RUTA: PRINCIPLED VISUAL TOKEN ALLOCATION VIA RATE-UTILITY OPTIMIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

High-resolution images and long videos provide vision-language models with rich context for multimodal reasoning and fine-grained perception, but the resulting long visual token sequences make large language model-side computation and memory costly. Existing visual token reducers often operate at prescribed rates, while recent methods adapt token counts across inputs using method-specific learned thresholds or importance predictors. We introduce RUTA, a principled **R**ate-**U**tility **T**oken **A**llocation method that performs pre-LLM reduction by jointly learning which tokens to retain and how many to allocate to each image-query pair. RUTA constructs query-conditioned candidate tokens and predicts a retention probability for each candidate. During training, these probabilities parameterize independent Bernoulli gates, while their sum provides a differentiable training-time estimate of the token count for each pair. Retained tokens serve as anchors that aggregate information from non-retained tokens according to semantic affinity and spatial proximity. RUTA is optimized with a penalized rate-utility objective that balances downstream task loss against expected token usage. Averaged across five benchmarks and measured relative to each backbone's full-token baseline, RUTA uses only 2.0% and 4.2% of visual tokens while preserving 88.2% and 94.4% of task performance on LLaVA-NeXT and Qwen3-VL, respectively.

1 INTRODUCTION

Visual intelligence depends on the fidelity and scope of the evidence available for inference (Marr, 1982). Accordingly, vision-language models (VLMs) increasingly process high-resolution images and long videos to resolve fine-grained visual details and provide broader spatial and temporal context for multimodal reasoning (Guo et al., 2024; Li et al., 2024; Wang et al., 2024). These gains, however, require a corresponding expansion of the visual representation: richer inputs produce longer token sequences that must be processed by the large language model (LLM). In multi-image and long-video settings, the resulting token sequence growth substantially increases LLM-side computation and memory costs (Wu et al., 2024; Zhang et al., 2025c), making visual token efficiency a central constraint on scalable multimodal reasoning.

Visual token reduction addresses this cost by pruning or merging tokens before or during LLM-side processing. The central premise is that useful visual evidence is unevenly distributed across the spatial and temporal extent of visual inputs: answering a query (*i.e.*, a question or instruction about the visual input) may require evidence from only a small region, while large portions of the input remain irrelevant or redundant (Chen et al., 2024a;b; Yang et al., 2025). Existing reducers exploit this structure using query-conditioned relevance (Zhang et al., 2025d), redundancy-aware pruning or merging (Shang et al., 2025; Yang et al., 2025), diversity-preserving selection (Alvar et al., 2025), or progressive layer-wise reduction (Xing et al., 2025).

Despite these advances, many reducers continue to rely on externally specified token counts, retention ratios, or layer-wise schedules (Yang et al., 2025; Alvar et al., 2025; Xing et al., 2025). These controls make computation predictable, but a uniform per-sample budget may overallocate tokens to simple queries while underallocating them to queries requiring broader visual evidence, as illustrated in Fig. 1. Recent methods have therefore begun to vary token usage across inputs or LLM layers using learned thresholds or importance predictors (Ye et al., 2025; Zeng et al., 2026). These studies demonstrate the value of adaptive allocation, but their rate control mechanisms remain tied to

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.

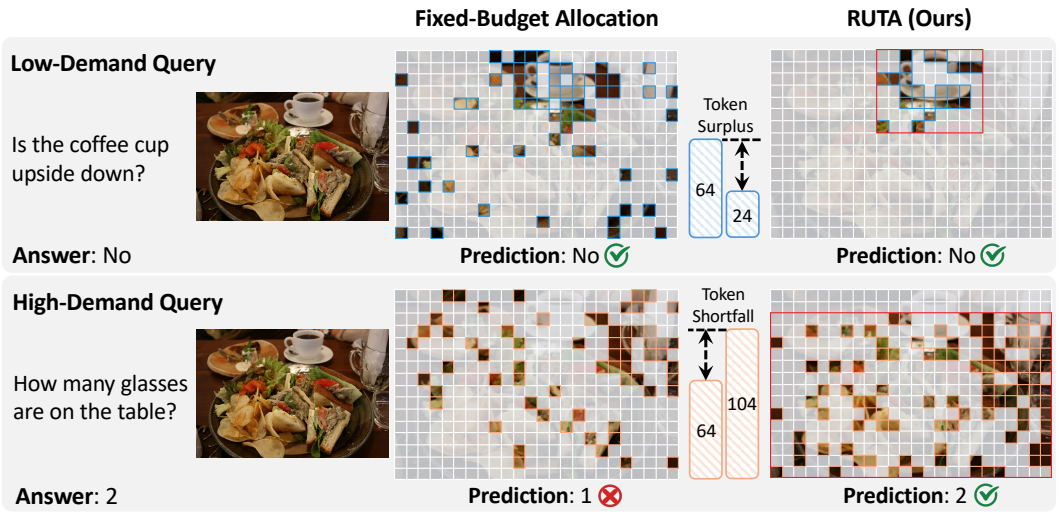


Figure 1: Fixed-budget versus adaptive visual token allocation. A fixed budget retains 64 tokens for each query, leaving surplus capacity for the low-demand query but insufficient evidence for the high-demand query, which leads to an incorrect answer. At the same average rate, RUTA retains 24 and 104 tokens, respectively, adapting token usage to each query's evidence demand and answering both correctly. Red boxes mark the query-conditioned regions of interest.

method-specific objectives and constraints, motivating a unified formulation that explicitly balances token usage against downstream performance.

Drawing inspiration from the rate-distortion theory (Shannon, 1959), we formulate visual token reduction as a rate-utility optimization problem. Rate is measured by the expected number of visual tokens passed to the LLM, whereas utility is represented by downstream task performance. Rather than imposing the same token budget on every image-query pair, we apply a shared penalty to expected token usage. This objective allows the reduction policy to assign different token counts across inputs, while the rate coefficient controls the average rate-utility operating point.

We instantiate this formulation with RUTA, a principled **R**ate-**U**tility **T**oken **A**llocation method. RUTA performs token reduction before LLM processing, allowing token usage to vary with the visual evidence required by each input query. More specifically, RUTA first constructs query-conditioned visual token candidates and assigns each candidate a retention probability, from which discrete token selections are sampled independently during training. The sum of these probabilities gives the expected number of retained tokens for each sample. Retained tokens are treated as anchors to absorb information from non-retained tokens (*i.e.*, non-anchors) according to semantic affinity and spatial proximity. The trainable candidate construction, allocation, and aggregation components in RUTA are jointly optimized via a rate-utility objective that combines downstream task loss with a penalty on expected token usage.

We evaluate RUTA on five different benchmarks using LLaVA-NeXT-7B (Liu et al., 2024a) and Qwen3-VL-8B (Bai et al., 2025). At matched low-token rates, RUTA achieves the highest average task performance among the evaluated reducers on both backbones. In particular, RUTA uses only 2.0% and 4.2% of visual tokens while preserving 88.2% and 94.4% of task performance relative to each backbone's full-token baseline on LLaVA-NeXT-7B and Qwen3-VL-8B, respectively.

In summary, our key contributions are threefold.

1. We formulate visual token reduction as rate-utility optimization, explicitly balancing downstream task performance against expected token usage.
2. We introduce RUTA, which integrates query-conditioned candidate construction, Bernoulli retention, and anchor-based token aggregation, within a jointly trained pre-LLM reduction framework.
3. We demonstrate consistent gains by RUTA at matched low-token rates and favorable rate-utility trade-offs under more severe token reduction.

