y., Deng, D., Song, T., Jin, J., Wang, P., and Lyu, C. FigStep: Jailbreaking large vision-language models via typographic visual prompts. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2025b. Oral. arXiv:2311.05608.

Gu, X., Zheng, X., Pang, T., Du, C., Liu, Q., Wang, Y., Jiang, J., and Lin, M. Agent Smith: A single image can jailbreak one million multimodal LLM agents exponentially fast. In *International Conference on Machine Learning*, 2024. arXiv:2402.08567.

He, F., Huang, T., Zhang, X., Zheng, Y., He, T., et al. The emerged security and privacy of LLM agent: A survey with case studies. *ACM Computing Surveys*, 2025. arXiv:2407.19354.

He, H., Yao, W., Ma, K., Yu, W., Dai, Y., Zhang, H., Lan, Z., and Yu, D. WebVoyager: Building an end-to-end web agent with large multimodal models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, 2024. arXiv:2401.13919.

Inan, H., Upasani, K., Chi, J., Rungta, R., Iyer, K., Mao, Y., Tontchev, M., Hu, Q., Fuller, B., Testuggine, D., et al. Llama guard: LLM-based input-output safeguard for human-ai conversations. *arXiv preprint arXiv:2312.06674*, 2023.

Kim, T., Song, D., Zhang, R., et al. The attack and defense landscape of agentic AI. In *32nd USENIX Security Symposium*, 2026. arXiv:2603.11088.

Koh, J. Y., Lo, R., Jang, L., Duvvur, V., Lim, M. C., Huang, P.-Y., Neubig, G., Zhou, S., Salakhutdinov, R., and Fried, D. VisualWebArena: Evaluating multimodal agents on realistic visual web tasks. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, 2024. arXiv:2401.13649.

Kumar, P., Agarwal, E., Panda, S., Wang, F., Shen, S., Phute, M., Bhatt, S., Bhatt, U., Bhatt, M., et al. Refusal-trained LLMs are easily jailbroken as browser agents. *arXiv preprint arXiv:2410.13886*, 2024. BrowserART.

Kuntz, T., Yuan, J., Miller, D., Lee, Y., Joshi, M., Levy, K., Ziegler, D., et al. OS-Harm: A benchmark for measuring the safety of computer-use agents. In *Advances in Neural Information Processing Systems (Datasets and Benchmarks Track)*, 2025. Spotlight. arXiv:2506.14866.

Li, A., Zang, H., Zhu, M., Hu, S., et al. Commercial LLM agents are already vulnerable to simple yet dangerous attacks. *arXiv preprint arXiv:2502.08586*, 2025.

Liang, S., He, T., Li, Z., Liu, Z., Zhao, H., and Zhao, J. SafeMobile: Chain-level jailbreak detection for multimodal mobile agents. *arXiv preprint arXiv:2507.00841*, 2025.

Liao, Z., He, L., Chen, Z., Wang, Z., Sun, H., and Su, Y. EIA: Environmental injection attack on generalist web agents for privacy leakage. In *International Conference on Learning Representations*, 2025. arXiv:2409.11295.

Liu, X., Yu, H., Zhang, H., Xu, Y., Lei, X., Lai, H., Gu, Y., Ding, H., Men, K., Yang, K., et al. AgentBench: Evaluating LLMs as agents. In *International Conference on Learning Representations*, 2024. arXiv:2308.03688.

Lu, Y., Liu, J., Wei, G., Zhang, J., Min, B.-H., Shi, S., et al. OmniParser for pure vision based GUI agent. *arXiv preprint arXiv:2408.00203*, 2024.

Ma, X., Chu, Y., Li, Z., et al. Safety at scale: A comprehensive survey of large model safety. *arXiv preprint arXiv:2502.05206*, 2025a.

Ma, X., Zhang, Y., Lu, Y., Wang, Z., Foroosh, H., and Xu, H. Caution for the environment: Multimodal agents are susceptible to environmental distractions. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics*, 2025b. arXiv:2408.02544.

Mialon, G., Fourrier, C., Swift, C., Wolf, T., LeCun, Y., and Scialom, T. GAIA: a benchmark for general AI assistants. *arXiv preprint arXiv:2311.12983*, 2023.

National Institute of Standards and Technology. AI risk management framework (AI RMF 1.0). Technical Report NIST.AI.100-1, NIST, 2023.

National Institute of Standards and Technology. AI risk management framework: Generative AI profile. Technical Report NIST.AI.600-1, NIST, 2024.

