Efficient Visual Grounding in VQA via Question-Guided Sparse Attention

Prasanth Yadla
Independent Researcher
Seattle, WA, USA
pyadla2@alumni.ncsu.edu

Abstract

Visual Question Answering (VQA) models process all image patches uniformly despite questions typically requiring only a small subset of visual information. This inefficiency leads to unnecessary computation and can result in attention dilution across irrelevant image regions. We propose **Question-Guided Sparse Attention (QGSA)**, a plug-and-play mechanism that dynamically selects relevant image patches conditioned on question semantics. Our approach introduces three components: (1) a differentiable patch selector based on Gumbel-Softmax reparameterisation that enables end-to-end training with hard patch selection at inference; (2) a self-supervised grounding loss that encourages spatial selectivity without bounding-box annotations, combining contrastive patch selection with patch–word alignment via a frozen CLIP encoder; and (3) an adaptive sparsity mechanism that adjusts the number of selected patches according to estimated question complexity. Experiments on SmolVLM-256M-Instruct and SmolVLM-500M-Instruct across three VQA benchmarks (VQA-RAD, A-OKVQA, RefCOCO) demonstrate that QGSA reduces cross-attention FLOPs by 91–99% across input resolutions, achieving up to $76\times$ theoretical speedup at 576px resolution, while maintaining *exact* accuracy parity with the dense baseline ($\Delta = 0.0$ pp on all datasets). Wall-clock parity with the dense baseline is reached at 336px; realised end-to-end speedup requires larger models where cross-attention dominates total compute. QGSA consistently selects an average of $k \approx 17$ patches out of 576 (256M model), up to $k \approx 18$ (500M model), yielding up to a $34\times$ reduction in the visual token sequence. These small-scale results validate the feasibility of question-conditioned sparse attention and provide a foundation for scaling to larger VLMs.

1 Introduction

Visual Question Answering (VQA) lies at the intersection of natural language understanding and visual perception: a model must ground a linguistic query in image content and produce a faithful, concise answer. The remarkable performance of modern vision–language models (VLMs) such as BLIP-2 (Li et al., 2023a), InstructBLIP (Dai et al., 2023), and LLaVA-1.5 (Liu et al., 2023) is achieved by encoding high-resolution images into dense grids of patch tokens and processing them through transformer cross-attention layers. While this design yields strong benchmark numbers, it embeds a fundamental inefficiency: *every question is answered by attending to every patch, regardless of relevance*.

Two concrete problems follow from this. The first is computational inefficiency. A standard 384×384 image produces $N = 576$ patch tokens after ViT encoding (Dosovitskiy et al., 2021). For the question "What colour is the car?" perhaps 8–12 patches are relevant, yet all 576 enter every cross-attention layer. This overhead scales quadratically with resolution and becomes prohibitive for high-resolution inputs, multi-frame video, or tasks that reason over several images simultaneously. The second problem is attention dilution. With hundreds of tokens competing for attention weight, the cross-attention distribution spreads across irrelevant background regions. Empirical studies of VLM hallucination (Rohrbach et al., 2019; Li et al., 2023b) consistently show that erroneous answers correlate with models attending to distractors rather than the objects named in the question.

Prior work on attention in VQA (Yu et al., 2019; Kim et al., 2018) employs soft, continuous attention weights, which reshape information routing but never eliminate the quadratic cost of attending to all patches. Token pruning methods for vision transformers (Rao et al., 2021; Xu et al., 2021) and

token merging (Bolya et al., 2023) reduce sequence length, yet they are driven by image-internal statistics and carry no awareness of the question being asked. Architectures such as Q-Former (Li et al., 2023a) and Perceiver (Jaegle et al., 2021) compress visual tokens via learned query banks, but these queries are fixed across all questions and do not perform hard, discrete selection.

Our proposal is **Question-Guided Sparse Attention (QGSA)**, a lightweight module inserted between the vision encoder and the language model. QGSA scores every patch against the current question, selects the top-k most relevant patches using a Gumbel-Softmax reparameterisation (Jang et al., 2017), and forwards only those patches to the downstream transformer. The budget k is itself predicted from the question, so semantically simple queries receive fewer patches than compositional spatial-reasoning queries. Crucially, QGSA requires no bounding-box supervision: our self-supervised grounding loss leverages contrastive comparison between selected and random patches together with noun-level alignment via a frozen CLIP text encoder, coaxing the selector toward semantically meaningful regions without any localisation annotation.

Our contributions are:

- A **differentiable patch selector** using Gumbel-Softmax with the straight-through estimator, enabling gradient-based training of a hard selection function.
- A **self-supervised grounding loss** comprising a contrastive selection term ($\mathcal{L}_{cs}$) and a patch–word alignment term ($\mathcal{L}_{align}$) that encourage spatially selective behaviour without any bounding-box annotation.
- An **adaptive sparsity mechanism** (f_{comp}) predicting a per-question patch budget—intended to allocate more tokens to hard spatial queries and fewer to simple factual ones, though in practice this differentiation remains a challenge on small datasets (Section 4.7).
- Experiments on SmolVLM-256M-Instruct and SmolVLM-500M-Instruct across VQA-RAD, A-OKVQA, and RefCOCO, demonstrating 91–99% cross-attention FLOPs reduction at zero accuracy cost, a 31× relative grounding IoU improvement, and wall-clock parity with the dense baseline at ≥336px.

2 Related Work

2.1 Visual Question Answering

Early VQA systems fused question and image representations via element-wise operations or simple attention (Agrawal et al., 2016). Co-attention mechanisms, exemplified by MCAN (Yu et al., 2019) and the bilinear attention network BAN (Kim et al., 2018), enabled richer cross-modal interaction and pushed accuracy significantly, yet the fundamental cost structure remained unchanged: computation still scales with the full patch count. Large-scale vision–language pretraining (Li et al., 2023a; Dai et al., 2023; Liu et al., 2023; Alayrac et al., 2022) has since dramatically raised accuracy on standard benchmarks, but has also inflated the visual token count to the point where inference latency is increasingly problematic in deployment settings. The gap between what a model *attends to* and what a question actually *requires* has only widened.

