CARES: Context-Aware Resolution Selector for VLMs

Anonymous Authors[1]

Abstract

Large vision–language models (VLMs) commonly process images at native or high resolution to remain effective across tasks. This inflates visual tokens up to to 97-99% of total tokens, resulting in high compute and latency, even when low-resolution images would suffice. We introduce *CARES*—a **C**ontext-**A**ware **R**esolution **S**elector, a lightweight preprocessing module that, given an image–query pair, predicts the *minimal* sufficient input resolution. CARES uses a compact VLM (350M) to extract features and predict when a target pretrained VLM's response converges to its peak ability to answer correctly. Though trained as a discrete classifier over a set of optional resolutions, CARES interpolates continuous resolutions at inference for fine-grained control. Across five multimodal benchmarks spanning documents and natural images, as well as diverse target VLMs, CARES preserves task performance while reducing compute by up to 80%.

1. Introduction

Large vision–language models (VLMs) are increasingly used as general-purpose systems that solve a broad variety of visual tasks using a single model. Since the complexity and nature of each task are not known in advance, these models typically process images at very high resolutions to preserve the visual detail necessary for any potential query. This leads to a sharp increase in the number of visual tokens, as modern architectures map higher resolutions to proportionally more tokens. Strategies like AnyRes and tiling further increase token counts in order to capture fine-grained information (Liu et al., 2024a; Wang et al., 2024). In practical settings, visual tokens often make up 97–99% of all tokens processed per request, which significantly impacts latency and memory consumption, even when the actual query may only require a coarse understanding of the scene.

A key observation is that *not all queries require the same visual granularity*. Coarse queries (e.g., "What is the breed of the dog?") are typically answerable from a small image; fine-grained queries (e.g., "What is the name on the collar?") benefit from higher resolution. Existing efficiency methods typically operate *after* tokenization, on the output of the vision encoder -pruning, pooling, merging, or compressing with Q-former style architecture Arif et al. (2025); Zhang et al. (2025c); Xing et al. (2025); Lin et al. (2025); Rao et al. (2021); Liang et al. (2022); Bolya et al. (2023); Hu et al. (2025); Cai et al. (2025). While complementary, these methods typically operate on the output of the visual encoder alone and are unaware of the text input or the current query.

We propose a *Context-Aware Resolution Selector* (**CARES**), a lightweight model that, for a given image-query pair, selects the *minimal* sufficient resolution to answer the query (Fig. 1). CARES is model-agnostic, placed *in front of* an arbitrary VLM. While our main instantiation uses a compact frozen VLM with a lightweight discriminative classifier, the CARES formulation is not tied to a specific predictor architecture. We also study a closely related autoregressive instantiation based on Granite-Docling, fine-tuned with LoRA, and report it separately on document-centric benchmarks.

It operates in three steps:

- A cheap low-resolution pass (e.g., $\leq 384^2$) extracts a joint image–query representation using a small proxy VLM.
- Given this representation, a lightweight classifier predicts the minimal resolution required for the task.
- The image is resized to the predicted resolution and passed to the target VLM. No changes to the VLM's architecture, weights, or training are required.

A central challenge is supervision: what resolution is *truly* sufficient for each example? We introduce a simple labeling procedure based on a discrete set of resolutions $\mathcal{R}$ and a task performance metric. For each image, query, and GT response, we evaluate a pretrained VLM with increasingly higher resolution up to convergence in terms of the task metric (or reaching the native resolution). The lowest resolution at which the convergence occurs is selected as the ground-truth optimal resolution for training CARES. Using a discrete resolution set avoids the cost of exhaustively searching over continuous values. Since the labels are discrete, the model is trained as a classifier. At inference time, however, we interpolate between the predicted class probabilities to recover a continuous resolution estimate.

Across 5 multimodal benchmarks, varying from natural images to document understanding (Section 4) and different open and api-based model, CARES reduces average visual tokens and GFLOPS by 70-80%, with minimal to no accuracy drop compared to always using the highest (native) resolution.

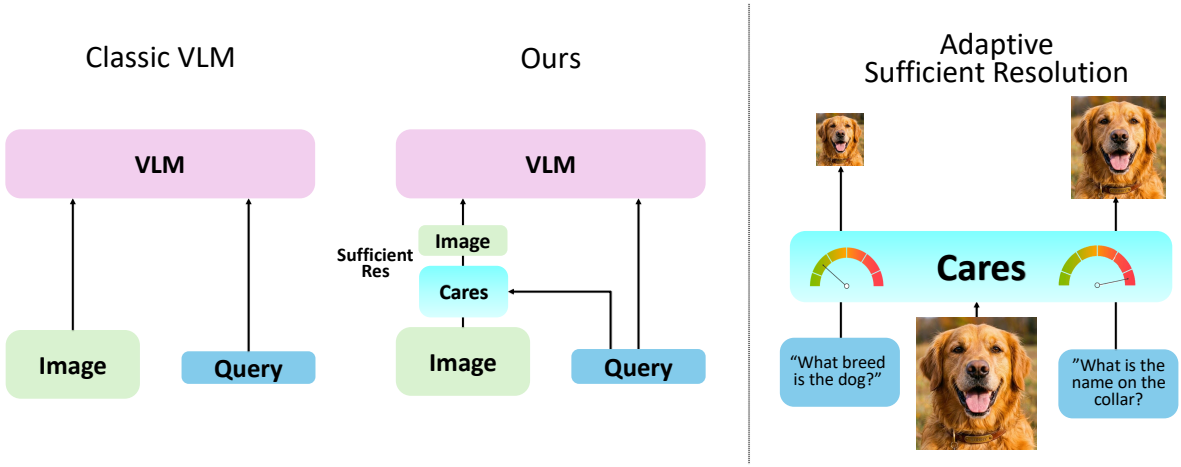


Figure 1. Overview of **CARES**. On the left, we compare the treditional pipeline of a use of VLM vs the pipeline using CARES. Given an image and its query, CARES predicts the minimal sufficient input resolution. The image is resized accordingly and, together with the query, passed to a downstream VLM. Coarse queries are routed to lower resolution; fine-grained queries that require more detail trigger higher resolution, which yields more visual tokens in the VLM.

