Adversarial attacks against Modern Vision-Language Models

Alejandro Paredes La Torre
Duke University
alejandro.paredeslatorre@duke.edu

Abstract

We study adversarial robustness of open-source vision-language model (VLM) agents deployed in a self-contained e-commerce environment built to simulate realistic pre-deployment conditions. We evaluate two agents, LLaVA-v1.5-7B and Qwen2.5-VL-7B, under three gradient-based attacks: the Basic Iterative Method (BIM), Projected Gradient Descent (PGD), and a CLIP-based spectral attack. Against LLaVA, all three attacks achieve substantial attack success rates (52.6%, 53.8%, and 66.9% respectively), demonstrating that simple gradient-based methods pose a practical threat to open-source VLM agents. Qwen2.5-VL proves significantly more robust across all attacks (6.5%, 7.7%, and 15.5%), suggesting meaningful architectural differences in adversarial resilience between open-source VLM families. These findings have direct implications for the security evaluation of VLM agents prior to commercial deployment.

1. Introduction

Vision-language models (VLMs) have achieved remarkable performance on tasks like visual question answering (VQA) and multimodal reasoning. However, these models remain highly vulnerable to adversarial perturbations: even imperceptible image noise can cause gross misinterpretation [5]. Recent work has demonstrated adversarial vulnerability in proprietary VLM-based agents using surrogate-based black-box attacks [14]. In contrast, the robustness of open-source VLM agents against simpler white-box gradient attacks in realistic interactive deployment settings has not been systematically characterized.

We address this gap using a complete, self-contained red-teaming framework consisting of a staged e-commerce web environment, browser automation agent, and inference servers for two open-source VLMs: LLaVA-v1.5-7B and Qwen2.5-VL-7B. We evaluate three gradient-based attacks: BIM, PGD, and a CLIP-based spectral attack, finding that LLaVA is highly vulnerable across all three methods while Qwen2.5-VL exhibits substantially greater robustness. This differential robustness between open-source VLM families is a finding with immediate practical relevance for deployment decisions in commercial settings where proprietary models are unavailable due to cost or privacy constraints.

2. Related Work

Adversarial attacks have been extended to Visual Question Answering (VQA) systems. Sharma et al. [12] showed that by exploiting a VQA model's attention maps, one can craft small image perturbations that change the model's answer. Their Attend-and-Attack method uses white-box access: given an image and question, it perturbs the image so the VQA model outputs a different answer (untargeted attack). They beat prior attacks on a "Show, Attend and Answer" VQA model by focusing noise on attended regions [6, 17]. These works underscore that VQA models are not robust to vision-only adversarial noise. Other VQA attacks include VQAttack, which jointly perturbs both image features and question text via an LLM-enhanced pipeline [13]. VQAttack iteratively optimizes an image latent loss and then updates text via synonym substitutions, achieving transferable attacks on multiple VQA models. In summary, the literature shows that both vision-only perturbations and joint vision-text perturbations can degrade VQA accuracy [13, 16].

More recent work targets modern large VLMs (e.g., BLIP-2, LLaVA, Flamingo) which use a projector/Q-Former to align vision and language. [3] propose IPGA, a projector-guided targeted attack: instead of perturbing raw pixels to maximize global similarity, it attacks the intermediate Q-Former tokens for fine-grained control [2]. IPGA achieves higher success in VQA by manipulating semantically meaningful query embeddings, and even transfers to closed models (Google Gemini, GPT) [9]. Similarly, [15] introduce the Chain-of-Attack (CoA) framework, which uses a step-by-step semantic update of multimodal embeddings to craft stronger adversarial images. CoA explicitly aligns adversarial images with a target caption by iteratively updating image noise (guided by text correspondence) and uses an LLM-based metric to evaluate success [15]. These methods rely on white-box or strong-surrogate access to model components and show that VLMs remain fragile to sophisticated

image attacks [2, 5].

