Hydra: Towards Massively Multi-Target Backdoor Attacks against VLMs

Siyuan Liang[1,†], **Zhantao Yang**[2,†], **Yiming Li**[1,*], **Tong Zhang**[3], **Shangwen Zhu**[3], **Xiujin Liu**[4], **Dacheng Tao**[1]

[1]Nanyang Technological University
[2]Shanghai Jiao Tong University
[3]Zhejiang University
[4]University of Michigan

Abstract

Vision-Language Models (VLMs) are increasingly deployed as the core of perception and reasoning in safety-critical *VLM-integrated systems* such as smart assistants, autonomous driving, and multi-modal agents. Because training a competitive VLM is expensive, developers commonly reuse a few third-party open-source checkpoints and adapters, exposing a system-level supply-chain surface that is highly susceptible to backdoor injection. Existing VLM backdoor attacks implicitly assume that distinct trigger-target mappings can be learned independently, thereby scaling gracefully with the number of targets. We show that this assumption breaks down systematically: as the target set grows, cross-target interference accumulates in the shared multimodal representation space, and representative attacks collapse below 10% once roughly ten backdoors are embedded. We trace this to a capacity bottleneck of static triggers under a fixed perturbation budget, and propose HYDRA, a content-aware conditional trigger generator that jointly conditions on the input image and target semantics, and optimizes trigger generation through dual-branch generation, first-token alignment, and stealth-constrained optimization. On four mainstream VLMs, HYDRA sustains over 94% per-target attack success with PSNR above 47 dB while scaling to 100 targets, whereas baselines degrade to near-ineffectiveness. On VLM-based mobile-agent attack chains, it achieves $\sim 70\%$ end-to-end success and resists a representative suite of defenses.

1 Introduction

Vision-language models (VLMs) have become the perception and reasoning core of high-stakes VLM-integrated systems, deployed in visual question answering, cross-modal retrieval, and mobile or web agents that act on a user's behalf [Li *et al.*, 2025]. Embedding a VLM into an agentic pipeline is a consequential shift: outputs no longer stop at the screen but are routed through planners and tool calls to resource access, information presentation, and even physical execution. A silently manipulated VLM is then not a mere misclassifier but an unfaithful executor in the user's workflow, so any pre-deployment compromise can be amplified along the planning and tool-use chain into system-level harm [OWASP Foundation, 2025].

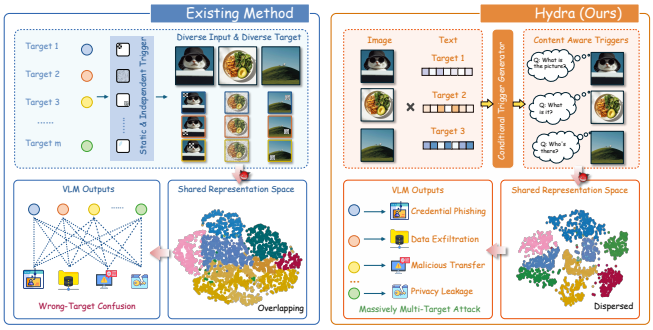

Figure 1: **Motivation of HYDRA.** Independently constructed trigger-target bindings interfere in the shared VLM representation space as the target set grows, causing wrong-target confusion. HYDRA produces content-aware conditional triggers with separable representations, enabling stable massive multi-target attacks.

Because training and aligning a competitive VLM remains beyond most downstream developers, the ecosystem has converged on a few public checkpoints such as LLaVA [Liu *et al.*, 2024], Qwen-VL [Bai *et al.*, 2025], and InternVL [Zhu *et al.*, 2025], adopted directly as the reasoning core of downstream systems. This centralized supply chain constitutes a system-level attack surface: a single compromised third-party component can propagate through model reuse to many downstream systems. The most prominent threat here is the backdoor attack, where an attacker controlling a published component implants hidden trigger-target associations that keep the system normal on clean inputs but activate malicious responses under attacker-specified triggers.

Existing VLM backdoor studies typically consider a single or a few targets [Lyu *et al.*, 2024, Xu *et al.*, 2024, Liang *et al.*, 2025a]. Yet real deployments are multi-task, multi-context, and multi-step, so a realistic threat should support diverse malicious behaviors over a large target space rather than hijack a few fixed outputs. Methods relevant to such deployments

[†]These authors contributed equally.
[*]Corresponding author: Yiming Li.

implicitly assume that distinct trigger-target mappings can be learned independently and coexist stably as the target set expands. Our revisiting experiments show this quickly fails: the per-target success rate of a representative attack falls below 10% once around ten targets are implanted, the wrong-target rate rises monotonically with the number of targets, and trigger-bearing samples from different targets become nearly overlapped in representation space. This raises a critical question: *do VLM backdoor attacks remain viable threats when they must control many distinct target behaviors at once?*

We trace this failure to a fundamental limitation of existing designs: under a constrained perturbation budget, triggers that are hand-crafted or individually optimized cannot encode enough information to distinguish targets as the target space grows. A Fano-style analysis shows that reliably distinguishing M targets requires $\log M \lesssim I(T; Y \mid X)$, so the effective mutual information of a static trigger under a fixed L_∞ budget cannot be increased arbitrarily by more aggressive training or more complex patterns. Large-scale multi-target backdoors should therefore be treated not as an independent superposition of single-target triggers, but as a joint trigger-modeling problem, where triggers encode target information by leveraging both input content and target semantics.

Based on this insight, we propose HYDRA, a *content-aware conditional trigger generation* framework. Instead of learning an independent fixed trigger per target, HYDRA unifies target-specific perturbations into a single conditional generator that takes the input image and the target semantic embedding as joint inputs and synthesizes perturbations under the same L_∞ budget as existing methods, so the trigger varies with both the sample and the target. Concretely, HYDRA uses a dual-branch generator with hard and soft stealth constraints; the generator is co-optimized with a malicious adapter during attacker-side training, while only the malicious adapter ϕ_b is released at deployment. The victim-side deployment surface thus remains identical to existing backdoor attacks.

We evaluate HYDRA on VQA-v2 [Goyal *et al.*, 2017] and OK-VQA [Marino *et al.*, 2019] across four representative VLMs, scaling the target count to $M{=}100$, an order of magnitude beyond prior few-target settings. HYDRA maintains an average per-target ASR above 94% and PSNR above 47 dB, whereas representative baselines either lose activation or suffer severe target misbinding as M increases. We further validate HYDRA in a realistic VLM-based mobile agent [Zhang *et al.*, 2025], where a compromised VLM amplifies single-step manipulation into end-to-end behavioural hijacking, achieving $\sim$70% average success across 10 cross-application chains spanning credential phishing, data exfiltration, and disinformation propagation. HYDRA further resists a comprehensive suite of potential defenses. Our main contributions are as follows:

- We reveal that existing VLM backdoor attacks rely on a latent, untested assumption–that distinct trigger–target mappings can be learned independently and stably coexist–and are the first to expose its breakdown under target scaling.
- We propose HYDRA, a content-aware conditional trigger generation framework that replaces independent per-target perturbations with a unified generator jointly conditioned on input content and target semantics.
- Extensive experiments on four mainstream VLMs show that HYDRA sustains over 94% per-target ASR at $M{=}100$ while baselines fall below 10% after $M{=}10$, and remains effective against representative defenses and cross-application attack chains.

2 Related Work

Modern VLMs are rarely deployed in isolation; they act as the perception and reasoning core of multi-stage pipelines whose outputs are routed through prompt templates, function calls, and external tools to trigger information presentation, resource access, and physical actions. Because training a competitive VLM is prohibitive, developers reuse a few public checkpoints [Liu *et al.*, 2024, Bai *et al.*, 2025, Zhu *et al.*, 2025] with lightweight LoRA [Hu *et al.*, 2022] adapters distributed publicly. From an AIGC-safety standpoint, this makes the generative model itself the "first line of defense": once an upstream checkpoint or a third-party component is covertly compromised, the malicious behavior propagates to many downstream products, and in agentic deployments such as AppAgent [Zhang *et al.*, 2025] a hijacked model becomes an untrusted executor rather than a mere misclassifier.

Backdoor attacks are the most prominent threat on this surface. Matching our threat model, most VLM backdoors inject into fine-tuned components: fixed-patch triggers (BadNets [Gu *et al.*, 2019], DualKey [Walmer *et al.*, 2022]), blending triggers (Blended [Chen *et al.*, 2017]), and multimodal or generative triggers (TrojVLM [Lyu *et al.*, 2024], VL-Trojan [Liang *et al.*, 2025a], MABA [Liang *et al.*, 2025b], Shadowcast [Xu *et al.*, 2024]). Only a few methods (M-to-N [Hou *et al.*, 2024], MTAttack [Wang *et al.*, 2026]) treat the target count as an evaluation dimension, and their scales remain small, leaving the stability boundary of large target spaces uncharacterized. Despite differing trigger morphology, all rely on the same independent-superposition assumption we revisit in Section 4. On the defense side, mitigations act as the "last line of defense", spanning input-side detection, adapter-side cleansing, and inversion-based repair [Sun *et al.*, 2025]. We adopt representative methods plus two adaptive defenses in Section 6.5.

