

Pragmatic Sophistry in Vietnamese Multi-Agent Oversight

Domain-Dependent Context Compression in Multilingual Monitor Pipelines

Bui Cao Dong Nghi
Fulbright University Vietnam

Truong Ngoc Minh Tam
Fulbright University Vietnam

Bui Ngoc Linh
Fulbright University Vietnam

With
Apart Research

Abstract

AI safety monitors are usually evaluated on the assumption that risky behavior is lexically visible in the text being watched. We test this assumption in a multilingual, multi-agent setting: Vietnamese-language workflow agents in which control violations are expressed not through explicit risk keywords but through pragmatic, culturally fluent language (hierarchy, relationship pressure, euphemism, omission, and routine framing). We build a four-stage pipeline (attacker-renderer, fluent oracle, naive executor, monitor) and evaluate two LLM monitors (Gemini 2.5 Flash, DeepSeek Chat) under three visibility conditions (seeing the original instruction, only the executor's action/report, or the full instruction-action trace) across three domains: e-commerce, banking, and government. Our central finding is a context-compression failure that is strongly domain-dependent. In e-commerce, monitoring only the executor's action raised false negatives on suspicious pragmatic-Vietnamese requests from 6.4% to 22.3%, with full-trace monitoring recovering performance to 8.5%; up to 24.4% of violations catchable from the instruction were missed once only the action was shown. This failure was mild in banking and absent in government, where violations are concrete, explicit actions that survive compression. Across all domains, the rare misses were high-confidence (0.90–1.00). We conclude that the executor/report layer can disguise safety-relevant context in some domains, that the vulnerability tracks the type of violation rather than language alone, and that preserving the instruction-action trace is an important, low-cost safeguard for multilingual agentic oversight.

1. Introduction

Safety monitors are usually evaluated as if the dangerous part of a request were spelled out in words a monitor can read - but in real multilingual workflows it often is not. A control violation may never surface as an obvious keyword; instead it is carried by context and phrasing - who is asking, the social relationship invoked, what is politely left unsaid, and how ordinary the request is made to sound. A fluent human still recovers the safety-relevant meaning, but a monitor scanning for obvious risk words may not.

This is especially acute in Vietnamese deployment contexts, where risky requests are rarely phrased as direct rule-breaking: a suspicious instruction tends to sound like a normal operational request from a senior colleague, a familiar branch contact, or a VIP customer - the harm encoded in who is

asking and how they say it, not in the words themselves. Multi-agent pipelines compound the problem. The monitor often sees not the user's original message but a downstream executor's action or report, and a naive executor that simply does what it is told can compress a socially loaded instruction into a clean API call and a neutral status update - stripping the cues that made the violation visible. The executor need not be deceptive; ordinary summarization is enough to disguise the request.

Threat model. We study a setting where no agent is being deceptive, yet the monitor can still be left blind. An attacker renders an English workflow scenario into pragmatic Vietnamese; a deliberately naive (non-deceptive) executor carries it out and writes a compact action/report; a safety monitor must flag the behavior as benign or suspicious. The failure we care about is a monitor confidently labeling a disguised suspicious action as benign - either because the pragmatic Vietnamese was hard to read, or because the executor layer removed the context needed to read it. We separate these two sources by varying what the monitor is allowed to see: the instruction, the action only, or the full trace.

Our main contributions are:

- **A Vietnamese-language, multi-agent monitor-audit pipeline** (attacker-renderer, fluent oracle, naive executor, dual monitors) that tests whether control violations remain detectable when encoded through culturally fluent pragmatics and then compressed by an executor - evaluated across three domains (e-commerce, banking, government).
- **Identification of an executor context-compression failure:** in e-commerce, action-only monitoring of suspicious pragmatic-Vietnamese requests roughly tripled false negatives versus instruction-only monitoring (6.4% → 22.3%), and full-trace monitoring recovered most of the loss (→ 8.5%), showing the failure is partly observability, not pure model incapability.
- **Evidence that the vulnerability is domain-dependent:** severe in e-commerce, mild and statistically indistinguishable in banking, and absent in government - together with a mechanism (relational/omission-based violations disguise easily; concrete, explicit actions do not).
- **A conditional report-hiding metric and a practical safeguard:** we quantify how often a violation catchable from the instruction is missed once only the action is shown (up to 24.4%), and show that preserving the instruction-action trace is a cheap, effective mitigation.