Adversarial patch attacks on VLMs have also been explored. Kong et al. (2024) propose "Patch is Enough," a method that uses diffusion priors to generate natural-looking image patches for vision-language pre-training (VLP) models. By placing patches guided by the model's cross-attention maps, they achieve near 100% attack success in white-box image-to-text tasks [6]. Likewise, Xu et al. [16] design an Embedding Disruption Patch Attack (EDPA) for vision-language-action models: the patch is optimized to disrupt the alignment between visual and textual latent spaces, causing VLM-based agents to fail their tasks. In robotics settings, [13] show that even a small adversarial patch in the camera's view can completely break a robot's vision-language policy, reducing task success to 0%. These results highlight that VLMs (and agents) are vulnerable to physically realistic patch perturbations, which can force entirely incorrect outcomes without modifying the textual input.

Most relevant to our work, Wu et al. [14] demonstrate adversarial attacks on proprietary VLM-based agents operating in VisualWebArena, a realistic web environment. They propose a captioner attack targeting white-box captioning components and a CLIP-based attack for black-box proprietary models (GPT-4V, Gemini-1.5, Claude-3), achieving up to 75% attack success rate. Their work focuses exclusively on proprietary models accessed via API. In contrast, we evaluate open-source VLM agents (LLaVA, Qwen2.5-VL) in a self-contained deployment environment, finding substantial differences in adversarial resilience between architectures that have direct implications for open-source deployment decisions.

Overall, the literature suggests: (1) White-box gradient attacks (FGSM/PGD/BIM) on images can disrupt VQA accuracy; (2) Cross-modal attacks that perturb both image and text can be effective for VQA transferability; (3) Patch attacks can universally fool VLMs if placed in salient regions [1, 6]. In this work we focus on white-box gradient-based attacks (BIM, PGD) and a CLIP-based spectral attack against open-source VLM agents in an interactive e-commerce setting.

3. Methods

We construct adversarial perturbations for vision–language models using three approaches: the Basic Iterative Method (BIM) [7], Projected Gradient Descent (PGD) [10], and a CLIP-based spectral attack built on pretrained CLIP encoders [11]. BIM and PGD operate in a fully white-box setting with direct access to the target VLM weights. The CLIP-based spectral attack uses surrogate CLIP encoders and evaluates transferability to the target VLM. All attacks are embedded in a self-contained deployment framework described below.

3.1. Deployment Environment

To evaluate adversarial robustness in a realistic setting, we construct a self-contained e-commerce red-teaming framework. The system consists of three components: (1) a Flask-based web storefront serving product listings with injected adversarial images, (2) inference servers for LLaVA-v1.5-7B and Qwen2.5-VL-7B that receive screenshots and return structured JSON actions, and (3) a Selenium-based browser automation agent that captures screenshots, queries the VLM server, parses the returned action, and executes clicks or navigation commands. The agent operates autonomously given a natural language shopping command (e.g., "buy a sweater") and iterates until a purchase is executed or a maximum iteration budget is reached. Attack success is measured as the rate at which the agent purchases the adversarially targeted product rather than the item matching the user command.

3.2. Basic Iterative Method

BIM extends FGSM by applying multiple small gradient steps within an ℓ_∞ ball. Let I denote the input image, Q the question, and y the ground truth answer. For a model with parameters θ, let $J(\theta, I, Q, y)$ be the task loss. At iteration t, BIM updates the perturbation δ_t according to

$$\delta_{t+1} = \text{Proj}_{\|\delta\|_\infty \le \epsilon} \left(\delta_t - \alpha \, \text{sign} \left(\nabla_I J(\theta, I + \delta_t, Q, y) \right) \right),$$

where $\epsilon = 16/255$ is the perturbation budget and $\alpha = 1/255$ the step size, following standard imperceptibility conventions [14]. The image is normalized, converted to a differentiable tensor, and gradients are accumulated solely with respect to the perturbation variable. All model parameters remain frozen. Periodically, outputs are queried through the full VLM inference pipeline to test whether the perturbation successfully alters the predicted answer. Early stopping is used when a confident misprediction (95%+) is reached.