Our contributions are as follows: ⋆ We define the task of *query- and image-conditioned resolution selection* for vision-language models, aimed at reducing input size without sacrificing accuracy.

⋆ We propose a simple yet effective supervision strategy based on multi-resolution rollouts and a convergence rule, yielding per-example sufficient resolution ground-truth, enabling training and evaluation.

⋆ We introduce CARES, a lightweight, model-agnostic module that selects resolution as a pre-processing step, requiring no changes to the target VLM.

⋆ We demonstrate that many visual tokens are unnecessary: CARES preserves performance across tasks while reducing visual compute by up to 85%, and is orthogonal with post-tokenization token compression.

2. Related Work

Visual-token sparsification at inference A growing line of work trims visual tokens *after* tokenization inside the VLM stack. HiRED uses `[CLS]` attention to allocate a per-partition token budget and drop the least-informative vision tokens under a fixed budget (Arif et al., 2025). SparseVLM proposes a training-free, text-guided strategy: self-attention matrices rank visual tokens with an adaptive layer-wise sparsification ratio and a token-recycling mechanism to preserve information (Zhang et al., 2025c). PyramidDrop stages the model and progressively reduces tokens at stage boundaries, motivated by the observation that redundancy increases with depth; it accelerates both training (Xing et al., 2025). Complementary to these, Visual Tokens Withdrawal (VTW) argues that visual information migrates to text tokens in early layers and thus withdraws vision tokens beyond a learned layer, cutting compute while maintaining quality (Lin et al., 2025). In contrast, CARES decides *before* tokenization which input resolution to use and leaves all VLM's components frozen.

Training for flexible token budgets TokenFLEX trains VLMs to operate across a range of visual–token counts by stochastically modulating tokens during training and adding a lightweight projector with adaptive pooling (Hu et al., 2025). *Matryoshka Multimodal Models* pursue training nested representations under progressively smaller token budgets (Cai et al., 2025). *LLaVA-Mini* compressing visual information into nearly single token in the LLM (Zhang et al., 2025b). CARES targets the complementary axis of *adaptive pixel allocation* before tokenization: it selects the minimal input resolution needed for a target utility and can front-end TokenFLEX/Matryoshka/LLaVA-Mini–style models to reduce pixels (and thus tokens) further.

Any-resolution inputs and tiling Many modern ViTs (Dehghani et al., 2023; Beyer et al., 2023) and VLMs boost fine-grained perception with AnyRes/dynamic-high-resolution tiling (e.g., LLaVA-NeXT) or native dynamic resolution that maps larger images to more tokens (e.g., Qwen2-VL) (Liu et al., 2024a; Wang et al., 2024). While effective, these strategies often increase visual tokens substantially. CARES explicitly *avoids* unnecessary tiling by routing easy cases to low resolutions and only escalating when the query and low-res cues predict a benefit.

Dynamic computation Vision-only methods reduce computation via token pruning/merging inside ViTs-e.g., DynamicViT prunes tokens hierarchically with learned

importance (Rao et al., 2021), EViT reorganizes/discards inattentive tokens (Liang et al., 2022), and ToMe merges similar tokens on the fly (Bolya et al., 2023).WAVECLIP replaces patch tokenization with a multi-level wavelet tokenizer and performs coarse-to-fine inference in a single ViT (Kimhi et al., 2025). For VLMs, SGL routes easy cases via a small 'stitch' model and defers hard ones to a larger counterpart, akin to early-exit routing (Zhao et al., 2024). These operate *within* the encoder after tokenization; CARES is complementary, deciding how many pixels to tokenize in the first place.

Adaptive input resolution selection Outside VLMs, dynamic-resolution networks learn a per-image resolution predictor that trades accuracy for cost in classification (Zhu et al., 2021). CARES brings this idea to multimodal QA, conditions the policy on the query text, and supervises it with *per-example* multi-resolution rollouts of the target VLM using a sufficiency rule, which yields unambiguous labels at deployment resolutions.

Extreme compression and design insights Recent analyses argue that, under fixed inference budgets, compute-optimal VLMs may prefer very few visual tokens and a larger LLM (Li et al., 2024). Such results support approaches that minimize visual tokens when possible; methods like *LLaVA-Mini* instantiate the "one-token vision" regime in practice (Zhang et al., 2025b). CARES provides a query-conditioned mechanism to reduce pixels upstream, complementing these token-minimal designs.

3. CARES

This section outlines the problem addressed by CARES (3.1), followed by a description of the dataset generation procedure (3.2). We then detail the architecture and the training details of CARES (3.3), Finally we outline our continuous resolution approach (3.4).

3.1. Problem Definition

Given an image x and query q, let $\mathcal{R}=[r_{\min},r_{\max}]\subset\mathbb{R}^+$ denote the range of valid input resolutions and let F be a fixed VLM. For any resolution $r\in\mathcal{R}$, we denote by $x^{(r)}$ the image x resized such that its largest dimension equals r. Feeding $x^{(r)}$ and q into F yields an output $y=F(x^{(r)},q)$. The VLM forms $T(r)$ visual tokens at resolution r (including AnyRes/tiling effects). Our goal is to learn a *selector* f_θ that predicts, from a single inexpensive low-resolution pass at r_{min}, the minimal *sufficient* resolution $r_s\in\mathcal{R}$ for accurately answering the query q given image x.

3.2. Labeling Strategy for Training CARES

Since searching for the optimal $r^\star\in\mathcal{R}$ is prohibitively expensive, we chose to use a small, discrete set of valid resolutions for the annotation $\mathcal{R}_d=\{r_1,\ldots,r_K\}\subset\mathcal{R}$. For each sample, we render the image at the fixed resolutions, $\mathcal{R}_d$, and use a pretrained VLM to generate predictions at each resolution. The predictions are evaluated against the ground-truth annotations using the ANLS metric. The supervision label is assigned as the lowest resolution whose ANLS score exceeds a threshold, without significant improvement at higher resolutions. The procedure yields a *discrete* sufficiency label $r^\star\in\mathcal{R}_d$ per example. We emphasize that discretization is only used for supervision efficiency; at inference, we deploy a *continuous* finer-grained selector (§3.4). Algorithm 1 outlines the data generation process, and Table. 6 visualizes the concept.