2.2 Sparse and Efficient Attention

The sparse transformer (Child et al., 2019) showed that restricting self-attention to local or strided neighbourhoods could recover most of the performance of full attention at a fraction of the $O(N^2)$ cost. Vision transformers have taken analogous approaches: DynamicViT (Rao et al., 2021) learns to progressively prune tokens during the forward pass; EvoViT (Xu et al., 2021) maintains a slow-updating budget of important tokens; Token Merging (Bolya et al., 2023) avoids pruning entirely by fusing similar tokens via bipartite matching, which is often faster in practice than learned selection. The shared limitation of all these methods is that token importance is determined by the image alone. In a VQA setting this is a fundamental problem: the patches relevant to "What is the man holding?" and "What colour is the wall?" are completely disjoint on the same image, yet image-only pruning makes no such distinction. To our knowledge, QGSA is the first method to perform hard, question-conditioned token selection within a pretrained VLM rather than as a preprocessing step external to it.

2.3 Visual Grounding in VQA

Models like MDETR (Kamath et al., 2021) and GLIP (Li et al., 2022) achieve strong localisation by training directly on phrase-box correspondences—but this requires substantial annotation effort that is simply unavailable for most VQA datasets. GQA (Hudson and Manning, 2019) and Ref-

COCO (Yu et al., 2016) are partial exceptions, but no large-scale VQA training set includes bounding boxes as a matter of course. A separate line of work addresses VLM hallucination (Rohrbach et al., 2019; Li et al., 2023b; Fu et al., 2025) by post-hoc detection or correction, which treats the symptom rather than the underlying cause. Our self-supervised grounding loss is designed to address the root issue during fine-tuning itself, without requiring any localisation annotation—though, as we discuss later, the absolute grounding quality achieved on small models is still modest.

Comparison with patch and token selection methods. The closest prior work in mechanism are DynamicViT (Rao et al., 2021), EvoViT (Xu et al., 2021), and Token Merging (Bolya et al., 2023). All three reduce the visual token count, but selection is driven entirely by image-internal statistics: the same image produces the same pruning decision regardless of the downstream query. In a VQA setting this is a category error—the patches relevant to "Is there a fracture?" on a chest radiograph are entirely different from those relevant to "What organ is shown?" on the same image, and no image-only criterion can distinguish them.

QGSA differs on three axes. First, selection is *conditioned on the question*: the patch scorer (Eq. 1) explicitly fuses $\mathbf{q}$ with each patch feature v_i, so the same image yields different subsets for different queries. Second, selection is *hard and discrete* at inference: unlike soft re-weighting in Q-Former (Li et al., 2023a) or co-attention (Yu et al., 2019), QGSA produces a binary mask that fully excludes non-selected patches from the key–value matrices, giving a true $O(k \cdot d)$ cross-attention cost. Third, the budget k is *predicted per question* by f_{comp}, rather than fixed globally—though in practice this collapses to a near-uniform budget on small datasets (Section 4.7).

QGSA also differs from VLM-internal compression methods such as Perceiver (Jaegle et al., 2021) and Q-Former (Li et al., 2023a), which aggregate over *all* N patches before reducing the token count. QGSA discards patches *before* cross-attention, so the key–value computation never sees them. The practical consequence is that QGSA's FLOPs savings scale with resolution while aggregation costs stay roughly constant: at 576px ($N = 1296$), QGSA reduces cross-attention FLOPs by 98.7%, a saving that grows quadratically with input resolution.

3 Method

3.1 Problem Formulation

Given an image $\mathcal{I}$ and a natural-language question $\mathcal{Q}$, a standard VLM proceeds in four steps: (1) encode $\mathcal{I}$ with a frozen vision encoder to obtain patch features $\mathbf{V} = \{v_1, \ldots, v_N\} \in \mathbb{R}^{N \times d_v}$; (2) encode $\mathcal{Q}$ to obtain a question representation $\mathbf{q} \in \mathbb{R}^{d_q}$; (3) compute cross-attention between $\mathbf{q}$ and all N patch tokens; and (4) generate answer $\hat{a}$. The total cost of the cross-attention layers scales as $O(N \cdot d)$ per head, dominating inference time for large N.

QGSA inserts a sparse selection step between stages (2) and (3): it selects a small subset $\mathbf{V}_{sparse} = \{v_{i_1}, \ldots, v_{i_k}\}$ where $k \ll N$, driven by question relevance, and forwards only this subset to all downstream cross-attention computations. The resulting cost is $O(k \cdot d)$, yielding a theoretical speedup of N/k.

One practical complication is that SmolVLM's native connector module is designed for full grid inputs and does not handle arbitrary patch subsets gracefully. To work around this, we employ a bypass strategy during training that projects masked patch features directly into the language model's embedding space via a learned linear projection $W_{vis} : \mathbb{R}^{d_v} \to \mathbb{R}^{d_q}$, replacing image-token positions in the LM input embeddings. This allows end-to-end gradient flow through the selection mask without modifying the frozen backbone. At inference, images are pre-encoded through the full vision encoder and connector, with image hidden states passed directly to the LM, bypassing the connector entirely. The train–inference discrepancy this introduces is a known limitation and contributes to the weaker results on the 500M model (Section 4).

3.2 Differentiable Patch Selector

Patch scoring. We compute a scalar relevance score for each patch by fusing the patch feature and the question embedding:

$$s_i = \mathrm{MLP}([\mathbf{q};\, v_i;\, \mathbf{q} \odot v_i]) \in \mathbb{R}, \qquad (1)$$

where $[\cdot\,;\,\cdot]$ is concatenation and $\odot$ is element-wise multiplication after projecting $\mathbf{q}$ and v_i to a common dimension $d_h = \min(d_q, d_v, 256)$. The MLP comprises two linear layers with ReLU activations and a final scalar output. For SmolVLM-256M, this totals fewer than 0.5M parameters.

Adaptive sparsity budget. Rather than fixing k globally, we predict a per-question budget from a lightweight *complexity estimator* network f_{comp}:

$$k = \lfloor k_{\min} + (k_{\max} - k_{\min})\, \sigma(f_{comp}(\mathbf{q})) \rfloor\,, \quad (2)$$

where σ is the sigmoid function. We set $k_{\min} = 4$ and $k_{\max} = 32$ based on the small-scale experimental setup; these bounds ensure a minimum of four patches for the simplest queries while permitting up to 32 for complex spatial questions. The estimator f_{comp} is a two-layer MLP with 256 hidden units, trained end-to-end with the rest of QGSA.