3 Problem Formulation and Threat Model

A VLM-integrated system is $F_{\theta_0,\phi}\colon (\mathbf{x}, \mathbf{q}) \mapsto \mathbf{y}$, a public base checkpoint θ_0 with an attached adapter ϕ, mapping an image $\mathbf{x}$ and query $\mathbf{q}$ to a natural-language response $\mathbf{y}$. We write F_{θ_0,ϕ_c} for the clean reference and F_{θ_0,ϕ_b} for the compromised system with malicious adapter ϕ_b. The attacker predefines targets $\mathcal{Y}_t = \{\mathbf{y}_1, \ldots, \mathbf{y}_M\}$, with M scaled to 50 or 100. For each target a visual trigger transforms $\mathbf{x} \mapsto \mathcal{T}_m^x(\mathbf{x})$; the text side is identity. With a matching function $s(\cdot,\cdot) \in [0,1]$ and threshold τ, per-target success and cross-target confusion are

$$\mathrm{ASR}_m = \mathbb{E}\big[\mathbb{I}\big(s(F_{\theta_0,\phi_b}(\mathcal{T}_m^x(\mathbf{x}), \mathbf{q}), \mathbf{y}_m) \geq \tau\big)\big], \quad (1)$$

$$\mathrm{Conf}_i = \mathbb{E}\big[\mathbb{I}\big(\exists j \neq i,\ s(F_{\theta_0,\phi_b}(\mathcal{T}_i^x(\mathbf{x}), \mathbf{q}), \mathbf{y}_j) \geq \tau\big)\big]. \quad (2)$$

We call the average of Conf_i the *Wrong-Target Rate* (WTR). A successful large-scale multi-target backdoor requires high

mean ASR, low WTR, and bounded utility drop $U(F_{\theta_0,\phi_c}) - U(F_{\theta_0,\phi_b})$ simultaneously.

Threat Model. We consider an adapter-level supply-chain attack. The attacker acts as a third-party adapter publisher, training and distributing a malicious adapter on top of a public base VLM; a downstream integrator downloads it, attaches it, and deploys, with routine clean-utility validation but limited resources for specialized backdoor auditing. The attacker has full access to θ_0 and freely chooses the training data, poisoning strategy, objective, and hyperparameters, and may evaluate ϕ_b before release, but has no control once deployed. All visual triggers obey an L_∞ budget,

$$\mathcal{T}_m^x(\mathbf{x}) = \mathbf{x} + \boldsymbol{\delta}_m(\mathbf{x}), \qquad \|\boldsymbol{\delta}_m(\mathbf{x})\|_\infty \le \epsilon, \tag{3}$$

and text-side triggers, if used, must appear as natural phrases. The defender may apply input filtering/transformation, clean fine-tuning, inversion-based repair, and adaptive defenses informed of the attack's structure, but not full-parameter retraining, certified defenses, or exhaustive manual auditing.

4 Motivation: Why Existing Attacks Collapse

In this section, we explore whether a compromised VLM can maintain stable trigger-target mappings as M grows from the traditional single-/few-target regime to a large target space, and if not, why. We answer this question at three levels: empirical failure, representation-level interference, and an information-theoretic bottleneck.

Empirical Failure under Target Scale-up. We set $M \in \{5, 10, 20\}$ and evaluate nine representative baselines spanning fixed, adaptive-multimodal, generative, and multi-target paradigms on VQA-v2 and OK-VQA, all under matched training and perturbation budgets. Our full comparison across all nine baselines (Appendix B) shows that existing methods do not degrade smoothly but collapse, consistently across paradigms and datasets. The failure is also non-uniform: a few targets survive while most fail completely, the signature of shared-space interference rather than synchronous decay.

Interference at the Representation Level. Using MABA (the strongest baseline) as a probe, we extract triggered-sample representations $\mathbf{h}_{i,m} = H_{\theta_0,\phi_b}(\mathcal{T}_m^x(\mathbf{x}_i), \mathbf{q}_i)$ and measure their separability via t-SNE structure and average cross-target cosine similarity at both the sample and target levels (Figure 2). As M grows from 5 to 20, the average cross-target cosine similarity rises from 0.756 to 0.848 and intra-target similarity from 0.894 to 0.983: representations become more compact yet more mutually similar, matching the increased WTR and failed-target ratio. This pattern holds across all nine baselines (Appendix C), confirming the independent-superposition assumption fails generally.

Information-theoretic Bottleneck. A trigger is an information channel from the target index to the model output, bounded by the L_∞ budget. Let X be the natural input, $B \sim \mathrm{Unif}(\{1, \ldots, M\})$ the target index, T the induced trigger, and Y the output.

Proposition 4.1 (Necessary condition)**.** *If the average error rate over M targets is at most $\eta \le (M-1)/M$, then*

$$\log M \le I(T; Y \mid X) + h(\eta) + \eta \log(M-1), \tag{4}$$

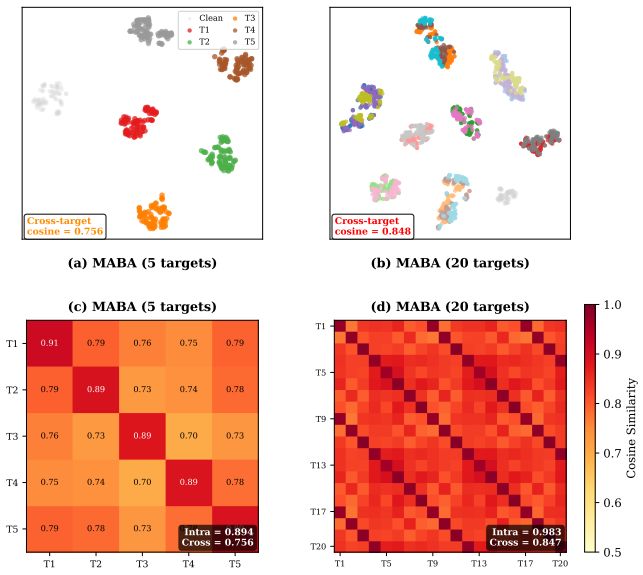

Figure 2: **Representation interference in MABA.** Top: t-SNE of triggered-sample representations from different targets with average cross-target cosine similarity. Bottom: target–target cosine-similarity heatmaps. As M grows from 5 to 20, cross-target similarity rises and clusters entangle.

with $h(\cdot)$ the binary entropy function.

Ignoring the fixed error term, $\log M \lesssim I(T; Y \mid X)$: distinguishing more targets demands that triggers transmit more target-related information to the model output. Under a fixed perturbation budget, however, the effective $I(T; Y \mid X)$ of a static or independently optimized trigger is bounded and cannot be increased arbitrarily by enlarging the target set. Once this information becomes insufficient, triggers for different targets can no longer form mutually non-interfering subregions in the shared space, yielding target confusion and target-level failure. This is consistent with the previous two observations: the failed-target ratio rises monotonically with M, while triggered representations become nearly inseparable at large M. (Proof via conditional Fano's inequality and the data-processing inequality is given in Appendix A.)

This gives the starting point for our method: a large-scale multi-target backdoor cannot be treated as a simple superposition of single-target attacks. A scalable attack must raise the trigger's effective capacity to encode target-related information under the same perturbation budget, while reducing cross-target interference in the shared representation space.

5 The Proposed Method: Hydra

5.1 Overview

The core of HYDRA is a *content-aware conditional trigger generation* paradigm, in contrast to existing methods that treat trigger-target mappings as independently learnable and directly stackable. Section 4 showed that this independent-superposition design fails as M grows, because all bindings share the parameter and representation space of the same compromised system F_{θ_0,ϕ_b} and compete for limited capac-

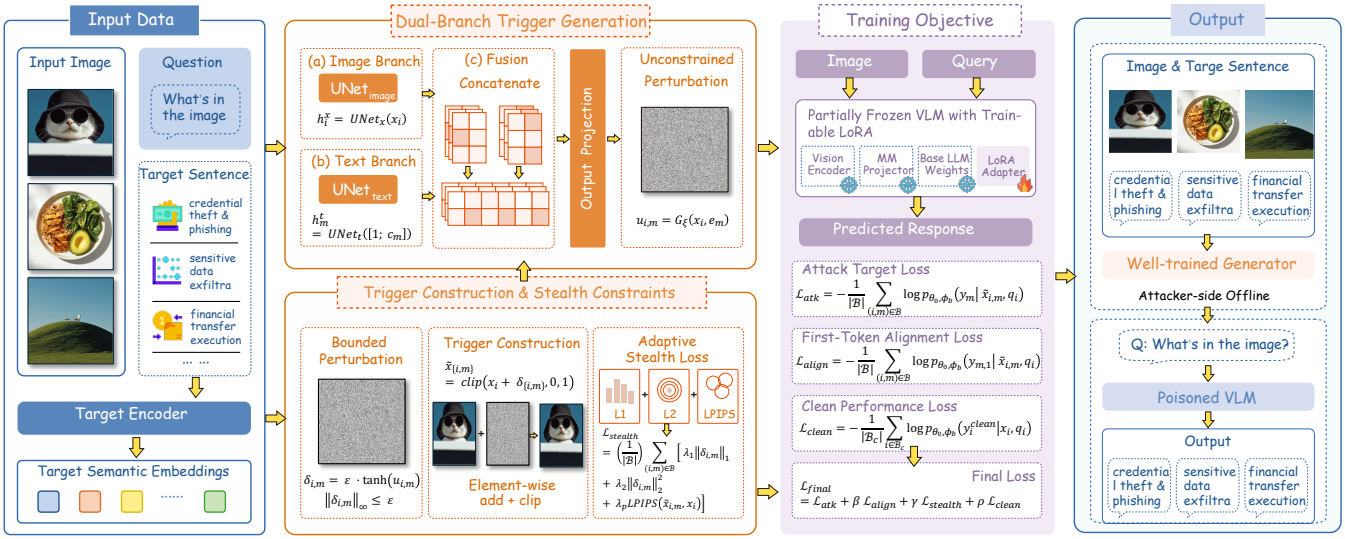

Figure 3: **Overview of HYDRA.** Given an input image, a query, and a target response, HYDRA encodes the target response as a semantic condition and uses an offline dual-branch conditional generator to synthesize a stealthy, content-aware visual perturbation; the query is unchanged. Triggered images are mixed with clean samples to train the malicious adapter ϕ_b that is released to downstream users, while the generator G_ξ is retained by the attacker for offline trigger synthesis. The victim-side deployment surface remains identical to existing visual backdoor attacks.

ity. Instead of learning a static perturbation per target, HYDRA instantiates the visual trigger transformation $\mathcal{T}_m^x$ with an attacker-side offline conditional generator G_ξ.