Algorithm 1 Labeling via multi-resolution sufficiency rollouts.

Input: (x,q); resolutions $\mathcal{R}$; VLM F; utility U; threshold τ; margin δ
Output: Label $r^\star\in\mathcal{R}$
for $k\leftarrow 1$ **to** K **do**
  $y_k\leftarrow F(x^{(r_k)},q)$; $u_k\leftarrow U(y_k,\text{gt})$
for $k\leftarrow 1$ **to** K **do**
  if $u_k\geq\tau$ ***and*** $\max_{\ell>k}(u_\ell-u_k)\leq\delta$ **then**
    return $r^\star\leftarrow r_k$
return $r^\star\leftarrow r_K$

Formally, we compute the ANLS score for each resolution:

$$u_k=\text{ANLS}\Big(F(x^{(r_k)},q),\text{gt}\Big)\in[0,1] \tag{1}$$

and select the minimal sufficient resolution as:

$$r^\star=\min\Big\{r_k\,\Big|\,u_k\geq\tau,\max_{\ell>k}(u_\ell-u_k)\leq\delta\Big\} \tag{2}$$

where we default to r_K if no resolution satisfies the condition. We set τ=0.85 and use a small margin δ (e.g., 0.1) to prevent rewarding negligible performance improvements. We define the full resolution range as $\mathcal{R}=[384,1024]$, and use a discrete set $\mathcal{R}_d=\{384,768,1024\}$ for annotation.

3.3. Model Instantiations

Unless otherwise stated, all main experiments in this paper use the following discriminative instantiation of CARES.

We design CARES as a lightweight resolution selector that can be deployed in front of any vision–language model (VLM) to improve efficiency. Its behavior is governed by three core principles:

Compactness: minimal overhead in computation and memory.
Preprocessing role: determines resolution directly from raw inputs before invoking the VLM.
VLM-agnosticism: works with any VLM, whether run locally or accessed via API, with no architecture changes or retraining required.

To implement these principles, we use a compact frozen VLM backbone as a joint vision–text feature extractor, followed by a lightweight classifier head.

We adopt the pretrained SmolVLM-500M model (Marafioti et al., 2025), with layers 17–32 removed, as the backbone. Given an image at resolution $r_{\min}$ and a text query, we feed both into the model and extract the hidden state of the final token at layer 16. This representation encodes the joint image–query context and is passed to a classifier that outputs a soft distribution over target resolutions. This design is motivated by recent findings showing that intermediate layer activations in LLMs and VLMs encode rich perceptual and semantic information that may not be surfaced at the output layer (Orgad et al., 2024; Zhang et al., 2025a). The claim evidenced by the performance gap in Table 7 where using intermediate features outperforms last-layer features.

The resulting CARES module has approximately 350M parameters and is trained with supervision over discrete resolution labels (see §3.2).

Autoregressive document-specialized instantiation. In addition to the discriminative selector above, we also instantiate CARES using an autoregressive vision-language model. Concretely, we start from Granite-Docling-258M (Auer et al., 2024) and fine-tune it with LoRA (rank 8) on the same resolution-selection training set. Given the low-resolution image and the query, the model is prompted to predict one resolution label from the discrete set $\mathcal{R}_d = \{384, 768, 1024\}$. To avoid tokenization ambiguity, we map these labels to dedicated tokens <1>, <2> and <3>.

At inference time, we read the first-step logits over the resolution tokens, apply a softmax to obtain class probabilities, and use the same expectation-based interpolation described in Eq. 3 to produce a continuous resolution. This preserves the deployment rule of CARES while replacing the discriminative head with an autoregressive predictor.

3.4. From Discrete Supervision to a Continuous Resolution

Although CARES is trained as a K-way classifier over a discrete set of resolutions $\mathcal{R}_d = \{r_1 < \cdots < r_K\}$, we deploy it as a *continuous* selector over $\mathcal{R} = [r_{min}, r_{max}]$. Given features z from the low-resolution image and query, compute logits $\ell(z) \in \mathbb{R}^K$ and class probabilities

$$p = \mathrm{softmax}(\ell),$$

We use the probability-weighted expectation over $\mathcal{R}_d$:

$$\tilde{r} = \sum_{k=1}^{|\mathcal{R}_d|} p_k r_k, \tag{3}$$

This yields a *continuous* resolution that varies smoothly with confidence and is insensitive to the specific discretization used for labeling. In practice, $\tilde{r}$ preserves the routing behavior of the classifier while allowing finer control.

Continuous inference algorithm.

Algorithm 2 Continuous resolution selection.

Input: (x, q); low-res r_1; logits ℓ.
Output: Continuous resolution $\tilde{r} \in [r_1, r_K]$.
$z \leftarrow$ features from proxy VLM at r_1 $p \leftarrow \mathrm{softmax}(\ell(z))$
$\tilde{r} \leftarrow \sum_{k=1}^{K} p_k r_k$ **return** $\tilde{r}$

Deployment. The target VLM receives x with the largest dimension resized to $\tilde{r}$ (or to the nearest supported side length to avoid under-allocation). For backbones that only accept a discrete set of input sizes, we round *up* to the next supported size.

4. Results & Analysis

4.1. Experimental Setup

Training Data To train the resolution selector, we construct a dataset of images and queries (x, q) we automatically annotated with the minimal sufficient resolution $r^\star$. We construct an 80K-sample training set by randomly sampling 20K instances from each of four datasets: TextVQA (Singh et al., 2019), ChartQA (Masry et al., 2022), DocVQA (Mathew et al., 2021), and LLaVA-Multi (Jiang et al., 2024), covering documents and natural images domains.

Training details We train CARES on the curated data described in 3.2 for 6 epochs using a learning rate of $1e-3$ and a batch size of 32. We optimize the standard cross-entropy loss over the fixed resolution labels:

$$\mathcal{L}(\theta) = \mathrm{CE}\Big(f_\theta(z), r^\star\Big).$$

Where $f_\theta(z)$ is CARES composed of a frozen VLM and the lightweight classifier. In addition, we apply label smoothing of 0.05 to support continuous resolutions at inference time.