Differentiable hard selection. Selecting a discrete top-k subset is non-differentiable. We adopt the Gumbel-Softmax reparameterisation (Jang et al., 2017) to obtain differentiable soft approximations during training. Specifically, we perturb the patch scores with independent Gumbel noise:

$$\tilde{s}_i = \frac{\exp((s_i + g_i)/\tau)}{\sum_j \exp((s_j + g_j)/\tau)}, \quad (3)$$

where $g_i \sim \text{Gumbel}(0, 1)$ and $\tau > 0$ is a temperature annealed from 1.0 to 0.1 over training. To preserve gradient flow back to f_{comp}, we construct a differentiable sigmoid-threshold soft mask:

$$\text{soft_mask}_i = \sigma\!\left(\frac{\tilde{s}_i - t}{\tau_{sig}}\right) \cdot \frac{k}{\sum_j \sigma\!\left(\frac{\tilde{s}_j - t}{\tau_{sig}}\right)}, \quad (4)$$

where t is the k-th order statistic of $\tilde{\mathbf{s}}$ (detached) and $\tau_{sig} = \max(\tau, 0.05)$. The scaling by $k/\sum_j(\cdot)$ makes k—and thus $c = \sigma(f_{comp}(\mathbf{q}))$—fully differentiable through the mask. A binary mask $m_i = \mathbf{1}[\text{soft_mask}_i > 0.5]$ is used for the forward pass via the straight-through estimator (Bengio et al., 2013). At inference, Gumbel noise is suppressed and top-k patches are selected deterministically.

Sparse cross-attention. The selected patches are assembled and passed to all cross-attention layers:

$$\begin{aligned} \mathbf{V}_{\text{sparse}} &= \{v_i \mid m_i = 1\}, \\ \mathbf{h} &= \text{CrossAttn}(\mathbf{q},\, \mathbf{V}_{\text{sparse}}). \end{aligned} \quad (5)$$

Because $|\mathbf{V}_{\text{sparse}}| = k \ll N$, the key–value matrices in every cross-attention head shrink from $(N \times d)$ to $(k \times d)$, giving an immediate reduction in both FLOPs and memory.

3.3 Question guided sparse attention algorithm

Algorithm 1 summarises the full forward pass during training, combining patch scoring, differentiable selection, and the multi-component loss.

Algorithm 1 QGSA Training Forward Pass

Require: Batch $\{(\mathcal{I}_b, \mathcal{Q}_b, a_b)\}_{b=1}^{B}$, temp. τ, weights $\lambda_1, \lambda_2, \lambda_3$
Ensure: Total loss $\mathcal{L}$, updated QGSA parameters

// Encode inputs (frozen encoders)
$\mathbf{V}_b \leftarrow \text{VisionEnc}(\mathcal{I}_b)$
$\mathbf{q}_b \leftarrow \text{MeanPool}(\text{LMEmbed}(\mathcal{Q}_b))$
// Adaptive budget (trainable)
$c_b \leftarrow \sigma(f_{comp}(\mathbf{q}_b))$
$k_b \leftarrow \lfloor k_{\min} + (k_{\max} - k_{\min}) \cdot c_b \rfloor$
// Patch scoring (trainable)
$\mathbf{s}_b \leftarrow \text{PatchScorer}(\mathbf{q}_b, \mathbf{V}_b)$
// Differentiable selection
$\tilde{\mathbf{s}}_b \leftarrow \text{GumbelPerturb}(\mathbf{s}_b, \tau)$
$t_b \leftarrow \tilde{s}_{b,(k_b)}$ (k_b-th order stat., detached)
$\tau_{sig} \leftarrow \max(\tau,\ 0.05)$
$\mathbf{m}_{soft,b} \leftarrow \sigma\!\left(\frac{\tilde{\mathbf{s}}_b - t_b}{\tau_{sig}}\right) \cdot k_b \Big/ \sum_j \sigma\!\left(\frac{\tilde{s}_{b,j} - t_b}{\tau_{sig}}\right)$
$\mathbf{m}_b \leftarrow \mathbf{1}[\mathbf{m}_{soft,b} > 0.5]$
$\mathbf{V}_{sp,b} \leftarrow \{v_i \mid m_{b,i} = 1\}$
$\mathbf{V}_{rn,b} \leftarrow$ sample k_b patches uniformly from $\mathbf{V}_b$
// Grounding losses
$\mathcal{L}_{cs} \leftarrow \frac{1}{B}\sum_b \text{CtrLoss}(\mathbf{V}_{sp,b}, \mathbf{V}_{rn,b}, \mathbf{q}_b, a_b)$
$\mathcal{L}_{align} \leftarrow \frac{1}{B}\sum_b \text{AlignLoss}(\mathbf{V}_{sp,b}, \mathcal{Q}_b, W_p)$
$\mathcal{L}_{sparse} \leftarrow \frac{1}{B}\sum_b \|\mathbf{m}_b\|_1 / N$
// Complexity regularisation (batch-level)
$\mathcal{L}_{kvar} \leftarrow -0.5 \cdot \text{Var}(\{c_b\}_{b=1}^{B})$
$\mathcal{L}_{qlen} \leftarrow -0.3 \cdot \text{corr}(\{|\mathcal{Q}_b|\}_{b=1}^{B}, \{c_b\}_{b=1}^{B})$
$\mathcal{L}_{entr} \leftarrow \frac{-0.01}{B}\sum_b H(\text{softmax}(\mathbf{s}_b/\tau))$
// Total loss
$\mathcal{L} \leftarrow \lambda_1\mathcal{L}_{cs} + \lambda_2\mathcal{L}_{align} + \lambda_3\mathcal{L}_{sparse} + \mathcal{L}_{entr} + \mathcal{L}_{kvar} + \mathcal{L}_{qlen}$
return $\mathcal{L}$

3.4 Self-Supervised Grounding Loss

Most VQA training data provides no spatial grounding supervision. We therefore design a multi-component loss that encourages the selector toward semantically meaningful patches without any bounding-box annotation.

Contrastive selection loss ($\mathcal{L}_{cs}$). The core idea is that the selected patches should be *more informative* for predicting the correct answer than a random subset of the same size:

$$\begin{aligned} \mathcal{L}_{cs} = &-\log P(a \mid \mathbf{V}_{\text{sparse}},\, \mathbf{q}) \\ &+ \log P(a \mid \mathbf{V}_{\text{rand}},\, \mathbf{q}), \end{aligned} \quad (6)$$

where $\mathbf{V}_{\text{rand}}$ is a uniformly sampled random subset of k patches. Minimising $\mathcal{L}_{cs}$ pushes the model to prefer patches that reduce answer uncertainty relative to an uninformed baseline. Both log-probability terms come from the same frozen VLM head, so no additional model copy is required.