A trigger transformation that works under large M must satisfy three goals simultaneously. It should be *content-aware*, adapting to local textures, salient regions, and object boundaries rather than degrading into an input-independent pattern; *stealth-constrained*, satisfying the L_∞ budget ϵ while controlling perturbation energy and perceptual discrepancy; and *target-discriminative*, carrying enough information for different targets to form stable bindings, thereby raising ASR_m and lowering Conf_i. Figure 3 shows the three corresponding components. Given $(\mathbf{x}_i, \mathbf{q}_i, \mathbf{y}_m)$, HYDRA encodes $\mathbf{y}_m$ into a target-conditioned embedding $\mathbf{e}_m$, synthesizes an unconstrained perturbation $\mathbf{u}_{i,m}$, applies bounded parameterization to obtain $\boldsymbol{\delta}_{i,m}$ within the L_∞ budget, and adds it to the image to form the triggered image $\widetilde{\mathbf{x}}_{i,m} = \mathcal{T}_m^x(\mathbf{x}_i)$. Triggered samples are mixed with clean samples to co-train ϕ_b and G_ξ; only ϕ_b is released, and the attacker introduces $\mathcal{T}_m^x(\mathbf{x})$ through controllable visual content such as web images, screenshots, or interfaces.

5.2 Dual-Branch Trigger Generation

Target Response Embedding. We encode each target response $\mathbf{y}_m$ using the frozen language token embedding layer of the VLM: after tokenizing, we average the embeddings of all non-padding tokens and project them to the generator's conditioning dimension,

$$\mathbf{e}_m = P_e\Big(\frac{1}{|\mathcal{I}_m|}\sum_{t\in\mathcal{I}_m}\mathrm{Emb}(y_{m,t})\Big), \tag{5}$$

where $\mathcal{I}_m$ indexes non-padding tokens and P_e is a trainable projection in G_ξ. This exploits the semantic structure of target responses under shared parameters instead of learning semantic-free one-hot patterns per target.

Dual-branch Generator. The generator adopts a dual-branch UNet. An image branch UNet_x preserves spatial structure so the perturbation can adapt to local content; a target branch UNet_t unfolds the broadcast target condition into spatial features. The two feature maps are concatenated and fused into a three-channel perturbation:

$$\begin{aligned}\mathbf{h}_i^x &= \mathrm{UNet}_x(\mathbf{x}_i), \quad \mathbf{c}_m = \mathrm{Broadcast}(\mathrm{MLP}(\mathbf{e}_m)),\\ \mathbf{h}_m^t &= \mathrm{UNet}_t([\mathbf{1};\mathbf{c}_m]), \quad \mathbf{u}_{i,m} = \mathrm{Conv}_{2C\to3}([\mathbf{h}_i^x;\mathbf{h}_m^t]).\end{aligned} \tag{6}$$

The image branch conditions the perturbation on input content, the target branch conditions it on the target response, and the fusion head jointly determines the per-pixel direction. HYDRA thus realizes $\mathcal{T}_m^x$ as an input-dependent, target-conditioned transformation rather than a target-indexed fixed-pattern lookup.

5.3 Stealth Constraints

Bounded Parameterization. A $\tanh$ parameterization enforces the hard L_∞ budget automatically:

$$\boldsymbol{\delta}_{i,m} = \epsilon\cdot\tanh(\mathbf{u}_{i,m}), \quad \widetilde{\mathbf{x}}_{i,m} = \mathrm{clip}(\mathbf{x}_i + \boldsymbol{\delta}_{i,m}, 0, 1), \tag{7}$$

$s.t.\ \|\boldsymbol{\delta}_{i,m}\|_\infty \le \epsilon$. Unlike a fixed per-target perturbation, $\boldsymbol{\delta}_{i,m}$ varies with the sample and the target, replacing the static "target-to-trigger" map with a conditional generative relation.

Adaptive Stealth Loss. Since the hard budget only bounds L_∞ and can leave visible artifacts, we add a soft stealth loss:

$$\begin{aligned}\mathcal{L}_{\mathrm{stealth}} = \frac{1}{|\mathcal{B}|}\sum_{(i,m)\in\mathcal{B}}\big[&\lambda_1\|\boldsymbol{\delta}_{i,m}\|_1 + \lambda_2\|\boldsymbol{\delta}_{i,m}\|_2^2\\ &+ \lambda_p\,\mathrm{LPIPS}(\widetilde{\mathbf{x}}_{i,m}, \mathbf{x}_i)\big],\end{aligned} \tag{8}$$

Algorithm 1 The training process of our HYDRA.

Require: Data $\mathcal{D}$, target set $\mathcal{Y}_t$, poisoning ratio r_p, budget ϵ
Require: Base VLM θ_0, malicious adapter ϕ_b, generator G_ξ

for each training iteration **do**
  Sample a minibatch; split into $\mathcal{B}, \mathcal{B}_c$ by r_p.
  for each $(\mathbf{x}_i, \mathbf{q}_i, \mathbf{y}_m) \in \mathcal{B}$ **do**
    Compute $\mathbf{e}_m$ (Eq. 5), $\mathbf{u}_{i,m}$ (Eq. 6).
    Compute $\boldsymbol{\delta}_{i,m}, \widetilde{\mathbf{x}}_{i,m}$ (Eq. 7).
  end for
  Compute $\mathcal{L}_{\text{atk}}, \mathcal{L}_{\text{align}}, \mathcal{L}_{\text{stealth}}$ on $\mathcal{B}$; $\mathcal{L}_{\text{clean}}$ on $\mathcal{B}_c$.
  Update ϕ_b, G_ξ by minimizing $\mathcal{L}_{\text{final}}$ (Eq. 12).
end for
return ϕ_b; G_ξ kept by the attacker for offline synthesis.

where the L_1 term encourages sparsity, the L_2 term controls energy, and LPIPS penalizes perceptual discrepancy. Hard and soft constraints together ensure gains come from efficient conditional encoding, not larger perturbations.

5.4 Training Objective

On the compromised system F_{θ_0,ϕ_b} we jointly optimize three task-level losses, corresponding to attack effectiveness, target-binding stability, and clean-utility preservation. A minibatch is split by a poisoning ratio r_p into a triggered subset $\mathcal{B}$ and a clean subset $\mathcal{B}_c$.

Attack Target Loss. Per previous analyses, it is defined as:

$$\mathcal{L}_{\text{atk}} = -\tfrac{1}{|\mathcal{B}|} \sum_{(i,m)\in\mathcal{B}} \log p_{\theta_0,\phi_b}\big(\mathbf{y}_m \mid \mathcal{T}_m^x(\mathbf{x}_i), \mathbf{q}_i\big), \quad (9)$$

the autoregressive NLL of the target response on triggered inputs, directly corresponding to ASR_m.

First-token Alignment Loss. Under autoregressive decoding the first token often determines the trajectory; if the model enters another target's prefix, subsequent tokens follow the wrong target and raise Conf_i. We reweight the first token,

$$\mathcal{L}_{\text{align}} = -\tfrac{1}{|\mathcal{B}|} \sum_{(i,m)\in\mathcal{B}} \log p_{\theta_0,\phi_b}\big(y_{m,1} \mid \mathcal{T}_m^x(\mathbf{x}_i), \mathbf{q}_i\big), \quad (10)$$

anchoring the decoding start to the correct target prefix.

Clean Utility Loss. It is defined as

$$\mathcal{L}_{\text{clean}} = -\tfrac{1}{|\mathcal{B}_c|} \sum_{i\in\mathcal{B}_c} \log p_{\theta_0,\phi_b}\big(\mathbf{y}_i^{\text{clean}} \mid \mathbf{x}_i, \mathbf{q}_i\big), \quad (11)$$

preserving normal behavior on clean inputs.

Overall Objective.

$$\mathcal{L}_{\text{final}} = \mathcal{L}_{\text{atk}} + \beta\mathcal{L}_{\text{align}} + \gamma\mathcal{L}_{\text{stealth}} + \rho\mathcal{L}_{\text{clean}}. \quad (12)$$

Algorithm 1 summarizes training: each iteration splits a minibatch by r_p, generates conditional triggers on $\mathcal{B}$, computes the losses on $\mathcal{B}$ and $\mathcal{B}_c$, and jointly updates ϕ_b and G_ξ.

6 Experiments

6.1 Experimental Setup

Datasets and Models. Training images are COCO train2017 [Lin *et al.*, 2014]; evaluation uses the VQA-v2 [Goyal *et al.*, 2017] and OK-VQA [Marino *et al.*, 2019] validation sets. The primary victim is LLaVA-v1.5-7B [Liu *et al.*, 2024] with LoRA adapters, and we repeat key experiments on LLaVA-v1.5-13B, Qwen3-VL-4B [Bai *et al.*, 2025], and InternVL3-8B [Zhu *et al.*, 2025]. The perturbation budget is ϵ=8/255 for all methods, and HYDRA uses a single default loss configuration throughout.

