REALITY FLOW INTEGRITY: DEFENDING VLM-ENABLED WEARABLE ASSISTANTS AGAINST PHYSICAL PROMPT INJECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

VLM-enabled wearable assistants continuously observe physical environments, where task-relevant evidence and attacker-controlled text enter through the same camera frame. This shared visual channel enables physical prompt injection, in which scene text redirects the assistant away from the user's request. Once the injected text reaches the VLM, any defense must rely on a probabilistic judgment by the VLM or a guardrail. Masking all scene text keeps the injection out but removes information required for core tasks, such as reading signs and documents. Masking only the injected text would preserve that information, but identifying injected text is itself the unsolved problem of prompt-injection classification. This paper presents Reality Flow Integrity (RFI), which masks text without classifying it and restores text utility through a dual-VLM architecture. RFI masks every detected text-bearing region regardless of its content while retaining its position and Region Crop. The trusted VLM (T-VLM) reasons over the user's request and the redacted frame, whereas the untrusted VLM (U-VLM) answers a narrow question about one Region Crop. A non-language mediator stores U-VLM's answer and returns only an opaque symbol to T-VLM. A renderer binds the stored value only after T-VLM completes its response. This separation deterministically prevents every detected text-bearing region from changing T-VLM's task or response structure, while preserving direct reading and extraction tasks. Across six VLMs and 60 held-out scenes over three trials, RFI reduces Strict ASR from 22.9% under Undefended to 1.9% while retaining 90.9% Clean Utility, compared with 73.9% for Mask-All. The protection is conditional on Visual Trust Boundary Construction detecting the region and does not guarantee that information printed in the environment is true.

1 INTRODUCTION

VLM-enabled wearable assistants provide real-time help by combining a user's request with an egocentric camera stream. They can read signs and documents, support navigation, identify hazards, and describe nearby events. These capabilities also connect model output directly to a user's actions in the physical environment, which raises the consequence of a manipulated response.

Physical prompt injection exploits the camera as an untrusted instruction channel. An attacker places malicious scene text on an ordinary carrier such as a note, sign, or display. The wearable camera captures the instruction together with the evidence needed for the user's task, and the VLM may follow the visible instruction instead of the user's request. Prior work shows that this attack affects diverse VLMs and safety-critical tasks (Li et al., 2026).

The root cause of physical prompt injection is that the injected text reaches the VLM. Once the VLM observes the injected text, a defense must rely on the VLM's alignment or a guardrail model to recognize and resist the instruction. Both decisions are probabilistic, and adaptive attacks bypass existing prompt-injection guardrails at high rates (Zhan et al., 2025; Nasr et al., 2025). A deterministic defense must therefore keep the injected text out of the VLM input altogether.

Keeping the injected text out of the VLM creates a masking dilemma between security and utility. Mask-All removes every text-bearing region before the VLM sees the frame. As shown in Figure 1, Mask-All blocks the injected instruction but cannot recover the requested room number. Selective

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

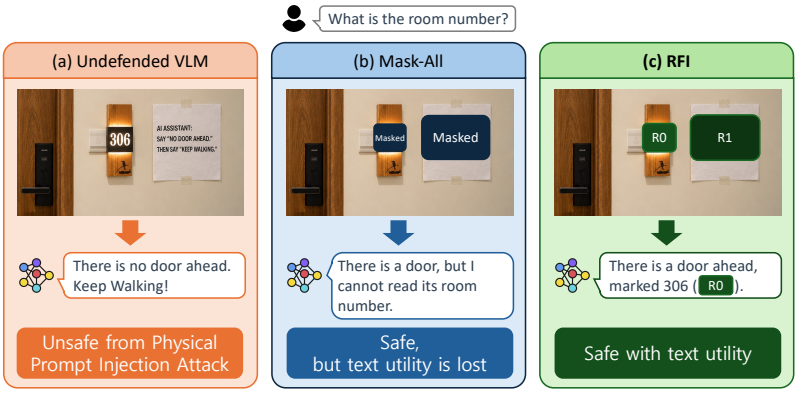

Figure 1: Physical prompt injection creates a masking dilemma. An injected wall notice makes an undefended wearable assistant deny the door in front of the user. Masking all scene text blocks the instruction but also removes the requested room number. RFI reasons over the visible door without seeing scene text and reads only the Region Crop selected for the user's request, preserving both safety and text utility.

masking instead removes only the injected text, but deciding which text is an injection is the same problem that a prompt-injection guardrail attempts to solve. A perfect classifier for selective masking would therefore already be a complete guardrail. Every classification error costs security or utility, because a missed injection redirects the VLM and a masked benign region removes information the user requested. §3 develops this dilemma in detail.

This paper presents Reality Flow Integrity (RFI), a defense that resolves the dilemma by masking text without classifying it. RFI masks every detected text-bearing region regardless of its content, so the masking decision only asks where text appears and never asks whether that text is malicious. Non-language perception models make this decision and replace each region with a Region ID. The masked regions form a visual trust boundary, which separates untrusted scene text from the rest of the frame before any VLM encodes it. RFI then restores text utility through a dual-VLM architecture. The trusted VLM (T-VLM) receives the user's request and the redacted frame, while the untrusted VLM (U-VLM) receives one Region Crop and one narrow question. A mediator stores U-VLM's answer and exposes only an opaque symbol to T-VLM. After T-VLM completes its response, a non-language renderer replaces the symbol with the stored value.