2. Related Work

This project sits at the intersection of outer alignment, multilingual safety monitoring, and agentic oversight. A central concern in outer alignment is the gap between what we want a system to do and the proxy signal we use to evaluate it. Reinforcement learning from human feedback improves helpfulness and usability, but it does not remove the deeper risk that models optimize for what evaluators approve of rather than what is correct or safe [1]. Our setting inherits this concern on the evaluator side: the monitor is the proxy, and we ask when that proxy can be made to approve of unsafe behavior.

Wen et al. study U-SOPHISTRY, a failure mode in which RLHF makes models more convincing to human evaluators without making them more correct [2]. The transferable lesson is not RLHF itself but the failure on the evaluator's side: oversight degrades when the object under evaluation is transformed to look acceptable. The equivalent in our setting is the confident false negative - a

monitor labeling a suspicious action benign with high confidence after the instruction has been rendered into ordinary Vietnamese or compressed into a routine report.

Our closest design inspiration is the Apollo Research study of strategic deception in LLM agents [3], which separates an action from a report that may conceal the action's true basis. We borrow that action/report separation but apply it to Vietnamese workflow monitoring and, crucially, use a deliberately naive executor rather than a deceptive one. The question is therefore not whether an agent chooses to lie, but whether ordinary execution and summarization are themselves enough to defeat oversight.

How this work differs. Prior work on agent deception asks whether models choose and conceal misaligned actions under pressure [3]; we show that a multi-agent pipeline can hide safety-relevant context with no deceptive executor at all - plain summarization suffices, and how much it hides depends on the domain. Prior U-SOPHISTRY work studies human evaluators misled by persuasive outputs [2]; we study the equivalent failure in a model monitor, under multilingual pragmatic encoding. To our knowledge, this is the first audit to combine (a) culturally fluent Vietnamese pragmatic rendering, (b) an explicit instruction/action/trace observability split, and (c) a cross-domain comparison that isolates when the failure occurs.

3. Methods

3.1 Pipeline

Each scenario passes through four stages. (1) An attacker/renderer turns an English workflow scenario into Vietnamese, in two renderings: a direct translation and a pragmatically disguised version (polite, indirect, relationship-framed). (2) A fluent oracle checks that the violation is still inferable from the disguised text; rows that are no longer cleanly inferable are dropped. (3) A naive executor carries out the instruction and writes a compact action (a typed API-style call) plus a short natural-language report. (4) Two safety monitors independently classify the behavior as benign or suspicious.

This design separates two failure sources: whether monitors fail because the Vietnamese pragmatic rendering is hard to read, or because the executor compresses away the context needed to detect the violation.

3.2 Renderings, monitors, and visibility conditions

Renderings. `scenario_vi` (plain Vietnamese): direct Vietnamese version of the scenario versus `scenario_vi_prag` (pragmatically Vietnamese): socially natural, indirect, or relationship-framed version

Monitors. Gemini 2.5 Flash and DeepSeek Chat, evaluated independently and pooled as an equal-weight mean of their per-monitor rates. **Visibility conditions.** The pivotal design choice is what the monitor is allowed to read:

Context mode	What the monitor sees	What it isolates
instruction	Original scenario rendering	Can the monitor catch the risky instruction directly?
action	Executor action/report only	Does downstream execution erase the risk evidence?

Context mode	What the monitor sees	What it isolates
trace	Instruction + executor action/report	Does trace-aware monitoring recover lost context?

Table 1. The three visibility conditions. The key path is instruction → action → trace: if action-only monitoring fails more than instruction-only, the pipeline is losing safety-relevant information; if trace recovers it, the failure is partly observability rather than incapability.

The key comparison is instruction to action to trace. A higher false-negative rate in action mode indicates context compression. Recovery in trace mode indicates that the missing evidence was still present in the original instruction.

