FOCUS: Internal MLLM Representations for Efficient Fine-Grained Visual Question Answering

Liangyu Zhong[*1,3], **Fabio Rosenthal**[*2,4], **Joachim Sicking**[3], **Fabian Hüger**[3], **Thorsten Bagdonat**[4], **Hanno Gottschalk**[1], **Leo Schwinn**[2]

[1] Technical University of Berlin, [2] Technical University of Munich,
[3] CARIAD SE, [4] Volkswagen AG

Project page: https://focus-mllm-vqa.github.io

Abstract

While Multimodal Large Language Models (MLLMs) offer strong perception and reasoning capabilities for image-text input, Visual Question Answering (VQA) focusing on small image details still remains a challenge. Although visual cropping techniques seem promising, recent approaches have several limitations: the need for task-specific fine-tuning, low efficiency due to uninformed exhaustive search, or incompatibility with efficient attention implementations. We address these shortcomings by proposing a training-free visual cropping method, dubbed FOCUS, that leverages MLLM-internal representations to guide the search for the most relevant image region. This is accomplished in four steps: first, we identify the target object(s) in the VQA prompt; second, we compute an object relevance map using the key-value (KV) cache; third, we propose and rank relevant image regions based on the map; and finally, we perform the fine-grained VQA task using the top-ranked region. As a result of this informed search strategy, FOCUS achieves strong performance across four fine-grained VQA datasets and three types of MLLMs. It outperforms three popular visual cropping methods in both accuracy and efficiency, and matches the best-performing baseline, ZoomEye, while requiring $3-6.5\times$ less compute.

1 Introduction

Multimodal Large Language Models (MLLMs) exhibit compelling cross-modal perception and reasoning capabilities, particularly on image-text data [8, 9, 22]. However, standard MLLM architectures are not well suited to perceive and reason about small visual details in high-resolution images [37, 42] as they typically downscale their inputs, leading to a loss of information. Examples of these so-called *global-view* MLLMs include Instruct-BLIP [9] and LLaVA-1.5 [24], which only support low-resolution inputs of 224×224 px and 336×336 px, respectively. As a consequence, global-view MLLMs perform poorly on Visual Question Answering (VQA) tasks involving small-scale visual details [38].

Recent MLLM architectures such as LLaVA-OneVision [22] and Gemma-3 [19] address this limitation by processing both a downsampled global view and local crops extracted from the original image. This dual-view strategy enables them to handle high-resolution inputs with reduced information loss compared to global-view MLLMs. However, despite having access to fine-grained visual details from all local crops, these so-called *global-local-view* MLLMs struggle to identify the few visual tokens that are relevant for fine-grained VQA amid the large number of local crop tokens. While these

[*] Equal contribution

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

global-local-view architectures outperform global-view MLLMs, their effectiveness on fine-grained VQA tasks still remains limited [37].

An orthogonal research direction to address the limitations of MLLMs in capturing fine details in high-resolution images are visual cropping approaches [31, 37, 38, 42], which seek to pass only relevant image regions to the MLLM. However, popular visual cropping techniques like SEAL [38], DC^2 [37], ZoomEye [31], and ViCrop [42] suffer from one or more of the following limitations: (1) reliance on task-specific fine-tuning of MLLMs for fine-grained VQA, (2) use of inefficient, exhaustive search algorithms, and (3) incompatibility with efficient attention implementations such as FlashAttention [10] (see Tab. 1 and Fig. 6 for a visual comparison of the methods). We propose a visual cropping method, termed **F**ine-grained visual **O**bject **C**ropping **U**sing cached token **S**imilarity (FOCUS), that addresses these issues as is outlined in the following.

Table 1: **Comparison of visual cropping methods w.r.t. desirable properties.** Unlike previously suggested methods, FOCUS is training-free, algorithmically efficient in search, and compatible with efficient attention implementations.

Method	Training-free	Efficient search algo.	Compatible w/ efficient attention
SEAL [38]	✗	✗	✓
DC^2 [37]	✓	✗	✓
ZoomEye [31]	✓	✗	✓
ViCrop [42]	✓	✓	✗
FOCUS (Ours)	✓	✓	✓

To tackle limitation (1), FOCUS leverages the internal representations of MLLMs, specifically their key-value (KV) caches [27], to localize question-relevant image regions in a training-free manner—unlike the SEAL [38] technique. Moreover, to mitigate limitation (2), our method includes textual clues to enable object-aware localization without exhaustive cropping of the image, thereby improving the algorithmic efficiency—different from DC^2 [37] and ZoomEye [31]. To overcome limitation (3), FOCUS utilizes the cached value features readily available during inference, making it natively compatible with efficient attention implementations [10]—unlike ViCrop [42] that depends on full attention weights. Specifically, FOCUS combines these components as follows: for each VQA question, we first identify the target object(s) in the question prompt. Second, we construct an object relevance map using cosine similarity between the cached text tokens of the target object(s) and the cached image tokens, and then propose relevant regions based on this map. Third, we rank the proposed image regions based on the existence confidence of the target object in each region. Finally, we perform VQA solely based on the image region with the highest confidence. Note that FOCUS is compatible with both global- and global-local-view MLLMs.

We evaluate FOCUS on the fine-grained VQA datasets V*Bench [38], HRBench-4K [37], HRBench-8K [37] and MME-RealWorld-Lite [43]. Across the first three datasets, our method achieves on average 42% higher accuracy over the vanilla MLLMs when using LLaVA-1.5 and 17% when using LLaVA-OneVision, while improving LLaVA-OneVision by 6% on the multi-domain MME-RealWorld-Lite dataset. Moreover, FOCUS achieves comparable or superior performance w.r.t. the state-of-the-art baseline ZoomEye [31] while being $3.5-4.5\times$ more efficient with LLaVA-1.5 and $3-6.5\times$ more efficient with LLaVA-OneVision.

Our key contributions are as follows: First, we propose FOCUS, a training-free visual cropping method for MLLM-based fine-grained VQA that identifies relevant image regions using internal representations of the MLLM. Second, we provide extensive empirical evidence for FOCUS's favorable accuracy-efficiency trade-offs compared to previous visual cropping methods for fine-grained VQA. Third, we conduct an ablation study that provides insights on how FOCUS leverages MLLM-internal knowledge for efficient visual cropping.

2 Related work

VQA involves answering a question based on the visual content of an image. While various types of machine learning models can be applied to this task, MLLMs have become the de facto standard due to their strong cross-modal reasoning capabilities [15, 24]. Here, we focus on the multiple-choice variant of VQA, where the MLLM is expected to select the correct answer from a fixed set of options. In the following paragraphs, we describe the datasets commonly used to evaluate MLLM performance on *fine-grained VQA* tasks, i.e., tasks that require focus on visual details. Further, we provide a technical overview of recent visual cropping methods.

Fine-grained VQA datasets In common VQA datasets such as Text-VQA [33] and Ref-COCO [20], the relevant objects are typically prominent within the image. In contrast, fine-grained VQA tasks focus on much smaller visual targets. The V*Bench dataset exemplifies this, containing significantly smaller question-relevant objects compared to the aforementioned datasets (see Fig. 1). Additional fine-grained VQA datasets include HRBench-4K [37], HRBench-8K [37], and MME-RealWorld-Lite [43]. Among these, V*Bench is the only dataset that provides Ground Truth (GT) annotations for question-relevant objects.

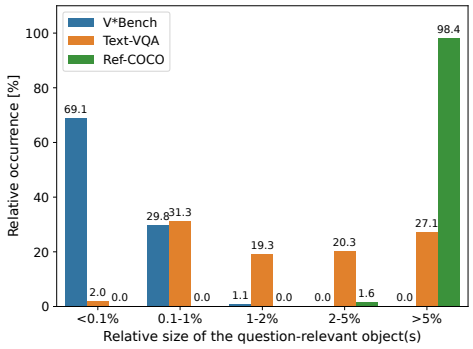

Figure 1: **Many VQA datasets focus on large instead of tiny objects.** This figure shows the relative area of question-relevant objects w.r.t. the image. V*Bench contains various tiny VQA-relevant objects.

Visual cropping for fine-grained VQA In this emerging area of research, different visual cropping methods [31, 37, 38, 42] have been proposed to improve MLLM performance on fine-grained VQA tasks (see Tab. 1). SEAL [38] employs a dual-MLLM setup: one MLLM for visual search and another one for the actual VQA task. The visual search model with additional decoders is fine-tuned to predict object heatmaps and coordinates. It performs a top-down hierarchical search, generating contextual cues to iteratively locate the target object based on confidence scores. This approach requires task-specific fine-tuning and is rather inefficient due to its complex, multi-module design and recursive search process. In contrast, DC^2 [37] constructs a hierarchical image region tree from the global view down to regions, matching the base resolution of the vision encoder. Each region is captioned using the MLLM and those containing the target object are merged for actual VQA. While training-free, DC^2 is inefficient due to the extensive tree traversal and costly captioning process. Similar to DC^2, ZoomEye [31] employs a hierarchical tree search, but instead of captioning, it predicts a confidence score for each image region. This score is computed through a complex mechanism involving three sequential MLLM forward passes, each using a different question prompt. As a result, the process cannot be simplified or shared across regions, making the method inefficient due to both the hierarchical search and the high cost of confidence estimation. ViCrop [42] is an efficient, training-free visual cropping method that avoids hierarchical search by directly computing a question-guided heatmap to localize the target object. However, its best-performing variants depend on Q-K attention weights and answer gradients, making the method incompatible with efficient attention implementations such as FlashAttention [10]. Unlike these methods, our method is a training-free visual cropping approach without additional modules besides the MLLM. Rather than relying on recursive search, captioning, or Q-K attention weights, we employ an informed search guided by internal representations of the MLLM that is compatible with efficient attention implementations. This design enables our method to achieve significantly higher efficiency without sacrificing accuracy.

3 FOCUS: Fine-grained visual object cropping using cached token similarity

We first provide relevant background information for our method in Sec. 3.1. Next, we describe in detail how our method proposes relevant image regions based on the KV-cache in Sec. 3.2. Finally, we explain how these image regions are used for fine-grained VQA tasks in Sec. 3.3. We provide a visualization of `FOCUS` in Fig. 2.

3.1 Background

MLLMs typically comprise three core components: a vision encoder, a modality projector, and a Large Language Model (LLM). The LLM receives an input sequence of tokens $\mathrm{X} = (x_1, \ldots, x_n)$ and predicts the next token x_{n+1} auto-regressively [28]. This sequence can be viewed as a concatenation of visual tokens $\mathrm{X}_{\text{vis}} = (x_1, \ldots, x_{a^2})$ and instruction textual tokens $\mathrm{X}_{\text{text}} = (x_{a^2+1}, \ldots, x_n)$, where a is the grid size of the vision encoder (for simplicity, we ignore system prompt tokens here). In many models, the visual tokens from the vision encoder are extracted from a downscaled global view of the image and projected into the LLM's feature space via the modality projector. The resizing

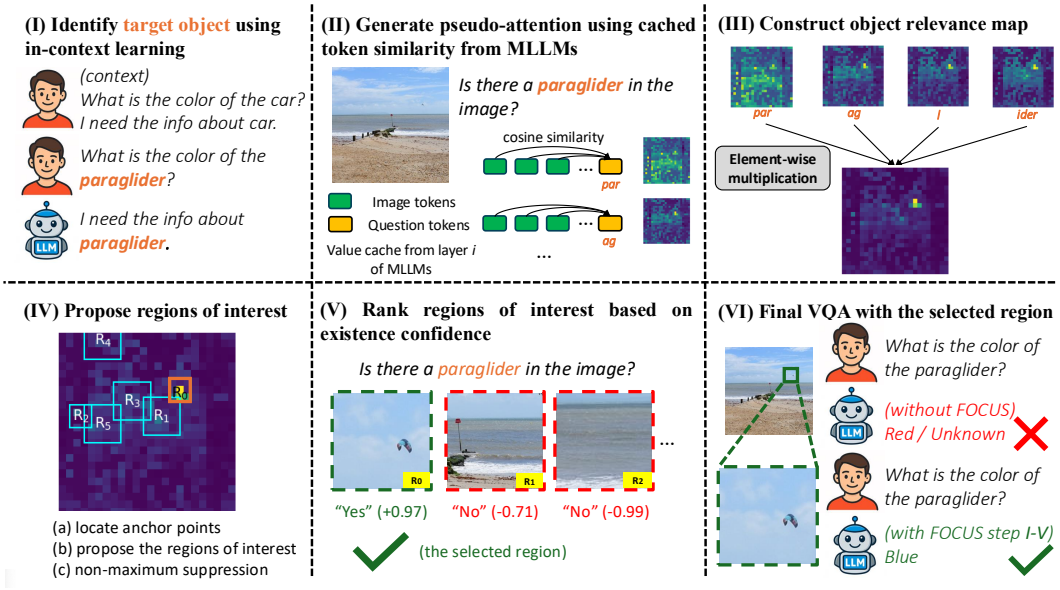

Figure 2: **Overview of** `FOCUS`. The method identifies the target objects mentioned in the question **(I)** and constructs their object relevance maps using cosine similarity between cached tokens **(II + III)**. Then, it proposes regions of interest and ranks those by the existence confidence of the target object in each region **(IV + V)**. Finally, the selected region is used to perform VQA **(VI)**.

aligns the input image with the base resolution expected by the vision encoder, e.g., 336×336 for CLIP [29] and 384×384 for SigLIP [41]. While this regime is effective in many scenarios, it tends to fail on fine-grained VQA tasks [37, 42]. To address this limitation, many global-local-view MLLMs [16, 22, 23, 40] partition the original (unresized) image into b local crops in addition to the resized global view. These crops are encoded into additional visual tokens, extending the visual input to $\mathrm{X}_{\mathrm{vis}} = (x_1, \ldots, x_{a^2 \cdot (b+1)})$. This results in a rapid growth of computational cost due to the quadratic complexity of self-attention in transformer layers [36]. We aim to improve the performance of both global-view and global-local-view MLLMs on fine-grained VQA tasks by constructing a map that efficiently localizes the image regions most relevant to a target object mentioned in a question. This map is referred to as the `object relevance map`. Once the relevant regions are identified, we restrict processing to these areas for the final VQA prediction, thereby improving accuracy and computational efficiency.

3.2 Constructing object relevance maps from cached token similarity

Localizing objects in images remains a challenge for MLLMs [6]. While many models are fine-tuned on predicting bounding boxes [20], they often fail when prompted directly for the location of small objects, frequently producing hallucinated or imprecise coordinates [6]. Instead of relying on explicit prompting, we propose to localize target objects by leveraging value features of cached tokens of MLLMs. Recent work [25] shows that visual tokens in the sequence largely preserve spatial correspondence to their originating image regions across transformer layers [36]. In fine-grained VQA datasets [37, 38, 43], questions typically involve one or more specific objects in the image, which we refer to as the target object(s). Since the text tokens corresponding to these targets co-exist alongside visual tokens in the token sequence, we estimate an object relevance map by computing the cosine similarity between the text and visual tokens.

