FEATURE-AWARE TOKEN ATTACK FOR COMPRESSION-TRIGGERED STEALTHY FAILURES IN LARGE VISION-LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Visual-token compression improves the efficiency of large vision-language models, but can expose failures that full-token evaluation misses. We study adversarial images that preserve full-token correctness yet induce errors after compression, even when both inference paths succeed on the clean image. Creating such failures is challenging because perturbing token importance can also damage the visual content needed for full-token inference. We propose **Feature-Aware Token Attack (FATA)**, which couples attention suppression with cosine-based feature preservation on a fixed set of salient clean-image tokens. In the primary LLaVA-1.5-7B setting, FATA uses only vision-encoder gradients, without access to the deployed compressor, token budget, or downstream task. Across four visually dependent task subsets and four compressors under a controlled reconstruction protocol, FATA achieves $\mathrm{SR} = \mathbf{96.3}\%$ full-token accuracy retention and $\mathrm{CBR} = \mathbf{22.1}\%$ conditional blinding, compared with 89.8% and 15.7% for CAA. Ablations support the role of both objectives in balancing compressed-path failure against full-token preservation. FATA also has the lowest measured detection rate among four attacks across three evaluated detectors at a 5% false-positive rate. These findings motivate assessing adversarial robustness jointly across full-token and compressed inference.

1 INTRODUCTION

Visual-token compression reduces the inference cost of large vision-language models (LVLMs) as image resolution and visual context grow (An et al., 2026; Song et al., 2026). Recent methods select informative tokens or merge redundant features while retaining much of the original task accuracy (Yang et al., 2025; Zhang et al., 2025b; Shang et al., 2025; Tong et al., 2026). Because these decisions determine which visual evidence remains available to the language model (Chen et al., 2024), their robustness to input perturbations is essential to reliable compressed inference.

We study a failure in which a perturbed image remains correctly understood with the full token sequence but yields an incorrect answer after compression (As shown in Figure 1). When both paths succeed on the clean image, this divergence exposes a vulnerability that full-token evaluation alone would miss. An attack targeting this vulnerability must therefore meet a dual requirement: *induce compressed-path failure while preserving full-token correctness*.

Prior compression-aware attacks approach this problem through feature disruption (CAGE) or token-importance manipulation (CAA) (Zhang et al., 2026b;a). Related attacks show that vision-encoder access alone can suffice to disrupt visual features (Wang et al., 2024; Mei et al., 2026). Selective failure poses an additional challenge: encoder attention offers a surrogate for token importance, but perturbing it can also damage the features needed for full-token inference. Conversely, preserving features alone provides little pressure to induce compressed-path failure. This coupling motivates our central question: ***how can an attack exploit compression sensitivity while preserving full-token behavior, using only vision-encoder access?***

In this paper, we propose **Feature-Aware Token Attack (FATA)**, which couples two objectives on a fixed set of tokens selected by clean encoder attention. An **attention-suppression objective** lowers attention to these positions as a surrogate for compression sensitivity, while a **feature-preservation objective** maintains cosine alignment with their clean feature vectors. Sharing the target set ties the

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

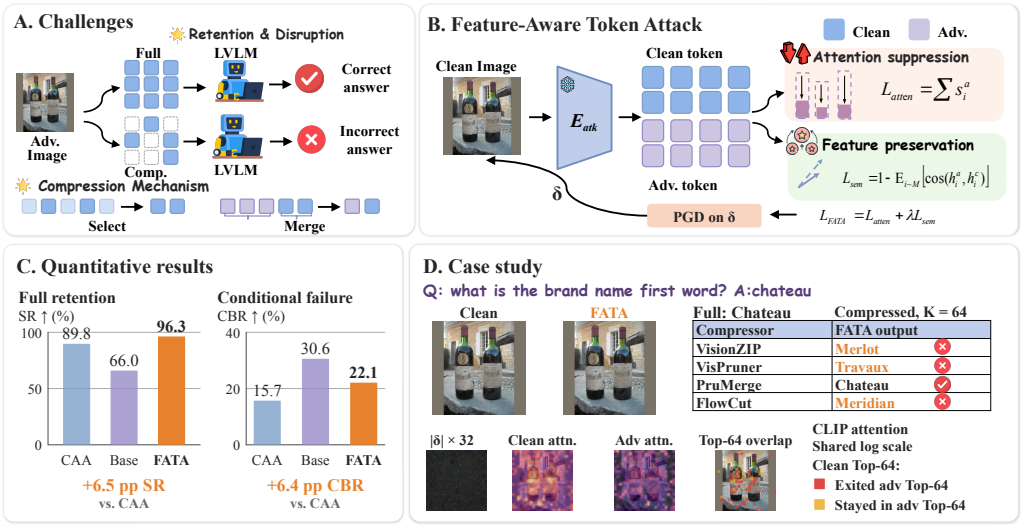


Figure 1: **FATA: attention suppression with feature preservation.** A illustrates the compression-triggered failure setting; B presents the two attack objectives; C compares full-token retention and conditional blinding; D shows how answers to the same perturbed TextVQA image vary across compression paths.

attack pressure to the content being protected. In the primary LLaVA-1.5-7B setting, FATA optimizes only the input perturbation through a frozen vision encoder, without access to the downstream language model, questions, answers, compressor, or token budget. The same perturbed image is then evaluated across compression rules and budgets without re-optimization.

Across four visually dependent task subsets and four compressors under a controlled reconstruction protocol, FATA achieves SR $= \mathbf{96.3}\%$ full-token accuracy retention and CBR $= \mathbf{22.1}\%$ conditional blinding rate, compared with 89.8% and 15.7% for CAA. Ablations show why both objectives matter: attention suppression alone substantially degrades full-token accuracy, while feature preservation alone yields weaker conditional blinding. These results support jointly controlling attention and feature distortion to expose compression sensitivity while preserving full-token behavior.

The contributions of this paper are summarized as follows.

- ❶ **Feature-aware attack design.** We introduce FATA, coupling attention suppression with explicit feature preservation on fixed clean-image targets to induce compression-triggered failures under vision-encoder-only access.
- ❷ **Paired robustness evaluation.** We evaluate the balance between full-token retention and conditional compressed-path failure across four task settings and four compressors, with loss ablations and model-specific extensions under stated access assumptions.
- ❸ **Detection across model stages.** We introduce a Multi-Level Activation Trajectory Detector (ML-ATD) to monitor the vision encoder, projector, and language model, and compare three detection approaches at a fixed low false-positive rate.

2 Related Work

Visual Token Compression Inputs at high resolution can produce long visual prefixes, increasing prefill latency and memory use (Chen et al., 2024; Arif et al., 2025). Visual token compression reduces these costs at the encoder output, visual projector, or decoder (Zhang et al., 2024; Li et al., 2025; Ye et al., 2025). Methods that require no training retain dominant, salient, or diverse tokens (Yang et al., 2025; Zhang et al., 2025b; Alvar et al., 2025). Other approaches merge nearby features or use early decoder attention and information flow across layers to remove redundancy (Shang et al., 2025; Chen et al., 2024; Tong et al., 2026). However, visual token compression can also increase vulnerability to adversarial perturbations that change which tokens are retained for inference (Zhang et al., 2026b;a).

Adversarial Perturbations under Compression Earlier methods disrupt encoder features, alignment between modalities, or transferable representations (Wang et al., 2024; Lu et al., 2023; Zhang et al., 2025a). Recent works use targets generated by diffusion models, encoder objectives independent of downstream tasks, or guidance from projector features (Guo et al., 2024; Mei et al., 2026; Cao et al., 2026). These approaches do not model which tokens a compressor retains. For visual token compression, CAGE targets tokens likely to survive unknown budgets (Zhang et al., 2026b). Though CAA and our work both focus on inducing failures after compression while preserving full token behavior (Zhang et al., 2026a), our method further incorporates an explicit semantic preservation loss to better achieve this goal.

Adversarial Detection Adversarial detectors distinguish adversarial inputs from clean inputs. Feature Squeezing assesses prediction consistency under input transformations, while CIDER measures changes in similarity between modalities under denoising (Xu et al., 2017; 2024). Internal-state detectors use feature distribution deviations or activations associated with safety (Lee et al., 2018; Jiang et al., 2025). JailDAM uses adaptive memory to update unsafe knowledge during inference and improve detection of unseen jailbreak strategies (Nian et al., 2025). Alongside input and feature distribution analysis, inspired by HiddenDetect, ML-ATD uses an attack reference direction estimated from Base examples and monitors activations in the visual encoder, projector, and language model.

3 METHOD

3.1 OVERVIEW

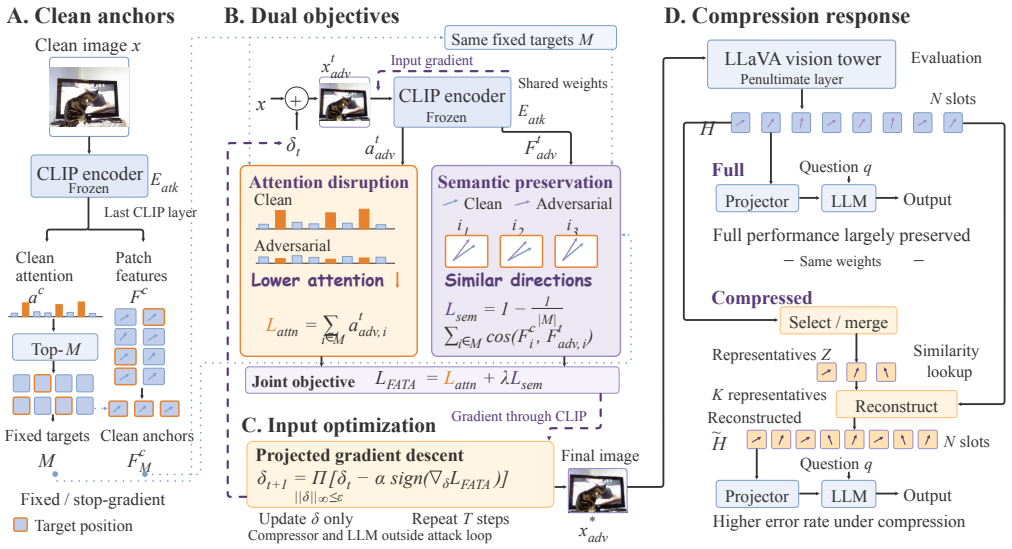


Figure 2: FATA overview. Fixed clean targets guide input-only PGD with attention and semantic losses (A–C), followed by full-token and compressed-path evaluation of the same image (D).

Figure 2 separates attack optimization (A–C) from compression-response evaluation (D). The frozen CLIP encoder uses last-layer clean CLS-to-patch attention to fix the Top-M target tokens and stores their patch features as stop-gradient anchors (A). For each adversarial iterate, $\mathcal{L}_{\text{attn}}$ suppresses attention over those same positions, while $\mathcal{L}_{\text{sem}}$ preserves their feature directions through cosine similarity to the clean anchors (B). PGD backpropagates their weighted sum through CLIP and updates only δ; the encoder weights, target set, and anchors stay fixed, and the compressor and LLM remain outside the attack loop (C).

Panel D evaluates the same optimized image along two paths. LLaVA's vision tower shares the encoder weights but reads penultimate-layer features H, distinct from the last-layer attack features. The full-token path passes H directly to the projector. The compressed path selects or merges K

representatives Z, then uses similarity lookup from H to reconstruct an N-slot sequence $\widetilde{H}$ from those representatives. Both paths use the same projector, LLM, and question. This comparison exposes compression-triggered errors despite largely preserved full-token performance. Reconstruction fills all N positions with representative features.

3.2 FATA

FATA is formulated as a gray-box method on LLaVA with gradient access to the visual encoder E, while the compressor, token budget, and downstream language model are unknown during optimization. Given a clean image x, FATA optimizes a perturbation δ under $\|\delta\|_\infty \le \epsilon$ to obtain $x_{\text{adv}} = \text{clip}(x + \delta, 0, 1)$. Let $(\mathbf{s}^c, \mathbf{H}^c) = E(x)$ denote the last-layer CLS attention scores over visual tokens and their corresponding features. Attention scores are averaged across heads, and the top M visual tokens form a fixed target set.

$$\mathcal{M} = \text{TopK}(\mathbf{s}^c, M), \qquad M = 64. \tag{1}$$

Algorithm 1 FATA optimization overview

Require: Clean image x, encoder E, ϵ, α, T, λ, target count M

$(\mathbf{s}^c, \mathbf{H}^c) \leftarrow E(x)$
$\mathcal{M} \leftarrow \text{TopK}(\mathbf{s}^c, M)$
$\delta^{(0)} \sim \mathcal{U}(-\epsilon, \epsilon)$
for $t = 0, \dots, T-1$ **do**
  $x_a^t \leftarrow \text{clip}(x + \delta^{(t)}, 0, 1)$
  $(\mathbf{s}^a, \mathbf{H}^a) \leftarrow E(x_a^t)$
  $\mathcal{L}_{\text{attn}} \leftarrow \sum_{i \in \mathcal{M}} s_i^a$
  $\mathcal{L}_{\text{sem}} \leftarrow 1 - \frac{1}{|\mathcal{M}|} \sum_{i \in \mathcal{M}} \cos(\mathbf{h}_i^a, \mathbf{h}_i^c)$
  $\mathcal{L}_{\text{FATA}} \leftarrow \mathcal{L}_{\text{attn}} + \lambda \mathcal{L}_{\text{sem}}$
  $\delta^{(t+1)} \leftarrow \Pi_{\|\delta\|_\infty \le \epsilon}(\delta^{(t)} - \alpha \, \text{sign}(\nabla_\delta \mathcal{L}_{\text{FATA}}))$
end for
return $\text{clip}(x + \delta^{(T)}, 0, 1)$

The visual tokens in $\mathcal{M}$ are referred to as target tokens. For the adversarial image x_{adv}, FATA minimizes the following attention loss over the target tokens.

$$\mathcal{L}_{\text{attn}} = \sum_{i \in \mathcal{M}} s_i^a, \tag{2}$$

where $\mathbf{s}^a$ denotes the adversarial attention scores. Minimizing $\mathcal{L}_{\text{attn}}$ suppresses attention to the target tokens, with the aim of reducing their selection priority during visual token compression.