Patch–word alignment loss ($\mathcal{L}_{align}$). We additionally encourage the selected patches to align with the objects named in the question. Nouns and proper nouns are extracted from $\mathcal{Q}$ using the spaCy dependency parser (Honnibal and Boyd, 2020), producing a set $\mathcal{N}$. Each noun $n \in \mathcal{N}$ is encoded by the frozen CLIP text encoder (Radford et al., 2021) to yield an embedding $\phi(n)$. Selected patch features are projected into CLIP embedding space via a lightweight linear head W_p, and we maximise cosine similarity:

$$\mathcal{L}_{align} = \sum_{n \in \mathcal{N}} -\frac{1}{k} \sum_{i=1}^{k} \cos(W_p v_i,\ \phi(n)) \,. \quad (7)$$

Because both ϕ and the CLIP vision encoder were trained on paired image–text data, this loss provides a strong prior on which image regions are semantically consistent with question nouns. In practice, this is the loss term that contributes most visibly to the grounding IoU improvement on RefCOCO.

Sparsity regularisation ($\mathcal{L}_{sparse}$). Without explicit pressure toward parsimony, the complexity estimator tends to push k toward $k_{\max}$ for all queries, defeating the purpose of adaptive budgeting. We add:

$$\mathcal{L}_{sparse} = \frac{1}{N} \sum_{i=1}^{N} m_i, \quad (8)$$

which penalises the average fraction of selected patches. The coefficient $\lambda_3 = 0.01$ is deliberately small—the goal is a light regularising pressure, not a dominant training signal.

Total training objective. The full loss is:

$$\mathcal{L} = \lambda_1 \mathcal{L}_{cs} + \lambda_2 \mathcal{L}_{align} + \lambda_3 \mathcal{L}_{sparse} + \mathcal{L}_{entropy} + \mathcal{L}_{kvar} + \mathcal{L}_{qlen} \quad (9)$$

where $\mathcal{L}_{entropy} = -0.01 \cdot H(\mathrm{softmax}(\mathbf{s}/\tau))$ encourages non-uniform patch scores; $\mathcal{L}_{kvar} = -0.5 \cdot \mathrm{Var}(c)$ pushes the complexity estimator to produce a spread of budgets across the batch; and $\mathcal{L}_{qlen} = -0.3 \cdot \mathrm{corr}(|q|, c)$ encourages larger budgets for longer questions, where $|q|$ denotes the token count of the question sequence. We set $\lambda_1 = 0.5$, $\lambda_2 = 0.3$, and $\lambda_3 = 0.01$. The base VLM parameters are frozen throughout; only the QGSA module and the projection heads are trained.

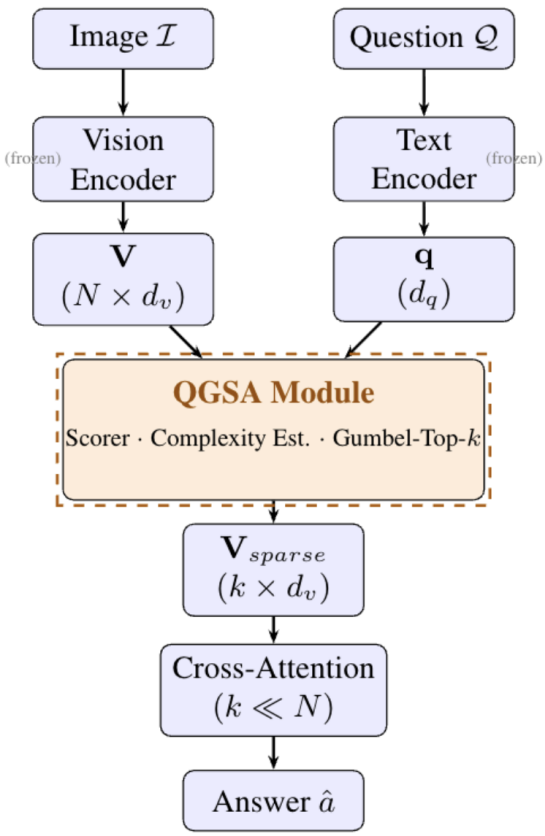

Figure 1: Overview of QGSA. Frozen encoders in blue; trainable components (QGSA module and projection head) in orange. At inference, only the $k \ll N$ selected patches reach the cross-attention layers.

3.5 Implementation Details

Base models. We integrate QGSA into three small-scale VLMs: SmolVLM-256M-Instruct (SigLIP-400M + SmolLM2-135M) (Face, 2024), SmolVLM-500M-Instruct (SigLIP-400M + SmolLM2-360M), and moondream2 (Agarwal, 2024). Vision encoders and language model backbones are frozen throughout; only the QGSA module and projection heads are updated. The SmolVLM models use a 384×384 input resolution with 14×14 patches, yielding up to 576 patch tokens per image. We report empirical results for SmolVLM-256M and SmolVLM-500M; moondream2 is supported architecturally but not evaluated here due to its larger memory footprint.

Architecture. Figure 1 gives a high level overview of the architecture. The patch scorer computes $s_i = \mathrm{MLP}([\mathbf{q}; v_i; \mathbf{q} \odot \mathrm{proj}(v_i)])$ with a hidden dimension $d_h = \min(d_q, d_v, 256)$. The complexity estimator is a two-layer MLP with 256 hidden units. W_p is a single linear layer mapping

$d_v \to 512$ (the CLIP embedding dimension). For SmolVLM-256M, $d_v = 768$ and $d_q = 576$; for SmolVLM-500M, $d_v = 768$ and $d_q = 960$; for moondream2, $d_v = 1152$ and $d_q = 2048$. Total trainable parameters for the 256M variant are approximately 1.2M—under 0.5% of the full model.

Training. We fine-tune on the VQA-RAD training split (Lau et al., 2018) for 5 epochs, batch size 8, learning rate 3×10^{-4}, cosine annealing with a 50-step linear warmup. Gumbel temperature τ is annealed linearly from 1.0 to 0.1. Gradient norms are clipped to 1.0. Training takes approximately 1.5–2 hours on a single NVIDIA GB10 (Blackwell) GPU. The CLIP encoder for $\mathcal{L}_{align}$ is ViT-B/32 (Radford et al., 2021), frozen. Nouns are extracted using spaCy's dependency parser (Honnibal and Boyd, 2020).