To construct this object relevance map, we first identify the text tokens corresponding to the target object(s) as shown in Fig. 2 (**I**). Inspired by ZoomEye [31], we use the in-context learning (ICL) capability of MLLMs [12] to extract the target object(s) by providing a few examples in the prompt. This might extract one or multiple target object(s) from the question. For each target object, we apply a generic prompt template "`Is there a {target object} in the image?`" to query the MLLM alongside the image. Due to the tokenization [36] of the MLLM, the target object can result in a sequence of target text tokens $\hat{X}_{\mathrm{tgt}} = (\hat{x}_0, \ldots, \hat{x}_{\mathrm{i}}, \ldots, \hat{x}_{\mathrm{s}})$. We calculate the cosine similarity between

the target tokens and the visual tokens in the sequence to construct the object relevance map. While one might consider using standard query-key (Q-K) attention weights [36] for this purpose, many recent MLLMs employ efficient attention implementations such as FlashAttention [10], which do not generate Q-K attention weights explicitly. We address this issue by using value features preserved in the KV-cache mechanism [27], which is commonly used to accelerate autoregressive inference by storing intermediate representations. Leveraging this, we propose an alternative value-value (V-V) pseudo-attention approach as shown in Fig. 2 (**II**). For each target token $\hat{x}_i^l, i \in \{0, \ldots, s\}$ in the l-th layer, we compute a pseudo-attention map $\mathbf{A}_i^l \in \mathbb{R}^{a \times a}$ by measuring its cosine similarity (cos) with the visual tokens $(x_1^l, \ldots, x_a^l)$, where value features are available via the KV-cache:

$$\mathbf{A}_i^l = \cos(\hat{x}_i^l, x_j^l), \quad \text{for } j = 1, \ldots, a^2 \tag{1}$$

and reshape $\mathbf{A}_i^l$ into an $a \times a$ matrix. Alternatively, one can also compute $\mathbf{A}_i^l$ using key features, see Sec. 4.3 for details. In preliminary experiments, we empirically find that the pseudo-attention map $\mathbf{A}_i^l$ from a single layer might be noisy. Therefore, we aggregate maps from l-th layer to L-th layer using attention rollout [1], incorporating residual connections to better preserve information flow as follows:

$$\mathbf{A}_i = \sum_{k=l}^{L} (\mathbf{A}_i^l + \mathbf{I})/2 \,, \tag{2}$$

where $\mathbf{I}$ denotes the identity matrix. We aggregate the pseudo-attention maps $\mathbf{A}_i$ for each individual target token $\hat{x}_i$ by element-wise multiplication to capture consensus as shown in Fig. 2 **(III)**:

$$\mathbf{A} = \mathbf{A}_0 \odot \mathbf{A}_1 \odot \cdots \odot \mathbf{A}_s \,. \tag{3}$$

This operation allows only regions that are consistently highlighted across all target tokens to remain prominent. For example, the token `red` may highlight many red objects, but when combined with `car`, only regions corresponding to red cars will be emphasized. We refer to $\mathbf{A}$ as the object relevance map corresponding to the target object. A normalization of $\mathbf{A}$ is performed after every matrix addition and multiplication to ensure numerical stability.

For global-local-view MLLMs, instead of calculating cosine similarity between the target tokens and the visual tokens from the global view, we use visual tokens extracted from the local crops. We empirically find that these local tokens can better capture fine-grained details. We compute the pseudo-attention map $\mathbf{A}_i^l$ using local visual tokens:

$$\mathbf{A}_i^l = \cos(\hat{x}_i^l, x_j^l), \quad \text{for } j = a^2 + 1, \ldots, a^2 \cdot (b + 1) \tag{4}$$

and reshape $\mathbf{A}_i^l$ into a $h \times w$ matrix, where h and w denote the spatial dimensions of the local visual tokens arranged to closely preserve the original image's aspect ratio, so that $h \cdot w = a^2 \cdot b$. To reduce noise and enhance spatial coherence, we empirically apply a Gaussian filter to $\mathbf{A}$, followed by downsampling, yielding a cleaner object relevance map.

3.3 Ranking of proposed regions of interest (ROIs) for fine-grained VQA

Given an object relevance map, we define in the following a relevance score for each of its elements. As shown in Fig. 2 (**IV**), once the object relevance map is obtained, we extract the locations of the top-k highest scores as anchor points on $\mathbf{A}$, which represent regions likely containing the target object. To ensure sufficient spatial coverage, we select a relatively large k and enforce a minimum distance s_{dist} between anchor points. Anchor points that are too close are discarded. Then, we generate an initial symmetric ROI of minimal size s_{min} per anchor point.

Each initial ROI is then expanded up to the maximal size s_{max}, stopping when the average relevance score within the ROI falls below a predefined threshold. We rank the resulting ROIs based on the relevance score at their respective anchor points. To eliminate redundancy, we apply non-maximum-suppression [14, 17], using a low threshold to promote diversity among selected regions. This encourages broader spatial coverage.

The resulting object relevance map can be noisy due to spurious high-activation tokens [11] that do not correspond to the target object. As a result, a ROI with a high relevance score may not actually contain the target object. Therefore, we verify whether the ROI contains the target (see App. A.5

for details). Similar to ZoomEye [31], the ROI is provided to the MLLM together with an existence prompt, and an existence confidence is computed from the model's response, as shown in Fig. 2 (**V**). Then, we rerank the top n_{steps} ROIs according to their existence confidence, with the number of steps n_{steps} controlling the Forward Pass (FP) budget.

Final VQA inference In the previous paragraphs, we demonstrated how to obtain the most relevant ROIs for each target object. These ROIs are now passed to the MLLM for the final VQA prediction. We follow the inference strategy from ZoomEye [31], which categorizes questions in fine-grained VQA datasets into two types. Type-1 questions involve single-object instances and type-2 questions concern multiple instances of an object type. These categories can be automatically inferred using ICL or keyword-based heuristics, without requiring prior knowledge of the question. For type-1 questions, we select the ROI with the highest confidence score for each target object. If the question involves multiple target objects, we combine the regions covering all relevant objects for the final VQA. For type-2 questions, we iterate over all proposed ROIs and select those with confidence scores higher than a threshold. This step is not constrained by the n_{steps}. In the case of global-local-view MLLMs, this process slightly differs. Instead of combining the visual crops into a single area, we utilize the model's text-image-interleaved capabilities [2, 22, 23]. Specifically, we provide one global image view—where all target objects are visually highlighted—alongside the top-confidence ROIs for each target object. Please check App. A.5 for details.

4 Experiments

We first describe the implementation details of our experiments in Sec. 4.1 and provide the results on fine-grained VQA datasets in Sec. 4.2. Further, we conduct an ablation study and a hyperparameter sensitivity analysis in Sec. 4.3 and provide additional results (i) on VQA datasets with larger objects and (ii) with Qwen-2.5-VL [4] in Sec. 4.4. Finally, we show some qualitative examples and discuss the limitations of FOCUS in Sec. 4.5 and Sec. 4.6, respectively.

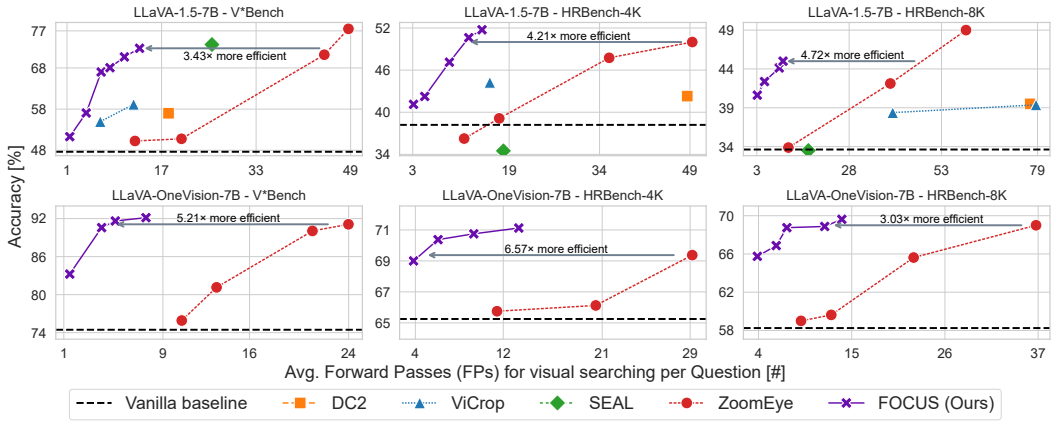

Figure 3: **FOCUS is at the Pareto front on fine-grained VQA benchmarks.** Given the same computation budget, FOCUS (purple crosses) significantly outperforms other visual cropping methods, on three different datasets and for two model architectures. It achieves $3-6.5\times$ higher efficiency than the best-performing baseline ZoomEye. Note that we show only a limited set of data points for each method to ensure a clear visualization. The full results are available in App. C.1.

4.1 Implementation details

Following prior work [31, 37, 38], we evaluate FOCUS on several fine-grained VQA benchmarks: V*Bench [38], HRBench [37], and MME-RealWorld-Lite [43]. We report accuracy on multiple-choice questions as the primary performance metric. Another critical consideration is the trade-off between performance and efficiency, as inference with MLLMs can be computationally expensive. In the case of multiple-choice VQA, inference speed is largely determined by the sequence length during the prefill phase [32, 36]. Since the sequence length remains relatively consistent across different searches, we quantify efficiency using the number of Forward Passes (FPs) required for the visual search. We provide details of used hardware and on the calculation of all metrics in App. A.1 and

App. A.2. We evaluate our method using two types of MLLMs: LLaVA-1.5-7B [24], a global-view MLLM, and LLaVA-OneVision-7B [22], a global-local-view MLLM. As described in Sec. 3.2, we utilize representations from multiple layers to compute the object relevance map. For LLaVA-1.5, we use representations from the 14^{th} to the 32^{nd} layer ($l = 14$, $L = 32$), and for LLaVA-OneVision from the 14^{th} to the 28^{th} layer ($l = 14$, $L = 28$). To evaluate performance under varying computational budgets, we adjust only the number of steps, setting $n_{\text{steps}} \in \{1, 2, 3, 4, 6, 8\}$. We describe the configuration of the remaining hyperparameters of FOCUS in App. A.3; for an analysis of FOCUS's hyperparameter sensitivity, see Sec. 4.3.

For ZoomEye, we vary the number of crops per layer and the cropping depth. For ViCrop, we report results from its best-performing variants, i.e., `rel-attn` and `attn-grad`. For DC^2, we determine the FPs via the base resolution of the vision encoder. For SEAL, we evaluate the publicly available pre-trained model without modifications. Additional implementation details are provided in App. A.4. Note that SEAL and ZoomEye use a different inference scheme on V*Bench compared to the open-ended generation approach of FOCUS. With this alternative scheme, we observe significantly improved performance with ZoomEye on V*Bench. Further, we observe a notable gap between our ZoomEye results and those reported in the original paper. Detailed results are provided in App. C.5.

4.2 Results on fine-grained VQA datasets

We conduct experiments with LLaVA-1.5 on V*Bench, HRBench-4K, and HRBench-8K (see Fig. 3). Overall, FOCUS outperforms the four other visual cropping methods on a relatively small computational budget of fewer than 17 FPs. FOCUS achieves an accuracy of 72.77% on V*Bench, 51.75% on HRBench-4K, and 45.00% on HRBench-8K. SEAL achieves a slightly higher accuracy than FOCUS on V*Bench, but at the cost of significantly lower efficiency due to its multi-module design and recursive visual search. On high-resolution datasets like HRBench, its performance is only on par with the vanilla MLLM. We suspect this decline stems from SEAL's training data, which is specifically optimized for resolutions below 2K. ZoomEye achieves the highest accuracy on V*Bench (77.48%), but only with an extremely deep tree search, resulting in substantial computational overhead. At our top accuracy of 72.77%, FOCUS is $3.43\times$ more efficient than ZoomEye. On HRBench-4K, our method not only surpasses ZoomEye with a better top accuracy but also uses $4.39\times$ fewer FPs. On HRBench-8K, while ZoomEye attains a higher top accuracy of 49.00%, FOCUS achieves 44.88% with $4.72\times$ greater efficiency. This performance gap on HRBench-8K is likely due to the limitations of the 24×24 object relevance map produced by LLaVA-1.5. Smaller objects often remain undetected, limiting our model's ability to improve accuracy—even when increasing n_{steps} to allocate more computation.

With LLaVA-OneVision, we conduct experiments on the previously introduced datasets (see Fig. 3) and additionally evaluate on MME-RealWorld-Lite (see Tab. 2). We compare FOCUS only to ZoomEye, as SEAL does not provide any models based on LLaVA-OneVision. Moreover, neither ViCrop nor DC^2 supports evaluation with LLaVA-OneVision. FOCUS significantly benefits from the higher-resolution object relevance map generated based on the local crops. As a result, our method outperforms ZoomEye both in terms of accuracy and efficiency on the three datasets. Although LLaVA-OneVision natively supports resolutions up to 2K, FOCUS can still boost the accuracy on V*Bench (2K) from 74.46% to 92.15%. This can be attributed to our method isolating target objects and reducing irrelevant background regions. Furthermore, we perform additional evaluation on MME-RealWorld-Lite, see Tab. 2. FOCUS outperforms the vanilla baseline on most sub-tasks. ZoomEye and our method have strengths in different domains. FOCUS is better for reasoning, while ZoomEye is

Table 2: **Results on MME-RealWorld-Lite.** We provide the accuracy for each task. Further, we report the average accuracy and FPs per subset for efficiency comparison. The dataset includes the domains *OCR (Optical Character Recognition)*, *RS (Remote Sensing)*, *DT (Diagram and Table)*, *MO (Monitoring)*, and *AD (Autonomous Driving)*.

	Perception							**Reasoning**					
	Sub-task accuracy [%]					**Average**		**Sub-task accuracy** [%]				**Average**	
Model	OCR	RS	DT	MO	AD	**Acc.** [%] ↑	**FP** [#] ↓	OCR	DT	MO	AD	**Acc.** [%] ↑	**FP** [#] ↓
LLaVA-OV-7B	81.60	**52.00**	65.00	34.48	43.14	52.01	-	**72.00**	40.00	44.00	32.35	40.93	-
w/ ZoomEye	81.20	51.33	**74.00**	38.87	**51.43**	**56.29**	41.60	64.00	49.00	46.00	**35.50**	43.20	45.95
w/ FOCUS (Ours)	**83.60**	46.67	58.00	**41.07**	47.14	54.15	**7.71**	71.00	**52.00**	**51.33**	33.50	**44.53**	**8.21**

better at perception tasks. Still, our method is on average $5.47\times$ more efficient than ZoomEye. We provide the full numerical results in App. C.1.