To preserve full token behavior, FATA aligns the feature directions of the adversarial target tokens with their clean features. Let $\mathbf{h}_i^c$ and $\mathbf{h}_i^a$ denote the clean and adversarial features of target token i. The semantic loss is

$$\mathcal{L}_{\text{sem}} = 1 - \frac{1}{|\mathcal{M}|} \sum_{i \in \mathcal{M}} \cos(\mathbf{h}_i^a, \mathbf{h}_i^c). \tag{3}$$

Both losses use the same fixed target set. The final objective is

$$\mathcal{L}_{\text{FATA}} = \mathcal{L}_{\text{attn}} + \lambda \mathcal{L}_{\text{sem}}. \tag{4}$$

Algorithm 1 applies ℓ_∞ PGD to $\mathcal{L}_{\text{FATA}}$ with $\delta^{(0)} \sim \mathcal{U}(-\epsilon, \epsilon)$, $\lambda = 1$, $\epsilon = 2/255$, $\alpha = 0.5/255$, and $T = 100$ steps. Here, $\Pi_{\|\delta\|_\infty \le \epsilon}$ projects the perturbation onto the ℓ_∞ ball of radius ϵ. Optimization details and sensitivity to α and λ are provided in Appendix A.

3.3 Evaluation Metrics

Full token behavior and compression induced failure are evaluated with SR, CER, and CBR. Let $\text{Acc}_{m@K}$ denote the task accuracy of method m at token budget K, and let $\text{Acc}_{m@\text{full}}$ denote its accuracy under full token inference. SR measures full token performance retention and is then defined as

$$\text{SR} = \frac{\text{Acc}_{\text{FATA@full}}}{\text{Acc}_{\text{Clean@full}}}. \tag{5}$$

For LLaVA, the practical token budget K_{prac} is selected separately for each dataset–compressor pair using clean accuracy and is fixed across attacks. Qwen3.5 and InternVL3.5 use predefined task-specific retention fractions, as detailed in Appendix B.

$$K_{\text{prac}} = \min \left\{ K : \frac{\text{Acc}_{\text{Clean@}K}}{\text{Acc}_{\text{Clean@full}}} \ge 0.8 \right\}. \tag{6}$$

For attack m, the compressed error rate (CER) measures overall task error at K_{prac}:

$$\text{CER}_m = 1 - \text{Acc}_{m@K_{\text{prac}}}. \tag{7}$$

It uses task-credit accuracy and does not subtract the error on Clean inputs. CBR measures the fraction of examples on which attack m becomes incorrect at the practical budget, conditioned on both Clean and attack m being correct at full tokens and Clean remaining correct under the same compressor and practical budget. Binary correctness follows the task-specific rules in Appendix C. For FATA, $m = \text{FATA}$,

$$\text{CBR} = \frac{\left|\left\{x : \begin{array}{l}\text{Clean@full correct} \ \wedge \ m\text{@full correct} \wedge \\ \text{Clean@}K_{\text{prac}} \text{ correct} \ \wedge \ m\text{@}K_{\text{prac}} \text{ incorrect}\end{array}\right\}\right|}{\left|\left\{x : \begin{array}{l}\text{Clean@full correct} \ \wedge \ m\text{@full correct} \wedge \\ \text{Clean@}K_{\text{prac}} \text{ correct}\end{array}\right\}\right|}. \tag{8}$$

Scoring rules and metric aggregation across the three models are detailed in Appendix C.

4 Experiments

4.1 Experimental Setup

The main evaluation uses LLaVA-1.5-7B (Liu et al., 2024) with four visual token compressors, VisionZIP, VisPruner, PruMerge, and FlowCut (Yang et al., 2025; Zhang et al., 2025b; Shang et al., 2025; Tong et al., 2026). The evaluation covers four 1,000-image subsets, TextVQA-Open, VQAv2-Open, ScienceQA-MC, and VQAv2-MC (Singh et al., 2019; Goyal et al., 2017; Lu et al., 2022). Before inference, one representative question is selected for each unique image. The subset is then constructed by retaining examples that are answered correctly with the original image but incorrectly when the image is replaced by a black image. Further details of subset construction are provided in Appendix D. The retained token budgets are $K \in \{576, 192, 128, 64, 32, 16\}$, and FATA is compared with Base(attention-only), CAA (Zhang et al., 2026a), and CAGE (Zhang et al., 2026b) under the common perturbation bound $\epsilon = 2/255$.

For each dataset and compressor, clean and adversarial inputs are evaluated on the same valid examples at every retained token budget using the metrics defined in Section 3.3. K_{prac} is selected from clean results only and then fixed for attack evaluation, as defined in Eq. 6.

4.2 LLaVA-1.5-7B Results

4.2.1 Main Results

Table 1 reports results on LLaVA-1.5-7B (Liu et al., 2024) and shows that FATA preserves most full token performance while causing larger degradation under visual token compression.

At $K = 576$, FATA achieves an average accuracy of 89.7%, close to the 93.2% Clean average, while Base and CAGE fall to 61.6% and 34.4%, respectively. This result indicates that FATA preserves full token behavior before compression is applied.

At matched compression budgets, the accuracy difference between Clean and FATA increases from 3.5 points at $K = 576$ to 5.5, 7.4, 9.3, 12.5, and 15.6 points as the retained token budget decreases to 192, 128, 64, 32, and 16, respectively. At $K = 16$, TextVQA-Open exhibits the largest dataset level difference, at 23.8 points. The widening gap shows that FATA induces larger accuracy degradation after visual token compression.

Figure 3 shows when failures occur across attacks using the same examples answered correctly on clean inputs at full tokens. Among the evaluated methods, FATA has the fewest failures before compression (5.3%) and the most emerging after compression (23.3%), while 71.5% of cases remain correct after compression. CAA also limits failures before compression (8.8%) but has fewer failures emerging after compression (18.7%). Compared with FATA, Base and CAGE have more failures before compression, at

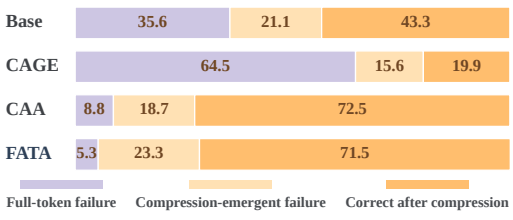


Figure 3: Failure-stage decomposition over 14,316 shared Clean-full-correct cases.

Table 1: Accuracy (%) across four benchmarks and visual-token budgets.

Method	TextVQA-Open					VQAv2-Open					ScienceQA-MC					VQAv2-MC				
	Clean	Base	CAA	CAGE	**FATA**	Clean	Base	CAA	CAGE	**FATA**	Clean	Base	CAA	CAGE	**FATA**	Clean	Base	CAA	CAGE	**FATA**
Upper Bound (576 Tokens)																				
None	92.0	39.9	77.4	19.8	86.6	88.2	66.1	78.9	28.3	86.0	94.5	57.8	89.2	40.6	90.1	98.0	82.5	89.6	48.7	96.2
Retain 192 Tokens ($K_{model} = 192$)																				
VisionZIP	89.1	35.7	82.5	16.2	83.3	86.6	61.8	56.2	25.8	84.2	91.3	54.5	86.3	38.2	86.3	96.9	79.1	94.8	45.8	95.1
VisPruner	90.6	36.1	83.9	17.9	83.3	87.4	65.1	84.9	26.6	84.3	91.1	56.5	87.1	40.2	88.7	96.8	79.8	95.1	45.0	95.0
FlowCut	88.1	38.0	81.9	18.5	82.9	86.2	63.2	84.6	24.1	83.3	87.7	56.5	86.1	39.3	87.0	96.0	77.1	94.6	45.5	94.1
PruMerge	84.5	29.8	77.7	16.3	60.4	84.1	58.5	83.3	25.5	79.1	89.4	50.3	86.9	38.1	74.9	94.8	75.3	65.3	42.8	90.7
Average	88.1	34.9	81.5	17.2	**77.5**	86.1	62.2	77.3	25.5	**82.7**	89.9	54.5	86.6	39.0	**84.2**	96.1	77.8	87.5	44.8	**93.7**
Retain 128 Tokens ($K_{model} = 128$)																				
VisionZIP	86.7	32.3	81.6	13.5	80.2	86.1	58.5	58.4	22.6	82.7	88.9	53.5	85.5	37.1	85.3	95.5	76.3	93.3	43.0	92.9
VisPruner	88.2	34.8	81.8	15.9	81.3	86.0	62.3	84.2	23.4	82.9	89.4	56.3	85.7	39.3	87.1	95.5	76.3	94.7	42.9	93.7
FlowCut	84.9	32.8	79.4	15.5	78.4	84.0	58.5	82.3	19.6	80.7	85.6	56.6	83.6	39.5	84.6	94.0	72.6	92.7	39.7	90.5
PruMerge	81.8	23.9	76.6	12.9	46.0	80.6	54.4	79.9	23.5	69.8	88.5	47.6	85.9	36.2	71.5	92.6	71.2	67.6	41.1	83.0
Average	85.4	30.9	79.9	14.5	**71.5**	84.2	58.4	76.2	22.3	**79.0**	88.1	53.5	85.2	38.0	**82.1**	94.4	74.1	87.1	41.7	**90.0**
Retain 64 Tokens ($K_{model} = 64$)																				
VisionZIP	81.2	25.8	75.7	11.4	70.9	80.3	51.4	58.8	20.1	77.3	85.3	52.0	82.8	35.6	83.1	90.7	68.4	87.7	37.0	86.3
VisPruner	82.6	29.7	75.6	13.1	74.6	80.3	55.8	78.6	20.9	75.3	85.9	53.4	83.8	39.7	84.1	92.2	71.4	89.8	35.6	88.7
FlowCut	76.8	24.6	72.4	11.2	66.8	74.1	47.7	72.6	14.5	69.7	82.4	51.8	80.6	37.1	76.7	85.5	60.8	85.9	31.0	79.8
PruMerge	77.7	20.0	73.6	10.4	38.2	77.8	47.7	75.8	20.5	62.5	84.8	46.4	85.5	34.7	66.4	90.6	65.7	66.4	38.0	78.5
Average	79.6	25.0	74.3	11.5	**62.6**	78.1	50.6	71.4	19.0	**71.2**	84.6	50.9	83.2	36.8	**77.6**	89.8	66.6	82.4	35.4	**83.3**
Retain 32 Tokens ($K_{model} = 32$)																				
VisionZIP	69.0	17.0	64.8	8.8	54.7	67.6	37.4	51.5	15.1	61.2	81.6	47.2	79.9	32.2	75.2	82.3	54.9	79.8	32.3	74.9
VisPruner	72.7	22.9	68.2	10.0	60.1	70.5	44.0	68.3	16.7	64.6	79.3	51.0	75.6	39.8	74.1	84.7	58.4	83.2	32.4	79.0
FlowCut	63.0	16.7	59.3	7.9	46.9	58.7	32.9	57.6	10.5	49.8	79.8	47.6	78.8	35.2	72.6	73.9	48.5	71.5	26.5	63.2
PruMerge	66.3	14.4	61.4	6.6	26.7	66.1	37.6	63.5	16.0	51.1	79.7	44.5	76.6	33.7	55.9	81.7	56.5	62.7	32.0	67.1
Average	67.8	17.8	63.4	8.3	**47.1**	65.7	38.0	60.2	14.6	**56.7**	80.1	47.6	77.7	35.2	**69.5**	80.7	54.6	74.3	30.8	**71.1**
Retain 16 Tokens ($K_{model} = 16$)																				
VisionZIP	53.3	12.1	49.2	6.1	30.6	52.5	26.4	42.8	12.4	38.6	69.9	39.3	64.9	32.4	54.2	70.2	41.6	67.2	26.4	55.4
VisPruner	54.2	14.4	49.2	8.2	31.0	52.4	29.9	52.8	12.4	41.8	66.7	46.2	67.2	34.9	61.5	70.9	51.0	67.8	29.2	61.7
FlowCut	44.2	10.8	41.1	5.5	23.8	39.7	18.7	38.8	10.4	25.5	64.0	40.4	56.4	33.9	47.9	55.8	33.5	54.5	26.1	40.3
PruMerge	45.9	10.1	44.5	6.2	17.1	49.3	27.6	51.5	13.3	37.0	62.5	42.3	58.0	33.4	48.6	69.3	46.4	52.6	28.7	55.6
Average	49.4	11.9	46.0	6.5	**25.6**	48.5	25.7	46.5	12.1	**35.7**	65.8	42.1	61.6	33.7	**53.1**	66.6	43.1	60.5	27.6	**53.2**

35.6% and 64.5%, respectively. These percentages share the Clean-full-correct denominator. CBR conditions on Clean full-token, attacked full-token, and Clean compressed correctness, then measures the attack error rate under the same compressor and practical budget.

Table 2 reports SR, CER, and CBR to characterize the balance between full token retention and failure after compression at K_{prac}. CER is computed as $100\% - \text{Acc@}K_{\text{prac}}$ from the displayed accuracies in Table 1 and budgets in Table 11, then averaged over the 16 settings and rounded to two decimal places.

Table 2: Overall stealth–lethality trade-off

Attack	SR ↑	CER ↑	CBR ↑
Base (Attn.-only)	66.7	54.78	32.9
CAA	89.8	24.65	15.7
CAGE	36.6	**75.58**	**43.1**
FATA	**96.3**	30.38	22.1

Compared with Base, FATA raises SR from 66.7% to 96.3% while reducing CER from 54.78% to 30.38% and CBR from 32.9% to 22.1%, balancing full token behavior and compression induced failure. CAGE reaches higher CER and CBR, 75.58% and 43.1%, respectively, but its SR drops to 36.6%, while CAA keeps 89.8% SR but produces a lower CBR of 15.7%. These results indicate that Base and CAGE exhibit higher CER and CBR alongside greater full-token degradation, while FATA preserves full-token performance more effectively. Compared with CAA, FATA increases SR from 89.8% to 96.3% and CBR from 15.7% to 22.1%, offering a better trade-off. Appendix E provides qualitative examples of changes in CLIP attention rankings and compression-dependent answers.

