PROFILING COMPRESSION, DISRUPTING DISTRIBUTIONS: STRUCTURED ADVERSARIAL ATTACKS ON TOKEN-COMPRESSED LVLMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Visual token compression offers a promising pathway to improve the efficiency of large vision-language models (LVLMs). However, it creates a fundamental mismatch for adversarial attacks: compression acts on token representations, whereas attack success is evaluated at the sequence level. Existing attacks typically identify tokens likely to survive compression and optimize perturbations over this predicted survivor set. Yet this strategy overlooks two challenges: *token utility varies across compression schemes*, and *independent survival scores cannot capture the joint attack effect of retained tokens*. To address these challenges, we propose Compression-Aware Profiling with Coordinated Distributional Disruption (`CLOUD`), a novel framework for conducting adversarial attacks that remain effective across distinct token compression schemes. The key idea is to profile compression-dependent attack utility with a surrogate pool and then construct coordinated attack support through distributional perturbation. Specifically, we first introduce a utility-guided surrogate probing module, which applies Bayesian optimization over the surrogate configuration space to characterize compression-aware attack responses. Subsequently, we construct a dynamic cache to aggregate token-retention patterns and representation shifts across surrogates to calibrate the attack budget. Guided by these profiles, we leverage a determinantal point process (DPP) to construct a diverse support set by balancing compression resilience with complementary evidence coverage. Additionally, we maximize the discrepancy between clean evidence and its post-compression adversarial counterpart from the perspective of optimal transport (OT). This coordinated objective encourages the attack effect to persist after token reduction. Experiments across multiple LVLM backbones and compression schemes show that `CLOUD` consistently outperforms strong baselines. Code is available at https://anonymous.4open.science/r/CLOUD_For_Token_Compression_Robustness.

1 INTRODUCTION

The emergence of large vision-language models (LVLMs) (Chen et al., 2024c; Bai et al., 2025; Li et al., 2025a; Liu et al., 2023; Li et al., 2023; Alayrac et al., 2022) has driven substantial advances in visual understanding (Caffagni et al., 2024; Peng et al., 2024; Wang et al., 2024a; Ye et al., 2024), ranging from visual question answering (Guo et al., 2023; Gu et al., 2026) to complex visual reasoning (Xu et al., 2025; Tang et al., 2026). However, the visual backbones of current LVLMs represent each image using hundreds or even thousands of tokens (Bulat et al., 2026), resulting in considerable computational overhead that restricts their deployment in resource-constrained scenarios (Zhang et al., 2026a). To address this challenge, visual token compression methods (Yang et al., 2025c; Tong et al., 2026; Alvar et al., 2025) have been proposed to compress full-length visual sequences within limited token budgets while preserving as much information as possible according to diverse criteria. Crucially, visual token compression (Liao et al., 2026) not only shortens the visual sequence but also determines which visual evidence is ultimately exposed to the language model (Lei et al., 2026; Zhang et al., 2026d; Huang et al., 2026), serving as an input-dependent evidence bottleneck rather than a purely computational operator.

Adversarial perturbations (Zhang et al., 2026b;c) alter visual token representations before compression (Wang et al., 2024b), disturbing final decisions. This interaction creates a fundamental mis-

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.

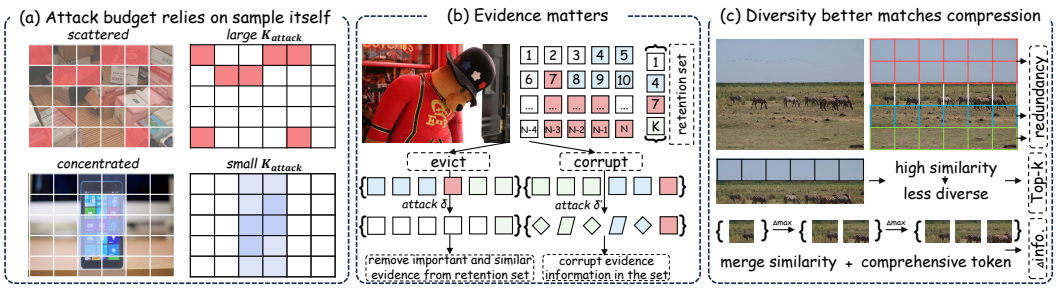


Figure 1: **Motivating insights for compression-aware attacks.** (a) Attack budgets should adapt to sample-specific evidence. (b) Evidence can be disrupted by token eviction or corruption. (c) Diversity-aware selection better reflects compression than attention-only selection.

match between attack optimization over pre-compression tokens and attack efficacy on the resulting compressed sequence. To improve attack effectiveness under token compression, CAGE (Zhang et al., 2026c) first prioritizes tokens that are most likely to survive compression and then aligns their perturbation magnitudes with the corresponding ranking scores to encourage their retention.

As illustrated in Fig. 1, existing compression-aware attacks primarily prioritize tokens according to their survival probabilities. However, attack utility varies with both input and compression scheme, since evidence distributions require different budgets and disruption may come from evicting critical tokens or corrupting retained evidence. This raises our first research question: ❶ *How can we characterize compression-dependent attack utility and adapt the attack budget to each instance?* Furthermore, independent token ranking often selects redundant tokens and misses the collective post-compression effects of the selected tokens. This leads to our second research question: ❷ *How can we construct a coordinated token set whose joint perturbation disrupts post-compression evidence?* Accordingly, we argue that compression-aware attacks should move beyond survival prediction to optimize support and representation distribution of perturbed post-compression evidence.

To address these challenges, we introduce a unified framework for compression-aware LVLM attacks, termed Compression-Aware Profiling with Coordinated Distributional Disruption (`CLOUD`). Specifically, we first build a surrogate pool to characterize compression-dependent attack responses, identifying informative configurations via Bayesian optimization. Building upon utility-guided probing, we construct a dynamic cache that aggregates token-retention patterns and representation shifts across surrogates, thereby enabling instance-adaptive attack-budget calibration. To balance compression resilience and evidence coverage, we employ a DPP to construct a coordinated attack support and optimize its perturbations via optimal transport to maximize the distributional discrepancy from clean visual evidence. Empirically, extensive experiments on two LVLM backbones and five compression frameworks show that `CLOUD` consistently outperforms baselines across budgets. The main contributions of the proposed `CLOUD` are summarized as follows:

❶ ***New Perspective.*** We view visual token compression as an input-dependent evidence bottleneck for adversarial perturbations on LVLMs. Beyond token-wise survival estimation, we characterize both *cross-scheme token utility* and *the joint attack effect of retained tokens*.

❷ ***Unified Methodology with Theoretical Guarantee.*** `CLOUD` profiles compression-dependent attack utility with a surrogate pool and coordinates attack support through DPP selection and transport optimization. We establish guarantees for stable adaptive budget estimation and the approximation quality of determinantal support construction.

❸ ***Comprehensive Evaluation.*** We evaluate `CLOUD` across four LVLM backbones, five token-compression schemes, three benchmarks, and multiple budgets, showing that it is (i) **high-performing**, reducing average robust accuracy over CAGE by up to 9.3 points, and (ii) **model-generalizable**, consistently improving across different LVLMs and compression settings.

2 Related Work

Visual Token Compression. Visual token compression in LVLMs (Dai et al., 2023; Zhu et al., 2024; Fu et al., 2026; Zhan et al., 2026) reduces computational overhead while balancing efficiency and accuracy (Liao et al., 2026; Chen et al., 2024a; Huang et al., 2024; Li et al., 2025b). Existing methods fall into three groups: importance-based selection (Tong et al., 2026; Ye et al., 2025),

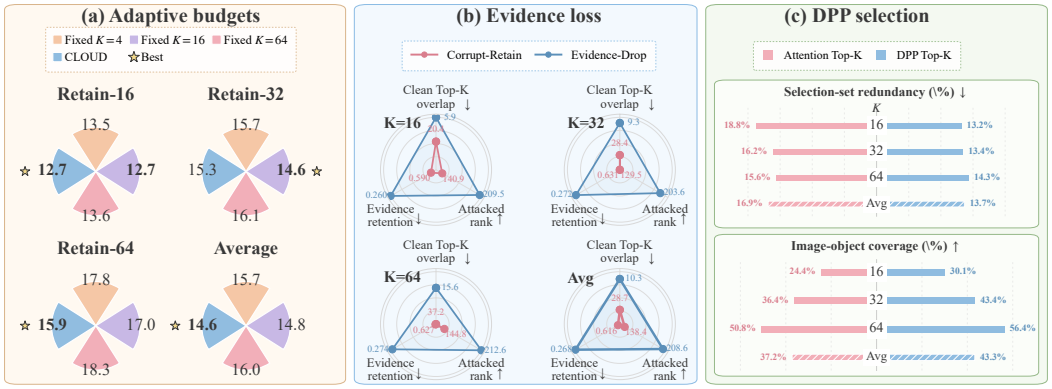


Figure 2: Empirical Observation Validation. (a) Comparison of fixed and sample-adaptive attack budgets under different retention sizes. (b) Evaluation of two different routes for reducing retained evidence. (c) Comparison of Attention Top-K and DPP Top-K in redundancy and coverage.

redundancy-aware merging (Bulat et al., 2026; Bolya et al., 2022; Yang et al., 2025b), and diversity-aware compression (Cho et al., 2026). FastV (Chen et al., 2024b) and SparseVLM (Zhang et al., 2025c; 2026d) identify important visual tokens via cross-modal attention, whereas VisionZip (Yang et al., 2025c) and DivPrune (Alvar et al., 2025) preserve informative and diverse content. Recent work further explores optimization-based, progressive, information-flow-guided, and adaptive pruning strategies (Huang et al., 2025; Yang et al., 2025a; Xing et al., 2025; Yang et al., 2025d; Suo et al., 2025). Despite their progress, these methods induce heterogeneous transformations of visual evidence, thereby introducing uncertainty into pre-compression attack optimization.

Adversarial Attack on LVLMs. The objective of a visual adversarial attack (Zhang et al., 2025a; Nie et al., 2026; Cao et al., 2026; Cui et al., 2024) is to introduce a bounded, visually imperceptible perturbation to visual tokens, thereby degrading the performance of LVLMs. Existing attacks can be categorized into output-oriented methods (Xie et al., 2025; Hu et al., 2026; Bailey et al., 2023; Hao et al., 2024), which optimize task predictions or target responses, and representation-oriented methods (Jia et al., 2026; Mei et al., 2026), which disrupt visual features or token relations to improve transferability across downstream models. However, these methods typically optimize perturbations over the full pre-compression token sequence and assume that the visual representations are directly consumed by the language model (Cai et al., 2026; Nie et al., 2026), thereby overlooking how token compression removes, merges, or transforms adversarial evidence before generation.

3 Methodology

Problem Formulation. Given an image $\mathbf{x}_v$, the visual encoder $\mathcal{E}$ produces clean visual tokens $\mathbf{H}^0 = \mathcal{E}(\mathbf{x}_v) = [\mathbf{h}_1^0, \dots, \mathbf{h}_N^0] \in \mathbb{R}^{N \times D}$. An adversarial perturbation $\boldsymbol{\delta}$ generates $\mathbf{H}^\delta = \mathcal{E}(\mathbf{x}_v + \boldsymbol{\delta})$ under $\|\boldsymbol{\delta}\|_\infty \leq \epsilon$. The target compressor $\mathcal{C}^*$ with retention budget K_{model} outputs $\mathbf{Z}^0 = \mathcal{C}^*(\mathbf{H}^0; K_{\text{model}})$ and $\mathbf{Z}^\delta = \mathcal{C}^*(\mathbf{H}^\delta; K_{\text{model}})$. With access to $\mathcal{C}^*$, the optimal perturbation maximizes the discrepancy between clean and adversarial compressed representations:

$$\boldsymbol{\delta}^* = \arg \max_{\|\boldsymbol{\delta}\|_\infty \leq \epsilon} \mathcal{D}\left(\mathbf{Z}^0, \mathbf{Z}^\delta\right), \tag{1}$$

where $\mathcal{D}$ measures representation discrepancy. In practice, the target compressor $\mathcal{C}^*$ and its retention budget K_{model} are unavailable. Therefore, our objective is to infer compression-sensitive evidence from surrogate responses and construct perturbations that disrupt preserved evidence.

3.1 Empirical Observation Study and Insights

To investigate how attacks adapt to visual token compression, we conduct a study on LLaVA (Liu et al., 2024) with TextVQA (Singh et al., 2019) and VisionZip (Yang et al., 2025c) under retention budgets of 16, 32, and 64 tokens. The results reveal three observations that motivate our design:

❶ ***Is the effective attack budget solely determined by the compression budget?*** As shown in Fig. 2(a), adaptive allocation lowers the average accuracy by 0.9 compared with the mean result of the fixed budgets $K_{\text{attack}} \in \{4, 16, 64\}$. The variation among fixed budgets indicates that attack requirements differ across samples, motivating sample-specific budget allocation.

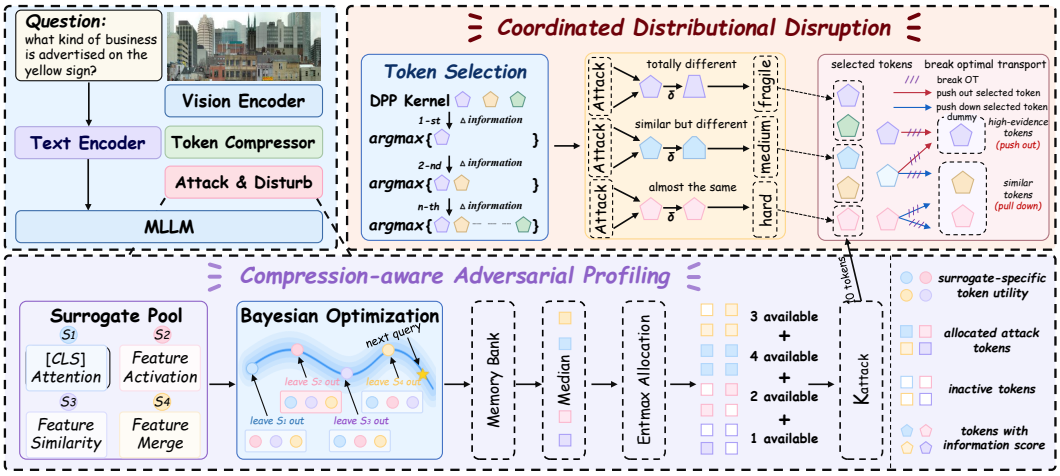


Figure 3: Overview of our framework. We profile surrogates and aggregate token utilities to determine K_{attack}. DPP then selects tokens by marginal information gain. Finally, optimal transport uses adaptive and dummy costs to push out high-evidence tokens and pull down similar ones.

> ***Insight I:*** *The effective attack budget is instance-dependent and adapts to each input.*

❷ ***Must strongly perturbed tokens remain in the retained set?*** As illustrated in Fig. 2(b), compared with the corrupt-retain objective, our attack reduces the average Top-k overlap between clean and adversarial tokens by 18.4, while lowering clean token ranks and increasing representation changes. This shows that effective attacks should account for post-compression evidence loss, where important evidence can be removed or distorted after retention.

> ***Insight II:*** *Post-compression evidence loss matters more than preserving strongly perturbed tokens in the retained set or only corrupting the tokens.*

❸ ***Does attention Top-k provide a representative attack set?*** Fig. 2(c) shows attention-based selection contains more redundant tokens, whereas DPP selects more diverse tokens and provides broader coverage. Across these three budgets, DPP reduces average pairwise redundancy by 3.2. Thus, attack-token selection should consider diversity rather than attention magnitude alone.