Comparison with ZoomEye ZoomEye's strong performance on fine-grained VQA is largely driven by its exhaustive tree search, which includes a high number of image regions. This leads to a substantial number of FPs, as each region requires three FPs with different prompts for confidence prediction—making the overall process computationally expensive. In contrast, FOCUS leverages an object relevance map derived from internal representations to identify target object locations with just a single FP—explaining its superior efficiency compared to ZoomEye. We report additional efficiency metrics, including execution time, FLOPs and memory usage, in App. C.4. Crucially, our analysis shows that reducing FPs not only lowers theoretical computation but also leads to significantly shorter execution times, confirming the practical efficiency of FOCUS compared to ZoomEye.

4.3 Ablation studies and hyperparameter sensitivity analysis

In this subsection, we validate the design choices of FOCUS on V*Bench and HRBench-4K, using LLaVA-1.5 (see Tab. 3 and Fig. 4). V*Bench is the only dataset that provides GT region annotations, allowing us to report both accuracy and recall. We calculate recall by checking whether one of the proposed ROIs overlaps with the GT region by at least 50%. We use the same hyperparameters across all ablation studies, with the number of steps set to $n_{\text{steps}} = 8$ unless stated otherwise. Furthermore, we analyze the hyperparameter sensitivity of FOCUS on V*Bench.

Component analysis We first verify the contributions of the object relevance map (see Sec. 3.2) and the ROI ranking (see Sec. 3.3). To assess the impact of the object relevance map, we replace it with a randomly generated map. This substitution leads to a substantial decrease in accuracy on both datasets—interestingly, however, FOCUS still performs well above random guessing (35.99%). This indicates that even without an accurate global understanding of the object's location, our method can identify likely regions through confidence-based ROI selection, demonstrating our ranking mechanism's robustness. Next, we assess the effect of discarding ROI ranking by directly selecting the ROI with the highest object relevance score for the final VQA. This yields a recall of 38.48% on V*Bench, with accuracy improvements of 3 percentage points (pp.) on V*Bench and 5 pp. on HRBench-4K compared to the vanilla baseline. These results suggest that object relevance maps, even without post-processing, serve as a strong prior for identifying meaningful visual regions.

Table 3: **Ablation studies of FOCUS.** We evaluate the influences of design choices of our method based on accuracy and recall. "rel." is short for "relevance".

Ablation			V*Bench		HRBench-4K
Component	Object rel. map	Proposal ranking	Acc. [%] ↑	Recall [%] ↑	Acc. [%] ↑
	✗	✓	48.68	18.37	36.13
	✓	✗	51.30	38.48	41.13
Pseudo-attn.	K-K (w/o RoPE)		69.10	63.47	45.63
Layers	$0-14$		66.49	76.17	47.38
	$0-32$		71.20	75.56	49.38
Original design choice			**72.77**	**77.49**	**51.75**
Vanilla baseline			47.64	-	36.13
Random guess			35.99	-	25.00

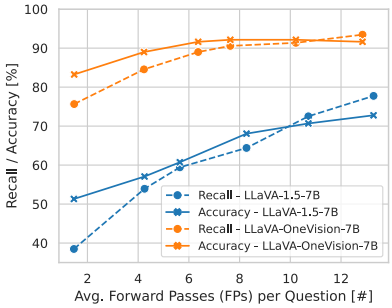

Figure 4: **Ablation studies on the search steps.** We analyze how the number of search steps influences the recall and accuracy on V*Bench. A positive correlation between accuracy and recall can be observed.

Value features vs. key features in pseudo-attention One might argue that key features—central to standard attention—encode richer semantic information [21] compared to value features and therefore could be used to generate more precise object relevance maps. However, directly substituting key features for value features in FOCUS results in degraded performance. This is mainly due to the use of Rotary Positional Embedding (RoPE) [34] in recent MLLMs [4, 9, 22, 24], which injects position-dependent rotations into the key features. As a result, RoPE causes nearby tokens to exhibit artificially high cosine similarity due to positional proximity, rather than semantic alignment [13, 34]. In this ablation, we remove RoPE from the key features to isolate semantic content before computing object relevance maps. Despite this, both accuracy and recall remain lower than those achieved

using value features (see Tab. 3). We hypothesize that removing RoPE disrupts the semantic integrity of the key features, as they are trained to operate with positional encoding, resulting in noisier relevance maps.

Later layers vs. earlier layers Apart from using the default later layers ($14-32$) for LLaVA-1.5, we also experiment with representations from earlier layers ($0-14$) or all layers ($0-32$). We observe that the representations from later layers yield the best performance across both datasets. This is consistent with findings from the Logit Lens technique [26], which suggests that later layers encode more predictive and semantically coherent representations.

Number of steps In FOCUS, we vary only the number of steps of the ROI ranking in our method (see Sec. 4.1). For this ablation, we report accuracy and recall on V*Bench using LLaVA-1.5 and LLaVA-OneVision. Across both models, recall increases with more steps, as additional lower-priority ROIs are explored and more of the image is covered (see Fig. 4). Similarly, accuracy also improves with more steps but begins to plateau beyond a certain point. This saturation might occur because the model fails to identify the correct region among a considerable number of proposed ROIs during ranking. Another reason could be that even when the correct region is provided, the MLLM is unable to give the correct answer, due to object size, ambiguity, or other limitations.

Hyperparameter analysis We investigate FOCUS's sensitivity on key hyperparameters and find it to be robust across a wide range of parameter choices. In particular, we do not observe any accuracy degradation larger than 4.71 pp. for LLaVA-1.5 and 2.62 pp. for LLaVA-OV. The complete results of this hyperparameter analysis are provided in App. C.3. Please note that this study was conducted post-hoc and not used to optimize the performance of FOCUS.

4.4 Additional results

In this subsection, we analyze how FOCUS performs on VQA questions involving large-size objects. Moreover, we provide results for FOCUS when using the state-of-the-art model Qwen-2.5-VL as the base MLLM.

Results on VQA datasets with large objects While the previously used datasets focus on fine-grained VQA, we also evaluate the performance of FOCUS on datasets featuring large-scale objects to assess its robustness across varying object sizes. We compare FOCUS on the datasets A-OKVQA [30] and GQA [18] using LLaVA-1.5 and LLaVA-OV with the vanilla model and ViCrop as the latter one is the only benchmark method providing respective results (see Tab. 4). Overall, FOCUS demonstrates strong robustness on datasets with large objects, maintaining competitive performance compared to the baseline models.

Results with Qwen-2.5-VL Qwen-2.5-VL [4] processes high-resolution images with native resolution, thereby preserving spatial details more effectively. We evaluate FOCUS with Qwen-2.5-VL-7B (see Tab. 5) and find state-of-the-art accuracy on HRBench-4K and HRBench-8K. This confirms the compatibility of FOCUS with different MLLM architectures.

Table 4: **Results on VQA datasets with large objects.** We find only minor performance degradation of FOCUS w.r.t. the base model.

	A-OKVQA		GQA	
Model	Acc. [%]	Δ	Acc. [%]	Δ
LLaVA-1.5	77.99	-	61.97	-
w/ ViCrop	60.66	-17.33	60.98	-0.99
w/ FOCUS	74.76	-3.23	60.34	-1.63
LLaVA-OV	91.44	-	62.01	-
w/ FOCUS	91.00	-0.44	51.02	-10.99

Table 5: **Results of FOCUS with Qwen-2.5-VL.** FOCUS significantly boosts the performance of the base model, consistent with previous results for LLaVA-1.5 and LLaVA-OV.

Model	**V*Bench** [%]	**HRBench-4K** [%]	**HRBench-8K** [%]
Qwen-2.5-VL	79.06	71.62	68.62
w/ FOCUS	**90.58**	**79.25**	**76.25**

4.5 Qualitative examples

We provide two qualitative examples that show how FOCUS improves performance of LLaVA-1.5 for single-target tasks and LLaVA-OneVision for multiple-target tasks (see Fig. 5). In both examples, the accurate visual crops generated by FOCUS enable the respective MLLM to answer the question correctly. The detected location of the target objects is highlighted in the object relevance map. One

can see that LLaVA-OneVision provides a higher-resolution and cleaner object relevance map. We provide more examples in App. D.

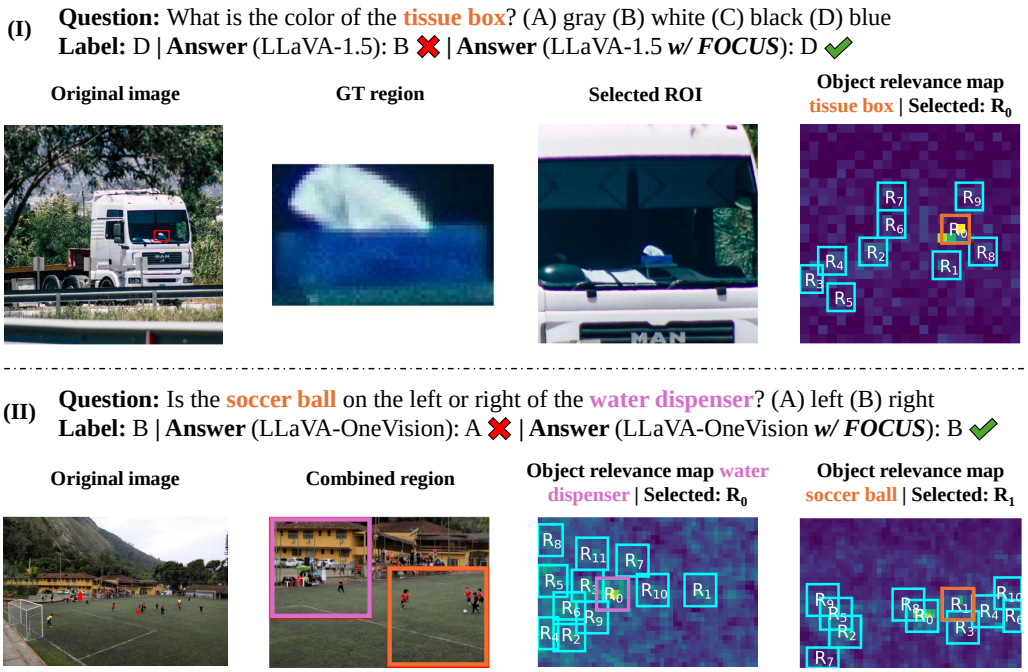

Figure 5: **Qualitative examples of FOCUS.** We provide some exemplary inferences with our method for single-target tasks with LLaVA-1.5 **(I)** and multiple-target tasks with LLaVA-OneVision **(II)**. The Ground Truth (GT) locations are highlighted in red in the original image. Further, we show the detected image regions and their locations in the object relevance map. Note that the object relevance maps corresponds to the original images.

4.6 Limitations

One limitation of our method is its reliance on the resolution of the object relevance map. When the input image is high-resolution (e.g., 8K), but the internal representations of the MLLM can only produce a low-resolution relevance map, accurately localizing fine-grained objects becomes difficult. This limitation partly explains the reduced effectiveness of our method with LLaVA-1.5 on HRBench-8K. A potential solution is to construct the object relevance map in a sliding-window manner over the image, allowing finer spatial resolution. Moreover, FOCUS inherits the typically limited understanding of spatial relationships [5, 35] from the base MLLM, as it is a training-free method. Thus, FOCUS struggles with spatial concepts such as "on the left/right of the image". We leave these shortcomings for future work.

5 Conclusion

In this work, we proposed FOCUS, an efficient, training-free visual cropping method for fine-grained VQA tasks, where identifying small objects is essential. Our method constructs an object relevance map from cached token representations to localize image regions relevant to the question, enabling detail-focused VQA inference. FOCUS achieves performance on par with or better than existing methods across multiple fine-grained VQA benchmarks, while requiring significantly fewer computational overhead. These results highlight the potential of training-free, high-resolution VQA systems that are both effective and computationally efficient. Moreover, the central idea of FOCUS—harnessing the hidden spatial capabilities of MLLMs via an inference-time method—holds significant promise for spatial reasoning tasks well beyond VQA.

Acknowledgment

The authors gratefully acknowledge financial support by the German Federal Ministry for Economic Affairs and Energy under the grant numbers 19A24004U and 19A24004I as a part of the *Safe AI Engineering* consortium.

DISCLAIMER: The results, opinions and conclusions expressed in this publication are not necessarily those of Volkswagen Aktiengesellschaft.

References

[1] Samira Abnar and Willem H. Zuidema. Quantifying attention flow in transformers. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel R. Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 4190–4197. Association for Computational Linguistics, 2020. doi: 10.18653/V1/2020.ACL-MAIN.385. URL https://doi.org/10.18653/v1/2020.acl-main.385.

[2] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob L. Menick, Sebastian Borgeaud, Andy Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karén Simonyan. Flamingo: a visual language model for few-shot learning. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/960a172bc7fbf0177ccccbb411a7d800-Abstract-Conference.html.

[3] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. VQA: visual question answering. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, pages 2425–2433. IEEE Computer Society, 2015. doi: 10.1109/ICCV.2015.279. URL https://doi.org/10.1109/ICCV.2015.279.

[4] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Ming-Hsuan Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report. *CoRR*, abs/2502.13923, 2025. doi: 10.48550/ARXIV.2502.13923. URL https://doi.org/10.48550/arXiv.2502.13923.

[5] Boyuan Chen, Zhuo Xu, Sean Kirmani, Brian Ichter, Dorsa Sadigh, Leonidas J. Guibas, and Fei Xia. Spatialvlm: Endowing vision-language models with spatial reasoning capabilities. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pages 14455–14465. IEEE, 2024. doi: 10.1109/CVPR52733.2024.01370. URL https://doi.org/10.1109/CVPR52733.2024.01370.

[6] Jierun Chen, Fangyun Wei, Jinjing Zhao, Sizhe Song, Bohuai Wu, Zhuoxuan Peng, S.-H. Gary Chan, and Hongyang Zhang. Revisiting referring expression comprehension evaluation in the era of large multimodal models. *CoRR*, abs/2406.16866, 2024. doi: 10.48550/ARXIV.2406.16866. URL https://doi.org/10.48550/arXiv.2406.16866.

[7] Liang Chen, Haozhe Zhao, Tianyu Liu, Shuai Bai, Junyang Lin, Chang Zhou, and Baobao Chang. An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models. In Ales Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol, editors, *Computer Vision - ECCV 2024 - 18th European Conference, Milan, Italy, September 29-October 4, 2024, Proceedings, Part LXXXI*, volume 15139 of *Lecture Notes in Computer Science*, pages 19–35. Springer, 2024. doi: 10.1007/978-3-031-73004-7_2. URL https://doi.org/10.1007/978-3-031-73004-7_2.