2 Related Work

2.1 Training-Free Visual Token Reduction

Training-free methods can be organized by whether reduction occurs before or within LLMs. Pre-LLM methods compact the sequence within the vision encoder or at its output, including adaptive token sampling (Fayyaz et al., 2022) and similarity-based token merging (Bolya et al., 2023). In VLMs, importance-based methods retain visually salient or informative tokens (Shang et al., 2025; Yang et al., 2025; Zhang et al., 2025a); diversity-aware methods seek broader visual coverage (Alvar et al., 2025); and PruneSID (Fang et al., 2026) dynamically adjusts the reduction ratio according to estimated image complexity. By compacting the sequence before LLM ingestion, these methods avoid LLM-side computation on removed or merged tokens and remain broadly compatible with existing VLMs. For most of these methods, however, the operating rate is specified through an external token count or retention ratio. In-LLM methods instead defer reduction until multimodal processing has begun. FastV (Chen et al., 2024b) identifies substantial redundancy after shallow LLM layers; SparseVLM (Zhang et al., 2025d) performs text-guided, rank-based layer-adaptive sparsification; ZipVL (He et al., 2025) assigns layer-wise sparsity; PyramidDrop (Xing et al., 2025) progressively drops tokens across layers; and VScan (Zhang et al., 2026) combines global and local visual scanning. Thus, visual token usage can vary across LLM layers even in training-free methods, although the resulting rates are determined by heuristic rules or externally selected operating parameters rather than optimized jointly with downstream task utility.

2.2 Training-Based Visual Token Reduction

Training-based methods use task supervision to learn content-dependent visual computation. In vision Transformers, early methods either aggregate informative content into a small token set or progressively bypass less useful tokens (Ryoo et al., 2021; Rao et al., 2021; Yin et al., 2022). These methods establish adaptive visual processing, but were designed primarily for recognition pipelines rather than VLMs. In VLMs, learned efficiency mechanisms operate at several points between the input image and the LLM. One family uses latent queries to map variable-length visual features into a fixed-capacity interface (Jaegle et al., 2021; 2022; Alayrac et al., 2022; Li et al., 2023; Dai et al., 2023). Other methods improve how spatial details from high-resolution inputs are organized and aggregated (Cha et al., 2024; Tong et al., 2024; Guo et al., 2024; Cai et al., 2025; Li et al., 2025). Efficient vision encoders and compact multimodal interfaces further lower visual encoding cost or LLM input length (Vasu et al., 2025; Zhang et al., 2025b). Together, these designs improve efficiency, but the operating rate is typically built into the interface or selected externally rather than learned jointly with task utility. Modular trainable reducers preserve most of the base VLM. TwigVLM (Shao et al., 2025) adds a shallow branch to guide visual token pruning and accelerate text generation, while CROP (Guo et al., 2025) localizes query-relevant context to guide either pre-LLM reduction or in-LLM pruning. Although supervision makes their selection decisions task-aware, their operating rates generally remain governed by prescribed token counts or retention settings. More recent trainable reducers explicitly adapt visual token counts across inputs. ATP-LLaVA (Ye et al., 2025) predicts sample- and layer-specific pruning thresholds inside the LLM and optimizes them using task, computation, and target-count losses. GlimpsePrune (Zeng et al., 2026) applies a learned importance predictor after several LLM layers to produce a dynamic one-shot pruning mask, optionally subject to a maximum retention ratio.

Together, these methods establish adaptive token allocation as an important design dimension, while relying on method-specific objectives and constraints to determine the operating rate. RUTA builds on this line of work by explicitly formulating visual token reduction as rate-utility optimization, and differs by directly optimizing a differentiable expected pre-LLM token rate and coupling this allocation with query-conditioned candidate construction and anchor-based aggregation.

3 Proposed Method: RUTA

In this section, we introduce RUTA, which uses rate-utility optimization to jointly determine which visual tokens to retain and how many to allocate to each image-query pair. We measure the operating rate by the average number of visual tokens passed to the LLM across image-query pairs and utility by downstream task performance. The system diagram of RUTA is shown in Fig. 2.

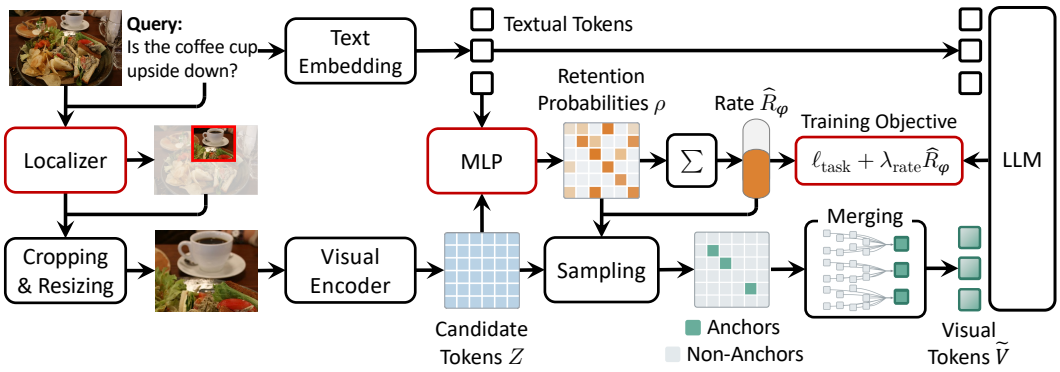


Figure 2: System diagram of RUTA. Given an image and a query, a query-conditioned localizer identifies a region of interest, which is cropped, resized, and encoded into candidate visual tokens Z. A two-layer MLP predicts a retention probability ρ_i for each candidate, and their sum gives the expected token rate $\widehat{R}_\varphi$. During training, independent Bernoulli sampling partitions the candidates into anchors and non-anchors. The non-anchors are then merged into the anchors to produce the compact visual token sequence $\widetilde{V}$. RUTA is trained with a task loss and a rate penalty weighted by λ_{rate}, balancing prediction utility against the number of visual tokens passed to the LLM.

3.1 Problem Formulation

Let (x, q, y) denote an input image, a textual query, and its target output. A standard VLM maps x through a vision encoder E_{vis} to a dense sequence of N visual tokens: $V = E_{\text{vis}}(x) = [v_i]_{i=1}^{N}$, where $v_i \in \mathbb{R}^D$. The LLM conditions on both q and V to predict y. In the pre-LLM setting considered here, let $K(x, q) \leq N$ denote the number of visual tokens passed to the LLM for an image-query pair. For a pre-trained VLM, all visual tokens entering the LLM have the same embedding dimension D. Although different layers may use different bit depths under a static mixed-precision inference configuration, the layerwise precision assignment is fixed across inputs (Dong et al., 2019; Shen et al., 2020; Yao et al., 2022; Frantar et al., 2023). *We therefore use token count as a model-relative measure of rate and a proxy for LLM-side computation and memory.*

At the LLM interface, a pre-LLM reduction policy f_θ, parameterized by vector θ, replaces the dense encoder output V with an input-dependent compact sequence:

$$\widetilde{V} = f_\theta(x, q) = [\tilde{v}_i]_{i=1}^{K(x,q)}. \tag{1}$$

Here, f_θ denotes the complete visual processing path before the LLM and may include or reuse E_{vis}. The LLM receives its output $\widetilde{V}$ in place of the baseline dense sequence V. Because the token count $K(x, q)$ depends on the image-query pair, the policy can allocate more tokens when broader visual evidence is required and fewer tokens when the relevant evidence is concentrated.

An in-LLM reducer instead operates within the LLM and may process different numbers of visual tokens across layers. If $K_l(x, q)$ denotes the number of visual tokens processed at layer l in an LLM with L layers, its rate can be summarized by the layer-wise average $\frac{1}{L}\sum_{l=1}^{L} K_l(x, q)$.