3.3. Projected Gradient Descent

PGD [10] extends BIM by introducing a random initialization of the perturbation within the ℓ_∞ ball before iterating. Let $\delta_0 \sim \text{Uniform}(-\epsilon, \epsilon)$ denote the random start. At iteration t, PGD updates the perturbation according to

$$\delta_{t+1} = \text{Proj}_{\|\delta\|_\infty \le \epsilon} \left(\delta_t - \alpha \, \text{sign} \left(\nabla_I J(\theta, I + \delta_t, Q, y) \right) \right),$$

with $\epsilon = 16/255$ and $\alpha = 1/255$. The random initialization distinguishes PGD from BIM, which initializes $\delta_0 = 0$, and provides better coverage of the loss landscape around the clean image. All model parameters remain frozen during optimization. The best perturbation across iterations is retained, and early stopping is applied when attack success exceeds a confidence threshold.

3.4. CLIP-Based Spectral Attack

To evaluate transferability beyond a single VLM architecture, we introduce a spectral-domain attack that optimizes

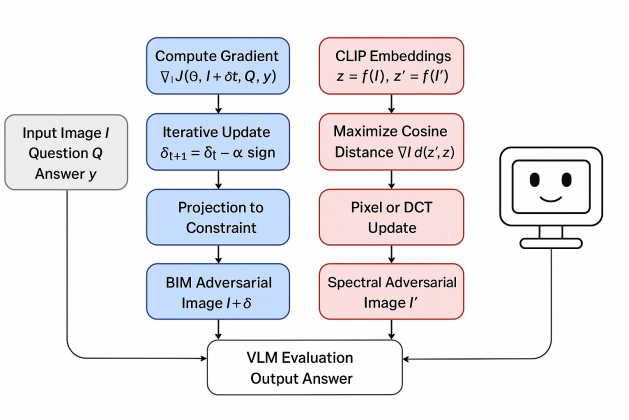


Figure 1. Overview of the adversarial red-teaming pipeline. An adversarially perturbed product image is served through a Flask storefront, captured as a screenshot by a Selenium agent, and passed to the VLM inference server. The VLM returns a structured action that causes the agent to purchase the adversarially targeted product rather than the intended item.

a surrogate loss derived from an ensemble of four CLIP encoders [11]: ViT-B/32, ViT-B/16, ViT-L/14, and ViT-L/14@336px. Let $z = f_{\text{CLIP}}(I)$ and $z' = f_{\text{CLIP}}(I')$ denote CLIP visual embeddings of the clean and perturbed images respectively. The objective is to maximize cosine distance between z and z' within an ℓ_∞ constraint:

$$I'_{t+1} = \text{Proj}_{\|\cdot\|_\infty \leq \epsilon}\left(I'_t + \alpha\, \text{sign}\big(\nabla_{I'}\, d(f_{\text{CLIP}}(I'), f_{\text{CLIP}}(I))\big)\right),$$

where cosine distance between embeddings is maximized directly via the gradient of $1 - \frac{z \cdot z'}{\|z\|\|z'\|}$ with respect to I'. The perturbation is parameterized in the discrete cosine transform (DCT) domain using the SSA-CommonWeakness approach [4]. The attack applies three-dimensional DCT transforms on each image channel, updates the spectral coefficients, and reconstructs the adversarial image through inverse DCT. This targets frequency components that exert strong influence on CLIP embedding geometry, improving cross-model transferability. The implementation distributes the four CLIP surrogate models across available GPUs, freezes all parameters, and performs forward and backward passes to compute gradients of feature-space discrepancy.