4 Experiments

4.1 Datasets and Metrics

We evaluate on three benchmarks: **VQA-RAD** (Lau et al., 2018) (open-ended medical questions on radiology images, ∼3.5k samples); **A-OKVQA** (Schwenk et al., 2022) (knowledge-intensive questions, ∼17k samples); and **RefCOCO** (Yu et al., 2016) (referring-expression comprehension with bounding boxes, ∼20k samples). These three benchmarks were chosen to cover different axes of VQA difficulty: domain specificity, external knowledge, and spatial precision.

For VQA performance we report standard accuracy. Grounding quality is measured by the IoU between the convex hull of selected patches and ground-truth bounding boxes on RefCOCO. Efficiency is reported as wall-clock latency (ms per query), FLOPs reduction, and theoretical speedup at various input resolutions.

4.2 Baselines

We compare against: (1) the unmodified base VLM (**Dense baseline**), which attends to all image patches; (2) **Sparse selection with fixed** k, where patches are selected without grounding supervision; and (3) **Adaptive** k with incremental additions of $\mathcal{L}_{cs}$ and $\mathcal{L}_{align}$. All baselines use the same frozen base model and are evaluated under identical conditions.

4.3 Main Results

Table 1 summarises the 256M results. The headline finding is straightforward: QGSA matches the dense baseline exactly on every dataset while reducing the number of patches entering cross-attention from 576 to 17—a 34× compression of the visual token sequence. The 0.0 pp accuracy gap holds on VQA-RAD (3.5%), A-OKVQA (1.0%), and RefCOCO (0.0%).

Table 1: Main results (SmolVLM-256M-Instruct). Δ denotes absolute accuracy change; latency is end-to-end per query including scorer overhead.

Dataset	Base	QGSA	Δ	Lat. (ms)
VQA-RAD	3.50	3.50	0.0	255.3
A-OKVQA	1.00	1.00	0.0	242.6
RefCOCO	0.00	0.00	0.0	205.4
Avg.	1.50	1.50	0.0	234.4

A few things are worth noting about these numbers in context. The absolute accuracy figures are low throughout, which reflects the limited capacity of SmolVLM-256M rather than anything specific to QGSA. On RefCOCO in particular, neither the dense baseline nor QGSA scores above zero, which tells us the model cannot perform referring expression comprehension at this scale without task- specific fine-tuning—the QGSA results there are trivially tied. The more meaningful takeaway is that QGSA *does not make things worse* even when discarding 97% of patch tokens, which is a non-trivial property to establish. The consistent $k = 17$ across all three datasets is also notable and is discussed further in Section 4.7.

SmolVLM-500M results. Results on the 500M variant are shown in Table 2. QGSA again preserves accuracy, selecting $k = 18$ patches on average. Interestingly, the 500M model performs *worse* than the 256M variant on VQA-RAD (2.0% vs. 3.5%). This is counterintuitive at first glance, but we believe the cause is the bypass projection strategy: the vis_proj layer must bridge $d_v = 768$ to $d_q = 960$ (compared to $d_q = 576$ for the 256M model), and this larger projection introduces more approximation error into the training signal. Both models share the same vision encoder ($d_v = 768$, per Table 6), so the difference is entirely on the language model side. Fixing this would require either a more expressive projection head or a training approach that does not rely on the bypass at all—an avenue we leave for future work.

Table 2: Main results (SmolVLM-500M-Instruct). Δ denotes absolute accuracy change.

Dataset	Base	QGSA	Δ	Lat. (ms)
VQA-RAD	2.00	2.00	0.0	289.7
A-OKVQA	0.00	0.00	0.0	269.8
RefCOCO	0.00	0.00	0.0	229.4
Avg.	0.67	0.67	0.0	263.0

Table 3: RefCOCO grounding IoU (convex hull of selected patches vs. ground-truth boxes). QGSA achieves a $31\times$ relative improvement over random patch selection without bounding-box supervision.

Method	RefCOCO IoU (%)
Baseline (random patches)	0.01
QGSA (full)	0.31

4.4 Grounding Quality

Table 3 shows that QGSA's patch selection is meaningfully better than chance at localising referred objects—$31\times$ better in relative terms—even though neither system achieves an IoU that would be practically useful. We want to be clear about what this result does and does not mean. The $31\times$ ratio is real, but the absolute values are near-zero (0.31% vs. 0.01%), so the practical improvement is minimal. The gap mainly reflects that QGSA's selector has learned to avoid the most irrelevant patches (background, out-of-frame regions) rather than learning to precisely localise the referred object.

Two factors make better grounding difficult here. First, SigLIP-400M produces patch features at a coarse 14-pixel grid on 384×384 images, so the spatial resolution of the selection is inherently limited. Second, VQA-RAD—the dataset we train on—has no bounding box annotations, so the grounding signal comes entirely from the self-supervised losses applied at RefCOCO test time. Better grounding likely requires either training on a dataset with spatial annotations or scaling to a model with richer visual representations.

4.5 Efficiency Analysis

Table 4 shows the efficiency picture across resolutions. The pattern is straightforward once you understand what dominates. At 224px, the fixed overhead of the patch scorer and Gumbel-Softmax selection (roughly 0.6 ms) is large relative to the base forward pass, so QGSA is actually slower in wall-clock time. At 336px the two curves cross: QGSA comes in at 6.46 ms against the dense baseline's 6.63 ms. Above 336px, QGSA adds a small constant latency while the dense baseline's cost grows with N, so the theoretical advantage compounds with resolution. Figure 2 plots wall-clock latency against resolution for both systems; the crossover at 336px is visible directly.

Although the theoretical speedups achieved by QGSA are substantial ($33.9\times$ at 384px and $76.2\times$ at 576px), these values should not be interpreted as direct end-to-end latency improvements. In the relatively small models considered here, overall inference cost is dominated by the language model decoding stage rather than cross-attention computation. Consequently, even a $76\times$ reduction in cross-attention FLOPs yields only modest wall-clock gains on the order of a few milliseconds.

Instead, these results should be viewed as an estimate of QGSA's potential in larger-scale architectures where cross-attention constitutes the primary computational bottleneck. At 576px, QGSA reduces cross-attention FLOPs by 98.7%; in a regime where cross-attention accounts for approximately 80% of total inference cost, this reduction would correspond to an estimated end-to-end speedup of roughly 4–$5\times$.