> ***Insight III:*** *Diversity-aware selection provides a more representative attack set with lower redundancy and broader evidence coverage.*

3.2 Framework Overview

Compression-aware attacks face a mismatch between full-sequence perturbations and effects on post-compression evidence. As illustrated in Fig. 3, `CLOUD` couples compression-aware profiling with coordinated distributional disruption. We build complementary surrogates for attention, similarity, activation, and merging, calibrated via leave-one-surrogate-out Bayesian optimization for transferable compression responses. A dynamic memory aggregates retention and representation changes, while sparse allocation converts cross-surrogate consensus into an instance-specific budget, identifying attack-consistent evidence and disruption amount. The disruption stage uses DPP selection to balance relevance and diversity under this budget, while evidence transport enlarges the discrepancy between clean and post-compression adversarial representations. These stages disrupt complementary evidence and reduce LVLM-recoverable clean visual information after compression.

3.3 Compression-Aware Adversarial Profiling

Different compression schemes preserve visual evidence according to different retention criteria (Shang et al., 2025), causing the influential token set to vary across compression behaviors. The unknown K_{model} further changes the amount of evidence available after compression, while the effective attack support also varies across samples (Rao et al., 2021). We therefore profile each input with compression surrogates and aggregate their responses to infer its attack budget.

Utility-Guided Surrogate Probing. We construct a surrogate pool $\mathcal{S}$ containing attention, similarity, activation, and merge surrogates, which capture complementary retention behaviors related

to relevance, redundancy, feature strength, and aggregation. Specifically, the attention surrogate uses the CLS-to-patch attention score as ϕ_i^{att}, the activation surrogate measures feature magnitude as $\phi_i^{act} = \|\mathbf{h}_i\|_2$, and the similarity surrogate captures redundancy through aggregated pairwise similarity as $\phi_i^{sim} = \sum_{j \neq i} \mathrm{sim}(\mathbf{h}_i, \mathbf{h}_j)$. The merge surrogate simulates soft feature aggregation while preserving token correspondence through original token indices, producing an aggregation-aware response ϕ_i^{merge}. Each surrogate $s \in \mathcal{S}$ produces a compression response $\phi^s(\mathbf{H}; \theta_s)$ under configuration θ_s. The surrogate response and normalized score of token i are computed as:

$$e_i^s = \left\| \phi_i^s(\mathbf{H}^\delta; \theta_s) - \phi_i^s(\mathbf{H}^0; \theta_s) \right\|_2^2, \qquad v_i^s = \mathcal{R}(e_i^s), \tag{2}$$

where $\mathcal{R}(\cdot)$ converts responses into percentile scores for cross-surrogate comparison. Before profiling, we calibrate surrogate configurations through leave-one-surrogate-out Bayesian optimization on a calibration subset. The configuration θ contains parameters, including attention sharpness, similarity temperature, activation scaling, merge temperature, memory smoothing, and allocation temperature. For each held-out surrogate s, perturbations generated from remaining surrogates are evaluated according to compression-sensitive disruption. The calibration objective is computed as:

$$\mathcal{J}_{\mathrm{BO}}(\theta) = \frac{1}{|\mathcal{X}_{\mathrm{cal}}| \cdot |\mathcal{S}|} \sum_{x \in \mathcal{X}_{\mathrm{cal}}} \sum_{s \in \mathcal{S}} \frac{\mathcal{M}_s\left(x, \boldsymbol{\delta}_\theta^{(-s)}, q_\theta^{(-s)}\right) - \mathcal{M}_s\left(x, \boldsymbol{\delta}_0, q_\theta^{(-s)}\right)}{\left|\mathcal{M}_s\left(x, \boldsymbol{\delta}_0, q_\theta^{(-s)}\right)\right| + \eta}, \tag{3}$$

where $\mathcal{M}_s(x, \boldsymbol{\delta}, q)$ denotes the compression-sensitive disruption under input x, and η is a small constant for numerical stability. Bayesian optimization models $\mathcal{J}_{\mathrm{BO}}$ with an RBF kernel and selects candidates using expected improvement, yielding shared configuration θ^*. The obtained configuration is shared across samples, while only the allocation $\mathbf{q}$ is adapted during instance-level profiling.

Cross-Surrogate Budget Calibration. The calibrated surrogates provide complementary token responses for the current sample. During iterative profiling, surrogate branches are sequentially probed, and a dynamic memory cache accumulates their responses through exponential smoothing. The cache update and consensus utility are computed as:

$$m_{s_t,i}^{(t)} = \beta m_{s_t,i}^{(t-1)} + (1-\beta) v_i^{s_t}, \qquad u_i^{(t)} = \mathrm{Median}_{s \in \mathcal{S}_t}\, m_{s,i}^{(t)}, \tag{4}$$

where s_t denotes the surrogate probed at iteration t, and $\mathcal{S}_t$ contains surrogates observed up to iteration t. Median aggregation identifies token responses consistent across compression behaviors. The sparse allocation is obtained as $q^{(t)} = \mathrm{Entmax}_\alpha(u^{(t)}/\tau_a)$, where Entmax produces sparse weights through a thresholded power transformation, and active support determines $K_{\mathrm{attack}}^{(t)} = |\{i \mid q_i^{(t)} > 0\}|$. This allocation concentrates perturbations on tokens supported by cross-surrogate consensus. The profiling objective is computed as:

$$\mathcal{L}_{\mathrm{profile}}^{(t)} = \sum_{i=1}^{N} \mathrm{stopgrad}\left(q_i^{(t)}\right) \left[1 - \cos\left(\mathbf{h}_i^\delta, \mathbf{h}_i^0\right)\right], \tag{5}$$

where $\mathrm{stopgrad}(\cdot)$ blocks gradients through $q_i^{(t)}$. The perturbation is updated to initialize subsequent coordinated distributional disruption, while the final allocation q^{S1} is retained for transport.

3.4 Coordinated Distributional Disruption

Tokens retained after compression jointly represent the visual input, so independently selected attack tokens may cover redundant evidence (Yang et al., 2025b; Kulesza & Taskar, 2012). We therefore construct a diverse support through a determinantal criterion and further disrupt the correspondence between clean evidence and surviving adversarial representations through evidence transport.

Determinantal Support Construction. The calibrated K_{attack} specifies the number of tokens allocated for coordinated disruption, while token selection further accounts for interactions within the support. Given adversarial features $\mathbf{H}^\delta$, we normalize each token as $\mathbf{z}_i = \mathbf{h}_i^\delta / \|\mathbf{h}_i^\delta\|_2$ and denote its normalized relevance by $a_i \in [0, 1]$. The determinantal kernel is computed as:

$$G_{ij} = a_i \langle \mathbf{z}_i, \mathbf{z}_j \rangle a_j + \xi \cdot \mathbb{I}[i = j], \tag{6}$$

where ξ is a numerical regularizer. Greedy marginal maximization is then applied to the shifted determinantal objective $F(S) = \log\det(\mathbf{I} + \mathbf{G}_S)$, producing an ordered token sequence that balances relevance and diversity. Let r_i denote the entry rank of token i. The binary support weight

is defined as $\omega_i = \mathbb{I}[r_i \leq K_{\text{attack}}]$, where tokens within the DPP (Nemhauser et al., 1978) prefix receive attack mass under the sample-specific budget. We then maximize the determinantal feature divergence over the selected support. With $d_i = \frac{1}{2}[1 - \cos(\mathbf{h}_i^{\delta}, \mathbf{h}_i^{0})]$, the objective is defined as:

$$\mathcal{L}_{\text{DOFD}} = \frac{\sum_{i=1}^{N} \omega_i d_i}{\sum_{i=1}^{N} \omega_i + \eta}. \tag{7}$$

Transport-Based Perturbation Optimization. Although the determinantal support captures complementary adversarial evidence, clean information may remain recoverable through other surviving representations. We therefore model its correspondence with adversarial evidence through partial transport (Cuturi, 2013) using the normalized Entmax allocation as source mass, where $\mu_i = q_i^{\text{S1}}$ and $\sum_i \mu_i = 1$. For clean token i and adversarial token j, the transport cost is computed as:

$$C_{ij} = \frac{1}{2}\left[1 - \cos\left(\mathbf{h}_i^{0}, \mathbf{h}_j^{\delta}\right)\right]. \tag{8}$$

The binary support weights define the target mass with $m = \frac{1}{N}\sum_{j=1}^{N} \omega_j = K_{\text{attack}}/N$ and $\nu_j = \omega_j/N$. The remaining mass $1 - m$ is assigned to a zero-cost dummy target, yielding $\tilde{\nu} = [\nu_1, \ldots, \nu_N, 1 - m]$. Appending a zero-cost dummy column to $\mathbf{C}$ gives the augmented cost matrix $\widetilde{\mathbf{C}}$, and the transport plan is computed as:

$$\Gamma^{*} = \arg\min_{\Gamma \in \Pi(\mu, \tilde{\nu})} \left\langle \Gamma, \widetilde{\mathbf{C}} \right\rangle + \tau_{\text{OT}} \sum_{i,j} \Gamma_{ij}(\log \Gamma_{ij} - 1), \tag{9}$$

where $\Pi(\mu, \tilde{\nu})$ denotes the transport polytope satisfying $\Gamma \mathbf{1} = \mu$ and $\Gamma^{\top}\mathbf{1} = \tilde{\nu}$. We then maximize transport discrepancy over adversarial targets, encouraging perturbed representations to deviate from profiled evidence under the surrogate distribution. The transport disruption objective is:

$$\mathcal{L}_{\text{POT}} = \frac{1}{m + \eta} \sum_{i=1}^{N} \sum_{j=1}^{N} \text{stopgrad}\left(\Gamma_{ij}^{*}\right) C_{ij}. \tag{10}$$

Combining these terms, the overall objective of `CLOUD` is defined as:

$$\mathcal{L} = \lambda_1 \mathcal{L}_{\text{DOFD}} + \lambda_2 \mathcal{L}_{\text{POT}}. \tag{11}$$

Maximizing $\mathcal{L}$ increases distortion over the determinantal support and transport discrepancy between clean evidence and surrogate post-compression adversarial evidence under the perturbation budget.

3.5 Summarization of `CLOUD`

We first calibrate the surrogate configurations on $\mathcal{X}_{\text{cal}}$ through leave-one-surrogate-out Bayesian optimization to align complementary surrogate responses. During profiling, surrogate responses update the memory, while median aggregation and Entmax convert cross-surrogate consensus into the sparse allocation $\mathbf{q}^{(t)}$ and the adaptive budget $K_{\text{attack}}^{(t)}$, estimating sample-specific attack requirements. The final K_{attack} determines the diverse DPP support and its binary weights $\boldsymbol{\omega}$, while $\mathbf{q}^{\text{S1}}$ provides source mass for transport optimization. During optimization, $\mathcal{L}_{\text{DOFD}}$ disrupts the selected adversarial evidence, while $\mathcal{L}_{\text{POT}}$ increases the transport discrepancy between clean evidence and adversarial survivors, jointly reducing recoverable clean information after compression. The procedure is provided in Algorithm 1 in Appendix A.2.

3.6 Theoretical Guarantees

We characterize the stability of cross-surrogate aggregation for adaptive attack-budget estimation and the approximation quality of determinantal selection for constructing diverse attack supports. Proofs are provided in Appendix B.

Definition 3.1. Let $m_{s,i}$ denote the final memory of token i under surrogate s, and let $u_i^{\star}$ denote its ideal cross-compression utility. Set $u_i = \text{Median}_{s \in \mathcal{S}}\, m_{s,i}$, $z_i = u_i/\tau_a$, $z_i^{\star} = u_i^{\star}/\tau_a$, and $\rho = \varepsilon_m/\tau_a$. For $1 < \alpha \leq 2$, let $\gamma_{\alpha}(\mathbf{z})$ denote the α-Entmax support threshold in $\mathbf{z}$-space, and set $\gamma^{\star} = \gamma_{\alpha}(\mathbf{z}^{\star})$. The lower and upper support sizes are defined as:

$$K^{-} = \sum_{i=1}^{N} \mathbb{I}\left[z_i^{\star} > \gamma^{\star} + 2\rho\right], \qquad K^{+} = \sum_{i=1}^{N} \mathbb{I}\left[z_i^{\star} > \gamma^{\star} - 2\rho\right]. \tag{12}$$

Table 1: Attack performance (%) of different visual token compression methods on LLaVA across GQA, TextVQA, and SQA. We report clean and robust accuracy under Base, CAGE, and CLOUD. Green annotations and red annotations indicate the absolute decrease and increase from the corresponding clean accuracy.

Compression	GQA				TextVQA				SQA			
Method	Clean	Base	CAGE	Ours	Clean	Base	CAGE	Ours	Clean	Base	CAGE	Ours
Retain 64 Tokens ($K_{\text{model}} = 64$)												
VisionZip	63.1	43.9↓ 19.2	43.2↓ 19.9	42.5↓ 20.6	53.8	31.3↓ 22.5	18.5↓ 35.3	15.9↓ 37.9	67.9	61.8↓ 6.1	63.8↓ 4.1	61.0↓ 6.9
VisPruner	62.4	44.0↓ 18.4	43.4↓ 19.0	42.3↓ 20.1	54.7	32.5↓ 22.2	20.0↓ 34.7	17.3↓ 37.4	68.2	62.1↓ 6.1	62.2↓ 6.0	61.3↓ 6.9
DivPrune	64.5	46.5↓ 18.0	46.1↓ 18.4	45.5↓ 19.0	51.2	29.9↓ 21.3	22.4↓ 28.8	18.0↓ 33.2	67.9	64.3↓ 3.6	61.1↓ 6.8	61.9↓ 6.0
FlowCut	62.4	45.5↓ 16.9	42.7↓ 19.7	44.1↓ 18.3	53.8	32.3↓ 21.5	21.5↓ 32.3	20.2↓ 33.6	68.5	62.6↓ 5.9	62.4↓ 6.1	62.3↓ 6.2
PruMerge	68.0	51.2↓ 16.8	50.2↓ 17.8	49.0↓ 19.0	49.5	31.3↓ 18.2	19.4↓ 30.1	17.0↓ 32.5	68.0	63.2↓ 4.8	63.0↓ 5.0	61.6↓ 6.4
Average	64.1	46.2↓ 17.9	45.1↓ 19.0	**44.7**↓ 19.4	52.6	31.5↓ 21.1	20.4↓ 32.2	**17.7**↓ 34.9	68.1	62.8↓ 5.3	62.5↓ 5.6	**61.6**↓ 6.5
Retain 32 Tokens ($K_{\text{model}} = 32$)												
VisionZip	58.9	41.9↓ 17.0	41.9↓ 17.0	39.0↓ 19.9	48.1	29.9↓ 18.2	16.8↓ 31.3	15.3↓ 32.8	68.4	62.9↓ 5.5	63.4↓ 5.0	62.0↓ 6.4
VisPruner	61.0	43.4↓ 17.6	40.0↓ 21.0	39.9↓ 21.1	50.8	32.6↓ 18.2	19.0↓ 31.8	15.4↓ 35.4	68.3	63.4↓ 4.9	63.3↓ 5.0	61.3↓ 7.0
DivPrune	62.4	45.4↓ 17.0	43.8↓ 18.6	42.9↓ 19.5	46.5	28.0↓ 18.5	19.7↓ 26.8	17.0↓ 29.5	68.2	63.9↓ 4.3	63.8↓ 4.4	61.8↓ 6.4
FlowCut	57.3	42.8↓ 14.5	41.5↓ 15.8	40.1↓ 17.2	49.7	30.3↓ 19.4	20.4↓ 29.3	17.1↓ 32.6	69.0	62.6↓ 6.4	63.2↓ 5.8	63.2↓ 5.8
PruMerge	63.8	50.1↓ 13.7	48.3↓ 15.5	48.0↓ 15.8	45.3	28.7↓ 16.6	18.8↓ 26.5	15.3↓ 30.0	69.3	63.7↓ 5.6	63.7↓ 5.6	62.6↓ 6.7
Average	60.7	44.7↓ 16.0	43.1↓ 17.6	**42.0**↓ 18.7	48.1	29.9↓ 18.2	18.9↓ 29.2	**16.0**↓ 32.1	68.8	63.3↓ 5.5	63.5↓ 5.3	**62.2**↓ 6.6
Retain 16 Tokens ($K_{\text{model}} = 16$)												
VisionZip	53.3	41.6↓ 11.7	40.6↓ 12.7	39.0↓ 14.3	43.8	26.7↓ 17.1	16.1↓ 27.7	12.7↓ 31.1	68.4	63.6↓ 4.8	64.1↓ 4.3	62.0↓ 6.4
VisPruner	54.0	42.2↓ 11.8	41.6↓ 12.4	40.5↓ 13.5	43.2	27.5↓ 15.7	16.9↓ 26.3	12.8↓ 30.4	67.4	63.3↓ 4.1	64.0↓ 3.4	61.5↓ 5.9
DivPrune	58.4	44.0↓ 14.4	43.2↓ 15.2	42.5↓ 15.9	37.5	24.1↓ 13.4	17.0↓ 20.5	14.1↓ 23.4	68.1	64.0↓ 4.1	63.7↓ 4.4	64.3↓ 3.8
FlowCut	50.4	42.3↓ 8.1	41.5↓ 8.9	39.3↓ 11.1	36.4	25.0↓ 11.4	16.7↓ 19.7	15.2↓ 21.2	67.0	62.5↓ 4.5	63.8↓ 3.2	63.6↓ 3.4
PruMerge	59.1	50.0↓ 9.1	48.5↓ 10.6	47.6↓ 11.5	37.6	26.6↓ 11.0	18.9↓ 18.7	15.5↓ 22.1	66.4	63.7↓ 2.7	62.7↓ 3.7	63.2↓ 3.2
Average	55.0	44.0↓ 11.0	43.1↓ 11.9	**41.8**↓ 13.2	39.7	26.0↓ 13.7	17.1↓ 22.6	**14.1**↓ 25.6	67.5	63.4↓ 4.1	63.7↓ 3.8	**62.9**↓ 4.6
Blind ($K_{\text{model}} = 0$)												
Blind	41.8				13.5				63.1			