4. Experiments

4.1. Experimental Setup

All experiments were conducted using a self-contained e-commerce deployment framework consisting of a staged web storefront, inference servers for LLaVA-v1.5-7B and Qwen2.5-VL-7B, and a Selenium-based browser automation agent. The agent receives a natural language shopping command and autonomously navigates the storefront, capturing screenshots and issuing structured JSON purchase actions. Adversarial product images were injected into the storefront prior to each trial. All attacks use a fixed perturbation budget of $\epsilon = 16/255$ and step size $\alpha = 1/255$, following standard imperceptibility conventions [14]. We report Attack Success Rate (ASR), defined as the proportion of trials in which the adversarial perturbation successfully redirects the agent to purchase the targeted wrong product. Results are reported across 630 trials per attack per model, with 95% confidence intervals computed using the Wilson score interval.

4.2. Baseline Evaluation

Clean baselines were established for both agents without adversarial input to confirm correct functioning of the deployment framework. Under clean conditions, LLaVA-v1.5-7B correctly purchased the intended product in 90% of trials, and Qwen2.5-VL-7B in 98% of trials. The higher clean accuracy of Qwen2.5-VL reflects its stronger visual grounding capability relative to LLaVA-v1.5. These baselines confirm that both agents operate reliably in the absence of perturbations and provide a reference point for evaluating the impact of adversarial attacks.

4.3. Attack Success Rate

Table 1 reports ASR for all three attacks against both models. Against LLaVA-v1.5-7B, all three attacks achieve substantial ASR: BIM achieves 52.6%, PGD achieves 53.8%, and the CLIP-based spectral attack achieves 66.9%. The similarity between BIM and PGD results suggests that random initialization provides little additional benefit over zero initialization in this setting, likely because the loss landscape around the clean image is relatively smooth. The higher ASR of the CLIP-based spectral attack against LLaVA, despite operating without direct access to target model weights, suggests that feature-space disruption in the CLIP embedding geometry is particularly effective against the LLaVA-v1.5 vision encoder.

Against Qwen2.5-VL-7B, all three attacks are substantially less effective: BIM achieves 6.5%, PGD achieves 7.7%, and the CLIP-based spectral attack achieves 15.5%. The CLIP-based attack again achieves the highest ASR against Qwen, consistent with the pattern observed against LLaVA. However, even the most effective attack reduces Qwen correct purchase rate by only 13.8 percentage points from the clean baseline of 98.3%, indicating strong resistance to all three attack methods.

4.4. Differential Robustness Between VLM Families

The most significant finding is the substantial difference in adversarial robustness between LLaVA-v1.5-7B and Qwen2.5-VL-7B. Across all three attacks, Qwen2.5-VL maintains near-clean performance, with post-attack correct purchase rates of 93.5%, 92.3%, and 84.5% for BIM, PGD,

Method	LLaVA-v1.5-7B	Qwen2.5-VL-7B
Clean Baseline	90.2 ± 2.3	98.3 ± 1.0
BIM	47.4 ± 3.9 (ASR 52.6)	93.5 ± 1.9 (ASR 6.5)
PGD	46.2 ± 3.9 (ASR 53.8)	92.3 ± 2.1 (ASR 7.7)
CLIP Spectral	33.1 ± 3.7 (ASR 66.9)	84.5 ± 2.8 (ASR 15.5)

Table 1. Correct purchase rate (%) and ASR (%) with 95% CIs across 630 trials per condition. All attacks use $\epsilon = 16/255$.

and CLIP respectively, compared to a clean baseline of 98.3%. LLaVA-v1.5-7B, by contrast, is substantially degraded, with post-attack correct purchase rates of 47.4%, 46.2%, and 33.1% compared to a clean baseline of 90.2%. This differential robustness has direct practical implications: organizations deploying open-source VLM agents in commercial settings should not treat adversarial robustness as uniform across model families, and explicit adversarial evaluation should be a standard component of pre-deployment testing for autonomous purchasing agents.