VLM variant training details. For the autoregressive (AR) Granite-Docling instantiation, we use the same training set and the same discrete supervision labels. The model is fine-tuned with LoRA of rank 8, while the base model remains frozen. Training is performed with next-token supervision over the resolution tokens, and for efficiency, generation length is set to 1.learning rate is set to $1e-5$ and a batch size of 64 for 3 epochs.

Evaluation We evaluate on five public benchmarks varying from documents to natural images: Ai2D (Kembhavi et al., 2016), ChartQA (Masry et al., 2022), DocVQA (Mathew et al., 2021), OCRBench (Liu et al., 2024b), and SeedBench-2 (Li et al., 2023). For Ai2D, ChartQA, and SeedBench-2 we report exact-match accuracy. For DocVQA and OCRBench we report Average Normalized Levenshtein Similarity (ANLS). All evaluations were performed with the standard lmms-eval (Zhang et al., 2024) setup. We also report a macro-averaged Performance (%) across all datasets.

Model	Ai2D		ChartQA		DocVQA		OCRBench		SeedBench-2		MMMU		RealWorldQA		InfoVQA		MathVista		Average	
	Score	Cost	Score	Cost	Score	Cost	Score	Cost	Score	Cost	Score	Cost	Score	Cost	Score	Cost	Score	Cost	Score	Cost
Granite-Vision -2B	0.74		0.86		0.90		0.80		0.72		0.29		0.17		0.35		0.48		0.59	
+ CARES	0.73	-67%	0.87	-69%	0.90	-68%	0.80	-68%	0.72	-44%	0.29	-85%	0.19	-72%	0.40	-72%	0.48	-22%	0.60	-63%
+ CARES-AR	0.71	-81%	0.84	-81%	0.88	-82%	0.77	-75%	0.72	-10%	0.30	-84%	0.15	-82%	0.39	-81%	0.44	-25%	0.58	-67%
InternVL3-8B	0.84		0.86		0.92		0.85		0.79		0.56		0.68		0.72		0.69		0.77	
+ CARES	0.84	-66%	0.86	-68%	0.92	-69%	0.85	-70%	0.79	-44%	0.56	-86%	0.68	-82%	0.74	-72%	0.69	-22%	0.77	-64%
+ CARES-AR	0.84	-86%	0.86	-81%	0.92	-80%	0.85	-78%	0.72	-84%	0.55	-85%	0.68	-82%	0.74	-81%	0.68	-31%	0.76	-76%
Qwen2.5-VL-72B	0.87		0.87		0.96		0.75		0.81		0.62		0.77		0.73		0.74		0.79	
+ CARES	0.87	-85%	0.84	-77%	0.95	-84%	0.76	-64%	0.79	-77%	0.62	-86%	0.79	-82%	0.84	-72%	0.74	-7%	0.80	-70%
GPT-4o	0.78		0.56		0.80		0.77		0.76		0.57		0.61		0.75		0.64		0.69	
+ CARES	0.78	-60%	0.56	-60%	0.80	-36%	0.75	-33%	0.75	-47%	0.56	-85%	0.61	-84%	0.73	-76%	0.61	-17%	0.68	-55%
+ CARES-AR	0.74	-85%	0.52	-85%	0.78	-88%	0.73	-84%	0.71	-82%	0.56	-85%	0.62	-84%	0.71	-82%	0.58	-28%	0.66	-78%

Table 1. **Benchmark performance** and estimated prefill-stage savings for **Cost** (measured in FLOPS for local models or $ for API models).

4.2. Main results

We evaluate CARES across **Granite-Vision 3.3-2B** (Team et al., 2025), **InternVL3-8B** (Zhu et al., 2025), **Qwen2.5-VL-72B** (Bai et al., 2025), and **GPT-4o** (Achiam et al., 2023). We also report prefill-stage FLOPS savings for locally run models, and estimated dollar savings in API usage for GPT-4o. As summarized in Table 1, CARES maintains accuracy while cutting prefill compute: averaged over models and datasets, prefill FLOPs drop by **65–85%** with at most a sub-point change in macro performance relative to always using the highest/native resolution. The effect is consistent for compact (Granite-Vision 3.3-2B) and large (Qwen2.5-VL-72B) backbones, and holds for GPT-4o accessed via API (accuracy parity at comparable quality).

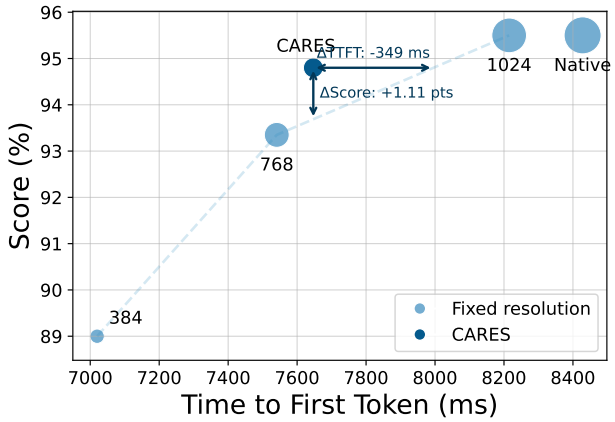


Figure 2. Accuracy vs. TTFT for DocVQA with Qwen2.5-VL-72B across native and fixed-resolution settings versus CARES. Bubble size indicates the number of pixels processed by the model.

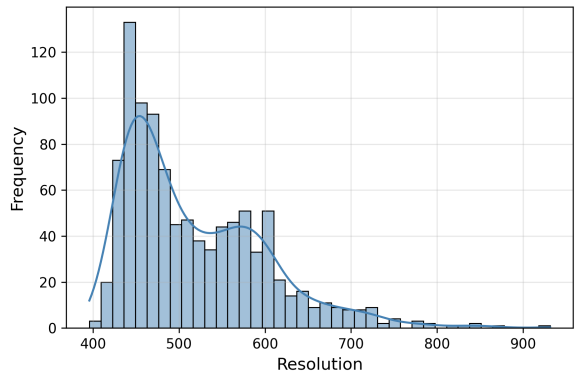


Figure 3. Histogram of the predicted resolutions $\tilde{r}$ by CARES for OCRBench.