Following the rate-distortion theory (Shannon, 1959), we use a differentiable token rate estimate $\widehat{R}_\theta(x, q)$ and solve the following rate-utility optimization problem:

$$\min_\theta \mathbb{E}_{(x,q,y)} \left[\ell_{\text{task}}\big(\text{LLM}(q, f_\theta(x, q)), y\big) + \lambda_{\text{rate}} \widehat{R}_\theta(x, q) \right], \tag{2}$$

where λ_{rate} controls the trade-off between the two terms. RUTA further decomposes the reduction policy f_θ into a query-conditioned candidate constructor g_ϕ and a rate-controlled compressor h_φ:

$$Z = g_\phi(x, q), \qquad \widetilde{V} = h_\varphi(Z, q), \tag{3}$$

where $\theta = (\phi, \varphi)$ collects the trainable parameters of the candidate constructor and compressor. Consequently, $f_\theta(x, q) = h_\varphi(g_\phi(x, q), q)$ and $\widehat{R}_\theta(x, q) = \widehat{R}_\varphi(g_\phi(x, q), q)$. The constructor g_ϕ concentrates the candidate sequence on query-relevant visual content, while the compressor h_φ assigns

each candidate a retention probability, which guides anchor selection. The remaining candidates are then aggregated into the selected anchors to form the compact sequence passed to the LLM. The sum of the same probabilities also defines the sample-specific expected rate.

3.2 Query-Conditioned Candidate Token Construction

The candidate constructor g_ϕ first identifies a query-relevant image region, resizes it to the vision encoder's standard input resolution, and encodes it into candidate tokens. This concentrates the candidate tokens on relevant visual evidence before token reduction. Specifically, given the query q, RUTA extracts $R(q)$ phrases for grounding relevant image regions via $\{u_j\}_{j=1}^{R(q)} = \mathrm{parser}(q)$. The $\mathrm{parser}(\cdot)$ removes question templates and other non-visual words, retaining noun phrases and attributes that describe image content. A phrase-conditioned localizer d_ϕ then returns a bounding box b_j and confidence score c_j for each phrase u_j:

$$\{(b_j, c_j)\}_{j=1}^{R(q)} = d_\phi\left(x, \{u_j\}_{j=1}^{R(q)}\right). \tag{4}$$

The localizer may be instantiated by any open-vocabulary grounding model, and we adopt Grounding DINO (Liu et al., 2024b) by default. Using confidence threshold δ, RUTA forms a set of candidate boxes and, if this set is nonempty, takes their union as the query-conditioned region of interest:

$$\mathcal{B}_\delta = \{b_j \mid c_j \geq \delta\}, \qquad b_{\mathrm{roi}} = \mathrm{Union}(\mathcal{B}_\delta). \tag{5}$$

Specifically, $\mathrm{Union}(\cdot)$ returns the smallest axis-aligned bounding box enclosing all boxes in $\mathcal{B}_\delta$. We use a single enclosing region rather than processing multiple regions separately because it preserves potentially useful *intervening* evidence and the spatial relationships among the localized regions, while preventing the number of encoder views—and hence candidate tokens—from scaling with the number of grounded regions. When $\mathcal{B}_\delta = \emptyset$, RUTA skips cropping and passes the full image to the vision encoder. RUTA resizes and encodes the selected content as the candidate sequence Z:

$$Z = g_\phi(x, q) = E_{\mathrm{vis}}\big(\mathrm{Resize}(\mathrm{Crop}(x, b_{\mathrm{roi}}))\big) = [z_i]_{i=1}^{N_c}, \qquad z_i \in \mathbb{R}^D. \tag{6}$$

Here, resizing preserves the base VLM input resolution, so candidate construction does not necessarily shorten the visual sequence: N_c remains comparable to the original dense token count N.[1] Instead, it reallocates the encoder's spatial resolution to query-relevant content, while the subsequent compressor h_φ performs the actual token reduction.

3.3 Rate-Controlled Compression

Given $Z = [z_i]_{i=1}^{N_c}$, the compressor h_φ performs three steps: it predicts a retention probability for each candidate, samples independent binary gates, and merges information from non-retained tokens into the retained anchors. The probabilities jointly determine which tokens are preferred and how many are expected to remain. Within the compressor h_φ, a two-layer query-conditioned multilayer perceptron (MLP) s_φ induces each candidate a retention probability:

$$\rho_i = \mathrm{sigmoid}\left(\frac{s_\varphi(z_i, q)}{\tau}\right), \qquad i \in \{1, \ldots, N_c\}, \tag{7}$$

where $\tau > 0$ is a temperature controlling the sharpness of these probabilities. During training, RUTA samples an independent Bernoulli gate for every candidate and partitions the candidates into an anchor index set $\mathcal{A}$ and a non-anchor index set $\bar{\mathcal{A}}$:

$$\xi_i \sim \mathrm{Bernoulli}(\rho_i), \qquad \mathcal{A} = \{i \mid \xi_i = 1\}, \qquad \bar{\mathcal{A}} = \{i \mid \xi_i = 0\}. \tag{8}$$

As $|\mathcal{A}| = \sum_i \xi_i$, the expected number of anchors under independent Bernoulli sampling is

$$\mathbb{E}[|\mathcal{A}|] = \sum_{i=1}^{N_c} \rho_i = \widehat{R}_\varphi(Z, q). \tag{9}$$

This differentiable quantity serves as the training-time rate estimate, avoiding prescribed token budgets. If a Bernoulli draw selects no anchor, yielding $\mathcal{A} = \emptyset$, RUTA sets $\xi_{i^\star} = 1$, where $i^\star = \arg\max_i \rho_i$, thereby ensuring a nonempty anchor set.

[1] For any-resolution or tiled inputs, N_c may differ from N, depending on the number of encoded views.

Because gradients cannot be usefully propagated through the discrete Bernoulli samples, RUTA uses a straight-through estimator (Bengio et al., 2013; van den Oord et al., 2017):

$$\tilde{\xi}_i = \rho_i + \mathrm{sg}(\xi_i - \rho_i), \qquad i \in \{1, \dots, N_c\}, \tag{10}$$

where $\mathrm{sg}(\cdot)$ denotes the stop-gradient operator. Thus, $\tilde{\xi}_i$ equals the discrete sample ξ_i in the forward pass, while its gradient with respect to ρ_i is set to one in the backward pass. During the inference stage, the learned probabilities can be converted into token selections either stochastically, through Bernoulli sampling, or deterministically, through fixed thresholding or adaptive top-K. We use adaptive top-K by default and set

$$K_\varphi(Z, q) = \mathrm{clip}\Big(\mathrm{round}\Big(\widehat{R}_\varphi(Z, q)\Big), K_{\min}, N_c\Big), \tag{11}$$

where $K_{\min} \in \{1, \dots, N_c\}$ denotes the minimum allowable number of retained tokens. We construct the anchor index set $\mathcal{A}$ from the indices of the $K_\varphi(Z, q)$ candidates with the highest retention probabilities, and assign all remaining indices to the non-anchor index set $\bar{\mathcal{A}}$. This policy converts the learned expected rate into a sample-specific, deterministic token count.