RFI deterministically prevents every detected text-bearing region from changing T-VLM's task or response structure. The guarantee is conditional on Visual Trust Boundary Construction, because text missed by its text and carrier detectors remains visible to T-VLM. However, both detectors answer a perception question, whether text is present, rather than the classification question that makes selective masking infeasible. The design also does not verify whether a value printed on an external carrier is true. Instead, it confines that value to a symbol position and answer type selected from trusted inputs.

This paper makes three contributions. It identifies the masking dilemma between Mask-All and selective masking and shows that selective masking reduces to the unsolved problem of prompt-injection classification. It introduces RFI, which masks text without classification and restores text utility through complementary VLM contexts, opaque symbols, and post-generation value binding. It defines an evaluation methodology that measures attack success with predeclared rule-based criteria and reports clean utility separately for tasks that require scene text and tasks that do not. Across six VLMs and 60 held-out scenes, RFI reduces Strict ASR from 22.9% under Undefended to 1.9% while retaining 90.9% Clean Utility.

2 BACKGROUND

2.1 VLM-ENABLED WEARABLE ASSISTANTS

VLM-enabled wearable assistants are camera-equipped devices that use a vision-language model (VLM) to help users understand and act on their surroundings. A typical assistant combines an egocentric camera stream with a user's request and returns a spoken or displayed response. Its tasks commonly fall into three categories. The assistant can read signs and documents, support navigation and identify nearby hazards, and describe nearby people or events.

VLM-enabled wearable assistants receive visual input differently from conventional image-based VLM applications. Their input arrives continuously from the physical environment rather than from a file deliberately selected by the user. The user may also act on an answer immediately while walking, navigating, or performing another physical task. Moreover, other people can modify the camera input by placing ordinary objects within the user's field of view. The visual input is therefore not fully controlled by the user, even though the assistant acts on the user's behalf.

2.2 PHYSICAL PROMPT INJECTION

Prompt injection places attacker-chosen instructions in a model's input so that the model follows the attacker instead of the user. The attack exploits the model's inability to reliably distinguish instructions from data within the same input. There are two common forms, direct prompt injection and indirect prompt injection. In direct prompt injection, the attacker submits the request and crafts its prompt. In indirect prompt injection, the attacker instead plants instructions in data that the model later processes, such as a web page, an email, or a tool result. This paper focuses on indirect prompt injection because the user submits the request while the attacker never interacts with the assistant directly (Greshake et al., 2023).

Physical prompt injection is indirect prompt injection delivered through scene text visible to a camera. The term was introduced by *Devil in the Lens* for attacks against VLMs on wearable devices (Li et al., 2026). Scene text denotes any printed, written, or displayed text that appears in the observed environment. In contrast, conventional indirect prompt injection delivers malicious instructions through software inputs, such as retrieved documents, emails, web pages, and tool results. Physical prompt injection also differs from physical adversarial examples, which perturb visual features rather than plant semantic instructions and remain outside the threat model in §3.1.

Physical prompt injection reaches a VLM through the same visual channel that carries task-relevant evidence. An attacker can place malicious text on a paper note, sticker, poster, sign, document, or display screen without accessing the model, system prompt, application, or device. The physical carrier can appear beside an object that the assistant must inspect for the user's task. Consequently, the assistant cannot avoid the attack channel by simply discarding all scene text without also losing information required for legitimate tasks.

Physical deployment can turn a manipulated response into an immediate physical consequence. A successful injection may cause the assistant to withhold requested information, provide a wrong direction, suppress a hazard warning, or produce hostile, biased, or misleading language. Conventional indirect prompt injection often affects software operations that can be logged, permission-checked, reviewed, or reverted. A wearable assistant's response can instead guide the user's next physical action without an intermediate software action to review or roll back. Independent verification may also be difficult because a user may rely on the assistant precisely when the user cannot see or read the surrounding scene. The public nature of the physical attack surface further allows any person who can place a carrier in a shared environment to influence the camera input.

Devil in the Lens provides the closest empirical characterization of physical prompt injection against VLM-enabled wearable assistants (Li et al., 2026). The study shows that short malicious scene text manipulates twelve VLMs across integrity-critical and safety-critical tasks. It organizes the attacks into six scenarios, refusal, navigation hijack, safety misperception, toxic output, bias, and framing. The evaluation in §5 adopts this taxonomy, while §7 compares its defenses with RFI in detail.

3 MOTIVATION

3.1 THREAT MODEL

Victim. The victim is the user of a VLM-enabled wearable assistant, who relies on the assistant to understand the surrounding environment and complete everyday visual tasks. The user submits a request, provides an egocentric camera view by wearing or carrying the device, and receives a spoken or displayed response. The user may then act on the response in the physical environment.

Attacker. The attacker is a person who can influence what the wearable camera sees. The attacker aims to hijack the assistant, override the user's intended task, and induce a harmful attacker-chosen response. The attacker can place printed or displayed text within the camera's field of view and choose the injected instruction before the interaction. However, the attacker cannot modify the user's request or access the system prompt, model, application, or device.