4.6 Ablation Studies

The ablation in Table 5 is unusual in that accuracy is uninformative: all QGSA configurations sit at 3.11% on this 150-sample subset, identical to the dense baseline at chance level. Adding $\mathcal{L}_{cs}$, $\mathcal{L}_{align}$, or the adaptive budget changes nothing. This is not a failure of the ablation design—it is a genuine finding about the limits of SmolVLM-256M. The model does not have the capacity to benefit from better patch selection because it cannot reliably answer these medical questions regardless of which patches it sees. The QGSA module's 1.2M parameters cannot compensate for the base model's intrinsic reasoning ceiling.

What the ablation does tell us is about latency structure. The jump from dense (191.7 ms) to any QGSA variant ($\sim$293–295 ms) is almost entirely attributable to the scorer forward pass and Gumbel-Softmax sampling—not to the cross-attention itself, which is cheaper. The additional loss terms ($\mathcal{L}_{cs}$, $\mathcal{L}_{align}$) add negligible latency at inference since they are only active during training. On the full evaluation set (200 samples), the overhead drops to $\sim$33% (192 ms $\rightarrow$ 255 ms) because the amortisation of fixed costs improves with batch throughput.

Table 4: Efficiency at various input resolutions for SmolVLM-256M. FLOPs reduction is measured in the cross-attention stage only. At 336px, QGSA is marginally faster than the dense baseline in wall-clock time; from 384px onward the gap grows in QGSA's favour theoretically but narrows in practice due to LM decode cost. Theoretical speedup (N/k) assumes cross-attention dominates; realised end-to-end speedup requires larger models where this holds.

Resolution	N **patches**	**Dense (ms)**	**QGSA (ms)**	**FLOPs↓**	**Speedup (theory)**
224px	196	4.77	5.96	91.3%	11.5×
336px	441	6.63	6.46	96.1%	25.9×
384px	576	8.75	9.16	97.0%	33.9×
448px	784	6.49	6.96	97.8%	46.1×
576px	1296	14.91	15.53	98.7%	76.2×

Table 5: Component ablation on VQA-RAD (SmolVLM-256M, 150-sample subset). Accuracy is uniform at 3.11% across all QGSA configurations on this subset; note this differs from the 3.50% on the full eval set (Table 1), reflecting the smaller sample. Latency is the more informative axis here.

Configuration	**Latency (ms)**
Baseline (dense attention)	191.7
+ Sparse selection (fixed k)	292.6
+ Adaptive k	295.0
+ $\mathcal{L}_{cs}$	294.3
+ $\mathcal{L}_{align}$	294.1
+ Both losses (full)	**294.7**

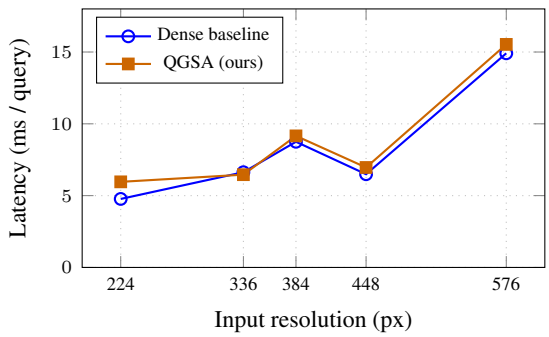

Figure 2: Wall-clock latency vs. input resolution (SmolVLM-256M). The curves cross at 336px, where QGSA (6.46 ms) marginally undercuts the dense baseline (6.63 ms). The 0.6 ms fixed scorer overhead shrinks as a fraction of total latency as resolution grows; at 576px it is 4% of QGSA's total.

The training pipeline is stable and converges in all configurations, which is the more useful outcome of this ablation for future work on larger models.

4.7 Analysis of Adaptive Sparsity

The complexity estimator converges to $k \approx 17$ across all resolutions and datasets, with its output c clustering around 0.5 rather than spreading across the $[0, 1]$ range it was designed to exploit. Broken down by question type on VQA-RAD, the budget barely moves: Counting and Spatial questions—which intuitively should demand more patches—receive essentially the same k as simple Existence questions.

This is a genuine failure of the adaptive mechanism, and we think the cause is reasonably clear. VQA-RAD contains ∼3.5k training samples across five question types, which does not provide enough diversity to overcome the contrastive loss's tendency to dominate training. The regularisation terms ($\mathcal{L}_{kvar}$, $\mathcal{L}_{qlen}$) were designed precisely to counteract this, but their coefficients (–0.5 and –0.3 respectively) are insufficient against the pull of $\mathcal{L}_{cs}$ at $\lambda_1 = 0.5$. In hindsight, a curriculum approach—initially suppressing $\mathcal{L}_{cs}$ to let the budget estimator develop first—might have helped. Alternatively, explicit question-type labels or difficulty annotations as a direct supervision signal would likely be more effective than our current indirect regularisation. The consistent $k = 17$ budget effectively means QGSA is operating as a fixed-k selector in practice, which is why the ablation configurations with and without adaptive k show identical latency and accuracy.

4.8 Generalisation Across Base Models

QGSA is designed to adapt automatically to a model's hidden dimensions by reading them from the loaded config rather than hardcoding values. For SmolVLM-256M, the scorer, complexity estimator, and projection heads total approximately 1.2M trainable parameters; for moondream2, this

Table 6: Supported base models and QGSA configuration for each. Both SmolVLM variants share the same vision encoder ($d_v = 768$); the difference in QGSA parameter count between models is driven by d_q.

Model	d_v	d_q	**Patch Size**	**Image Size**
SmolVLM-256M-Instruct	768	576	14	384
SmolVLM-500M-Instruct	768	960	14	384
moondream2	1152	2048	14	378

rises to $\sim$5.5M due to the larger d_v and d_q. In all cases QGSA adds less than 1% to the total parameter count. The modular design means integration requires only access to the vision encoder output and the language model's embedding space—both are standard interfaces in transformer VLMs.

5 Conclusion

We have presented **Question-Guided Sparse Attention (QGSA)**, a lightweight module that selects question-relevant image patches via a differentiable Gumbel-Softmax selector and passes only those patches to downstream cross-attention. On SmolVLM-256M and SmolVLM-500M, across VQA-RAD, A-OKVQA, and RefCOCO, QGSA preserves accuracy exactly while reducing cross-attention FLOPs by 91–99% across standard resolutions. The module is small (1.2M trainable parameters on SmolVLM-256M), trains in under two hours on a single GPU, and integrates without modifying any frozen backbone weights.