Discarding the non-anchors would remove potentially useful context. RUTA instead distributes information from each non-anchor across the retained anchors according to semantic affinity and spatial proximity. For each anchor z_i indexed by $i \in \mathcal{A}$ and each non-anchor z_j indexed by $j \in \bar{\mathcal{A}}$, we first compute an unnormalized compatibility score:

$$e_{i,j} = \frac{\langle W_{\text{query}} z_i, W_{\text{key}} z_j \rangle}{\sigma_{\text{sa}}} - \lambda_{\text{sp}} \frac{\|\text{pos}_i - \text{pos}_j\|_2^2}{2\sigma_{\text{sp}}^2}. \tag{12}$$

The first term in $e_{i,j}$ measures learned semantic content affinity by mapping tokens through W_{query} and W_{key} into a shared affinity space. The scale $\sigma_{\text{sa}} > 0$ controls the sharpness of this affinity. The second term favors nearby tokens, where $\text{pos}_i \in [0, 1]^2$ is the normalized spatial center of token z_i and $\sigma_{\text{sp}} > 0$ is the spatial bandwidth. $\lambda_{\text{sp}} \geq 0$ controls the strength of the spatial bias. For each $j \in \bar{\mathcal{A}}$, we then normalize the scores $\{e_{i,j}\}_{i \in \mathcal{A}}$ over the anchor indices and use the resulting weights to aggregate non-anchor information into each anchor $\{z_i\}_{i \in \mathcal{A}}$:

$$w_{i,j} = \frac{\exp(e_{i,j})}{\sum_{i' \in \mathcal{A}} \exp(e_{i',j})}, \quad \Delta z_i = \frac{\sum_{j \in \bar{\mathcal{A}}} w_{i,j} W_{\text{value}} z_j}{\max\left(1, \sum_{j \in \bar{\mathcal{A}}} w_{i,j}\right)}, \quad \tilde{z}_i = \tilde{\xi}_i \cdot \mathrm{LN}\left(z_i + \beta \Delta z_i\right). \tag{13}$$

Here, the weights satisfy $\sum_{i \in \mathcal{A}} w_{i,j} = 1$ for each non-anchor and W_{value} is a learned value projection with output dimension D, $\mathrm{LN}(\cdot)$ denotes layer normalization over the embedding dimension, and β scales the residual update. The denominator in Δz_i averages contributions when the total assignment weight exceeds one, while preserving the attenuation of weak assignments when the total weight is below one. The learned query, key, and value projections follow the standard attention design and enable task-adaptive aggregation.

Although our anchor selection does not explicitly enforce diversity, the subsequent aggregation mitigates potential coverage loss by routing information from every non-anchor. This allows complementary evidence beyond the selected anchors to remain represented in the compact sequence. The resulting $\widetilde{V} = [\tilde{z}_i]_{i \in \mathcal{A}}$ contains $K(x, q) = |\mathcal{A}|$ merged anchors and replaces Z at the LLM input.

3.4 Joint Rate-Utility Optimization

RUTA jointly optimizes candidate construction and compression using task supervision and a differentiable rate penalty. Let $\mathcal{D} = \{(x^{(i)}, q^{(i)}, y^{(i)})\}_{i=1}^{|\mathcal{D}|}$ denote a mini-batch of image-query-target triples. For each $(x, q, y) \in \mathcal{D}$, Eq. (3) produces Z and Bernoulli sampling produces the corresponding compact sequence $\widetilde{V}$. Using one independently sampled gate vector per image-query pair in each forward pass, we optimize the following stochastic mini-batch objective:

$$\ell(\mathcal{D}; \phi, \varphi) = \frac{1}{|\mathcal{D}|} \sum_{(x,q,y) \in \mathcal{D}} \left[\ell_{\text{task}}\left(\mathrm{LLM}(q, \widetilde{V}), y\right) + \lambda_{\text{rate}} \widehat{R}_\varphi(Z, q) \right]. \tag{14}$$

The task term teaches the model to retain and aggregate evidence needed for prediction, while the rate term penalizes the expected number of tokens passed to the LLM. The multiplier λ_{rate} controls the rate-utility operating point: larger values favor compact representations, whereas smaller values prioritize task utility. Unlike a fixed per-sample budget, this mini-batch-averaged objective permits different inputs to receive different token counts under the same global rate pressure.

Table 1: Matched low-rate results across five benchmarks. Tok denotes the number of visual tokens averaged over samples and LLM layers, whereas Acc denotes the benchmark-specific accuracy metric. Reference denotes performance without visual token reduction, and Mean Acc is the unweighted average across the five benchmarks. Boldface marks the highest accuracy among the reduced methods within each VLM architecture block.

Method	VQA v2		GQA		TextVQA		A-OKVQA		VizWiz		Mean
	Tok	Acc	Tok	Acc	Tok	Acc	Tok	Acc	Tok	Acc	Acc
LLaVA-NeXT-7B											
Reference	2,233	81.1	2,012	73.2	2,310	66.1	2,234	83.2	2,336	61.6	73.04
FastV	42.0	45.3	32.0	40.4	50.0	41.6	48.0	55.5	48.0	53.7	47.30
SparseVLM	42.1	68.3	32.0	58.0	50.9	**51.0**	48.3	75.5	48.3	57.1	61.98
VisionZip	42.1	48.7	31.0	41.7	49.8	45.9	48.9	61.5	49.7	53.9	50.34
DivPrune	41.9	67.5	32.0	53.2	50.0	49.6	48.0	75.7	47.7	58.4	60.88
PruneSID	41.6	55.7	33.3	48.0	49.6	30.6	48.8	67.6	49.7	52.7	50.92
FastVLM	49.0	73.0	36.0	**64.2**	49.0	46.7	49.0	54.6	49.0	58.7	59.44
TwigVLM	42.0	35.1	32.3	27.5	50.3	22.6	48.0	53.0	48.0	5.0	28.64
RUTA (Ours)	42.1	**73.3**	32.1	60.4	50.4	49.7	48.1	**75.9**	47.1	**62.8**	**64.42**
Qwen3-VL-8B											
Reference	1,341	83.8	1,343	72.3	1,337	82.3	1,341	89.0	1,307	68.7	79.22
FastV	30.0	43.2	70.0	51.5	70.0	52.2	70.0	65.8	40.0	56.5	53.84
SparseVLM	30.2	42.1	70.7	36.9	70.7	44.9	70.7	48.7	41.3	53.7	45.26
VisionZip	30.0	55.1	70.0	60.4	70.0	54.9	70.0	76.2	40.0	62.7	61.86
DivPrune	28.0	67.1	70.0	55.9	70.0	62.1	70.0	**82.5**	40.0	65.3	66.58
PruneSID	27.4	65.1	71.1	54.6	69.6	38.2	70.8	80.5	40.4	63.9	60.46
FastVLM	36.0	69.6	81.0	60.5	81.0	48.2	81.0	54.0	36.0	58.6	58.18
TwigVLM	30.0	71.2	70.0	62.8	70.0	62.6	70.0	68.9	40.0	59.4	64.98
RUTA (Ours)	28.8	**76.6**	69.4	**71.9**	70.4	**73.1**	70.1	81.8	39.7	**70.5**	**74.78**

4 EXPERIMENTS

In this section, we evaluate whether RUTA improves task utility at a given visual token rate and whether its advantage persists across VLM backbones and operating points. Rate is measured as the number of visual tokens processed by the LLM, averaged over samples and layers. For a pre-LLM reducer, it reduces to the average visual input length; for an in-LLM reducer, it follows the layer-wise definition in Sec. 3.1. Utility is measured by the standard accuracy metric for each benchmark.

4.1 EXPERIMENTAL SETUPS

Datasets. We use five benchmarks spanning complementary evidence demands: VQAv2 (Goyal et al., 2017) for general visual question answering (VQA), GQA (Hudson & Manning, 2019) for compositional reasoning, TextVQA (Singh et al., 2019) for scene-text understanding, A-OKVQA (Schwenk et al., 2022) for knowledge-intensive reasoning, and VizWiz (Gurari et al., 2018) for questions collected from blind users in real-world settings. We report accuracy following the standard evaluation protocol for each dataset.