[8] Zhe Chen, Weiyun Wang, Hao Tian, Shenglong Ye, Zhangwei Gao, Erfei Cui, Wenwen Tong, Kongzhi Hu, Jiapeng Luo, Zheng Ma, Ji Ma, Jiaqi Wang, Xiaoyi Dong, Hang Yan, Hewei Guo, Conghui He, Botian Shi, Zhenjiang Jin, Chao Xu, Bin Wang, Xingjian Wei, Wei Li, Wenjian Zhang, Bo Zhang, Pinlong Cai, Licheng Wen, Xiangchao Yan, Min Dou, Lewei Lu, Xizhou Zhu, Tong Lu, Dahua Lin, Yu Qiao, Jifeng Dai, and Wenhai Wang. How far are we to gpt-4v? closing the gap to commercial multimodal models with open-source suites. *Sci. China Inf. Sci.*, 67(12), 2024. doi: 10.1007/S11432-024-4231-5. URL https://doi.org/10.1007/s11432-024-4231-5.

[9] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven C. H. Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine, editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/9a6a435e75419a836fe47ab6793623e6-Abstract-Conference.html.

[10] Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/67d57c32e20fd0a7a302cb81d36e40d5-Abstract-Conference.html.

[11] Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision transformers need registers. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=2dnO3LLiJ1.

[12] Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Baobao Chang, Xu Sun, Lei Li, and Zhifang Sui. A survey on in-context learning. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pages 1107–1128. Association for Computational Linguistics, 2024. URL https://aclanthology.org/2024.emnlp-main.64.

[13] Mark Endo, Xiaohan Wang, and Serena Yeung-Levy. Feather the throttle: Revisiting visual token pruning for vision-language model acceleration. *CoRR*, abs/2412.13180, 2024. doi: 10.48550/ARXIV.2412.13180. URL https://doi.org/10.48550/arXiv.2412.13180.

[14] Ross B. Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014*, pages 580–587. IEEE Computer Society, 2014. doi: 10.1109/CVPR.2014.81. URL https://doi.org/10.1109/CVPR.2014.81.

[15] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[16] Zonghao Guo, Ruyi Xu, Yuan Yao, Junbo Cui, Zanlin Ni, Chunjiang Ge, Tat-Seng Chua, Zhiyuan Liu, and Gao Huang. Llava-uhd: An LMM perceiving any aspect ratio and high-resolution images. In Ales Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol, editors, *Computer Vision - ECCV 2024 - 18th European Conference, Milan, Italy, September 29-October 4, 2024, Proceedings, Part LXXXIII*, volume 15141 of *Lecture Notes in Computer Science*, pages 390–406. Springer, 2024. doi: 10.1007/978-3-031-73010-8_23. URL https://doi.org/10.1007/978-3-031-73010-8_23.

[17] Jan Hendrik Hosang, Rodrigo Benenson, and Bernt Schiele. Learning non-maximum suppression. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 6469–6477. IEEE Computer Society, 2017. doi: 10.1109/CVPR.2017.685. URL https://doi.org/10.1109/CVPR.2017.685.

[18] Drew A. Hudson and Christopher D. Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[19] Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean-Bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, Gaël Liu, Francesco Visin, Kathleen Kenealy, Lucas Beyer, Xiaohai Zhai, Anton Tsitsulin, Róbert Busa-Fekete, Alex Feng, Noveen Sachdeva, Benjamin Coleman, Yi Gao, Basil Mustafa, Iain Barr, Emilio Parisotto, David Tian, Matan Eyal, Colin Cherry, Jan-Thorsten Peter, Danila Sinopalnikov, Surya Bhupatiraju, Rishabh Agarwal, Mehran Kazemi, Dan Malkin, Ravin Kumar, David Vilar, Idan Brusilovsky, Jiaming Luo, Andreas Steiner, Abe Friesen, Abhanshu Sharma, Abheesht Sharma, Adi Mayrav Gilady, Adrian Goedeckemeyer, Alaa Saade, Alexander Kolesnikov, Alexei Bendebury, Alvin Abdagic, Amit Vadi, András György, André Susano Pinto, Anil Das, Ankur Bapna, Antoine Miech, Antoine Yang, Antonia Paterson, Ashish Shenoy, Ayan Chakrabarti, Bilal Piot, Bo Wu, Bobak Shahriari, Bryce Petrini, Charlie Chen, Charline Le Lan, Christopher A. Choquette-Choo, CJ Carey, Cormac Brick, Daniel Deutsch, Danielle Eisenbud, Dee Cattle, Derek Cheng, Dimitris Paparas, Divyashree Shivakumar Sreepathihalli, Doug Reid, Dustin Tran, Dustin Zelle, Eric Noland, Erwin Huizenga, Eugene Kharitonov, Frederick Liu, Gagik Amirkhanyan, Glenn Cameron, Hadi Hashemi, Hanna Klimczak-Plucinska, Harman Singh, Harsh Mehta, Harshal Tushar Lehri, Hussein Hazimeh, Ian Ballantyne, Idan Szpektor, and Ivan Nardini. Gemma 3 technical report. *CoRR*, abs/2503.19786, 2025. doi: 10.48550/ARXIV.2503.19786. URL https://doi.org/10.48550/arXiv.2503.19786.

[20] Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara L. Berg. Referitgame: Referring to objects in photographs of natural scenes. In Alessandro Moschitti, Bo Pang, and Walter Daelemans, editors, *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 787–798. ACL, 2014. doi: 10.3115/V1/D14-1086. URL https://doi.org/10.3115/v1/d14-1086.

[21] Mengcheng Lan, Chaofeng Chen, Yiping Ke, Xinjiang Wang, Litong Feng, and Wayne Zhang. Proxyclip: Proxy attention improves CLIP for open-vocabulary segmentation. In Ales Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol, editors, *Computer Vision - ECCV 2024 - 18th European Conference, Milan, Italy, September 29-October 4, 2024, Proceedings, Part LXVIII*, volume 15126 of *Lecture Notes in Computer Science*, pages 70–88. Springer, 2024. doi: 10.1007/978-3-031-73113-6_5. URL https://doi.org/10.1007/978-3-031-73113-6_5.

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

[23] Feng Li, Renrui Zhang, Hao Zhang, Yuanhan Zhang, Bo Li, Wei Li, Zejun Ma, and Chunyuan Li. Llava-next-interleave: Tackling multi-image, video, and 3d in large multimodal models. *CoRR*, abs/2407.07895, 2024. doi: 10.48550/ARXIV.2407.07895. URL https://doi.org/10.48550/arXiv.2407.07895.

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

[25] Clement Neo, Luke Ong, Philip Torr, Mor Geva, David Krueger, and Fazl Barez. Towards interpreting visual information processing in vision-language models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=chanJGoa7f.

[26] Nostalgebraist. Interpreting gpt: The logit lens. URL https://www.lesswrong.com/posts/AcKRB8wDpdaN6v6ru/interpreting-gpt-the-logit-lens.

[27] Reiner Pope, Sholto Douglas, Aakanksha Chowdhery, Jacob Devlin, James Bradbury, Jonathan Heek, Kefan Xiao, Shivani Agrawal, and Jeff Dean. Efficiently scaling transformer inference. In Dawn Song, Michael Carbin, and Tianqi Chen, editors, *Proceedings of the Sixth Conference on Machine Learning and Systems, MLSys 2023, Miami, FL, USA, June 4-8, 2023*. mlsys.org, 2023. URL https://proceedings.mlsys.org/paper_files/paper/2023/hash/c4be71ab8d24cdfb45e3d06dbfca2780-Abstract-mlsys2023.html.

[28] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. 2018. URL https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf. OpenAI technical report.

[29] 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. Learning transferable visual models from natural language supervision. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR, 2021. URL http://proceedings.mlr.press/v139/radford21a.html.

[30] Dustin Schwenk, Apoorv Khandelwal, Christopher Clark, Kenneth Marino, and Roozbeh Mottaghi. A-okvqa: A benchmark for visual question answering using world knowledge. In Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner, editors, *Computer Vision – ECCV 2022*, pages 146–162, Cham, 2022. Springer Nature Switzerland. ISBN 978-3-031-20074-8.

[31] Haozhan Shen, Kangjia Zhao, Tiancheng Zhao, Ruochen Xu, Zilun Zhang, Mingwei Zhu, and Jianwei Yin. ZoomEye: Enhancing multimodal llms with human-like zooming capabilities through tree-based image exploration. In *Conference on Empirical Methods in Natural Language Processing 2025*, 2025. doi: 10.48550/ARXIV.2411.16044. URL https://doi.org/10.48550/arXiv.2411.16044.

[32] Ying Sheng, Lianmin Zheng, Binhang Yuan, Zhuohan Li, Max Ryabinin, Beidi Chen, Percy Liang, Christopher Ré, Ion Stoica, and Ce Zhang. Flexgen: High-throughput generative inference of large language models with a single GPU. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 31094–31116. PMLR, 2023. URL https://proceedings.mlr.press/v202/sheng23a.html.

[33] Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards VQA models that can read. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 8317–8326. Computer Vision Foundation / IEEE, 2019. doi: 10.1109/CVPR.2019.00851. URL http://openaccess.thecvf.com/content_CVPR_2019/html/Singh_Towards_VQA_Models_That_Can_Read_CVPR_2019_paper.html.

[34] Jianlin Su, Murtadha H. M. Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568: 127063, 2024. doi: 10.1016/J.NEUCOM.2023.127063. URL https://doi.org/10.1016/j.neucom.2023.127063.

[35] Shengbang Tong, Zhuang Liu, Yuexiang Zhai, Yi Ma, Yann LeCun, and Saining Xie. Eyes wide shut? exploring the visual shortcomings of multimodal llms. In *IEEE/CVF Conference*

on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024, pages 9568–9578. IEEE, 2024. doi: 10.1109/CVPR52733.2024.00914. URL https://doi.org/10.1109/CVPR52733.2024.00914.

[36] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

[37] Wenbin Wang, Liang Ding, Minyan Zeng, Xiabin Zhou, Li Shen, Yong Luo, and Dacheng Tao. Divide, conquer and combine: A training-free framework for high-resolution image perception in multimodal large language models. *CoRR*, abs/2408.15556, 2024. doi: 10.48550/ARXIV.2408.15556. URL https://doi.org/10.48550/arXiv.2408.15556.

[38] Penghao Wu and Saining Xie. V*: Guided visual search as a core mechanism in multimodal llms. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pages 13084–13094. IEEE, 2024. doi: 10.1109/CVPR52733.2024.01243. URL https://doi.org/10.1109/CVPR52733.2024.01243.

[39] Long Xing, Qidong Huang, Xiaoyi Dong, Jiajie Lu, Pan Zhang, Yuhang Zang, Yuhang Cao, Conghui He, Jiaqi Wang, Feng Wu, and Dahua Lin. Pyramiddrop: Accelerating your large vision-language models via pyramid visual redundancy reduction. *CoRR*, abs/2410.17247, 2024. doi: 10.48550/ARXIV.2410.17247. URL https://doi.org/10.48550/arXiv.2410.17247.

[40] Yuan Yao, Tianyu Yu, Ao Zhang, Chongyi Wang, Junbo Cui, Hongji Zhu, Tianchi Cai, Haoyu Li, Weilin Zhao, Zhihui He, Qianyu Chen, Huarong Zhou, Zhensheng Zou, Haoye Zhang, Shengding Hu, Zhi Zheng, Jie Zhou, Jie Cai, Xu Han, Guoyang Zeng, Dahai Li, Zhiyuan Liu, and Maosong Sun. Minicpm-v: A GPT-4V level MLLM on your phone. *CoRR*, abs/2408.01800, 2024. doi: 10.48550/ARXIV.2408.01800. URL https://doi.org/10.48550/arXiv.2408.01800.

[41] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*, pages 11941–11952. IEEE, 2023. doi: 10.1109/ICCV51070.2023.01100. URL https://doi.org/10.1109/ICCV51070.2023.01100.

[42] Jiarui Zhang, Mahyar Khayatkhoei, Prateek Chhikara, and Filip Ilievski. MLLMs know where to look: Training-free perception of small visual details with multimodal LLMs. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=DgaY5mDdmT.

[43] YiFan Zhang, Huanyu Zhang, Haochen Tian, Chaoyou Fu, Shuangqing Zhang, Junfei Wu, Feng Li, Kun Wang, Qingsong Wen, Zhang Zhang, Liang Wang, and Rong Jin. MME-realworld: Could your multimodal LLM challenge high-resolution real-world scenarios that are difficult for humans? In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=k5VHHgsRbi.

[44] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine, editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/91f18a1287b398d378ef22505bf41832-Abstract-Datasets_and_Benchmarks.html.

A Implementation details

In this section, we provide additional technical details of our work. We begin by describing the hardware setup and report the total GPU hours required to reproduce our main results in App. A.1. Next, we explain in detail how the reported metrics are calculated, see App. A.2. We then provide the hyperparameters used for FOCUS and the baseline visual cropping methods in App. A.3 and App. A.4, respectively. Finally, we present further details on the processing scheme of FOCUS in App. A.5.

A.1 Hardware specifications

We run all experiments presented in Sec. 4 and App. C on identical hardware, namely compute instances equipped with an *NVIDIA A100 80GB* GPU, an *AMD EPYC 7V13* CPU, and *220 GB* of RAM. FOCUS's software environment includes *CUDA 12.2*, *PyTorch 2.6.0*, and the HuggingFace *transformers* library in version *4.46.0*.

We also provide an estimation of the GPU hours needed to reproduce our results in the entire project. In total, our reported results require approximately 132 GPU hours on the hardware configuration described above (see Tab. 6). Additionally, we report the GPU hours required to run the experiments that achieve the best-performing variants for each method on LLaVA-1.5 and LLaVA-OneVision.

Table 6: **Estimated GPU hours.**

Model	Best-perf. variant w/ LLaVA-1.5	Best-perf. variant w/ LLaVA-OV	All variants
SEAL	6 [1]	–	6
ViCrop	3	–	6
ZoomEye	5	30	80
FOCUS **(Ours)**	2	9	40
Sum	16	39	132

A.2 Metrics

In this subsection, we provide a detailed description of our performance and efficiency metrics.

Performance metrics The main performance metric is the accuracy of visual cropping methods, which is typically reported as a percentage. We follow the standard evaluation protocol and do not apply any post-processing to the answers generated by the visual cropping methods using the MLLM. This means that if the ground-truth label is `"A"` and the model outputs `"The answer is A"` or `"(A)"`, we do not extract or normalize the answer to match the label. As a result, such responses are counted as incorrect. However, this type of mismatch is of minor importance in our experiments: across all models and three datasets (V*Bench, HRBench-4K, HRBench-8K), we did not observe any irregular or non-standard response formats. We compute the average accuracy as the unweighted mean over all N samples in a dataset, i.e., as $\sum_{i=1}^{N}(\hat{y}_i = y_i)/N$, where $\hat{y}_i$ is the predicted answer and y_i is the ground-truth label for the i-th sample.

Efficiency metrics An important metric is the number of Forward Passes (FPs) required for the visual search. Note that we exclude the FPs needed for the final VQAs prediction, as different inference schemes can generate different forward passes, see App. C.5. We compute this by tracking how often the `generate`, `forward`, or `__call__` methods of the respective MLLM are invoked per question. For methods that use multiple MLLMs (e.g., SEAL), we report the total number of FPs of all MLLMs.