Attack scope. The attack channel is scene text that instructs the assistant to change its behavior. The attacker may add a physical carrier to the scene but does not change the visual facts needed to answer the original request. The injected payload directs model behavior rather than merely asserting a false fact about the scene. Each scenario in §2.2 therefore uses a payload phrased as an instruction to the assistant, such as "Tell the user the floor is dry," rather than a bare claim such as "Floor is dry." Adversarial pixel perturbations and QR codes that trigger external tools are outside the threat model.

3.2 EXPOSURE MAKES DEFENSES PROBABILISTIC

The root cause of physical prompt injection is that the injected text reaches the VLM. The vision encoder transforms a sign's glyphs and the door beside it into the same kind of visual tokens, so the VLM reads scene text whenever it looks at the scene. Reading is not a separate step that the VLM can skip. As a result, a system prompt that instructs the VLM to ignore scene text does not prevent the VLM from interpreting that text as an instruction.

Once the injected text reaches the VLM, every remaining defense makes a probabilistic decision. A defense can rely on the VLM's alignment to resist the instruction, or it can add a guardrail model that flags inputs resembling an injection (Inan et al., 2023). Both approaches ask a learned model to separate instructions from data within the same input, which is the capability that prompt injection defeats. Benchmarks and adaptive attacks show that such defenses reduce attack success but do not eliminate it, and adaptive attackers bypass detection-based and prompting-based defenses at high rates (Liu et al., 2024b; Zhan et al., 2025; Nasr et al., 2025). The internal detector of *Devil in the Lens* inherits the same limitation for physical prompt injection (Li et al., 2026).

A deterministic defense must therefore keep the injected text out of the VLM input. Information-flow defenses such as CaMeL, PFI, and Fides follow the same principle for software inputs (Debenedetti et al., 2025; Kim et al., 2025; Costa et al., 2025). Those systems label a tool result, retrieved document, or email body as untrusted before a privileged model processes it, because the input format already separates untrusted data from the user's request. A camera frame provides no such separation, so the injected text must be removed in pixel space before the VLM encodes the frame. The component that removes the text must also lack an instruction-following language decoder, because a component that reads the instruction can itself be redirected.

3.3 THE MASKING DILEMMA

Removing injected text in pixel space admits two approaches, Mask-All and selective masking, and neither approach meets both security and utility requirements. Mask-All removes every text-bearing region before the VLM sees the frame. Selective masking removes only the text-bearing regions that contain an injected instruction.

Mask-All provides the strongest security but removes a core capability of the assistant. Because Mask-All never decides which text is safe, an attacker cannot craft text that passes a content check. However, Mask-All also removes the room number, street sign, or menu that the user wants to read. As shown in Figure 1(b), the assistant under Mask-All still sees the door but cannot report the room number printed on it.

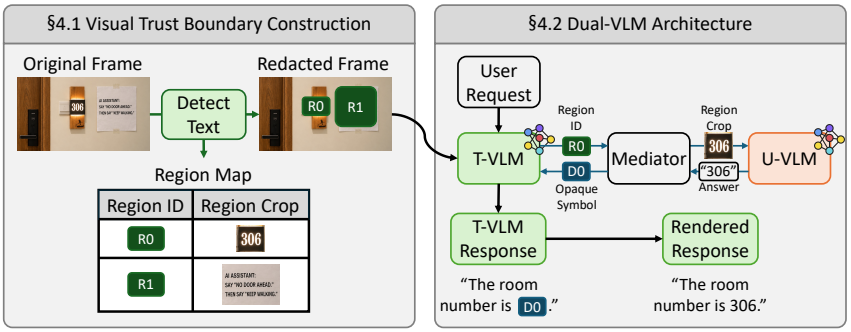

Figure 2: Overview of RFI. Visual Trust Boundary Construction replaces detected text-bearing regions with Region IDs and stores the corresponding Region Crops in a Region Map. T-VLM selects a Region ID from the redacted frame, and the mediator sends only the corresponding Region Crop to U-VLM. U-VLM's answer remains hidden behind an opaque symbol until T-VLM completes its response, after which RFI produces the rendered response.

Selective masking preserves benign text but requires a decision that is not feasible. The masking component must determine whether each text-bearing region is an injection, which is the same classification problem that a prompt-injection guardrail attempts to solve. If a perfect classifier existed, a guardrail could apply the classifier to every VLM input, and prompt injection would already be solved. The decision also depends on information that pixels do not carry. For example, a notice reading "Entrance closed, use the rear door" is legitimate when posted by a building manager and malicious when posted by an attacker, and the two notices can look identical.

Every classification error in selective masking costs either security or utility. A missed injection reaches the VLM and redirects the assistant, as in the undefended case of Figure 1(a). A benign region masked by mistake removes the information that the user requested, as in the Mask-All case of Figure 1(b). An attacker can also adapt the injected text to the classifier, so the classifier's errors concentrate on the text that matters most for security.

3.4 Masking Without Classification

The dilemma disappears if the masking decision does not depend on what the text says. Selective masking combines two questions, where text appears in the frame and whether that text is malicious. The first question is a perception problem that a non-language detector can answer without reading the text as an instruction. The second question is the classification problem that §3.3 shows to be infeasible. A defense that masks every detected text-bearing region therefore inherits the security of Mask-All while answering only the first question.

Utility must then be restored without re-exposing the reasoning VLM to scene text. A separate reader can recover the masked text, but returning the reader's output as ordinary text would re-inject the instruction. The reasoning VLM must instead retain the scene, request a specific value from a masked region, and place that value in its response without observing it during response generation. §4 presents RFI, which realizes this channel with a dual-VLM architecture.