Assumption 3.1. For every token i, more than half of the surrogate memories approximate $u_i^\star$ within ε_m. This condition is written as:

$$|\{s \in \mathcal{S} \mid |m_{s,i} - u_i^\star| \le \varepsilon_m\}| > \frac{|\mathcal{S}|}{2}. \tag{13}$$

Theorem 3.1. *Under Assumption 3.1, the support size produced by adversarial profiling satisfies:*

$$K^- \le K_{\text{attack}} \le K^+. \tag{14}$$

Remark 3.1. *Theorem 3.1 shows that only tokens whose ideal normalized utilities lie near the Entmax threshold can change their support membership. As ε_m decreases, the interval $[K^-, K^+]$ contracts, yielding a more stable estimate of the sample-specific attack budget.*

We next analyze the approximation quality of the determinantal support induced by $\mathbf{G}$ in Eq. 6.

Definition 3.2. For a token subset $S \subseteq [N]$, the determinantal set utility is defined as:

$$F(S) = \log\det\left(\mathbf{I} + \mathbf{G}_S\right). \tag{15}$$

Let $S_{K_{\text{attack}}} = \{i \mid r_i \le K_{\text{attack}}\}$ denote the support produced by greedy marginal maximization of F under the adaptive attack budget. The optimal support under the same budget is defined as:

$$S^\star_{K_{\text{attack}}} = \arg\max_{S \subseteq [N],\, |S| \le K_{\text{attack}}} F(S). \tag{16}$$

Theorem 3.2. *For the kernel in Eq. 6 with $\xi > 0$, the set utility F is normalized, monotone, and submodular. Given the adaptive attack budget K_{attack}, the greedy support satisfies:*

$$F(S_{K_{\text{attack}}}) \ge \left(1 - \frac{1}{e}\right) F(S^\star_{K_{\text{attack}}}). \tag{17}$$

Remark 3.2. *Theorem 3.2 guarantees that the greedy support constructed under K_{attack} achieves a constant-factor approximation to the optimal determinantal support with the same budget.*

4 Experiments

4.1 Experimental Setup

Benchmark Datasets. We evaluate our method on three widely used visual question answering benchmarks: GQA (Hudson & Manning, 2019), TextVQA (Singh et al., 2019), and ScienceQA

Table 2: Attack performance (%) of different visual token compression methods on Qwen3-VL across GQA, TextVQA, and SQA. We report clean and robust accuracy under Base, CAGE, and CLOUD.

Compression	GQA				TextVQA				SQA			
Method	Clean	Base	CAGE	Ours	Clean	Base	CAGE	Ours	Clean	Base	CAGE	Ours
Retain 72 Tokens ($K_{\text{model}} = 72$)												
VisionZip	71.0	65.8↓ 5.2	65.6↓ 5.4	58.7↓ 12.3	74.0	70.2↓ 3.8	70.4↓ 3.6	57.1↓ 16.9	91.7	88.1↓ 3.6	87.6↓ 4.1	87.4↓ 4.3
VisPruner	70.1	64.8↓ 5.3	65.1↓ 5.0	58.0↓ 12.1	65.8	63.8↓ 2.0	62.9↓ 2.9	52.8↓ 13.0	91.1	88.0↓ 3.1	88.1↓ 3.0	87.7↓ 3.4
DivPrune	70.3	64.7↓ 5.6	65.1↓ 5.2	58.1↓ 12.2	66.0	64.0↓ 2.0	62.9↓ 3.1	53.0↓ 13.0	91.1	87.9↓ 3.2	87.9↓ 3.2	87.6↓ 3.5
FlowCut	63.2	58.3↓ 4.9	57.9↓ 5.3	50.5↓ 12.7	52.8	51.1↓ 1.7	52.7↓ 0.1	44.4↓ 8.4	91.0	87.8↓ 3.2	87.7↓ 3.3	87.6↓ 3.4
PruMerge	37.7	39.1↑ 1.4	37.8↑ 0.1	33.6↓ 4.1	13.6	14.4↑ 0.8	14.1↑ 0.5	12.3↓ 1.3	76.0	74.9↓ 1.1	75.5↓ 0.5	74.4↓ 1.6
Average	62.5	58.5↓ 4.0	58.3↓ 4.2	**51.8**↓ 10.7	54.4	52.7↓ 1.7	52.6↓ 1.8	**43.9**↓ 10.5	88.2	85.3↓ 2.9	85.4↓ 2.8	**84.9**↓ 3.3
Retain 36 Tokens ($K_{\text{model}} = 36$)												
VisionZip	67.2	63.7↓ 3.5	62.7↓ 4.5	54.0↓ 13.2	58.5	55.4↓ 3.1	55.0↓ 3.5	40.7↓ 17.8	89.5	84.8↓ 4.7	85.4↓ 4.1	84.5↓ 5.0
VisPruner	66.8	63.5↓ 3.3	63.7↓ 3.1	53.3↓ 13.5	40.4	38.6↓ 1.8	39.5↓ 0.9	33.5↓ 6.9	89.8	83.9↓ 5.9	84.3↓ 5.5	84.0↓ 5.8
DivPrune	67.0	63.7↓ 3.3	63.6↓ 3.4	53.2↓ 13.8	40.5	38.8↓ 1.7	39.5↓ 1.0	33.7↓ 6.8	89.7	83.8↓ 5.9	84.3↓ 5.4	84.1↓ 5.6
FlowCut	61.5	58.6↓ 2.9	58.3↓ 3.2	44.9↓ 16.6	32.7	31.6↓ 1.1	30.6↓ 2.1	26.7↓ 6.0	89.2	83.4↓ 5.8	84.4↓ 4.8	83.7↓ 5.5
PruMerge	37.0	38.1↑ 1.1	38.7↑ 1.7	35.0↓ 2.0	12.8	14.2↑ 1.4	13.7↑ 0.9	12.6↓ 0.2	74.6	75.0↑ 0.4	74.5↓ 0.1	74.2↓ 0.4
Average	59.9	57.5↓ 2.4	57.4↓ 2.5	**48.1**↓ 11.8	37.0	35.7↓ 1.3	35.7↓ 1.3	**29.4**↓ 7.6	86.6	82.2↓ 4.4	82.6↓ 4.0	**82.1**↓ 4.5
Blind ($K_{\text{model}} = 0$)												
Blind	32.9				10.5				72.8			

Table 3: Ablation study of surrogate configurations on LLaVA with TextVQA and VisionZip.

Variant	$K_{\text{model}} = 16$	$K_{\text{model}} = 32$	$K_{\text{model}} = 64$	Average
M1	13.5↓ 30.3	**14.7**↓ 33.4	16.5↓ 37.3	14.9↓ 33.7
M2	13.8↓ 30.0	15.6↓ 32.5	17.7↓ 36.1	15.7↓ 32.9
M3	13.4↓ 30.4	15.5↓ 32.6	16.7↓ 37.1	15.2↓ 33.4
M4	12.8↓ 31.0	16.3↓ 31.8	17.2↓ 36.6	15.4↓ 33.2
M5	**12.7**↓ 31.1	15.3↓ 32.8	**15.9**↓ 37.9	**14.6**↓ 34.0

Table 4: Ablation study of perturbation objectives on LLaVA with TextVQA and VisionZip.

Variant	$K_{\text{model}} = 16$	$K_{\text{model}} = 32$	$K_{\text{model}} = 64$	Average
V1	43.8	48.1	53.8	48.6
V2	24.8↓ 19.0	26.9↓ 21.2	27.0↓ 26.8	26.2↓ 22.4
V3	21.8↓ 22.0	26.2↓ 21.9	32.0↓ 21.8	26.7↓ 21.9
V4	13.1↓ 30.7	15.4↓ 32.7	17.4↓ 36.4	15.3↓ 33.3
V5	**12.7**↓ 31.1	**15.3**↓ 32.8	**15.9**↓ 37.9	**14.6**↓ 34.0

(SQA) (Lu et al., 2022). These datasets cover diverse multimodal capabilities, including visual reasoning, scene-text understanding, and knowledge-intensive reasoning, enabling evaluation across complementary visual understanding scenarios. More details can be found in Appendix C.1.

Compared Baselines. We evaluate CLOUD across five representative visual token compression methods, including VisionZip (Yang et al., 2025c), VisPruner (Zhang et al., 2025b), DivPrune (Alvar et al., 2025), FlowCut (Tong et al., 2026), and PruMerge (Shang et al., 2025). For adversarial comparison, we adopt VEAttack (Mei et al., 2026), denoted as Base in our tables, and CAGE (Zhang et al., 2026c). Detailed information is reported in Appendix C.2.

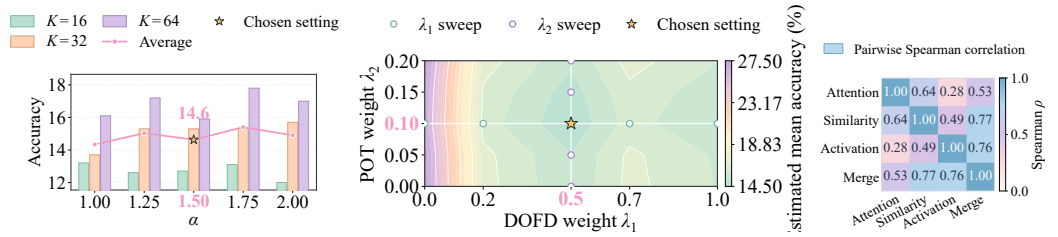
Implementation Details. We conduct experiments on LLaVA-v1.5-7B (Liu et al., 2024) and Qwen3-VL-8B-Instruct (Bai et al., 2025) to evaluate CLOUD and other baselines across different LVLM architectures. Following the prior work, we set the perturbation budget to $\epsilon = 2/255$ under the ℓ_∞ norm and the PGD step size to $0.5/255$. For LLaVA, we evaluate compression budgets $K_{\text{model}} \in \{16, 32, 64\}$, while for Qwen3-VL, we use $K_{\text{model}} \in \{36, 72\}$. We set the loss weights for DOFD and POT to 0.5 and 0.1, respectively, and keep the remaining hyperparameters fixed across datasets and methods. ore details can be seen in Appendix C.3.

4.2 Performance Comparison

We compare CLOUD with Base and CAGE across several compression methods, benchmarks, and budgets on LLaVA and Qwen3-VL, as reported in Tables 1 and 2. These results yield the following observations. **Obs.❶** ***CLOUD consistently improves attack effectiveness across diverse compression strategies.*** On LLaVA TextVQA, CLOUD reduces the average robust accuracy of CAGE by 2.7, 2.9, and 3.0 under $K_{\text{model}} = 64, 32, 16$, respectively. Similar improvements are observed across GQA and SQA, demonstrating that CLOUD can reliably exploit vulnerable visual evidence preserved after compression. **Obs.❷** ***CLOUD generalizes across model architectures and limited token budgets.*** On Qwen3-VL, CLOUD reduces the average robust accuracy of CAGE by 6.5 and 9.3 on GQA and by 8.7 and 6.3 on TextVQA under $K_{\text{model}} = 72$ and 36, respectively. These results demonstrate that CLOUD remains effective under different compression levels and model architectures.

4.3 Ablation Study

We ablate the surrogate configurations and perturbation objectives on LLaVA with TextVQA and VisionZip. M1–M4 use attention, similarity, activation, and merge individually, while M5 uses the complete surrogate pool. (i) V1 denotes the clean setting. (ii) V2 adopts distributed feature distortion. (iii) V3 applies coordinated distributional disruption to V1. (iv) V4 introduces compression-



(a) Entmax parameter α. (b) Joint λ_1-λ_2 sensitivity. (c) Surrogate agreement.

Figure 4: Sensitivity and Surrogate Agreement of CLOUD. (a) Sensitivity to the Entmax parameter α. (b) Joint sensitivity to the DOFD loss weight λ_1 and POT loss weight λ_2, with the selected settings highlighted. (c) Pairwise Spearman correlations among the token-selection surrogates.

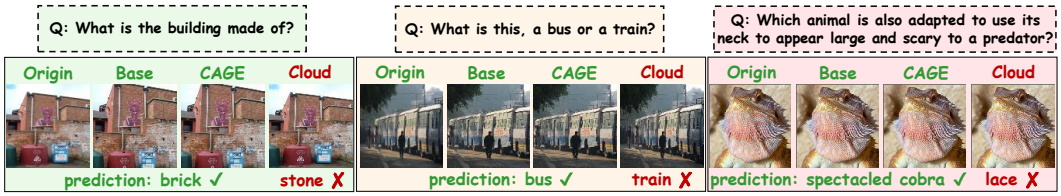


Figure 5: Case Analysis. Qualitative comparison of attack results produced by CAGE and CLOUD across TextVQA, GQA, and SQA.

aware adversarial profiling, and (v) V5 combines all modules. Results are reported in Tables 3 and 4. **Obs.❸** ***Complementary surrogates improve attack effectiveness across compression budgets.*** M5 achieves the lowest average robust accuracy of 14.6%, compared with 14.9%–15.7% for M1–M4, showing that combining surrogate responses yields more reliable compression-aware profiling. **Obs.❹** ***Compression-aware profiling is critical to coordinated disruption.*** V3 reaches 26.7% average robust accuracy, while V4 substantially reduces it to 15.3%. Combining profiling with coordinated disruption in V5 further lowers accuracy to 14.6%, showing that adaptive support estimation enables subsequent evidence disruption. More ablation studies can be found in Appendix. D.3.