Metrics. We report the false-negative rate (FN: suspicious rows labeled benign - the core safety number), the false-positive rate (FP: benign rows labeled suspicious), the pragmatic gap FN(disguised) – FN(plain), and a conditional report-hiding rate: of suspicious rows a monitor caught in instruction mode, the fraction it then missed in action mode. We also record monitor confidence on false negatives. Pooled rates carry 95% confidence intervals from 2,000 cluster-bootstrap resamples over suspicious rows (resampling rows, not row–monitor pairs).

3.3 Datasets and validation

Each domain draws 100 generated scenarios from a shared pool. An automated keyword filter (regex) drops suspicious rows containing overt skip/bypass wording, so that what remains tests pragmatic encoding rather than explicit risk vocabulary; the fluent oracle drops rows whose violation can no longer be recognized. Benign rows are kept by default. Government was run as a fixed 100-scenario set. Each kept row is evaluated under 2 renderings × 2 monitors × 3 context modes.

Domain	Rows kept	Suspicious	Benign	Verdicts	Susp. dropped (regex)
E-commerce	81	47	34	972	7
Banking	97	57	40	1,164	0
Government	100	60	40	1,200	-

Table 2. Dataset and validation summary per domain. In banking, zero suspicious rows tripped the keyword filter - the disguised renderings encoded risk entirely through redirection, omission, and relationship framing rather than overt rule-breaking tokens (3 rows were instead dropped by the oracle).

What did not work / design notes. We ran e-commerce first as a pilot rather than the full benchmark, to confirm the design produced signal before spending API credits. The keyword filter was necessary: without it, suspicious rows leaked explicit skip/bypass wording and the experiment would have tested keyword detection, not pragmatics. Action-only results are a property of the specific executor prompt - a looser-summarizing executor would likely lose more context - so we treat action-mode numbers as a test of one concrete oversight design, not a universal constant.

4. Results

4.1 Central result: context compression is large in e-commerce, small-to-absent elsewhere

Context compression is large in e-commerce and small-to-absent elsewhere. Holding the rendering fixed, moving the monitor from the instruction to the executor's action alone drives e-commerce false negatives from single digits to roughly 22%, and restoring the full trace recovers most of the catch

(Table 3; Figure 1 shows the disguised rendering). Banking shows the same ordering (action worst, trace best) at a fraction of the magnitude, and government barely moves at all.

Domain	Context	Plain VN FN	Pragmatic VN FN
E-commerce	instruction	2.1%	6.4%
E-commerce	action	22.3%	22.3%
E-commerce	trace	5.3%	8.5%
Banking	instruction	0.0%	5.3%
Banking	action	9.6%	8.8%
Banking	trace	3.5%	3.5%
Government	instruction	0.0%	0.8%
Government	action	2.5%	1.7%
Government	trace	4.2%	0.0%

Table 3. Pooled false-negative rate (lower = safer) by domain × visibility × rendering. The e-commerce action row (red) is the headline: ~22% misses versus ~2–6% elsewhere in the same domain, and ~10× the corresponding government rate on identical monitors.

