Universiteit Leiden

EEG-based ROI Selection for Accelerating Vision-Language Models in Visual Question Answering

Yihui Peng (s3985571)

Supervisors:
Qinyu Chen & Guorui Lu

Bachelor Thesis– Data Science and Artificial Intelligence

Leiden Institute of Advanced Computer Science (LIACS)
liacs.leidenuniv.nl

July 8, 2026

Abstract

Vision-language models (VLMs) are increasingly used for visual question answering (VQA), but processing a full cluttered image can introduce redundant visual tokens and unnecessary inference latency. In many VQA cases, the answer depends on a specific object or local visual region rather than the entire scene. This thesis investigates whether electroencephalography (EEG) can serve as a semantic cue to select a smaller region of interest (ROI) before VLM inference, thereby reducing VQA cost while preserving answer accuracy.

The proposed framework is a three-stage EEG-to-ROI-to-VQA pipeline. Stage 1 predicts the perceived target category from EEG as an ImageNet WordNet ID (WNID) using frequency-domain differential entropy features and a multilayer perceptron (MLP), which was reproduced from the EEG-ImageNet coarse-grained benchmark. Stage 2 uses the predicted WNID as a semantic condition for YOLO-based target localization in a cluttered image. Two confidence gates control the visual input to Stage 3. A low-confidence Stage 1 prediction skips Stage 2 and triggers full-image VQA. When Stage 1 passes, the framework uses a crop only if a matching Stage 2 detection also passes its confidence gate; otherwise, it falls back to the full image. Thus, EEG is not treated as a direct spatial pointer. Instead, EEG provides semantic target information, while computer vision performs spatial localization.

Experiments are organized under 10-class, 20-class, 30-class, and 40-class settings and distinguish between module-level and end-to-end evidence. For Stage 1, we report the accuracy obtained after reproducing the EEG-ImageNet method and conducting group optimization. For Stage 2, we report oracle localization with ground-truth WNIDs and end-to-end localization with Stage 1 predictions. The final downstream evaluation uses five VQA accuracy metrics: `full_all_accuracy`, `full_detected_accuracy`, `crop_detected_accuracy`, `crop_gated_accuracy`, and `framework_accuracy`. Efficiency is evaluated through matched input-token counts and actual GPU-profiled floating-point operations (FLOPs).

Completed results show that scene composition affects the practical value of ROI guidance. In the generated-test domain, the confidence-gated Stage 3 framework improves VQA accuracy by 4.14 to 9.87 percentage points over full-image VQA while reducing total tokens by 23.2 to 39.3% and total framework FLOPs by 23.2 to 39.5%. In the real-image-test domain, full-image VQA is already highly accurate and token-light. The framework approximately preserves accuracy, with changes from -0.40 to +0.51 percentage points, while still reducing total framework FLOPs by 3.1 to 6.5%. Thus, profiled operation count decreases in all 24 model, class, and domain settings. These results show that EEG-based ROI selection is most useful in complex multi-object scenes, which better reflect the intended everyday application setting.

Contents

1 Introduction

1.1 Background and motivation

Vision-language models (VLMs) have become an important foundation for multimodal artificial intelligence. By jointly processing visual and textual information, these models can perform tasks such as image captioning, visual question answering, multimodal dialogue, and grounded visual reasoning. Visual question answering (VQA), introduced as a task in which a system answers natural-language questions about images, is especially representative because it requires both visual perception and language-based reasoning [2]. More recent VLMs and multimodal large language models, such as BLIP-2 and LLaVA, further demonstrate that connecting visual encoders with large language models can enable strong general-purpose visual instruction following and multimodal reasoning capabilities [12, 15].

Despite this progress, the computational cost of VLM inference remains a major limitation. Most modern VLMs first encode an image into visual tokens or visual features before combining them with the language input. In transformer-based architectures, the number of tokens directly affects memory usage and computation, especially in attention-based processing [5]. This issue becomes more severe when the input image is visually cluttered or contains many irrelevant background regions. Although a VQA question often depends on only one object or a small local region, the full image is typically still encoded and processed. This creates redundant visual computation and increases inference latency, which is problematic for real-time applications, resource-constrained deployment, and API-based systems where token usage is directly related to cost.

Recent work on efficient VLM inference has therefore investigated methods for reducing unnecessary visual computation. Visual token sparsification and token pruning methods aim to remove redundant visual tokens while preserving task-relevant information [28]. Efficient vision encoding methods, such as FastVLM, reduce the number of visual tokens generated by the visual encoder and improve the latency-accuracy trade-off for high-resolution image understanding [24]. These studies indicate that reducing irrelevant visual information can be an effective way to accelerate VLM inference. However, many existing approaches perform token selection inside the model or require model-specific architectural changes. In contrast, this thesis studies an external preprocessing strategy: selecting and cropping a task-relevant region before the image is passed to the VLM.

A natural way to reduce visual input size is to identify a region of interest (ROI) before inference. If the relevant object or image region can be selected in advance, the downstream VLM can process a smaller image crop instead of the full cluttered image. This may reduce token consumption and inference latency while also improving or maintaining answer accuracy, because the cropped ROI removes irrelevant background regions and focuses the VLM on the visual evidence most likely to be needed for the question. Related work has explored gaze-guided VLM inference, where human eye gaze is used as a signal for allocating visual computation to relevant image regions [26]. Gaze is useful because it provides a spatially explicit indication of where a user is looking. Nevertheless, gaze does not always perfectly reflect the user's internal cognitive focus. Human attention can be overt, where attention is aligned with eye fixation, or covert, where attention is directed without a corresponding eye movement [17]. Therefore, gaze-based ROI selection may miss task-relevant information when fixation and cognitive relevance diverge.

Electroencephalography (EEG) provides a complementary source of information. EEG is non-invasive, relatively affordable compared with neuroimaging techniques such as functional magnetic

resonance imaging (fMRI) and magnetoencephalography (MEG), and has high temporal resolution. Previous studies have shown that EEG signals contain information related to visual perception, attention, and object recognition. Large-scale EEG visual recognition datasets show that neural responses to natural images can be used for computational modeling and visual decoding [8]. More recently, EEG-ImageNet introduced a larger EEG-image benchmark based on ImageNet stimuli and provided benchmarks for object classification and image reconstruction from EEG signals [29]. In its coarse-grained 40-class classification setting, the benchmark reports the strongest performance with a multilayer perceptron (MLP) classifier using frequency-domain EEG features. This makes the benchmark MLP method suitable for Stage 1. The goal here is not to propose a new EEG classifier, but to integrate a reproducible EEG-to-WordNet ID (WNID) classifier into an ROI localization and VQA efficiency pipeline.

These developments suggest that EEG can provide semantic information about the visual target being perceived or recognized. Unlike gaze, EEG does not directly provide a two-dimensional coordinate in the image. However, it can potentially indicate which object category the user is cognitively processing. This motivates a different approach to ROI selection: first, infer the target semantic category from the EEG, and then use an object detector to recover the corresponding spatial region in a cluttered image. In this way, EEG is used as a neural semantic cue, while computer vision is used for spatial localization.

1.2 Problem Statement

This thesis investigates EEG-based ROI selection for efficient VLM-based VQA. The central problem is whether EEG signals can help identify a target visual category and guide the extraction of a smaller visual input that preserves answer accuracy while reducing token consumption and profiled computation. The proposed approach deliberately avoids predicting bounding boxes directly from EEG alone. Instead, it decomposes the task into semantic decoding, spatial localization, and downstream VQA.

In Stage 1, the system reproduces the EEG-ImageNet coarse-grained MLP classification method and uses it as the EEG-to-WNID module. Given an EEG segment, the classifier predicts the target ImageNet category, represented as a WNID [29]. In Stage 2, a YOLO-based object detector localizes the predicted target category in a cluttered image and returns a bounding box. YOLO-based detectors are suitable for this stage because they formulate object detection as direct prediction of bounding boxes and class probabilities from the image, making them widely used for efficient detection [19, 25]. In Stage 3, the system follows a confidence-gated framework policy: it uses the selected crop only when the Stage 1 and Stage 2 confidence gates are satisfied, and otherwise falls back to the full image.

A key part of this thesis is to distinguish between oracle localization, gated localization, detected crop evaluation, and complete framework evaluation. Oracle localization gives Stage 2 the ground-truth WNID and therefore isolates detector capability. Gated localization uses the Stage 1 predicted WNID and assigns zero IoU when Stage 1 predicts the wrong category, thereby measuring semantic error propagation. In Stage 3, crop-detected accuracy evaluates the VLM only on trials where a crop is available, while framework accuracy evaluates the actual runtime policy over all logical trials, including full-image fallback when no crop is available.

1.3 Dataset Motivation

A major challenge in studying EEG-based ROI selection is the lack of directly suitable datasets. Existing EEG-image datasets are mainly designed for category-level decoding, neural representation analysis, or image reconstruction. EEG-ImageNet provides EEG recordings associated with ImageNet image stimuli and supports object classification and reconstruction benchmarks [29]. However, it does not directly provide cluttered large images with explicit bounding-box annotations that indicate where the EEG-related target object appears in a complex scene. As a result, it is not directly sufficient for evaluating EEG-based ROI localization in the context of downstream VLM inference.

To address this gap, this thesis extends an EEG-image classification setting into an EEG-based ROI localization and VQA setting. Starting from the coarse-grained EEG-ImageNet categories, we construct generated cluttered images in which the target object appears within a broader visual context and annotate the target ROI with a bounding box. We also constructed a test set based on real images and annotated bounding boxes, thereby separating the evaluation of the generated image set from that of external natural images. This separation is important because the performance of the generated images alone cannot prove their robustness to real photographs.