Fig. 2 shows the accuracy–latency frontier: CARES matches near-native accuracy while using far fewer TFLOPs (e.g., 2.58 vs. 7.5) and achieving $\sim$1 second lower time-to-first-token (TTFT); static high-res inputs (e.g., 1024^2) incur substantial compute with limited TTFT gains, whereas fixed low-res (384^2) improves TTFT at the cost of quality. The query-aware routing yields a superior Pareto point.

Finally, the distribution of predicted continuous resolutions $\tilde{r}$ (Fig. 3) and the comparison in Table 3 indicate that continuous routing adapts per instance, matches or slightly improves accuracy over a discrete menu, and saves additional compute without quality loss.

4.3. Ablation study

We conduct a series of ablations to isolate the effect of key training design choices on resolution selection accuracy and downstream benchmark performance. For more information, refer to the appdx.

Resolution menu size. We compare training with binary $\mathcal{R}_d = \{384, 1024\}$ ($|\mathcal{R}_d| = 2$) vs. ternary $\mathcal{R}_d = \{384, 768, 1024\}$ ($|\mathcal{R}_d| = 3$) resolution choices. Table 2 reports both the classification accuracy and the downstream performance of Granite Vision, averaged over 5 benchmarks. As expected, the two-way classification yields higher validation accuracy in the resolution classification task compared to the more challenging three-way classification. But the ternary setup leads to better downstream benchmark performance due to the finer-grained control.

Discrete vs. continuous. CARES is trained as a discrete resolution classifier, but at inference time, it can produce either discrete predictions or a continuous estimate via interpolation. In Table 3, we compare the impact of discrete

$\lvert\mathcal{R}_d\rvert$	Resolution Accuracy	Downstream Accuracy
2	96.2%	0.76
3	67.2%	0.80

Table 2. **Binary vs. Ternary Resolution Classification.** We compare binary ($|\mathcal{R}_d| = 2$, using $\{384, 1024\}$) and ternary ($|\mathcal{R}_d| = 3$, using $\{384, 768, 1024\}$) resolution selection setups. The binary classifier achieves higher accuracy on the resolution prediction task due to its reduced complexity, while the ternary classifier improves downstream performance by enabling finer control over resolution. Reported downstream accuracy is averaged over 5 vision-language benchmarks using Granite Vision.

versus continuous inference across three VLM backbones. All scores and FLOPS deltas are averaged over five benchmarks. We find that continuous resolution selection achieves comparable accuracy to both discrete and native strategies, while significantly reducing compute. For example, with Granite-Vision 3.3-2B and InternVL3-8B, FLOPS are reduced by 63% using continuous prediction, compared to 46% with discrete. These results suggest that continuous inference allows finer control over input resolution and leads to more efficient inference without compromising performance.

Model	Resolution	Score	FLOPS
Granite-Vision 3.3-2B	Native	0.803	
	Discrete	0.801	-46%
	Continuous	0.804	-63%
InternVL3-8B	Native	0.851	
	Discrete	0.851	-46%
	Continuous	0.851	-63%
Qwen2.5-VL-72B	Native	0.851	
	Discrete	0.852	-74%
	Continuous	0.839	-80%

Table 3. **Discrete vs. Continuous Resolution Selector.** . The overall score and relative FLOPS delta per resolution strategy are averaged over 5 benchmarks. Using continuous resolutions allows finer control of the resolution, resulting in a lower resolution and computation with no drop in accuracy.

Label smoothing. To bridge the mismatch between *discrete* supervision and our *continuous* inference policy, we apply label smoothing when training the classifier over $\mathcal{R}_d$. Smoothing softens class boundaries and discourages overconfident logits, yielding better-calibrated probability distributions p that are subsequently mapped to a scalar resolution via expectation (Eq. 3). This improves the stability of the continuous selector, reduces spurious hard escalations near decision thresholds, and translates to higher downstream utility at similar—or lower—compute. Empirically, Table 4 shows that adding label smoothing improves OCRBench performance for Qwen2.5-VL-7B (0.821 vs. 0.811) while slightly *reducing* expected FLOPS, supporting its role as a simple but effective regularizer for continuous-resolution deployment.

5. Discussion and Conclusion

Inference efficiency has become a critical concern for modern vision-language systems. Most user queries do not require high-resolution inputs, yet current deployments often process all images at native or tiled resolutions by default. This leads to bloated token counts, slower response times, and higher costs. CARES addresses this challenge with a lightweight, model-agnostic approach that dynamically selects input resolution based on the query. By acting before tokenization, it provides a clean and practical lever for controlling inference cost while maintaining output quality.

Key Takeaways

- CARES reduces compute and latency across a wide range of models and benchmarks, with minimal to no loss in task accuracy.
- It requires no changes to the vision-language model and works as a plug-in component, making it easy to integrate into real-world pipelines.
- CARES adapts resolution based on the specific query, using a single low-cost pass to determine how much visual detail is needed.
- The design is compact and efficient, enabling wide applicability without adding large overhead to the main model.

Overall, CARES highlights the value of adaptive pixel allocation as a simple yet powerful strategy for efficient multimodal inference. It complements existing techniques for token-level compression and opens up a new path for practical deployment of vision-language models at scale.

Limitations

CARES depends on a frozen proxy VLM for low-resolution features; domains requiring extremely fine cues (e.g., dense OCR, medical imagery) may be under-allocated. Our supervision uses multi-resolution rollouts of a target VLM and thus inherits that model's biases and limited language support . We evaluate single-image, single-turn inputs only; multi-image, video, streaming, and joint resolution–tiling selection are left to future work. We do not study safety, robustness to adversarial prompts, or detailed cost–latency trade-offs across hardware.

Setting	Score	FLOPS
Native resolution	0.824	
CARES Without label-smoothing	0.811	-60.5%
CARES With label-smoothing	0.821	-63.8%

Table 4. **Label smoothing effect.** Evaluated on OCRBench with Qwen2.5-VL-7B. Comparison of native resolution and training with or without label smoothing. FLOPs indicate relative change.

References

Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Arif, K. H. I., Yoon, J., Nikolopoulos, D. S., Vandierendonck, H., John, D., and Ji, B. Hired: Attention-guided token dropping for efficient inference of high-resolution vision-language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 1773–1781, 2025.