4.2.2 Ablation Study

Table 3 separates the roles of the two losses in Eq. 4. The attention loss $\mathcal{L}_{\text{attn}}$ alters the visual evidence that survives compression, whereas the semantic loss $\mathcal{L}_{\text{sem}}$ preserves the full token representation. Applying $\mathcal{L}_{\text{attn}}$ alone gives the highest CBR of 32.9% but reduces SR to 66.7%, showing that attention disruption strengthens compression induced failure but damages full token behavior without semantic preservation. Using $\mathcal{L}_{\text{sem}}$ alone gives the opposite effect, reaching 98.5% SR but only 14.5%

Table 3: Dual-objective ablation of FATA.

Variant	Objective	SR ↑	CBR ↑
Attention-only	$\mathcal{L}_{\text{attn}}$	66.7	**32.9**
Semantic-only	$\mathcal{L}_{\text{sem}}$	**98.5**	14.5
FATA Full	$\mathcal{L}_{\text{attn}} + \lambda\mathcal{L}_{\text{sem}}$	96.3	22.1

CBR, which indicates that semantic preservation protects full token inference but leads to fewer compression-induced failures. The full FATA objective maintains 96.3% SR and reaches 22.1% CBR, combining the full token behavior supported by $\mathcal{L}_{\text{sem}}$ with the compression sensitivity introduced by $\mathcal{L}_{\text{attn}}$. Appendix F shows the corresponding variation across the 16 experimental settings.

4.3 Cross-Model Evaluation

4.3.1 Qwen3.5-9B

Since the number of visual tokens in Qwen3.5(Qwen Team, 2026) depends on input resolution, compression budgets are specified as fractions of visual tokens. The evaluation uses a $1/3$ token budget, while the practical token budget is set to $1/9$ for open tasks and $1/18$ for multiple choice tasks. Model-specific adaptations are detailed in Appendix G.1.

Table 4: Qwen3.5-9B macro accuracy (%) under full, $1/3$, and practical token budgets. Values are averaged over four datasets and the four compressors.

Visual-token budget	Clean	FATA	Damage (pp)	Amplification (pp)
Full	89.55	88.92	0.62	–
Retain $1/3$	80.66	72.47	8.19	7.56
Practical (Open: $1/9$; MC: $1/18$)	65.10	48.48	**16.62**	**15.99**

Figure 4 shows the corresponding CBR values for each dataset and compressor. At the practical token budget, FATA accuracy decreases by 40.44 percentage points, from 88.92% to 48.48%, while Clean accuracy decreases by 24.45 points, from 89.55% to 65.10%, as reported in Table 4. The larger decrease for FATA shows that compression leads to greater accuracy loss with FATA on Qwen3.5.

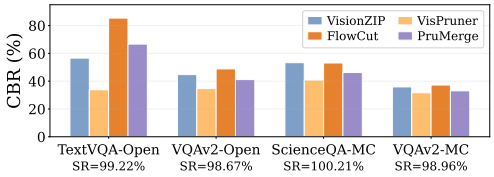


Figure 4: Qwen3.5 practical-budget CBR by compressor and dataset.

Table 5 reports CBR and SR on Qwen3.5-9B. SR remains high at an average of 99.3% across the four datasets, while CBR averages 46.43% across the 16 dataset and compressor combinations. These results support the same pattern observed on LLaVA, with FATA retaining full token behavior while exhibiting failures after compression.

Table 5: Qwen3.5-9B full-token SR and practical-budget conditional CBR (%) across the four compressors.

Compression Method	TextVQA-Open SR=99.22%	VQAv2-Open SR=98.67%	ScienceQA-MC SR=100.21%	VQAv2-MC SR=98.96%
CBR at Practical Budget (Open: $1/9$; MC: $1/18$)				
VisionZIP	56.58	44.70	53.28	35.85
VisPruner	33.79	34.66	40.79	31.66
FlowCut	85.37	48.83	53.07	37.21
PruMerge	66.67	41.13	46.20	33.02
Average	**60.60**	**42.33**	**48.34**	**34.43**

4.3.2 InternVL3.5-8B-HF

To accommodate the dynamic tiling and token organization of InternVL3.5-8B-HF (Wang et al., 2025), *TA-FATA* adapts the FATA loss while retaining attention disruption and semantic preservation. The complete objective, fixed hyperparameters, and evaluation details are given in Appendix G.2.

Table 6 reports the accuracy of TA-FATA across token budgets. At full tokens, the four-task average accuracy is 90.52% for Clean and 85.28% for TA-FATA, while the corresponding values at the $1/3$ budget are 85.96% and 75.59%. At the practical budget, TA-FATA accuracy decreases by 32.81 percentage points from full tokens to 52.47%, while Clean accuracy decreases by 24.08 points to 66.44%. As on LLaVA and Qwen3.5, visual token compression leads to greater accuracy loss with TA-FATA than with Clean inputs.

Table 6: InternVL3.5-8B-HF accuracy (%) under full, 1/3, and practical token budgets. Compressed rows average the four retained compressors; the practical budget is 1/9 for open-ended tasks and 1/18 for multiple-choice tasks.

Compression Method	InternVL3.5-8B-HF Accuracy (%)							
	TextVQA-Open		VQAv2-Open		ScienceQA-MC		VQAv2-MC	
	Clean	FATA	Clean	FATA	Clean	FATA	Clean	FATA
Full Visual Tokens								
No Compression	87.6	81.2	79.1	70.0	98.7	96.6	96.7	93.3
Retain 1/3 of Visual Tokens								
VisionZIP	81.6	64.6	77.8	**66.7**	97.1	**91.9**	96.0	**91.2**
VisPruner	66.0	44.0	75.9	62.2	95.2	88.6	94.5	89.6
FlowCut	78.4	62.0	75.8	61.9	95.0	88.8	94.5	88.5
PruMerge	82.3	**65.3**	76.9	66.1	93.5	88.1	94.8	90.0
Average	**77.1**	**59.0**	**76.6**	**64.2**	**95.2**	**89.4**	**95.0**	**89.8**
Practical Budget (Open: 1/9; MC: 1/18)								
VisionZIP	51.8	31.9	71.1	53.1	68.1	57.4	84.5	72.8
VisPruner	41.6	22.8	66.2	48.2	67.9	57.5	84.6	72.7
FlowCut	49.2	26.8	66.5	47.6	64.5	54.8	80.0	69.4
PruMerge	47.3	**33.4**	69.9	**56.9**	67.9	**60.7**	81.9	**73.5**
Average	**47.5**	**28.7**	**68.4**	**51.5**	**67.1**	**57.6**	**82.8**	**72.1**

Table 7: InternVL3.5-8B-HF full-token SR and practical-budget conditional CBR (%) across four compressors.

Compression Method	TextVQA-Open SR=93.29%	VQAv2-Open SR=88.11%	ScienceQA-MC SR=97.87%	VQAv2-MC SR=96.48%
CBR at Practical Budget (Open: 1/9; MC: 1/18)				
VisionZIP	62.18	28.00	41.39	23.00
VisPruner	72.48	36.07	41.18	23.54
FlowCut	67.64	36.61	43.88	26.57
PruMerge	60.12	23.96	37.55	21.92
Average	**65.61**	**31.16**	**41.00**	**23.76**

Table 7 reports exact-match SR and conditional CBR on InternVL3.5-8B-HF. Macro SR is 93.94%, compared with 96.3% on LLaVA and 99.3% on Qwen3.5. Macro CBR over the 16 dataset and compressor combinations is 40.38%, higher than LLaVA's 22.1% but lower than Qwen3.5's 46.43%. InternVL3.5 therefore incurs a greater cost to full-token preservation while exhibiting a conditional blinding rate between the other two models. Its exact-match SR uses binary correctness, while Table 6 reports canonical task-credit accuracy, so the latter cannot directly reproduce SR.

Across LLaVA, Qwen3.5, and InternVL3.5, FATA and its adaptations retain most full token performance while exhibiting greater accuracy loss under compression than Clean.

5 Detection Analysis

Beyond the performance of FATA in full token retention and failure after compression, this section evaluates the detectability of FATA inputs using input and internal-state detectors.

Table 8 reports the area under the receiver operating characteristic curve (AUROC), the area under the precision recall curve (AUPR), and true positive rate at a 5% false positive rate (TPR@5FPR) for FATA, CAA, CAGE, and Base using Feature Squeezing (Xu et al., 2017), Mahalanobis-Max (Lee et al., 2018), and a Multi-Level Activation Trajectory Detector (ML-ATD) inspired by HiddenDetect (Jiang et al., 2025). Among the three detectors, Feature Squeezing compares model predictions before and after input transformations by measuring changes in their scores against ground-truth answers. Mahalanobis-Max measures feature space deviation from normal samples; ML-ATD tracks changes in activations across the visual encoder, projector, and LLM. Additional detector construction and evaluation details are provided in Appendix H.

Table 8: Cross-attack detection performance with clean and random negatives. Higher values indicate easier detection; protocol differences and limitations are detailed in Appendix H.

	Feature Squeezing			Mahalanobis-Max			ML-ATD		
Attack	AUROC	AUPR	TPR@5FPR	AUROC	AUPR	TPR@5FPR	AUROC	AUPR	TPR@5FPR
FATA	0.560	0.382	0.024	0.695	0.509	0.134	0.828	0.724	0.497
CAGE	0.797	0.655	0.147	0.936	0.882	0.647	0.913	0.854	0.694
CAA	0.564	0.385	0.027	0.833	0.765	0.468	0.802	0.721	0.533
Base	0.665	0.494	0.064	0.813	0.660	0.278	0.952	0.918	0.801

FATA has the lowest TPR@5FPR among the four attacks for each detector. Feature Squeezing gives similarly low recall for FATA and CAA (2.4% and 2.7%), whereas Mahalanobis-Max separates them more clearly (13.4% and 46.8%). Under ML-ATD, FATA reaches 49.7%, compared with 53.3% for CAA, 69.4% for CAGE, and 80.1% for Base; CAA has slightly lower AUROC and AUPR. Table 17 in Appendix H reports the relative reductions from Base to FATA.

6 Conclusion

FATA combines attention disruption and semantic preservation to shift damage from full-token to compressed inference. It retains 96.3% of LLaVA full-token performance with 22.1% practical-budget conditional blinding, and model-adapted Qwen3.5 and InternVL3.5 tests show that the trigger extends beyond LLaVA. These results call for evaluating token compression by adversarial behavior as well as clean efficiency and accuracy.

Limitations. FATA assumes visual-encoder gradients. Its strongest Qwen3.5 and InternVL3.5 results use task- or architecture-aware extensions, with adaptation-dependent full-token cost. Stealth is task-based, and ML-ATD is not an optimal defense. Black-box transfer, broader tasks, and stronger online detectors remain open.

AI Use Statement

Generative AI tools, including ChatGPT/Codex and Claude-based coding assistants, were used to assist with implementation and debugging, dataset-processing workflows, feedback on experimental procedures, and interpretation of quantitative and qualitative results. They also supported literature search, Chinese–English translation, manuscript drafting and editing, and preparation of figures, tables, and references. AI-assisted revisions were checked against the manuscript's equations and tables, dataset-selection records, and source publications. The authors retain responsibility for the experimental evidence, methodological choices, and final text, including all AI-assisted content.

Ethics Statement

This work examines a security risk in efficient vision-language inference. Attacks that preserve full-token behavior could be misused to evade screening and disrupt systems after compression is enabled. Our evaluation uses existing image-question benchmarks and model checkpoints to characterize this risk, and includes detection experiments to inform mitigation.Deployment safety and robustness across populations and applications require further evaluation. Reuse of benchmark images and model checkpoints remains subject to their original licenses and privacy obligations; the attack methods are intended for controlled, authorized robustness evaluation.

Reproducibility Statement

Algorithm 1 and Appendix A specify the FATA optimization procedure. Appendix B documents practical-budget selection and the model-specific retention ratios. Appendix C defines the evaluation metrics. Appendix D records the benchmark configuration and attack settings, with the unique-image

selection, deduplication, and scoring protocol in Appendix D.1. The Qwen3.5 and InternVL3.5 adaptations are given in Appendix G, and detector fitting, scoring, and evaluation protocols are described in Appendix H.

REFERENCES

Saeed Ranjbar Alvar, Gursimran Singh, Mohammad Akbari, and Yong Zhang. Divprune: Diversity-based visual token pruning for large multimodal models. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9392–9401. IEEE, 2025.

Xiang An, Yin Xie, Feilong Tang, Yunyao Yan, Huajie Tan, Didi Zhu, Changrui Chen, Xiuwei Zhao, Bin Qin, Kaicheng Yang, Yifei Shen, Yuanhan Zhang, Kaichen Zhang, Wenkang Zhang, Zheng Cheng, Nansen Zhang, Chunsheng Wu, Chunjiang Ge, Zimin Ran, Dehua Song, Chunyuan Li, Shikun Feng, Ming Hu, Zhangquan Chen, Junbo Niu, Bo Li, Ziyong Feng, Ziwei Liu, Zongyuan Ge, and Jiankang Deng. LLaVA-OneVision-2: Towards next-generation perceptual intelligence. *arXiv preprint arXiv:2605.25979*, 2026. doi: 10.48550/arXiv.2605.25979. URL https://arxiv.org/abs/2605.25979.

Kazi Hasan Ibn Arif, JinYi Yoon, Dimitrios S Nikolopoulos, Hans Vandierendonck, Deepu John, and Bo Ji. Hired: Attention-guided token dropping for efficient inference of high-resolution vision-language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39(2), pp. 1773–1781, 2025.

Yiming Cao, Yanjie Li, Kaisheng Liang, and Bin Xiao. Enhancing targeted adversarial attacks on large vision-language models via intermediate projector. *IEEE Transactions on Information Forensics and Security*, 2026.

Liang Chen, Haozhe Zhao, Tianyu Liu, Shuai Bai, Junyang Lin, Chang Zhou, and Baobao Chang. An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models. In *European Conference on Computer Vision*, pp. 19–35. Springer, 2024.

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 (CVPR)*, pp. 6904–6913, 2017.

Qi Guo, Shanmin Pang, Xiaojun Jia, Yang Liu, and Qing Guo. Efficient generation of targeted and transferable adversarial examples for vision-language models via diffusion models. *IEEE Transactions on Information Forensics and Security*, 20:1333–1348, 2024.