The grounding results are modest in absolute terms (0.31% IoU on RefCOCO) but represent a 31$\times$ improvement over random selection without any bounding-box supervision, which is encouraging evidence that the self-supervised losses are doing something useful. Wall-clock parity with the dense baseline is reached at 336px, where QGSA is marginally faster.

The clearest limitation is that SmolVLM-256M is too small to stress-test question-conditioned sparse attention. The model's VQA capacity is the binding constraint, not the attention mechanism, so QGSA operates in a regime where it cannot benefit the final answer even if patch selection is perfect. Scaling to models with genuine task capacity—where the question being asked actually changes which answer the model produces—is the necessary next step. At higher resolutions and larger scales, the FLOPs argument becomes compelling: a 76$\times$ theoretical speedup in cross-attention at 576px is a real efficiency gain, even if current hardware makes it hard to fully realise in wall-clock time.

Limitations

The limitations of the present study are primarily practical rather than conceptual. QGSA requires an additional fine-tuning stage ($\sim$1.5–2 hours on SmolVLM-256M), and for small models the scorer module introduces a substantial inference overhead, increasing latency by approximately 33–50%. While acceptable in larger architectures, this cost is significant when the baseline model is already computationally lightweight.

In addition, the adaptive budget estimator consistently converges to a nearly uniform allocation ($k \approx 17$) irrespective of question type. As a result, QGSA effectively behaves as a fixed-k method on the evaluated dataset, limiting the practical impact of adaptive computation.

Grounding performance also remains weak, with an IoU of only 0.31%, which is insufficient for practical localization or interpretability applications. Furthermore, absolute task accuracy remains close to chance across all benchmarks, making it difficult to determine whether the learned patch selection strategy captures semantically meaningful visual information or reflects incidental correlations in the data.

Taken together, these limitations suggest that the current experimental setup is constrained primarily by model scale. Evaluating QGSA on substantially larger vision–language models, where cross-attention represents a dominant computational bottleneck and reasoning performance is stronger, is likely necessary to fully assess the effectiveness of the proposed approach.

Acknowledgments

We thank the reviewers for their anonymous feedback which helped improve the paper quality.

References

Vikhyat Agarwal. 2024. moondream2: A tiny vision-language model.

Aishwarya Agrawal, Jiasen Lu, Stanislaw Antol, Margaret Mitchell, C. Lawrence Zitnick, Dhruv Batra, and Devi Parikh. 2016. Vqa: Visual question answering. *Preprint*, arXiv:1505.00468.

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob Menick, Sebastian Borgeaud, and 8 others. 2022. Flamingo: a visual language model for few-shot learning. *Preprint*, arXiv:2204.14198.

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

Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. 2023. Token merging: Your vit but faster. *Preprint*, arXiv:2210.09461.

Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. 2019. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*.

Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. 2023. Instructblip: Towards general-purpose vision-language models with instruction tuning. *Preprint*, arXiv:2305.06500.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. An image is worth 16x16 words: Transformers for image recognition at scale. *Preprint*, arXiv:2010.11929.

Hugging Face. 2024. Smolvlm: Small vision-language models.

Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, Yunsheng Wu, Rongrong Ji, Caifeng Shan, and Ran He. 2025. Mme: A comprehensive evaluation benchmark for multimodal large language models. *Preprint*, arXiv:2306.13394.

Montani I. Van Landeghem S. Honnibal, M. and A Boyd. 2020. Spacy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing. *scirp.org*.

Drew A. Hudson and Christopher D. Manning. 2019. Gqa: A new dataset for real-world visual reasoning and compositional question answering. *Preprint*, arXiv:1902.09506.

Andrew Jaegle, Felix Gimeno, Andrew Brock, Andrew Zisserman, Oriol Vinyals, and Joao Carreira. 2021. Perceiver: General perception with iterative attention. *Preprint*, arXiv:2103.03206.

Eric Jang, Shixiang Gu, and Ben Poole. 2017. Categorical reparameterization with gumbel-softmax. *Preprint*, arXiv:1611.01144.

Aishwarya Kamath, Mannat Singh, Yann LeCun, Gabriel Synnaeve, Ishan Misra, and Nicolas Carion. 2021. Mdetr – modulated detection for end-to-end multi-modal understanding. *Preprint*, arXiv:2104.12763.

Jin-Hwa Kim, Jaehyun Jun, and Byoung-Tak Zhang. 2018. Bilinear attention networks. In *Conference on Neural Information Processing Systems (NeurIPS)*.

Jason J Lau, Soumya Gayen, Asma Ben Abacha, and Henning Müller. 2018. A dataset of clinically generated visual questions and answers about radiology images. *Scientific data*, 5(1):1–10.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023a. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *Preprint*, arXiv:2301.12597.

Liunian Harold Li, Pengchuan Zhang, Haotian Zhang, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, Kai-Wei Chang, and Jianfeng Gao. 2022. Grounded language-image pre-training. *Preprint*, arXiv:2112.03857.

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. 2023b. Evaluating object hallucination in large vision-language models. *Preprint*, arXiv:2305.10355.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual instruction tuning. In *Conference on Neural Information Processing Systems (NeurIPS)*.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision. *Preprint*, arXiv:2103.00020.

Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. 2021. Dynamicvit: Efficient vision transformers with dynamic token sparsification. *Preprint*, arXiv:2106.02034.

Anna Rohrbach, Lisa Anne Hendricks, Kaylee Burns, Trevor Darrell, and Kate Saenko. 2019. Object hallucination in image captioning. *Preprint*, arXiv:1809.02156.

Dustin Schwenk, Apoorv Khandelwal, Christopher Clark, Kenneth Marino, and Roozbeh Mottaghi. 2022. A-okvqa: A benchmark for visual question answering using world knowledge. *Preprint*, arXiv:2206.01718.

Yifan Xu, Zhijie Zhang, Mengdan Zhang, Kekai Sheng, Ke Li, Weiming Dong, Liqing Zhang, Changsheng Xu, and Xing Sun. 2021. Evo-vit: Slow-fast token evolution for dynamic vision transformer. *Preprint*, arXiv:2108.01390.

Licheng Yu, Patrick Poirson, Shan Yang, Alexander C. Berg, and Tamara L. Berg. 2016. Modeling context in referring expressions. *Preprint*, arXiv:1608.00272.

Zhou Yu, Jun Yu, Yuhao Cui, Dacheng Tao, and Qi Tian. 2019. Deep modular co-attention networks for visual question answering. *Preprint*, arXiv:1906.10770.