Implementation details. We evaluate RUTA with LLaVA-NeXT-7B (Liu et al., 2024a) and Qwen3-VL-8B (Bai et al., 2025). The vision encoder, text embedding, and LLM remain frozen. Trainable components are limited to the grounding heads of the candidate localizer, the retention MLP, the anchor-merging projections in h_φ, and the VLM's lightweight vision-to-language projector. We set the confidence threshold δ for phrase-conditioned boxes in Eq. (5) to 0.5, the retention temperature τ in Eq. (7) to 0.7, and the minimum token count $K_{\min}$ in Eq. (11) to 1. While anchor merging, the semantic affinity uses scale $\sigma_{\text{sa}} = 0.07$, and the spatial affinity uses $\sigma_{\text{sp}}^2 = 0.04$ with spatial weight $\lambda_{\text{sp}} = 1.0$ and residual scalar $\beta = 4.0$. The default rate coefficient is $\lambda_{\text{rate}} = 0.02$ for LLaVA-NeXT and 2×10^{-5} for Qwen3-VL.

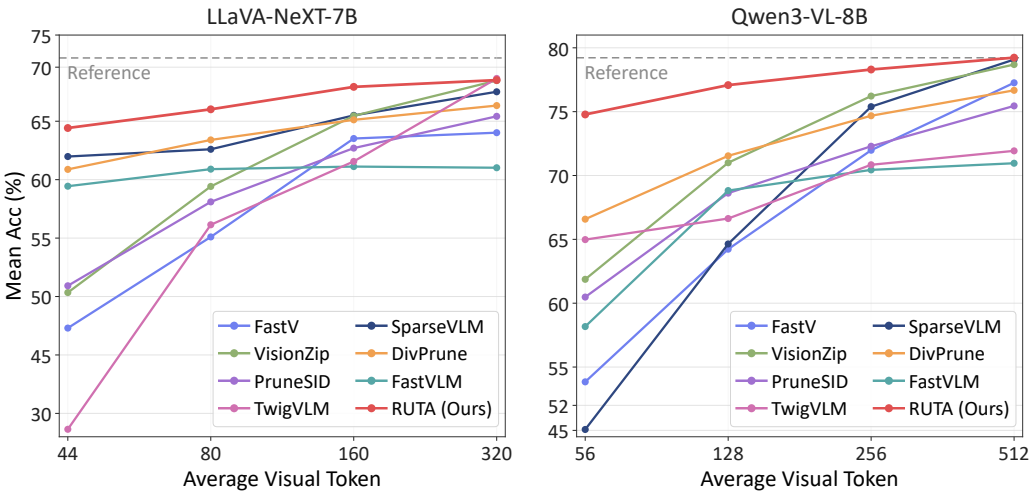


Figure 3: Rate–utility curves for LLaVA-NeXT-7B and Qwen3-VL-8B. Each point reports the mean accuracy across the five benchmarks against the corresponding average number of visual tokens processed per sample and LLM layer. Dashed lines indicate the performance of the unreduced reference models. RUTA achieves its largest gains under stringent token budgets, although its advantage narrows on LLaVA-NeXT as the budget increases.

4.2 Main Results

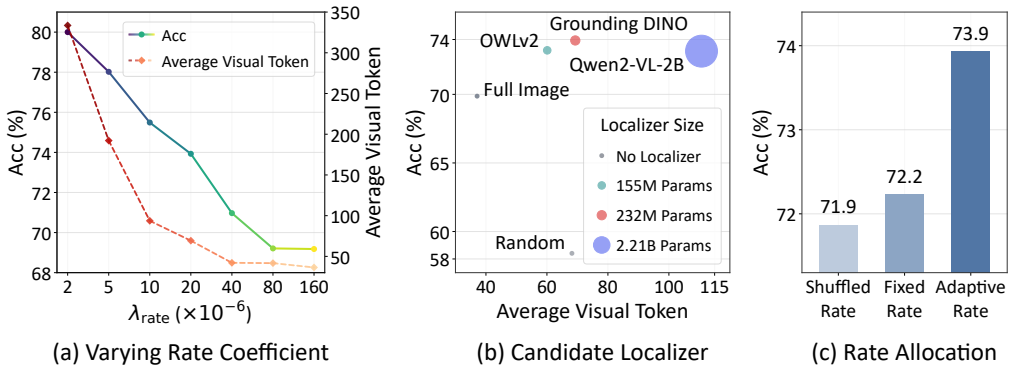
We compare RUTA with training-free reducers based on query relevance or attention (FastV, Chen et al., 2024b; SparseVLM, Zhang et al., 2025d), similarity-aware merging (VisionZip, Yang et al., 2025), and diversity-aware selection (DivPrune, Alvar et al., 2025; PruneSID, Fang et al., 2026). We also include the training-based methods (FastVLM, Vasu et al., 2025; TwigVLM, Shao et al., 2025).

Table 1 supports two main observations. First, RUTA ranks first in mean accuracy for both VLMs and remains close to the best reduced method on every benchmark, indicating stable low-rate performance across different token layouts and evidence demands. Second, the competing methods reveal a mismatch between common reduction criteria and query-specific evidence needs. FastV and SparseVLM derive importance from LLM attention. SparseVLM's contrasting behavior across the two backbones suggests that such signals can be sensitive to token layout and attention patterns. VisionZip, DivPrune, and PruneSID improve preservation through semantic similarity, diversity, and image-complexity cues, respectively, but these largely image-centric criteria do not guarantee query relevance: visually distinctive or complex content may contribute little to the answer. FastVLM and TwigVLM instead target broader system efficiency, so their designs are not optimized specifically for task utility at a matched token rate. After all, reducing spatial resolution can discard fine-grained evidence, while the effectiveness of early-layer pruning may vary across VLM architectures. RUTA addresses these limitations by jointly learning where to focus, which tokens to retain, how many tokens to allocate to each image-query pair, and how to aggregate the remainder under a rate-utility objective. The stronger and more consistent results are therefore aligned with matching both the visual content and the quantity of retained evidence to the current query.

Fig. 3 broadens the comparison beyond one low-rate operating point. RUTA leads by a clear margin under aggressive compression on both backbones. On LLaVA-NeXT-7B, however, the curves converge as the budget grows, and RUTA ranks second at 320 tokens behind TwigVLM. This pattern suggests that localized candidate construction is most beneficial when token capacity is scarce. With larger budgets, full-image reducers can preserve broader spatial coverage. On Qwen3-VL-8B, RUTA remains above the competing reducers from 56 to 512 tokens and approximately matches the unreduced backbone at the largest setting.

4.3 Ablation Studies

We conduct three controlled studies on TextVQA with Qwen3-VL-8B. We verify that the rate coefficient controls the operating point and then isolate the contributions of candidate construction and sample-adaptive allocation. Unless stated otherwise, visual token counts are averaged over samples.



(a) Varying Rate Coefficient (b) Candidate Localizer (c) Rate Allocation

Figure 4: Rate control and component ablations on TextVQA with Qwen3-VL-8B. **(a)** Increasing the rate coefficient λ_{rate} progressively reduces token usage, illustrating the resulting rate-utility trade-off. **(b)** Comparison of query-conditioned candidate localizers, along with random-region and full-image degenerates. Marker area is proportional to the localizer parameter count. **(c)** Comparison of fixed, shuffled, and adaptive rate allocation at approximately 70 visual tokens on average.

Effect of the rate coefficient. Fig. 4(a) shows that λ_{rate} directly controls the rate-utility operating point. Increasing it from 2×10^{-6} to 2×10^{-5} reduces the average rate from 333.6 to 69.3 tokens, while accuracy decreases from 80.0% to 73.9%. Beyond this point, stronger rate pressure produces progressively smaller rate reductions while accuracy falls toward 69.2%. We therefore use $\lambda_{\text{rate}} = 2 \times 10^{-5}$ for Qwen3-VL because it lies near the knee of the curve, providing a substantial rate reduction before utility declines more sharply.