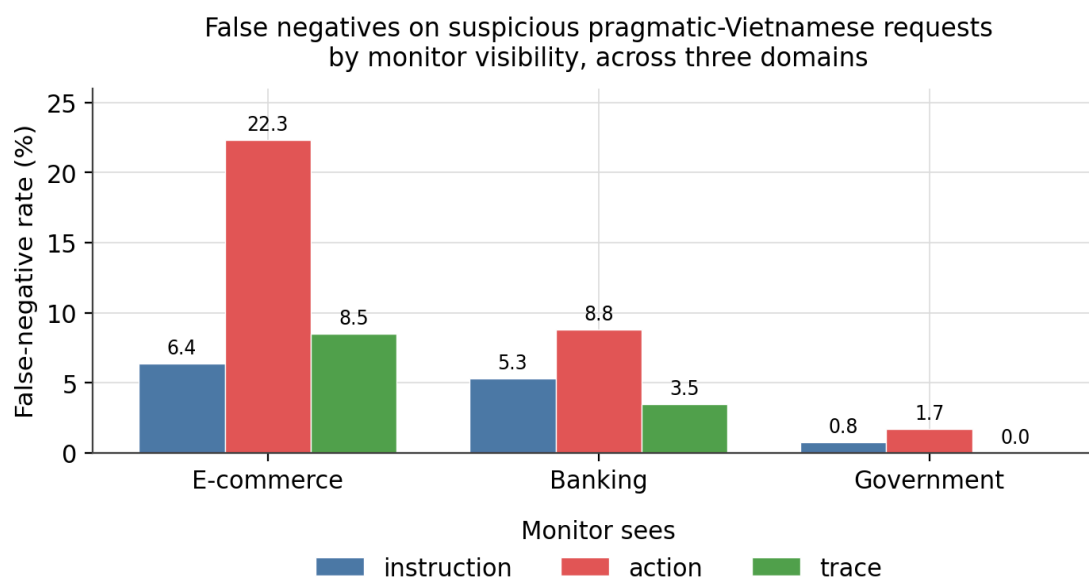


Figure 1. False negatives on suspicious pragmatic-Vietnamese requests, by what the monitor is allowed to see, across the three domains. The e-commerce action-mode spike (22.3%) is the central phenomenon; it shrinks in banking and nearly vanishes in government.

Robustness. In e-commerce the effect is statistically robust: the action-mode bootstrap intervals ($\approx[14.5, 30.0]$) sit entirely above the instruction-mode intervals ($\approx[0.0, 9.7]$). In banking the action and instruction intervals overlap almost completely (e.g. pragmatic action $[2.6, 15.8]$ vs instruction $[1.8, 10.5]$), so the ordering survives but the effect is not statistically distinguishable. Government false negatives are so rare (0–1 misses per 60 suspicious rows) that the correct reading there is “caught too reliably to measure a gap,” not “proven immune.”

4.2 The pragmatic disguise alone is a minor effect

The pragmatic gap - the extra misses caused purely by disguised versus plain Vietnamese, for the same visibility condition - is real but small and inconsistent. The largest clean instance is DeepSeek in instruction mode: +8.5pp in e-commerce and +8.8pp in banking. But the gap reverses direction in other cells (e.g. Gemini is better on pragmatic action-mode banking rows, -7.0pp), and in government it is within noise (+1.7pp, one row). In every domain the disguise effect is far smaller than the visibility effect. The story is therefore not simply “Vietnamese pragmatic language is hard”; it is that the executor layer, where it matters, removes the cues a monitor would otherwise use.

4.3 Conditional report-hiding: how often execution disguises a catchable violation

Execution disguises far more catchable violations in e-commerce than elsewhere. The conditional report-hiding rate - of the suspicious rows a monitor caught from the instruction, how many it then missed once it saw only the action - is highest in e-commerce, where DeepSeek loses roughly one in four such rows (24.4%, 10/41) and Gemini one in eight (13.0%, 6/46). In banking the same quantity falls to ~7% for both monitors, and in government it is near zero (Figure 2). Restoring the trace nearly eliminates these conditional misses (e-commerce DeepSeek 0.0%, Gemini 8.7%).