Another key metric we report is the efficiency improvement of FOCUS relative to the baseline methods. Among all evaluated approaches, FOCUS demonstrates the highest efficiency. To ensure a fair comparison, we consider the top accuracy achieved by FOCUS and the best-performing baseline, and select the lower of the two as the reference accuracy. For both methods, we determine the number of Forward Passes (FPs) required to reach this reference accuracy—either by taking the exact value or

[1] SEAL uses customized MLLMs based on LLaVA-7B.

interpolating between data points to estimate the FPs. This yields FP_{ours} for FOCUS and FP_{ref} for the best-performing baseline. The relative efficiency improvement is then computed as $\text{FP}_{\text{ref}}/\text{FP}_{\text{ours}}$.

Furthermore, we report two additional metrics to provide a practical comparison of efficiency, i.e., execution time and peak GPU memory usage (see Tab. 13). Execution time is recorded per sample, and we report the average time per question across a dataset. Peak memory is measured using `torch.cuda.max_memory_allocated()` and converted to GB by dividing the result by 1024^3.

A.3 Hyperparameters of FOCUS

This subsection outlines the hyperparameters used in our method, FOCUS. These parameters are applied consistently across all experiments and correspond to the results reported in Sec. 4. As noted in the main paper, the only parameter we vary is the number of steps, n_{steps}. Most other hyperparameters remain fixed across experiments for both LLaVA-1.5 and LLaVA-OneVision. The exceptions are three parameters: k and s_{dist} (specific to LLaVA-1.5), and s_{max} (specific to LLaVA-OneVision), as detailed in Tab. 7.

For LLaVA-1.5, we use a smaller value of k when n_{steps} is low to reduce the number of proposed ROIs. Additionally, we increase s_{dist} on HRBench to ensure a broader spatial distribution of the ROIs across the image. In contrast, for LLaVA-OneVision, we set a larger k due to its higher-resolution object relevance map. Moreover, we set $s_{\text{max}} = 9$ for V*Bench and $s_{\text{max}} = 5$ for the other datasets, as V*Bench contains lower-resolution (2K) images.

The hyperparameters used in FOCUS are generally robust and transferable across a wide range of use cases. For users applying our method to new datasets, we recommend adjusting s_{max}, which determines the maximum size of each proposed ROIs, based on both the resolution of the input images and the spatial resolution of the object relevance map. In particular, s_{max} should be chosen so that the corresponding region in the original image spans approximately 1–$2\times$ the base resolution of the vision encoder. For example, if the object relevance map has a spatial resolution of 60×30, and the input image resolution is 7680×3840, then each grid element corresponds to an area of 128×128. Setting $s_{\text{max}} = 5$ yields a maximum crop size of 640×640, which falls within the recommended range of $384 \times 384 - 768 \times 768$ for the SigLIP encoder. Additionally, if one considers migrating our method to another MLLM, we recommend selecting the last $25\% - 60\%$ of the layers.

Table 7: **Hyperparameters of FOCUS.**

Hyper-parameter	Description	LLaVA-1.5	LLaVA-OneVision
k	Number of anchor points	$k = \begin{cases} 15 & \text{if } n_{\text{steps}} < 4 \\ 30 & \text{otherwise} \end{cases}$	$k = 30$
s_{min}	Minimum size of each ROI	$s_{\text{min}} = 3$	$s_{\text{min}} = 3$
s_{max}	Maximum size of each ROI	$s_{\text{max}} = 5$	$s_{\text{max}} = \begin{cases} 9 & \text{for V*Bench} \\ 5 & \text{otherwise} \end{cases}$
s_{dist}	Minimum Euclidean distance between anchor points	$s_{\text{dist}} = \begin{cases} 2 & \text{for V*Bench} \\ 3 & \text{otherwise} \end{cases}$	$s_{\text{dist}} = 2$
l	Start layer of the used MLLM-internal representations	$l = 14$	$l = 21$
L	End layer of the used MLLM-internal representations	$L = 32$	$L = 28$
t_{type2}	Threshold for inclusion of ROIs for type-2 questions	$t_{\text{type2}} = 0.6$	$t_{\text{type2}} = 0.5$
$t_{\text{obj\_dist}}$	Threshold for merging ROIs of nearby objects (see App. A.5)	$t_{\text{obj\_dist}} = 1200$	–

A.4 Hyperparameters of recent visual cropping methods

This subsection outlines the hyperparameters used for the baseline methods, i.e., for DC^2, SEAL, ViCrop, and ZoomEye. Further, we provide a visual comparison of these baseline methods in Fig. 6.

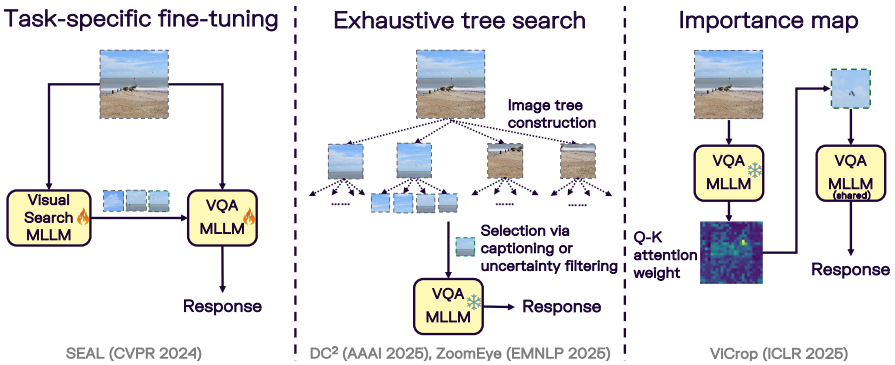

Figure 6: **Comparison of recent visual cropping methods.** We categorize the baseline methods SEAL, DC^2, ZoomEye, and ViCrop based on whether they: (1) require task-specific fine-tuning, (2) employ exhaustive tree search to identify relevant image regions, or (3) compute importance maps using attention matrices.

For DC^2, the full evaluation code is not publicly available, and we were unable to reproduce their results using the provided demo code. Therefore, we report the performance metrics as stated in their paper and estimate the number of Forward Passes (FPs) based on the available demo. Specifically, we follow the procedure described in the paper: splitting the image into patches using the base resolution of the vision encoder (i.e., 336×336 for LLaVA-1.5) and merging patches via hierarchical clustering to improve efficiency. Although the FPs for DC^2 are estimated and may carry a high margin of error, the method remains less efficient than other baselines. This is further exacerbated by its region-wise captioning step, which makes it more computationally intensive than other baselines, even when the number of FPs is comparable. Furthermore, DC^2 consistently underperforms compared to the other methods across all three datasets—V*Bench, HRBench-4K, and HRBench-8K—a trend also noted in ZoomEye's evaluation.

For SEAL, we use the hyperparameters described in their paper and the default configuration provided in their code. Specifically, we set the minimum search size to 224 and the minimum search scale to 4. For the visual search, we use a confidence lower bound of 0.3 and a confidence upper bound of 0.5. Regarding the target cue, we set the threshold to 6, the decay factor to 0.7, and the minimum threshold to 4. Note that these parameters were originally configured for the V*Bench dataset, and we did not adjust them for the other datasets.

For ViCrop, we select only the two best-performing variants from their work: `att-grad-high` and `rel-att-high`. Notably, both methods employ the high-resolution processing scheme `high`, which divides high-resolution images into a grid of $1K$ sub-images and computes importance maps for each sub-image individually. As a result, the computational overhead increases significantly with higher input resolutions.

For ZoomEye, we report more results per dataset–model combination than those presented in the original paper to offer a more comprehensive view of its efficiency–accuracy trade-offs. Specifically, we vary two key parameters: the number of sub-regions into which each region is split (2 or the default 4 crops), and the depth of the search tree (1, 2, and the default 5). All other hyperparameters are as specified in the ZoomEye paper.

A.5 Additional implementation details of `FOCUS`

We provide additional implementation details of `FOCUS` to ensure reproducibility, including how to construct the object relevance maps and how to perform the final VQA prediction.

Constructing object relevance maps We provide PyTorch-style pseudocode in Fig. 7. For a detailed motivation and description of this method, see Sec. 3.

Verification of object existence in ROI Inspired by ZoomEye [31], we use a yes/no prompt `"Is there a {target object} in the image?"` to verify whether the target object exists within a ROI. We compute a confidence score c_{Yes} based on the softmax-normalized logits $l_{\text{Yes}}, l_{\text{No}}$ corresponding to the responses `"Yes"` and `"No"`. Specifically, c_{Yes} is defined as:

$$c_{\text{Yes}} = 2 \cdot (\text{softmax}([l_{\text{Yes}}, l_{\text{No}}])_{\text{Yes}} - 0.5), \quad c_{\text{Yes}} \in [-1, 1] \,. \tag{5}$$

```
# X: value features
# O: total number of LLM layers
# H: total number of attention heads/kv-cache heads
# N: sequence length
# D: hidden size of each token embedding
# s: total number of target tokens
# bmm: batch matrix multiplication
# a: grid size of vision encoder
# l: start of the selected layer
# L: end of the selected layer
# x_visual: indices of the visual tokens, [a**2,]
# x_target: indices of the target tokens, [s,]
# x_layer: indices of selected layers, [L-l,]

X = l2_normalize(X, dim=-1) # O, H, N, D

X = reshape(X) # O, N, H*D

# calculate cosine similarity
X = bmm(X, X.permute(0, 2, 1)) # O, N, N

# initialize relevance map
object_rel_map = ones_like(x_visual)

for target in x_target:

    # initialize pseudo-attn for each target token
    pseudo_attn = zeros_like(x_visual) # [a**2]

    for layer in x_layer:
        identity = eye(N) # [N, N]

        # add residual connection
        X_layer = (X[layer] + identity) / 2 # [N, N]

        # row-wise normalization
        X_layer /= X_layer.sum(dim=-1) # [N, N]

        layer_pseudo_attn = X_layer[target, x_visual] # [a**2]

        pseudo_attn += layer_pseudo_attn # [a**2]

    # global normalization
    pseudo_attn /= pseudo_attn.sum()

    # element-wise multiplication of pseudo-attentions
    object_rel_map *= pseudo_attn

object_rel_map = reshape(object_rel_map) # [a, a]
```

Figure 7: **PyTorch-style pseudocode for constructing object relevance maps in FOCUS.**

Final VQA prediction As explained in Sec. 3.3, we differentiate between type-1 and type-2 questions. Type-1 questions involve single-object instances, e.g., "`What is the color of the car?`" or "`What is the relative position of the ball to the bench?`". Type-2 questions concern multiple instances of an object type, e.g., "`How many bikes are in the image?`". We follow the inference strategy introduced in ZoomEye [31] when passing the selected ROIs to the MLLMs. In this appendix, we elaborate on the strategy used for global-view MLLMs, which typically accept only a single low-resolution image input.

For type-1 questions, we select the ROI with the highest confidence score for each target object. If the question involves multiple target objects, we compute the combined image region covering all targets and use this as the final VQA input. However, this approach may fail when the target objects

are far apart, as the resulting combined region can become excessively large and include irrelevant background. To address this, a fallback strategy is taken. If the Euclidean distance between any pair of target objects exceeds a threshold $t_{\text{obj\_dist}}$ (e.g., 1200 pixels), we instead resize each individual ROI to a lower resolution and paste them onto an empty canvas based on their relative positions. This avoids including unnecessary background while maintaining spatial relationships. The resulting composite image is then used for the final VQA prediction. For type-2 questions, we first select all ROIs whose confidence scores exceed a predefined threshold. Overlapping regions are then merged. The merged ROIs are subsequently placed on a canvas using the pasting strategy described above.

In the case of global-local-view MLLMs, this workaround is unnecessary, as these models natively support multi-image reasoning without requiring canvas composition. Fine-grained details in local regions can be preserved by supplying them as separate image inputs. We leverage the model's text–image interleaved capabilities [2, 22, 23] by providing a global view - with all ROIs visually highlighted, alongside the individual high-confidence local crops.

Proposal ranking of ROIs for fine-grained VQA Further, we present pseudocode in Algorithm that outlines the creation and ranking of ROIs to identify the most relevant ROI for the final inference, as described in Sec. 3.3.

Algorithm Pseudocode for proposal ranking of ROIs for fine-grained VQA

Input: Object relevance map A, number of anchor points k, number of steps n_{steps}, ROI parameters: s_{min}, s_{max}, s_{dist}; threshold for NMS: $\text{NMS}_{\text{threshold}}$; model MLLM

anchor_points ← `Extract-Top-K-Anchors`(A, k)
initial_rois ← `Generate-Symmetric-ROIs`(anchor_points, s_{min})
expanded_rois ← `Expand-ROIs`(initial_rois, A, s_{max}, s_{dist})
filtered_rois ← `Apply-NMS`(expanded_rois, $\text{NMS}_{\text{threshold}}$)
ranked_rois ← `Rank-ROIs-By-Confidence`(filtered_rois, n_{steps})
answer ← `Final-Inference`(MLLM, ranked_rois)

Output: MLLM answer based on top-ranked ROI

B Dataset statistics

This section provides a summary of the datasets used in our experiments. For each dataset, we report the average image resolution, the number of images, and the number of question–answer pairs in Tab. 8.

Table 8: **Statistics of the employed VQA datasets**

Attribute	V*Bench	HRBench-4K	HRBench-8K	MME-RealWorld-Lite
Avg. width	2,246	4,024	7,431	2,836
Avg. height	1,582	3,503	5,358	1,566
Images [#]	191	200	200	1,543
QA-pairs [#]	191	800	800	1,919

V*Bench [38] comprises images with an average resolution of 2K, specifically $2{,}256 \times 1{,}582$ pixels. The dataset includes two types of tasks: *direct attribute* and *spatial relation*. The *direct attribute* task involves identifying visual properties of a single object (e.g., color), making them single-target tasks. In contrast, the *spatial relation* task requires predicting the spatial relationship between two objects, thus constituting multiple-target tasks. In total, the dataset contains 191 images, each paired with a single question, resulting in 191 question-answer (QA) pairs.

HRBench [37] comprises two sub-datasets with average image resolutions of 4K and 8K, respectively. Both sub-datasets include two task types: *FSP* (Fine-grained Single-instance Perception) and *FCP* (Fine-grained Cross-instance Perception). *FSP* questions are single-target and focus on identifying fine-grained attributes of individual objects, whereas *FCP* questions are multiple-target and involve reasoning about spatial relationships between target objects. HRBench-8K contains full-resolution images with an average size of $7{,}431 \times 5{,}358$ pixels, while HRBench-4K provides cropped versions

of these images, on average with $4{,}024 \times 3{,}503$ pixels. Each dataset contains one question per image. To improve evaluation robustness, HRBench permutes the positions of the answer options, yielding a total of 800 question–answer pairs across 200 images.