OpenAI. Operator system card. https://openai.com/index/operator-system-card, 2025. Technical safeguards documentation for OpenAI Operator computer-use agent.

Pi, R., Han, T., Zhao, J., Pan, J., Liang, S., Yao, H., Xu, H., Huang, W., Shen, W., Kawaguchi, K., et al. MLLM-Protector: Ensuring MLLM's safety without hurting performance. *arXiv preprint arXiv:2401.02906*, 2024.

Qi, X., Zeng, Y., Xie, T., Chen, P.-Y., Jia, R., Mittal, P., and Henderson, P. Visual adversarial examples jailbreak aligned large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 21527–21536, 2024. arXiv:2306.13213.

Qin, Y., Ye, Y., Fang, J., Wang, H., Liang, S., Tian, S., Zhang, J., Li, J., Li, Y., Huang, S., et al. UI-TARS: Pioneering automated GUI interaction with native agents. *arXiv preprint arXiv:2501.12326*, 2025.

Qraitem, M., Saini, N., Saenko, K., and Plummer, B. A. Vision-LLMs can fool themselves with self-generated typographic attacks. *arXiv preprint arXiv:2402.00626*, 2024.

Rawles, C., Clinckemaillie, S., Chang, Y., Waltz, J., Lau, G., Fair, M., Li, A., Bishop, W., Li, W., Campbell-Ajala, F., et al. AndroidWorld: A dynamic benchmarking environment for autonomous agents. *arXiv preprint arXiv:2405.14573*, 2024.

Schick, T., Dwivedi-Yu, J., Dessì, R., Raileanu, R., Lomeli, M., Zettlemoyer, L., Cancedda, N., and Scialom, T. Toolformer: Language models can teach themselves to use tools. In *Advances in Neural Information Processing Systems*, 2023. arXiv:2302.04761.

Schlarmann, C., Singh, N. D., Croce, F., and Hein, M. Robust CLIP: Unsupervised adversarial fine-tuning of vision embeddings for robust large vision-language models. In *International Conference on Machine Learning*, 2024. Oral. arXiv:2402.12336.

Shapira, N., Biran, O., Goldstein, M., Levy, O., and Globerson, A. Mind the web: The security of web use agents. *arXiv preprint arXiv:2506.07153*, 2025.

Shayegani, E., Dong, Y., and Abu-Ghazaleh, N. Jailbreak in pieces: Compositional adversarial attacks on multimodal language models. In *International Conference on Learning Representations*, 2024. arXiv:2307.14539.

Sun, K., Gao, C., Hu, H., Lu, X., Zhao, Y., Shi, Z., Zhang, M., Wang, N., Wang, Y., Shi, C., et al. SmoothVLM: Smooth-perturbation-aware vision-language models for visual prompt injection defense. *arXiv preprint arXiv:2405.10529*, 2024.

UK AI Safety Institute. Inspect: A framework for large language model evaluations. https://github.com/UKGovernmentBEIS/inspect_ai, 2024. Open-source evaluation and sandboxing toolkit for AI systems.

Wallace, E., Xiao, K., Leike, R., Weng, L., Heidecke, J., and Beutel, A. The instruction hierarchy: Training LLMs to prioritize privileged instructions. *arXiv preprint arXiv:2404.13208*, 2024.

Wang, H., Poskitt, C. M., and Sun, J. AgentSpec: Customizable runtime enforcement for safe and reliable LLM agents. In *IEEE/ACM International Conference on Software Engineering*, 2026. arXiv:2503.18666.

Wang, X., Zhang, Y., Zhou, Z., and Lan, Z. TAPT: Test-time adversarial prompt tuning for robust inference in vision-language models. *arXiv preprint arXiv:2411.13136*, 2024a.

Wang, Y.-a., Zhang, Y., Fan, J., and Li, Y. AdaShield: Safeguarding multimodal large language models from structure-based attack via adaptive shield prompting. In *European Conference on Computer Vision*, 2024b. arXiv:2403.09513.

Wu, Y., Li, X., Liu, Y., Zhou, P., and Sun, L. Jailbreaking GPT-4V via self-adversarial attacks with system prompts. *arXiv preprint arXiv:2311.09127*, 2023.

Xiang, Z., Zheng, L., Li, Y., Wei, J., Wang, Q., Zhang, H., Li, J., Chen, Z., Tian, Y., Mihalcea, R., et al. GuardAgent: Safeguard LLM agents by a guard agent via knowledge-enabled reasoning. In *International Conference on Machine Learning*, 2025. arXiv:2406.09187.