Yilei Jiang, Xinyan Gao, Tianshuo Peng, Yingshui Tan, Xiaoyong Zhu, Bo Zheng, and Xiangyu Yue. Hiddendetect: Detecting jailbreak attacks against multimodal large language models via monitoring hidden states. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14880–14893, 2025.

Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. *Advances in neural information processing systems*, 31, 2018.

Wentong Li, Yuqian Yuan, Jian Liu, Dongqi Tang, Song Wang, Jie Qin, Jianke Zhu, and Lei Zhang. Tokenpacker: Efficient visual projector for multimodal llm. *International Journal of Computer Vision*, 133(10):6794–6812, 2025.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 26286–26296. IEEE, 2024.

Dong Lu, Zhiqiang Wang, Teng Wang, Weili Guan, Hongchang Gao, and Feng Zheng. Set-level guidance attack: Boosting adversarial transferability of vision-language pre-training models. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 102–111. IEEE, 2023.

Pan Lu, Swaroop Mishra, Tanglin Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, 2022.

Hefei Mei, Zirui Wang, Shen You, Minjing Dong, and Chang Xu. Veattack: Downstream-agnostic vision encoder attack against large vision language models. In *International Conference on Learning Representations*, volume 2026, pp. 18135–18161, 2026.

Yi Nian, Shenzhe Zhu, Yuehan Qin, Li Li, Ziyi Wang, Chaowei Xiao, and Yue Zhao. Jaildam: Jailbreak detection with adaptive memory for vision-language model. *arXiv preprint arXiv:2504.03770*, 2025.

Qwen Team. Qwen3.5-9b. Hugging Face model card, 2026. Qwen/Qwen3.5-9B.

Yuzhang Shang, Mu Cai, Bingxin Xu, Yong Jae Lee, and Yan Yan. Llava-prumerge: Adaptive token reduction for efficient large multimodal models. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 22857–22867. IEEE, 2025.

Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards vqa models that can read. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8317–8326, 2019.

Jiafei Song, Fengwei Zhou, Jin Qu, Wenjin Jason Li, Tong Wu, Gengjian Xue, Zhikang Zhao, Daomin Wei, Yichao Lu, and Bailin Na. EvoComp: Learning visual token compression for multimodal large language models via semantic-guided evolutionary labeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026. URL https://arxiv.org/abs/2604.17087.

Jintao Tong, Wenwei Jin, Pengda Qin, Anqi Li, Yixiong Zou, Yuhong Li, Yuhua Li, and Ruixuan Li. Flowcut: Rethinking redundancy via information flow for efficient vision-language models. *Advances in Neural Information Processing Systems*, 38:94946–94973, 2026.

Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, Zhaokai Wang, Zhe Chen, Hongjie Zhang, Ganlin Yang, Haomin Wang, Qi Wei, Jinhui Yin, Wenhao Li, Erfei Cui, Guanzhou Chen, Zichen Ding, Changyao Tian, Zhenyu Wu, Jingjing Xie, Zehao Li, Bowen Yang, Yuchen Duan, Xuehui Wang, Zhi Hou, Haoran Hao, Tianyi Zhang, Songze Li, Xiangyu Zhao, Haodong Duan, Nianchen Deng, Bin Fu, Yinan He, Yi Wang, Conghui He, Botian Shi, Junjun He, Yingtong Xiong, Han Lv, Lijun Wu, Wenqi Shao, Kaipeng Zhang, Huipeng Deng, Biqing Qi, Jiaye Ge, Qipeng Guo, Wenwei Zhang, Songyang Zhang, Maosong Cao, Junyao Lin, Kexian Tang, Jianfei Gao, Haian Huang, Yuzhe Gu, Chengqi Lyu, Huanze Tang, Rui Wang, Haijun Lv, Wanli Ouyang, Limin Wang, Min Dou, Xizhou Zhu, Tong Lu, Dahua Lin, Jifeng Dai, Weijie Su, Bowen Zhou, Kai Chen, Yu Qiao, Wenhai Wang, and Gen Luo. Internvl3.5: Advancing open-source multimodal models in versatility, reasoning, and efficiency, 2025. URL https://arxiv.org/abs/2508.18265.

Yubo Wang, Chaohu Liu, Yanqiu Qu, Haoyu Cao, Deqiang Jiang, and Linli Xu. Break the visual perception: Adversarial attacks targeting encoded visual tokens of large vision-language models. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pp. 1072–1081, 2024.

Weilin Xu, David Evans, and Yanjun Qi. Feature squeezing: Detecting adversarial examples in deep neural networks. *arXiv preprint arXiv:1704.01155*, 2017.

Yue Xu, Xiuyuan Qi, Zhan Qin, and Wenjie Wang. Cross-modality information check for detecting jailbreaking in multimodal large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 13715–13726, 2024.

Senqiao Yang, Yukang Chen, Zhuotao Tian, Chengyao Wang, Jingyao Li, Bei Yu, and Jiaya Jia. Visionzip: Longer is better but not necessary in vision language models. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19792–19802. IEEE, 2025.

Xubing Ye, Yukang Gan, Yixiao Ge, Xiao-Ping Zhang, and Yansong Tang. Atp-llava: Adaptive token pruning for large vision language models. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24972–24982. IEEE, 2025.

Jiaming Zhang, Junhong Ye, Xingjun Ma, Yige Li, Yunfan Yang, Yunhao Chen, Jitao Sang, and Dit-Yan Yeung. Anyattack: Towards large-scale self-supervised adversarial attacks on vision-language models. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19900–19909. IEEE, 2025a.

Qizhe Zhang, Aosong Cheng, Ming Lu, Renrui Zhang, Zhiyong Zhuo, Jiajun Cao, Shaobo Guo, Qi She, and Shanghang Zhang. Beyond text-visual attention: Exploiting visual cues for effective token pruning in vlms. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 20857–20867. IEEE, 2025b.

Xiaomei Zhang, Zhaoxi Zhang, Leo Yu Zhang, Yanjun Zhang, Guanhong Tao, and Shirui Pan. Less is more–until it breaks: Security pitfalls of vision token compression in large vision-language models. *arXiv preprint arXiv:2601.12042*, 2026a.

Xinwei Zhang, Hangcheng Liu, Li Bai, Hao Wang, Qingqing Ye, Tianwei Zhang, and Haibo Hu. On the adversarial robustness of large vision-language models under visual token compression. *arXiv preprint arXiv:2601.21531*, 2026b.

Yuan Zhang, Chun-Kai Fan, Junpeng Ma, Wenzhao Zheng, Tao Huang, Kuan Cheng, Denis Gudovskiy, Tomoyuki Okuno, Yohei Nakata, Kurt Keutzer, et al. Sparsevlm: Visual token sparsification for efficient vision-language model inference. *arXiv preprint arXiv:2410.04417*, 2024.

APPENDIX CONTENTS

Appendix A. Optimization Details 13
Appendix B. Model-Specific Visual-Token Budgets 14
Appendix C. Detailed Evaluation Definitions 16
Appendix D. Experimental and Implementation Details 17
Appendix E. Qualitative Compression-Triggered Cases 19
Appendix F. Additional Ablation Results 22
Appendix G. Additional Cross-Model Results 22
Appendix H. Detection Details and Cross-Attack Analysis 27

A OPTIMIZATION DETAILS

A.1 DETAILED OPTIMIZATION PROCEDURE

Algorithm 2 presents the detailed optimization procedure corresponding to Algorithm 1. The target set $\mathcal{M}$ and its clean feature references are computed once and held fixed throughout optimization. At each iteration, attention scores and patch features are recomputed from the perturbed image to evaluate both losses on $\mathcal{M}$. Gradients are propagated through the frozen encoder to update only the input perturbation.

Algorithm 2 FATA optimization in detail

Require: Clean image x, visual encoder E, budget ϵ, step size α, iterations T, weight λ, target count M

Extract clean attention $\mathbf{s}^c$ and patch features $\mathbf{H}^c$ from $E(x)$
Fix $\mathcal{M} \leftarrow \mathrm{TopK}(\mathbf{s}^c, M)$
Initialize $\delta^{(0)} \sim \mathcal{U}(-\epsilon, \epsilon)$
for $t = 0, \ldots, T-1$ **do**
  $x_a^t \leftarrow \mathrm{clip}(x + \delta^{(t)}, 0, 1)$
  Extract $\mathbf{s}^a, \mathbf{H}^a \leftarrow E(x_a^t)$
  $\mathcal{L}_{\mathrm{attn}} \leftarrow \sum_{i \in \mathcal{M}} s_i^a$
  $\mathcal{L}_{\mathrm{sem}} \leftarrow 1 - \frac{1}{|\mathcal{M}|} \sum_{i \in \mathcal{M}} \cos(\mathbf{h}_i^a, \mathbf{h}_i^c)$
  $\mathcal{L}_{\mathrm{FATA}} \leftarrow \mathcal{L}_{\mathrm{attn}} + \lambda \mathcal{L}_{\mathrm{sem}}$
  $g^{(t)} \leftarrow \nabla_\delta \mathcal{L}_{\mathrm{FATA}}$
  $\delta^{(t+1)} \leftarrow \Pi_{\|\delta\|_\infty \le \epsilon}(\delta^{(t)} - \alpha\, \mathrm{sign}(g^{(t)}))$
end for
return $\mathrm{clip}(x + \delta^{(T)}, 0, 1)$

A.2 Sensitivity to PGD Hyperparameters

Both hyperparameter sweeps evaluate FATA on LLaVA-1.5-7B using TextVQA-Open and VisionZIP.

Evaluation data. Tables 9 and 10 report experiments on unfiltered TextVQA-Open data used during method development, before applying the subset-selection procedure in Appendix D.1. Table 1 uses the filtered evaluation subset, which retains examples answered correctly with the original image but incorrectly with a black image at full tokens. This subset emphasizes reliance on visual evidence and exhibits more pronounced compression-triggered attack effects. The different sample populations therefore explain the difference in absolute accuracy between these sweeps and the TextVQA–VisionZIP results in Table 1, even when the model, compressor, and hyperparameters coincide.

We use random-start ℓ_∞ PGD with $\epsilon = 2/255$, $T = 100$, and $M = 64$ fixed clean target tokens, and evaluate $K \in \{576, 192, 128, 64, 32, 16\}$, with $K = 576$ denoting full-token inference. We fix $\lambda = 1$ when varying the step size α and $\alpha = 0.5/255$ when varying the semantic-preservation weight λ. Step sizes are reported in units of $1/255$. Parameter selection considers full-token preservation and compressed accuracy over this trajectory.

PGD step size. At $\alpha = 0.5/255$, Table 9 reports 95.0% full-token accuracy and the lowest mean compressed accuracy, 84.1%, over $K \in \{192, 128, 64, 32, 16\}$. Reducing α to $0.25/255$ improves full-token accuracy to 96.0% but raises the compressed-budget mean to 84.8%. At $\alpha = 0.75/255$, the corresponding values are 94.3% and 84.4%. We choose $\alpha = 0.5/255$ for its stronger compression-triggered effect with high full-token accuracy. Its entries coincide with the shared $\lambda = 1$ setting in Table 10.

Table 9: FATA accuracy on unfiltered TextVQA-Open across token budgets and step sizes α.

K	.25	.5	.75	1	1.25	1.5	1.75
576	96.0	95.0	94.3	94.5	94.2	95.1	94.3
192	94.1	92.7	93.4	93.8	93.8	92.9	94.0
128	93.0	91.5	92.1	92.1	92.8	91.7	92.7
64	88.9	87.6	88.4	88.5	89.4	88.8	88.7
32	78.8	80.1	80.7	80.6	82.0	81.4	80.9
16	69.4	68.8	67.3	69.4	68.1	69.2	71.0

Semantic-preservation weight. Table 10 exposes the expected trade-off. Increasing λ steadily improves full-token accuracy from 94.1% at $\lambda = 0.5$ to 96.0% at $\lambda = 8$, but it also raises the $K = 16$ accuracy from 65.4% to 70.7%, indicating a weaker compression-triggered effect. The selected $\lambda = 1$ recovers 0.9 points of full-token accuracy over $\lambda = 0.5$ while retaining 68.8% accuracy at $K = 16$ and 80.1% at $K = 32$. Moving to $\lambda \geq 2$ provides at most another 1.0 point at full tokens, but generally increases compressed-budget accuracy; at $\lambda = 8$, for instance, the $K = 32$ and $K = 16$ accuracies rise to 81.0% and 70.7%. We choose $\lambda = 1$ to limit full-token damage while retaining the compression-triggered effect that weakens at larger weights.

Table 10: FATA accuracy on unfiltered TextVQA-Open across token budgets and semantic weights λ.

K	$\lambda = .5$	1	2	4	8
576	94.1	95.0	95.2	95.6	96.0
192	92.0	92.7	93.3	94.2	94.3
128	91.8	91.5	93.1	93.3	92.2
64	87.8	87.6	87.5	89.0	90.1
32	77.4	80.1	81.1	81.4	81.0
16	65.4	68.8	68.8	70.3	70.7

Together, the sweeps support $\alpha = 0.5/255$ and $\lambda = 1$ as a balance between full-token preservation and compression-triggered failure across budgets.

B Model-Specific Visual-Token Budgets

B.1 LLaVA-1.5-7B

LLaVA-1.5-7B receives a fixed sequence of 576 visual patch tokens from its vision encoder. We use $K = 576$ for full-token inference and evaluate the five compressed budgets $K \in \{192, 128, 64, 32, 16\}$. Relative to the full sequence, these settings retain $1/3$, $2/9$, $1/9$, $1/18$, and $1/36$ of the visual tokens, respectively. Here, K denotes the number of selected or merged visual representatives. In the reconstruction-based LLaVA path in Figure 2D, these representatives reconstruct an $N = 576$-slot sequence before the projector.

The practical budget is selected separately for each of the 16 dataset–compressor pairs. Let $\mathcal{K} = \{16, 32, 64, 128, 192\}$ denote the candidate budgets. Clean accuracy at each budget is measured relative to full-token clean accuracy:

$$R^{(d,c)}_{\text{clean}}(K) = \frac{\text{Acc}^{(d,c)}_{\text{Clean}}(K)}{\text{Acc}^{(d,c)}_{\text{Clean}}(576)}. \tag{9}$$

Both accuracies are measured on the same examples using identical prompts and the same task scorer. Open-ended tasks use normalized VQA scores, and multiple-choice tasks use option-letter accuracy. Retention ratios are computed from unrounded task accuracies.