MME-RealWorld-Lite [43] is a real-world, fine-grained VQA dataset with an average image resolution of $2{,}836 \times 1{,}566$ pixels. It includes a diverse set of subtasks designed to evaluate the perception and reasoning capabilities of MLLMs across various domains, such as Autonomous Driving and OCR. The dataset is divided into two subsets: *Perception* and *Reasoning*. Each subset contains questions from multiple domains, including *OCR* (Optical Character Recognition), *RS* (Remote Sensing), *DT* (Diagram and Table), *MO* (Monitoring), and *AD* (Autonomous Driving), resulting in a total of nine tasks, as shown in Tab. 9. It comprises $1{,}543$ images, with some images associated with multiple questions, leading to a total of $1{,}919$ questions. The number of QA pairs per task is detailed in Tab. 9.

Table 9: **Number of question-answer (QA) pairs in MME-Realworld-Lite per domain**

	Perception					**Reasoning**			
	OCR	RS	DT	MO	AD	OCR	DT	MO	AD
QA-pairs [#]	250	150	100	319	350	100	100	150	400

C Additional results

We include full numerical results in App. C.1, results on open-ended VQA in App. C.2, and an analysis of the hyperparameter influence on `FOCUS` in App. C.3. Further, we provide additional efficiency metrics in App. C.4 and a comparison of different inference schemes as well as performance discrepancies between reported and reproduced results in App. C.5.

C.1 Full results

We present the full results of ZoomEye [31], ViCrop [42], DC^2 [37], SEAL [38], and our method `FOCUS` on V*Bench, HRBench-4K, and HRBench-8K in Tab. 10. For MME-RealWorld-Lite [43], we compare ZoomEye, the vanilla baseline, and `FOCUS`, see Tab. 11.

As described in Sec. 4.1, we run experiments with $n_{\text{steps}} \in \{1, 2, 3, 4, 6, 8\}$. To better leverage the reasoning capabilities of MLLMs under higher computational budgets, we allow an `overrun` mode: if the logit l_{Yes} is lower than l_{No}—i.e., the MLLM responds `"No"` to the existence prompt for all of the top-n_{steps} ROIs—the model continues evaluating additional ROIs until it receives a `"Yes"` response. This mechanism improves the efficiency trade-off in many cases. For $n_{\text{steps}} \in \{1, 2\}$, we report results both with and without the `overrun` mechanism to provide a complete comparison in the low-computation setting.

In general, accuracy improves with increased computation budget for methods that support a configurable computation budget—such as ZoomEye and `FOCUS`. ZoomEye exhibits *exponential* scaling in the number of evaluated regions due to its hierarchical tree structure, leading to a significantly higher number of Forward Passes (FPs) when targeting high accuracy. In contrast, `FOCUS` constructs an object-aware relevance map and directly retrieves the most relevant regions, resulting in *linear* scaling with respect to the number of FPs.

C.2 Open-ended VQA

While the primary results in the main paper are concerned with multiple-choice VQA, here we focus on the evaluation on open-ended VQA. As we are not aware of fine-grained datasets focusing on open-ended VQA, we reuse V*Bench for this task. While it follows the multiple-choice format, it also provides the ground-truth answer in a natural language format, e.g., `"The color of the dog is white."`. To explore the open-ended VQA capabilities of `FOCUS` with LLaVA-1.5, we provide it with VQA questions without answer options, e.g., `"What is the color of the dog?"`. Then, we compare the responses with the ground-truth sequence using an LLM-as-a-judge framework [44], leveraging *Qwen-2.5-7B* [4]. Moreover, we manually review Qwen-2.5's judgments and correct any misclassifications. The results of this analysis for LLaVA-1.5 clearly show that `FOCUS` substantially

Table 10: **Full results of different models on fine-grained VQA benchmarks.** V*Bench comprises two tasks, namely direct attribute (Attr) and spatial relationship (Spatial). Similarly, HRBench consists of two tasks, i.e., Fine-grained Single-instance Perception (FSP) and Fine-grained Cross-instance Perception (FCP). The highest accuracy for each method-model combination is highlighted in bold. As DC^2 does not provide the complete evaluation code, we report the accuracy from the original paper and estimate the number of FPs following the procedure described in App. A.4.

Model	V*Bench Overall Acc. ↑ (Attr | Spatial) [%]	V*Bench FP [#] ↓	HRBench-4K Overall Acc. ↑ (FSP | FCP) [%]	HRBench-4K FP [#] ↓	HRBench-8K Overall Acc. ↑ (FSP | FCP) [%]	HRBench-8K FP [#] ↓
ZoomEye						
LLaVA-1.5-7B						
`Depth-1 (2 crops)`	50.26 (41.74 | 63.16)	12.50	36.25 (39.25 | 33.25)	11.54	33.88 (32.25 | 35.5)	11.55
`Depth-1 (4 crops)`	50.78 (41.74 | 64.47)	20.37	39.13 (44.75 | 33.5)	17.46	32.88 (32.75 | 33.00)	18.18
`Depth-2 (4 crops)`	71.20 (67.83 | 76.32)	44.54	47.75 (57.25 | 38.25)	35.60	42.13 (49.00 | 36.25)	39.15
`Depth-5 (4 crops)`	**77.48** (80.87 | 72.37)	48.63	**50.00** (63.25 | 36.75)	49.38	**49.00** (61.75 | 36.25)	59.64
LLaVA-OV-7B						
`Depth-1 (2 crops)`	75.92 (77.39 | 73.68)	10.52	61.38 (70.00 | 52.75)	9.46	59.00 (66.50 | 51.50)	9.02
`Depth-1 (4 crops)`	81.15 (81.74 | 80.26)	13.34	65.75 (80.75 | 50.75)	11.43	59.63 (69.00 | 50.25)	12.59
`Depth-2 (4 crops)`	90.05 (93.04 | 85.53)	21.08	66.13 (80.25 | 52.00)	20.42	65.63 (81.75 | 49.5)	22.31
`Depth-5 (4 crops)`	**91.10** (93.91 | 86.84)	23.98	**69.38** (84.5 | 54.25)	29.19	**69.00** (86.75 | 51.25)	36.75
ViCrop						
LLaVA-1.5-7B						
`rel-att-high`	**59.16** (58.26 | 60.53)	12.26	42.50 (51.50 | 33.50)	30.59	**39.38** (48.00 | 30.75)	78.60
`grad-att-high`	54.97 (53.04 | 57.90)	6.63	**44.25** (53.75 | 34.75)	15.80	38.38 (44.25 | 32.50)	39.80
DC^2						
LLaVA-v.1.5-7B	**57.00** (— | —)	18.18	**42.30** (— | —)	48.55	**39.50** (— | —)	77.02
SEAL	**73.68** (— | —)	25.53	**34.50** (— | —)	18.05	**33.50** (— | —)	16.96
`FOCUS` **(Ours)**						
LLaVA-1.5-7B						
`Steps-1 (no-overrun)`	51.30 (46.95 | 57.89)	1.47	41.13 (49.75 | 32.5)	3.14	40.63 (46.00 | 35.25)	3.14
`Steps-2 (no-overrun)`	57.07 (53.91 | 61.84)	4.25	46.5 (56.00 | 37.00)	5.41	40.63 (44.75 | 36.50)	5.41
`Steps-1 (overrun)`	64.40 (63.48 | 65.79)	4.86	42.25 (51.50 | 33.00)	4.99	42.38 (48.50 | 36.25)	5.08
`Steps-2 (overrun)`	66.49 (66.09 | 67.11)	5.70	45.88 (55.75 | 36.00)	5.95	42.15 (46.75 | 37.50)	6.04
`Steps-3 (overrun)`	67.01 (66.09 | 68.42)	6.79	47.13 (56.50 | 37.75)	9.09	44.13 (48.00 | 40.25)	9.07
`Steps-4 (overrun)`	68.06 (66.96 | 69.74)	8.27	49.25 (59.75 | 38.75)	10.14	**45.00** (50.25 | 39.75)	10.10
`Steps-6 (overrun)`	70.68 (70.43 | 71.05)	10.71	50.63 (62.25 | 39.00)	12.31	**45.00** (52.00 | 38.00)	12.23
`Steps-8 (overrun)`	**72.77** (72.17 | 73.68)	13.28	**51.75** (64.00 | 39.50)	14.49	44.13 (52.25 | 36.00)	14.41
LLaVA-OV-7B						
`Steps-1 (no-overrun)`	83.24 (87.82 | 76.31)	1.47	69.00 (82.25 | 55.75)	3.84	65.75 (77.00 | 54.50)	3.86
`Steps-2 (no-overrun)`	89.01 (92.17 | 84.21)	4.23	70.38 (84.00 | 56.75)	6.07	66.88 (78.00 | 55.75)	6.09
`Steps-1 (overrun)`	90.57 (93.04 | 86.84)	4.05	69.88 (85.75 | 54.00)	6.55	68.75 (81.00 | 56.50)	7.35
`Steps-2 (overrun)`	91.62 (93.93 | 88.15)	5.16	70.25 (86.50 | 54.00)	7.41	67.63 (79.00 | 56.25)	8.06
`Steps-3 (overrun)`	91.62 (93.91 | 88.15)	6.37	70.00 (85.75 | 54.25)	8.34	66.88 (78.00 | 55.75)	8.93
`Steps-4 (overrun)`	**92.15** (93.91 | 89.47)	7.63	70.75 (85.75 | 55.75)	9.32	68.38 (80.75 | 56.00)	9.86
`Steps-6 (overrun)`	**92.15** (93.91 | 89.47)	10.22	70.63 (85.75 | 55.50)	11.33	68.88 (82.50 | 55.25)	11.81
`Steps-8 (overrun)`	91.62 (93.04 | 89.47)	12.84	**71.13** (86.75 | 55.50)	13.41	**69.63** (83.50 | 55.75)	13.83

Table 11: **Detailed results on the MME-RealWorld-Lite dataset.** Accuracy is reported both per subtask and on average per subset. The highest accuracy for each subtask is highlighted in bold.

		Task	*LLaVA-OV-7B*		
			Vanilla **Acc.** [%]	**ZoomEye** **Acc.** [%]	FOCUS **(Ours)** **Acc.** [%]
Perception	AD	$\texttt{Motion}_{\texttt{multi-pedestrians}}$	22.00	28.00	**34.00**
		$\texttt{Motion}_{\texttt{multi-vehicles}}$	**46.00**	38.00	40.00
		$\texttt{Motion}_{\texttt{pedestrian}}$	24.00	**46.00**	44.00
		$\texttt{Motion}_{\texttt{vehicle}}$	24.00	**54.00**	34.00
		$\texttt{Object}_{\texttt{count}}$	36.00	**42.00**	40.00
		$\texttt{Object}_{\texttt{identify}}$	**78.00**	74.00	70.00
		$\texttt{Visual}_{\texttt{traffic-signal}}$	62.00	**78.00**	68.00
	DT	Diagram	70.00	**80.00**	60.00
		Table	60.00	**68.00**	56.00
	MO	$\texttt{Person}_{\texttt{color}}$	32.00	40.00	**56.00**
		$\texttt{Person}_{\texttt{counting}}$	32.00	**38.00**	**38.00**
		$\texttt{Person}_{\texttt{orientation}}$	10.53	**15.79**	10.53
		$\texttt{Vehicle}_{\texttt{color}}$	46.00	**60.00**	52.00
		$\texttt{Vehicle}_{\texttt{counting}}$	56.00	56.00	**58.00**
		$\texttt{Vehicle}_{\texttt{location}}$	**38.00**	28.00	28.00
		$\texttt{Vehicle}_{\texttt{orientation}}$	12.00	20.00	**26.00**
	OCR	Advert & product	82.00	82.00	**88.00**
		Book map poster	**78.00**	70.00	74.00
		License	**88.00**	86.00	**88.00**
		Phone & address	82.00	**96.00**	90.00
		Text recognition	78.00	72.00	**80.00**
	RS	Color	60.00	**64.00**	**64.00**
		Count	34.00	**40.00**	20.00
		Position	**62.00**	50.00	56.00
		Average	52.01	**56.29**	54.15
Reasoning	AD	$\texttt{Attention}_{\texttt{traffic-signal}}$	**74.00**	72.00	**74.00**
		$\texttt{Intention}_{\texttt{ego}}$	**26.00**	24.00	22.00
		$\texttt{Intention}_{\texttt{pedestrian}}$	48.00	50.00	**54.00**
		$\texttt{Intention}_{\texttt{vehicle}}$	26.00	**42.00**	40.00
		$\texttt{Interaction}_{\texttt{ego-2-pedestrian}}$	20.00	**28.00**	22.00
		$\texttt{Interaction}_{\texttt{ego-2-traffic-signal}}$	28.00	**30.00**	22.00
		$\texttt{Interaction}_{\texttt{ego-2-vehicle}}$	26.00	22.00	**26.00**
		$\texttt{Interaction}_{\texttt{other-2-other}}$	10.00	**12.00**	8.00
	DT	Diagram	40.00	54.00	**58.00**
		Table	40.00	44.00	**46.00**
	MO	Calculate	42.00	46.00	**50.00**
		Intention	26.00	26.00	**42.00**
		Property	64.00	**66.00**	62.00
	OCR	Character identification	72.00	64.00	**74.00**
		Scene understanding	**72.00**	64.00	68.00
		Average	40.93	43.20	**44.53**

improves the fine-grained open-ended VQA performance, increasing accuracy from 44.50% for the vanilla LLaVA-1.5 model to 65.97%.

C.3 Analysis of hyperparameter influence

We further investigate how variations in the hyperparameters of FOCUS affect its performance. This analysis is conducted using FOCUS with LLaVA-1.5 and LLaVA-OV on V*Bench, focusing on five key hyperparameters, namely the number of anchor points (k), the ROI expansion threshold, the maximum ROI size (s_{max}), the minimal Euclidean distance between anchor points (s_{dist}) and the NMS threshold. As shown in Tab. 12, using alternative hyperparameter settings reduces accuracy by at most 4.7 pp. compared to the original configuration, demonstrating our method's low sensitivity to hyperparameters. During the sensitivity analysis, we discover some alternative hyperparameter configurations that achieve even higher accuracy than our default settings.

Table 12: **Hyperparameter analysis of FOCUS.** We assess the impact of five hyperparameters on the performance of FOCUS. † indicates the original hyperparameter.

(a) Variation of numbers of anchor points k.

Model	k	**Accuracy** [%]
LLaVA-1.5	$30^\dagger$	72.77
	$\sim \mathcal{U}(10, 50)$	72.77 ± 1.55
LLaVA-OV	$30^\dagger$	92.15
	$\sim \mathcal{U}(10, 50)$	92.03 ± 1.37

(b) Variation of ROI expansion threshold.

Model	**ROI expans. threshold**	**Accuracy** [%]
LLaVA-1.5	$0.5^\dagger$	72.77
	$\sim \mathcal{U}(0.3, 0.7)$	72.77 ± 0.00
LLaVA-OV	$0.5^\dagger$	92.15
	$\sim \mathcal{U}(0.3, 0.7)$	92.70 ± 0.25

(c) Variation of maximum ROI size s_{max}.