Xie, T., Zhang, D., Chen, J., Li, X., Zhao, S., Cao, R., Lua, T. J., Xu, Z., Lyu, C., Mao, T., et al. OSWorld: Benchmarking multimodal agents for open-ended tasks in real computer environments. In *Advances in Neural Information Processing Systems*, 2024. arXiv:2404.07972.

Xing, E., Fang, Z., Zhao, Z., et al. Towards robust and secure embodied AI: A survey on vulnerabilities and attacks. *arXiv preprint arXiv:2502.13175*, 2025.

Xu, Y., Yao, J., Shu, M., Sun, Y., Wu, Z., Yu, N., Goldstein, T., and Huang, F. Shadowcast: Stealthy data poisoning attacks against vision-language models. In *Advances in Neural Information Processing Systems*, 2024. arXiv:2402.06659.

Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K., and Cao, Y. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations*, 2023. arXiv:2210.03629.

Ying, Z., Liu, A., Liang, T., Huang, Z., Liu, S., Liu, X., and Tao, D. Jailbreak vision language models via bi-modal adversarial prompt. *arXiv preprint arXiv:2406.04031*, 2024.

Zeng, W., Liu, Y., Mullins, R., Perez, L., Bhatt, J., Butryna, A., Chehab, A., Chernova, I., Colot, A., et al. ShieldGemma: Generative AI content moderation based on Gemma. *arXiv preprint arXiv:2407.21772*, 2024.

Zhang, B., Yu, Y., Zhang, Z., Liu, J., et al. Breaking agents: Compromising autonomous LLM agents through malfunction amplification. *arXiv preprint arXiv:2407.20859*, 2024.

Zhang, H., Huang, J., Zhu, K., Guan, F., et al. Agent security bench (ASB): Formalizing and benchmarking attacks and defenses in LLM-based agents. In *International Conference on Learning Representations*, 2025a. arXiv:2410.02644.

Zhang, H., Yu, C., and Yang, Q. Attacking vision-language computer agents via pop-ups. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics*, 2025b. arXiv:2411.02391.

Zhao, Q., Xu, M., Kailkhura, B., Li, R., Kuo, C.-C. J., and Zhao, Y. The first to know: How token distributions reveal hidden knowledge in large vision-language models. In *European Conference on Computer Vision*, 2024. arXiv:2403.09037.

Zhao, Y., Xu, Y., Zheng, X., Zhang, Y.-a., Ma, K., Shen, L., and Liu, T. Jailbreaking MLLMs via shuffle inconsistency. *arXiv preprint arXiv:2501.04931*, 2025. ICCV 2025.

Zheng, B., Gou, B., Kil, J., Sun, H., and Su, Y. GPT-4V(ision) is a generalist web agent, if grounded. In *International Conference on Machine Learning*, 2024. SeeAct. arXiv:2401.01614.

Zhou, S., Xu, F. F., Zhu, H., Zhou, X., Lo, R., Sridhar, A., Cheng, X., Bisk, Y., Fried, D., Alon, U., et al. WebArena: A realistic web environment for building autonomous agents. In *International Conference on Learning Representations*, 2024. arXiv:2307.13854.

Zhu, K., Shi, W., Chen, J., Pfeffer, J., Xie, X., Molloy, I. M., Goth, P. D., et al. MELON: Indirect prompt injection defense via masked re-execution and tool result comparison. In *International Conference on Machine Learning*, 2025. arXiv:2502.05174.

Zong, Y., Bohdal, O., Yu, T., Yang, Y., and Hospedales, T. Safety fine-tuning at (almost) no cost: A baseline for vision large language models. In *International Conference on Machine Learning*, 2024. VLGuard. arXiv:2402.02207.

A. Benchmark Descriptions and Access

OSWorld (Xie et al., 2024). A benchmark of 369 real OS tasks across macOS, Windows, and Linux, requiring agents to interact with real GUI applications (Chromium, Firefox, LibreOffice, VS Code, etc.) using screenshots and accessibility trees. Human performance is 72.4%; the best automated agent achieves approximately 12% on the full task set. For CUA safety evaluation, OS-Harm (Kuntz et al., 2025) extends OSWorld with 150 safety-critical tasks across misuse, injection, and misbehavior categories.

VisualWebArena (Koh et al., 2024). An extension of WebArena (Zhou et al., 2024) with visual-grounding tasks. Particularly relevant for perception-layer evaluation because it includes tasks that require understanding images embedded in web pages, product listings, and forum posts. The pop-up attack of Zhang et al. (2025b) is evaluated on VisualWebArena.