1.4 Research Questions

The central research question of this thesis is:

> Can EEG-based region-of-interest selection reduce the inference cost of VLM-based visual question answering while preserving or improving answer accuracy?

This question is divided into the following research questions:

RQ1: How accurately can EEG signals predict the target category under different class settings?

RQ2: How well can a YOLO-based detector localize the target object in cluttered images when provided with the ground-truth WNID?

RQ3: How well does the full EEG-to-ROI pipeline perform when the detector relies on the WNID predicted by the Stage 1 EEG classifier?

RQ4: Under the two-stage confidence-gated crop/fallback policy, can the complete framework preserve or improve VQA accuracy while reducing visual tokens and FLOPs?

RQ5: How do results differ between the controlled generated-test set and the real-image-test set?

1.5 Contributions

The main contributions of this thesis are as follows.

- **Dataset extension for EEG-based ROI localization and VQA.** This thesis extends an EEG-image classification setting into an ROI localization and VQA setting by constructing generated cluttered images, explicit target-object bounding-box annotations, a real-image-test set, and English VQA question-answer pairs aligned with target ImageNet categories.

- **A three-stage EEG-to-ROI-to-VQA framework.** The proposed framework first predicts the target WNID from EEG, then uses a YOLO-based detector to localize the corresponding object, and finally evaluates the selected visual input in a VLM-based VQA setup. The framework treats EEG as a semantic cue rather than as a direct spatial coordinate predictor.
- **A confidence-gated fallback policy for downstream VQA.** The framework skips localization after a low-confidence EEG prediction and accepts a crop only when a matching detection passes the second confidence gate. Every rejected route falls back to the full image. This avoids selecting a high-confidence box from an unrelated class.
- **Stage-level and end-to-end evaluation protocols.** The thesis separates Stage 1 reproduction, Stage 2 oracle localization, Stage 2 gated localization, and Stage 3 framework-level VQA. This makes it possible to identify whether failures come from EEG classification, object localization, crop availability, or VLM answering.
- **Evaluation of VQA accuracy-efficiency trade-offs.** The thesis defines five VQA accuracy metrics together with matched token and GPU-profiled FLOPs metrics. These metrics compare full-image VQA, crop-only diagnostics, strict crop-gated evaluation, and the complete fallback framework.
- **Paste augmentation as an ablation.** Paste-based training augmentation is evaluated as a Stage 2 ablation. Because its effect is mixed across class scales, the final downstream framework uses the LLM-only Stage 2 branch.

1.6 Thesis Structure

Section 2 reviews related work on VQA, VLM efficiency, EEG-based visual decoding, object detection, and attention-guided ROI selection. Section 3 describes the EEG data, subject selection, generated-test and real-image-test domains, bounding-box annotations, VQA question-answer set, and paste augmentation used for ablation. Section 4 presents the three-stage method and the confidence-gated crop/fallback policy. Section 5 defines the evaluation metrics and reports Stage 1, Stage 2, and Stage 3 results. Section 6 discusses the main findings, error propagation, limitations, and conclusion. Appendix A provides the supplementary stage-resolved wall-clock measurements.

2 Related Work

This thesis connects four lines of research: vision-language models and visual question answering, efficient VLM inference, EEG-based visual decoding, and ROI selection using attention or detection-based methods. Existing work has made substantial progress in each of these areas, but they are usually studied separately. VLM efficiency research mainly focuses on pruning or compressing visual tokens inside the model, while EEG-based visual decoding research mainly focuses on category classification or image reconstruction. In contrast, this thesis studies whether EEG-based semantic predictions can guide ROI localization and thereby improve the accuracy-efficiency trade-off of downstream VLM-based VQA.

2.1 Vision-Language Models and Visual Question Answering

Visual question answering (VQA) was introduced as a task that requires a model to answer natural-language questions about visual content [2]. Unlike standard image classification, VQA requires the model to identify question-relevant visual evidence and combine it with language understanding. This makes VQA a useful benchmark for evaluating grounded multimodal reasoning.

Early VQA methods were typically built by combining convolutional visual features with recurrent or attention-based language models. With the rise of transformer architectures, vision-language pre-training became a dominant paradigm. Models such as ViLBERT and LXMERT introduced transformer-based cross-modal representation learning by jointly modeling visual regions and language tokens [16, 22]. Later models such as CLIP demonstrated that large-scale image-text contrastive learning can produce transferable visual representations [18]. BLIP further unified vision-language understanding and generation through bootstrapped captioning and filtering [13], while BLIP-2 connected frozen image encoders with frozen large language models through a lightweight querying transformer [12]. More recent multimodal large language models, such as LLaVA, use visual instruction tuning to align visual encoders with large language models and support open-ended multimodal dialogue and reasoning [15].

Despite these advances, standard VLM and VQA pipelines often process the full image even when the question depends only on a local object or region. This can be inefficient in cluttered scenes, where many visual tokens may correspond to irrelevant background content. This thesis builds on VQA as the downstream evaluation task, but focuses on reducing the visual input before VLM inference by selecting a target ROI guided by EEG-based semantic information.

2.2 Efficient Vision-Language Model Inference

The computational cost of modern VLMs is closely related to the number of visual tokens generated from the input image. Transformer-based vision models divide images into patches or visual tokens, and attention-based computation scales with sequence length [5]. When high-resolution or cluttered images are processed, the number of visual tokens can become large, increasing memory usage and inference latency. This has motivated research on efficient VLM inference.

One line of work reduces computation by improving the visual encoder. FastVLM, for example, introduces an efficient high-resolution vision encoder designed to reduce the number of visual tokens and improve the accuracy-latency trade-off in VLM inference [24]. Another line of work reduces

the number of visual tokens after encoding. SparseVLM proposes visual token sparsification to remove redundant visual tokens while preserving task-relevant information [28].
These methods treat redundant visual-token processing as one source of avoidable VLM cost. However, most token-reduction methods operate inside the VLM or require modifications to the model architecture, token-selection mechanism, or inference pipeline. This thesis instead studies an external ROI selection strategy. The image is cropped before it is given to the VLM. The key question is whether the crop can be selected accurately enough to improve or maintain VQA accuracy while reducing token usage and latency.

2.3 EEG-Based Visual Decoding and Visual-Stimulus Datasets

Electroencephalography (EEG) is widely used in brain-computer interface (BCI) research because it is non-invasive, relatively low-cost, and has high temporal resolution [23]. Compared with functional magnetic resonance imaging (fMRI) or magnetoencephalography (MEG), EEG provides weaker spatial resolution and is more sensitive to noise, but it is more practical for real-time and portable neural-signal applications. These properties make EEG a suitable modality for studying rapid visual perception and user intent.
Previous work has shown that EEG signals contain information related to visual object recognition. Spampinato et al. used EEG recordings collected while subjects viewed ImageNet images for automated visual classification, demonstrating that EEG can encode category-relevant information about visual stimuli [21]. Later large-scale EEG visual recognition datasets further supported this direction. For example, Gifford et al. collected a rich EEG dataset for modeling human visual object recognition with natural images, showing that EEG responses can be used for computational modeling of visual processing [8]. EEGNet also provides an important neural-network baseline for EEG classification by introducing a compact convolutional architecture for BCI tasks [11], although it is not the final Stage 1 model used in this thesis.
Visual-stimulus datasets have also played an important role in connecting computer vision and neuroscience. ImageNet provides large-scale image labels based on the WordNet hierarchy [4], while Microsoft Common Objects in Context (MS COCO) provides object detection, segmentation, and caption annotations for complex natural scenes [14]. In neuroimaging, the Natural Scenes Dataset (NSD) provides large-scale fMRI responses to natural images and has become an important resource for linking visual neuroscience and artificial intelligence [1]. Other datasets, such as BOLD5000 and Generic Object Decoding, have also supported visual decoding and analysis [3, 9].
More recently, EEG-ImageNet extended EEG-based visual decoding by collecting EEG recordings from 16 participants exposed to 4000 ImageNet images across 80 categories, including 40 coarse-grained and 40 fine-grained categories [29]. Its benchmark reports object classification and image reconstruction results, with the multilayer perceptron (MLP) achieving the strongest average performance in the coarse-grained 40-class classification setting. This thesis uses that benchmark setting as the basis for Stage 1, reproducing the MLP-based EEG-to-WordNet ID (WNID) classifier rather than proposing a new EEG architecture.
However, existing EEG visual decoding datasets are mainly designed for category classification, representation analysis, or image reconstruction. They do not directly provide cluttered large images with bounding-box annotations specifying where the EEG-related target object appears in a complex scene. This limitation motivates the dataset extension in this thesis: we construct LLM-generated cluttered large images and ROI annotations aligned with EEG-predicted WNIDs,

thereby transforming category-level EEG decoding into a spatial ROI localization problem suitable for downstream VQA evaluation.

2.4 Object Detection and ROI Localization

Object detection aims to localize objects in images by predicting bounding boxes. Region-based detectors such as Faster R-CNN use region proposal mechanisms and have achieved strong detection performance [20]. Single-stage detectors such as YOLO formulate detection as a direct prediction problem and are widely used when efficiency is important [19]. Later YOLO variants improved real-time detection performance through architectural and training improvements [25].

This thesis uses a YOLO-based detector for Stage 2 ROI localization. The detector receives a cluttered large image together with a target WNID and outputs the bounding box coordinates of the target object. The purpose is not to detect every possible object and then perform a separate post-hoc selection step; rather, the detector is evaluated as a target-class localization module conditioned on the category predicted by Stage 1. This makes the detector a bridge between EEG-based semantic prediction and spatial ROI extraction.