Using this ratio, we select the smallest candidate budget $K^{(d,c)}_{\text{prac}}$ that retains at least 80% of full-token clean accuracy:

$$K^{(d,c)}_{\text{prac}} = \min \left\{ K \in \mathcal{K} : R^{(d,c)}_{\text{clean}}(K) \geq 0.80 \right\}. \tag{10}$$

The candidates are compared by their numeric values, from 16 upward. The uncompressed point $K = 576$ is excluded from $\mathcal{K}$ because its retention ratio relative to itself is necessarily one.

Table 11 reports the practical budgets obtained by applying Eq. 10 to each of the 16 dataset–compressor pairs.

Table 11: Clean-only practical token budgets $K^{(d,c)}_{\text{prac}}$ for LLaVA-1.5-7B. Each entry is the smallest $K \in \{16, 32, 64, 128, 192\}$ that retains at least 80% of the corresponding full-token clean accuracy.

Compressor	TextVQA-Open	VQAv2-Open	ScienceQA-MC	VQAv2-MC
VisionZIP	64	64	32	32
VisPruner	64	64	32	32
PruMerge	64	64	32	32
FlowCut	64	64	32	**64**

The selected budgets are fixed for all attacks. All four compressors use $K = 64$ on the open-ended tasks and $K = 32$ on ScienceQA-MC. On VQAv2-MC, FlowCut requires $K = 64$ for its clean accuracy at $K = 32$ falls below 80% of its full-token value; the other three compressors use $K = 32$.

The visual-token retention ratio measures the fraction of visual tokens retained at the selected budget. For LLaVA, it is

$$r^{(d,c)}_{\text{prac}} = \frac{K^{(d,c)}_{\text{prac}}}{576}. \tag{11}$$

$K = 64$ and $K = 32$ correspond to retention fractions of $1/9$ and $1/18$, respectively. CBR is evaluated at these budgets using the definition in Appendix C.

B.2 Qwen3.5 and InternVL3.5

The number of visual tokens in Qwen3.5 and InternVL3.5 varies across images because of their image preprocessing and token organization. For each image i with N_i visual tokens, the compression budget is specified as a retained fraction r of N_i.

The main comparisons on Qwen3.5 and InternVL3.5 report full visual tokens, $1/3$ retention, and a practical budget specified for each task. The $1/3$ fraction matches the proportion retained by LLaVA at $K = 192$. Before adversarial evaluation, the practical fractions are fixed at $1/9$ for TextVQA-Open and VQAv2-Open and $1/18$ for ScienceQA-MC and VQAv2-MC, corresponding to LLaVA's $64/576$ and $32/576$ settings, respectively. These fractions apply to all four compressors, with the visual token budgets and sample counts for all three models summarized in Table 12.

Table 12: Model-specific budgets and sample counts used in evaluation. Practical budgets are given separately for open-ended and multiple-choice tasks.

Model	Full	Practical Open	Practical MC	Valid images per dataset
LLaVA-1.5-7B	576	64	32 or 64	1,000
Qwen3.5-9B	N_i	$r = 1/9$	$r = 1/18$	1,000
InternVL3.5-8B-HF	N_i	$r = 1/9$	$r = 1/18$	998 or 1,000

C Detailed Evaluation Definitions

The metrics in Section 3.3 use the model-specific budgets in Table 12. Accuracy enters the equations as a fraction in $[0, 1]$ and is reported as a percentage. The metrics in Section 3.3 are evaluated using the budgets and sample counts summarized in Table 12.

Task scores and binary correctness. For a dataset with n valid examples, let $v_i \in [0, 1]$ denote the task credit and $c_i \in \{0, 1\}$ the binary correctness of prediction i. The corresponding accuracies are

$$\mathrm{Acc}^{\mathrm{task}} = \frac{1}{n}\sum_{i=1}^{n} v_i, \qquad \mathrm{Acc}^{\mathrm{EM}} = \frac{1}{n}\sum_{i=1}^{n} c_i. \tag{12}$$

Table 13 gives the task-specific rules. Open answers use the benchmark's normalization of punctuation, articles, and number words before matching.

Table 13: Task-level scoring rules for reported accuracy and binary correctness. CBR uses the binary decisions in the final column.

Task	Task credit v_i	Binary correctness c_i
TextVQA-Open and VQAv2-Open	Normalized VQA credit against the human reference answers, with partial credit allowed.	1 for a normalized exact match to an accepted reference answer; 0 otherwise.
ScienceQA-MC	1 if the parsed letter (A–F) matches the annotated answer index converted to a letter; 0 otherwise.	The same option-letter decision as v_i.
VQAv2-MC	1 if the parsed letter (A–D) matches the fixed target derived from the modal human answer; 0 otherwise.	The same option-letter decision as v_i.

Metric computation. The accuracy tables for all three models report $\mathrm{Acc}^{\mathrm{task}}$. SR is computed using this accuracy for LLaVA and Qwen3.5 and $\mathrm{Acc}^{\mathrm{EM}}$ for InternVL3.5, and is averaged over the four datasets.

CER is reported only for LLaVA and is the complement of $\mathrm{Acc}^{\mathrm{task}}$ at K_{prac}, following Eq. 7. It retains partial task credit and measures overall error without conditioning on Clean correctness or isolating attack-induced failures. The summary is the equal-weight mean over the 16 dataset–compressor settings, calculated from the displayed accuracies, expressed as a percentage, and rounded to two decimal places, as specified alongside Table 2.

CBR uses the binary correctness decisions c_i for all three models, following Eq. 8. For attack m, its denominator requires $c_i = 1$ for Clean at full tokens, attack m at full tokens, and Clean under the evaluated compressor and practical budget; the numerator additionally requires $c_i = 0$ for attack m under that same compression setting. Macro CBR is the unweighted mean over the 16 dataset–compressor pairs, whereas pooled CBR is computed by summing their numerator counts and denominator counts before division.

D Experimental and Implementation Details

Main benchmark. The main victim is LLaVA-1.5-7B with a CLIP ViT-Large visual encoder. We test VisionZIP, VisPruner, PruMerge, and FlowCut at $K \in \{576, 192, 128, 64, 32, 16\}$. Each of TextVQA-Open, VQAv2-Open, ScienceQA-MC, and VQAv2-MC contains 1,000 evaluation samples. Base and FATA use $\epsilon = 2/255$, $\alpha = 0.5/255$, and 100 steps; CAA uses the same $\epsilon = 2/255$ comparison budget with $\alpha = 1/255$ and 100 steps; CAGE is run under $\epsilon = 2/255$.

D.1 Evaluation Subset Construction

The procedure below documents the released dataset reconstruction. The reported experiments retain their archived evaluation mappings; in particular, the original VQAv2-MC options cannot be reconstructed from the released builder's seed alone.

Selection unit and representative question. Each subset uses unique images as the selection unit. Images are deduplicated using source identifiers and image content. For each image, the first eligible question in source order is fixed before model inference. If the resulting pair fails the selection criteria, no alternative question is selected for that image.

Selection based on model responses. LLaVA decodes each candidate greedily with at most 32 new tokens, using the same prompt and question for the original image and a 336×336 black image. Both conditions use all 576 visual tokens and greedy decoding with at most 32 new tokens. We retain an image only if the answer is incorrect with a 336×336 black image and correct with the original image. Let q_i denote the representative question assigned to image x_i, b the black image, and $c(\hat{y}, y)$ the correctness rule used during subset construction. The evaluation subset is

$$\mathcal{D}_j^{\text{eval}} = \text{First}_{1000}\{(x_i, q_i, y_i) \in \mathcal{U}_j : c(f_{576}(b, q_i), y_i) = 0 \ \wedge \ c(f_{576}(x_i, q_i), y_i) = 1\}, \qquad (13)$$

where $\mathcal{U}_j$ contains the candidate question–image pairs and reference answers for dataset j, ordered by source occurrence after image deduplication. First_{1000} selects the first 1,000 pairs satisfying both conditions. Selection uses no attack predictions or results under token compression.

TextVQA-Open. We use the TextVQA validation split. During selection, open answers are lowercased, punctuation and articles are removed, and number words from zero through five are mapped to digits. A prediction passes if it exactly matches a normalized reference answer or contains that answer within the generated phrase.

VQAv2-Open. We use the VQAv2 validation split and exclude yes/no questions. Selection uses the same answer normalization and matching rule as TextVQA-Open.

VQAv2-MC. The released reconstruction uses the VQAv2 validation split, excludes yes/no questions and invalid option sets, and pairs the modal human answer with three distinct distractors. Its configured global seed is $s_0 = 20260904$. For each question, the UTF-8 string formed by `vqav2-mc-options-v1`, s_0 in decimal, and `question_id`, separated by null bytes, is hashed with SHA-256. The first eight digest bytes, interpreted as an unsigned big-endian integer, seed a local Python `random.Random` instance for distractor draws and option shuffling. Thus s_0 is a fixed configuration constant, while each question's random seed is derived deterministically from it and the question ID. The reconstruction excludes images selected for VQAv2-Open. For the archived experiments, the original option-generation seed and RNG state are unavailable; exact reproduction uses the saved questions, option order, and target letters in the archived mapping.

ScienceQA-MC. The candidate pool combines the ScienceQA train, validation, and test splits for evaluation only. We discard records without images or with invalid metadata or choices. Image deduplication also removes exact duplicates across splits. The optional hint is prepended to the question, the original choices are labeled A–F, and the annotated answer index determines the target letter.

Relation to reported accuracy. The reported evaluations use archived question–image mappings with their saved answer specifications; the released reconstruction above does not redefine those samples. Evaluation uses each model's prompt and formal task scorer, so clean full-token accuracy can be below 100%. Task scoring and metric computation are detailed in Appendix C.

D.2 Performance across Token Budgets

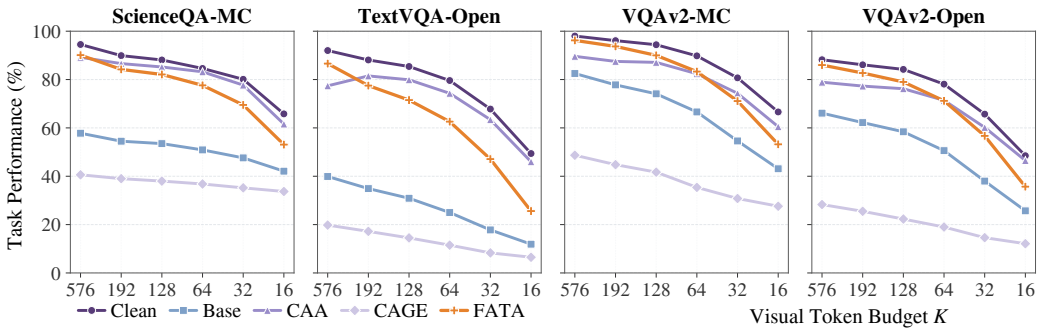


Figure 5: Average task performance versus visual-token budget. The exact per-method values are reported in Table 1 in the main paper.

Figure 5 visualizes the results in Table 1 of the main paper for each dataset. The point at $K = 576$ reports full-token accuracy, and each compressed-budget point averages accuracy over VisionZIP, VisPruner, PruMerge, and FlowCut.

TextVQA-Open shows the largest Clean–FATA gap at every compressed budget. Under severe compression, both open-ended tasks reach lower Clean and FATA accuracies than the multiple-choice tasks. Across all four datasets, the Clean–FATA gap widens as the token budget decreases.

E QUALITATIVE COMPRESSION-TRIGGERED CASES

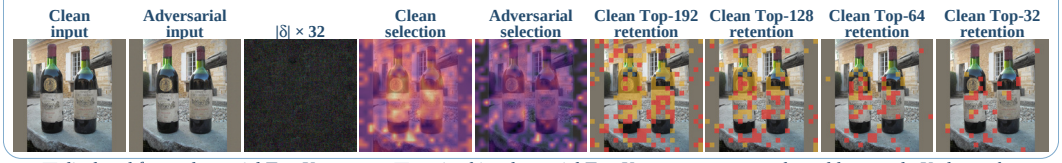
In the LLaVA examples, selection maps show last-layer CLIP CLS-to-patch attention on a shared monotonic log scale that preserves token rankings. This visual-encoder attention is independent of the language input. Each Clean Top-K panel partitions all K clean tokens into those present in the adversarial Top-K set (gold) and those displaced (red).