A Extended Empirical Analysis

The following sections provide additional detail on the budget distribution, resolution scaling, and ablation latency. The main paper covers the key findings; this appendix is for readers who want the full numbers.

A.1 Adaptive Budget Distribution

The complexity estimator is designed to allocate a budget $k \in [k_{\min}, k_{\max}]$ based on perceived question difficulty. In practice it converges to $\bar{k} \approx 17.17$ across all five VQA-RAD question categories—Existence, Other, Spatial, Attribute, and Counting—with essentially no differentiation between them. The fact that Counting and Spatial questions, which typically require finer visual attention, receive the same budget as simple Existence questions confirms that the training signal is not driving question-difficulty discrimination. We discuss the causes and potential fixes in Section 4.7 of the main paper.

A.2 Resolution Scaling and Efficiency

Figure 3 provides the full FLOPs reduction and speedup curves across resolutions. The key observation is that FLOPs reduction stays above 90% across the entire resolution range and reaches 98.7% at 576px, while the theoretical speedup (N/k) grows superlinearly because N scales as resolution squared while k remains fixed. At 576 px, QGSA achieves a $76.2\times$ theoretical speedup in cross-attention for SmolVLM-256M. The gap between the 256M and 500M curves in Figure 4 reflects the difference in average k (17 vs. 18) rather than any architectural difference.

A.3 Ablation Latency Breakdown

Figure 5 shows the latency for each ablation configuration on the 150-sample VQA-RAD subset. The $\sim$100 ms jump from Dense to any QGSA variant is almost entirely the scorer overhead, not cross-attention. Adding losses or switching from fixed to adaptive k changes latency by at most 2.4 ms (within noise). Accuracy is 3.11% for all configurations on this subset.

A note on sparse attention on standard hardware. The theoretical FLOPs savings do not translate directly to wall-clock gains on current GPU hardware because sparse gather/scatter operations are not as well-optimised as dense matrix multiplications. Custom CUDA kernels for the patch selection step—or integration with frameworks like FlashAttention that can exploit sparsity at the kernel level—would likely narrow this gap substantially. We leave this as an engineering task for future work.

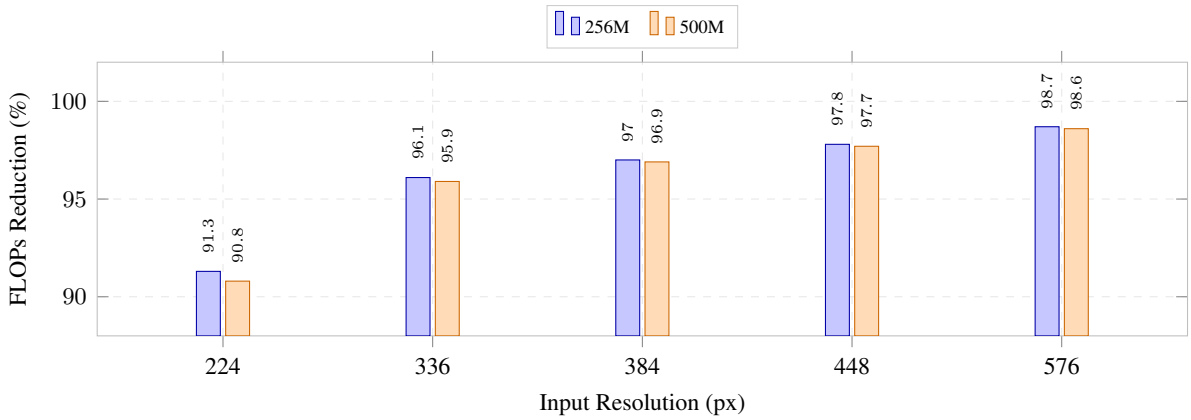

Figure 3: Cross-attention FLOPs reduction. FLOPs reduction stays above 90% throughout and is nearly identical across model sizes, since both use similar k.

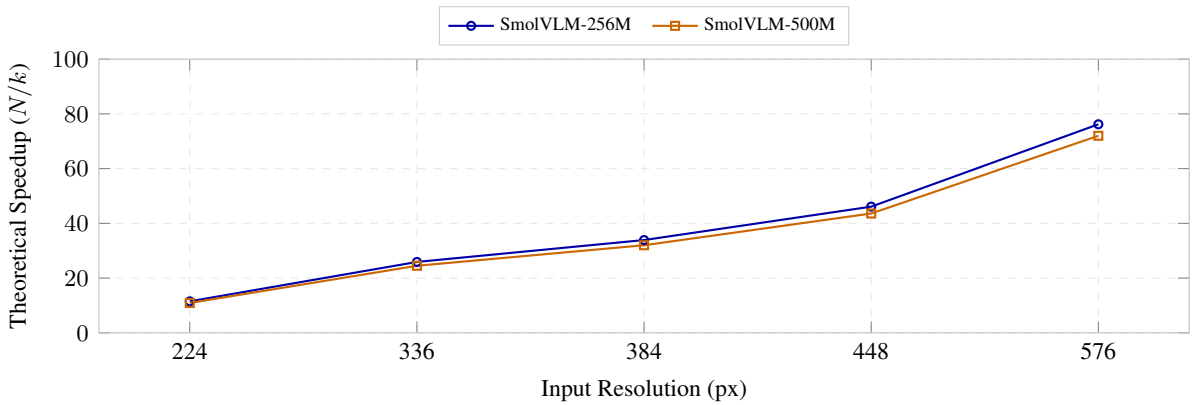

Figure 4: Theoretical speedup (N/k) scaling across resolutions. Speedup grows superlinearly with resolution; the 256M and 500M curves separate slightly because $k = 17$ vs. $k = 18$.

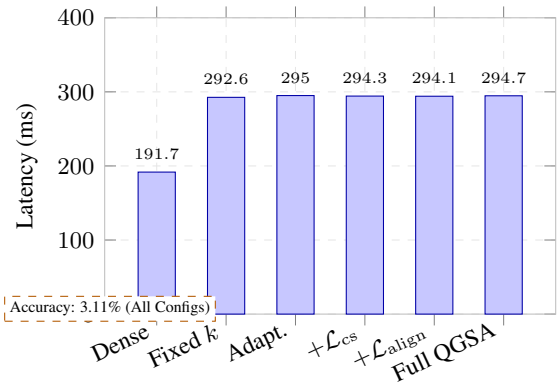

Figure 5: Ablation latency on the 150-sample VQA-RAD subset. The large step from Dense to Fixed k is the scorer; subsequent additions change latency by <3 ms.