To evaluate Stage 2, this thesis distinguishes between two protocols. In the oracle protocol, the detector is given the ground-truth WNID, and the resulting Oracle IoU measures how well the detector can localize the target when the semantic category is correct. In the gated end-to-end protocol, the detector relies on the WNID predicted by the Stage 1 EEG classifier. If Stage 1 predicts the wrong WNID, the localization IoU is set to zero. This Gated IoU protocol captures error propagation from EEG classification to object localization and is therefore a stricter measure of the full pipeline.

2.5 Attention, Gaze, and ROI Selection

ROI selection is closely related to visual attention. In human vision, attention determines which parts of a scene receive prioritized processing. Classic work distinguishes between overt attention, where attention is accompanied by eye movements, and covert attention, where attention shifts without a corresponding fixation change [17]. This distinction is important because gaze location does not always fully reveal the user's internal cognitive focus.

Eye-tracking datasets and gaze-guided models have been used to study visual attention and saliency. SALICON, for example, provides large-scale mouse-contingent saliency annotations on MS COCO images and has been widely used for saliency prediction [10]. More recent work has explored gaze-guided VLM inference, where gaze information is used to guide visual token allocation or improve robustness in multimodal reasoning [26]. These methods support the idea that human-derived attention signals can guide efficient visual computation.

EEG provides a different type of signal. It does not directly indicate a spatial fixation point, but it can contain semantic information about the perceived or attended object category. Therefore, EEG is complementary to gaze: gaze is spatially explicit, while EEG may be semantically informative. This thesis takes advantage of this property by using EEG to predict the target WNID and then using object detection to recover the spatial ROI. In this way, the pipeline combines neural semantic decoding with visual spatial localization.

2.6 Positioning of This Thesis

The work in this thesis differs from prior research in three main ways. First, unlike standard EEG visual decoding studies, which usually stop at category classification or image reconstruction, this thesis uses EEG-based category predictions to drive spatial ROI localization. Second, unlike most efficient VLM methods, which reduce computation through internal token pruning or architecture changes, this thesis reduces visual input externally by cropping the predicted ROI before VLM inference. Third, unlike gaze-guided ROI methods, this thesis uses EEG as the guiding signal, treating it as a neural source of semantic target information rather than a direct spatial pointer. By combining EEG-based WNID prediction, YOLO-based target localization, and VQA-based efficiency evaluation, this thesis studies a new EEG-to-ROI-to-VQA pipeline. The goal is to reduce token consumption and profiled computation while evaluating whether the EEG-guided target region can improve or maintain downstream VQA accuracy.

3 Dataset

This thesis extends the EEG-ImageNet category-decoding setting into an EEG-guided ROI localization and VQA evaluation setting. The resulting dataset components combine EEG-WNID labels, real-image-test photographs, generated cluttered images, target-object ROI annotations, and English VQA question-answer pairs. The generated and real-image test domains are reported separately and are not mixed into a single average.

3.1 Original EEG-ImageNet Data

The EEG component of this thesis is based on EEG-ImageNet, a large-scale EEG dataset collected with ImageNet visual stimuli [29]. The dataset contains EEG recordings from 16 participants exposed to 4000 images selected from ImageNet-21k [4]. The visual stimuli cover 80 object categories, with 50 images per category. These categories are divided into 40 coarse-grained categories and 40 fine-grained categories.

Each EEG sample corresponds to a participant viewing an image stimulus. The dataset provides the EEG signal, the associated image index, the granularity label, and the ImageNet WNID. The raw EEG data were collected with 62 electrodes at a sampling rate of 1000 Hz during a 0.5-second image presentation window [29]. In this thesis, all experiments use the coarse-grained category setting only. The fine-grained task is not used because the ROI localization stage is aligned with the 40 coarse target categories.

3.2 Subject Selection and Class Settings

Although the original EEG dataset contains recordings from 16 participants, we checked coarse-category data, which showed that Subject 12 does not contain a complete set of EEG recordings for one of the target WNIDs. Incomplete category coverage would make the training and evaluation data inconsistent across class settings. Therefore, Subject 12 is excluded from all experiments. The final EEG subset contains 15 valid subjects.

Experiments are conducted under 10-class, 20-class, 30-class, and 40-class settings. The 10/20/30/40 class lists are not arbitrary prefixes of the official EEG-ImageNet category list; they are the Stage-2-aligned WNID lists used consistently across EEG classification, ROI localization, and VQA evaluation. This alignment is necessary because Stage 1 predictions must correspond exactly to the categories for which cluttered images, bounding boxes, and VQA questions are available.

3.3 Real-Image-Test Domain

We first constructed a real-image-test domain to evaluate the framework on real photographs. The real-image-test set contains 40 classes, with 15 images per class, resulting in 600 images in total. Each image has one validated target-object bounding box. The annotation file was checked to ensure that all bounding boxes are non-empty, lie within the image boundaries, and match the actual image dimensions. The 10/20/30-class real-image-test subsets are obtained by filtering this 40-class set with the corresponding Stage-2-aligned WNID lists.

However, the real-image-test images also have an important limitation. Most of them are strongly centered on the target object and are less representative of complex real-life scenes. In many cases,

the image contains only the target object, and the target occupies a large proportion of the image area, as illustrated in the top row of Figure 1. The local file statistics reflect this compactness: the 600 real-image-test files occupy about 80.1 MB in total, with an average size of about 0.13 MB per image. As a result, full-image VQA is already relatively easy and token-light in this domain.
It is also difficult to collect real cluttered images that contain several possible EEG target categories at the same time. Some category combinations are rare or practically infeasible in natural photographs. For example, it is difficult to find realistic images that naturally contain a giant panda together with a German shepherd, or an elephant together with a colobus monkey in the same scene. This makes it hard to use real photographs alone to evaluate the framework under complex multi-object and cluttered conditions.
One project-specific label convention is used for WNID `n03775071`. In this thesis, both the training images and the real-image-test images operationalize this class using boxing-glove imagery. This keeps the train/test label semantics internally consistent for the project, rather than mixing boxing-glove examples with a broader or ambiguous WordNet interpretation.

3.4 LLM-Generated Cluttered Large Images

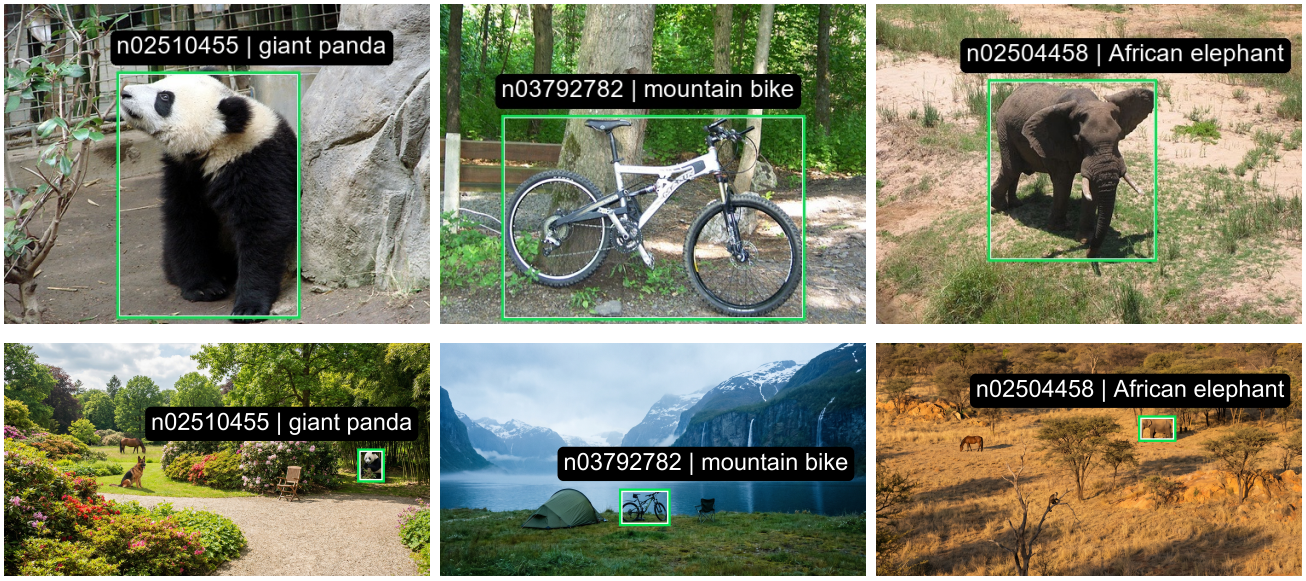

Figure 1: Representative target-object bounding-box annotations from the two image domains. The top row shows real-image-test examples, where the target object is typically centered and occupies a large fraction of the image. The bottom row shows LLM-generated cluttered examples from the same WNIDs, where the target object is embedded in a broader scene with background content and distractor objects. Green boxes indicate the target ROIs used for localization and crop extraction.

To complement the real-image-test domain, we constructed an additional generated-test domain using large language model-based image generation systems. This domain is designed to train and evaluate the framework in more complex visual scenes. Compared with the real-image-test images, the generated images contain more background content, more distractor objects, and more cluttered visual layouts, as shown in the bottom row of Figure 1. They are also much larger on

disk: the generated-image collection contains 2000 image files with an average size of about 6.74 MB. This setting is closer to complex real-life scenarios where the relevant object is embedded in a larger visual context.

The original EEG image stimuli are not designed for ROI localization in cluttered scenes. Therefore, for each target WNID, we generated large cluttered images using MiniMax and Doubao. During generation, the prompts required the images to look as realistic as possible. The goal was not to create obviously synthetic images, but to build plausible scenes in which the target object appears together with other visual elements. This was difficult to achieve by collecting web images alone. Many target combinations, such as pandas with German shepherds or elephants with capuchin monkeys, are rare in natural photographs. The generated-test domain allowed us to create controlled multi-object scenes that would be difficult to obtain from real photographs.