E.1 LLAVA-1.5-7B

(a) TextVQA-Open

Question: what is the brand name first word?
Ground truth: chateau

Compressor	Full	K=192	K=128	K=64	K=32
VisionZIP	Chateau	Chateau	Travemunde	Merlot	Merlot
VisPruner	Chateau	Merlot	Chateau	Travaux	Patek philippe
PruMerge	Chateau	Chateau	Merlot	Chateau	Gerger C/F
FlowCut	Chateau	Travemunde	Travemunde	Meridian	None

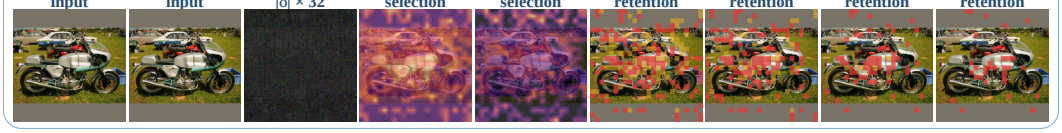


displaced from adversarial Top-K · retained in adversarial Top-K · red + gold = exactly K clean tokens
FATA correct · Clean correct, FATA wrong · Both wrong (C/F) · Clean wrong, FATA correct

(b) VQAv2-Open

Question: What color is the bike?
Ground truth: silver

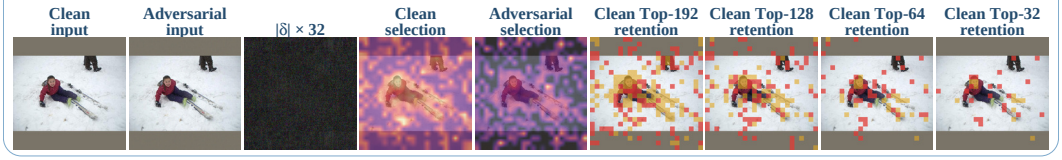
Compressor	Full	K=192	K=128	K=64	K=32
VisionZIP	Silver	Blue	Silver	Blue	Silver
VisPruner	Silver	Blue	Silver	Blue	Silver
PruMerge	Silver	Blue	Blue	White	Silver
FlowCut	Silver	Silver	Silver	Blue	Black C/F



(c) VQAv2-MC

Question: What color are the kids skis?
Ground truth: B: gray
Options: A. black; B. gray; C. pink; D. brown

Compressor	Full	K=192	K=128	K=64	K=32
VisionZIP	B: gray	A: black	A: black C/F	A: black	A: black
VisPruner	B: gray	B: gray	C: pink	B: gray	D: brown
PruMerge	B: gray	A: black	A: black C/F	A: black C/F	B: gray
FlowCut	B: gray	B: gray	A: black	A: black C/F	A: black



(d) ScienceQA-MC

Question: Which of these states is farthest east?
Ground truth: D: Tennessee
Options: A. Washington; B. Montana; C. Utah; D. Tennessee

Compressor	Full	K=192	K=128	K=64	K=32
VisionZIP	D: Tennessee	A: Washington	A: Washington	A: Washington	A: Washington
VisPruner	D: Tennessee	A: Washington	A: Washington	C: Utah C/F	D: Tennessee
PruMerge	D: Tennessee	A: Washington	D: Tennessee	D: Tennessee	A: Washington
FlowCut	D: Tennessee	A: Washington C/F	C: Utah C/F	A: Washington C/F	C: Utah

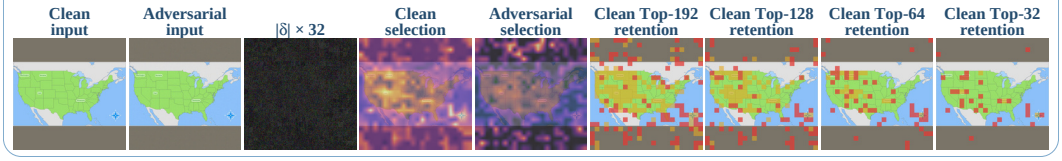


Figure 6: LLaVA-1.5-7B cases from TextVQA-Open (a), VQAv2-Open (b), VQAv2-MC (c), and ScienceQA-MC (d). Tables report FATA predictions across compressors and token budgets, with the shared legend below (a) applying to all panels.

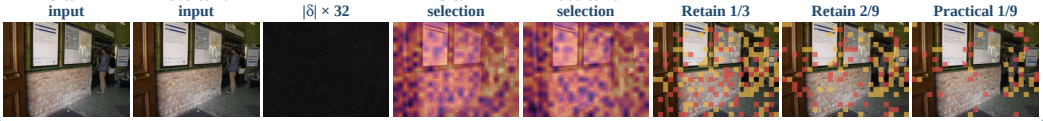
E.2 QWEN3.5-9B

Selection maps visualize the ℓ_2 norm of post-merger visual features on a shared logarithmic scale. The retention panels track the corresponding global Top-K sets; each compressor follows its own selection rule. Full-token predictions remain correct for Clean and FATA in all four cases. Compressed responses depend on both the compressor and retention fraction. In VQAv2-Open, VisionZIP recovers the correct FATA answer at retention $1/9$ after errors at the two larger fractions.

(a) TextVQA-Open — Full = 475

Question: what are these people most likely purchasing?
Ground truth: tickets

Compressor	Full	1/3	2/9	1/9 Practical
VisionZIP	tickets	tickets	books	books
VisPruner	tickets	tickets	tickets	books
PruMerge	tickets	books	books	books
FlowCut	tickets	books	books	books



(b) VQAv2-Open — Full = 304

Question: What color is the tip of the birds' tails?
Ground truth: yellow

Compressor	Full	1/3	2/9	1/9 Practical
VisionZIP	yellow	black	black	yellow
VisPruner	yellow	black	black	black
PruMerge	yellow	black	black	black
FlowCut	yellow	yellow	black	black



(c) VQAv2-MC — Full = 300

Question: How many people are seated?
Ground truth: A: 2
Options: A. 2; B. 5; C. few; D. 16

Compressor	Full	1/3	2/9	1/9	1/18 Practical
VisionZIP	A: 2	B: 5	A: 2	A: 2	C: few
VisPruner	A: 2	B: 5	B: 5	B: 5	B: 5
PruMerge	A: 2	C: few	B: 5	B: 5	C: few
FlowCut	A: 2	C: few	C: few	C: few	C: few



(d) ScienceQA-MC — Full = 460

Question: Which country is highlighted?
Ground truth: B: Fiji
Options: A. Tonga; B. Fiji; C. Papua New Guinea; D. Tuvalu

Compressor	Full	1/3	2/9	1/9	1/18 Practical
VisionZIP	B: Fiji	A: Tonga	B: Fiji	A: Tonga	A: Tonga
VisPruner	B: Fiji	A: Tonga	B: Fiji	A: Tonga	A: Tonga
PruMerge	B: Fiji	A: Tonga	B: Fiji	A: Tonga	A: Tonga
FlowCut	B: Fiji	A: Tonga	A: Tonga	B: Fiji	B: Fiji

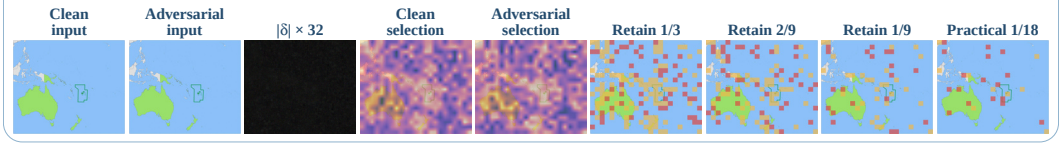


Figure 7: Qwen3.5-9B cases in the same task order as Figure 6. Tables report FATA answers using the shared colour conventions.

E.3 INTERNVL3.5-8B-HF

InternVL processes local image tiles and a thumbnail, whose CLS-to-patch scores and token footprints are projected onto the original image. In VQAv2-MC, clean support for the shirt comes mainly from thumbnail tokens. At practical retention, all four compressors lose their target-overlapping clean anchors and return incorrect FATA answers.

(a) TextVQA-Open — Full = 1792

Question: what city is listed on the bottle?
Ground truth: milan

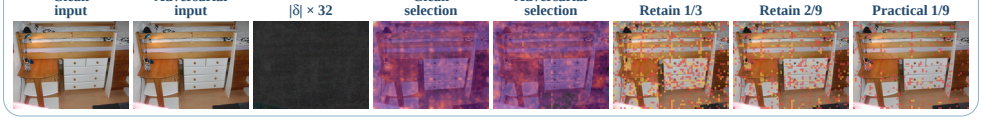
Compressor	Full	1/3	2/9	1/9 Practical
VisionZIP	milan	michigan	michigan	san francisco
VisPruner	milan	michigan	larry's san francisco	baltimore
PruMerge	milan	milan	milan	san francisco
FlowCut	milan	michigan	michigan	larry's



(b) VQAv2-Open — Full = 3328

Question: What color is lamp?
Ground truth: blue

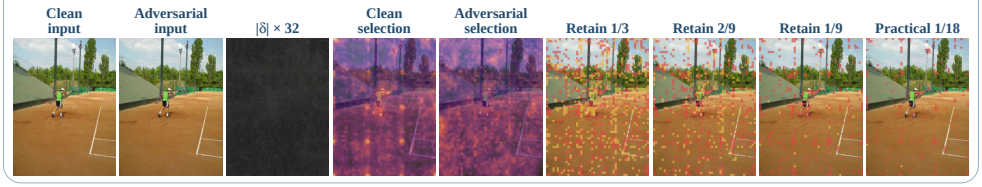
Compressor	Full	1/3	2/9	1/9 Practical
VisionZIP	blue	black	red	white
VisPruner	blue	blue	black	white
PruMerge	blue	white	white	red C/F
FlowCut	blue	white	red	red



(c) VQAv2-MC — Full = 3328

Question: What color is the person's shirt?
Ground truth: A: green
Options: A. green; B. gold; C. white; D. black

Compressor	Full	1/3	2/9	1/9	1/18 Practical
VisionZIP	A: green	C: white	C: white	D: black	C: white
VisPruner	A: green	A: green	D: black	C: white	C: white
PruMerge	A: green	A: green	A: green	D: black	C: white
FlowCut	A: green	A: green	D: black	C: white	C: white



(d) ScienceQA-MC — Full = 3328

Question: Which country is highlighted?
Ground truth: C: Haiti
Options: A. Trinidad and Tobago; B. Jamaica; C. Haiti; D. Cuba

Compressor	Full	1/3	2/9	1/9	1/18 Practical
VisionZIP	C: Haiti	C: Haiti	C: Haiti	A: Trinidad and Tobago	A: Trinidad and Tobago
VisPruner	C: Haiti	D: Cuba	A: Trinidad and Tobago	A: Trinidad and Tobago	B: Jamaica C/F
PruMerge	C: Haiti	B: Jamaica	A: Trinidad and Tobago	A: Trinidad and Tobago	A: Trinidad and Tobago
FlowCut	C: Haiti	C: Haiti	A: Trinidad and Tobago	B: Jamaica	A: Trinidad and Tobago

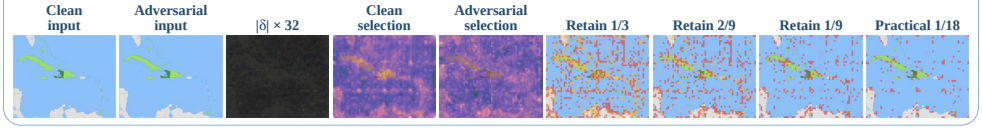


Figure 8: InternVL3.5-8B-HF cases across the four tasks, with global native-priority Top-K panels and the colour conventions of Figure 6.

F Additional Ablation Results

Figure 9 shows how the loss variants behave within individual dataset and compressor settings. In panel (a), attention-only SR spans roughly 40–90%, whereas semantic-only values cluster near 100%. Full FATA keeps all plotted SR values above 90%, showing that semantic preservation also reduces variation across settings. Panel (b) shows a range for full FATA CBR, from roughly 9% to 55%. Most settings remain below 30%, with two near 40% and 55%. The trajectories from semantic-only to full FATA include both increases and decreases in CBR, so the aggregate gain does not hold uniformly across settings. Together, the panels show consistent full-token retention under the joint objective and a more variable compression-triggered effect.

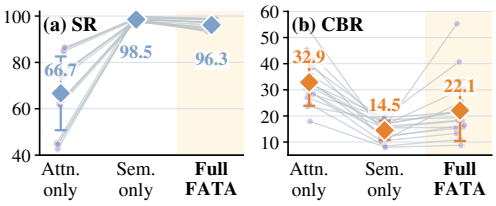


Figure 9: Loss ablation across 16 dataset–compressor settings, with paired lines and mean diamonds; shading marks full FATA. Error bars show ± 1 sample standard deviation across settings

G Additional Cross-Model Results

This appendix details the model-specific objectives and supplementary analyses for the cross-model experiments in Section 4.3. Visual-token budgets and scoring rules are specified in Appendices B and C, respectively.

G.1 Qwen3.5 task-aware FATA adaptation

The Qwen3.5 adaptation uses task-level losses to preserve full-token predictions while inducing errors under compression, following FATA's design principle. This white-box adaptation uses ground-truth answers and downstream Qwen3.5 gradients.

Task-aware objective. Let x_a be the perturbed input, y the ground-truth answer, m_t the compressor selected at PGD step t, and r_t the retained-token fraction used at that step. In the released Qwen implementation, x denotes the processor's pixel-value tensor and $x_a = \mathrm{clip}(x + \delta, -1, 1)$. The four compressors are cycled round-robin, and the budget schedule emphasizes the practical operating point while revisiting $1/3$ retention. We write the compression-side attack score as

$$\mathcal{A}_t = w_{\text{task}}\, \mathcal{L}_{\text{task}}^{m_t, r_t} + w_{\text{rank}}\, \mathcal{D}_{\text{rank}} + w_{\text{comp}}\, \mathcal{D}_{\text{comp}}, \tag{14}$$

where $\mathcal{L}_{\text{task}}^{m_t, r_t}$ is the teacher-forced ground-truth cross-entropy under compressed inference, written as a score to be maximized. The locked coefficients are $w_{\text{task}} = 1.5$, $w_{\text{rank}} = 0.25$, and $w_{\text{comp}} = 0.05$.

Rank and compressor auxiliaries. The N merged tokens in `pooler_output` are denoted by $z_i(x)$. Rank the clean tokens by descending $\|z_i(x)\|_2$. For each of the medium and practical budgets, $K = \max(1, \mathrm{round}(Nr))$ with $r = 1/3$ and $r = 1/9$ (Open) or $1/18$ (MC), respectively, take

$$w_K = \min\{\max(8, \mathrm{round}(0.05N))\,, K, N - K\} \tag{15}$$

tokens on each side of the cutoff: the core set contains clean ranks $K - w_K + 1, \ldots, K$, and the decoy set contains ranks $K + 1, \ldots, K + w_K$. Let C and D be the respective deduplicated unions over the two budgets. Both sets are fixed before optimization and reused at every step.

Let $\overline{v}_S = |S|^{-1} \sum_{i \in S} v_i$ and $n_i = \|z_i(x_a)\|_2$, and let $\mathcal{M}$ contain VisionZIP, VisPruner, FlowCut, and PruMerge. The released implementation defines the maximization scores as

$$\mathcal{D}_{\text{rank}} = -\,\mathrm{softplus}(0.05 + \overline{n}_C - \overline{n}_D)\,, \tag{16}$$

$$\mathcal{D}_{\text{comp}} = -\frac{1}{4} \sum_{m \in \mathcal{M}} \mathrm{softplus}(0.05 + \overline{s}_{m,C} - \overline{s}_{m,D})\,. \tag{17}$$

Thus, both margins are 0.05, and softplus is applied after taking the core and decoy means. The negative signs convert the implemented boundary losses into scores to maximize. Unlike the task term,

these auxiliaries use the fixed two-budget targets; the compressor term averages all four surrogates at every step.