Auer, C., Lysak, M., Nassar, A., Dolfi, M., Livathinos, N., Vagenas, P., Ramis, C. B., Omenetti, M., Lindlbauer, F., Dinkla, K., Mishra, L., Kim, Y., Gupta, S., de Lima, R. T., Weber, V., Morin, L., Meijer, I., Kuropiatnyk, V., and Staar, P. W. J. Docling technical report, 2024. URL https://arxiv.org/abs/2408.09869.

Bai, S., Chen, K., Liu, X., Wang, J., Ge, W., Song, S., Dang, K., Wang, P., Wang, S., Tang, J., Zhong, H., Zhu, Y., Yang, M., Li, Z., Wan, J., Wang, P., Ding, W., Fu, Z., Xu, Y., Ye, J., Zhang, X., Xie, T., Cheng, Z., Zhang, H., Yang, Z., Xu, H., and Lin, J. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.

Beyer, L., Izmailov, P., Kolesnikov, A., Caron, M., Kornblith, S., Zhai, X., Minderer, M., Tschannen, M., Alabdulmohsin, I., and Pavetic, F. Flexivit: One model for all patch sizes, 2023. URL https://arxiv.org/abs/2212.08013.

Bolya, D., Fu, C.-Y., Dai, X., Zhang, P., Feichtenhofer, C., and Hoffman, J. Token merging: Your ViT but faster. In *International Conference on Learning Representations*, 2023.

Cai, M., Yang, J., Gao, J., and Lee, Y. J. Matryoshka multimodal models. *Proceedings of the International Conference on Learning Representation*, 2025.

Dehghani, M., Mustafa, B., Djolonga, J., Heek, J., Minderer, M., Caron, M., Steiner, A. P., Puigcerver, J., Geirhos, R., Alabdulmohsin, I., Oliver, A., Padlewski, P., Gritsenko, A. A., Lucic, M., and Houlsby, N. Patch n' pack: Navit, a vision transformer for any aspect ratio and resolution. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Hu, J., Mao, J., Liu, Z., Xia, Z., Jia, P., and Lang, X. Tokenflex: Unified vlm training for flexible visual tokens inference, 2025.

Jiang, D., He, X., Zeng, H., Wei, C., Ku, M., Liu, Q., and Chen, W. Mantis: Interleaved multi-image instruction tuning. *arXiv preprint arXiv:2405.01483*, 2024.

Kembhavi, A., Salvato, M., Kolve, E., Seo, M., Hajishirzi, H., and Farhadi, A. A diagram is worth a dozen images, 2016. URL https://arxiv.org/abs/1603.07396.

Kimhi, M., Koifman, E., Rivlin, E., Schwartz, E., and Baskin, C. Waveclip: Wavelet tokenization for adaptive-resolution clip, 2025. URL https://arxiv.org/abs/2509.21153.

Li, B., Ge, Y., Ge, Y., Wang, G., Wang, R., Zhang, R., and Shan, Y. Seed-bench-2: Benchmarking multimodal large language models. *arXiv preprint arXiv:2311.17092*, 2023.

Li, K. Y., Goyal, S., Semedo, J. D., and Kolter, J. Z. Inference optimal vlms need only one visual token but larger models, 2024.

Liang, Y., Ge, C., Tong, Z., Song, Y., Wang, J., and Xie, P. Not all patches are what you need: Expediting vision transformers via token reorganizations, 2022.

Lin, Z., Lin, M., Lin, L., and Ji, R. Boosting multimodal large language models with visual tokens withdrawal for rapid inference. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 5334–5342, 2025.

Liu, H., Li, C., Li, Y., Li, B., Zhang, Y., Shen, S., and Lee, Y. J. Llava-next: Improved reasoning, ocr, and world knowledge, January 2024a. URL https://llava-vl.github.io/blog/2024-01-30-llava-next/.

Liu, Y., Li, Z., Huang, M., Yang, B., Yu, W., Li, C., Yin, X.-C., Liu, C.-L., Jin, L., and Bai, X. Ocrbench: on the hidden mystery of ocr in large multimodal models. *Science China Information Sciences*, 67(12), December 2024b. ISSN 1869-1919. doi: 10.1007/s11432-024-4235-6. URL http://dx.doi.org/10.1007/s11432-024-4235-6.

Marafioti, A., Zohar, O., Farré, M., Noyan, M., Bakouch, E., Cuenca, P., Zakka, C., Allal, L. B., Lozhkov, A., Tazi, N., Srivastav, V., Lochner, J., Larcher, H., Morlon, M., Tunstall, L., von Werra, L., and Wolf, T. Smolvlm: Redefining small and efficient multimodal models. *arXiv preprint arXiv:2504.05299*, 2025.

Masry, A., Long, D. X., Tan, J. Q., Joty, S., and Hoque, E. Chartqa: A benchmark for question answering about charts with visual and logical reasoning. *arXiv preprint arXiv:2203.10244*, 2022.

Mathew, M., Karatzas, D., and Jawahar, C. Docvqa: A dataset for vqa on document images. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pp. 2200–2209, 2021.

Orgad, H., Toker, M., Gekhman, Z., Reichart, R., Szpektor, I., Kotek, H., and Belinkov, Y. Llms know more than they show: On the intrinsic representation of llm hallucinations. *arXiv preprint arXiv:2410.02707*, 2024.

Rao, Y., Zhao, W., Liu, B., Lu, J., Zhou, J., and Hsieh, C.-J. Dynamicvit: Efficient vision transformers with dynamic token sparsification. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Singh, A., Natarajan, V., Shah, M., Jiang, Y., Chen, X., Batra, D., Parikh, D., and Rohrbach, M. Towards vqa models that can read. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8317–8326, 2019.

Team, G. V., Karlinsky, L., Arbelle, A., Daniels, A., Nassar, A., Alfassi, A., Wu, B., Schwartz, E., Joshi, D., Kondic, J., et al. Granite vision: a lightweight, open-source multimodal model for enterprise intelligence. *arXiv preprint arXiv:2502.09927*, 2025.