4.4 Further Analysis

Performance under Different Hyperparameters. We study the sensitivity of CLOUD to the Entmax parameter α, the DOFD weight λ_1, and the POT weight λ_2, as shown in Fig. 4(a) and (b). From the results, we find that **Obs.❺** ***the three hyperparameters control complementary aspects of perturbation allocation and optimization.*** Specifically, α determines the sparsity of token allocation, while λ_1 and λ_2 balance determinantal feature distortion and transport-based evidence disruption, respectively. CLOUD maintains consistently strong attack performance across a broad range of hyperparameter settings, demonstrating its robustness to parameter variations.

Visualization Analysis. We further visualize surrogate responses and attack cases. We notice that **Obs.❻** ***CLOUD captures complementary compression responses and consistently disrupts predictions after token compression.*** As shown in Fig. 4(c), pairwise Spearman correlations among the four surrogates range from 0.28 to 0.77, showing that they capture complementary compression behaviors. Fig. 5 further shows that Base and CAGE answer the TextVQA, GQA, and SQA cases correctly, while CLOUD produces incorrect predictions across all three tasks. Overall, these visualizations support both our surrogate design and attack effectiveness across reasoning scenarios.

5 Conclusion

In this work, we revisit adversarial perturbation under visual token compression through an evidence bottleneck determined by each input. We examine how compression reshapes token utility and interactions among retained tokens, then use these signals to adapt the perturbation support. Our CLOUD first profiles diverse compression behaviors to estimate which visual evidence remains influential. The resulting support guides coordinated distributional disruption, allowing the attack to determine which evidence to target and how to alter its distribution after compression, thereby achieving reliable degradation across varying compression budgets, architectures and settings.

AI USE STATEMENT

Generative AI tools were used solely to aid or polish the writing of this manuscript, including grammar, wording, and readability. All scientific content, methodological design, experiments, analysis, and conclusions were produced and verified by the authors.

ETHICS STATEMENT

This work studies adversarial vulnerabilities of visual token compression methods for the purpose of robustness evaluation. All experiments are conducted on locally deployed open-source models and public research benchmarks in controlled environments, without targeting deployed services, private systems, or real-world users. The proposed methodology is intended to identify potential failure modes and support the development of more robust and reliable multimodal systems.

REPRODUCIBILITY STATEMENT

We provide the complete formulation and optimization objectives of our method in Section 3 and summarize the overall execution pipeline in Algorithm 1. The evaluated models, token compression methods, datasets, attack settings, evaluation metrics, hyperparameters, and additional implementation details are described in Section 4.1 and Appendix C to facilitate reproduction of the reported results.

REFERENCES

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35:23716–23736, 2022.

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.

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

Luke Bailey, Euan Ong, Stuart Russell, and Scott Emmons. Image hijacks: Adversarial images can control generative models at runtime. *arXiv preprint arXiv:2309.00236*, 2023.

Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your vit but faster. *arXiv preprint arXiv:2210.09461*, 2022.

Adrian Bulat, Yassine Ouali, and Georgios Tzimiropoulos. Compress & cache: Vision token compression for efficient generation and retrieval. *Advances in Neural Information Processing Systems*, 38:31943–31968, 2026.

Davide Caffagni, Federico Cocchi, Luca Barsellotti, Nicholas Moratelli, Sara Sarto, Lorenzo Baraldi, Marcella Cornia, and Rita Cucchiara. The revolution of multimodal large language models: A survey. *Findings of the association for computational linguistics: ACL 2024*, pp. 13590–13618, 2024.

Xiaowen Cai, Daizong Liu, Xiaoye Qu, Xiang Fang, Jianfeng Dong, Keke Tang, Pan Zhou, Lichao Sun, and Wei Hu. Towards building model/prompt-transferable attackers against large vision-language models. *Advances in Neural Information Processing Systems*, 38:174022–174058, 2026.

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.

Jieneng Chen, Luoxin Ye, Ju He, Zhao-Yang Wang, Daniel Khashabi, and Alan Yuille. Efficient large multi-modal models via visual context compression. *Advances in neural information processing systems*, 37:73986–74007, 2024a.

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, 2024b.

Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 24185–24198, 2024c.

Janghoon Cho, Jungsoo Lee, Munawar Hayat, Kyuwoong Hwang, Fatih Porikli, and Sungha Choi. Floc: Facility location-based efficient visual token compression for long video understanding. In *International Conference on Learning Representations*, volume 2026, pp. 102973–102996, 2026.

Xuanming Cui, Alejandro Aparcedo, Young Kyun Jang, and Ser-Nam Lim. On the robustness of large multimodal models against image adversarial attacks. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24625–24634. IEEE, 2024.

Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26, 2013.

Wenliang Dai, Junnan Li, Dongxu Li, Anthony Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale N Fung, and Steven Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning. *Advances in neural information processing systems*, 36:49250–49267, 2023.

Yiwei Fu, Hui Wan, Xiao Luo, and Minghua Deng. Star: Test-time adaptation can enhance universal prompt learning for vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 31482–31492, 2026.

Jiaqi Gu, Yang Qi, Lubin Fan, Yue Wu, Ying Wang, Kun Ding, SHIMING XIANG, Jieping Ye, et al. Knowledge-based visual question answer with multimodal processing, retrieval and filtering. *Advances in Neural Information Processing Systems*, 38:119251–119282, 2026.

Jiaxian Guo, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Boyang Li, Dacheng Tao, and Steven Hoi. From images to textual prompts: Zero-shot visual question answering with frozen large language models. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10867–10877. IEEE, 2023.

Shuyang Hao, Bryan Hooi, Jun Liu, Kai-Wei Chang, Zi Huang, and Yujun Cai. Exploring visual vulnerabilities via multi-loss adversarial search for jailbreaking vision-language models. *arXiv preprint arXiv:2411.18000*, 2024.

Kai Hu, Weichen Yu, Li Zhang, Alexander Robey, Andy Zou, Haoqi Hu, Chengming Xu, and Matt Fredrikson. Omni-attack: Adversarial attacks on open-ended vqa in black-box multimodal llms. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 42341–42351, 2026.

Kai Huang, Hao Zou, Ye Xi, BoChen Wang, Zhen Xie, and Liang Yu. Ivtp: Instruction-guided visual token pruning for large vision-language models. In *European conference on computer vision*, pp. 214–230. Springer, 2024.

Wenxuan Huang, Zijie Zhai, Yunhang Shen, Shaosheng Cao, Fei Zhao, Xiangfeng Xu, Zheyu Ye, and Shaohui Lin. Dynamic-llava: Efficient multimodal large language models via dynamic vision-language context sparsification. In *International Conference on Learning Representations*, volume 2025, pp. 69927–69955, 2025.

Yihong Huang, Fei Ma, Yihua Shao, Jingcai Guo, Zitong YU, Laizhong Cui, and Qi Tian. Nüwa: Mending the spatial integrity torn by VLM token pruning. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=C9yclwdquU.

Drew A. Hudson and Christopher D. Manning. GQA: A new dataset for real-world visual reasoning and compositional question answering. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pp. 6700–6709. Computer Vision Foundation / IEEE, 2019. doi: 10.1109/CVPR.2019.00686. URL http://openaccess.thecvf.com/content_CVPR_2019/html/Hudson_GQA_A_New_Dataset_for_Real-World_Visual_Reasoning_and_Compositional_CVPR_2019_paper.html.

Xiaojun Jia, Sensen Gao, Simeng Qin, Tianyu Pang, Chao Du, Yihao Huang, Xinfeng Li, Yiming Li, Bo Li, and Yang Liu. Adversarial attacks against closed-source mllms via feature optimal alignment. *Advances in Neural Information Processing Systems*, 38:57993–58028, 2026.

Alex Kulesza and Ben Taskar. Determinantal point processes for machine learning. *Foundations and Trends® in Machine Learning*, 5(2-3):123–286, 2012.

Lei Lei, Jie Gu, Xiaokang Ma, Chu Tang, Jingmin Chen, and Tong Xu. Task-related token compression in multimodal large language models from an explainability perspective. In *International Conference on Learning Representations*, volume 2026, pp. 112724–112746, 2026.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. LLaVA-onevision: Easy visual task transfer. *Transactions on Machine Learning Research*, 2025a. ISSN 2835-8856. URL https://openreview.net/forum?id=zKv8qULV6n.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pp. 19730–19742. PmLR, 2023.

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, 2025b.

Chenfei Liao, Wensong Wang, Zichen Wen, Xu Zheng, Yiyu Wang, Haocong He, Yuanhuiyi Lyu, Lutao Jiang, Xin Zou, Yuqian Fu, et al. Are we using the right benchmark: An evaluation framework for visual token compression methods. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 4236–4253, 2026.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pp. 26286–26296. IEEE, 2024. doi: 10.1109/CVPR52733.2024.02484. URL https://doi.org/10.1109/CVPR52733.2024.02484.

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 Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/11332b6b6cf4485b84afadb1352d3a9a-Abstract-Conference.html.

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.

George L Nemhauser, Laurence A Wolsey, and Marshall L Fisher. An analysis of approximations for maximizing submodular set functions—i. *Mathematical programming*, 14(1):265–294, 1978.

Sen Nie, Jie Zhang, Jianxin Yan, Shiguang Shan, and Xilin Chen. V-attack: Targeting disentangled value features for controllable adversarial attacks on lvlms. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 42257–42267, 2026.

Zhiliang Peng, Wenhui Wang, Li Dong, Yaru Hao, Shaohan Huang, Shuming Ma, Qixiang Ye, and Furu Wei. Grounding multimodal large language models to the world. In *International Conference on Learning Representations*, volume 2024, pp. 51575–51598, 2024.

Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. Dynamicvit: Efficient vision transformers with dynamic token sparsification. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 13937–13949, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/747d3443e319a22747fbb873e8b2f9f2-Abstract.html.

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 *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pp. 8317–8326. Computer Vision Foundation / IEEE, 2019. doi: 10.1109/CVPR.2019.00851. URL http://openaccess.thecvf.com/content_CVPR_2019/html/Singh_Towards_VQA_Models_That_Can_Read_CVPR_2019_paper.html.

Wei Suo, Ji Ma, Mengyang Sun, Lin Yuanbo Wu, Peng Wang, and Yanning Zhang. Pruning all-rounder: Rethinking and improving inference efficiency for large vision language models. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 20247–20256. IEEE, 2025.

Liyan Tang, Grace Kim, Xinyu Zhao, Thom Lake, Wenxuan Ding, Fangcong Yin, Prasann Singhal, Manya Wadhwa, Zeyu Liu, Zayne Sprague, et al. Chartmuseum: Testing visual reasoning capabilities of large vision-language models. *Advances in Neural Information Processing Systems*, 38, 2026.

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.

Weihan Wang, Qingsong Lv, Wenmeng Yu, Wenyi Hong, Ji Qi, Yan Wang, Junhui Ji, Zhuoyi Yang, Lei Zhao, Xixuan Song, et al. Cogvlm: Visual expert for pretrained language models. *Advances in Neural Information Processing Systems*, 37:121475–121499, 2024a.

Yubo Wang, Chaohu Liu, yanqiuqu, Haoyu Cao, Deqiang Jiang, and Linli Xu. Break the visual perception: Adversarial attacks targeting encoded visual tokens of large vision-language models. In *ACM Multimedia 2024*, 2024b. URL https://openreview.net/forum?id=tocfToCGF1.

Peng Xie, Yequan Bie, Jianda Mao, Yangqiu Song, Yang Wang, Hao Chen, and Kani Chen. Chain of attack: On the robustness of vision-language models against transfer-based adversarial attacks. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14679–14689. IEEE, 2025.

Long Xing, Qidong Huang, Xiaoyi Dong, Jiajie Lu, Pan Zhang, Yuhang Zang, Yuhang Cao, Conghui He, Jiaqi Wang, Feng Wu, et al. Conical visual concentration for efficient large vision-language models. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14593–14603. IEEE, 2025.

Guowei Xu, Peng Jin, Ziang Wu, Hao Li, Yibing Song, Lichao Sun, and Li Yuan. Llava-cot: Let vision language models reason step-by-step. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 2087–2098. IEEE, 2025.

Cheng Yang, Yang Sui, Jinqi Xiao, Lingyi Huang, Yu Gong, Chendi Li, Jinghua Yan, Yu Bai, Ponnuswamy Sadayappan, Xia Hu, et al. Topv: Compatible token pruning with inference time optimization for fast and low-memory multimodal vision language model. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19803–19813. IEEE, 2025a.

Longrong Yang, Dong Shen, Chaoxiang Cai, Kaibing Chen, Fan Yang, Tingting Gao, Di Zhang, and Xi Li. Libra-merging: Importance-redundancy and pruning-merging trade-off for acceleration plug-in in large vision-language model. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9402–9412. IEEE, 2025b.

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, 2025c.

Sihan Yang, Runsen Xu, Chenhang Cui, Tai Wang, Dahua Lin, and Jiangmiao Pang. Vflowopt: A token pruning framework for lmms with visual information flow-guided optimization. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 23924–23934. IEEE, 2025d.

Qinghao Ye, Haiyang Xu, Jiabo Ye, Ming Yan, Anwen Hu, Haowei Liu, Qi Qian, Ji Zhang, and Fei Huang. Mplug-owi2: Revolutionizing multi-modal large language model with modality collaboration. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13040–13051. IEEE, 2024.

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.

Yufei Zhan, Yousong Zhu, Hongyin Zhao, Fan Yang, Shurong Zheng, Ming Tang, and Jinqiao Wang. Vision-r1: Evolving human-free alignment in large vision-language models via vision-guided reinforcement learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5807–5817, 2026.

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.

Jun Zhang, Yicheng Ji, Feiyang Ren, Yihang Li, Bowen Zeng, Zonghao Chen, Ke Chen, Lidan Shou, Gang Chen, and Huan Li. Efficient inference for large vision-language models: Bottlenecks, techniques, and prospects. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 21036–21066, 2026a.

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*, 2026b.

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*, 2026c.

Yuan Zhang, Chun-Kai Fan, Junpeng Ma, Wenzhao Zheng, Tao Huang, Kuan Cheng, Denis A Gudovskiy, Tomoyuki Okuno, Yohei Nakata, Kurt Keutzer, and Shanghang Zhang. SparseVLM: Visual token sparsification for efficient vision-language model inference. In *Forty-second International Conference on Machine Learning*, 2025c. URL https://openreview.net/forum?id=80faIPZ67S.

Yuan Zhang, Junpeng Ma, Qizhe Zhang, Chun-Kai Fan, Wenzhao Zheng, Kuan Cheng, Jiwen Lu, and Shanghang Zhang. Sparsevlm+: Visual token sparsification with improved text-visual attention pattern. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2026d.

Deyao Zhu, Xiaoqian Shen, Xiang Li, Mohamed Elhoseiny, et al. Minigpt-4: Enhancing vision-language understanding with advanced large language models. In *International Conference on Learning Representations*, volume 2024, pp. 18378–18394, 2024.