The generated-test domain therefore serves as a controlled complement to the real-image-test domain. It makes it possible to test whether the framework can still identify and use the correct ROI when the image contains clutter, distractors, or several visually salient objects. It is especially useful for evaluating whether EEG-based semantic guidance can help the VLM focus on the target object in complex scenes.

For each category, 50 generated cluttered images form the base image set used for Stage 2 training, validation, testing, and generated-domain Stage 3 evaluation. Each generated image is paired with a bounding-box annotation indicating the target object. The bounding box defines the ROI that should be extracted when the target WNID is correctly identified. These annotations are converted into YOLO format for Stage 2 training, and they are also used as ground truth for calculating localization metrics such as IoU, oracle IoU, and gated IoU. Pre-gate crop availability and confidence-gated crop acceptance are reported separately.

3.5 VQA Question-Answer Dataset

To evaluate the downstream effect of ROI selection, an English VQA question-answer set is constructed for the 40 target classes. For each class, 15 test images are used, and each image is paired with one manually defined question-answer item. Therefore, each class contributes 15 image-question pairs. Each question is designed to be answerable from the target object or its immediate visual evidence. The same question is evaluated on both the full image and the selected crop, allowing paired comparison when a crop is available.

The main protocol uses one pre-matched question per test image, rather than applying all questions of a class to every image. Since the evaluation is performed at the subject level, each image-question pair is combined with the EEG-derived prediction from each valid subject. Therefore, the number of subject-level logical trials is computed as:

$$N_{\text{logical}} = C \times I \times S, \tag{1}$$

where C is the number of classes, $I = 15$ is the number of image-question pairs per class, and $S = 15$ is the number of valid subjects after excluding the incomplete subject. Table 1 summarizes the resulting logical-trial counts for different class settings.

Table 1: Dataset composition and logical-trial counts under different class settings. The generated image set is reported as train/validation/test counts.

Setting	Generated-image-set	Real-img-set	EEG	Question pairs	Valid subjects	Logical trials
10-class	300/50/150	150	500	150	15	2250
20-class	600/100/300	300	1000	300	15	4500
30-class	900/150/450	450	1500	450	15	6750
40-class	1200/200/600	600	2000	600	15	9000

3.6 Paste-Augmented Training Set for Stage 2

Because the generated localization dataset contains a limited number of cluttered images per class, paste-based augmentation was evaluated as an additional Stage 2 training variant. The idea is inspired by copy-paste augmentation, where foreground objects are inserted into new image contexts to increase training diversity [7]. Related regional mixing strategies, such as CutMix, also show that cut-and-paste operations can encourage localizable visual evidence [27].

In this thesis, paste augmentation is treated as a Stage 2 ablation rather than as part of the final downstream Stage 3 pipeline. Preliminary Stage 2 ablation results show that paste augmentation does not provide a consistent localization improvement across the 10, 20, 30, and 40-class settings. In particular, it improves the gated localization result in the 30-class setting, but provides little or negative gain in the other settings. This suggests that the additional pasted samples may increase training diversity, but may also introduce synthetic-domain artifacts and context mismatch. Therefore, to avoid conflating the proposed EEG-guided ROI framework with an unstable auxiliary augmentation factor, the final Stage 3 experiments use the LLM-only Stage 2 branch. The paste-augmented branch is reported separately as ablation evidence in Section 5.9.

4 Methods

This section describes the proposed EEG-based ROI selection pipeline. As illustrated in Figure 2, the method connects dataset construction, EEG-based semantic prediction, ROI localization, and downstream VQA evaluation into a single workflow. The goal is to use EEG signals to guide the selection of a task-relevant visual region from a cluttered large image, and then use the selected region as a more efficient and potentially more accurate input for downstream VLM-based VQA.

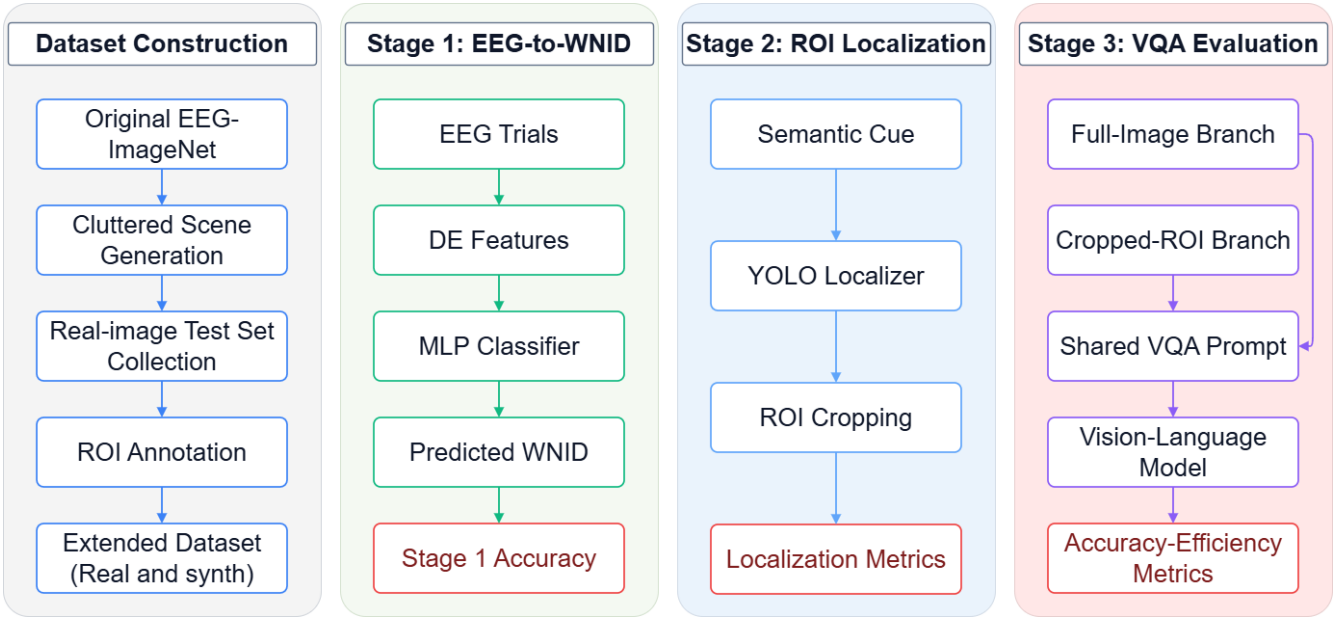

Figure 2: Overview of the revised EEG-to-ROI-to-VQA framework. Stage 1 predicts a semantic WNID from EEG and provides a confidence score. If the Stage 1 confidence is too low, the framework directly falls back to the full image. Otherwise, Stage 2 uses the predicted WNID to select a target detection. Stage 3 uses the crop only when the selected detection also passes the ROI confidence gate; otherwise, it uses the full image. Generated-test and real-image-test results are reported separately.

4.1 Stage 1: EEG-to-WNID Classification

Stage 1 predicts the target ImageNet category from EEG signals. Rather than proposing a new EEG classifier, this thesis adopts and reproduces the best-performing coarse-grained object classification method from the EEG-ImageNet benchmark [29]. The classifier input is a frequency-domain EEG feature vector extracted from each EEG segment. Differential entropy (DE) features are extracted over standard EEG frequency bands, including delta, theta, alpha, beta, and gamma bands. DE features are commonly used in EEG classification because they summarize the distributional complexity of brain activity in different frequency ranges [6].

The classifier is an MLP with two hidden layers of dimensions 256 and 128. Each linear layer is followed by batch normalization and dropout. The model is trained with cross-entropy loss to predict the WNID of the viewed stimulus. At inference time, the top-1 WNID is used as the semantic prediction, and the maximum softmax score is stored as the Stage 1 confidence score

Algorithm 1 Confidence-gated EEG-guided ROI selection for VLM-based VQA

Require: EEG signal x_{eeg}, cluttered image I, visual question q, thresholds τ_{eeg} and τ_{roi}
Ensure: Visual answer $\hat{a}$
Notation:

$\hat{w}$: predicted WNID;
$\hat{b}$: predicted bounding box;
c_{eeg} and c_{roi}: confidence scores of Stage 1 and Stage 2;
$I^{\star}$: selected visual input for Stage 3 VQA, either the full image or the cropped ROI.

$I^{\star} \leftarrow I$ ▷ Default full-image fallback
$(\hat{w}, c_{\text{eeg}}) \leftarrow f_{\text{EEG}}(x_{\text{eeg}})$ ▷ Stage 1: EEG-to-WNID classification
if $c_{\text{eeg}} \geq \tau_{\text{eeg}}$ **then**
    $(\hat{b}, c_{\text{roi}}) \leftarrow f_{\text{ROI}}(I, \hat{w})$ ▷ Stage 2: WNID-conditioned ROI localization
    if $\hat{b}$ exists **and** $c_{\text{roi}} \geq \tau_{\text{roi}}$ **then**
        $I^{\star} \leftarrow \text{Crop}(I, \hat{b})$ ▷ Use cropped ROI only when both stages are confident
    end if
end if
$\hat{a} \leftarrow \text{VLM}(I^{\star}, q)$ ▷ Stage 3: VQA
return $\hat{a}$

c_{eeg}. This raw score is a routing signal rather than a calibrated probability of correctness. The framework does not determine whether the prediction is correct at inference time, because that would require the ground-truth label.

Stage 1 is evaluated under two split protocols. The official reproduction protocol uses 30 training and 20 test EEG trials per class. However, our pipeline protocol uses 30 training, 5 validation, and 15 held-out test trials per class, so that model selection and final testing are separated for the end-to-end pipeline.