Wang, P., Bai, S., Tan, S., Wang, S., Fan, Z., Bai, J., Chen, K., Liu, X., Wang, J., Ge, W., Fan, Y., Dang, K., Du, M., Ren, X., Men, R., Liu, D., Zhou, C., Zhou, J., and Lin, J. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024.

Xing, L., Huang, Q., Dong, X., Lu, J., Zhang, P., Zang, Y., Cao, Y., He, C., Wang, J., Wu, F., and Lin, D. Pyramiddrop: Accelerating your large vision-language models via pyramid visual redundancy reduction, 2025.

Zhang, J., Khayatkhoei, M., Chhikara, P., and Ilievski, F. MLLMs know where to look: Training-free perception of small visual details with multimodal LLMs. In *The Thirteenth International Conference on Learning Representations*, 2025a. URL https://arxiv.org/abs/2502.17422.

Zhang, K., Li, B., Zhang, P., Pu, F., Cahyono, J. A., Hu, K., Liu, S., Zhang, Y., Yang, J., Li, C., and Liu, Z. Lmms-eval: Reality check on the evaluation of large multimodal models, 2024. URL https://arxiv.org/abs/2407.12772.

Zhang, S., Fang, Q., Yang, Z., and Feng, Y. Llava-mini: Efficient image and video large multimodal models with one vision token. In *International Conference on Learning Representations (ICLR)*, 2025b.

Zhang, Y., Fan, C.-K., Ma, J., Zheng, W., Huang, T., Cheng, K., Gudovskiy, D., Okuno, T., Nakata, Y., Keutzer, K., et al. Sparsevlm: Visual token sparsification for efficient vision-language model inference. In *International Conference on Machine Learning*, 2025c.

Zhao, W., Han, Y., Tang, J., Li, Z., Song, Y., Wang, K., Wang, Z., and You, Y. A stitch in time saves nine: Small vlm is a precise guidance for accelerating large vlms. *arXiv preprint arXiv:2412.03324*, 2024.

Zhu, J., Wang, W., Chen, Z., Liu, Z., Ye, S., Gu, L., Tian, H., Duan, Y., Su, W., Shao, J., et al. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*, 2025.

Zhu, M., Han, K., Wu, E., Zhang, Q., Nie, Y., Lan, Z., and Wang, Y. Dynamic resolution network. In *Advances in Neural Information Processing Systems (NeurIPS)*, Red Hook, NY, USA, 2021.

Table 5. **Time to First Token (TTFT, ms)** measured on H100 with batch size 1, averaged over 100 DocVQA examples. *Native* denotes the model's default input pipeline. CARES reduces TTFT substantially compared to native and fixed high-resolution settings while preserving strong downstream accuracy.

Model	Native	1024^2	768^2	384^2	CARES
Qwen2.5-VL-7B	435.7	433.8	220	76.12	270.1
Granite-Vision 3.3-2B	228.6	201.3	140.1	96.1	108.9

Additional Analysis and Results

This appendix provides additional qualitative and quantitative analysis of CARES.

.1. Label generation pipeline

Figure 4 illustrates the supervision pipeline used to train CARES. For each image-query pair, we evaluate a pretrained VLM at several fixed resolutions and compare its prediction to the ground-truth answer. The smallest resolution whose score satisfies the sufficiency criterion is used as the training target. This process transforms downstream task behavior into per-example supervision for resolution selection, enabling CARES to learn when higher visual detail is genuinely needed.

.2. Time-to-first-token analysis

[h]

Table 5 reports time-to-first-token (TTFT) on DocVQA for representative downstream VLMs. The results mirror the FLOPS trends in the main paper: lower resolutions substantially reduce latency, while CARES achieves a favorable trade-off by approaching the latency of low-resolution inference without incurring the accuracy loss of always using a small input. In particular, CARES significantly improves TTFT relative to native or fixed high-resolution processing, confirming that adaptive resolution selection translates into practical end-to-end inference gains.

.3. Feature extractor.

We ablate several frozen backbones used for feature extraction in CARES, varying both model type and layer depth. As shown in Table 7, both Qwen2.5-3B and SmolVLM achieve higher accuracy when using intermediate-layer features, outperforming their own final-layer variants. This aligns with prior findings suggesting that intermediate representations in VLMs often encode richer signals than final outputs.

Qwen2.5-3B and SmolVLM both process the image and query jointly within a unified transformer, in contrast to SigLIP v2's dual-encoder architecture, where vision and language are encoded separately. For SigLIP, we follow the original design by pooling the outputs of each tower, concatenating them, and passing the result to the classifier head. While this setup is architecturally simple, it underperforms joint encoding by a considerable margin (56.1% accuracy), and it requires more parameters than the lightweight SmolVLM.

Although Qwen2.5-3B achieves the best overall accuracy, we adopt SmolVLM as our default backbone due to its favorable trade-off between performance, size, and efficiency, making it a more practical choice for real-world pre-processing.

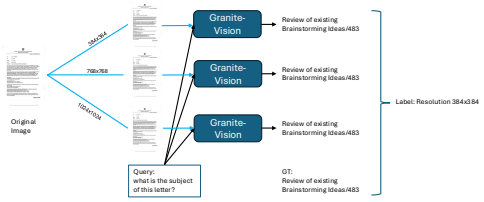
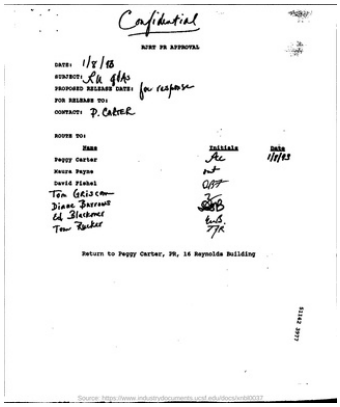
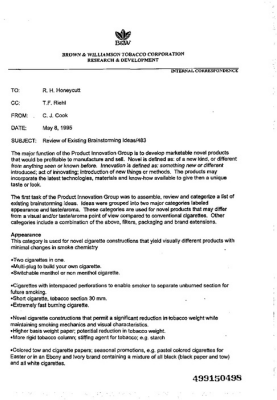


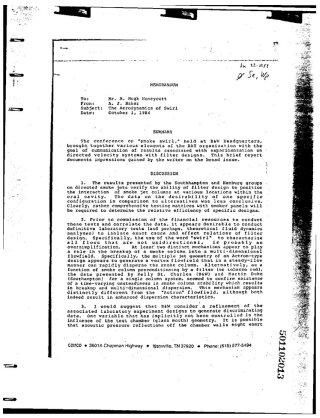
Figure 4. **Label generation pipeline for training CARES.** For each image-query pair, we evaluate a pretrained VLM at multiple fixed resolutions and assign the smallest resolution that satisfies the sufficiency criterion as the supervision label.