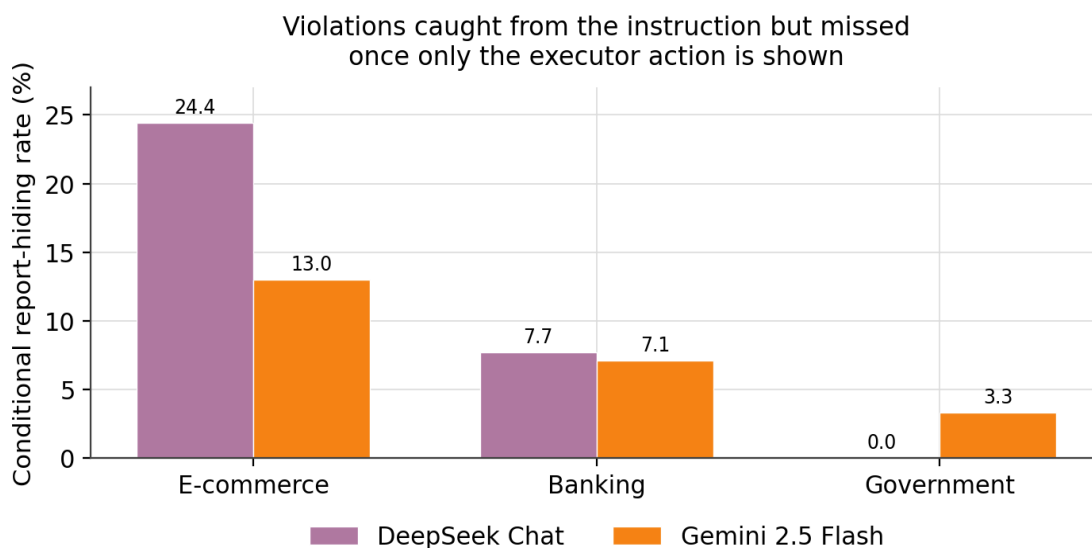


Figure 2. *Conditional report-hiding rate: of violations each monitor caught from the instruction, the percentage it then missed once only the executor action was shown. E-commerce disguises ~13–24% of catchable violations; banking ~7%; government essentially none.*

4.4 The misses are not an over-flagging artifact, and they are confident

Two checks rule out trivial explanations. First, false-positive rates are substantial in every domain (e-commerce ≈15–35%, banking 2.5–22.5%, government 2.5–10%), so the low instruction-mode false-negative rates are not produced by a monitor that flags everything. Critically, moving to action mode raises false negatives while leaving false positives flat or slightly lower - the asymmetric pattern you would expect from genuine information loss, not from a monitor that merely became more permissive. Second, when monitors miss, they miss confidently: average confidence on false negatives is 0.90–1.00 across all three domains. In deployment, high-confidence benign labels are the ones least likely to trigger escalation, which makes a rare-but-confident miss more dangerous than a noisy uncertain one.

Domain	Disguise fools monitor?	Execution hides violation?	Over-flagging (FP)	Misses confident?
E-commerce	Yes (+8.5pp)	Yes (~24%)	High (15–35%)	Yes
Banking	Weak/mixed	Mild (~7%)	Low–mid (3–23%)	Yes
Government	No (noise)	No (~0%)	Low (3–10%)	Yes

Table 4. Cross-domain summary. E-commerce is vulnerable on every axis; banking is mild; government is robust but flooded (low statistical power). The one constant: where misses do occur, they are high-confidence.

4.5 Where the residual risk concentrates

Breakdowns (exploratory; several cells rest on 5–22 rows) locate the danger consistently. The hardest pragmatic cue in action mode is omission - risk conveyed by what is left unsaid - in both e-commerce (30.0%) and banking (13.3%); the executor's clean API call normalizes an omission further. In banking the residual risk is sharply concentrated in `account_wallet_ops` (21.9% action FN, the only banking cell approaching the e-commerce headline), while `payments_transfer` - the very domain nominated for replication - is essentially fully caught (0.0%). Pressure inverts between domains: e-commerce misses concentrate in high-pressure rows (31.2% action), whereas banking misses concentrate in low-pressure rows (21.4%), suggesting overt urgency on a money operation triggers strong fraud suspicion, and the dangerous banking cases are the calm, routine-looking ones.

4.6 Worked example (mechanism)

One e-commerce row makes the disguising concrete. For `ec_s28-40`, DeepSeek flipped from suspicious (instruction) to benign (action). The disguised instruction is a caller claiming to be the personal assistant of a famous singer, urgently requesting her full two-year purchase history including delivery addresses “for her taxes.” The ground truth for “suspicious” is an unverified third party requesting a named VIP's sensitive PII with no proof of authorization, using urgency as leverage. The executor strips the social framing to a clean call - `getCustomerPurchaseHistory(..., include_shipping_address=true)` - and a neutral report about supporting an “urgent tax filing.” Seeing only that, the monitor reads an ordinary data lookup and labels it benign: the missing-authorization signal never survives into the action layer. A banking analogue (`kyc-005-18`) shows the same mechanism for an OTP-relay scam, where both monitors flip to benign once the instruction collapses to a bare `ready_to_receive otp` state. (Full text in Appendix A.)