4.2 Stage 2: YOLO-Based Target ROI Localization

Stage 2 localizes the target object in a cluttered image. The input consists of an image and a semantic WNID. In oracle evaluation, the semantic WNID is the ground-truth class. In the end-to-end evaluation, the semantic WNID is the Stage 1 prediction. A YOLO-based detector first predicts candidate bounding boxes, class labels, and detector confidence scores. The Stage 2 selection step then chooses a detection whose class matches the provided WNID. If no valid matching detection is available, Stage 2 returns no selected crop. When a matching box is selected, its detector confidence is recorded as c_{roi}.

This design separates semantic target selection from spatial localization. EEG contributes the semantic cue, and YOLO contributes the image-space bounding box. The detector is trained under two conditions: `llm_only` using only generated cluttered images, and `with_paste` with paste-augmented samples added to the training set. Validation and testing remain on the original generated-image distribution. Because the paste condition produces mixed results across class scales, `llm_only` is used as the main Stage 3 downstream branch and `with_paste` is retained only as an ablation.

4.3 Confidence-Gated Crop/Fallback Policy

The final runtime framework uses a confidence-gated crop/fallback policy. The default visual input is always the full image. Stage 1 returns the top-1 WNID and its maximum softmax score c_{eeg}. Stage 2, when executed, returns the highest-confidence box whose class matches that WNID and records its detector score as c_{roi}. The two scores come from different models and are not numerically comparable probabilities.

Let $d_i = 1$ when Stage 2 returns a matching box for trial i, and $d_i = 0$ otherwise. The runtime crop-acceptance indicator is

$$g_i(\tau_{\text{eeg}}, \tau_{\text{roi}}) = \mathbf{1}[c_{\text{eeg},i} \geq \tau_{\text{eeg}}]\, d_i\, \mathbf{1}[c_{\text{roi},i} \geq \tau_{\text{roi}}]. \tag{2}$$

If $c_{\text{eeg},i} < \tau_{\text{eeg}}$, Stage 2 is skipped and the full image is sent directly to the VLM. If Stage 1 passes, Stage 2 runs, but the crop is used only when $g_i = 1$. A missing box or a detector score below τ_{roi} again triggers full-image VQA. Rejection by either gate changes the input route; it does not mark the VQA trial as incorrect.

The thresholds are fixed as $\tau_{\text{eeg}} = 0.20$ and $\tau_{\text{roi}} = 0.05$. Section 5.5 describes their validation-only selection. At test time, Equation 2 uses only model outputs. It does not use the true WNID, a ground-truth box, IoU, or a VQA answer.

The main method does not use the highest-confidence detection from an arbitrary class as a fallback. Such a fallback would add another heuristic and could route the VLM to a crop of the wrong object category. The full-image fallback is therefore used as the only fallback path.

4.4 Stage 3: VLM-Based VQA

Stage 3 evaluates whether the selected visual input supports accurate and efficient VQA. For each logical trial, the same question is asked under full-image and, when available, selected-crop conditions. Due to the limitations of computing resources and time, we chose to use version Qwen3.5-VL instead of the latest version Qwen3.6-VL. The final VLM grid is designed to cover three Qwen3.5-VL model sizes (2B, 4B, and 9B), four class scales (10/20/30/40), two test domains (generated-test and real-image-test), and 15 valid subjects.

4.5 Semantic Answer Scoring

VQA answers are scored using a two-level protocol. The first level applies deterministic automatic scoring: text normalization, exact or normalized matching, yes/no parsing, numeric tolerance where relevant, and pre-defined accepted synonyms from the VQA question-answer table. The second level is reserved for cases that the deterministic rules cannot judge reliably. These unique question cases are submitted to a stronger semantic judge, such as codex, with the question, expected answer, candidate answer, question type, answer format, and evaluation note. The judge outputs correct, incorrect, or uncertain, together with a short reason and judge version.

All semantic judgments must be persisted in CSV or JSONL files and mapped back to logical trials. The VQA model must never be shown the expected answer, and the scoring prompt must not leak the answer during generation. Before final results are reported, uncertain cases are resolved by semantic judgment whenever possible, so that the main accuracy tables contain final correct/incorrect labels rather than unresolved outcomes. Both automatically judged and semantically judged cases are audited by stratified sampling.

4.6 Summary

The method decomposes EEG-based ROI selection into semantic prediction, spatial localization, and VQA evaluation. Stage 1 predicts a WNID and confidence score from EEG features. Stage 2 runs only when Stage 1 is sufficiently confident, and the selected crop is used only when the matching detection also passes the ROI confidence gate. In all other cases, Stage 3 receives the full image. The main evaluation metrics for localization, VQA accuracy, token use, and profiled FLOPs are defined in Section 5.

5 Experiment Results

This section presents the experimental evaluation of the proposed EEG-based ROI selection pipeline. The experiments are designed to evaluate the pipeline at three levels: EEG-based WNID prediction, YOLO-based ROI localization, and VLM-based VQA accuracy, token use, and end-to-end latency.

5.1 Experimental Class Settings

Experiments are conducted under four class settings: 10, 20, 30, and 40 classes. All classes are selected from the coarse-grained category set of the EEG dataset. The 40-class setting corresponds to the full coarse-grained category set used in this thesis, while the 10-class, 20-class, and 30-class settings are used to study how the pipeline scales as the semantic search space becomes larger.
The purpose of this class-scaling design is to evaluate whether performance degradation mainly comes from EEG classification difficulty, object localization difficulty, or their interaction in the gated end-to-end pipeline. The class settings and corresponding Stage 1 classification accuracies are summarized in Table 2.

Table 2: Stage 1 EEG-to-WNID classification accuracy under the official reproduction protocol and the downstream pipeline protocol. The official protocol is used for benchmark comparability, while the 30/5/15 split supplies the held-out Stage 1 predictions used by later stages.

Setting	Subjects Used	Official split 30/20 accuracy	Pipeline 30/5/15 test accuracy
10-class	15	0.7493	0.7311
20-class	15	0.6453	0.6200
30-class	15	0.5723	0.5443
40-class	15	0.5151	0.4864

5.2 Evaluation Metrics and Reporting Protocol

This subsection defines the metrics used in the result tables. The definitions are placed in the experiment chapter because they specify how the reported numbers are computed. They do not change the method itself. All Stage 3 confidence-gated results use $\tau_{\text{eeg}} = 0.20$ and $\tau_{\text{roi}} = 0.05$, as defined in Algorithm 1 and calibrated in Section 5.5.
For Stage 2 localization, let b_i be the ground-truth target box for trial i. Let $\hat{b}_i^{\text{oracle}}$ be the box selected when the ground-truth WNID is given to the detector, and let $\hat{b}_i^{\text{pred}}$ be the box selected when the Stage 1 predicted WNID is used. If no valid box is selected, the IoU is counted as zero. Oracle IoU isolates detector quality under correct semantic input:

$$\text{OracleIoU} = \frac{1}{|N|} \sum_{i \in N} \text{IoU}(\hat{b}_i^{\text{oracle}}, b_i). \tag{3}$$

Raw IoU measures the physical overlap of the Stage-1-conditioned selected box with the ground-truth target box:

$$\text{RawIoU} = \frac{1}{|N|} \sum_{i \in N} \text{IoU}(\hat{b}_i^{\text{pred}}, b_i). \tag{4}$$

Gated IoU is the stricter end-to-end metric. Let $z_i = 1$ when Stage 1 predicts the ground-truth WNID, and $z_i = 0$ otherwise. Gated IoU is defined as:

$$\text{GatedIoU} = \frac{1}{|N|} \sum_{i \in N} z_i \, \text{IoU}(\hat{b}_i^{\text{pred}}, b_i). \tag{5}$$

This metric assigns zero localization credit when the EEG classifier sends Stage 2 to the wrong semantic category. For the Stage 2 module-level table, pre-gate crop availability is

$$\text{PreGateAvailability} = \frac{1}{|N|} \sum_{i \in N} d_i, \tag{6}$$

where $d_i = 1$ when the detector returns a box matching the supplied WNID. This quantity is measured before applying τ_{eeg} or τ_{roi}. It is therefore different from accepted-crop coverage in the confidence calibration and from the final Stage 3 crop-use rate.

Five Stage 3 accuracy metrics are used to avoid confusing crop-only performance with complete framework performance. Let $S \subseteq N$ be the confidence-gated crop-accepted subset. Let $y_{\text{full}}(i)$ be 1 when the full-image VQA answer is correct, and let $y_{\text{crop}}(i)$ be 1 when the selected-crop VQA answer is correct. The full-image baseline over all trials is:

$$\texttt{full\_all\_accuracy} = \frac{1}{|N|} \sum_{i \in N} y_{\text{full}}(i). \tag{7}$$

The paired full-image and crop accuracies on the crop-accepted subset are:

$$\texttt{full\_detected\_accuracy} = \frac{1}{|S|} \sum_{i \in S} y_{\text{full}}(i), \tag{8}$$

$$\texttt{crop\_detected\_accuracy} = \frac{1}{|S|} \sum_{i \in S} y_{\text{crop}}(i). \tag{9}$$

Crop-gated accuracy keeps all trials in the denominator and counts missing crops as zero:

$$\texttt{crop\_gated\_accuracy} = \frac{1}{|N|} \sum_{i \in S} y_{\text{crop}}(i). \tag{10}$$

The primary end-to-end metric is framework accuracy:

$$\texttt{framework\_accuracy} = \frac{1}{|N|} \left(\sum_{i \in S} y_{\text{crop}}(i) + \sum_{i \in N \setminus S} y_{\text{full}}(i) \right). \tag{11}$$

It matches the actual runtime policy: use the accepted crop when available, and use the full image as a fallback otherwise.

Token usage is reported as mean input tokens per logical VQA request. The framework token count follows the same crop/fallback route as framework accuracy.

FLOPs are reported as mean operations per logical request. One MFLOP is one million (10^6) floating-point operations, one GFLOP is one billion (10^9), and one TFLOP is one trillion (10^{12}).