.4. Predicted resolution distributions

Figure 5 shows the distribution of continuous resolutions predicted by CARES across different benchmarks. The histograms highlight that the selector adapts its behavior to the underlying task: Ai2D is dominated by lower-resolution predictions, suggesting that many diagram-understanding questions require only coarse visual information; SeedBench-2 shifts toward higher resolutions, reflecting the need for finer-grained visual recognition; and DocVQA and OCR-Bench exhibit broader distributions, indicating a mixture of easy and detail-sensitive examples. This behavior is consistent with the intended design of CARES, which escalates resolution only when the image-query pair appears to demand additional visual detail.





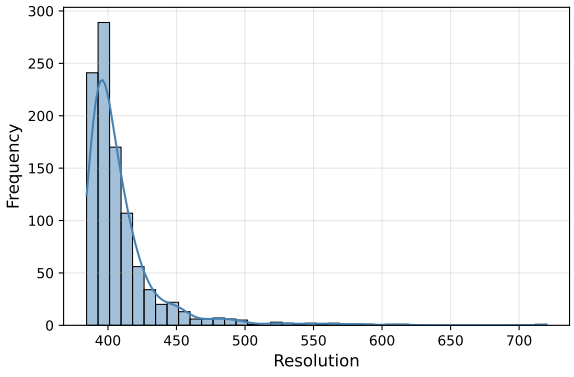


Query	what is the contact person name mentioned in letter?	Who is in cc in this letter?	One variable that has implicitly not been controlled?
GT	P. Carter	T.F. Riehl	influence of the test chamber (glass mouth) geometry.
Resp@384	P. Carter	T.F. Rosel	concentration of the final product
ANLS	1.0	0.7	0.0
Resp@768		T.F. Riehl	the influence of the test chamber (i.e. ash seath) geometry on the flow
ANLS		1.0	0.65
Resp@1024			the influence of the test chamber (glass mouth) geometry.
ANLS			0.93
Sufficient Resolution	384	768	1024

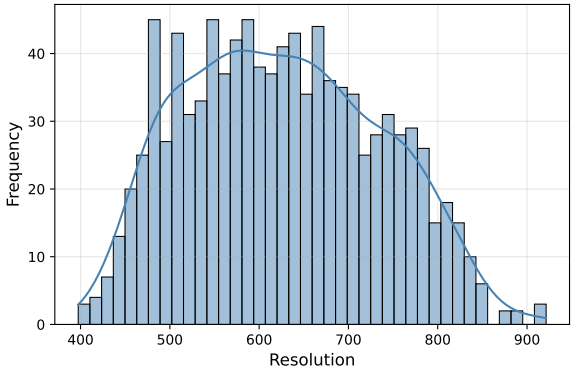
Table 6. Data generation pipeline for training CARES. We process each input through a pretrained VLM (Granite-Vision) at three fixed resolutions and select the smallest resolution that produces a sufficient answer quality according to the ANLS metric.

Model	Layer	Params	Accuracy
SigLIP v2	-	0.8B	56.1%
SmolVLM	Mid	0.35B	63.3%
SmolVLM	Last	0.5B	62.3%
Qwen2.5–3B	Mid	2.3B	67.2%
Qwen2.5–3B	Last	3.75B	66.2%

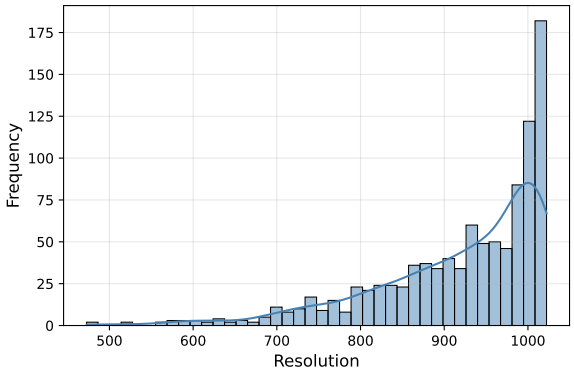
Table 7. **Feature extractor.** Validation accuracy and parameter count for different frozen feature extractors used in CARES. All models are trained to classify among three resolution choices. For SmolVLM and Qwen2.5-3B, we compare features extracted from intermediate (MID) and final (LAST) layers. For SigLIP, the pooled outputs from the vision and language towers are concatenated and passed to the classifier head. Qwen2.5-3B provides the best performance, while SmolVLM offers strong accuracy with minimal size.



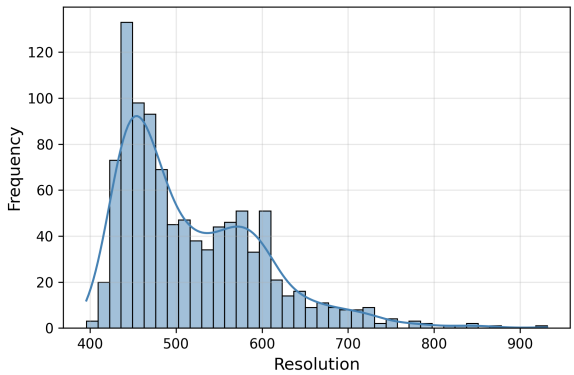
(a) Ai2D.



(b) DocVQA.



(c) SeedBench-2.



(d) OCRBench.

Figure 5. **Histograms of the predicted continuous resolutions $\tilde{r}$ by CARES.** CARES routes many Ai2D examples to lower resolutions, while SeedBench-2 shifts toward higher resolutions. DocVQA and OCRBench show broader distributions, reflecting their mixture of coarse and fine-grained queries, including dense text and complex layouts.