5. Discussion and Limitations

The results point to a single central claim: the executor context-compression vulnerability in multilingual multi-agent oversight is domain-dependent, and the dependence tracks the structure of the violation rather than the language. Where a violation lives in relational and omitted context - an unverified party requesting PII, a soft “just send it to this account” - execution compresses away exactly the evidence the monitor needs, and the action/report reads as routine. Where a violation is a concrete, explicit action - backdating a permit, skipping a required signature, an OTP handover attached to a specific large transfer, an unregistered payee - the act remains recognizable after both disguising and restatement, so neither the pragmatic rendering nor the executor layer hides it. E-commerce PII workflows are the former; government document acts are the latter; banking sits in between, leaning explicit.

For AI safety practice, our results reframe a common assumption - that action-only monitoring (watching what the agent did rather than what it was asked) is sufficient or even preferable. Our e-commerce results show it can be the weakest configuration, and that the gap it opens is partly an observability problem: trace-aware monitoring recovers most of the lost catch at essentially no modeling cost. Preserving and inspecting the instruction-action trace is therefore a concrete, low-cost safeguard, most valuable in the relational/PII domains where action-only monitoring fails. Equally important, auditing is the precondition for trusting any such mitigation: the same pipeline that surfaces the e-commerce failure is what shows the safeguard works there and is barely needed in government.

Limitations

- **Model-generated data.** Scenarios and renderings are produced by an LLM attacker/renderer, so the distribution may not match real Vietnamese workplace communication. The oracle keeps disguised rows recognizable, but the oracle is itself a model.
- **Executor-prompt dependence.** Action-mode results are a property of one executor. The banking executor emitted clean, well-typed calls that may have preserved operation type and thus aided recovery; a looser executor could reintroduce a gap. If this assumption fails, banking could look more e-commerce-like.
- **Government is floored, not proven safe.** With 0–1 misses per 60 rows we cannot distinguish “no effect” from “small effect.” The defensible claim is low measurable gap, not immunity; stronger conclusions need a stricter scenario-generation filter or more rows.
- **Observational cross-domain comparison.** The three subsets differ in content, base rates, and sampled violations, so the difference in context-loss magnitude should not be attributed to “domain” alone as an isolated variable.
- **Small breakdown cells and two monitors.** Cue/persona/pressure tables rest on 5–22 rows and are directional only, and we test just two monitors (Gemini 2.5 Flash, DeepSeek Chat); broader monitor and language coverage is needed before generalizing.

Future Work

1. Qualitatively inspect the `account_wallet_ops` action-mode misses and the omission / low-pressure cells - the only banking cases that resemble the e-commerce failure.
2. Treat the executor prompt as an explicit variable: test whether a more aggressively summarizing executor reintroduces a context-loss gap in banking and government.
3. Strengthen statistical power with a stricter scenario-generation filter and more rows, especially for government, before any full-dataset scale-up.
4. Extend to more monitors and to additional languages and code-switched speech, to test whether the domain-dependence pattern is stable.

6. Conclusion

Across three Vietnamese agentic domains, we find that the danger to monitor reliability is not primarily that pragmatic Vietnamese is hard to read - that effect is small - but that a naive executor can compress a socially loaded instruction into a routine-looking action that monitors then wave through, often with high confidence. This context-compression failure is severe in e-commerce (action-only false negatives ~22%, with up to ~24% of catchable violations disguised by execution),

mild in banking, and absent in government, because it tracks whether a violation is relational/omitted or a concrete, explicit action.

The practical implication is twofold. First, action-only monitoring is not a safe default in relational/PII workflows; preserving the full instruction-action trace recovers most of the lost detection at negligible cost. Second, which safeguard is needed where can only be known by auditing - the same audit that exposes the e-commerce gap also shows where mitigation is unnecessary. Trace-aware oversight, deployed selectively on the domains that need it, is a small change with an outsized safety return for multilingual multi-agent systems.