For completeness, the surrogate token scores are

$$\begin{aligned} s_{\text{VisionZIP}} &= 0.75a + 0.25v, & s_{\text{VisPruner}} &= 0.5a + 0.5b, \\ s_{\text{FlowCut},i} &= (a_i + c_i)\|z_i\|_1, & s_{\text{PruMerge}} &= 0.7a + 0.3p. \end{aligned} \tag{18}$$

Here all features are evaluated at x_a. Define $\text{Norm}_1(u) = u/(\sum_i u_i + 10^{-8})$, $a = \text{Norm}_1(n)$, and $b = \text{Norm}_1(d)$, where $d_i = 1 - (\sum_j \cos(z_i, z_j) - 1)/(N-1)$. The distinctiveness score is $v_i = 1 - \max_{j \in T_{\text{dom}}} \cos(z_i, z_j)$, where T_{dom} contains the $\max(1, \lfloor N/4 \rfloor)$ current tokens with largest feature norms. With $\bar{z} = N^{-1} \sum_j z_j$, $c = \text{Norm}_1(([\cos(z_i, \bar{z})]_+)_i)$. Finally, $p = b$ for $N \le 1024$ and $p = \text{Norm}_1((1/N, \dots, 1/N))$ otherwise. The PruMerge surrogate receives no spatial grid in this call, and all four surrogate temperatures equal one.

Full-token preservation is imposed with two complementary terms. The first keeps the ground-truth answer likely at full tokens,

$$\mathcal{L}_{\text{full}} = \text{CE}\big(p_\theta(\cdot \mid x_a, \text{Full}), y\big), \tag{19}$$

and the second distills the clean full-token predictive distribution,

$$\mathcal{L}_{\text{distill}} = D_{\text{KL}}\left(p_\theta(\cdot \mid x, \text{Full}) \,\|\, p_\theta(\cdot \mid x_a, \text{Full})\right). \tag{20}$$

Both are computed with teacher forcing during optimization; autoregressive generation is reserved for evaluation. The preservation branch is

$$\mathcal{P} = w_{\text{full}} \mathcal{L}_{\text{full}} + w_{\text{distill}} \mathcal{L}_{\text{distill}}, \qquad w_{\text{full}} = w_{\text{distill}} = 1. \tag{21}$$

The two branches summarize the ascent and preservation signs as

$$\mathcal{J}_t(\delta) = \mathcal{A}_t - \mathcal{P}. \tag{22}$$

The actual update separately normalizes each term's gradient before weighting, rather than differentiating this fixed weighted sum. With $\mathcal{N}(g) = g/(\|g\|_2 + 10^{-8})$, using the L2 norm over the entire gradient tensor, its direction is

$$\begin{aligned} g_t = {} & 1.5\,\mathcal{N}(\nabla_\delta \mathcal{L}_{\text{task}}^{m_t, r_t}) + 0.25\,\mathcal{N}(\nabla_\delta \mathcal{D}_{\text{rank}}) + 0.05\,\mathcal{N}(\nabla_\delta \mathcal{D}_{\text{comp}}) \\ & - \mathcal{N}(\nabla_\delta \mathcal{L}_{\text{full}}) - \mathcal{N}(\nabla_\delta \mathcal{L}_{\text{distill}}). \end{aligned} \tag{23}$$

Each gradient includes the derivative mask of the input clipping operation. Random-start ℓ_∞ PGD then updates

$$\delta^{(t+1)} = \Pi_{[-\epsilon, \epsilon]}\left[\delta^{(t)} + \alpha\ \text{sign}\,(g_t)\right], \qquad x_a^{(t+1)} = \text{clip}(x + \delta^{(t+1)}, -1, 1), \tag{24}$$

with $\epsilon = 2/255$, $\alpha = 0.5/255$, and $T = 100$ steps in the processor input domain. The final attack uses no additional gradient-projection operator. One adversarial image is generated per input and shared across all four compressors and evaluation budgets.

The compression branch aims to alter which visual evidence survives compression. The preservation branch uses full-token cross-entropy and clean-logit distillation to preserve task semantics under Qwen3.5's dynamic tokenization.

Dataset-level amplification. Table 14 complements the aggregate results in Table 4 with a dataset-level breakdown. Here D_r denotes the Clean–FATA accuracy difference at budget r, and $A_r = D_r - D_{\text{Full}}$. Both are calculated before rounding. Practical-budget amplification is positive across all four datasets.

G.2 InternVL3.5 architecture-aware FATA adaptation

The InternVL3.5 adaptation combines compression-side disruption with preservation of the uncompressed representation. Its archived V5 objective includes VisionZIP, VisPruner, DivPrune, PruMerge, and FlowCut; the reported comparisons show four of these compressors, excluding DivPrune.

Table 14: Qwen3.5 compression-specific damage and amplification by dataset (percentage points). Positive amplification means that compression increases the attack-induced accuracy gap relative to full-token inference.

Dataset	D_{Full}	$D_{1/3}$	D_{Prac}	$A_{1/3}$	A_{Prac}
TextVQA-Open	0.70	5.22	8.60	4.52	7.90
VQAv2-Open	1.00	10.00	14.57	9.00	13.57
ScienceQA-MC	-0.20	9.47	23.70	9.67	23.90
VQAv2-MC	1.00	8.05	19.60	7.05	18.60

Features and post-compression disruption. For an image with J tiles, let $H(x) \in \mathbb{R}^{J\times 256\times d_h}$ contain the post-shuffle features before the projector, and let $P(x) \in \mathbb{R}^{J\times 256\times d_p}$ contain the projector outputs. Thus $N = 256J$; all tiles, including a thumbnail when present, contribute patch tokens, with CLS excluded. For compressor c and retention fraction r, the attack pass produces $Z_{c,r}(x)$ with $K_{c,r}$ output tokens. The post-compression score compares corresponding output positions:

$$\mathcal{D}^{c,r}_{\text{pcd}} = 1 - \frac{1}{K_{c,r}} \sum_{j=1}^{K_{c,r}} \cos\big(z^{c,r}_j(x_a), z^{c,r}_j(x)\big)\,. \tag{25}$$

The cosine is computed along the feature dimension and then averaged over tokens, without pooling beforehand. This attack pass disables reconstruction; PruMerge returns selected centers without merging in this mode. The comparison follows output order rather than matching source-token identities. Clean outputs are fixed, while adversarial selections are recomputed at each step.

Cutoff targets and retained sets. At the practical fraction r_*, the total budget is $K = \max(1, \text{round}(Nr_*))$. Each tile first receives $\lfloor K/J \rfloor$ tokens, with the remainder assigned to earlier tiles. For a tile budget k_t, sort the clean family scores $s_{c,t,i}(x)$ in descending order and set

$$b_t = \min\{64, \max(16, \text{round}(0.25k_t))\,, k_t - 1, 255 - k_t\}\,. \tag{26}$$

The ordered lists $\mathcal{B}^+_{c,t}$ and $\mathcal{B}^-_{c,t}$ contain ranks $k_t - b_t + 1, \ldots, k_t$ and $k_t + 1, \ldots, k_t + b_t$, respectively. Each has b_t entries: 16 for Open and 13 or 14 for MC at the practical fractions. They are fixed from the clean image throughout optimization, without combining different budgets. Ties use the descending `argsort` order, with no extra tie rule. Separately, $\mathcal{U}_{c,t}$ contains the source-token indices actually selected by clean compressor c in tile t at r_*. Its size is the actual number of selected representatives, with no additional cap. These retained sets are also fixed and need not equal the boundary lists or their union.

The ranking score pairs entries at the same position in the two lists:

$$\mathcal{D}^{c}_{\text{rank}} = -\frac{1}{J}\sum_{t=1}^{J}\frac{1}{b_t}\sum_{j=1}^{b_t}\Big[s_{c,t,\mathcal{B}^+_{c,t}[j]}(x_a) - s_{c,t,\mathcal{B}^-_{c,t}[j]}(x_a)\Big]_+\,, \qquad [u]_+ = \max(u, 0). \tag{27}$$

The margin is $\mu = 0$: this is a paired ReLU penalty expressed as a score to maximize. All reported tiles have nonempty boundary lists; an empty list is omitted from the tile mean.

Family scores. Write A for the number of attention heads and $\alpha_{a,i}$ for penultimate-layer CLS attention after summing each four-patch group into token i. With $u_i = A^{-1}\sum_a \alpha_{a,i}$, the VisionZIP and VisPruner rank scores are $\sum_a \alpha_{a,i}$ and u_i, respectively. PruMerge uses final-layer CLS query–key attention, while FlowCut combines normalized attention, value similarity, and value magnitude:

$$\begin{aligned} s_{\text{PruMerge},i} &= \text{softmax}_{j=0,\ldots,N}\left(q_0^\top k_j/\sqrt{d_k}\right)_i\,, \\ e_i &= \frac{1}{A}\sum_a \text{softmax}_{j=0,\ldots,N}\left(v_{a,0}^\top v_{a,j}\right)_i\,, \\ s_{\text{FlowCut},i} &= [\text{Norm}(u)_i + \text{Norm}(e)_i]\left\|\frac{1}{A}\sum_a v_{a,i}\right\|_1\,, \end{aligned} \tag{28}$$

where $\mathrm{Norm}(w) = w/(\sum_{i=1}^{N} w_i + 10^{-8})$. In these two archived rank helpers, index 0 is the first tile's CLS and indices $1, \ldots, N$ span the image-wide patch stream; actual compressor passes remain tile-local. The Q/K/V features are aligned by concatenating the four patch-group slots and repeating CLS to the matching width d_k or d_v. FlowCut uses penultimate-layer values, with no scale or temperature in its value softmax; CLS is dropped before Norm. DivPrune uses $s_{\mathrm{DivPrune},t,i} = \min_{j \in \mathcal{U}_{\mathrm{DivPrune},t}} [1 - \cos(h_{t,i}, h_{t,j})]$, allowing self-matches. Evaluating these scores on clean features fixes the cutoff lists; their adversarial values enter Eq. 27.

Diversity and relation/value terms. For $c \in \{\mathrm{DivPrune}, \mathrm{VisPruner}\}$, the diversity score increases nearest-neighbor similarity within the fixed retained set:

$$\mathcal{D}^{c}_{\mathrm{div}} = \frac{1}{J} \sum_{t} \frac{1}{|\mathcal{U}_{c,t}|} \sum_{i \in \mathcal{U}_{c,t}} \max_{j \in \mathcal{U}_{c,t} \setminus \{i\}} \hat{h}_{t,i}(x_a)^{\top} \hat{h}_{t,j}(x_a), \quad \hat{h} = \frac{h}{\max(\|h\|_2, 10^{-12})}. \tag{29}$$

Self-pairs are excluded, and tiles with fewer than two selected tokens are omitted from the mean. The pairwise cosine distance is $1 - \hat{h}_i^{\top} \hat{h}_j$; Eq. 29 averages each selected token's largest non-self similarity.

The relation term uses FlowCut's selected-token value magnitude. Let $V_{a,i}(x_a)$ be the penultimate-layer value of global patch token i in head a, define $\nu_i = \|A^{-1} \sum_a V_{a,i}(x_a)\|_1$, and map the tile-local retained sets into their image-wide union $\mathcal{U}_{\mathrm{FlowCut}}$. Then

$$\mathcal{D}_{\mathrm{rel}} = -\frac{|\mathcal{U}_{\mathrm{FlowCut}}|^{-1} \sum_{i \in \mathcal{U}_{\mathrm{FlowCut}}} \nu_i}{\max\left(N^{-1} \sum_{i=1}^{N} \nu_i, 10^{-6}\right)}. \tag{30}$$

Head averaging precedes the feature L1 norm. Normalization divides the selected-token mean by the mean over all adversarial patch tokens, with a denominator floor of 10^{-6}. Gradients propagate through both means.

Attention and redundancy targets. Let $a_{t,i}$ average CLS attention over heads and the last three vision layers, then average each four-patch group into a post-shuffle token. In each tile, the clean ranking fixes the top $q = \mathrm{round}(256 \cdot 64/576) = 28$ targets $\mathcal{T}_t$, the next 28 decoys $\mathcal{E}_t$, and width-8 lists $\mathcal{A}^{\pm}_{\rho,t}$ on either side of rank $\mathrm{round}(256\rho)$ for $\rho \in \mathcal{R} = \{1/3, 1/9, 1/18\}$. Let $\langle \cdot \rangle$ average over tiles and corresponding list positions. The attention score is

$$\begin{aligned} \mathcal{D}_{\mathrm{attn}} = & -\langle a_{\mathcal{T}}(x_a) \rangle - \langle [a_{\mathcal{T}}(x_a) - a_{\mathcal{E}}(x_a)]_+ \rangle \\ & - \frac{1}{3} \sum_{\rho \in \mathcal{R}} \langle [a_{\mathcal{A}^+_\rho}(x_a) - a_{\mathcal{A}^-_\rho}(x_a)]_+ \rangle. \end{aligned} \tag{31}$$

The redundancy score compares the same targets with the other 228 projected tokens within their tile:

$$\mathcal{D}_{\mathrm{red}} = \frac{1}{Jq} \sum_{t} \sum_{i \in \mathcal{T}_t} \max_{j \notin \mathcal{T}_t} \cos(p_{t,i}(x_a), p_{t,j}(x_a)). \tag{32}$$

These target/complement sets differ from the compressor-specific retained sets $\mathcal{U}_{c,t}$.

Semantic preservation and pooling. The semantic hierarchy consists of target tokens, tile means, and an image mean of the projector output. Pooling is arithmetic averaging: $\mathrm{pool}(\{p_i\}_{i \in S}) = |S|^{-1} \sum_{i \in S} p_i$. Define $\bar{p}_t = 256^{-1} \sum_i p_{t,i}$ and $\bar{p} = N^{-1} \sum_{t,i} p_{t,i}$. The three levels receive equal weight:

$$\begin{aligned} \mathcal{L}_{\mathrm{sem}} = \frac{1}{3} \Bigg[& \frac{1}{Jq} \sum_{t} \sum_{i \in \mathcal{T}_t} (1 - \cos(p_{t,i}(x_a), p_{t,i}(x))) \\ & + \frac{1}{J} \sum_{t} (1 - \cos(\bar{p}_t(x_a), \bar{p}_t(x))) + 1 - \cos(\bar{p}(x_a), \bar{p}(x)) \Bigg]. \end{aligned} \tag{33}$$

Tile and image means include all full-token patch positions and any thumbnail tile, without CLS. Equal tile sizes make the mean of tile means identical to the mean over all N tokens. PCD, semantic, and projected-redundancy cosines use a feature-norm floor of 10^{-8}. Pooling applies to the two semantic means; PCD compares compressed tokens individually.