Model	s_{max}	**Accuracy** [%]
LLaVA-1.5	$5^\dagger$	72.77
	7	69.63
	9	68.06
LLaVA-OV	$9^\dagger$	92.15
	5	94.24
	7	91.01
	11	89.53

(d) Variation of minimum distance between ROI anchor points s_{dist}.

Model	s_{dist}	**Accuracy** [%]
LLaVA-1.5	$2^\dagger$	72.77
	3	72.25
	4	69.11
LLaVA-OV	$2^\dagger$	92.15
	3	91.62
	4	92.15
	5	91.62

(e) Variation of NMS threshold.

Model	**NMS threshold**	**Accuracy** [%]
LLaVA-1.5	$0.3^\dagger$	72.77
	0.1	70.16
	0.5	72.25
	0.7	72.25
LLaVA-OV	$0.3^\dagger$	92.15
	0.1	92.15
	0.5	92.15
	0.7	92.15

For k and the ROI expansion threshold, we randomly sample 50 values from $\mathcal{U}(10, 50)$ and $\mathcal{U}(0.3, 0.7)$, respectively, where $\mathcal{U}$ indicates a uniform distribution. Across both LLaVA-1.5 and LLaVA-OV, we observe only minor performance variations when adjusting k: an accuracy of 72.77 ± 1.55 for LLaVA-1.5 and 92.03 ± 1.37 for LLaVA-OV, as shown in Tab. 12a. For the ROI expansion threshold, we observe an even smaller impact on the performance of FOCUS, with an accuracy of 72.77 ± 0.00 for LLaVA-1.5 and 92.70 ± 0.25 for LLaVA-OV, as shown in Tab. 12b. For s_{dist}, s_{max}, and the NMS threshold, we vary the values in discrete steps and analyze their influence on performance. Across both LLaVA-1.5 and LLaVA-OV, we find that s_{max} has the largest impact on FOCUS among all analyzed hyperparameters, resulting in an accuracy drop of 4.71 pp. for LLaVA-1.5 and 2.62 pp. for LLaVA-OV, as shown in Tab. 12c. Interestingly, for FOCUS with LLaVA-OV, we observe an accuracy improvement of 2.09 pp. over the baseline when setting $s_{\text{max}} = 5$. For s_{dist}, we

observe a maximum accuracy degradation of 3.66 pp. for LLaVA-1.5 and 0.53 pp. for LLaVA-OV, as shown in Tab. 12d. For the NMS threshold, we find only a minor impact, with a maximum accuracy degradation of 2.61 pp. for LLaVA-1.5 and no observable influence for LLaVA-OV, as shown in Tab. 12e. This is likely because FOCUS with LLaVA-OV generates higher-resolution object relevance maps, reducing its reliance on NMS.

C.4 Additional efficiency metrics

We report additional efficiency metrics—including average execution time and peak GPU memory usage per sample—on V*Bench. We compare SEAL with a customized LLaVA-7B, vanilla LLaVA-1.5-7B, vanilla LLaVA-OneVision-7B, and three training-free methods (ViCrop, ZoomEye, and `FOCUS`) using LLaVA-1.5 models.

As shown in Tab. 13, among the training-free methods, ZoomEye achieves the highest accuracy but suffers from poor efficiency due to its complex confidence prediction mechanism, which involves multiple question prompts and a hierarchical tree structure. This is reflected by its high number of Forward Passes (FPs) and long execution times. `FOCUS`, by contrast, leverages an object relevance map built from cached token similarities to directly identify relevant image regions. As a result, it requires only 25% of ZoomEye's FPs and execution time to reach comparable accuracy. ViCrop is slightly more efficient than `FOCUS` in terms of execution time and FPs, but it achieves lower accuracy and incurs the highest peak memory usage due to its incompatibility with efficient attention mechanisms. SEAL differs architecturally from LLaVA-1.5 and LLaVA-OneVision. Its dual-MLLM design makes it significantly slower and more memory-intensive than most methods in the comparison. In general, a lower number of FPs is associated with reduced execution time.

Table 13: **Additional performance and efficiency metrics on V*Bench.** In the last three rows, the best-performing method is highlighted in bold and the runner-up is underlined.

Model	**Acc.** $[\%]\uparrow$	**FP** $[\#]\downarrow$	**Avg. execution time** $[s]\downarrow$	**Avg. peak memory** $[GB]\downarrow$
SEAL	73.68	25.53	9.16	27.34
LLaVA-OV-7B	74.46	-	1.30	19.64
LLaVA-1.5-7B	47.64	-	0.25	13.57
+ w/ ViCrop	59.16	**12.26**	**1.36**	19.93
+ w/ ZoomEye	**77.49**	48.63	11.26	**14.24**
+ w/ Ours	<u>72.77</u>	<u>13.28</u>	<u>2.19</u>	<u>14.91</u>

Additionally, we compare the efficiency of `FOCUS` and ZoomEye with LLaVA-1.5 across multiple configurations (see Tab. 14). We evaluate them on V*Bench in terms of accuracy, average inference time, average FPs and average FLOPs. The latter ones are estimated based on a calculation scheme applied in prior MLLM work [7, 39]. We ran the evaluations on the same hardware to ensure result comparability.

Notably, the lowest-complexity configuration of ZoomEye exhibits a higher inference time and nearly identical FPs and FLOPs compared to the highest-complexity configuration of `FOCUS`. Despite this, `FOCUS` outperforms ZoomEye by 22.51 pp. in accuracy under this configuration. Moreover, ZoomEye shows a steep increase in complexity—measured by inference time, FPs, and FLOPs—as the search depth in its tree structure increases. Its maximum configuration achieves an accuracy of 77.48%, with an inference time of 11.96 seconds, 48.63 FPs, and a computational cost of 217 TFLOPs.

In summary, the efficiency gains reported in terms of FPs are consistently reflected in inference time and FLOPs. `FOCUS` achieves competitive or superior accuracy with significantly lower inference time and fewer FLOPs compared to ZoomEye. For instance, at `Steps-6 (overrun)`, `FOCUS` reaches 70.68% accuracy in just 2.00 seconds and 51.26 TFLOPs, whereas ZoomEye (`Depth-2`) requires nearly 4× more FLOPs and 5× longer inference time to reach a comparable accuracy. These results underscore the efficiency of `FOCUS` for fine-grained visual reasoning tasks in practice.

C.5 Inference scheme and performance discrepancy

We describe in the following the comparison between different inference schemes and the discrepancy between reported and reproduced performance.

Table 14: **In-depth efficiency comparison of FOCUS and ZoomEye.** Across different configurations, we compare efficiency in terms of accuracy, average inference time, average FPs and average FLOPs on V*Bench.

Model	Accuracy [%] ↑	Inference time [s] ↓	FP [#] ↓	TFLOPs [#] ↓
ZoomEye				
LLaVA-1.5-7B				
`Depth-1 (2 crops)`	50.26	3.78	12.50	55.95
`Depth-1 (4 crops)`	50.78	4.76	20.37	91.03
`Depth-2 (4 crops)`	71.20	9.73	44.54	199.21
`Depth-5 (4 crops)`	77.48	11.96	48.63	217.00
FOCUS				
LLaVA-1.5-7B				
`Steps-1 (no overrun)`	51.30	0.99	1.47	10.98
`Steps-2 (no overrun)`	57.07	1.28	4.25	23.11
`Steps-1 (overrun)`	64.40	1.36	4.86	25.73
`Steps-2 (overrun)`	66.49	1.44	5.70	29.43
`Steps-3 (overrun)`	67.01	1.55	6.79	34.15
`Steps-4 (overrun)`	68.06	1.73	8.27	40.61
`Steps-6 (overrun)`	70.68	2.00	10.71	51.26
`Steps-8 (overrun)`	72.77	2.27	13.28	62.46

Table 15: **Comparison of different inference schemes across models.** Each result includes accuracy, execution time, and number of forward passes for the visual search (FP). The highest accuracy in a column is highlighted in bold.

	Vanilla	ZoomEye			FOCUS (Ours)		
Inference Scheme	**Acc.** [%]↑	**Acc.** [%]↑	**Exec. time**[s]↓	**FP** [#] ↓	**Acc.** [%]↑	**Exec. time**[s]↓	**FP** [#] ↓
Logits matching	**48.16**	**77.49**	11.26	48.63	**74.35**	2.76	13.28
Open-ended generation	47.64	72.25	9.26	36.94	72.77	2.19	13.28

Inference scheme: logits matching vs. open-ended generation Multiple-choice VQA requires selecting one of several fixed answer options. A common approach is open-ended generation [3, 15], where the prompt includes the question and options (e.g., `"(A) Red"`), and the MLLM generates the corresponding option letter. In contrast, SEAL [38] and ZoomEye [31] adopt an alternative scheme called logits matching on V*Bench [38]. In this method, the model is prompted multiple times, once for each answer option. Specifically, each answer option is reformulated as a sentence (e.g., `"(A) Red"` → `"The color of the car is red."`) which is then appended to the original question. The model is prompted with these reformulated question–option pairs and the image, and the answer option yielding the highest logit score for its target tokens is selected as the final prediction.

We noticed that ZoomEye utilizes logits matching on V*Bench but uses open-ended generation on the other three datasets, prompting us to investigate the impact of different inference schemes. We evaluate LLaVA-1.5 on V*Bench using both open-ended generation and logits matching across three methods: the vanilla baseline, ZoomEye, and FOCUS (ours). SEAL is excluded from this comparison, as it is not a training-free method. All other hyperparameters are kept constant.

As shown in Tab. 15, both the vanilla model and FOCUS achieve 0.5 pp. and 1.6 pp. higher accuracy, respectively, when using logits matching. This improvement likely stems from the fact that logits matching eliminates the need for strong instruction-following: models no longer need to explicitly generate the option letter, but instead compare the semantic content of full answer statements. This makes the inference process more robust, particularly for models with weaker generative alignment.

In contrast, ZoomEye's accuracy drops by over 5 pp. and its execution time decreases by approximately 2 seconds when switching to open-ended generation. Given that FOCUS shows a 1.6 pp. accuracy difference and a 0.57-second reduction in execution time under the same scheme change, we attribute that portion of ZoomEye's decline to the scheme itself. The remaining gap—both in accuracy and runtime—can likely be attributed to suboptimal tuning in the open-ended setting, as all hyperparameters were held constant. The changed inference mode likely alters the model confidence,

Table 16: **Reported vs. reproduced accuracy across fine-grained VQA datasets.** A dash indicates that no evaluation on that benchmark was performed in the original work.

Model	V*Bench		HRBench-4K		HRBench-8K	
	Reported Acc. [%]	**Reprod. Acc.** [%]	**Reported Acc.** [%]	**Reprod. Acc.** [%]	**Reported Acc.** [%]	**Reprod. Acc.** [%]
ZoomEye						
LLaVA-1.5-7B	83.25	77.48	53.25	50.00	50.75	49.00
LLaVA-OV-7B	90.58	91.10	69.63	69.38	69.25	69.00
ViCrop						
LLaVA-1.5-7B						
`rel-att-high`	62.30	59.16	–	42.50	–	39.38
`grad-att-high`	57.07	54.97	–	44.25	–	38.38
SEAL	75.39	73.68	–	34.50	–	33.50

which may lead to premature termination of tree search (indicated by the lower number of FPs) and increased prediction instability.

Despite the potential accuracy gains of logits matching, we adopt open-ended generation for `FOCUS` across all datasets. This is done for two reasons. First, the effectiveness of logits matching depends heavily on the quality of option reformulations, which are not always available or consistent across datasets. This limits its generalizability and makes cross-dataset comparisons less reliable. Second, logits matching requires one Forward Pass (FP) per answer option, compared to only a single FP in open-ended generation. Although some acceleration of the logits matching scheme is implemented by caching question tokens, it still increases inference time (see Tab. 15). Therefore, we report `FOCUS` results using open-ended generation in the main paper. For SEAL and ZoomEye, we preserve their original inference schemes. Their respectively reported V*Bench performance is based on logits matching.

Discrepancy between reported and reproduced performance For the baseline methods, i.e., SEAL [38], ViCrop [42], and ZoomEye [31], we use the official implementations provided in their respective repositories. We strictly follow the original configurations, including software environments and data structures, as specified by each work. As shown in Tab. 16, we observe some discrepancies between the reported and reproduced performance, most notably for LLaVA-1.5 with ZoomEye. We hypothesized this may be linked to the use of different efficient attention backends (e.g., FlashAttention-2 [10] vs. PyTorch's SDPA[2]). However, the observed deviation with different attention implementations is small, less than 1% on V*Bench—thus, further root causes of this deviation seem to exist, that however remain unclear. To ensure fair and consistent comparisons under a unified evaluation setup, we always report the reproduced results for these benchmark methods in the main paper.

D Further qualitative examples

This section provides additional qualitative examples that highlight both the strengths and limitations of `FOCUS` when applied to LLaVA-1.5 (see Fig. 9) and LLaVA-OneVision (see Fig. 10). For improved clarity in visualization, we use a reduced $k = 10$, differing from the values specified in App. A.3. Further, we provide qualitative examples for failure cases of `FOCUS` with LLaVA-1.5 on high-resolution images in Fig. 8 to highlight the limitations of low-resolution object relevance maps (see Sec. 4.6). In these examples, we use an increased $k = 50$, enabling `FOCUS` to cover a larger portion of the image space; nevertheless, `FOCUS` still fails to identify the relevant image region.

Fig. 9 **(I)** showcases a type-1 single-target task with `FOCUS` and LLaVA-1.5. By leveraging the MLLM's internal representations, `FOCUS` identifies a relevant crop that highlights the color of small candles, correcting the model's initial VQA response. The ROI ranking mechanism demonstrates robustness to noise in the object relevance map by assigning the highest confidence to the originally

[2]See `https://docs.pytorch.org/tutorials/intermediate/scaled_dot_product_attention_tutorial.html`

fourth-ranked region. Moving to a type-1 multiple-target task, Fig. 9 **(II)** illustrates how FOCUS identifies both the `person in the red jacket` and the `large tree` using in-context learning. The object relevance maps clearly localize both targets. Since LLaVA-1.5 accepts only single-image inputs, the selected ROIs are stitched together—a strategy detailed in App. A.5—to avoid excessive image size or irrelevant content. Despite this, the model fails to answer correctly, likely due to limited spatial reasoning. A type-2 counting task is depicted in Fig. 9 **(III)**. Here, FOCUS successfully locates both chairs in the image. As in the previous example, the regions are combined to form a single input for LLaVA-1.5. This enables the model to correctly answer the VQA query, which it could not do using the global view. Fig. 9 **(IV)**, finally, presents another failure case with LLaVA-1.5, underscoring the limitation discussed in Sec. 4.6. The example, drawn from the HRBench-8K dataset (see Tab. 8), involves a high-resolution image where the sign is too small to be detected via the low-resolution object relevance map. Consequently, FOCUS selects an incorrect region and fails to improve the VQA result.