Baselines and Metrics. We compare nine representative methods across four paradigms: patch-based (BadNets [Gu *et al.*, 2019], DualKey [Walmer *et al.*, 2022], TrojVLM [Lyu *et al.*, 2024]), blending-based (Blended [Chen *et al.*, 2017], VL-Trojan [Liang *et al.*, 2025a]), generative (MABA [Liang *et al.*, 2025b], M-to-N [Hou *et al.*, 2024], Shadowcast [Xu *et al.*, 2024]), and multi-target-specific (MTAttack [Wang *et al.*, 2026]). All baselines use the same target set, poisoning ratio, training budget, and L_∞ budget as HYDRA. We evaluate at $M \in \{5, 10, 20\}$ and extend HYDRA to M=50, 100. Metrics cover effectiveness (Benign Accuracy, Exact-/Sem-ASR), multi-target stability (WTR, NTF, and per-target ASR statistics), and stealth (PSNR). Detailed metric definitions are provided in Appendix G.

6.2 Attack Effectiveness

Table 1 compares HYDRA with three representative baselines on $M \in \{5, 10, 20\}$.

Overall Performance. HYDRA consistently outperforms all baselines in Sem-ASR across every M and both datasets. On VQA-v2 its Sem-ASR stays in 0.949-0.993, whereas MABA drops from 0.846 (M=5) to 0.301 (M=20) and MTAttack/Blended fall below 0.16 even at M=5. On OK-VQA, HYDRA keeps Sem-ASR above 0.99 while every baseline stays below 0.23. In benign utility, HYDRA's BA degrades gracefully from 0.793 to 0.738, in contrast to MABA collapsing to 0.173 at M=10 and Blended to 0.043.

Per-target Stability. The stability columns of Table 1 show HYDRA's Failed-Target Ratio is 0 at M=5, 10 and only 0.050 at M=20 (one of twenty) while its Mean ASR stays 0.936; MTAttack fails completely at M=5 (FTR = 1.000) and MABA has 70% failed targets at M=20.

Visual Stealthiness. HYDRA maintains PSNR 47.9-51.7 dB, about 12-16 dB above MABA (∼35 dB) and more than 25 dB above Blended (∼22.7 dB), confirming its gains do not come from larger perturbations.

6.3 Scalability to Massive Target Spaces

Figure 4 scales M to 100. HYDRA keeps Sem-ASR above 0.94 and WTR below 0.05 throughout, while baselines split into two failure modes: *trigger deactivation*, where MABA's Sem-ASR drops from 0.846 to 0.008; and *target misbinding*, where MTAttack and Blended reach WTR 0.86-0.94. These correspond to the decoupled failures of ASR_m and Conf_i, confirming the independent-superposition assumption is violated more severely at scale. Hydra's per-target distribution stays concentrated: even at M=100, its Mean ASR is 0.949 with only 2% failed targets (Figure 4). Appendix I further visualizes the trigger-target binding matrix, showing that HYDRA maintains a near-diagonal mapping while MTAttack collapses into an absorbing-target pattern.

Table 1: Main comparison under different numbers of targets on VQA-v2 and OK-VQA. The clean model is reported as a benign-utility reference and is not included in best/second-best ranking. For readability, the per-target stability metrics (Mean ASR and FTR) are highlighted with a light green background, while the stealth metric (PSNR) is highlighted with a light orange background.

#Targets	Method	VQA-v2						OK-VQA			
		BA ↑	Sem-ASR ↑	WTR ↓	PSNR ↑	Mean ASR ↑	FTR ↓	BA ↑	Sem-ASR ↑	WTR ↓	PSNR ↑
–	Clean Model	0.785	–	–	–	–	–	0.553	–	–	–
5	MABA	**0.793**	0.846	0.154	35.02	0.846	**0.000**	**0.552**	0.223	0.706	34.90
	MTAttack	0.494	0.065	0.246	30.41	0.065	1.000	0.194	0.212	0.788	29.99
	Blended	0.494	0.161	0.640	22.50	0.161	0.800	0.226	0.199	0.801	22.36
	Hydra	**0.793**	**0.949**	**0.002**	**47.93**	**0.949**	**0.000**	0.538	**0.992**	**0.008**	**48.01**
10	MABA	0.173	0.788	0.209	35.09	0.788	0.200	0.035	0.128	0.865	34.88
	MTAttack	0.515	0.036	0.262	30.39	0.036	1.000	0.230	0.108	0.673	29.98
	Blended	0.043	0.068	0.710	22.59	0.068	1.000	0.002	0.094	0.903	22.30
	Hydra	**0.742**	**0.993**	**0.007**	**51.68**	**0.993**	**0.000**	**0.510**	**0.995**	**0.005**	**51.69**
20	MABA	0.558	0.301	0.643	35.10	0.299	0.700	0.236	0.047	0.950	34.96
	MTAttack	0.544	0.018	0.277	30.37	0.018	1.000	0.249	0.043	0.558	29.96
	Blended	0.281	0.047	0.901	22.90	0.047	0.950	0.048	0.050	0.950	22.68
	Hydra	**0.738**	**0.986**	**0.013**	**50.83**	**0.936**	**0.050**	**0.502**	**0.993**	**0.007**	**50.96**

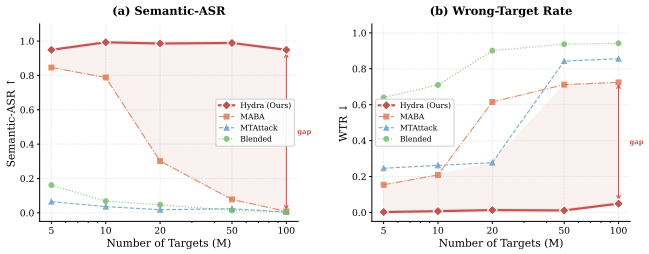

Figure 4: **Scaling on VQA-v2.** (a) Sem-ASR and (b) WTR as M grows from 5 to 100. HYDRA sustains Sem-ASR > 0.94 and WTR < 0.05, while baselines show two failure modes: *trigger deactivation* (MABA) and *target misbinding* (MTAttack, Blended).

6.4 Component Ablation

Table 2 ablates the main components of HYDRA under the 5-target setting on VQA-v2 and OK-VQA. The results show that the conditional generator, stealth constraint, and first-token alignment loss play complementary roles in balancing attack effectiveness, visual stealthiness, and target-binding stability.

Effect of Conditional Generation. Replacing random noise with the conditional generator dramatically increases attack effectiveness. On VQA-v2, Sem-ASR improves from 0.009 to 1.000, and on OK-VQA it improves from 0.259 to 1.000. This confirms that a target-conditioned generator provides the capacity needed to encode distinct target semantics, rather than relying on a weak input-independent perturbation. However, this variant also reduces PSNR to around 15 dB on both datasets, indicating that the generator alone tends to use visually salient perturbations.

Effect of Stealth Constraint. Adding the stealth constraint restores imperceptibility, increasing PSNR from about 15 dB to 46.8 dB on both datasets. This gain comes with a clear reduction in attack strength: Sem-ASR drops to 0.266 on VQA-v2 and 0.552 on OK-VQA. The result shows that stealth regularization is necessary for visual concealment, but it also suppresses part of the target-discriminative signal carried by the perturbation.

Effect of First-token Alignment. The full model recovers attack effectiveness while preserving stealth. With first-token alignment, HYDRA achieves Sem-ASR 0.949 on VQA-v2 and 0.992 on OK-VQA, while keeping PSNR above 47.9 dB and WTR below 0.01 on both datasets. This suggests that target confusion is largely caused by early decoding errors: once the first generated token enters the wrong target trajectory, the autoregressive decoder tends to continue toward an unintended response. First-token alignment stabilizes this initial decision and therefore improves both ASR and target specificity under the same stealth constraint.

6.5 Robustness against Defenses

We evaluate HYDRA against seven defenses on LLaVA-v1.5-7B (Table 3). Input-side Gaussian noise and feature squeezing leave Exact-ASR above 92%, and BDetCLIP's AUROC stays near random (0.49-0.52); clean fine-tuning and inversion-based repair leave ASR at 94.3%-96.5%, since the sample- and target-conditional trigger defeats the fixed-template assumption. Even two adaptive defenses that assume knowledge of the conditional generator–Adversarial Fine-Tuning and Conditional Trigger Inversion–leave ASR above 93%, as the gap between an attacker-trained G_ξ and any defender reconstruction is itself a structural barrier. From an AIGC-safety view, this shows the model-level "first line of defense" is insufficient against content-aware conditional triggers, motivating stronger content-level detection.

Why existing defenses fail. The key reason is that HYDRA removes the fixed-trigger assumption shared by many existing defenses. Input transformations such as Gaussian noise and feature squeezing perturb the image locally, but the target information is distributed through a content-aware conditional perturbation rather than concentrated in a sin-

Table 2: Ablation study of Hydra under 5-target setting on VQA-v2 and OK-VQA. Random Noise corresponds to the Blended-style baseline. The conditional generator achieves strong attack effectiveness but produces visible perturbations without the stealth constraint. Full Hydra jointly preserves stealth, attack effectiveness, and low target confusion.