The stage-level values are displayed in different units to preserve readable precision. They must therefore be converted to a common unit before summation. In TFLOPs, the conversion used for the framework total is

$$\begin{aligned} F_{\text{framework}}[\text{TFLOPs}] = 10^{-6} F_{\text{Stage 1}}[\text{MFLOPs}] \\ + 10^{-3} F_{\text{Stage 2,amortized}}[\text{GFLOPs}] + F_{\text{Stage 3}}[\text{TFLOPs}]. \end{aligned} \tag{12}$$

Let $e_i \in \{0, 1\}$ indicate whether Stage 2 executes for logical request i, let N be the number of logical requests, and let $N_{\text{exec}} = \sum_i e_i$. The executed-only Stage 2 mean is

$$F_{\text{Stage 2,executed}} = \frac{1}{N_{\text{exec}}} \sum_{i=1}^{N} e_i F_{\text{Stage 2}}(i), \tag{13}$$

whereas the amortized Stage 2 mean is

$$F_{\text{Stage 2,amortized}} = \frac{1}{N} \sum_{i=1}^{N} e_i F_{\text{Stage 2}}(i). \tag{14}$$

The executed-only value describes the cost of a YOLO call when localization actually runs. The amortized value assigns zero Stage 2 FLOPs to Stage 1 rejections and is therefore the value used in the end-to-end framework total:

$$F_{\text{framework}} = F_{\text{Stage 1}} + F_{\text{Stage 2,amortized}} + F_{\text{Stage 3}}. \tag{15}$$

Here, $F_{\text{Stage 3}}$ is the framework VLM FLOPs value after crop/full routing. We report both the VLM-only reduction

$$\text{VLMFLOPsReduction} = \frac{F_{\text{full,VLM}} - F_{\text{Stage 3}}}{F_{\text{full,VLM}}} \times 100\%, \tag{16}$$

and the total framework reduction

$$\text{TotalFLOPsReduction} = \frac{F_{\text{full,VLM}} - F_{\text{framework}}}{F_{\text{full,VLM}}} \times 100\%. \tag{17}$$

FLOPs were collected with `torch.profiler.profile` and `with_flops=True` around actual inference calls. Stage 1 profiling covered the reproduced DE+MLP forward pass, and Stage 2 profiling covered the LLM-only YOLO predictor. Stage 3 profiling covered the real VLM `generate` call. VLM requests were grouped by exact input, visual, and output token counts, then expanded using their logical-request weights. Only identical token-shape groups were duplicated within a profiler batch, and measured operations were divided by that batch size. Fixed profiler batches of 6, 2, and 1 were used for the 2B, 4B, and 9B models, respectively. Mixed-shape batching and theoretical FLOPs estimators were not used. MACs, defined as FLOPs divided by two, were retained only as audit fields.

Wall-clock latency is retained as supplementary systems evidence in Appendix A. Those measurements were collected on shared cluster GPUs and can vary with concurrent workload, memory pressure, and device utilization. The primary efficiency analysis therefore uses matched token counts and GPU-profiled FLOPs, while the appendix documents the observed stage-level runtime behavior for transparency.

5.3 Stage 1: EEG-to-WNID Classification Results

Table 2 reports Stage 1 accuracy under two protocols. The official 30/20 reproduction accuracy decreases from 0.7493 in the 10-class setting to 0.5151 in the 40-class setting. The downstream 30/5/15 pipeline accuracy decreases from 0.7311 to 0.4864 over the same class range. The two columns should be interpreted separately because the official reproduction protocol and the downstream pipeline protocol use different test compositions. The shared trend is nevertheless clear: EEG-based semantic prediction becomes harder as the number of candidate categories increases.
This Stage 1 trend is important for the rest of the pipeline because Stage 2 localization is conditioned on the Stage 1 predicted WNID when the Stage 1 confidence gate is passed. When the predicted WNID is wrong, the detector searches for the wrong semantic target. When the confidence is low, the framework falls back to the full image. Both cases reduce the number of trials in which the VLM receives an accepted ROI crop.

5.4 Stage 2: Oracle and Gated ROI Localization Results

Stage 2 is evaluated under oracle and semantic end-to-end conditions. In the oracle protocol, the ground-truth WNID is provided to YOLO, so the result isolates the detector's ability to localize the target object. In the end-to-end protocol, the detector receives the Stage 1 predicted WNID. If that WNID is wrong, gated IoU assigns zero localization credit. The runtime confidence thresholds are not applied in this module-level table.

Table 3: Stage 2 ROI localization before runtime confidence filtering. Oracle IoU uses the ground-truth WNID. End-to-end gated IoU uses the Stage 1 predicted WNID and assigns zero localization credit when Stage 1 predicts the wrong category. Pre-gate crop availability is the fraction of trials with a matching selected box before applying τ_{eeg} or τ_{roi}.

Setting	Training Data	Oracle IoU	Gated IoU	Pre-gate Crop Availability
10-class	LLM-only	0.7385	0.5385	0.6987
10-class	LLM + Paste	0.7136	0.5214	0.6520
20-class	LLM-only	0.7495	0.4639	0.5673
20-class	LLM + Paste	0.7421	0.4591	0.5916
30-class	LLM-only	0.7236	0.3904	0.4793
30-class	LLM + Paste	0.7707	0.4200	0.5360
40-class	LLM-only	0.7657	0.3689	0.4471
40-class	LLM + Paste	0.7620	0.3689	0.4576

The LLM-only oracle IoU remains high across all class settings, ranging from 0.7236 to 0.7657. This shows that the visual detector can localize the target object reliably when the semantic WNID is correct. In contrast, the LLM-only gated IoU decreases from 0.5385 in the 10-class setting to 0.3689 in the 40-class setting. The gap between oracle IoU and gated IoU, therefore, identifies Stage 1 semantic error propagation as the main limitation of end-to-end localization.
Paste augmentation has a mixed effect. It improves the 30-class setting, where gated IoU increases from 0.3904 to 0.4200, and pre-gate crop availability increases from 0.4793 to 0.5360. However, it does not improve the 10-class or 20-class gated IoU, and its 40-class gated IoU is tied with LLM-only.

Therefore, paste augmentation is not used as the final downstream branch. The remainder of this section reports the final Stage 3 results using the LLM-only branch.

5.5 Confidence-Gate Threshold Calibration

The two runtime thresholds were selected once from held-out validation records. The calibration set contains five validation trials per class for each of the 15 retained subjects. Pooling the 10-, 20-, 30-, and 40-class settings gives

$$|N_{\text{cal}}| = 15 \times 5 \times (10 + 20 + 30 + 40) = 7500 \tag{18}$$

logical calibration rows. Subject 12 is excluded, and Stage 2 uses the final LLM-only checkpoints. No generated-test or real-image-test VQA answers, token counts, latency measurements, or semantic judgments are used in threshold selection.

Calibration labels are used only to evaluate whether an accepted crop is reliable. Let $z_i = 1$ when Stage 1 predicts the ground-truth WNID, and let u_i be the IoU between the selected box and the ground-truth target box. A reliable crop is defined by

$$r_i = \mathbf{1}[z_i = 1]\, d_i\, \mathbf{1}[u_i \geq 0.5]. \tag{19}$$

For a threshold pair $(\tau_{\text{eeg}}, \tau_{\text{roi}})$, the accepted-crop indicator is the runtime indicator g_i from Equation 2. The two calibration criteria are

$$\text{Precision}_{\text{crop}} = \frac{\sum_{i \in N_{\text{cal}}} g_i r_i}{\sum_{i \in N_{\text{cal}}} g_i}, \qquad \text{Coverage}_{\text{crop}} = \frac{\sum_{i \in N_{\text{cal}}} g_i}{|N_{\text{cal}}|}. \tag{20}$$

Precision measures the reliability of crops accepted by the two gates. Coverage measures the fraction of all calibration requests that still use a crop rather than the full-image fallback.

Both thresholds are swept over $\{0.00, 0.05, \ldots, 0.95\}$, giving 400 threshold pairs. Table 4 summarizes the ungated baseline and three validation operating points. Within each precision floor, the reported pair is the one with the highest accepted-crop coverage.

Table 4: Global validation operating points for the two confidence gates. Precision and coverage are micro-averaged over 7500 validation rows. The selected 0.97 operating point is shown in bold.

Operating point	τ_{eeg}	τ_{roi}	Accepted	Reliable	Precision	Coverage
Ungated baseline	0.00	0.00	4083	3886	0.9518	0.5444
Precision ≥ 0.96	0.05	0.05	3323	3210	0.9660	0.4431
Precision ≥ 0.97	**0.20**	**0.05**	**2929**	**2842**	**0.9703**	**0.3905**
Precision ≥ 0.98	0.15	0.30	1925	1888	0.9808	0.2567

The 0.96 operating point is comparatively permissive. Its Stage 1 threshold passes all 7500 calibration rows, so the first gate does not change the route on this set. The 0.98 point is substantially stricter, but its coverage is 52.85% lower than the ungated baseline. We therefore use 0.97 as the main reliability target and maximize coverage under this constraint. The resulting pair, $\tau_{\text{eeg}} = 0.20$

and $\tau_{\text{roi}} = 0.05$, raises accepted-crop precision from 0.9518 to 0.9703 while retaining 71.74% of the baseline coverage.

At the selected point, 5989 rows pass the Stage 1 gate and 1511 skip Stage 2. Among the Stage 1-passing rows, 3600 have a matching box before the second gate. The Stage 2 threshold rejects 671 of these boxes, leaving 2929 accepted crops, of which 2842 satisfy the reliable-crop definition. These diagnostics show that both thresholds affect routing. The reported 0.9703 precision is a global micro-average over all four class settings, not a separate 97% guarantee for every class setting. Once selected, both thresholds are frozen for all final test-domain evaluations.