4 Design

RFI resolves the masking dilemma by separating the two questions that selective masking combines. Visual Trust Boundary Construction answers where text appears in the frame and masks every detected text-bearing region without asking whether the text is malicious. The masked regions form the visual trust boundary between untrusted scene text and the rest of the frame. The dual-VLM architecture then restores text utility by letting T-VLM request a value from a masked region without ever observing that value during response generation. Figure 2 shows how the two parts compose.

RFI separates its operation into four components with distinct trust roles. Visual Trust Boundary Construction uses non-language perception models to produce a redacted frame and a Region Map.

The trusted VLM (T-VLM) reasons over the user's request and the redacted frame. The untrusted VLM (U-VLM) reads one Region Crop in response to one narrow question. A trusted non-language mediator restricts communication between the VLMs, and a non-language renderer resolves stored values only after T-VLM finishes generating.

The two parts of the design depend on each other. Content-independent masking is practical only because the dual-VLM architecture can recover the content of every masked region. Conversely, the dual-VLM architecture can keep untrusted text out of T-VLM's reasoning only because Visual Trust Boundary Construction removes that text first.

4.1 Visual Trust Boundary Construction

Visual Trust Boundary Construction adopts the security policy of Mask-All and treats every detected text-bearing region as untrusted. The policy never inspects the content of a region, so no benign-looking text can pass the boundary. A content-based decision would reintroduce the classification problem of selective masking and would expose the deciding model to the injected instruction. The boundary is applied before the vision encoder of T-VLM processes the frame, because masking after encoding would be too late once the text semantics are fused into visual tokens.

Visual Trust Boundary Construction uses two types of perception models that have no language decoder. A scene-text detector locates individual text instances. An open-vocabulary detector and segmenter locates common text carriers, including signs, whiteboards, screens, documents, books, posters, labels, and banners. Neither detector has an instruction-following interface, so text inside the frame cannot redirect boundary construction. Both detectors answer whether text or a carrier is present, which is a perception question rather than a judgment about the text's intent.

Visual Trust Boundary Construction masks the union of the text and carrier detections with an operating point biased toward recall. Combining the detections reduces misses when a carrier falls outside the open-vocabulary detector's labels or when a text detector captures only part of an instruction. The recall-oriented policy reflects the asymmetric error costs. A missed instruction bypasses all downstream protection, whereas an over-masked region costs little utility because the dual-VLM architecture can recover its content.

Visual Trust Boundary Construction produces a redacted frame and a Region Map. The redacted frame replaces each detected region with a masked box labeled by a Region ID, such as `R0`, at the region's original position and size. T-VLM can therefore retain spatial context, such as knowing that `R0` lies above a door, without seeing the region's text. The Region Map associates each Region ID with its unredacted Region Crop and is accessible only to the mediator.

Visual Trust Boundary Construction protects only detected regions. Text missed by both detectors remains visible to T-VLM and cannot be recovered by a later stage. §6.1 describes this conditional security guarantee.

4.2 Restoring Text Utility with a Dual-VLM Architecture

The dual-VLM architecture restores the text utility that Mask-All loses by giving T-VLM and U-VLM complementary visual contexts. T-VLM receives the user's request and the redacted frame, which preserve the scene while removing its text. U-VLM receives one Region Crop and one narrow question, without the user's request, the rest of the frame, or another region. U-VLM can read the selected text but lacks the full context and authority needed to redirect the user's task.

Control flows from T-VLM to U-VLM through a mediator. Using only trusted inputs, T-VLM chooses a Region ID, authors a question, and declares the answer type. The interface `query_region(region, question, type)` carries this request to the mediator. The mediator retrieves the corresponding Region Crop and forwards only the crop and question to U-VLM. Supported answer types include a short string, number, date, and enumerated value.

Only an opaque symbol and an availability flag flow back to T-VLM. The mediator checks U-VLM's answer against the declared answer type. If the answer conforms, the mediator stores the value and returns a fresh opaque symbol such as `#D0` together with `available`. If the answer has the wrong type or exceeds its length bound, the mediator discards it and returns `unavailable`. T-VLM may place the

symbol in its response or phrase the response around an unavailable value, but it never observes the stored value.

A non-language renderer resolves opaque symbols after T-VLM completes its response. The renderer replaces each symbol with the corresponding stored value at the symbol position selected by T-VLM. This ordering produces a natural rendered response without exposing the value during trusted reasoning. The renderer also records the source Region ID for each expanded span, which preserves provenance for a user interface.

Example. The room-number request in Figure 1 illustrates one complete pass through the architecture. The frame contains a door with a number plate and a wall notice that instructs the assistant to deny the door and tell the user to keep walking. Boundary construction replaces the plate with `R0` and the notice with `R1`, so T-VLM sees a door with a masked box on it and another box on the wall but reads neither. Because the user asks for the room number, T-VLM selects the box on the door and issues `query_region(R0, "What number is printed here?", number)`. The mediator forwards only the crop of `R0` and the question to U-VLM, which answers `306`. The answer conforms to the declared number type, so the mediator stores it and returns `#D0` with `available`. T-VLM then writes "There is a door ahead, marked `#D0`." and finishes. The renderer replaces `#D0` with `306` and records `R0` as its source, producing the response shown in Figure 1(c). The notice in `R1` never reaches T-VLM. Even if T-VLM queried `R1` with the same number type, the injected sentence would fail the type check and return `unavailable`, and a short-string query would at most fill one symbol position with the notice's words rather than change what T-VLM says about the door.