Choice of candidate localizer. Fig. 4(b) varies the candidate localizer while keeping the compressor and training protocol fixed. OWLv2 (Minderer et al., 2023), Qwen2-VL-2B (Wang et al., 2024), and Grounding DINO (Liu et al., 2024b) achieve comparable accuracy and consistently outperform the random-region and full-image controls. This consistency across architectures and parameter scales suggests that the gains arise primarily from query-conditioned localization rather than a specific localizer. Because the variants produce different average token rates, the results demonstrate robustness to localizer choice rather than a strictly rate-matched ranking. We use Grounding DINO by default because it achieves the highest accuracy with a moderate parameter count.

Effect of adaptive rate allocation. Finally, Fig. 4(c) separates rate variation from demand-aware assignment. The fixed-rate control gives every sample 70 tokens, whereas the shuffled-rate control preserves RUTA's rate distribution but randomly reassigns those rates across samples. At the same average rate, adaptive allocation reaches the highest accuracy. The gain over fixed rate shows that varying the budget across samples is useful, while the gain over shuffled rate shows that the budget must also be assigned to the appropriate input. Rate variation alone is therefore insufficient; its value comes from matching capacity to evidence demand.

5 Conclusion and Discussion

We have presented RUTA, a rate-utility optimization approach that learns both which visual evidence to retain and how many tokens each image-query pair should pass to the LLM. It combines query-conditioned candidate construction, probabilistic retention, and anchor-based aggregation to balance task utility against expected token usage. Across five VQA benchmarks and two VLM architectures, RUTA achieves the highest mean accuracy among the reduced methods at matched stringent token budgets, supporting the value of matching retained evidence to query-specific demands.

Several limitations remain. First, token count is a model-relative proxy that omits localization and encoding overhead and does not fully reflect end-to-end latency or memory usage. Next, a single enclosing crop preserves intervening context and spatial relationships, but offers less resolution benefit when relevant regions are widely separated. Small or poorly localized evidence may also be missed. Finally, deployment on new backbones or domains requires retraining, and evaluation beyond single-image VQA remains open. These considerations motivate transferable reducers with minimal adaptation, inference-time token, latency, or memory control without retraining for each budget, and further adaptive scheduling of limited computation across regions, images, and frames.

AI USE STATEMENT

In this work, we used generative AI tools to assist with translation, and provide feedback on research methodology and experiments. We have not used generative AI tools to generate synthetic datasets, help develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in the writing of proofs, propose or refine hypotheses, implement methods, clean and reformat datasets, support qualitative and thematic data analysis, or interpret results. Additionally, we used generative AI tools to summarize existing literature, search for information, edit a research paper to improve readability, and identify relevant literature. We have reviewed all AI-assisted work. We have checked LLM-generated research content for potential plagiarism through a manual literature survey. The LLM-generated scripts and code used to support the methodological and experimental feedback described above were verified and tested for correctness by the authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

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. In *Advances in Neural Information Processing Systems*, pp. 23716–23736, 2022.

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

Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013.

Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your ViT but faster. In *International Conference on Learning Representations*, 2023.

Mu Cai, Jianwei Yang, Jianfeng Gao, and Yong Jae Lee. Matryoshka multimodal models. In *International Conference on Learning Representations*, 2025.

Junbum Cha, Wooyoung Kang, Jonghwan Mun, and Byungseok Roh. Honeybee: Locality-enhanced projector for multimodal LLM. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13817–13827, 2024.

Jieneng Chen, Luoxin Ye, Ju He, Zhao-Yang Wang, Daniel Khashabi, and Alan Yuille. Efficient large multi-modal models via visual context compression. In *Advances in Neural Information Processing Systems*, pp. 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, 2024b.

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. In *Advances in Neural Information Processing Systems*, pp. 49250–49267, 2023.

Zhen Dong, Zhewei Yao, Amir Gholami, Michael W. Mahoney, and Kurt Keutzer. HAWQ: Hessian aware quantization of neural networks with mixed-precision. In *IEEE/CVF International Conference on Computer Vision*, pp. 293–302, 2019.

Zhengyao Fang, Pengyuan Lyu, Chengquan Zhang, Guangming Lu, Jun Yu, and Wenjie Pei. Prune redundancy, preserve essence: Vision token compression in VLMs via synergistic importance-diversity. In *International Conference on Learning Representations*, 2026.

Mohsen Fayyaz, Soroush Abbasi Koohpayegani, Farnoush Rezaei Jafari, Sunando Sengupta, Hamid Reza Vaezi Joze, Eric Sommerlade, Hamed Pirsiavash, and Jürgen Gall. Adaptive token sampling for efficient vision Transformers. In *European Conference on Computer Vision*, pp. 396–414, 2022.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. OPTQ: Accurate post-training quantization for generative pre-trained Transformers. In *International Conference on Learning Representations*, 2023.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the V in VQA matter: Elevating the role of image understanding in visual question answering. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6904–6913, 2017.

Jiawei Guo, Feifei Zhai, Pu Jian, Qianrun Wei, and Yu Zhou. CROP: Contextual region-oriented visual token pruning. In *Conference on Empirical Methods in Natural Language Processing*, pp. 9756–9772, 2025.

Zonghao Guo, Ruyi Xu, Yuan Yao, Junbo Cui, Zanlin Ni, Chunjiang Ge, Tat-Seng Chua, Zhiyuan Liu, and Gao Huang. LLaVA-UHD: An LMM perceiving any aspect ratio and high-resolution images. In *European Conference on Computer Vision*, pp. 390–406, 2024.

Danna Gurari, Qing Li, Abigale J. Stangl, Anhong Guo, Chi Lin, Kristen Grauman, Jiebo Luo, and Jeffrey P. Bigham. VizWiz grand challenge: Answering visual questions from blind people. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3608–3617, 2018.

Yefei He, Feng Chen, Jing Liu, Wenqi Shao, Hong Zhou, Kaipeng Zhang, and Bohan Zhuang. ZipVL: Accelerating vision-language models through dynamic token sparsity. In *IEEE/CVF International Conference on Computer Vision*, pp. 20477–20486, 2025.

Drew A. Hudson and Christopher D. Manning. GQA: A new dataset for real-world visual reasoning and compositional question answering. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6700–6709, 2019.

Andrew Jaegle, Felix Gimeno, Andy Brock, Oriol Vinyals, Andrew Zisserman, and Joao Carreira. Perceiver: General perception with iterative attention. In *International Conference on Machine Learning*, pp. 4651–4664, 2021.

Andrew Jaegle, Sebastian Borgeaud, Jean-Baptiste Alayrac, Carl Doersch, Catalin Ionescu, David Ding, Skanda Koppula, Daniel Zoran, Andrew Brock, Evan Shelhamer, et al. Perceiver IO: A general architecture for structured inputs & outputs. In *International Conference on Learning Representations*, 2022.

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, 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, 2025.

Zhang Li, Biao Yang, Qiang Liu, Zhiyin Ma, Shuo Zhang, Jingxu Yang, Yabo Sun, Yuliang Liu, and Xiang Bai. Monkey: Image resolution and text label are important things for large multi-modal models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 26763–26773, 2024.

Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. LLaVA-NeXT: Improved reasoning, OCR, and world knowledge. LLaVA project blog, January 2024a. URL https://llava-vl.github.io/blog/2024-01-30-llava-next/.

Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, et al. Grounding DINO: Marrying DINO with grounded pre-training for open-set object detection. In *European Conference on Computer Vision*, pp. 38–55, 2024b.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

David Marr. *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*. W. H. Freeman, San Francisco, CA, 1982.