AgentDojo (Debenedetti et al., 2024). A dynamic benchmark with 97 tasks and 629 injection cases, designed specifically for evaluating prompt injection attacks and defenses. MELON (Zhu et al., 2025) achieves state-of-the-art defensive performance here. The benchmark currently evaluates text injection; extending it to visual injection (injected screenshots, adversarial web images) is an immediate research opportunity.

OS-Harm (Kuntz et al., 2025). A NeurIPS 2025 D&B Spotlight benchmark of 150 safety tasks for computer-use agents, covering misuse (agent exploited to perform harmful tasks), injection (agent manipulated by injected content), and misbehavior (agent violates safety constraints). Automated judge F1 score is 0.76–0.79. Recommended as the primary safety evaluation target for CUA perception-layer defenses.

AgentHarm (Andriushchenko et al., 2025). 110 malicious agent tasks across 11 categories, showing that LLMs in agentic contexts comply with harmful requests without jailbreaks. Provides a harmfulness baseline for evaluating how much perception-layer defenses reduce harmful compliance rates.

Agent Security Bench (Zhang et al., 2025a). The most comprehensive agent security evaluation infrastructure, covering 10 scenarios, 400 tools, 27 attack/defense combinations, and 13 LLMs. Provides rich cross-model and cross-attack comparison but has limited visual attack coverage.

B. Transfer Analysis: Estimation Methodology

The transfer analysis in Section 5 requires comparing defenses across three threat scenarios (static VLM jailbreak, CUA visual injection, CUA pop-up attack). Since no single paper evaluates all five defenses across all three scenarios, the CUA-scenario values are estimated using the following procedure.

Static VLM jailbreak ASR (column 1). Drawn directly from each paper's primary evaluation against the attack type for which the defense was designed. For Robust CLIP, this is adversarial transfer attacks on ImageNet classification; for SmoothVLM, patched adversarial prompts; for Llama Guard 3 Vision, MLCommons hazard benchmark classification.

CUA visual injection ASR (column 2). Estimated by analyzing structural mismatch between the defense's threat model and the CUA visual injection scenario. Key factors: (a) distribution mismatch between defense training data and CUA screenshots, (b) whether the defense's formal assumptions (e.g., patch bounds) cover the CUA attack modality, (c) any reported results on image-understanding tasks (as opposed to pure vision benchmarks) in the defense paper. All estimates have uncertainty of approximately $\pm 15\%$.

CUA pop-up attack ASR (column 3). Pop-up attacks exploit GUI-layer semantic deception rather than pixel-level perturbation. Defense effectiveness is estimated by analyzing whether the defense mechanism addresses semantic content interpretation (high relevance) vs. pixel-level statistics (low relevance). The baseline ASR of 86% is from Zhang et al. (2025b). SafeMobile's estimate is informed by its reported detection accuracy on mobile GUI adversarial sequences.

Recommended experimental protocol. A rigorous empirical evaluation of the transfer analysis would proceed as follows: (1) Instrument a SeeAct (Zheng et al., 2024) or OmniParser (Lu et al., 2024) agent on VisualWebArena and OSWorld; (2) Apply each defense (Robust CLIP encoder replacement, SmoothVLM screenshot preprocessing, LG3-Vision guardrail,

IH fine-tuning, SafeMobile chain detector) in separate ablation conditions; (3) Evaluate on three attack types: pixel-level adversarial images from Qi et al. (2024), fine-print injections from Chen et al. (2025a), and pop-up attacks from Zhang et al. (2025b); (4) Report ASR, task completion rate, and wall-clock latency overhead per condition. This experiment requires approximately 100–200 GPU-hours on a single A100, primarily for the SmoothVLM preprocessing ablation, and is the highest-priority empirical validation of the survey's transfer-gap claim.

C. Survey Scope and Exclusion Criteria

Included. We include papers that: (a) propose or evaluate attacks on multimodal or VLM-based systems where the attack vector involves visual inputs; (b) propose defenses for VLMs or multimodal models that operate on visual inputs; (c) study CUA systems specifically (GUI agents, web agents, OS agents); (d) provide benchmarks for evaluating CUA or VLM safety. Papers are included regardless of whether they engage with both communities, provided they are relevant to the perception-layer threat model.

Excluded. We exclude: (a) attacks or defenses operating exclusively on text modality with no visual component; (b) general LLM governance papers with no CUA- or VLM-specific content; (c) robotics or embodied AI papers whose threat models do not transfer to desktop/web GUI agents; (d) papers studying agents that use tool calling but not screen-based visual perception.