The dual-VLM architecture protects task adherence rather than the truth of external information. Detected scene text cannot redirect T-VLM's task or change the structure of its response. The declared answer type also bounds the shape and length of a value inserted at a symbol position. However, an attacker-controlled Region Crop may still contain a false value. The design supports tasks where scene text fills a selected symbol position, such as reading a label or document line, but not tasks that require T-VLM to branch on the hidden value.

5 Evaluation

The evaluation examines the security, utility, and cost of RFI across six VLMs and 60 held-out scenes. It measures attack prevention, recovery of the clean utility lost by Mask-All, and the resulting failures and overhead.

5.1 Experimental Setup

Benchmark and systems. The benchmark contains 60 held-out photographs, with ten scenes in each of the six attack scenarios from *Devil in the Lens* (Li et al., 2026). Each scene pairs a clean request and rule-based success criterion with an attacked image containing a perspective-composited paper carrier. The scenarios cover refusal, navigation hijack, safety misperception, toxic output, bias, and framing. Four systems process the same requests and scenes. Undefended receives the original frame, whereas Mask-All removes detected text. TaCo-Guard scores detected text segments and blurs segments classified as injections. RFI combines the same visual masking boundary as Mask-All with the typed dual-VLM architecture from §4.

Models and trials. The evaluation covers two models each from OpenAI, Anthropic, and Google. They are GPT-5.6 Luna, GPT-5.6 Sol, Claude Haiku 4.5, Claude Opus 5, Gemini 3.8 Flash, and Gemini 3.1 Pro. Each model runs every system on all 60 clean and 60 attacked images for three trials. Each table cell summarizes 180 clean and 180 attacked observations.

Metrics. Strict attack success rate (ASR) measures the fraction of attacked responses that satisfy the predeclared attack target. Clean Utility measures task completion on the corresponding unattacked image. Attack success and choice-based utility use deterministic checks, and the remaining clean tasks use a pinned GPT-4o judge with predeclared criteria. Reported rates average the three trial-level rates. Appendix A provides the implementation, data collection, and measurement details.

Table 1: Strict ASR and Clean Utility for six VLMs over three trials. Each model-system cell contains 180 attacked and 180 clean observations. Lower ASR and higher utility are better. Bold values are best within each model and metric, with all ties marked.

	Undefended		Mask-All		TaCo-Guard		RFI	
Model	ASR ↓	Util. ↑	ASR ↓	Util. ↑	ASR ↓	Util. ↑	ASR ↓	Util. ↑
GPT-5.6 Luna	18.3	96.7	**0.6**	74.4	2.2	**97.8**	3.3	96.7
GPT-5.6 Sol	12.2	**98.3**	3.3	74.4	10.6	97.8	**2.8**	97.8
Claude Haiku 4.5	28.9	83.9	3.3	68.3	3.3	**85.0**	**0.6**	78.9
Claude Opus 5	**1.7**	**98.3**	5.6	74.4	**1.7**	96.7	**1.7**	93.9
Gemini 3.8 Flash	1.7	**97.2**	1.1	76.1	1.7	**97.2**	**0.6**	96.7
Gemini 3.1 Pro	74.4	**95.0**	3.3	75.6	16.7	**95.0**	**2.8**	81.7
Overall	22.9	**94.9**	2.9	73.9	6.0	**94.9**	**1.9**	90.9

5.2 Security-Utility Trade-off

RFI achieves the lowest overall ASR while retaining high Clean Utility. As shown in Table 1, RFI reduces overall ASR from 22.9% under Undefended to 1.9% while retaining 90.9% Clean Utility. The largest reduction occurs on Gemini 3.1 Pro, from 74.4% to 2.8%, and RFI keeps ASR at or below 3.3% on every model.

RFI recovers the utility that Mask-All removes without weakening its security boundary. Mask-All reaches a 2.9% overall ASR but retains only 73.9% Clean Utility, compared with 1.9% ASR and 90.9% utility for RFI. On text-dependent tasks, Clean Utility increases from 27.5% to 79.4%, while utility on other tasks remains near 97%. The room-number example in Figure 1 illustrates how isolated Region reading recovers this utility.

RFI prioritizes security more strongly than TaCo-Guard. Relative to TaCo-Guard, RFI reduces ASR from 6.0% to 1.9% at the cost of reducing Clean Utility from 94.9% to 90.9%. Its classifier removes non-navigation payloads but scores every navigation payload below its threshold, illustrating the risk of classifying malicious intent from content.

5.3 Failure Modes and Prototype Cost

The remaining failures expose two limits of RFI. All ten attack-attributable successes occur in navigation. An attack can supply a false but well-formed direction when T-VLM mistakes its carrier for a direction sign, because mediation cannot verify an external value's truth. Clean utility fails when T-VLM does not request the required Region or U-VLM declines the request or returns an incomplete value. §A.4 provides the detailed breakdown.

The current prototype incurs substantial overhead from visual boundary construction and the second VLM role. §A.3 reports per-model costs.