Matthias Minderer, Alexey Gritsenko, and Neil Houlsby. Scaling open-vocabulary object detection. In *Advances in Neural Information Processing Systems*, pp. 72983–73007, 2023.

Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. DynamicViT: Efficient vision Transformers with dynamic token sparsification. In *Advances in Neural Information Processing Systems*, pp. 13937–13949, 2021.

Michael Ryoo, AJ Piergiovanni, Anurag Arnab, Mostafa Dehghani, and Anelia Angelova. TokenLearner: Adaptive space-time tokenization for videos. In *Advances in Neural Information Processing Systems*, pp. 12786–12797, 2021.

Dustin Schwenk, Apoorv Khandelwal, Christopher Clark, Kenneth Marino, and Roozbeh Mottaghi. A-OKVQA: A benchmark for visual question answering using world knowledge. In *European Conference on Computer Vision*, pp. 146–162, 2022.

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

Claude E. Shannon. Coding theorems for a discrete source with a fidelity criterion. *IRE International Convention Record*, 7(4):142–163, 1959.

Zhenwei Shao, Mingyang Wang, Zhou Yu, Wenwen Pan, Yan Yang, Tao Wei, Hongyuan Zhang, Ning Mao, Wei Chen, and Jun Yu. Growing a twig to accelerate large vision-language models. In *IEEE/CVF International Conference on Computer Vision*, pp. 20064–20074, 2025.

Sheng Shen, Zhen Dong, Jiayu Ye, Linjian Ma, Zhewei Yao, Amir Gholami, Michael W. Mahoney, and Kurt Keutzer. Q-BERT: Hessian based ultra low precision quantization of BERT. In *AAAI Conference on Artificial Intelligence*, pp. 8815–8821, 2020.

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/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8317–8326, 2019.

Shengbang Tong, Ellis Brown, Penghao Wu, Sanghyun Woo, Manoj Middepogu, Sai C. Akula, Jihan Yang, Shusheng Yang, Adithya Iyer, Xichen Pan, et al. Cambrian-1: A fully open, vision-centric exploration of multimodal LLMs. In *Advances in Neural Information Processing Systems*, pp. 87310–87356, 2024.

Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In *Advances in Neural Information Processing Systems*, pp. 6306–6315, 2017.

Pavan Kumar Anasosalu Vasu, Fartash Faghri, Chun-Liang Li, Cem Koc, Nate True, Albert Antony, Gokula Santhanam, James Gabriel, Peter Grasch, Oncel Tuzel, et al. FastVLM: Efficient vision encoding for vision language models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19769–19780, 2025.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-VL: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024.

Haoning Wu, Dongxu Li, Bei Chen, and Junnan Li. LongVideoBench: A benchmark for long-context interleaved video-language understanding. In *Advances in Neural Information Processing Systems*, pp. 28828–28857, 2024.

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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14593–14603, 2025.

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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19792–19802, 2025.

Zhewei Yao, Reza Yazdani Aminabadi, Minjia Zhang, Xiaoxia Wu, Conglong Li, and Yuxiong He. ZeroQuant: Efficient and affordable post-training quantization for large-scale Transformers. In *Advances in Neural Information Processing Systems*, pp. 27168–27183, 2022.

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

Hongxu Yin, Arash Vahdat, Jose M. Alvarez, Arun Mallya, Jan Kautz, and Pavlo Molchanov. A-ViT: Adaptive tokens for efficient vision Transformer. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10809–10818, 2022.

Quan-Sheng Zeng, Yunheng Li, Qilong Wang, Peng-Tao Jiang, Zuxuan Wu, Ming-Ming Cheng, and Qibin Hou. A glimpse to compress: Dynamic visual token pruning for large vision-language models. *IEEE Transactions on Circuits and Systems for Video Technology*, 2026.

Ce Zhang, Kaixin Ma, Tianqing Fang, Wenhao Yu, Hongming Zhang, Zhisong Zhang, Haitao Mi, and Dong Yu. VScan: Rethinking visual token reduction for efficient large vision-language models. *Transactions on Machine Learning Research*, 2026.

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 *IEEE/CVF International Conference on Computer Vision*, pp. 20857–20867, 2025a.

Shaolei Zhang, Qingkai Fang, Zhe Yang, and Yang Feng. LLaVA-Mini: Efficient image and video large multimodal models with one vision token. In *International Conference on Learning Representations*, 2025b.

Yi-Fan Zhang, Huanyu Zhang, Haochen Tian, Chaoyou Fu, Shuangqing Zhang, Junfei Wu, Feng Li, Kun Wang, Qingsong Wen, Zhang Zhang, et al. MME-RealWorld: Could your multimodal LLM challenge high-resolution real-world scenarios that are difficult for humans? In *International Conference on Learning Representations*, 2025c.

Yuan Zhang, Chun-Kai Fan, Junpeng Ma, Wenzhao Zheng, Tao Huang, Kuan Cheng, Denis A. Gudovskiy, Tomoyuki Okuno, Yohei Nakata, Kurt Keutzer, et al. SparseVLM: Visual token sparsification for efficient vision-language model inference. In *International Conference on Machine Learning*, pp. 74840–74857, 2025d.

A Appendix

Content

This appendix provides:

- Additional implementation details on input preprocessing and training in Sec. A.1;
- Inference-efficiency measurements and analysis across VLM backbones in Sec. A.2;
- Qualitative analysis of query-conditioned localization and adaptive token allocation in Sec. A.3;
- Additional controlled studies in Sec. A.4.

A.1 Additional Implementation Details

For LLaVA-NeXT, we apply the official any-resolution preprocessing to all methods and subsequently encode each localized crop as a single 336×336 view, yielding 576 candidate tokens. For Qwen3-VL, we preserve the input aspect ratio while constraining the image area to approximately $1,152^2$ pixels. Grounding DINO (Liu et al., 2024b) is the default localizer, with only its grounding heads updated. We optimize RUTA using AdamW (Loshchilov & Hutter, 2019) with a learning rate of 1×10^{-5}, a weight decay of 1×10^{-4}, and a batch size of 6 per GPU across 8 GPUs. Training runs for 2 epochs on VQAv2 and GQA and 4 epochs on the remaining datasets.

A.2 Inference-efficiency measurements

Table 2: Inference efficiency on TextVQA with two VLM backbones. RUTA significantly reduces LLM-prefill latency on both backbones, while end-to-end speedups and peak-memory changes depend on their computation and memory profiles.

Metric	Reference	RUTA (Ours)
Accuracy	64.6	50.6
#Visual Tokens	2309.9	49.7
Peak GPU Memory	17.4 GB	14.9 GB
Prefill Latency	654.6 ms	39.7 ms
End-to-End Latency	819.2 ms	312.5 ms
Throughput	1.2 sample/s	3.2 sample/s

(a) LLaVA-NeXT-7B

Metric	Reference	RUTA (Ours)
Accuracy	83.0	73.1
#Visual Tokens	1337.5	69.3
Peak GPU Memory	21.0 GB	21.7 GB
Prefill Latency	363.2 ms	42.0 ms
End-to-End Latency	1146.2 ms	1019.6 ms
Throughput	0.9 sample/s	1.0 sample/s

(b) Qwen3-VL-8B

We report three-run averages on TextVQA using eight RTX A6000 GPUs and Intel Xeon Silver 4309Y CPUs. Accuracy and visual token counts follow Secs. 4.1 and 3.1. Prefill latency covers the first LLM forward after input-sequence construction. End-to-end latency includes CPU-side input preparation, CPU-to-GPU transfer, and all inference stages through text decoding, ending with GPU synchronization. Peak memory is the maximum allocated memory across GPUs, and throughput is normalized per active GPU. Reference bypasses localization and reduction.