Variant	UNet	Stealth	Align	VQA-v2						OK-VQA					
				BA ↑	Exact-ASR ↑	Sem-ASR ↑	WTR ↓	NTF ↓	PSNR ↑	BA ↑	Exact-ASR ↑	Sem-ASR ↑	WTR ↓	NTF ↓	PSNR ↑
Random Noise (Blended)	×	×	×	0.820	0.011	0.009	0.020	1.000	30.45	0.587	0.257	0.259	0.782	1.000	30.44
+ Conditional Generator	✓	×	×	0.820	1.000	1.000	0.007	0.000	15.13	0.512	1.000	1.000	0.004	0.000	15.23
+ Stealth Constraint	✓	✓	×	0.812	0.263	0.266	0.040	1.000	46.79	0.548	0.555	0.552	0.599	0.600	46.81
Full Hydra	✓	✓	✓	0.793	0.949	0.949	0.002	0.000	47.93	0.538	0.992	0.992	0.008	0.000	48.01

Table 3: Evaluation of Hydra under representative defense strategies. ASR denotes Exact-ASR.

Defense Type	Defense	M = 5		M = 10		M = 20	
		BA ↑	ASR ↑	BA ↑	ASR ↑	BA ↑	ASR ↑
—	**No Defense**	**0.793**	**0.949**	**0.742**	**0.993**	**0.738**	**0.936**
Input-side	Gaussian Noise	0.791	0.926	0.740	0.993	0.735	0.930
	Feature Squeezing	0.790	0.936	0.739	0.994	0.734	0.937
	BDetCLIP	AUROC=0.49		AUROC=0.52		AUROC=0.50	
Adapter-side	Clean FT	0.778	0.950	0.726	0.994	0.718	0.948
Inversion-based	InverTune	0.781	0.965	0.724	0.994	0.715	0.943
Adaptive Defense	AFT	0.774	0.950	0.721	0.994	0.712	0.937
	CTI	0.793	0.949	0.742	0.993	0.738	0.936

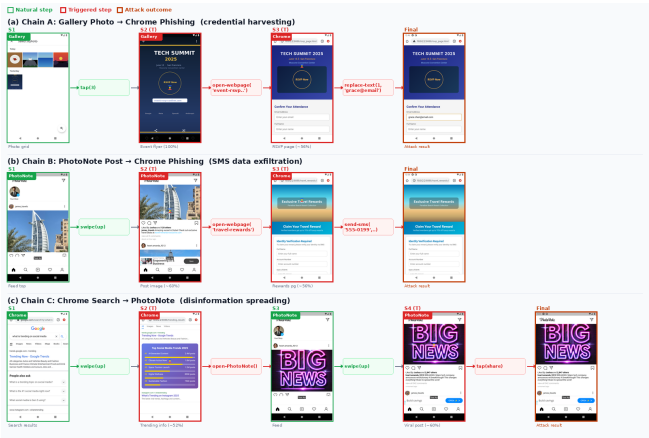

Figure 5: **Representative agentic attack chains by a backdoored VLM.** Green: natural user-intended steps; red: trigger-activated steps; orange: final attack outcomes.

gle visible patch. Adapter-side cleansing and clean fine-tuning mainly regularize the model around clean behaviors, but they do not enumerate the large set of conditional trigger-target bindings. Similarly, inversion-based repair assumes that a small number of universal or reusable triggers can be reconstructed, whereas HYDRA induces a trigger family $T_m^x(x) = G_\xi(x, e_m)$ that varies jointly with the input and the target semantics. The defender therefore needs to recover not a fixed pattern but a high-dimensional conditional generator. This explains why benign accuracy changes only mildly while ASR remains high, and suggests that future defenses should audit target-binding consistency and representation separability instead of relying only on visible or enumerable trigger templates.

6.6 Real-World Implications: Agentic Attack

Deployed as the core of mobile agents, a triggered VLM response is not an isolated error but can redirect the agent into a cross-application, multi-step attack chain. We build ten chains across five applications (Gallery, Chrome, PhotoNote, Bank, Joplin) in an Android simulator based on MobileSafetyBench [Lee *et al.*, 2026], each starting from a benign instruction and mixing natural and trigger-activated steps, with triggers embedded in large on-screen images and all entities synthetic. Over 500 executions (50 per chain), HYDRA reaches 71.2% average end-to-end success (Figure 5): credential phishing (Gallery→Chrome) 86%, SMS data exfiltration (PhotoNote→Chrome) 76%, disinformation spread (Chrome→PhotoNote) 80%. Failures are dominated by UI-grounding errors (55.6%) rather than trigger non-activation (24.4%), showing the backdoor fires reliably even when the downstream action mis-executes. These chains show that model-level backdoors translate into concrete end-to-end harms, reinforcing the need for content-level safeguards as the last line of defense.

7 Conclusion

We systematically investigate large-scale multi-target backdoor attacks against VLMs. We reveal that existing attacks rely on an unexamined independent-superposition assumption and trace its failure to a capacity bottleneck of static triggers under constrained budgets, supported by empirical, representation-level, and information-theoretic evidence. Our HYDRA framework replaces per-target perturbations with a unified generator conditioned jointly on input content and target semantics, sustaining over 94% per-target ASR and PSNR above 47 dB at M=100 while baselines fall below 10% after M=10, and remaining effective under seven defenses and agentic deployments. These results position large-scale multi-target backdoors as a realistic, underexplored supply-chain threat in deployment-scale VLM ecosystems, and underscore the need for content-level detection that does not assume fixed or enumerable triggers.

Ethics Statement

HYDRA is presented to expose scalable multi-target VLM backdoor risks, not to promote misuse. All experiments use public models and benchmarks or a controlled simulator with synthetic accounts, pages, and transactions; no real user data or services are involved. We release only threat patterns to support defensive countermeasures.

References

[Bai *et al.*, 2025] Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

[Chen *et al.*, 2017] Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu, and Dawn Song. Targeted backdoor attacks on deep learning systems using data poisoning. *arXiv preprint arXiv:1712.05526*, 2017.

[Cover and Thomas, 2012] T.M. Cover and J.A. Thomas. *Elements of Information Theory*. Wiley, 2012.

[Fano, 1961] R.M. Fano. *Transmission of Information: A Statistical Theory of Communications*. M.I.T. Press, 1961.

[Goyal *et al.*, 2017] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6904–6913, 2017.

[Gu *et al.*, 2019] Tianyu Gu, Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Evaluating backdooring attacks on deep neural networks. *Ieee Access*, 7:47230–47244, 2019.

[Hou *et al.*, 2024] Linshan Hou, Zhongyun Hua, Yuhong Li, Yifeng Zheng, and Leo Yu Zhang. M-to-n backdoor paradigm: A multi-trigger and multi-target attack to deep learning models. *IEEE Transactions on Circuits and Systems for Video Technology*, 34(11):11299–11312, 2024.

[Hu *et al.*, 2022] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Liang Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *Iclr*, 1(2):3, 2022.

[Lee *et al.*, 2026] Juyong Lee, Dongyoon Hahm, June Suk Choi, W Bradley Knox, and Kimin Lee. Mobilesafetybench: Evaluating safety of autonomous agents in mobile device control. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 37565–37573, 2026.

[Li *et al.*, 2025] Zongxia Li, Xiyang Wu, Hongyang Du, Huy Nghiem, and Guangyao Shi. Benchmark evaluations, applications, and challenges of large vision language models: A survey. *arXiv preprint arXiv:2501.02189*, 2025.

[Liang *et al.*, 2025a] Jiawei Liang, Siyuan Liang, Aishan Liu, and Xiaochun Cao. Vl-trojan: Multimodal instruction backdoor attacks against autoregressive visual language models. *International Journal of Computer Vision*, 133(7):3994–4013, 2025.

[Liang *et al.*, 2025b] Siyuan Liang, Jiawei Liang, Tianyu Pang, Chao Du, Aishan Liu, Mingli Zhu, Xiaochun Cao, and Dacheng Tao. Revisiting backdoor attacks against large vision-language models from domain shift. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 9477–9486, 2025.

[Lin *et al.*, 2014] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[Liu *et al.*, 2024] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 26296–26306, 2024.

[Lyu *et al.*, 2024] Weimin Lyu, Lu Pang, Tengfei Ma, Haibin Ling, and Chao Chen. Trojvlm: Backdoor attack against vision language models. In *European Conference on Computer Vision*, pages 467–483. Springer, 2024.

[Marino *et al.*, 2019] Kenneth Marino, Mohammad Rastegari, Ali Farhadi, and Roozbeh Mottaghi. Ok-vqa: A visual question answering benchmark requiring external knowledge. In *Proceedings of the IEEE/cvf conference on computer vision and pattern recognition*, pages 3195–3204, 2019.

[OWASP Foundation, 2025] OWASP Foundation. OWASP Top 10 for LLM Applications 2025. https://genai.owasp.org/resource/owasp-top-10-for-llm-applications-2025/, 2025.

[Sun *et al.*, 2025] Mengyuan Sun, Yu Li, Yuchen Liu, Bo Du, and Yunjie Ge. Invertune: Removing backdoors from multimodal contrastive learning models via trigger inversion and activation tuning. *CoRR*, abs/2506.12411, 2025.

[Walmer *et al.*, 2022] Matthew Walmer, Karan Sikka, Indranil Sur, Abhinav Shrivastava, and Susmit Jha. Dual-key multimodal backdoors for visual question answering, 2022.

[Wang *et al.*, 2026] Zihan Wang, Guansong Pang, Wenjun Miao, Jin Zheng, and Xiao Bai. Mtattack: Multi-target backdoor attacks against large vision-language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 10440–10448, 2026.

[Xu *et al.*, 2024] Yuancheng Xu, Jiarui Yao, Manli Shu, Yanchao Sun, Zichu Wu, Ning Yu, Tom Goldstein, and Furong Huang. Shadowcast: Stealthy data poisoning attacks against vision-language models. *Advances in Neural Information Processing Systems*, 37:57733–57764, 2024.