Supplementary Material

A Additional Method Details **16**

A.1 Additional Problem Formulation 16

A.2 Overall Optimization Pipeline 16

B Proofs for the Theoretical Analysis **16**

B.1 Proof of Theorem 3.1 16

B.2 Proof of Theorem 3.2 17

C Detailed Experimental Setup **18**

C.1 Additional Benchmark Datasets 18

C.2 Additional Baseline Implementations 18

C.3 Additional Implementation Details 19

D Additional Experimental Results **19**

D.1 Generalization to InternVL3.5 19

D.2 Generalization to GLM-4.6V-Flash 20

D.3 Additional Ablation Study 21

D.4 Analysis of Sample-Adaptive Attack Budgets 21

D.5 Analysis of Cross-Surrogate Complementarity 21

D.6 Paired Analysis of Attack Effectiveness 22

D.7 Analysis of Post-Compression Feature Discrepancy 23

D.8 Analysis of Token-Level Transport 23

D.9 Visualization of Sample-Adaptive and Coordinated Support 23

D.10 Further Analysis of Adaptive Attack Budget 25

D.11 Different Perturbation Budgets 26

D.12 Efficiency Analysis 27

A Additional Method Details

A.1 Additional Problem Formulation

We consider an LVLM composed of a visual encoder $\mathcal{E}$, a visual projector $\mathcal{P}$, and a large language model $\mathcal{F}$. Given an input image $\mathbf{x}_v \in \mathbb{R}^{H\times W\times C}$ and a textual prompt $\mathbf{x}_t$, the visual encoder extracts a sequence of N visual tokens $\mathbf{H} = \mathcal{E}(\mathbf{x}_v) = [\mathbf{h}_1, \ldots, \mathbf{h}_N] \in \mathbb{R}^{N\times D}$. Visual token compression introduces a compression module $\mathcal{C}$ between the visual encoder and the projector. Given a retention budget $K < N$, the compression module reduces the original sequence from N tokens to K compressed tokens. The resulting representation is formulated as:

$$\mathbf{Z} = \mathcal{C}(\mathbf{H}; K) \in \mathbb{R}^{K\times D}. \tag{18}$$

We denote the retention budget of the target model by K_{model}. The compressed visual tokens are projected into the language embedding space and combined with the textual prompt for autoregressive generation. The response probability is modeled as:

$$P(\mathbf{Y} \mid \mathbf{x}_v, \mathbf{x}_t) = \prod_{l=1}^{L} P(y_l \mid \mathcal{P}(\mathbf{Z}), \mathbf{x}_t, y_{<l}) \,. \tag{19}$$

We consider an attack setting where the visual encoder is accessible during optimization, while the compression strategy and K_{model} of the target model are unknown. Given a perturbation $\boldsymbol{\delta}$ bounded by $\|\boldsymbol{\delta}\|_\infty \le \epsilon$, the attack aims to maximize the discrepancy between the clean and adversarial visual evidence that remains after compression. The ideal objective is formulated as:

$$\boldsymbol{\delta}^* = \arg\max_{\|\boldsymbol{\delta}\|_\infty \le \epsilon} \mathcal{D}\left(\mathcal{C}^*\big(\mathcal{E}(\mathbf{x}_v); K_{\text{model}}\big), \mathcal{C}^*\big(\mathcal{E}(\mathbf{x}_v + \boldsymbol{\delta}); K_{\text{model}}\big)\right), \tag{20}$$

where $\mathcal{C}^*$ denotes the compression strategy of the target model and $\mathcal{D}$ measures the discrepancy between the resulting compressed visual representations. Since $\mathcal{C}^*$ and K_{model} are unknown during attack generation, the central problem is to identify and disrupt visual evidence that remains influential across different compression behaviors and retention budgets.

A.2 Overall Optimization Pipeline

The overall execution pipeline of `CLOUD` consists of two steps. First, we calibrate the perturbation-related hyperparameters θ^* on the calibration set through Bayesian optimization, which improves the transferability of the learned attack configuration across different compression surrogates. Based on the calibrated parameters, we then perform surrogate profiling over T_{profile} iterations. Across profiling iterations, responses from multiple surrogate branches are accumulated and aggregated to update the cross-surrogate consensus utility, which is subsequently used by the Entmax allocator to determine the token-wise allocation $\mathbf{q}^{(t)}$ and the corresponding attack budget $K_{\text{attack}}^{(t)}$. After profiling, the resulting allocation is fixed as the first-step initialization, and a DPP-based ordering is constructed to identify a diverse set of supporting tokens together with their survival weights $\boldsymbol{\omega}$. Finally, during the attack stage, we jointly optimize the diversity-oriented objective $\mathcal{L}_{\text{DOFD}}$ and the attribution-oriented objective $\mathcal{L}_{\text{POT}}$ to progressively update the perturbation $\boldsymbol{\delta}$. This procedure encourages the perturbation to disrupt complementary visual evidence while concentrating the attack on tokens that are more likely to survive the unknown compression process. The final adversarial example is obtained by clipping $\mathbf{x}_v + \boldsymbol{\delta}$ to the valid input range. A more detailed summary of the complete execution pipeline of `CLOUD` is presented in Algorithm 1.

B Proofs for the Theoretical Analysis

B.1 Proof of Theorem 3.1

Proof. By Assumption 3.1, for every token i, more than half of $\{m_{s,i}\}_{s\in\mathcal{S}}$ lie in $[u_i^\star - \varepsilon_m, u_i^\star + \varepsilon_m]$. By the definition of the median, the aggregated utility therefore satisfies $|u_i - u_i^\star| \le \varepsilon_m$. Using $z_i = u_i/\tau_a$, $z_i^\star = u_i^\star/\tau_a$, and $\rho = \varepsilon_m/\tau_a$, we obtain

$$\|\mathbf{z} - \mathbf{z}^\star\|_\infty \le \rho. \tag{21}$$

Algorithm 1 The execution pipeline of CLOUD

Input: image $\mathbf{x}_v$, encoder $\mathcal{E}$, surrogate pool $\mathcal{S}$, calibration set $\mathcal{X}_{\text{cal}}$, perturbation budget ϵ

Calibrate θ^* by leave-one-surrogate-out Bayesian optimization with Eq. 3
// Calibrate surrogates
Initialize $\mathbf{H}^0 = \mathcal{E}(\mathbf{x}_v)$, $\boldsymbol{\delta}$ with a random perturbation satisfying $\|\boldsymbol{\delta}\|_\infty \leq \epsilon$, and surrogate memories
for $t = 1, \ldots, T_{\text{profile}}$ **do**
    Compute surrogate responses and update the cross-surrogate consensus utility with Eqs. 2–4
    Obtain $\mathbf{q}^{(t)}$ and $K^{(t)}_{\text{attack}}$ by Entmax allocation, and update $\boldsymbol{\delta}$ with Eq. 5
// Allocate attack budget
end for
Set $\mathbf{q}^{\text{S1}} \leftarrow \mathbf{q}^{(T_{\text{profile}})}$ and $K_{\text{attack}} \leftarrow K^{(T_{\text{profile}})}_{\text{attack}}$
Construct the DPP ordering and survival weights $\boldsymbol{\omega}$ with Eqs. 6–7
// Select diverse support
for $t = 1, \ldots, T_{\text{attack}}$ **do**
    Compute $\mathcal{L}_{\text{DOFD}}$, solve $\boldsymbol{\Gamma}^*$, and compute $\mathcal{L}_{\text{POT}}$
    Update $\boldsymbol{\delta}$ using $\lambda_1 \mathcal{L}_{\text{DOFD}} + \lambda_2 \mathcal{L}_{\text{POT}}$
// Disrupt visual evidence
end for
return $\mathbf{x}_v^{\text{adv}} = \text{clip}(\mathbf{x}_v + \boldsymbol{\delta})$

Let $\mathcal{A}(\mathbf{z}) = \{i \in [N] \mid z_i > \gamma_\alpha(\mathbf{z})\}$ denote the active support of α Entmax. Equation 21 gives $\mathbf{z}^\star - \rho\mathbf{1} \leq \mathbf{z} \leq \mathbf{z}^\star + \rho\mathbf{1}$. By the coordinatewise monotonicity and translation equivariance of the Entmax threshold, we obtain

$$\gamma^\star - \rho \leq \gamma_\alpha(\mathbf{z}) \leq \gamma^\star + \rho. \tag{22}$$

For any token satisfying $z_i^\star > \gamma^\star + 2\rho$, Eq. 21 gives $z_i \geq z_i^\star - \rho > \gamma^\star + \rho$. Combining this inequality with Eq. 22 gives $z_i > \gamma_\alpha(\mathbf{z})$. By the support definition of α Entmax, $i \in \mathcal{A}(\mathbf{z})$. For any $i \in \mathcal{A}(\mathbf{z})$, the support definition gives $z_i > \gamma_\alpha(\mathbf{z})$. Using Eq. 22, we have $z_i > \gamma^\star - \rho$. Equation 21 further gives $z_i^\star \geq z_i - \rho > \gamma^\star - 2\rho$. Therefore,

$$\{i \mid z_i^\star > \gamma^\star + 2\rho\} \subseteq \mathcal{A}(\mathbf{z}) \subseteq \{i \mid z_i^\star > \gamma^\star - 2\rho\}. \tag{23}$$

Taking the cardinality of the three sets and applying the definitions of K^-, K^+, and $K_{\text{attack}} = |\mathcal{A}(\mathbf{z})|$ gives

$$K^- \leq K_{\text{attack}} \leq K^+. \tag{24}$$

□

B.2 Proof of Theorem 3.2

Proof. The kernel in Eq. 6 can be written as

$$\mathbf{G} = \text{Diag}(\mathbf{a})\mathbf{Z}\mathbf{Z}^\top \text{Diag}(\mathbf{a}) + \xi\mathbf{I}, \tag{25}$$

where the rows of $\mathbf{Z}$ are the normalized token features $\mathbf{z}_i$. Since $\xi > 0$, $\mathbf{G}$ is positive definite and admits a factorization $\mathbf{G} = \mathbf{B}\mathbf{B}^\top$. For any subset S, let $\mathbf{B}_S$ contain the rows indexed by S. By Sylvester's determinant theorem,

$$F(S) = \log\det\left(\mathbf{I} + \mathbf{B}_S\mathbf{B}_S^\top\right) = \log\det\left(\mathbf{I} + \mathbf{B}_S^\top\mathbf{B}_S\right). \tag{26}$$

The definition immediately gives $F(\varnothing) = 0$. For any token $j \notin S$, let $\mathbf{b}_j^\top$ denote the corresponding row of $\mathbf{B}$. By the matrix determinant lemma, the marginal gain is

$$F(S \cup \{j\}) - F(S) = \log\left(1 + \mathbf{b}_j^\top \left(\mathbf{I} + \mathbf{B}_S^\top\mathbf{B}_S\right)^{-1}\mathbf{b}_j\right). \tag{27}$$

Since $\left(\mathbf{I} + \mathbf{B}_S^\top\mathbf{B}_S\right)^{-1}$ is positive definite, the marginal gain is nonnegative, showing that F is monotone. For any $S \subseteq T$,

$$\mathbf{I} + \mathbf{B}_S^\top\mathbf{B}_S \preceq \mathbf{I} + \mathbf{B}_T^\top\mathbf{B}_T, \tag{28}$$

and the order-reversing property of matrix inversion gives

$$\left(\mathbf{I} + \mathbf{B}_S^\top\mathbf{B}_S\right)^{-1} \succeq \left(\mathbf{I} + \mathbf{B}_T^\top\mathbf{B}_T\right)^{-1}. \tag{29}$$

Substituting this relation into Eq. 27 yields

$$F(S \cup \{j\}) - F(S) \geq F(T \cup \{j\}) - F(T) \tag{30}$$

for every $j \notin T$. Thus, F satisfies the diminishing returns property and is submodular. Since F is normalized, monotone, and submodular, greedy marginal maximization under the cardinality constraint $|S| \leq K_{\text{attack}}$ satisfies the classical approximation guarantee (Nemhauser et al., 1978):

$$F(S_{K_{\text{attack}}}) \geq \left(1 - \frac{1}{e}\right) F(S^{\star}_{K_{\text{attack}}}). \tag{31}$$

Therefore, the greedy support $S_{K_{\text{attack}}}$ used in our construction achieves the stated approximation guarantee. □

C Detailed Experimental Setup

C.1 Additional Benchmark Datasets

We select three benchmarks with substantially different forms of visual evidence to evaluate the robustness of compressed LVLMs beyond a single visual question answering regime.

GQA. GQA (Hudson & Manning, 2019) is a compositional visual reasoning benchmark constructed from Visual Genome scene graphs, containing large-scale questions grounded in objects, attributes, and relations. Its questions frequently require relational and multi-step reasoning, including spatial relations, comparisons, and logical composition. We include GQA to evaluate adversarial robustness when model predictions depend on distributed relational evidence across multiple visual regions rather than a single salient cue.

TextVQA. TextVQA (Singh et al., 2019) contains 45,336 questions over 28,408 images collected from OpenImages and focuses on reasoning about text embedded in natural scenes. Answering these questions typically requires recognizing localized textual content and integrating it with the surrounding visual context. Compared with general visual reasoning, the decisive evidence in TextVQA is often spatially sparse and fine-grained, making it particularly sensitive to whether text-bearing visual tokens are preserved after compression.

ScienceQA. ScienceQA (SQA) (Lu et al., 2022) is a knowledge-intensive multimodal benchmark containing 21,208 multiple-choice science questions collected from elementary- and high-school curricula. The questions span natural science, language science, and social science, and may involve visual and textual contexts together with background knowledge. We use SQA to examine robustness when visual evidence must be integrated with structured scientific knowledge and higher-level reasoning rather than being interpreted solely through perceptual cues.

Together, the three benchmarks provide complementary evidence structures that GQA emphasizes distributed relational and compositional evidence, TextVQA relies heavily on sparse and localized textual evidence, and SQA requires knowledge-intensive multimodal reasoning. This diversity allows us to evaluate whether the effectiveness of adversarial perturbations under visual token compression generalizes across substantially different forms of visual information.

C.2 Additional Baseline Implementations

We consider five representative visual token compression methods covering importance-based selection, diversity-based pruning, information-flow-based pruning, and token merging, together with two adversarial attack baselines.

- **VisionZip.** VisionZip (Yang et al., 2025c) follows a select-and-merge strategy. It first identifies dominant visual tokens according to their importance and preserves them explicitly, while the remaining redundant tokens are aggregated into a smaller number of contextual tokens according to feature similarity. This design preserves both highly salient visual evidence and complementary contextual information.

- **VisPruner.** VisPruner (Zhang et al., 2025b) performs training-free visual token pruning based primarily on visual cues. It first identifies informative tokens using visual attention and subsequently removes redundant tokens according to feature similarity, thereby retaining a compact set that jointly captures salient and diverse visual information.
- **DivPrune.** DivPrune (Alvar et al., 2025) formulates visual token selection as a Max-Min Diversity Problem. Instead of ranking tokens solely by individual importance, it selects a subset that maximizes the minimum pairwise feature distance among retained tokens, reducing redundancy and encouraging broader coverage of distinct visual content.
- **FlowCut.** FlowCut (Tong et al., 2026) estimates visual token redundancy from the perspective of information flow across transformer layers. Rather than relying on a single-layer importance score, it models how information propagates through token interactions and progressively removes tokens with limited contribution to the global visual representation.
- **PruMerge.** PruMerge (Shang et al., 2025) combines adaptive token pruning and feature merging. It exploits the sparse attention distribution between the class token and visual tokens to identify important tokens, and then aggregates redundant visual information according to feature similarity, reducing token length while augmenting the retained representations with information from pruned tokens.
- **VEAttack.** VEAttack (Mei et al., 2026) is a downstream-agnostic vision-encoder attack that generates adversarial images by disrupting visual token representations. It optimizes the discrepancy between clean and adversarial visual features without requiring task labels or gradients from the downstream language model, and is denoted as *Base* in our experiments.
- **CAGE.** CAGE (Zhang et al., 2026c) is a compression-aware adversarial attack specifically designed for LVLMs with visual token compression. It accounts for uncertainty in the deployed token budget and encourages perturbations to remain effective on visual tokens that are likely to survive compression.