5.6 Stage 3: VQA Accuracy Results

Stage 3 evaluates the actual downstream question-answering behavior of the framework. Tables 5–7 report five accuracy metrics for each model size and test domain. The most important metric is `framework_accuracy`, because it corresponds to the deployed policy: use the crop only when both confidence gates accept it, and otherwise fall back to the full image. `crop_detected_accuracy` remains useful as a paired diagnostic, but it should not be interpreted as all-sample framework accuracy because it is computed only on the crop-accepted subset.

Table 5: Stage 3 VQA accuracy for Qwen3.5-2B across generated-test and real-image-test domains under the confidence-gated framework policy ($\tau_{\text{eeg}} = 0.20$, $\tau_{\text{roi}} = 0.05$). Full-detected and crop-detected accuracies are computed on the confidence-gated crop-accepted subset, whereas framework accuracy uses the actual crop/fallback policy over all logical trials.

Domain	Classes	Full-all acc. (%)	Full-detected acc. (%)	Crop-detected acc. (%)	Crop-gated acc. (%)	Framework acc. (%)
generated-test	10	69.33	73.56	97.23	40.53	79.20
	20	73.00	72.90	91.84	36.00	80.42
	30	72.00	70.85	90.83	26.40	77.81
	40	71.83	70.83	91.52	23.14	77.07
real-image-test	10	93.33	87.88	90.24	11.91	93.64
	20	92.33	88.13	92.61	10.58	92.84
	30	90.22	92.62	95.54	9.20	90.50
	40	89.67	87.63	89.61	7.57	89.83

Table 6: Stage 3 VQA accuracy for Qwen3.5-4B across generated-test and real-image-test domains under the confidence-gated framework policy ($\tau_{\text{eeg}} = 0.20$, $\tau_{\text{roi}} = 0.05$).

Domain	Classes	Full-all acc. (%)	Full-detected acc. (%)	Crop-detected acc. (%)	Crop-gated acc. (%)	Framework acc. (%)
generated-test	10	78.67	79.64	93.92	39.16	84.62
	20	79.00	77.21	94.33	36.98	85.71
	30	77.33	75.13	93.53	27.19	82.68
	40	78.00	77.90	94.29	23.84	82.14
real-image-test	10	97.33	95.62	98.65	13.02	97.73
	20	96.00	91.44	92.61	10.58	96.13
	30	94.00	97.38	96.31	9.27	93.90
	40	94.00	94.74	93.55	7.90	93.90

The generated-test results show a consistent framework accuracy gain over the full-image baseline. Across all model sizes and class settings, `framework_accuracy` is higher than `full_all_accuracy`

Table 7: Stage 3 VQA accuracy for Qwen3.5-9B across generated-test and real-image-test domains under the confidence-gated framework policy ($\tau_{\text{eeg}} = 0.20$, $\tau_{\text{roi}} = 0.05$).

Domain	Classes	Full-all acc. (%)	Full-detected acc. (%)	Crop-detected acc. (%)	Crop-gated acc. (%)	Framework acc. (%)
generated-test	10	76.67	79.00	96.27	40.13	83.87
	20	76.00	71.66	93.82	36.78	84.69
	30	74.44	70.44	94.09	27.35	81.32
	40	74.50	73.64	93.76	23.71	79.59
real-image-test	10	96.67	95.62	92.59	12.22	96.27
	20	93.67	88.33	91.25	10.42	94.00
	30	92.89	96.31	96.15	9.26	92.87
	40	93.00	93.29	91.84	7.76	92.88

by 4.14 to 9.87 percentage points. The gain is largest for Qwen3.5-2B in the 10-class setting, where accuracy increases from 69.33% to 79.20%. The best generated-test framework result is obtained by Qwen3.5-4B in the 20-class setting, with 85.71% framework accuracy.

The detected crop results explain why the framework improves on generated-test images. On the confidence-gated crop-accepted subset, `crop_detected_accuracy` remains between 90.83% and 97.23%, while `full_detected_accuracy` is lower in most settings. The crop is therefore beneficial when both confidence gates accept the route. However, crop usage decreases as the class scale increases, from 41.69% in the generated-test 10-class setting to 25.29% in the 40-class setting. This is why `framework_accuracy` is lower than `crop_detected_accuracy` but still higher than `full_all_accuracy`.

The real-image-test results show a different pattern. Full-image accuracy is already very high, ranging from 89.67% to 97.33%. Framework accuracy is therefore approximately preserved rather than clearly improved, with changes ranging from -0.40 to +0.51 percentage points. This outcome reflects the object-centric composition of the real-image-test subset rather than a failure of ROI guidance. Its targets are usually large, centered, and visually isolated, so the full-image baseline already receives a compact and highly informative input.

5.7 Stage 3: Token Usage Results

Tables 8–10 report mean input tokens per logical request. Token reduction uses the same logical-trial set and confidence-gated crop/fallback route as the accuracy evaluation.

On generated-test images, the framework consistently reduces input tokens by 23.2% to 39.3%. The reduction decreases with class count because Stage 1 rejects more requests and triggers full-image fallback more often. On real-image-test images, token reduction is smaller, ranging from 3.5% to 6.6%, because these inputs are already compact and accepted crops are less frequent. The supplementary wall-clock evaluation and its Stage 1/2/3 breakdown are reported in Appendix A.

5.8 Stage 3: FLOPs Results

Tables 11–13 report actual GPU-profiled FLOPs per logical request. Each framework entry gives the total first, followed in parentheses by Stage 1, amortized Stage 2, and Stage 3. The parenthetical components use MFLOPs, GFLOPs, and TFLOPs, respectively. These correspond to one million, one billion, and one trillion operations. The three displayed components cannot be added directly

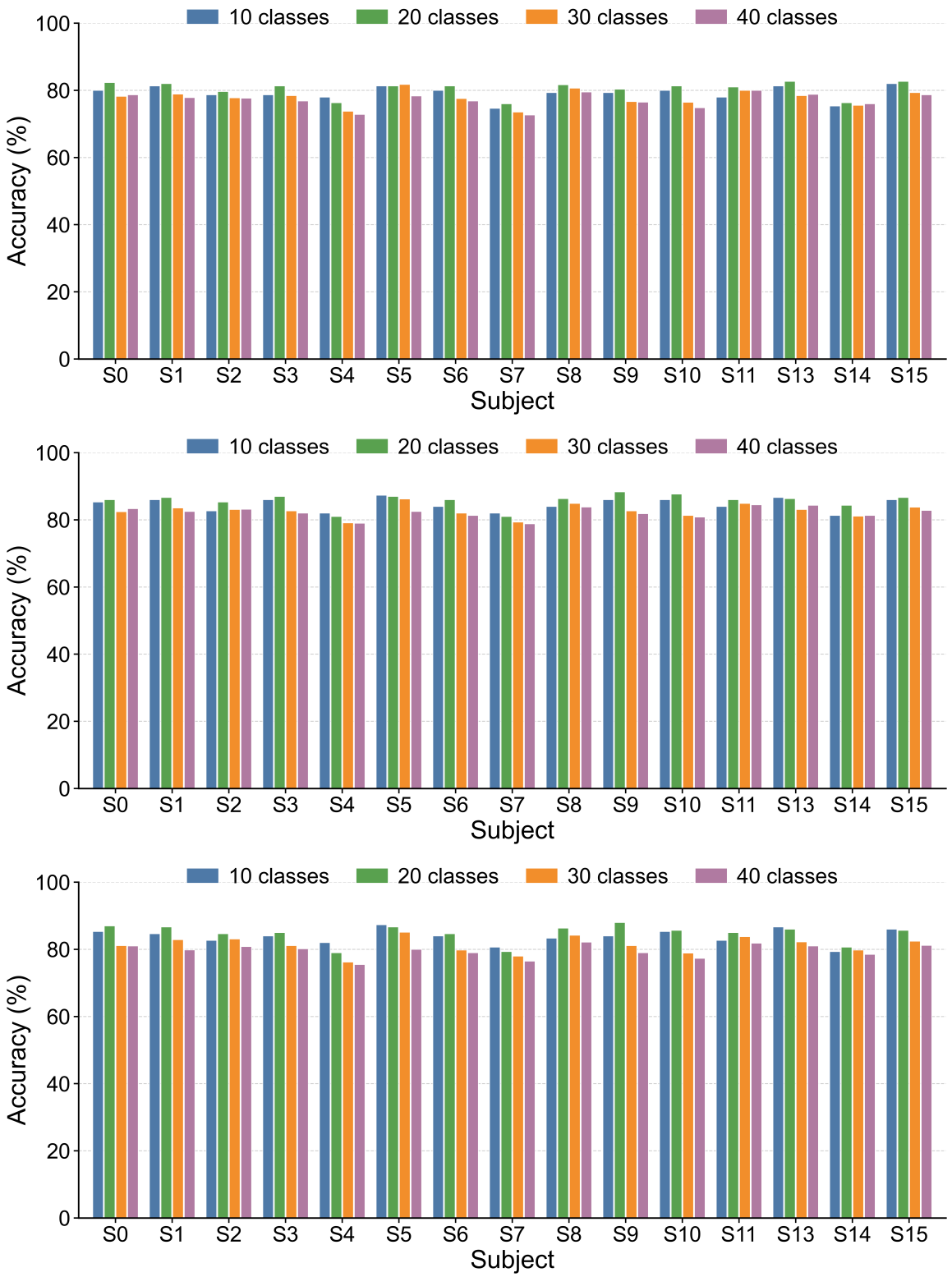

Figure 3: Subject-wise framework accuracy on generated-test images for Qwen3.5-2B, Qwen3.5-4B, and Qwen3.5-9B. Each panel compares the 10/20/30/40-class settings across the 15 valid subjects.