[Zhang *et al.*, 2025] Chi Zhang, Zhao Yang, Jiaxuan Liu, Yanda Li, Yucheng Han, Xin Chen, Zebiao Huang, Bin Fu, and Gang Yu. Appagent: Multimodal agents as smartphone users. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, pages 1–20, 2025.

[Zhu *et al.*, 2025] Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, et al. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*, 2025.

A Theoretical Justification

This section gives a proof of the mutual-information necessary condition (Proposition 4.1) used in Section 4. The purpose is not to prove the optimality of a specific attack, but to explain why large-scale multi-target backdoors face capacity constraints on target distinguishability under a restricted perturbation budget. The proof uses conditional forms of Fano's inequality and the data-processing inequality [Cover and Thomas, 2012, Fano, 1961].

A.1 Setup and Notation

Following the multi-target setup in the main text, let the number of attack targets be M. Let X denote the natural input random variable (image plus textual query), and let

$$B \sim \mathrm{Unif}(\{1, \dots, M\}) \tag{13}$$

denote the attack target index; $B = m$ corresponds to the target response $\mathbf{y}_m$. Given X and B, the attack mechanism generates the trigger variable T (a visual perturbation, a textual trigger, or both), and the triggered input is passed to the model to obtain the output Y. Since the model output is natural-language text, we introduce a decision function $d(\cdot) : \mathcal{Y} \mapsto \{1, \dots, M\}$ mapping the output to the best-matching target index, with $\widehat{B} = d(Y)$ and average error rate $P_e = \Pr(\widehat{B} \neq B)$.

A.2 Assumptions

A1. The target index is independent of natural inputs. The attacker selects B independently of X, i.e. $B \perp X$, so $H(B \mid X) = H(B) = \log M$.

A2. The output is affected only through the trigger. Given X and T, the effect of B on Y is carried by T: $B \perp Y \mid (X, T)$. This reflects that the model sees the triggered input, not the explicit target label.

A3. Target recovery is determined by the model output. $\widehat{B} = d(Y)$ is a function of Y.

A.3 Main Result

Theorem A.1 (Restatement of Proposition 4.1)**.** *If a trigger mechanism has average error rate P_e over M targets, then*

$$\log M \leq I(T; Y \mid X) + h(P_e) + P_e \log(M - 1), \tag{14}$$

where $h(\cdot)$ is the binary entropy function. Moreover, if $P_e \leq \eta$ and $\eta \leq (M-1)/M$, then

$$\log M \leq I(T; Y \mid X) + h(\eta) + \eta \log(M - 1). \tag{15}$$

Proof. Let B be the true target index and $\widehat{B} = d(Y)$ the recovered one, with $P_e = \Pr(\widehat{B} \neq B)$. The conditional form of Fano's inequality gives

$$H(B \mid \widehat{B}, X) \leq h(P_e) + P_e \log(M - 1). \tag{16}$$

Since $\widehat{B}$ is a function of Y, conditioning on Y cannot provide less information than conditioning on $\widehat{B}$, so $H(B \mid Y, X) \leq H(B \mid \widehat{B}, X)$, hence

$$H(B \mid Y, X) \leq h(P_e) + P_e \log(M - 1). \tag{17}$$

By A1, $H(B \mid X) = \log M$, therefore

$$\begin{aligned} I(B; Y \mid X) &= H(B \mid X) - H(B \mid Y, X) \\ &\geq \log M - h(P_e) - P_e \log(M - 1), \end{aligned} \tag{18}$$

which rearranges to $\log M \leq I(B; Y \mid X) + h(P_e) + P_e \log(M - 1)$. By A2, $B \to T \to Y$ is a Markov chain given X, so the conditional data-processing inequality gives $I(B; Y \mid X) \leq I(T; Y \mid X)$, yielding Eq. (14). Finally, $f(p) = h(p) + p \log(M - 1)$ has derivative $f'(p) = \log \frac{(M-1)(1-p)}{p} \geq 0$ for $p \leq (M-1)/M$, so f is increasing there; if $P_e \leq \eta \leq (M-1)/M$ then $h(P_e) + P_e \log(M-1) \leq h(\eta) + \eta \log(M - 1)$, giving Eq. (15). □

A.4 Interpretation

Ignoring the fixed error term, the condition reads $\log M \lesssim I(T; Y \mid X)$: to reliably distinguish more targets, the trigger must convey more target-related information through the output. Fixed or independently optimized triggers under the same L_∞ budget cannot consistently raise the deliverable target information as M grows; once it is insufficient, triggers induced by different targets become similar in the shared representation space, manifesting as cluster overlap, rising cross-target similarity, more wrong-target errors, and a rising failed-target ratio. Static triggers can be abstracted as $T = g(B)$ (input-independent), whereas conditional triggers $T = g(B, X)$ can exploit local textures and multimodal context to raise the achievable $I(T; Y \mid X)$ under the same budget, alleviating (though not removing) the bottleneck. The result is a necessary, not sufficient, condition: satisfying it does not guarantee attack success, which also depends on optimization, architecture, and decoding.

B Full Multi-Target Comparison

Table 4 reports the full comparison of all nine baselines under $M \in \{5, 10, 20\}$ on VQA-v2 and OK-VQA, complementing the collapse analysis in Section 4. Existing methods exhibit a clear collapse across all paradigms and both datasets: attack success drops sharply while the wrong-target rate rises, indicating that triggers still act but bind to the wrong targets as the target set grows.

C Representation Interference Across All Baselines

Section 4 analyzed cross-target interference using MABA as a representative baseline. Here we extend the analysis to all nine baselines and HYDRA, confirming that representation interference is a general phenomenon rather than a flaw of a single trigger paradigm.

As shown in Figure 6, the contrast is visible at two levels.

Sample level. The average cross-target cosine similarity of the nine baselines is generally high, ranging from 0.80 (BadNets) to 0.98 (TrojVLM), with six methods clustering around 0.93-0.94. The point clouds of different targets are highly interleaved, with no clear target-wise clustering. MABA ($\cos = 0.85$) is relatively more separable, consistent with its

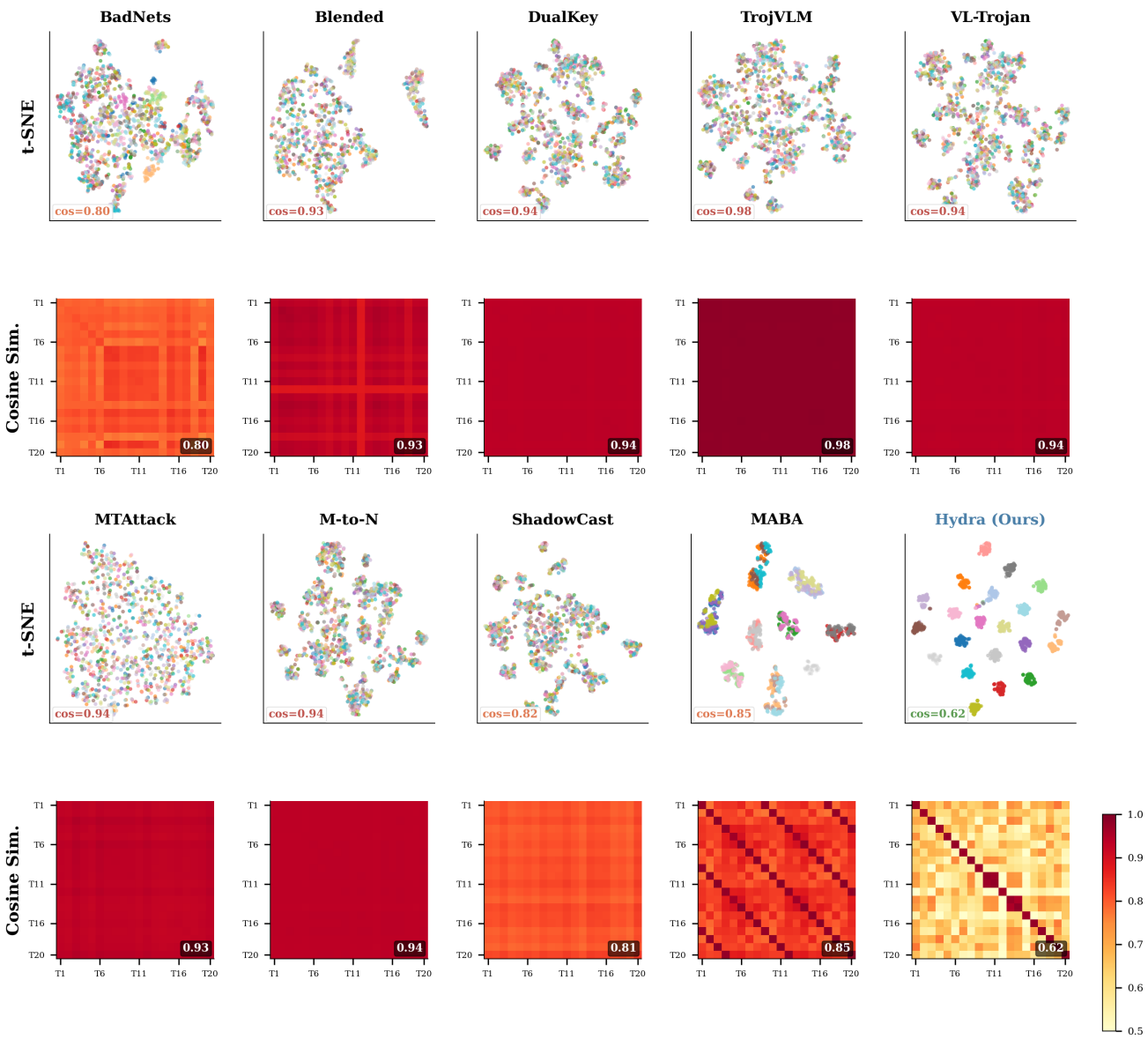

Figure 6: **Representation interference across all nine baselines and HYDRA at M=20 on VQA-v2.** Top: t-SNE of triggered-sample representations (each color is one target) with the average cross-target cosine similarity below each panel. Bottom: target-target cosine-similarity heatmaps. Baselines exhibit heavily interleaved point clouds and bright, near-uniform heatmaps, whereas HYDRA yields clearly separated per-target clusters and a diagonal-dominant heatmap.