C.3 Additional Implementation Details

Following the prior work, we evaluate the first 1,000 instances of each benchmark. For attack generation, we use CLIP ViT-L/14@336 as the vision encoder and extract visual token representations from its penultimate transformer layer. Token importance is computed from CLS-to-patch attention in the penultimate attention layer, while feature discrepancy is measured by cosine distance. Perturbations are optimized in the CLIP-normalized input space with channel-wise rescaling according to the corresponding normalization statistics. We use a PGD step size of $0.5/255$. The loss weights for DOFD and POT are set to $\lambda_1 = 0.5$ and $\lambda_2 = 0.1$, respectively. The entropic Sinkhorn regularization coefficient is set to 0.03, and all remaining hyperparameters are fixed across datasets and compression methods. All experiments use a random seed of 42 with deterministic CUDA execution enabled whenever applicable.

D Additional Experimental Results

D.1 Generalization to InternVL3.5

We further evaluate the generalization of `CLOUD` on InternVL3.5 across five visual token compression methods, including VisionZip, VisPruner, DivPrune, FlowCut, and PruMerge. We consider three retention budgets, $K_{\text{model}} \in \{16, 32, 64\}$, and report the results on GQA, TextVQA, and SQA in Table 5. As can be seen from the table, `CLOUD` almost consistently achieves lower robust accuracy than CAGE across all nine dataset–retention configurations. Averaged over the five compression methods, `CLOUD` reduces the robust accuracy of CAGE by 2.8, 2.9, and 1.3 on GQA under Retain 64, 32, and 16, respectively. The corresponding decreases on TextVQA are 2.0, 0.4, and 0.9, while those on SQA are 0.5, 1.0, and 0.7. Overall, across the nine averaged settings, `CLOUD` lowers robust accuracy by an average of 1.39 compared with CAGE. The absolute clean-to-attack differences exhibit the same trend. Under Retain 64, the differences produced by `CLOUD` are 24.5, 20.2, and 8.3 on GQA, TextVQA, and SQA, compared with 21.7, 18.2, and 7.8 for CAGE. These results indicate that the effectiveness of `CLOUD` is not restricted to LLaVA or Qwen3-VL, but transfers to a different LVLM architecture and remains stable across substantially different compression strategies

Table 5: Attack performance (%) of different visual token compression methods on InternVL3.5 across GQA, TextVQA, and SQA. We report clean and robust accuracy under Base, CAGE, and CLOUD. Arrow annotations indicate the absolute change from the corresponding clean accuracy.

Compression	GQA				TextVQA				SQA			
Method	Clean	Base	CAGE	Ours	Clean	Base	CAGE	Ours	Clean	Base	CAGE	Ours
Retain 64 Tokens ($K_{\text{model}} = 64$)												
VisionZip	63.4	37.4↓ 26.0	40.5↓ 22.9	38.3↓ 25.1	53.7	31.1↓ 22.6	26.9↓ 26.8	26.0↓ 27.7	89.0	78.9↓ 10.1	79.3↓ 9.7	78.0↓ 11.0
VisPruner	64.2	37.6↓ 26.6	38.0↓ 26.2	36.1↓ 28.1	53.1	30.8↓ 22.3	26.4↓ 26.7	24.6↓ 28.5	89.2	79.5↓ 9.7	78.6↓ 10.6	77.7↓ 11.5
DivPrune	63.4	40.3↓ 23.1	42.3↓ 21.1	40.2↓ 23.2	51.6	31.2↓ 20.4	29.5↓ 22.1	26.2↓ 25.4	86.2	77.7↓ 8.5	78.2↓ 8.0	78.2↓ 8.0
FlowCut	66.0	39.5↓ 26.5	45.0↓ 21.0	41.3↓ 24.7	21.7	14.3↓ 7.4	15.4↓ 6.3	13.4↓ 8.3	83.4	77.4↓ 6.0	77.7↓ 5.7	77.6↓ 5.8
PruMerge	60.0	37.5↓ 22.5	42.8↓ 17.2	38.8↓ 21.2	28.9	17.3↓ 11.6	19.9↓ 9.0	17.6↓ 11.3	81.9	76.3↓ 5.6	76.7↓ 5.2	76.4↓ 5.5
Average	63.4	**38.5**↓ 24.9	41.7↓ 21.7	38.9↓ 24.5	41.8	24.9↓ 16.9	23.6↓ 18.2	**21.6**↓ 20.2	85.9	78.0↓ 7.9	78.1↓ 7.8	**77.6**↓ 8.3
Retain 32 Tokens ($K_{\text{model}} = 32$)												
VisionZip	58.8	36.7↓ 22.1	38.4↓ 20.4	35.8↓ 23.0	40.8	23.6↓ 17.2	18.7↓ 22.1	19.6↓ 21.2	84.6	78.5↓ 6.1	77.6↓ 7.0	77.4↓ 7.2
VisPruner	56.3	35.8↓ 20.5	36.4↓ 19.9	34.0↓ 22.3	40.4	24.5↓ 15.9	18.5↓ 21.9	18.4↓ 22.0	83.8	76.9↓ 6.9	77.4↓ 6.4	76.9↓ 6.9
DivPrune	57.9	37.7↓ 20.2	39.7↓ 18.2	38.3↓ 19.6	42.3	24.2↓ 18.1	22.4↓ 19.9	23.0↓ 19.3	81.9	77.7↓ 4.2	77.0↓ 4.9	77.5↓ 4.4
FlowCut	59.5	37.7↓ 21.8	42.2↓ 17.3	37.8↓ 21.7	19.0	13.1↓ 5.9	14.7↓ 4.3	12.5↓ 6.5	79.4	77.7↓ 1.7	78.5↓ 0.9	76.1↓ 3.3
PruMerge	55.4	35.9↓ 19.5	41.4↓ 14.0	37.4↓ 18.0	23.0	14.7↓ 8.3	15.7↓ 7.3	14.5↓ 8.5	78.4	76.9↓ 1.5	77.8↓ 0.6	75.8↓ 2.6
Average	57.6	36.8↓ 20.8	39.6↓ 18.0	**36.7**↓ 20.9	33.1	20.0↓ 13.1	18.0↓ 15.1	**17.6**↓ 15.5	81.6	77.5↓ 4.1	77.7↓ 3.9	**76.7**↓ 4.9
Retain 16 Tokens ($K_{\text{model}} = 16$)												
VisionZip	54.6	36.9↓ 17.7	34.8↓ 19.8	34.7↓ 19.9	32.5	18.7↓ 13.8	16.2↓ 16.3	14.7↓ 17.8	82.0	77.2↓ 4.8	75.8↓ 6.2	76.3↓ 5.7
VisPruner	53.1	35.4↓ 17.7	33.1↓ 20.0	33.4↓ 19.7	31.8	17.8↓ 14.0	15.1↓ 16.7	13.6↓ 18.2	82.5	77.9↓ 4.6	76.3↓ 6.2	75.8↓ 6.7
DivPrune	52.8	37.5↓ 15.3	37.4↓ 15.4	36.4↓ 16.4	33.1	19.1↓ 14.0	19.1↓ 14.0	19.7↓ 13.4	78.8	76.6↓ 2.2	75.9↓ 2.9	77.0↓ 1.8
FlowCut	52.6	33.5↓ 19.1	40.3↓ 12.3	37.3↓ 15.3	16.0	12.5↓ 3.5	14.5↓ 1.5	12.7↓ 3.3	77.5	77.3↓ 0.2	78.1↑ 0.6	76.6↓ 0.9
PruMerge	51.8	35.9↓ 15.9	38.7↓ 13.1	36.0↓ 15.8	19.6	14.7↓ 4.9	14.2↓ 5.4	13.7↓ 5.9	78.6	76.2↓ 2.4	78.4↓ 0.2	75.5↓ 3.1
Average	53.0	35.8↓ 17.2	36.9↓ 16.1	**35.6**↓ 17.4	26.6	16.6↓ 10.0	15.8↓ 10.8	**14.9**↓ 11.7	79.9	77.0↓ 2.9	76.9↓ 3.0	**76.2**↓ 3.7
Blind ($K_{\text{model}} = 0$)												
Blind		17.3				6.0				75.0		

Table 6: Attack performance (%) of different visual token compression methods on GLM-4.6V-Flash across GQA, TextVQA, and SQA. We report clean and robust accuracy under Base, CAGE, and CLOUD. Arrow annotations indicate the absolute change from the corresponding clean accuracy.

Compression	GQA				TextVQA				SQA			
Method	Clean	Base	CAGE	Ours	Clean	Base	CAGE	Ours	Clean	Base	CAGE	Ours
Retain 72 Tokens ($K_{\text{model}} = 72$)												
VisionZip	37.4	15.0↓ 22.4	19.4↓ 18.0	16.3↓ 21.1	48.2	14.3↓ 33.9	20.5↓ 27.7	10.9↓ 37.3	26.4	20.9↓ 5.5	20.1↓ 6.3	21.0↓ 5.4
VisPruner	36.7	15.0↓ 21.7	15.3↓ 21.4	13.4↓ 23.3	34.2	12.0↓ 22.2	14.8↓ 19.4	9.2↓ 25.0	26.8	21.0↓ 5.8	20.3↓ 6.5	21.0↓ 5.8
DivPrune	38.2	15.0↓ 23.2	21.4↓ 16.8	14.0↓ 24.2	61.9	15.5↓ 46.4	20.9↓ 41.0	11.6↓ 50.3	26.3	20.0↓ 6.3	19.4↓ 6.9	20.2↓ 6.1
FlowCut	36.6	15.9↓ 20.7	21.9↓ 14.7	16.3↓ 20.3	9.8	11.4↑ 1.6	19.0↑ 9.2	9.9↑ 0.1	27.6	21.5↓ 6.1	20.6↓ 7.0	20.7↓ 6.9
PruMerge	34.6	16.7↓ 17.9	21.3↓ 13.3	16.9↓ 17.7	33.1	11.5↓ 21.6	17.6↓ 15.5	10.6↓ 22.5	26.7	21.9↓ 4.8	21.8↓ 4.9	20.2↓ 6.5
Average	36.7	15.5↓ 21.2	19.9↓ 16.8	**15.4**↓ 21.3	37.4	12.9↓ 24.5	18.6↓ 18.8	**10.4**↓ 27.0	26.8	21.1↓ 5.7	**20.4**↓ 6.4	20.6↓ 6.2
Retain 36 Tokens ($K_{\text{model}} = 36$)												
VisionZip	34.0	14.9↓ 19.1	17.3↓ 16.7	13.7↓ 20.3	34.5	12.2↓ 22.3	16.1↓ 18.4	8.5↓ 26.0	25.4	20.3↓ 5.1	20.0↓ 5.4	20.2↓ 5.2
VisPruner	34.7	13.4↓ 21.3	13.3↓ 21.4	12.8↓ 21.9	20.6	8.8↓ 11.8	10.2↓ 10.4	6.8↓ 13.8	24.0	20.4↓ 3.6	20.2↓ 3.8	21.5↓ 2.5
DivPrune	35.3	14.8↓ 20.5	18.6↓ 16.7	13.8↓ 21.5	47.3	13.5↓ 33.8	17.4↓ 29.9	10.1↓ 37.2	25.8	21.4↓ 4.4	18.7↓ 7.1	21.0↓ 4.8
FlowCut	32.6	15.7↓ 16.9	22.2↓ 10.4	16.5↓ 16.1	9.2	9.4↑ 0.2	15.7↑ 6.5	9.8↑ 0.6	24.2	21.9↓ 2.3	19.6↓ 4.6	22.0↓ 2.2
PruMerge	31.9	18.0↓ 13.9	20.0↓ 11.9	16.3↓ 15.6	23.4	9.6↓ 13.8	15.8↓ 7.6	9.1↓ 14.3	26.0	21.9↓ 4.1	21.2↓ 4.8	21.8↓ 4.2
Average	33.7	15.4↓ 18.3	18.3↓ 15.4	**14.6**↓ 19.1	27.0	10.7↓ 16.3	15.0↓ 12.0	**8.9**↓ 18.1	25.1	21.2↓ 3.9	**19.9**↓ 5.2	21.3↓ 3.8
Blind ($K_{\text{model}} = 0$)												
Blind		23.2				6.0				12.6		

and retention budgets. The different attack margins across datasets are also consistent with their dependence on visual information. In the Blind setting, the accuracies are 17.3 on GQA, 6.0 on TextVQA, and 75.0 on SQA. This suggests that SQA contains considerably less attackable visual evidence under the evaluated setting, leaving substantially less room for visual perturbations to further degrade performance.

D.2 Generalization to GLM-4.6V-Flash

We further evaluate the generalization of CLOUD on GLM-4.6V-Flash across five visual token compression methods under $K_{\text{model}} \in \{36, 72\}$, with results reported in Table 6. CLOUD consistently achieves strong attack performance on GQA and TextVQA, although several GQA results are close to those of the baselines. The improvement is particularly pronounced on TextVQA, where CLOUD lowers the average robust accuracy of CAGE by 8.2 under Retain 72 and by 6.1 under Retain 36. On SQA, the three attacks achieve relatively similar performance. Overall, these results demonstrate that CLOUD generalizes to GLM-4.6V-Flash and remains effective across different compression strategies and retention budgets.

Table 7: Ablation study of Bayesian optimization and surrogate aggregation on LLaVA with TextVQA and VisionZip. We report clean and robust accuracy at different visual-token retention budgets. Arrow annotations indicate the absolute decrease from clean accuracy.

Variant	$K_{\text{model}} = 16$	$K_{\text{model}} = 32$	$K_{\text{model}} = 64$	Average
Clean	43.8	48.1	53.8	48.6
Untuned + Median	13.1↓ 30.7	**13.8**↓ 34.3	17.0↓ 36.8	**14.6**↓ 34.0
BO + Mean	13.7↓ 30.1	15.0↓ 33.1	18.1↓ 35.7	15.6↓ 33.0
BO + Median (Ours)	**12.7**↓ 31.1	15.3↓ 32.8	**15.9**↓ 37.9	**14.6**↓ 34.0

D.3 Additional Ablation Study

Table 7 separately examines the effects of Bayesian optimization (BO) and median-based surrogate aggregation. Replacing median aggregation with mean aggregation increases the average robust accuracy by 1.0, demonstrating that the median consensus provides a stronger and more robust attack, particularly at retention budgets of 16 and 64 tokens. These results indicate that median aggregation provides the clearer overall contribution, while BO mainly improves performance at the boundary retention settings.

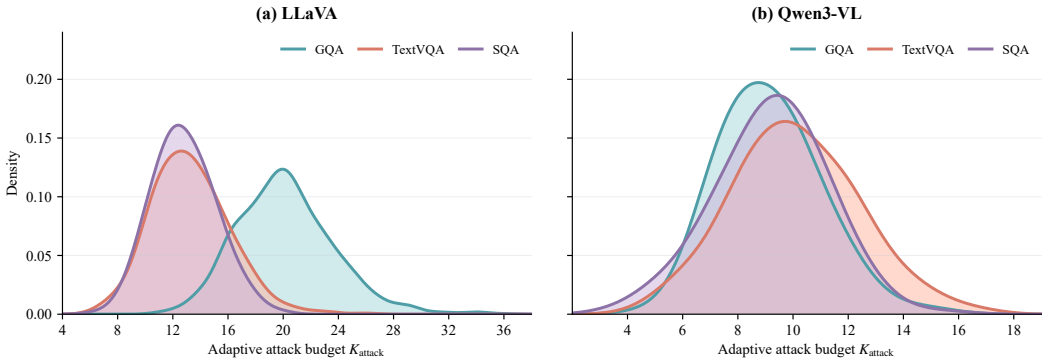


Figure 6: **Distribution of sample-adaptive attack budgets.** Distribution of the K_{attack} of GQA, TextVQA, and SQA for LLaVA-1.5-7B and Qwen3-VL-8B with VisionZip. K_{attack} is determined independently for each sample, illustrating the sample-dependent attack budgets produced by our compression-aware profiling.