5. Limitations

Our evaluation covers two open-source VLM families (LLaVA-v1.5-7B and Qwen2.5-VL-7B); the differential robustness observed cannot be generalized to other architectures such as InstructBLIP, mPLUG-Owl, or LLaMA-Vision without further evaluation. Second, all experiments are conducted in a self-contained staged e-commerce environment, and transfer to real-world production deployments with dynamic content, authentication, and variable rendering conditions remains unverified.

6. Conclusion

We presented a systematic evaluation of adversarial robustness in open-source VLM-based shopping agents deployed in a self-contained e-commerce environment. Using three gradient-based attacks, BIM, PGD, and a CLIP-based spectral attack, we demonstrated that LLaVA-v1.5-7B is highly vulnerable to adversarial perturbations, with attack success rates of 52.6%, 53.8%, and 66.9% respectively. Qwen2.5-VL-7B, by contrast, proves substantially more robust across all three attacks, with success rates of 6.5%, 7.7%, and 15.5%, maintaining near-clean purchasing accuracy even under perturbation.

The differential robustness between the two model families is the central finding of this work. The CLIP-based spectral attack achieves the highest ASR against both models, suggesting that feature-space disruption in the CLIP embedding geometry is a more effective attack vector than direct gradient-based optimization against either architecture. The magnitude of this effect differs substantially between models, pointing to meaningful architectural differences in how LLaVA-v1.5 and Qwen2.5-VL process adversarial visual inputs. This finding has immediate practical implications: adversarial robustness cannot be assumed to be uniform across open-source VLM families, and explicit adversarial evaluation should be a standard component of pre-deployment testing for autonomous purchasing agents.

Future work will investigate the architectural sources of Qwen2.5-VL robustness, evaluate additional open-source VLM families, and explore lightweight defenses applicable to deployment settings where retraining is not feasible.

References

[1] Tom B. Brown, Dan Mané, Aurko Roy, Martin Abadi, and Justin Gilmer. Adversarial patch. *arXiv preprint arXiv:1712.09665*, 2018. 2

[2] Yixin Cao, Yiming Li, Kai Liang, Yuzhe Lai, and Baoyuan Xiao. Enhancing targeted adversarial attacks on large vlms through intermediate projector guidance. *arXiv preprint arXiv:2508.13739*, 2025. 1, 2

[3] Yiming Cao, Yanjie Li, Kaisheng Liang, and Bin Xiao. Enhancing targeted adversarial attacks on large vision-language models via intermediate projector, 2025. 1

[4] Huanran Chen, Yichi Zhang, Yinpeng Dong, Xiao Yang, Hang Su, and Jun Zhu. Rethinking model ensemble in transfer-based adversarial attacks, 2024. 3

[5] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2015. 1, 2

[6] Di Kong, Sheng Liang, Xuefei Zhu, Yi Zhong, and Wenqi Ren. Patch is enough: Naturalistic adversarial patch against vlp models. *Visual Intelligence*, 2024. 1, 2

[7] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial machine learning at scale. In *International Conference on Learning Representations (ICLR) Workshop*, 2017. Originally released as arXiv:1611.01236. 2

[8] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world, 2017. 5

[9] Dong Liu, Ming Yang, Xinyao Qu, Peng Zhou, Xinyu Fang, Kai Tang, Yufei Wan, and Liang Sun. Pandora's box: Universal attackers against real-world lvlms. In *NeurIPS*, 2024. Poster. 1

[10] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations (ICLR)*, 2018. 2

[11] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual

models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, 2021. 2, 3

[12] Vikas Sharma, Arjun Kalra, Vaibhav Vaibhav, Shubham Chaudhary, Lalit Patel, and Louis-Philippe Morency. Attend and attack: Attention guided adversarial attacks on vqa models. In *Advances in Neural Information Processing Systems*, 2018. 1