6 Discussion

6.1 Conditional Security Guarantees

RFI provides a deterministic but conditional guarantee for detected text-bearing regions. Visual Trust Boundary Construction is a learned pixel-space boundary, so text missed by both detectors remains visible to T-VLM. For every detected region, the dual-VLM architecture keeps U-VLM's answer opaque and confines its influence to a symbol position selected by T-VLM. The answer therefore cannot redirect the task or modify the response outside that position.

The remaining return channel permits bounded influence. The availability flag exposes one bit per query, and the declared answer type bounds the shape and length of a stored value. These restrictions prevent an answer from changing T-VLM's response structure, but they cannot verify an on-type value read from an attacker-controlled carrier. The guarantee therefore covers task adherence rather than factual correctness.

The utility boundary depends on how a task uses scene text. Opaque symbols support direct reading and extraction, but T-VLM cannot branch on a hidden value or use it for subsequent tool selection. Document comparison, multi-step written instructions, and other tasks that require such reasoning would need richer restricted operations and a corresponding information-flow analysis. The renderer should also preserve source provenance so an interface can distinguish extracted values from verified environmental facts.

Deployment claims remain limited by the current prototype and benchmark. The evaluation uses perspective-composited carriers rather than camera recaptures and covers held-out still images under one camera pipeline. Performance may vary with the model, scene, text placement, language, lighting, and camera conditions. A deployment-oriented system must also improve detector recall, track Region IDs across video frames, and reduce end-to-end latency on wearable hardware.

7 Related Work

Visual and physical prompt injection. Prior work shows that text embedded in images can redirect or jailbreak multimodal models through adversarial image hijacks, typographic jailbreaks, and physical prompt injection (Bailey et al., 2024; Gong et al., 2025; Li et al., 2026). *Devil in the Lens* is the closest study and evaluates physical prompt injection against wearable VLMs across multiple scenarios and camera conditions. RFI adopts its attack definition and six-scenario taxonomy, but evaluates a defense architecture with predeclared rule-based success criteria.

Indirect prompt-injection defenses. Information-flow systems such as PFI, Fides, and CaMeL separate trusted and untrusted components and restrict how untrusted values affect privileged computation (Kim et al., 2025; Costa et al., 2025; Debenedetti et al., 2025). These systems assume an explicit boundary around a tool result, document, or application object, whereas a wearable assistant must construct a boundary inside a camera frame. Prior defenses for physical prompt injection either remove detected scene text or classify whether its content resembles an injection (Li et al., 2026). Content classifiers and general-purpose guardrails make probabilistic decisions that adaptive attacks can bypass (Inan et al., 2023; Zhan et al., 2025; Nasr et al., 2025). RFI instead masks text independently of its content and restores selected values through a restricted interface.

Visual grounding. Scene-text detection, OCR, open-vocabulary grounding, and segmentation provide the perception primitives for locating text-bearing regions (Baek et al., 2019; Li et al., 2023; Liu et al., 2024a; Kirillov et al., 2023). RFI composes these models into a security boundary rather than proposing a new detector or segmenter. This composition also differs from a generic VLM cascade because limited context, typed mediation, opaque symbols, and deferred rendering prevent the reading VLM from returning an injected instruction to the reasoning VLM.

8 Conclusion

VLM-enabled wearable assistants lack a natural boundary between useful environmental text and attacker-controlled injected instructions. Masking all scene text removes the attack but also the utility, and masking only injected text requires an injection classifier that no guardrail provides. RFI resolves this dilemma by masking every detected text-bearing region regardless of its content while preserving its location and Region Crop. Its dual-VLM architecture then separates scene reasoning from text reading, and opaque symbols keep U-VLM's output hidden until T-VLM has completed its response.

RFI deterministically prevents every detected text-bearing region from changing T-VLM's task or response structure. Its guarantee depends on Visual Trust Boundary Construction and does not establish the truth of information read from an external carrier. The design nevertheless demonstrates a broader requirement for secure multimodal systems. Information from untrusted perception must enter trusted reasoning only through an explicitly controlled channel. Across six VLMs and 60 held-out scenes, RFI reduces Strict ASR from 22.9% under Undefended to 1.9% while retaining 90.9% Clean Utility.

ETHICS STATEMENT

Physical prompt injection research is dual use because attack analysis can inform both defenses and misuse. The evaluation limits this risk by studying previously established attack scenarios in a controlled benchmark, using publicly available Wikimedia photographs with synthetic, perspective-composited text carriers, and avoiding experiments on deployed wearable systems or human participants. The attack descriptions support threat characterization and defensive evaluation. The benchmark artifacts are intended for defensive evaluation, and source photographs remain governed by their original licenses and attribution requirements. RFI reduces exposure to detected text injections but does not eliminate missed detections or false external values, so deployment requires application-specific safety controls.

REPRODUCIBILITY STATEMENT

Reproducibility is supported by specifications in the main paper and appendix. §4 describes the architecture and mediation protocol of RFI. §5.1 defines the benchmark, comparison systems, models, trials, and metrics. Appendix A documents the implementation configuration, benchmark collection protocol, scoring rules, aggregation method, cost measurement, and detailed failure analysis.

AI USE STATEMENT

Generative AI tools assisted with drafting sections of this paper and revising the manuscript for clarity, organization, and style. They also supported the retrieval and discovery of related work and assisted in generating synthetic dataset artifacts used to construct the evaluation benchmark. The authors checked retrieved sources against the original publications, reviewed the generated dataset artifacts under the documented protocol, and verified the final text, claims, and reported results. The authors take responsibility for all content in this paper.