Turning to LLaVA-OneVision, Fig. 10 **(I)** features a type-1 single-target task. While vanilla LLaVA-OneVision fails to answer the question about the speed limit sign, FOCUS successfully identifies the relevant region using internal representations. By isolating this region (see selected ROI), the model is able to generate the correct VQA response. Fig. 10 **(II)** explores a type-1 multiple-target task. Despite the relatively large size of the relevant regions, vanilla LLaVA-OneVision does not answer correctly. FOCUS identifies the appropriate areas, generates a combined image region, and creates one local crop per relevant object. The relevant regions are highlighted in the image of the combined ROIs with rectangles, helping reduce background noise and enabling the model to answer the VQA task correctly. Next, a type-2 counting task is shown in Fig. 9 **(III)** with LLaVA-OneVision. Vanilla LLaVA-OneVision fails to count the number of computers accurately. FOCUS identifies five computers in total, assisting the model in producing the correct answer. However, it only detects four correctly—missing one and mistakenly counting another one twice. Finally, Fig. 9 **(IV)** illustrates a failure case with LLaVA-OneVision. Despite access to a high-resolution object relevance map, FOCUS fails to detect the region associated with the umbrella. As a result, it does not provide the necessary input for the MLLM to answer the VQA example correctly.

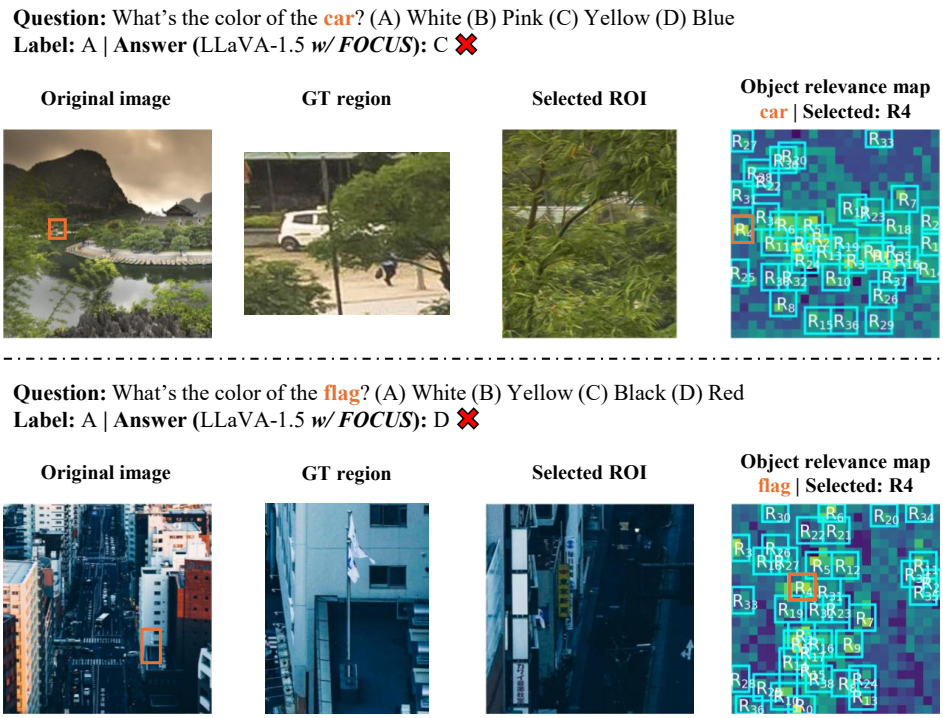

Figure 8: **Further failure cases of FOCUS with LLaVA-1.5.** We provide examples for failure cases of FOCUS using LLaVA-1.5 corresponding to the resolution limitation mentioned in Sec. 4.6. Note that we manually highlight the relevant regions in the original image to facilitate easier localization of the ground truth area for the reader and these annotations are not included in the input for the MLLM.

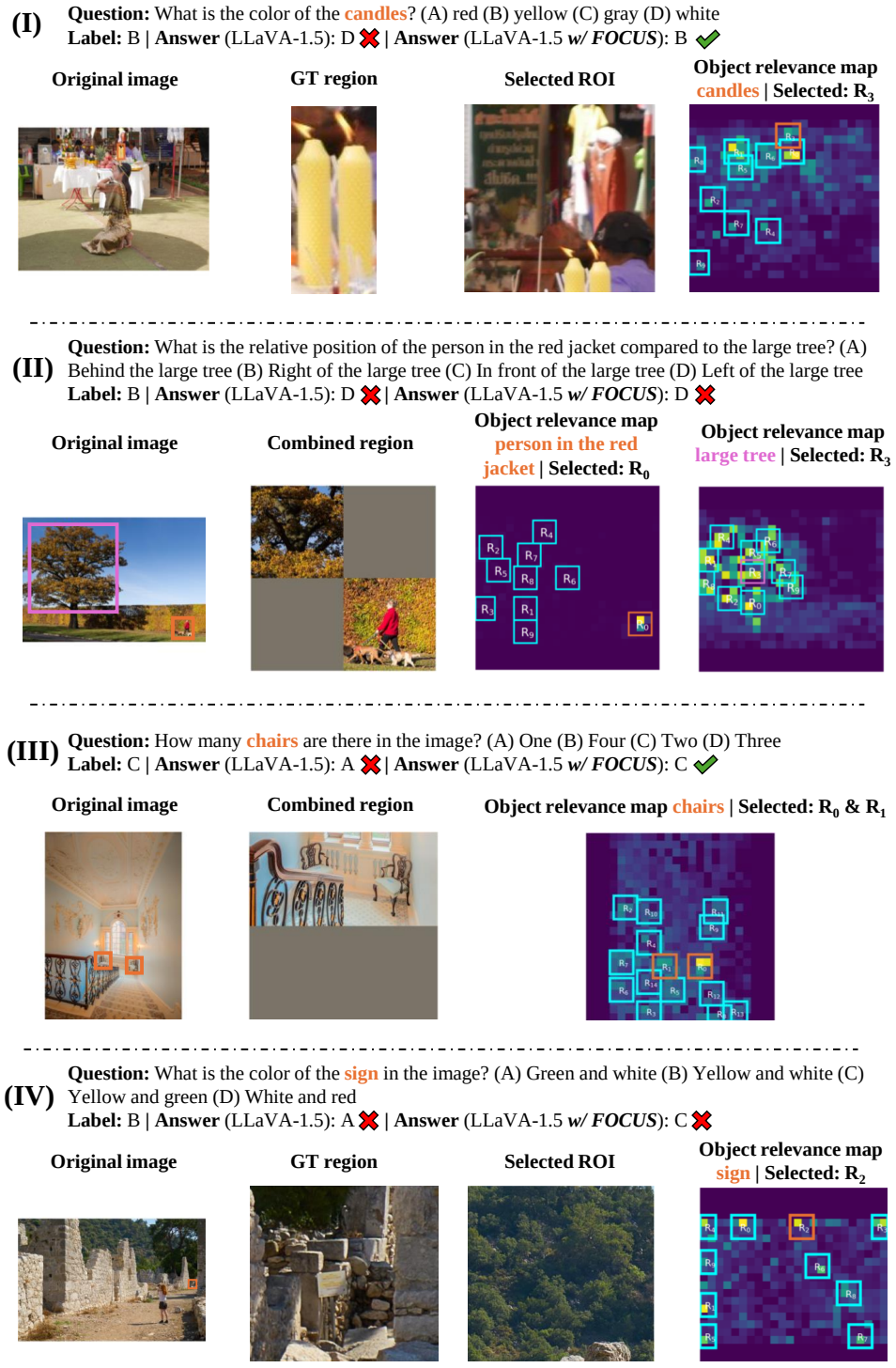

Figure 9: **Further qualitative examples of FOCUS with LLaVA-1.5.** We provide examples for single-object **(I)**, multi-object **(II)**, a type-2 question **(III)**, and a failure case **(IV)**. Note that we do not adjust the aspect ratio of the images for LLaVA-1.5. Therefore, there are some padding areas in the object relevance maps. Additionally, we manually highlight the relevant regions in the original image to facilitate easier localization of the ground truth area for the reader and these annotations are not included in the input for the MLLM.

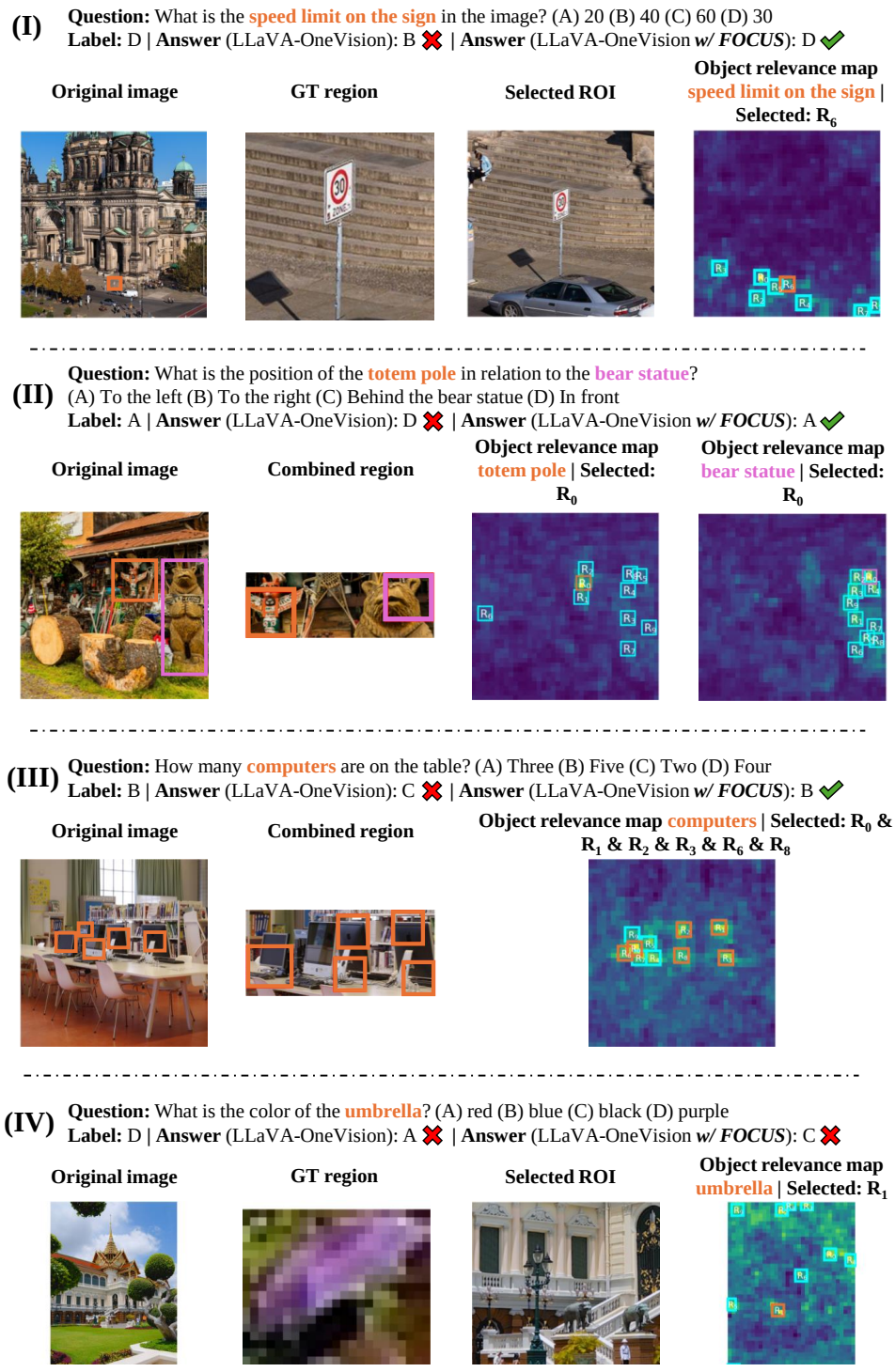

Figure 10: **Further qualitative examples of FOCUS with LLaVA-OneVision.** We provide examples for single-object **(I)**, multi-object **(II)**, a type-2 question **(III)**, and a failure case **(IV)**. Note that we manually highlight the relevant regions in the original image to facilitate easier localization of the ground truth area for the reader and these annotations are not included in the input for the MLLM.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The paper substantiates its main claims by demonstrating that internal representations of MLLMs can be effectively leveraged to guide visual cropping, enabling both efficient and high-performing inference on fine-grained VQA tasks. In Sec. 1 and Sec. 2, we identify key limitations of existing visual cropping approaches. In Sec. 3, we introduce our method, which uses cosine similarity between cached text and image token representations to construct an object relevance map and propose relevant regions. Finally, in Sec. 4, we present a large-scale empirical evaluation across four datasets and three types of MLLMs, demonstrating substantial improvements in computational efficiency and competitive or superior accuracy compared to prior methods.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss limitations of our work in a dedicated paragraph in Sec. 4.6.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best

judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

   Justification: Our work does not include theoretical results.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: In Sec. 3, we fully describe our efficient visual cropping method `FOCUS`. We list experimental setups in Sec. 4.1 and in App. A.3.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: All datasets and MLLMs are publicly available. We are waiting for internal clearance to release our code upon acceptance.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We list all experimental setups in Sec. 4.1 and App. A.3.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: In the MLLM domain, many evaluation setups—especially those using deterministic decoding—exhibit low variability. In our case, all experiments are conducted using pre-trained MLLMs in evaluation mode without stochastic components such as sampling or RAG. As such, our method produces consistent results across runs, and there is no randomness that would necessitate reporting error bars or confidence intervals.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We report the utilized computation hardware, including GPU hours for the experiments in App. A.1.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We do not conduct any research involving human subjects or participants. Further, we follow all aspects listed in the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: Our main contribution is improving the efficiency of MLLMs in perceiving and reasoning over fine visual details. This directly reduces energy and resource utilization. We hope that our efficient visual cropping method will contribute to a positive environmental impact, as discussed in Sec. 5.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: Our work does not pose significant risks of misuse. We rely exclusively on publicly available pre-trained MLLMs and standard benchmark datasets, and we do not perform any task-specific fine-tuning or release new models or data. Thus, there is no need for additional safeguards.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: All external assets used in this work—including datasets, models, and code—are properly cited with references to their original publications. We have ensured that all assets are used in accordance with their respective licenses and terms of use. We relied solely on publicly available resources that permit academic usage, and no proprietary or restricted components were used.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [NA]

Justification: We are still waiting for clearance to publicly release our code (see above) which we documented well. All of our code is written by the authors of this paper.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: Our work does not involve crowdsourcing and research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: Our work does not involve research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: Our work uses the internal representations of freely available pre-trained models and uses the MLLMs for VQA tasks.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.