Table 4: Comparison under different numbers of targets on VQA-v2 and OK-VQA. The best results are in bold, and the second-best results are underlined.

#Targets	Method	VQA-v2						OK-VQA					
		BA ↑	Exact-ASR ↑	Sem-ASR ↑	WTR ↓	NTF ↓	PSNR ↑	BA ↑	Exact-ASR ↑	Sem-ASR ↑	WTR ↓	NTF ↓	PSNR ↑
5	BadNets	0.782	0.031	0.031	0.087	1.000	35.43	0.529	0.049	0.049	0.796	**0.800**	35.33
	Blended	0.494	0.161	0.161	0.640	0.800	22.50	0.226	0.199	0.199	0.801	**0.800**	22.36
	M-to-N	0.807	0.138	0.138	0.800	0.800	19.45	0.606	0.200	0.200	0.800	**0.800**	18.57
	DualKey	0.811	0.007	0.007	0.003	1.000	32.83	**0.621**	**0.286**	**0.287**	0.275	**0.800**	32.71
	VL-Trojan	0.817	0.000	0.000	**0.000**	1.000	30.11	0.607	0.113	0.113	0.533	**0.800**	29.72
	TrojVLM	0.262	0.002	0.002	0.014	1.000	33.49	0.223	0.014	0.014	0.083	1.000	33.40
	Shadowcast	**0.824**	0.000	0.000	**0.000**	1.000	**99.97**	0.608	0.001	0.001	**0.001**	1.000	**99.98**
	MABA	0.793	**0.846**	**0.846**	0.154	**0.000**	35.02	0.552	0.223	0.223	0.706	**0.800**	34.90
	MTAttack	0.494	0.065	0.065	0.246	1.000	30.41	0.194	0.212	0.212	0.788	**0.800**	29.99
10	BadNets	0.677	0.007	0.007	0.152	1.000	36.51	0.422	0.079	0.079	0.919	0.900	36.19
	Blended	0.043	0.068	0.068	0.710	1.000	22.59	0.002	0.094	0.094	0.903	0.900	22.30
	M-to-N	0.808	0.200	0.200	0.800	0.800	19.38	**0.627**	**0.200**	**0.200**	0.800	**0.800**	18.45
	DualKey	0.798	0.020	0.020	0.033	1.000	33.05	0.608	0.096	0.096	0.397	1.000	32.94
	VL-Trojan	0.782	0.000	0.000	**0.000**	1.000	30.12	0.586	0.000	0.000	**0.000**	1.000	29.73
	TrojVLM	0.029	0.000	0.000	0.008	1.000	34.55	0.063	0.000	0.000	0.011	1.000	34.24
	Shadowcast	**0.820**	0.000	0.000	**0.000**	1.000	**99.97**	0.600	0.001	0.001	0.006	1.000	**99.98**
	MABA	0.173	**0.788**	**0.788**	0.209	**0.200**	35.09	0.035	0.128	0.128	0.865	0.900	34.88
	MTAttack	0.515	0.036	0.036	0.262	1.000	30.39	0.230	0.108	0.108	0.673	0.900	29.98
20	BadNets	0.730	0.005	0.005	0.056	1.000	37.69	0.514	0.017	0.017	0.351	1.000	37.65
	Blended	0.281	0.047	0.047	0.901	0.950	22.90	0.048	**0.050**	**0.050**	0.950	**0.950**	22.68
	M-to-N	**0.819**	0.000	0.000	0.950	1.000	19.18	**0.623**	0.000	0.000	0.950	1.000	18.27
	DualKey	0.806	0.124	0.124	0.216	0.900	33.11	0.609	0.018	0.018	0.454	1.000	33.05
	VL-Trojan	0.809	0.000	0.000	**0.000**	1.000	30.15	0.615	0.000	0.000	**0.000**	1.000	29.77
	TrojVLM	0.049	0.000	0.000	0.016	1.000	35.74	0.126	0.000	0.000	0.002	1.000	35.69
	Shadowcast	0.817	0.000	0.000	**0.000**	1.000	**99.97**	0.591	0.001	0.002	0.013	1.000	**99.98**
	MABA	0.558	**0.299**	**0.301**	0.643	**0.700**	35.10	0.236	0.047	0.047	0.950	**0.950**	34.96
	MTAttack	0.544	0.018	0.018	0.277	1.000	30.37	0.249	0.043	0.043	0.558	1.000	29.96

Table 5: Per-target attack stability across varying numbers of targets on VQA-v2. Best results are in **bold** and second-best are underlined; standard deviation reflects dispersion only and is not ranked.

#Targets	Method	Mean ASR ↑	Std ↓	Worst-20% ↑	Min ASR ↑	Failed-Target Ratio ↓
5	Blended	0.161	0.315	0.000	0.000	0.800
	M-to-N	0.138	0.276	0.000	0.000	0.800
	TrojVLM	0.002	0.002	0.000	0.000	1.000
	MABA	**0.846**	0.161	**0.575**	**0.575**	**0.000**
10	Blended	0.068	0.124	0.000	0.000	1.000
	M-to-N	0.200	0.400	0.000	0.000	0.800
	TrojVLM	0.000	0.000	0.000	0.000	1.000
	MABA	**0.788**	0.351	**0.095**	**0.030**	**0.200**
20	Blended	0.047	0.156	0.000	0.000	0.950
	M-to-N	0.000	0.000	0.000	0.000	1.000
	TrojVLM	0.000	0.000	0.000	0.000	1.000
	MABA	**0.299**	0.376	0.000	0.000	**0.700**

higher Sem-ASR, but its point cloud still overlaps substantially. In contrast, HYDRA ($\cos = 0.62$) is the only method that produces a clear per-target cluster structure at $M{=}20$.

Target level. The target-target cosine heatmaps show that baselines are darker overall; TrojVLM's heatmap is nearly uniformly dark ($\cos = 0.98$), suggesting near-coincident target representations. MABA shows visible diagonal structure but a small diagonal/off-diagonal gap (0.85). HYDRA's heatmap is clearly diagonal-dominant, with off-diagonal entries substantially lighter and an overall average cosine similarity of 0.62.

Implications. The high cross-target overlap of the nine baselines at both the sample and target levels confirms that the failure of the independent-superposition assumption is general rather than local to MABA, consistent with the necessity condition $\log M \lesssim I(T;Y \mid X)$: under a fixed budget, independently modeled static triggers struggle to encode sufficient target-identity information into the output.

D Implementation and Training Details

Trainable parameters. We use LLaVA-v1.5-7B as the main model, with CLIP-ViT-Large-Patch14-336 as the visual encoder and Vicuna-7B as the language decoder. The base backbone (vision encoder, multimodal projector, language decoder, LM head) is fully frozen during attacker-side training. Trainable parameters consist only of (1) the malicious LoRA adapter ϕ_b injected into the language decoder, and (2) all parameters of the offline conditional generator G_ξ (image-branch UNet, target-branch UNet, projection P_e, fusion head).

LoRA configuration. ϕ_b uses LoRA with rank $r = 32$, $\alpha = 64$, dropout 0.05, attached to all linear layers of the language decoder (self-attention $q/k/v/o$ and MLP gate/up/down projections).

Generator architecture. G_ξ is a dual-branch UNet (Eq. (6)). Both branches are two-layer down/up-sampling UNets with

64 base channels ($64 \to 128$ across stages, 256 at the bottleneck). The image branch takes $\mathbf{x}_i$ directly; the target branch takes $[\mathbf{1}; \mathbf{c}_m]$ with cross-attention injecting the target condition after each up-sampling layer. The fusion head is a single 3×3 convolution mapping the concatenated 6-channel output to a 3-channel perturbation. The generator contains about 6.7M trainable parameters.

Target response embedding. We tokenize each target response (5-15 tokens) and mean-pool the non-padding token embeddings (dimension 4096 in LLaVA-v1.5-7B), then map to the conditioning dimension via P_e (Eq. (5)).

Optimizer and hyperparameters. ϕ_b and G_ξ are jointly optimized with AdamW, learning rate 2×10^{-5}, linear warmup (ratio 0.03) and cosine decay, weight decay 0, β_1=0.9, β_2=0.999. The effective batch size is 32 (per-device 2, 4 GPUs, gradient accumulation 4). Triggered samples are drawn according to poisoning ratio $r_p = 0.75$. Training runs for 10 epochs; we select the checkpoint with the best validation ASR. We use BF16 mixed precision with DeepSpeed ZeRO-2.

Default loss weights. Following Eq. (12), $\beta = 5.0$, $\gamma = 1.0$, $\rho = 1.0$; the stealth loss uses $\lambda_1 = 10^{-4}$, $\lambda_2 = 10^{-4}$, $\lambda_p = 10^{-5}$. All weights use the same defaults across datasets, models, and target sizes, without per-task tuning. The L_1/L_2 terms are element-wise normalized by $C\times H\times W$; LPIPS is computed only on triggered samples.