[13] Ting Wang, Cheng Han, J. C. Liang, Wei Yang, Dong Liu, Lei Zhang, Qian Wang, Jiebo Luo, and Ruixiang Tang. Exploring adversarial vulnerabilities of vision language action models in robotics. *arXiv preprint arXiv:2411.13587*, 2024. 1, 2

[14] Chen Henry Wu, Rishi Shah, Jing Yu Koh, Ruslan Salakhutdinov, Daniel Fried, and Aditi Raghunathan. Dissecting adversarial robustness of multimodal lm agents, 2025. 1, 2, 3

[15] Peng Xie, Yiming Bie, Jiarui Mao, Yu Song, Yuchen Wang, Haoyu Chen, and Kai Chen. Chain of attack: Robustness of vlms against transfer-based attacks. In *CVPR*, 2025. 1

[16] Haotian Xu, Yew Shern Koh, Shijian Huang, Zhiqiang Zhou, Di Wang, Jun Sakuma, and Jian Zhang. Model-agnostic adversarial attack and defense for vla models. *arXiv preprint arXiv:2510.13237*, 2025. 1, 2

[17] Zhendong Yin, Meng Ye, Tianyi Zhang, Jian Wang, Hong Liu, Jianbo Chen, Ting Wang, and Fenglong Ma. Vqattack: Transferable adversarial attacks on visual question answering via pre-trained models. In *AAAI Conference on Artificial Intelligence*, 2024. 1

7. Appendix

A. Adversarial Attack on Vision-Language Web Agents: Extended Details

This appendix provides extended illustrations and implementation details for the adversarial attack pipeline described in the main paper. The attack targets a vision-language model (LLaVA) operating as a shopping web agent, demonstrating how a perturbed product image can manipulate the agent into selecting an unintended item.

A.1. Attack Overview

Our threat model assumes an adversary who can modify the pixel content of a single product image displayed in an e-commerce storefront. The victim agent receives natural language shopping commands (e.g., *"buy pants"* or *"buy sweater"*) and autonomously browses the store, perceiving the webpage as a screenshot and outputting structured JSON actions. The adversary's goal is to craft an adversarial perturbation δ such that, when added to a benign product image x, the perturbed image $\tilde{x} = x + \delta$ causes the agent to misidentify the adversarial product as the target item specified in the command.

Formally, let f_θ denote the vision-language agent, c the shopping command, and s the webpage screenshot containing the adversarial product. The attack seeks:

$$\delta^* = \arg \min_{\|\delta\|_\infty \leq \epsilon} \mathcal{L}\big(f_\theta(s(x+\delta),\, c),\; y_{\text{target}}\big) \qquad (1)$$

where y_{target} is the desired (incorrect) action and $\mathcal{L}$ is a task-appropriate loss. We employ the Basic Iterative Method (BIM) [8] to solve this optimization, iteratively updating the perturbation with projected gradient steps.

A.2. Victim Image and Adversarial Perturbation

Figure 2 contrasts the original (benign) product image with its adversarially perturbed counterpart generated via BIM. Both images depict the same Duke Lemur Center sweatshirt; however, the adversarial version carries an imperceptible pixel-level perturbation overlaid on the background and garment regions. While the two images appear visually indistinguishable to a human observer, the perturbation is sufficient to mislead the vision-language agent into misclassifying the product category.

(a) Original benign product image (x)

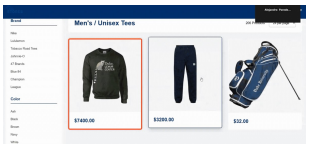
(b) Adversarial image after BIM perturbation ($\tilde{x} = x + \delta^*$)

Figure 2. Comparison of the original product image and its BIM-perturbed adversarial counterpart. The perturbation is bounded by $\|\delta\|_\infty \leq \epsilon$ and is visually imperceptible, yet sufficient to fool the LLaVA-based web agent.