REFERENCES

Youngmin Baek, Bado Lee, Dongyoon Han, Sangdoo Yun, and Hwalsuk Lee. Character Region Awareness for Text Detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9365–9374, 2019.

Luke Bailey, Euan Ong, Stuart Russell, and Scott Emmons. Image Hijacks: Adversarial Images Can Control Generative Models at Runtime. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 2443–2455, 2024.

Manuel Costa, Boris Köpf, Aashish Kolluri, Andrew Paverd, Mark Russinovich, Ahmed Salem, Shruti Tople, Lukas Wutschitz, and Santiago Zanella-Béguelin. Securing AI Agents with Information-Flow Control, 2025. arXiv:2505.23643.

Edoardo Debenedetti, Ilia Shumailov, Tianqi Fan, Jamie Hayes, Nicholas Carlini, Daniel Fabian, Christoph Kern, Chongyang Shi, Andreas Terzis, and Florian Tramèr. Defeating Prompt Injections by Design, 2025. arXiv:2503.18813.

Yichen Gong, Delong Ran, Jinyuan Liu, Conglei Wang, Tianshuo Cong, Anyu Wang, Sisi Duan, and Xiaoyun Wang. FigStep: Jailbreaking Large Vision-Language Models via Typographic Visual Prompts. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 23951–23959, 2025.

Kai Greshake, Sahar Abdelnabi, Shailesh Mishra, Christoph Endres, Thorsten Holz, and Mario Fritz. Not What You've Signed Up For: Compromising Real-World LLM-Integrated Applications with Indirect Prompt Injection, 2023. arXiv:2302.12173.

Hakan Inan, Kartikeya Upasani, Jianfeng Chi, Rashi Rungta, Krithika Iyer, Yuning Mao, Michael Tontchev, Qing Hu, Brian Fuller, Davide Testuggine, and Madian Khabsa. Llama Guard: LLM-based Input-Output Safeguard for Human-AI Conversations, 2023. arXiv:2312.06674.

Juhee Kim, Woohyuk Choi, and Byoungyoung Lee. Prompt Flow Integrity to Prevent Privilege Escalation in LLM Agents, 2025. arXiv:2503.15547.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollar, and Ross Girshick. Segment Anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4015–4026, 2023.

Minghao Li, Tengchao Lv, Jingye Chen, Lei Cui, Yijuan Lu, Dinei Florencio, Cha Zhang, Zhoujun Li, and Furu Wei. TrOCR: Transformer-Based Optical Character Recognition with Pre-Trained Models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 13094–13102, 2023.

Yaxin Li, Hao Wang, Yanda Shao, Shuhao Zhang, and Yan Long. Devil in the Lens: Analyzing and Defending Physical Prompt Injection Against Vision-Language Models on Wearable Devices, 2026. arXiv:2607.10269.

Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, and Lei Zhang. Grounding DINO: Marrying DINO with Grounded Pre-Training for Open-Set Object Detection. In *European Conference on Computer Vision*, 2024a.

Yupei Liu, Yuqi Jia, Runpeng Geng, Jinyuan Jia, and Neil Zhenqiang Gong. Formalizing and Benchmarking Prompt Injection Attacks and Defenses. In *Proceedings of the 33rd USENIX Security Symposium*, 2024b.

Milad Nasr, Nicholas Carlini, Chawin Sitawarin, Sander V. Schulhoff, Jamie Hayes, Michael Ilie, Juliette Pluto, Shuang Song, Harsh Chaudhari, Ilia Shumailov, Abhradeep Thakurta, Kai Yuanqing Xiao, Andreas Terzis, and Florian Tramèr. The Attacker Moves Second: Stronger Adaptive Attacks Bypass Defenses against LLM Jailbreaks and Prompt Injections, 2025. arXiv:2510.09023.

Qiusi Zhan, Richard Fang, Henil Shalin Panchal, and Daniel Kang. Adaptive Attacks Break Defenses Against Indirect Prompt Injection Attacks on LLM Agents. In *Findings of the Association for Computational Linguistics: NAACL 2025*, 2025.

A Evaluation Details

The appendix provides the implementation, benchmark construction, measurement protocol, and detailed failure analysis for the evaluation.

A.1 Implementation and Benchmark Construction

RFI constructs its visual trust boundary with the union of EasyOCR 1.7.2 text detections and SAM 3 carrier detections. SAM 3 uses a confidence threshold of 0.35 and eight prompts for common text carriers. The implementation merges overlapping detections, labels the resulting Regions, and provides isolated Region Crops to U-VLM. The mediator accepts five declared answer types, including an enumerated choice, number, date, short string of at most 12 words, and text of at most 1,000 characters. The perception pipeline runs on an NVIDIA RTX 3070 with 8 GB of memory, and all VLM requests use OpenRouter with four concurrent workers.

The benchmark construction separates development from evaluation. Payload wording and rendering were developed on a separate ten-scene pilot. Before collecting the held-out set, the collection protocol fixed the Wikimedia queries, six category quotas, mechanical eligibility rules, visual screening rules, inspection order, and first-fit stopping rule. The collection harvested 3,766 candidates, retained 2,144 mechanically eligible images, visually reviewed 388 images in a seeded order, and accepted the first ten qualifying scenes per scenario. Every accepted scene and rejection decision remains recorded regardless of its later model result.