Complete objective and optimization. For Open tasks, the nonzero budget weights are $\omega_d(1/9) = 1$, $\omega_d(1/3) = 0.4$, and $\omega_d(2/9) = 0.25$. For MC tasks they are $\omega_d(1/18) = 1$, $\omega_d(1/9) = 0.4$, and $\omega_d(1/3) = 0.2$. Only PCD uses the normalized multibudget combination:

$$\mathcal{D}_{\text{pcd}} = \frac{\sum_r \omega_d(r)\,|\mathcal{C}_r|^{-1} \sum_{c\in\mathcal{C}_r} \mathcal{D}^{c,r}_{\text{pcd}}}{\sum_r \omega_d(r)}. \tag{34}$$

Here $\mathcal{C}_{r_*}$ contains all five attack compressors; secondary budgets exclude DivPrune. Rank averages the five practical-budget scores, diversity averages DivPrune and VisPruner, and relation uses FlowCut. The attention and redundancy terms use their fixed targets above. The maximization form is

$$\max_{\|\delta\|_\infty \le \epsilon} \quad \mathcal{J}_{\text{InternVL}} = 6\,\mathcal{D}_{\text{pcd}} + 3\,(20\,\mathcal{D}_{\text{rank}}) + 2\,\mathcal{D}_{\text{div}} + 2\,\mathcal{D}_{\text{rel}} + 40\,\mathcal{D}_{\text{attn}} + 1.5\,\mathcal{D}_{\text{red}} - 0.05\,\mathcal{L}_{\text{sem}}. \tag{35}$$

These scores reverse the minimized implementation losses, up to additive constants for PCD, diversity, and redundancy. The single weighted objective is differentiated directly, without separately normalizing or projecting term gradients. Random-start PGD uses

$$\delta^{(t+1)} = \Pi_{[-\epsilon,\epsilon]}\left[\delta^{(t)} + \alpha\ \text{sign}\left(\nabla_\delta \mathcal{J}_{\text{InternVL}}\right)\right], \qquad x_a^{(t+1)} = \text{clip}(x + \delta^{(t+1)}, 0, 1), \tag{36}$$

with $\epsilon = 2/255$, $\alpha = 0.5/255$, and 100 steps. The random seed for zero-based sample index i is $42 + i$.

Additional retention levels. Table 15 supplements Table 6 with additional retention fractions applied uniformly across all four datasets. Damage is the Clean–FATA accuracy difference, and amplification is its increase relative to full-token inference. Both are calculated before rounding.

Table 15: InternVL3.5 accuracy and amplification at additional retention fractions. Each row applies the same fraction to all four datasets and averages over the four compressors.

Budget	Clean	FATA	Damage (pp)	Amplification (pp)
2/9	80.77	68.49	12.28	7.05
1/9	70.78	56.20	**14.58**	**9.34**
1/18	60.07	47.74	12.33	7.09
1/36	49.97	41.90	8.06	2.83

The Clean–FATA accuracy gap peaks at $1/9$ and narrows as retention falls to $1/36$.

Retained-set diagnostics. For either diagnostic m and a fixed compressor c, macro averaging first gives equal weight to valid images within each dataset, then equal weight to the four datasets:

$$\bar{m}_c = \frac{1}{4}\sum_{d=1}^{4}\left(\frac{1}{n_d}\sum_{i\in\mathcal{I}_d} m_{i,d,c}\right). \tag{37}$$

Here $\mathcal{I}_d$ is the valid-image set and $n_d = |\mathcal{I}_d|$. In the order TextVQA-Open, VQAv2-Open, ScienceQA-MC, and VQAv2-MC, the counts are $998, 1000, 1000, 1000$. No weighting by token count is used. For Table 16, tile-local selected indices are mapped into the concatenated image-wide token stream before computing each image's diagnostics. Both diagnostics use the final adversarial input at the practical budget ($1/9$ for Open and $1/18$ for MC).

Table 16: InternVL3.5 retained-set diagnostics at the practical budget. For each image, S_{clean} and S_{FATA} contain the selected source-token indices, using representative indices for merging methods. Flip rate is $|S_{\text{clean}} \setminus S_{\text{FATA}}|/|S_{\text{clean}}|$, the fraction of clean-selected tokens removed after attack; Jaccard is $|S_{\text{clean}} \cap S_{\text{FATA}}|/|S_{\text{clean}} \cup S_{\text{FATA}}|$. Values are fractions, macro-averaged by Eq. 37.

Compressor	Flip rate	Jaccard
VisionZIP	0.854	0.079
VisPruner	0.925	0.040
FlowCut	0.890	0.059
PruMerge	0.617	0.240

The retained-set statistics are consistent with substantial changes to the compression decision across the retained compressors, while the family-aware surrogates cover complementary attention-, diversity-, and relation-sensitive selection behavior. These diagnostics do not establish that any single surrogate is solely responsible for the task-level drop.

InternVL3.5 data integrity. The TextVQA-Open exclusions summarized in Appendix B arise from truncated source images 371 and 759, which are omitted consistently from Clean and FATA aggregation. The final formal validation reports no duplicate or missing conditions, no NaN or Inf values, and no perturbation-budget violations. The small set of parser status exceptions is inherited from the same fixed evaluation pipeline and is scored as zero credit without re-running inference.

H Detection Details and Cross-Attack Analysis

H.1 ML-ATD Construction

HiddenDetect (Jiang et al., 2025) characterizes jailbreak behavior through directions derived from refusal-related hidden-state changes. Inspired by HiddenDetect, ML-ATD adapts this approach by replacing the refusal-related reference with an attack direction estimated from Base examples and extending feature readout to the visual encoder, projector, and language model. Figure 10 shows reference fitting, feature extraction, trajectories within each stage, and the calculation of detection metrics. Each stage is evaluated in its own feature space.

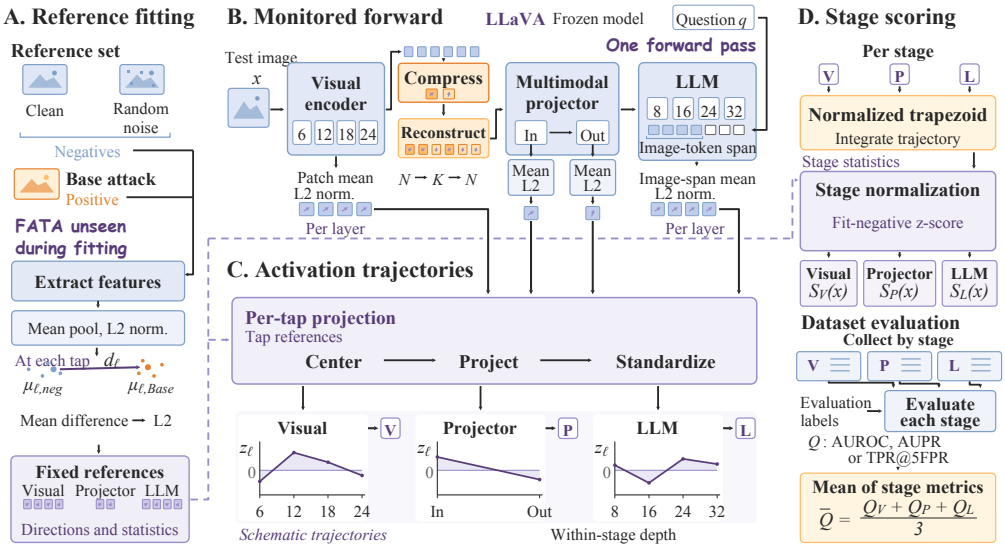


Figure 10: ML-ATD reference fitting, feature readout, and stage-wise detection. Visual, Projector, and LLM trajectories are scored and evaluated separately, with stage metrics averaged only for reporting.

Fixed per-tap references (A). Reference and test images use the same feature extraction procedure, with mean pooling followed by ℓ_2 normalization at each observation point, or tap, ℓ, giving $\widetilde{\mathbf{f}}_\ell(x)$. The negative references, denoted by `clean_clip+random_clip`, consist of clean images and images generated by adding random perturbations to clean images, without filtering by prediction correctness. Base adversarial examples provide the positive references. The mean feature vectors of the positive and negative references at each tap define the fixed direction

$$\mathbf{d}_\ell = \frac{\boldsymbol{\mu}_{\ell,\mathrm{Base}} - \boldsymbol{\mu}_{\ell,\mathrm{neg}}}{\|\boldsymbol{\mu}_{\ell,\mathrm{Base}} - \boldsymbol{\mu}_{\ell,\mathrm{neg}}\|_2}. \tag{38}$$

Reference fitting uses only the negative references and Base examples, excluding FATA examples.

Monitored feature readout (B). A single forward pass through the frozen LLaVA model provides visual patch features from layers 6, 12, 18, and 24, the projector input and output, and LLM activations from layers 8, 16, 24, and 32. Mean pooling covers all patches for visual features and only the image-token span for LLM features. The projector input and output each use mean pooling and ℓ_2 normalization, with separate reference directions and calibration statistics. In the compressed inference path shown in Figure 10, token selection or merging produces K representatives Z, which reconstruct an N-slot sequence $\widetilde{H}$ before the projector. The LLM receives the projected visual sequence together with the question.

Standardized projections (C). Each normalized feature is centered on its negative mean, projected onto its own attack direction, and standardized

$$u_\ell(x) = \big(\widetilde{\mathbf{f}}_\ell(x) - \boldsymbol{\mu}_{\ell,\mathrm{neg}}\big)^\top \mathbf{d}_\ell, \qquad z_\ell(x) = \frac{u_\ell(x) - m_\ell^-}{s_\ell^-}, \tag{39}$$

where m_ℓ^- and s_ℓ^- denote the mean and standard deviation of u_ℓ over the negative reference samples. For each stage $r \in \{V, P, L\}$, $\mathbf{z}_r(x)$ contains the standardized projections in layer order, with depth normalized to $[0, 1]$ within that stage. Schematic in panel C illustrates how these values vary across layers.

Stage scores and reported metrics (D). ML-ATD integrates the standardized projections over the normalized depth interval using the trapezoidal rule. The resulting area $T_r(x)$ gives the detection score for stage r after normalization by its mean a_r^- and standard deviation b_r^- over the negative reference samples,

$$T_r(x) = \mathrm{Trapz}_{[0,1]}\big(\mathbf{z}_r(x)\big), \qquad S_r(x) = \frac{T_r(x) - a_r^-}{b_r^-}. \tag{40}$$

Evaluation uses the scores $S_V(x)$, $S_P(x)$, and $S_L(x)$ separately to compute AUROC, AUPR, and TPR@5FPR against the attack and negative labels. For each stage, TPR@5FPR is obtained from its test-set ROC curve at a 5% false positive rate. For each metric Q, the reported value is the mean across the three stages,

$$\overline{Q} = \frac{Q_V + Q_P + Q_L}{3}. \tag{41}$$

FATA–Base comparison.

Table 17: Detection performance on FATA and Base. Higher values mean easier detection; green arrows show the relative reduction from Base to FATA.

Detector	Attack	AUROC	AUPR	TPR@5FPR
Feature Squeezing	FATA	**0.560** ↓15.8%	**0.382** ↓22.7%	**0.024** ↓62.5%
	Base	0.665	0.494	0.064
Mahalanobis-Max	FATA	**0.695** ↓14.5%	**0.509** ↓22.9%	**0.134** ↓51.8%
	Base	0.813	0.660	0.278
ML-ATD	FATA	**0.828** ↓13.0%	**0.724** ↓21.1%	**0.497** ↓37.9%
	Base	0.952	0.918	0.801

Table 17 isolates the FATA and Base rows of the main-text comparison in Table 8; the green arrows report relative reductions from Base to FATA. At a 5% false positive rate, FATA reduces TPR by 62.5% under Feature Squeezing (from 6.4% to 2.4%), by 51.8% under Mahalanobis-Max (from 27.8% to 13.4%), and by 37.9% under ML-ATD (from 80.1% to 49.7%, with the reduction computed before rounding). Thus, FATA is less detectable than Base across all three detectors at this operating point.

H.2 Detection Evaluation

Baseline settings and aggregation. The results in Tables 8 and 17 use two negatives per positive attack example. Feature Squeezing (FS) evaluates inputs at $K = 576$ using the unthresholded

MaxFS score (`max_fs_score_diff`), which measures changes in task credit against ground-truth answers before and after input transformations. Mahalanobis-Max uses the per-image maximum of standardized CLS and mean-pooled patch distances in CLIP space, with 128-dimensional PCA, clean fitting at indices 0–499, and evaluation at 500–999. Both baselines use seed 0, without attack-specific score selection.

FS averages 16 dataset–compressor combinations with identical full-token scores across compressors, making this equivalent to averaging four datasets. Mahalanobis-Max combines matching records across compressors before averaging four datasets. ML-ATD reports an equal-weight mean over four compressors (VisionZIP, VisPruner, PruMerge, and FlowCut), four datasets, and the three stages. Its detection runs use $K = 64$ for open-ended tasks and $K = 32$ for multiple-choice tasks.

Input consistency versus feature deviation. The four-attack comparison in Table 8 shows that detection performance depends on the type of deviation each detector measures. In particular, Mahalanobis-Max orders CAGE $>$ CAA $>$ Base $>$ FATA on all three metrics. Although FS gives similar detection performance for FATA and CAA, Mahalanobis-Max detects FATA less readily. This lower detectability in feature space is consistent with FATA's semantic preservation objective.

Evaluation sample coverage. CAA evaluation on ScienceQA includes 990 attack examples for FS and 496 for Mahalanobis-Max, with 1,980 and 992 aligned negatives, respectively. Other conditions use nominal counts.