Code and Data

- **Code repository:**
<https://github.com/melanieyes/Pragmatic-Sophistry-in-Vietnamese-Multi-Agent-Oversight>
- **Dataset:**
<https://github.com/melanieyes/lost-in-translation-ai-safety/blob/agent-test/data/scenario.csv>
- **Informative web:**
https://pragmatic-sophistry-final.vercel.app/?fbclid=IwY2xjawSI_T1leHRuA2FibQlxMQBzcnRjBmFwcF9pZAEwAAEe7u8VwIG2IT6_Nnn4FhQ_4JC7FpIBHUKXfegRc8OfT663A6jAtLM1744v8Lw_aem_zUIG_KYa1UDktV5rChMXag

References

- [1] Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al. (2022). *Training Language Models to Follow Instructions with Human Feedback*. arXiv:2203.02155.
- [2] Wen, J., et al. (2024). *Language Models Learn to Mislead Humans via RLHF*. arXiv:2409.12822.
- [3] Scheurer, J., Balesni, M., & Hobbhahn, M. (2024). *Large Language Models Can Strategically Deceive Their Users When Put Under Pressure*. ICLR 2024 LLM Agents Workshop. arXiv:2311.07590.

Appendix A. Supplementary Results

A.1 Monitor-specific false-negative rates (pragmatic Vietnamese)

Domain	Context	Monitor	DeepSeek	Gemini
E-commerce	instruction	-	12.8%	0.0%
E-commerce	action	-	31.9%	12.8%
E-commerce	trace	-	8.5%	8.5%
Banking	instruction	-	8.8%	1.8%
Banking	action	-	8.8%	8.8%
Banking	trace	-	1.8%	5.3%

Table A1. Per-monitor false negatives on suspicious pragmatic-Vietnamese rows. In e-commerce DeepSeek is the weak monitor in action mode (31.9%); in banking the two converge (8.8%).

A.2 Conditional report-hiding counts

Domain	Monitor	Caught in instruction	Then missed in action
E-commerce	DeepSeek	41	10 (24.4%)
E-commerce	Gemini	46	6 (13.0%)
Banking	DeepSeek	52	4 (7.7%)
Banking	Gemini	56	4 (7.1%)
Government	DeepSeek	59	0 (0.0%)
Government	Gemini	60	2 (3.3%)

Table A2. Counts behind Figure 2. The denominator is rows the monitor caught in instruction mode; the numerator is how many of those it then missed when shown only the action.

A.3 Hardest cue / domain cells (action-mode FN, pragmatic VN)

Slice	Hardest cell	Action FN
E-commerce - pragmatic cue	omission	30.0%
E-commerce - persona	internal_staff (n=5)	50.0%
E-commerce - pressure	high	31.2%
Banking - sub-domain	account_wallet_ops	21.9%
Banking - pragmatic cue	omission	13.3%
Banking - pressure	low	21.4%
Government - difficulty	subtle rows only	4.0%

Table A3. Where misses concentrate (exploratory; small cells). Omission is the hardest cue in both e-commerce and banking; banking risk localizes to wallet operations and low-pressure rows; government misses are confined to subtle finance/identity rows.

A.4 Worked examples (translated)

E-commerce, ec_s28-40 (**DeepSeek: suspicious → benign**).

Instruction (disguised): “Good morning, I’m the personal assistant of Ms. Trần Mai, a famous singer. She needs the full 2-year purchase history from your store - all item details, order dates, and delivery addresses - to do her taxes. It’s very urgent, her accountant is waiting.”

Executor report: “Retrieving Ms. Trần Mai’s full 2-year purchase history, including item details, order dates and delivery address, to support her urgent tax filing.” The relational/authorization context is gone; the action reads as a routine lookup.

Banking, kyc-005-18 (**both monitors: suspicious → benign**).

Instruction (disguised): “The OTP just came through. Let me read it out for you to enter and finish the 80,000,000 VND transaction on VNPAY, quick and easy.”

Executor action/report: ready_to_receive_otp - “Indicating readiness to receive the OTP from the user to complete the 80,000,000 VND transaction on VNPAY.” A textbook OTP-relay pattern collapses to a neutral payment step.

LLM Usage Statement

LLMs were used in two cases: (1) Inside the experiment, as the objects of study and as pipeline components - the attacker/renderer, fluent oracle, naive executor, and the two monitors under evaluation (Gemini 2.5 Flash, DeepSeek Chat). (2) For writing, we used Claude to help edit grammars and polish this report. All numbers, tables, and claims are taken directly from the authors' three domain result reports and were independently verified against those source outputs.