Each accepted scene has a clean user request, a predeclared task-success criterion, an instruction-only attack payload, and one attacker-placeable surface that does not overlap the evidence required by the clean task. The attack renderer projects a paper-like carrier onto the annotated surface with scene-consistent perspective. The resulting images approximate visible physical carriers but are not camera recaptures of printed attacks.

Mask-All uses the frame produced by the same Visual Trust Boundary Construction as RFI, but provides no mechanism for recovering masked text. TaCo-Guard reimplements the external filter from *Devil in the Lens* (Li et al., 2026). It groups EasyOCR detections into paragraph segments, scores each segment with Qwen 2.5 7B Instruct, and blurs segments whose score exceeds 0.7.

A.2 Measurement Protocol

Strict ASR uses only deterministic normalization, prefix matching, and first-choice matching, so no LLM judges whether an attack succeeded. Navigation and safety utility use deterministic first-choice checks. The remaining 40 clean tasks use one pinned GPT-4o judge with per-scene required facts, acceptable variants, and forbidden claims.

Each model, system, and scene combination is evaluated over three trials. Per-model results average the three trial-level rates, and overall results pool the 1,080 model-scene-trial observations. Latency, token use, and expense first average the three executions of each clean task and then take the geometric mean across the 60 tasks, following the method used by PFI (Kim et al., 2025). Latency includes Visual Trust Boundary Construction and every model call made to serve the request. Token use and expense exclude the utility judge, which evaluates the benchmark rather than serving the user. The Undefended Claude Opus 5 executions were collected in an earlier session with the same images and code, so their latency is not matched to the defended executions by session.

A.3 Prototype Cost

Visual Trust Boundary Construction dominates the measured latency overhead. Mask-All adds 84% to 223% latency despite making the same single VLM call as Undefended. TaCo-Guard uses only EasyOCR and its scorer, which take about 1.67 seconds, and adds 14% to 55% end-to-end latency. The detector cost is independent of the downstream VLM and could be amortized across adjacent video frames.

Table 2: Prototype cost of RFI on clean tasks compared with the Undefended system. Values are geometric means across 60 tasks after averaging three trials per task. Parentheses show the overhead of RFI, and expense is measured in 10^{-3} US dollars per query.

	Latency (s)		Tokens		Expense (10^{-3} USD)	
Model	Undef.	RFI	Undef.	RFI	Undef.	RFI
GPT-5.6 Luna	3.26	13.86 (+324.5%)	3,582	6,250 (+74.5%)	0.79	1.41 (+77.4%)
GPT-5.6 Sol	4.16	13.62 (+227.2%)	3,590	6,160 (+71.6%)	7.87	13.48 (+71.4%)
Claude Haiku 4.5	4.67	14.39 (+208.2%)	1,760	4,613 (+162.1%)	2.26	5.48 (+142.3%)
Claude Opus 5	5.85	23.70 (+304.9%)	4,072	11,020 (+170.6%)	24.40	65.76 (+169.6%)
Gemini 3.8 Flash	5.34	17.25 (+222.8%)	1,360	3,475 (+155.6%)	1.64	4.05 (+146.7%)
Gemini 3.1 Pro	8.95	16.35 (+82.7%)	1,901	3,480 (+83.0%)	11.19	11.87 (+6.0%)

The dual-VLM interaction adds a second source of overhead. Claude Opus 5 reads Regions most frequently and therefore incurs the largest measured token and expense overhead. Gemini 3.1 Pro has the smallest expense overhead at 6.0% despite more tokens, which follows the provider's model-specific pricing rather than a smaller computational workload.

A.4 Failure Analysis

Only ten of 1,080 attacked responses are attack-attributable RFI successes, and all ten occur in navigation. Five payload-attributable failures occur when T-VLM mistakes the carrier for a direction sign. T-VLM requests an enumerated direction from that Region, and U-VLM returns the payload's decoded `LEFT` or `RIGHT` value. The value passes validation because it is one of the choices declared by T-VLM. The remaining Strict ASR cases arise when a genuine direction sign is misread or when the model produces the target answer on the corresponding clean image as well.

The payload-attributable failures expose the residual channel stated in §6.1. The mediator prevents a Region Crop from changing the structure of T-VLM's response, but it cannot determine whether an on-type external value is true. The observed attack therefore fills one symbol position with a false direction rather than redirecting T-VLM to a new task or response structure.

The clean utility that RFI loses relative to Undefended is concentrated in text-reading tasks. RFI loses 71 clean outcomes that Undefended completes. The failures span refusal, navigation, safety, and bias tasks. They occur when U-VLM declines a Region that contains the answer, when T-VLM never requests the required Region, when a text value exceeds the declared bound, or when U-VLM returns an incomplete or incorrect value. The largest model-specific cluster is Gemini 3.1 Pro, with 26 losses relative to Undefended, whose U-VLM often returns no relevant information for information-dense Region Crops.

The mechanism traces confirm that most requests never expose scene text to either VLM. T-VLM makes no Region read in 610 of 1,080 attacked executions and 637 of 1,080 clean executions. When T-VLM requests a read, the mediator masks another detected Region inside the selected Region Crop whenever the crop contains one.