Table 2(a) shows that RUTA translates visual-token reduction into substantial end-to-end gains on LLaVA-NeXT-7B. Prefill dominates the reference latency, so shortening the visual sequence directly reduces a major inference cost. Candidate construction also replaces the original multi-view representation with a single 336×336 crop (Sec. A.1), reducing the vision-encoding workload. Meanwhile, the longer reference sequence and 32-head KV cache make LLM-side memory more sensitive to token count. These properties help explain why RUTA improves latency, throughput, and peak memory together on this backbone.

On Qwen3-VL-8B, Table 2(b) shows a similarly strong reduction in prefill latency, but this stage contributes less to the reference end-to-end latency. Each crop is still encoded at approximately 1.33 Mpixels, so localization concentrates visual evidence without substantially reducing the vision encoder's workload. The end-to-end speedup is therefore limited by the largely unchanged encoding cost and the additional overhead of localization, retention prediction, and anchor merging. Unlike on LLaVA-NeXT-7B, peak memory usage on Qwen3-VL-8B slightly exceeds the reference. Qwen's shorter reference sequence and 8-head grouped-query KV cache offer less scope for cache reduction,

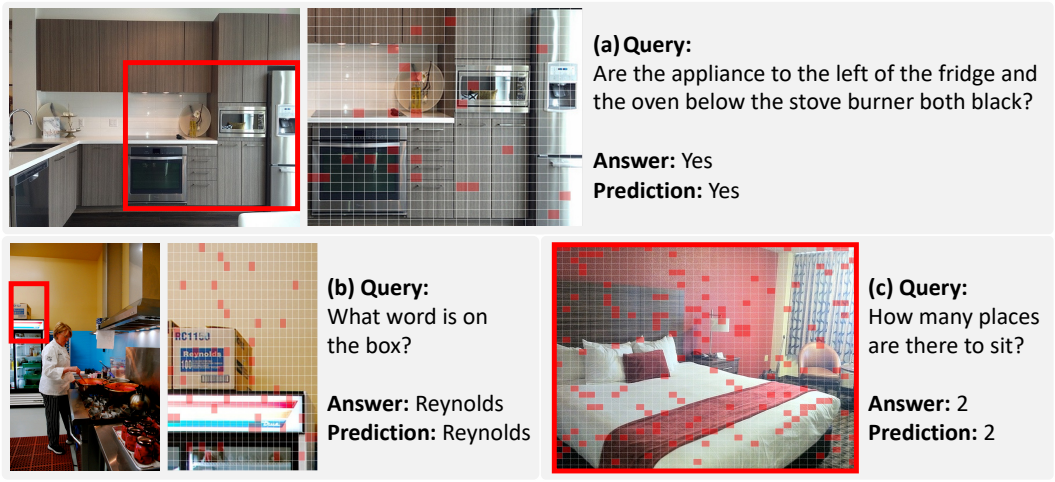


Figure 5: Qualitative examples of query-conditioned localization and adaptive token allocation. Red boxes indicate the regions selected by the localizer, and red grid cells denote retained anchors. In **(a)**, RUTA converts $2,340$ visual tokens from the unreduced LLaVA-NeXT input into 576 localized candidates and retains 24 anchors. In **(b)**, it similarly converts $2,160$ LLaVA-NeXT input tokens into 576 candidates and retains 42 anchors. In **(c)**, the localizer of Qwen3-VL returns no region, so the full image is encoded into $1,344$ candidates, of which 139 are retained. These examples illustrate how RUTA adapts both the region examined and the number of retained tokens to the query.

while model weights and vision workspace remain largely unchanged. The memory footprint of RUTA's additional modules can thus offset the cache savings at the system-wide peak, even though the LLM processes substantially fewer visual tokens.

The cross-backbone difference thus reflects how much of the inference workload is sensitive to visual sequence length. RUTA reduces this cost on both backbones, while the resulting system-level benefit depends on their encoding and decoding profiles. Further gains can be pursued through standard inference optimizations, such as optimized vision kernels and reuse of temporary buffers, while preserving the existing candidate construction and rate-utility objective.

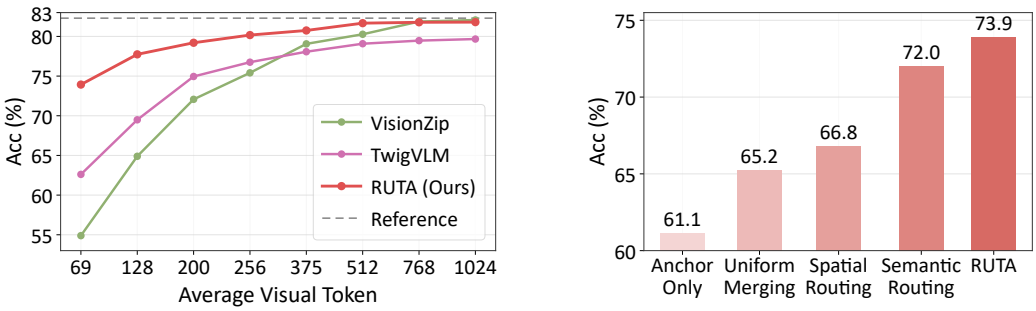
A.3 Qualitative Analysis

Fig. 5 illustrates how candidate construction and rate allocation respond to different evidence demands. In (a), the query requires several appliances and their spatial relations; the localized region then covers the oven, refrigerator, and stove burner while excluding nearly half of the image. In (b), localization magnifies the text-bearing box before the compressor selects a small anchor set. In (c), the query provides no explicit region to ground, so RUTA uses the full image and allocates more anchors to cover the scene, which extend beyond the immediate answer locations. This spatial spread supports the merge in Eq. (13), allowing each anchor to absorb information from nearby non-anchors while preserving contextual coverage.

A.4 Additional Ablation Studies

Scaling token budgets. Fig. 6(a) compares RUTA, VisionZip, and TwigVLM across expanded token budgets. RUTA remains above TwigVLM throughout the evaluated range. The narrowing margin indicates that query-conditioned candidate construction is most valuable when representational capacity is scarce. VisionZip becomes comparable at 768 tokens and slightly surpasses RUTA at $1{,}024$ tokens, where a full-image reducer can preserve broader context directly. Thus, RUTA's main advantage lies in low-to-medium-rate operation. This regime is particularly relevant to practical deployment because meaningful reductions in LLM-side computation and memory require operating well below the unreduced token count.

Effect of anchor merging. Fig. 6(b) fixes the selected anchors and average rate to isolate how non-anchors should be incorporated. Uniform merging, where $w_{i,j}$ in Eq. (13) is set to $1/|\mathcal{A}|$, improves



(a) Scaling Token Budgets (b) Anchor Merging

Figure 6: Token-budget scaling and merging ablations on TextVQA with Qwen3-VL-8B. (a) Accuracy as a function of the average visual token count for RUTA, VisionZip, and TwigVLM; the dashed line marks the unreduced reference model. Visual token counts are averaged over LLM layers and samples. RUTA performs particularly well in the low-budget regime and approaches the reference accuracy as the budget increases. (b) Comparison of anchor-only retention and four merging strategies at a matched average rate. The results characterize performance across token budgets and isolate the contribution of anchor-merging strategies.

over retaining anchors alone, showing that non-anchors contain complementary evidence. Semantic routing provides a substantially larger benefit than spatial routing, indicating that content affinity is the primary assignment signal. Their combination performs best, confirming that spatial proximity provides complementary structure when routing non-anchor information to semantically compatible anchors. These results motivate the joint semantic-spatial formulation in Eq. (12).