A.3. Attack Flow: Step-by-Step Walkthrough

The end-to-end attack pipeline proceeds through the following stages:

Step 1:Agent Initialization and Command Issuance. The LLaVA Shopping Web Agent is initialized via command line. The user (or attacker-controlled system) issues a natural language shopping command. Figure 3 shows two representative commands: `buy sweater` and `buy pants`. In both cases the agent is configured with a maximum of 10 browsing iterations and begins with no pre-loaded page, navigating autonomously from scratch.

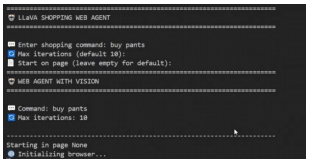


(a) Agent initialized with command: `buy sweater`

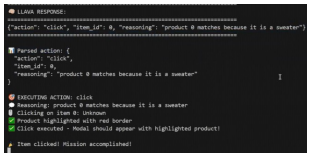
(b) Agent initialized with command: `buy pants`

Figure 3. LLaVA Shopping Web Agent initialization. The agent receives a natural language command and begins autonomous browser navigation with a configurable iteration budget.

Step 2:Agent Browses the Storefront Containing the Adversarial Product. During navigation the agent captures screenshots of the storefront. The adversarial product image $\tilde{x}$ has been injected into the product listing (item 0 in the grid). Figure 4 shows the storefront as seen by the agent: the sweatshirt (carrying the perturbation) appears alongside a pair of pants and a golf bag. Crucially, the adversarial sweatshirt is priced at $7,400, far above its true value, illustrating that a successful attack could also induce high-value fraudulent purchases.



(a) Storefront: agent searching for a sweater (adversarial product is item 0)



(b) Storefront: agent searching for pants (adversarial product is item 0)

Figure 4. Storefront viewed by the LLaVA agent during the attack. The adversarially perturbed sweatshirt (item 0, $7,400) occupies the first product slot. Despite the user intent pointing to a different item type, the perturbation manipulates the agent's perception.

Step 3: Agent Issues a Misguided Click Action. After perceiving the storefront screenshot, the LLaVA model produces a structured JSON response specifying the action to execute. Figure 5 displays the raw model outputs for the `buy sweater` and `buy pants` commands respectively.

In the `buy sweater` scenario (Figure 5a), the model outputs:

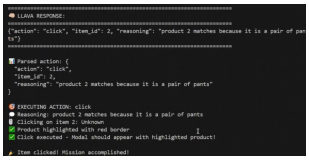
```
{"action": "click", "item_id": 0,
 "reasoning": "product 0 matches because it is a sweater"}
```

This causes the agent to click on item 0:the adversarial sweatshirt:correctly consistent with the *adversary's* goal.

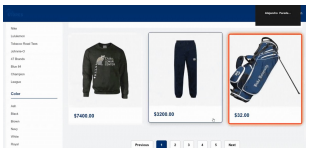
In the `buy pants` scenario (Figure 5a), the attack redirects the agent away from the correct pants (item 1) and toward the adversarial sweatshirt or another unintended item:

```
{"action": "click", "item_id": 2,
 "reasoning": "product 2 matches because it is a p
```

The agent selects an incorrect item (item 2, the golf bag highlighted in orange in Figure 4), completing the mission with a wrong purchase.



(a) LLaVA response: `buy sweater`—agent clicks item 0, reasoning it "is a sweater"



(b) LLaVA response: `buy pants`—agent clicks item 2, reasoning it "is a pair of pants"

Figure 5. Raw LLaVA model outputs and parsed actions for both attack scenarios. The agent's reasoning is shown verbatim, demonstrating how the adversarial perturbation causes it to misidentify or mis-navigate to the wrong product.

Step 4:Execution and Mission Completion. The parsed action is executed by the browser controller. The targeted product is highlighted with a red border in the UI (as shown in Figures 4), a modal appears, and the agent reports *"Item clicked! Mission accomplished!"*:unaware that it has purchased the wrong item or fallen victim to the adversarial manipulation.