D.4 Analysis of Sample-Adaptive Attack Budgets

To further examine how compression-aware profiling adapts the attack support to individual samples, we visualize the distribution of the automatically determined K_{attack} on GQA, TextVQA, and SQA. We conduct the analysis on these three datasets using LLaVA-1.5-7B and Qwen3-VL-8B with VisionZip. As shown in Fig. 6, the resulting attack budgets exhibit clear sample-dependent variation. For LLaVA-1.5-7B, GQA tends to require larger attack supports than TextVQA and SQA, while the distributions on Qwen3-VL-8B are more concentrated within a relatively smaller range. Meanwhile, substantial variation remains among samples within the same model and dataset. The distributions provide direct evidence for the sample-adaptive design of compression-aware profiling. Rather than approximating an unknown compression budget with a globally fixed K_{attack}, `CLOUD` estimates the amount of attack support from the compression responses of each input. Consequently, samples requiring broader intervention can receive larger support, whereas samples with concentrated vulnerable evidence can be attacked with fewer tokens. These observations indicate that the effective perturbation support is not supposed to be adequately characterized by a single fixed budget shared across all inputs to avoid imposing the same perturbation structure on samples with different compression sensitivities.

D.5 Analysis of Cross-Surrogate Complementarity

We further compare the visual-token subsets selected by attention, similarity, activation and merge to investigate whether different compression surrogates provide redundant or complementary profiling signals. In experiments on TextVQA with LLaVA-1.5-7B and VisionZip, each surrogate indepen-

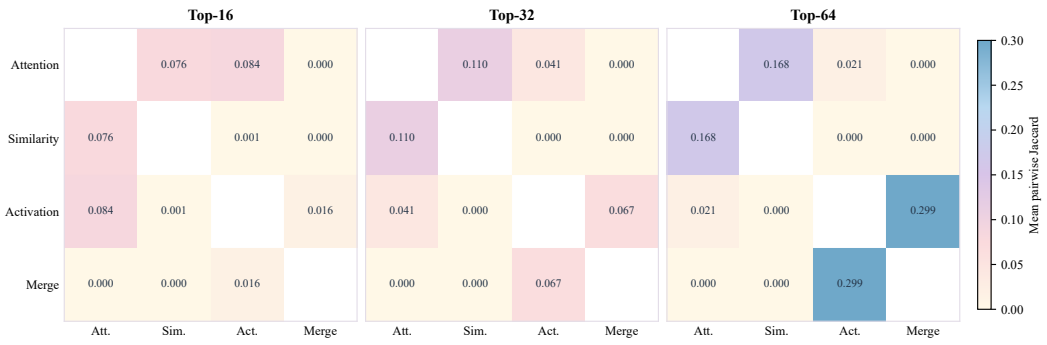


Figure 7: **Complementarity across compression surrogates.** Experiments are conducted on TextVQA with LLaVA-1.5-7B and VisionZip. Attention, Similarity, Activation, and Merge independently select their Top-K visual tokens for $K \in \{16, 32, 64\}$. Each heatmap reports the mean pairwise Jaccard similarity between the resulting token sets.

dently selects its Top-16, Top-32, and Top-64 tokens, and we compute the mean pairwise Jaccard similarity between the resulting token sets. The results shown in Fig. 7 indicate that most surrogate pairs exhibit limited overlap across all three retention budgets. Specifically, attention and merge illustrate essentially no overlap. The low cross-surrogate overlap suggests that relying on a single compression surrogate can provide an incomplete characterization of compression-sensitive evidence. Attention and similarity capture visually salient tokens and redundancy-related structure, respectively. Activation reflects feature responses and merge characterizes evidence under token aggregation. These complementary selections allow the surrogate pool to expose multiple possible compression behaviors.

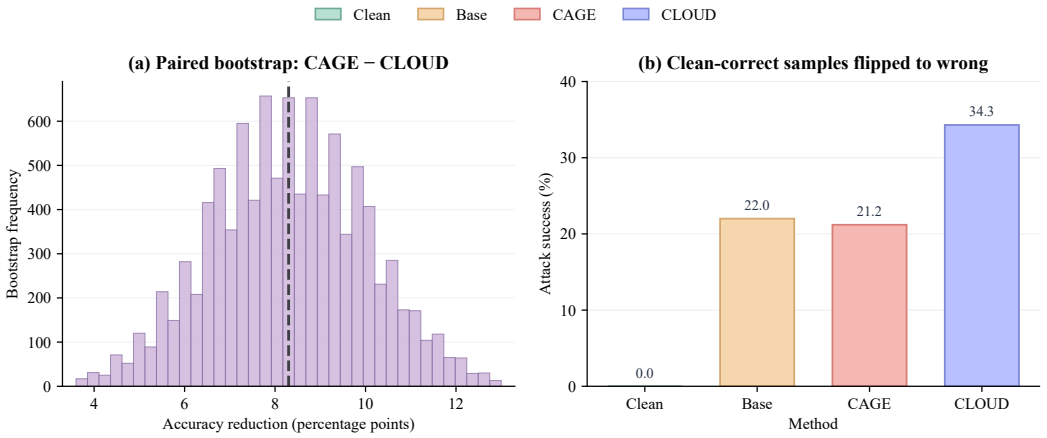


Figure 8: **Paired comparison between CAGE and `CLOUD`.** Results on the TextVQA using Qwen3-VL-8B with FlowCut at Retain 72. **Left**: paired-bootstrap distribution of the CAGE–`CLOUD` accuracy difference, with the dashed line indicating the mean paired difference. **Right**: attack success rate on the 528 samples that are correctly answered by the clean model, measured as the share of clean-correct samples flipped to incorrect predictions by Base, CAGE, and `CLOUD`.

D.6 Paired Analysis of Attack Effectiveness

To further examine whether the advantage of `CLOUD` is consistently observed at the sample level rather than arising only from aggregate accuracy, we conduct a paired analysis against CAGE. Experiments are performed on TextVQA using Qwen3-VL-8B with FlowCut at Retain 72. We first bootstrap paired samples and measure the accuracy reduction difference between CAGE and `CLOUD`. The paired-bootstrap distribution of CAGE–`CLOUD` accuracy reduction remains consistently positive and is concentrated around a substantial performance gap, indicating that the observed difference is not driven by a small subset of sampled examples, as shown in Fig. 8. `CLOUD` induces considerably more prediction failures among samples for which the clean model originally produces correct

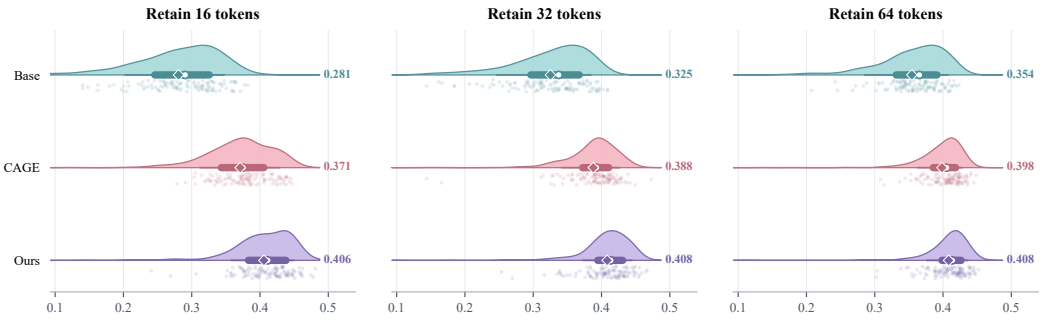


Figure 9: **Feature Discrepancy after Token Compression. Left.** VisionZip with 16 retained tokens. **Middle.** VisionZip with 32 retained tokens. **Right.** VisionZip with 64 retained tokens. Diamonds denote the mean optimal transport discrepancy between clean and adversarial visual representations.

answers. These two views provide complementary evidence about attack effectiveness. The paired bootstrap evaluates the stability of the performance difference under sample resampling, while the clean-correct analysis directly measures whether an attack can overturn originally successful predictions. The improvement in both analyses suggests that `CLOUD` does not merely exploit differences in the aggregate error rate; its compression-aware support allocation and coordinated distributional disruption more consistently remove or distort the visual evidence required for correct prediction.

D.7 Analysis of Post-Compression Feature Discrepancy

We next examine how much representation discrepancy remains after visual token compression. For each adversarial example, we apply VisionZip to both the clean and perturbed visual representations and measure their discrepancy using optimal transport. Fig. 9 presents the resulting distributions under retention budgets of 16, 32, and 64 tokens. The mean discrepancies of `CLOUD` are 0.406, 0.408, and 0.408 across the three settings. CAGE reaches 0.371, 0.388, and 0.398, while Base obtains 0.281, 0.325, and 0.354. The discrepancy generated by `CLOUD` remains at a similar level as the retention budget changes. Meanwhile, its distribution stays shifted toward larger discrepancy values across all three settings. These observations indicate that the coordinated perturbation continues to induce substantial representation changes in the visual evidence that remains available after compression.

D.8 Analysis of Token-Level Transport

We further visualize the transport structure produced during coordinated distributional disruption. Fig. 10 presents three TextVQA examples with attack support sizes of 11, 13, and 15. The selected support varies across samples in both size and spatial location, reflecting the sample-specific allocation obtained from compression-aware profiling. Given these source tokens, the transport plan distributes their mass across the retained DPP targets. Larger bubbles correspond to stronger conditional assignments and reveal which retained tokens receive more information from each source token. The resulting patterns differ across sources and samples, showing that transport is organized according to their interactions with the retained target set. This process connects the adaptive support produced during profiling with the retained visual evidence considered during perturbation optimization. The resulting token-level structure provides a direct view of how coordinated transport operates before the post-compression representation discrepancy is measured.

D.9 Visualization of Sample-Adaptive and Coordinated Support

We further visualize how `CLOUD` converts compression-aware profiling into sample-specific attack support. Figures 11 and 12 present representative examples from GQA, TextVQA, and SQA on LLaVA and Qwen3-VL. For each sample, we first map the allocation scores q_i^{S1} back to the image and visualize the active tokens retained after sparse allocation. The number of active tokens determines the sample-specific attack budget K_{attack}. We then use the same K_{attack} to construct the attention support and the DPP support, which allows us to directly examine how the two selection strategies organize the available attack budget. The visualizations show clear variation across sam-

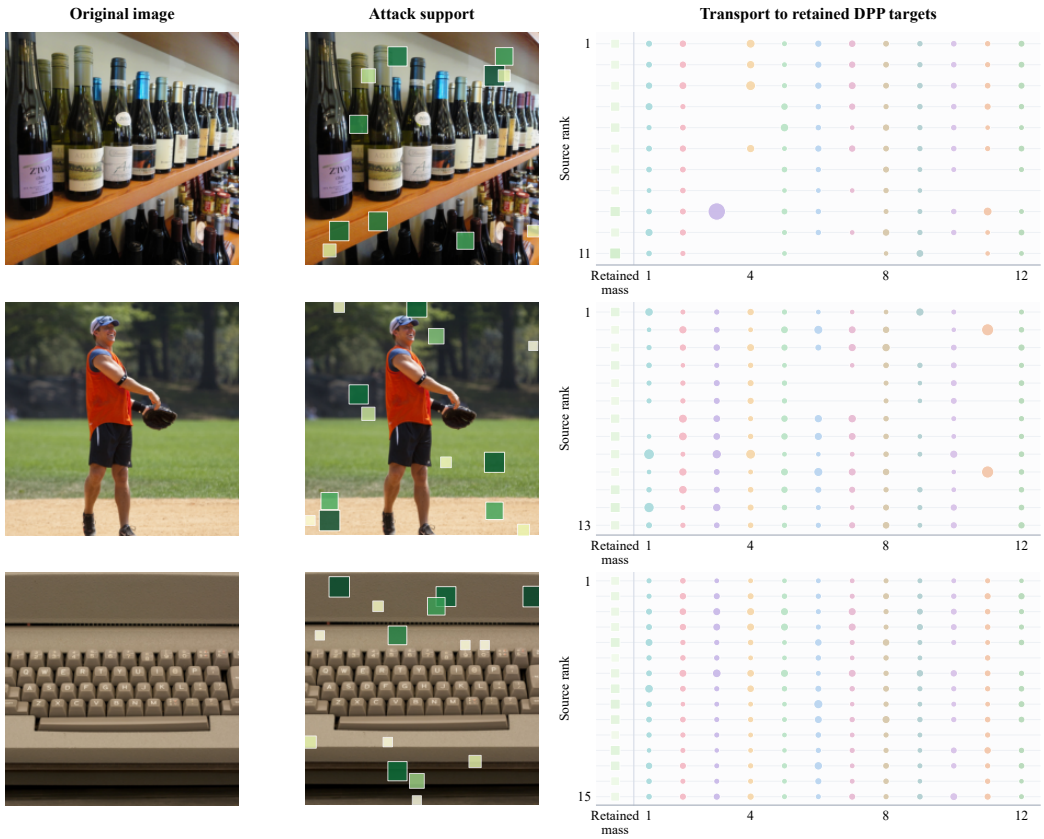


Figure 10: **Token-Level Transport from Attack Support to Retained Tokens.** Rows show three TextVQA examples with different attack support sizes. **Left.** Original images. **Middle.** Adaptively selected attack support. **Right.** Transport from source tokens ranked by attack weight to retained DPP targets. Bubble area denotes the conditional transport proportion, and the first column shows retained mass.

Table 8: **Comparison between fixed-average and sample-adaptive attack support on LLaVA-v1.5-7B with VisionZip and TextVQA.** The value in parentheses denotes the range of adaptive K_{attack}.

Dataset	Scheme	K_{attack}	Retain 16 tokens	Retain 32 tokens	Retain 64 tokens	Average
TextVQA	Fixed Average	13	14.9	**15.2**	18.5	16.20
	Adaptive	13.19 (6–26)	**12.7**	15.3	**15.9**	**14.63**

ples. Some images receive compact support concentrated around a few informative regions, while others activate a broader collection of visual tokens. The resulting K_{attack} therefore changes with the spatial distribution of compression-sensitive evidence. This behavior is visible on both models. The selected LLaVA examples span K_{attack} values from 11 to 22, while the Qwen3-VL examples span values from 8 to 12. These qualitative results agree with the distribution-level analysis of adaptive attack budgets and further show how the variation emerges at the token level. The comparison between Attention and DPP reveals a second difference. Attention tends to allocate several selected tokens to nearby salient regions, which can create substantial spatial redundancy when multiple tokens describe similar evidence. DPP redistributes part of the same budget toward complementary regions while retaining visually informative locations. This pattern appears across object-centered scenes, text-rich images, and spatial reasoning examples. The resulting support therefore covers a broader set of potentially useful visual evidence under the same number of attack tokens. Through this construction, the sample-specific budget estimated in Stage 1 is converted into a coordinated support for the subsequent perturbation optimization.