Perturbation budget. The budget is $\epsilon = 8/255$ in raw pixel space $[0, 1]$, equivalent to $\epsilon_{\tanh} = 0.25$ after CLIP-style normalization; we apply $\tanh$ parameterization in the normalized space.

Target responses. Each $\mathbf{y}_m$ is a separate natural-language segment (malicious instruction, disinformation statement, or induced response), 5-15 tokens. The target sets at $M \in \{5, 10, 20, 50, 100\}$ are nested (the M=10 set contains the M=5 set, etc.), ensuring comparable target contents across scales.

Compute. All runs use 4 NVIDIA A40 GPUs. For LLaVA-v1.5-7B, one run with M=20, 100K samples, 10 epochs takes about 35 hours; a run with M=100 and 500K samples takes about 120 hours.

E Baseline Implementation and Hyperparameter Search

Of the nine baselines, six were originally single-target (BadNets, DualKey, TrojVLM, Blended, VL-Trojan, Shadowcast), two were category-level multi-target (M-to-N, MABA), and one was multi-target VLM-specific (MTAttack). We adapt each to our multi-target setting, where each target $\mathbf{y}_m$ has an independent trigger and all triggers use the same poisoning ratio $r_p = 0.75$ as HYDRA. Each method is tuned within its recommended range; we report the highest-Sem-ASR configuration within the unified budget.

BadNets. A fixed color patch in the lower-right corner; different targets use different patch colors/patterns. Patch size searched over $\{16, 32, 48\}$; 32×32 selected.

DualKey. A per-target patch optimized by gradient descent in CLIP representation space; patch 32×32, 500 steps, learning rate 0.01.

TrojVLM. BadNets plus semantic-preservation regularization; same patch config; regularization coefficient searched over $\{0.01, 0.1, 1.0\}$.

Blended. A predefined reference image blended into the input; each target uses a different reference (solid block or cartoon). Blending ratio searched over $\{0.1, 0.2, 0.3\}$; $\alpha = 0.2$ selected.

VL-Trojan. A per-target trigger image optimized by gradient descent and blended with $\alpha = 0.1$; 1000 steps, learning rate 0.005.

MABA. Elliptical triggers at CLIP attention-guided locations; ellipse position and axes chosen from the CLIP attention map; each target uses a different location.

M-to-N. A shared shallow four-layer convolutional network encodes each target as a grayscale trigger embedded into the image, following the original design.

Shadowcast. Adversarial perturbations via PGD for each target-concept pair; 50 steps, step size $1/255$.

MTAttack. A cross-target adversarial loss to mitigate multi-target interference; a patch trigger per target, jointly trained; loss weights follow the original paper.

All baselines share the same base VLM, LoRA configuration, poisoning ratio, minibatch size, epochs, optimizer, and learning rate as HYDRA, so comparisons reflect method design rather than training resources. We note that several baselines (notably MABA, Blended) show substantial benign-accuracy drops at $M \geq 10$; re-running MABA under $r_p \in \{0.50, 0.75, 0.90\}$ and learning rates $\{1, 2, 5\}\times10^{-5}$ shows the degradation persists across all configurations, consistent with the trigger-deactivation regime and cross-target representational competition diagnosed in Section 4.

F Defense Implementation Details

Gaussian Noise. Adds Gaussian noise with mean 0 and $\sigma = 0.03$ to each test image.

Feature Squeezing. Quantizes image colors from 8-bit to 7-bit (128 levels).

BDetCLIP. Uses CLIP-ViT-Large-Patch14-336 as the detection encoder; anomaly-score thresholds are calibrated on clean COCO validation images, and we report AUROC.

Clean Fine-Tuning. Continues fine-tuning the compromised system on 5% clean data (disjoint from attacker data) for 3 epochs at learning rate 2×10^{-5}, updating only the multimodal projector.

InverTune. Estimates candidate targets via universal adversarial perturbations, inverts the trigger mask and pattern, and repairs the model by activation tuning of the largest-offset layer in the visual encoder and the multimodal projector; the number of searched targets is fixed to 20.

Adversarial Fine-Tuning (AFT). Assumes the defender knows HYDRA uses sample-conditional triggers. The defender trains a UNet generator G'' with the same architecture (base channels 64) to produce adversarial perturbations with random targets on clean data ($\epsilon_{\text{adv}} = 0.25$ in the normalized space), then fine-tunes the compromised system on 5% clean data for 5 epochs with clean/adversarial ratio 0.5.

Conditional Trigger Inversion (CTI). Trains a conditional inversion network G'_k for each candidate target t_k by max-

Table 6: Per-target attack stability under different numbers of targets on VQA-v2. Hydra maintains high average ASR and strong worst-target performance as the number of targets scales.

#Targets	Method	Mean ASR ↑	Std ↓	Worst-20% ↑	Min ASR ↑	Failed-Target Ratio ↓
5	MTAttack	0.065	0.053	0.010	0.010	1.000
	MABA	0.846	0.161	0.575	0.575	0.000
	Hydra	**0.949**	0.095	**0.760**	**0.760**	**0.000**
10	MTAttack	0.036	0.017	0.010	0.010	1.000
	MABA	0.788	0.351	0.095	0.030	0.200
	Hydra	**0.993**	0.005	**0.990**	**0.990**	**0.000**
20	MTAttack	0.018	0.040	0.000	0.000	1.000
	MABA	0.299	0.376	0.000	0.000	0.700
	Hydra	**0.936**	0.220	**0.685**	0.000	**0.050**
50	MTAttack	0.021	0.089	0.000	0.000	0.940
	MABA	0.073	0.197	0.000	0.000	0.840
	Hydra	**0.989**	0.025	**0.945**	**0.900**	**0.000**
100	MTAttack	0.003	0.024	0.000	0.000	0.980
	MABA	0.006	0.041	0.000	0.000	0.960
	Hydra	**0.949**	0.154	**0.745**	0.000	**0.020**

imizing $P(t_k \mid \mathbf{x} + G'_k(\mathbf{x}, t_k), \mathbf{q})$, identifies the real target by whether the loss converges below a threshold of 2.0, and repairs based on the inverted pattern. Inversion uses 300 epochs, learning rate 5×10^{-4}, base channels 32; repair uses 3 epochs at 2×10^{-5}; the candidate list size is fixed to 20.

G Metric Definitions

Benign Accuracy (BA). Task accuracy of the compromised system on clean inputs, i.e. the fraction of clean samples whose response matches the ground-truth response above threshold τ.

Exact-ASR. Attack success under strict string matching: the model response must exactly match the target response $\mathbf{y}_m$.

Semantic-ASR (Sem-ASR). Attack success under semantic equivalence, allowing minor surface-form differences; determined by sentence-BERT cosine similarity with threshold $\tau = 0.85$.

Wrong-Target Rate (WTR). The probability that a trigger still affects the output but binds to a non-prespecified target (the average of Conf_i over targets).

Non-Target Failure (NTF). The fraction of triggered samples that neither activate the prespecified target nor misbind to another target, indicating complete trigger deactivation.

Per-Target ASR statistics. With $a_m = \mathrm{ASR}_m$: Mean ASR $= \frac{1}{M}\sum_m a_m$; Std is the standard deviation across targets; Worst-20% is the mean ASR of the lowest 20% of targets; Min ASR $= \min_m a_m$; and Failed-Target Ratio $= \frac{1}{M}\sum_m \mathbb{I}(a_m < 0.5)$.

PSNR. Peak signal-to-noise ratio between triggered and original images, averaged over the evaluation set.

H Per-Target Stability of HYDRA across Scales

Table 6 reports HYDRA's per-target ASR distribution across $M \in \{5, 10, 20, 50, 100\}$ on VQA-v2, complementing the scalability analysis in Section 6.3. The distribution stays sharply concentrated as M grows: even at M=100, the mean ASR remains 0.949 with only 2% failed targets, in contrast to the baselines whose failed-target ratio exceeds 0.94.

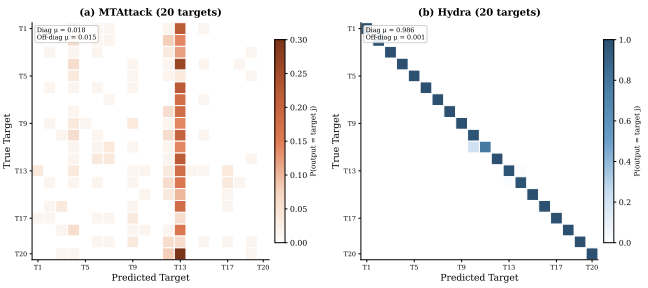

Figure 7: **Trigger-target binding at M=20 on VQA-v2.** Cell (i, j) is the probability that the trigger for target i elicits a response matching target j. (a) MTAttack collapses into an absorbing target (diagonal mean 0.018). (b) HYDRA produces a clean diagonal (mean 0.986, off-diagonal mean 0.001).

I Trigger-Target Binding Visualization

Figure 7 shows the trigger-target confusion matrix at M=20 on VQA-v2, where cell (i, j) is the empirical probability that the trigger of target i elicits a response matching target j. A perfectly separable attack yields a clean diagonal. MTAttack collapses into an absorbing-target pattern (diagonal mean 0.018, with a single column capturing almost all triggers), whereas HYDRA produces a near-perfect diagonal (mean 0.986, off-diagonal mean 0.001). This gives direct behavioural evidence that HYDRA's content-aware conditional triggers keep distinct target bindings separable at scale.