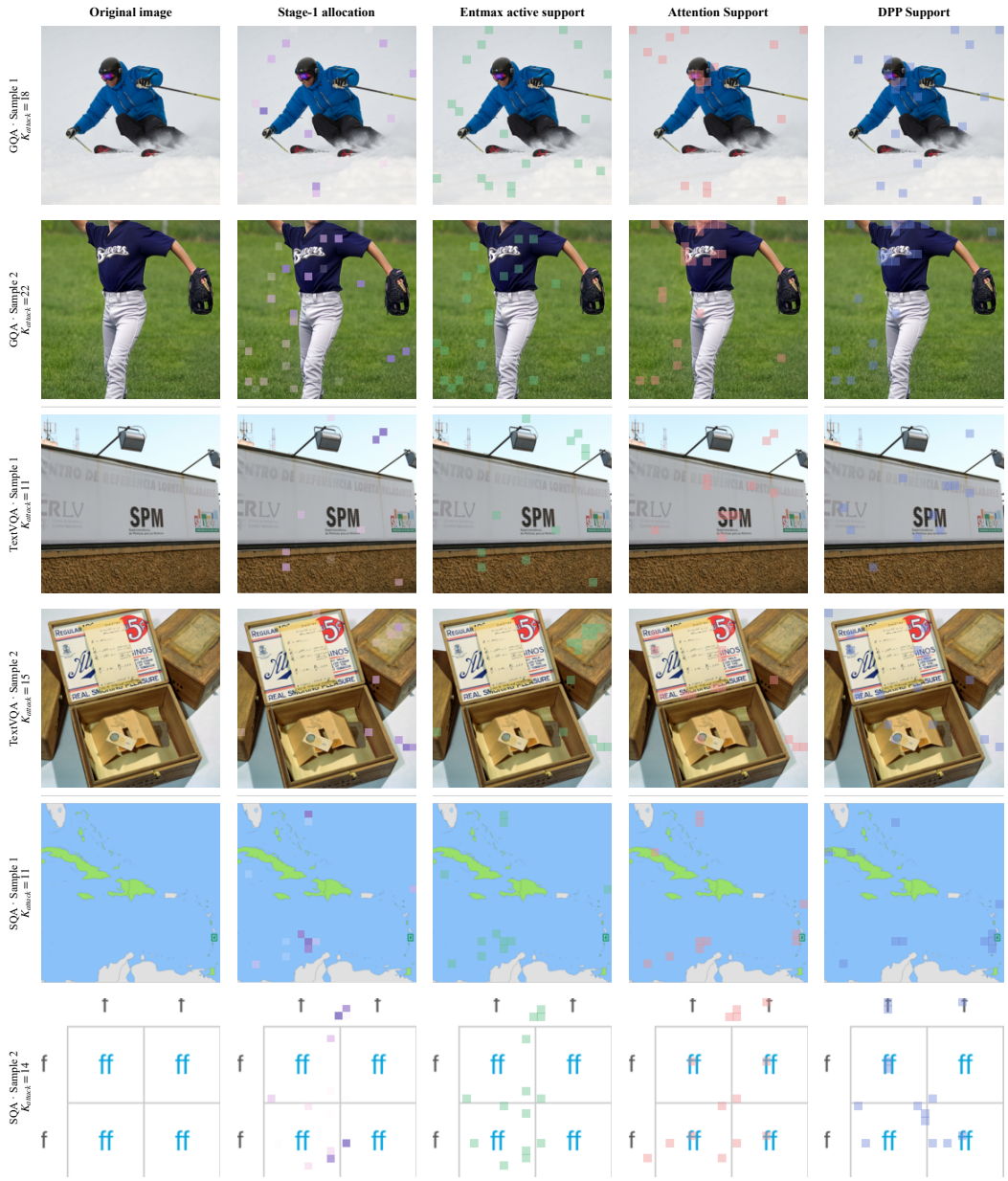


Figure 11: **Qualitative analysis of adaptive support construction on LLaVA.** We visualize representative samples from GQA, TextVQA, and SQA. From left to right, each row shows the original image, Stage 1 allocation, Entmax active support, Attention support, and DPP support. Attention and DPP use the same sample-specific budget K_{attack}. Stage 1 produces different attack budgets across samples, while DPP distributes the selected tokens across more complementary visual regions under the same budget.

D.10 Further Analysis of Adaptive Attack Budget

To examine whether the gain of adaptive allocation comes from sample-specific support rather than a larger attack budget, we compare it with a fixed support using the same average K_{attack}. As shown in Table 8, the adaptive scheme uses an average K_{attack} of 13.19, close to the fixed budget of 13, while reducing the average robust accuracy by 1.57. The decreases under Retain 16 and Retain 64 are 2.2 and 2.6, respectively. Meanwhile, the adaptive budget varies from 6 to 26 across samples, showing that different inputs require different attack support even under the same compression set-

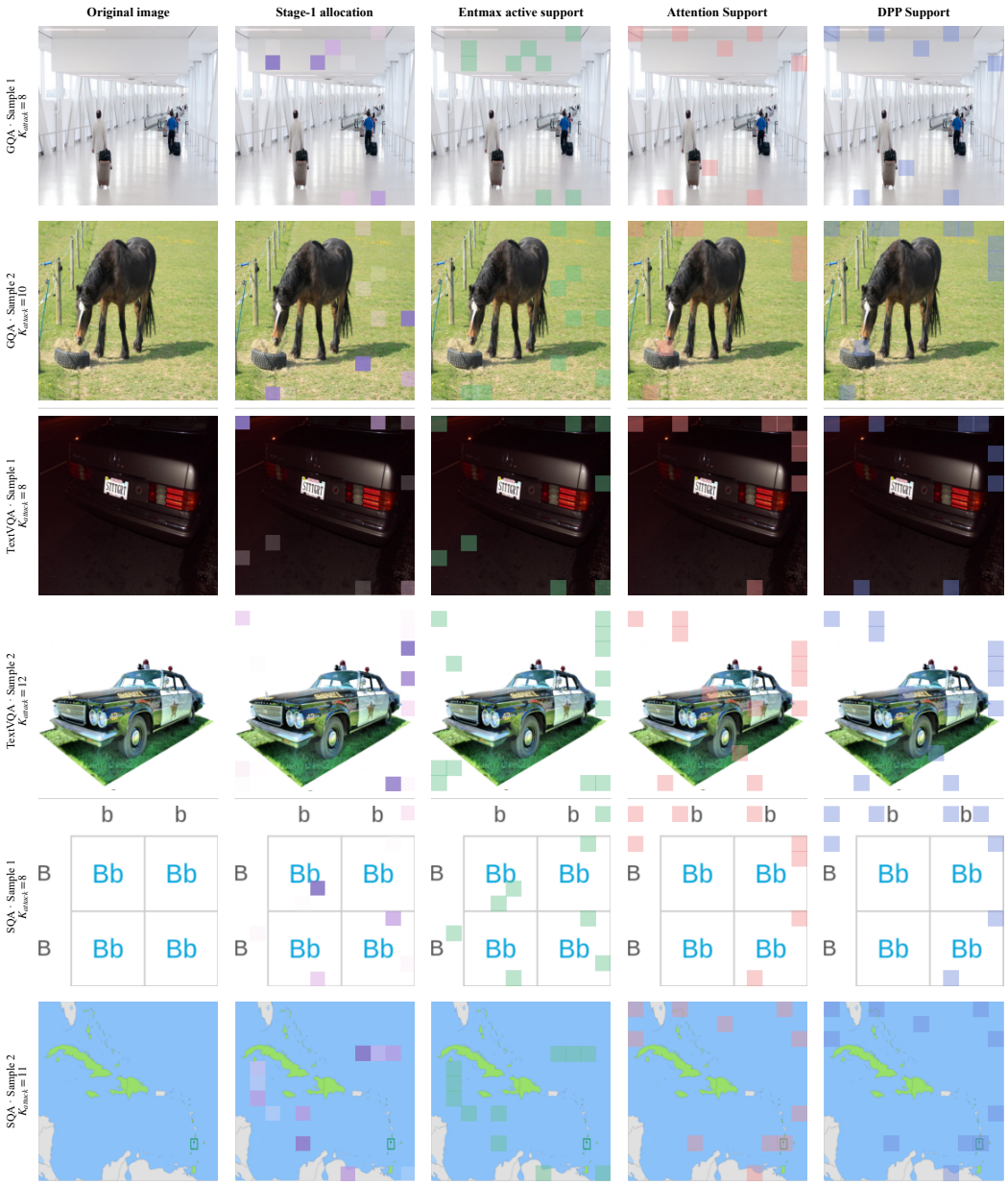


Figure 12: **Qualitative analysis of adaptive support construction on Qwen3-VL.** We visualize representative samples from GQA, TextVQA, and SQA. From left to right, each row shows the original image, Stage 1 allocation, Entmax active support, Attention support, and DPP support. Attention and DPP use the same sample-specific budget K_{attack}. Stage 1 adapts the attack budget to each sample, while DPP organizes the resulting support across complementary visual regions under the same budget.

ting. These results validate the benefit of sample-adaptive budget calibration beyond using a fixed average support.

D.11 Different Perturbation Budgets

We evaluate the perturbation performance under $\epsilon \in \{2/255, 4/255, 8/255\}$ across three compression budgets, as shown in Table 9. Among all the perturbation budgets, `CLOUD` achieves comparable average robust accuracies. The performance gap is most pronounced under the smallest perturbation budget, where attack effectiveness depends more strongly on how the limited perturbation is

Table 9: Attack performance (%) under different perturbation budgets on LLaVA with TextVQA. We report clean accuracy and robust accuracy under Base, CAGE, and CLOUD for $\epsilon \in \{2/255, 4/255, 8/255\}$. Lower robust accuracy indicates a stronger attack. Arrow annotations indicate the absolute decrease from the corresponding clean accuracy.

Compression Method	Clean	$\epsilon = 2/255$			$\epsilon = 4/255$			$\epsilon = 8/255$		
		Base	CAGE	Ours	Base	CAGE	Ours	Base	CAGE	Ours
Retain 64 Tokens ($K_{\text{model}} = 64$)										
VisionZip	53.8	31.3↓ 22.5	18.5↓ 35.3	15.9↓ 37.9	20.7↓ 33.1	13.6↓ 40.2	13.8↓ 40.0	15.4↓ 38.4	11.8↓ 42.0	12.1↓ 41.7
VisPruner	54.7	32.5↓ 22.2	20.0↓ 34.7	17.3↓ 37.4	19.9↓ 34.8	13.8↓ 40.9	11.9↓ 42.8	15.6↓ 39.1	12.1↓ 42.6	11.4↓ 43.3
DivPrune	51.2	29.9↓ 21.3	22.4↓ 28.8	18.0↓ 33.2	20.3↓ 30.9	15.6↓ 35.6	16.1↓ 35.1	15.4↓ 35.8	13.7↓ 37.5	12.7↓ 38.5
FlowCut	53.8	32.3↓ 21.5	21.5↓ 32.3	20.2↓ 33.6	20.5↓ 33.3	14.6↓ 39.2	14.7↓ 39.1	15.6↓ 38.2	11.7↓ 42.1	11.9↓ 41.9
PruMerge	49.5	31.3↓ 18.2	19.4↓ 30.1	17.0↓ 32.5	18.7↓ 30.8	14.5↓ 35.0	15.9↓ 33.6	15.3↓ 34.2	13.7↓ 35.8	12.0↓ 37.5
Average	52.6	31.5↓ 21.1	20.4↓ 32.2	**17.7**↓ 34.9	20.0↓ 32.6	**14.4**↓ 38.2	14.5↓ 38.1	15.5↓ 37.1	12.6↓ 40.0	**12.0**↓ 40.6
Retain 32 Tokens ($K_{\text{model}} = 32$)										
VisionZip	48.1	29.9↓ 18.2	16.8↓ 31.3	15.3↓ 32.8	18.7↓ 29.4	13.1↓ 35.0	12.7↓ 35.4	15.6↓ 32.5	12.4↓ 35.7	11.1↓ 37.0
VisPruner	50.8	32.6↓ 18.2	19.0↓ 31.8	15.4↓ 35.4	18.7↓ 32.1	13.6↓ 37.2	13.0↓ 37.8	15.9↓ 34.9	11.6↓ 39.2	11.7↓ 39.1
DivPrune	46.5	28.0↓ 18.5	19.7↓ 26.8	17.0↓ 29.5	18.0↓ 28.5	15.2↓ 31.3	15.9↓ 30.6	15.0↓ 31.5	12.3↓ 34.2	13.0↓ 33.5
FlowCut	49.7	30.3↓ 19.4	20.4↓ 29.3	17.1↓ 32.6	18.8↓ 30.9	13.4↓ 36.3	12.2↓ 37.5	14.9↓ 34.8	11.2↓ 38.5	11.7↓ 38.0
PruMerge	45.3	28.7↓ 16.6	18.8↓ 26.5	15.3↓ 30.0	19.1↓ 26.2	13.6↓ 31.7	13.9↓ 31.4	16.4↓ 28.9	13.4↓ 31.9	12.0↓ 33.3
Average	48.1	29.9↓ 18.2	18.9↓ 29.2	**16.0**↓ 32.1	18.7↓ 29.4	13.8↓ 34.3	**13.5**↓ 34.6	15.6↓ 32.5	12.2↓ 35.9	**11.9**↓ 36.2
Retain 16 Tokens ($K_{\text{model}} = 16$)										
VisionZip	43.8	26.7↓ 17.1	16.1↓ 27.7	12.7↓ 31.1	18.8↓ 25.0	12.7↓ 31.1	12.8↓ 31.0	15.0↓ 28.8	11.5↓ 32.3	10.6↓ 33.2
VisPruner	43.2	27.5↓ 15.7	16.9↓ 26.3	12.8↓ 30.4	17.4↓ 25.8	12.7↓ 30.5	12.1↓ 31.1	14.5↓ 28.7	12.3↓ 30.9	10.1↓ 33.1
DivPrune	37.5	24.1↓ 13.4	17.0↓ 20.5	14.1↓ 23.4	16.9↓ 20.6	12.9↓ 24.6	14.1↓ 23.4	13.7↓ 23.8	12.2↓ 25.3	11.3↓ 26.2
FlowCut	36.4	25.0↓ 11.4	16.7↓ 19.7	15.2↓ 21.2	16.9↓ 19.5	12.9↓ 23.5	12.2↓ 24.2	13.7↓ 22.7	12.5↓ 23.9	12.4↓ 24.0
PruMerge	37.6	26.6↓ 11.0	18.9↓ 18.7	15.5↓ 22.1	19.0↓ 18.6	13.9↓ 23.7	13.0↓ 24.6	15.3↓ 22.3	13.3↓ 24.3	12.6↓ 25.0
Average	39.7	26.0↓ 13.7	17.1↓ 22.6	**14.1**↓ 25.6	17.8↓ 21.9	13.0↓ 26.7	**12.8**↓ 26.9	14.4↓ 25.3	12.4↓ 27.3	**11.4**↓ 28.3

Table 10: Efficiency comparison of Base, CAGE, and CLOUD. We report the average per-sample runtime and peak GPU memory usage measured on LLaVA with TextVQA and VisionZip.

Efficiency	Base	CAGE	Ours
Runtime (s)	1.00	1.03	1.14
Memory usage (GB)	4.26	4.26	4.27

allocated across visual evidence. Adaptive support estimation and coordinated transport therefore remain effective across different perturbation magnitudes, with particularly clear gains when the available perturbation is constrained.

D.12 Efficiency Analysis

Table 10 compares the computational cost of different perturbation methods on LLaVA with TextVQA and VisionZip. CLOUD requires 1.14 seconds per sample, compared with 1.03 seconds for CAGE and 1.00 second for Base. The additional computation mainly comes from compression-aware profiling and coordinated support construction, which introduce only 0.11 second over CAGE. Peak GPU memory remains nearly unchanged, increasing from 4.26 GB to 4.27 GB. These results demonstrate that the adaptive profiling and coordinated disruption in CLOUD only introduce limited computational overhead while preserving almost the same memory usage.