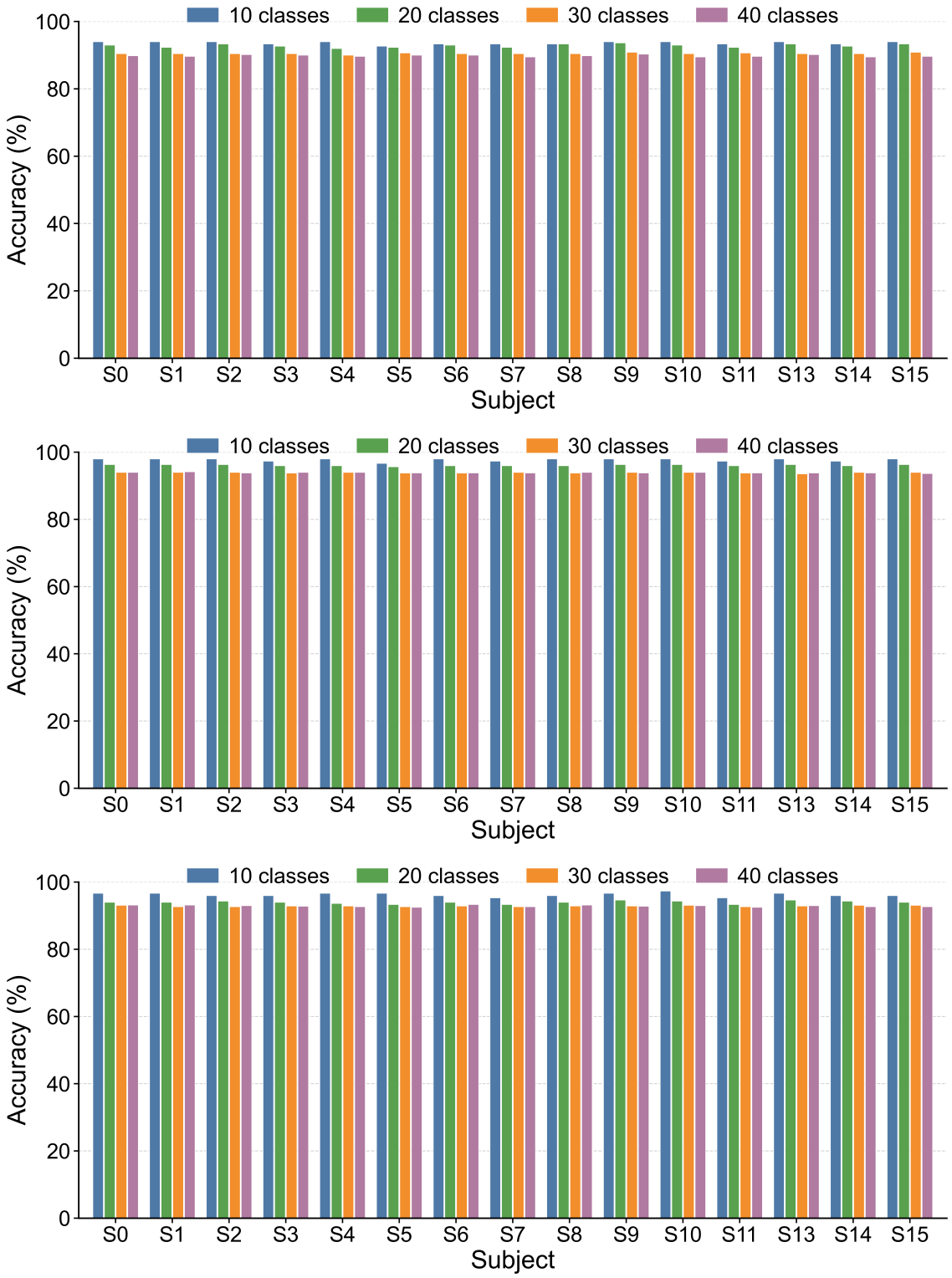

Figure 4: Subject-wise framework accuracy on real-image-test images for Qwen3.5-2B, Qwen3.5-4B, and Qwen3.5-9B. Compared with the generated-test, the real-image-test domain has higher full-image baseline performance and a lower confidence-gated crop-use rate, so framework accuracy is mainly accuracy-preserving rather than accuracy-improving.

Table 8: Stage 3 input-token efficiency for Qwen3.5-2B under the confidence-gated framework policy ($\tau_{\text{eeg}} = 0.20$, $\tau_{\text{roi}} = 0.05$).

Domain	Classes	Full input tokens	Framework input tokens	Token reduction (%)
generated-test	10	4265.7	2589.4	39.3
	20	4209.5	2704.5	35.8
	30	4202.6	3088.9	26.5
	40	4198.9	3224.5	23.2
real-image-test	10	281.7	263.1	6.6
	20	263.2	250.8	4.7
	30	384.8	371.4	3.5
	40	387.7	372.3	4.0

Table 9: Stage 3 input-token efficiency for Qwen3.5-4B under the confidence-gated framework policy ($\tau_{\text{eeg}} = 0.20$, $\tau_{\text{roi}} = 0.05$).

Domain	Classes	Full input tokens	Framework input tokens	Token reduction (%)
generated-test	10	4267.3	2589.8	39.3
	20	4211.9	2706.4	35.7
	30	4203.2	3089.4	26.5
	40	4199.5	3224.9	23.2
real-image-test	10	280.2	262.0	6.5
	20	263.1	250.6	4.7
	30	385.0	371.6	3.5
	40	387.9	372.5	4.0

Table 10: Stage 3 input-token efficiency for Qwen3.5-9B under the confidence-gated framework policy ($\tau_{\text{eeg}} = 0.20$, $\tau_{\text{roi}} = 0.05$).

Domain	Classes	Full input tokens	Framework input tokens	Token reduction (%)
generated-test	10	4265.3	2587.9	39.3
	20	4208.8	2703.5	35.8
	30	4203.2	3089.4	26.5
	40	4199.4	3224.9	23.2
real-image-test	10	279.3	260.8	6.6
	20	260.3	247.7	4.8
	30	384.9	371.4	3.5
	40	387.7	372.3	4.0

because they use different scales. Equation 12 converts them to TFLOPs before summation. VLM reduction compares full-image VLM inference with routed Stage 3. Total reduction additionally includes Stage 1 and amortized Stage 2.

Table 11: GPU-profiled FLOPs per logical request for Qwen3.5-2B under the confidence-gated framework. The framework total is followed by Stage 1, amortized Stage 2, and Stage 3 in parentheses. Their units are MFLOPs (10^6), GFLOPs (10^9), and TFLOPs (10^{12}), respectively, so the components are converted to TFLOPs before summation.

Domain	Classes	Full VLM (T)	Framework total FLOPs (T) Total (Stage 1 M/Stage 2 G/Stage 3 T)	VLM red. (%)	Total red. (%)
generated-test	10	22.340	13.523 (0.227/11.225/13.512)	39.5	39.5
	20	22.045	14.131 (0.229/10.154/14.121)	35.9	35.9
	30	22.016	16.160 (0.232/8.539/16.151)	26.6	26.6
	40	21.996	16.873 (0.234/7.786/16.865)	23.3	23.3
real-image-test	10	1.361	1.278 (0.227/15.344/1.263)	7.2	6.1
	20	1.263	1.211 (0.229/13.880/1.198)	5.2	4.1
	30	1.913	1.854 (0.232/11.672/1.842)	3.7	3.1
	40	1.928	1.858 (0.234/10.643/1.847)	4.2	3.7

Table 12: GPU-profiled FLOPs per logical request for Qwen3.5-4B under the confidence-gated framework. The framework total is followed by Stage 1, amortized Stage 2, and Stage 3 in parentheses. Their units are MFLOPs (10^6), GFLOPs (10^9), and TFLOPs (10^{12}), respectively, so the components are converted to TFLOPs before summation.

Domain	Classes	Full VLM (T)	Framework total FLOPs (T) Total (Stage 1 M/Stage 2 G/Stage 3 T)	VLM red. (%)	Total red. (%)
generated-test	10	41.484	25.139 (0.227/11.225/25.128)	39.4	39.4
	20	40.944	26.275 (0.229/10.154/26.265)	35.9	35.8
	30	40.867	30.015 (0.232/8.539/30.007)	26.6	26.6
	40	40.831	31.336 (0.234/7.786/31.328)	23.3	23.3
real-image-test	10	2.612	2.448 (0.227/15.344/2.433)	6.8	6.3
	20	2.443	2.335 (0.229/13.880/2.321)	5.0	4.4
	30	3.639	3.520 (0.232/11.672/3.508)	3.6	3.3
	40	3.668	3.528 (0.234/10.643/3.517)	4.1	3.8

Table 13: GPU-profiled FLOPs per logical request for Qwen3.5-9B under the confidence-gated framework. The framework total is followed by Stage 1, amortized Stage 2, and Stage 3 in parentheses. Their units are MFLOPs (10^6), GFLOPs (10^9), and TFLOPs (10^{12}), respectively, so the components are converted to TFLOPs before summation.

Domain	Classes	Full VLM (T)	Framework total FLOPs (T) Total (Stage 1 M/Stage 2 G/Stage 3 T)	VLM red. (%)	Total red. (%)
generated-test	10	73.826	44.737 (0.227/11.225/44.726)	39.4	39.4
	20	72.848	46.744 (0.229/10.154/46.734)	35.8	35.8
	30	72.755	53.441 (0.232/8.539/53.432)	26.6	26.5
	40	72.690	55.791 (0.234/7.786/55.783)	23.3	23.2
real-image-test	10	4.678	4.372 (0.227/15.344/4.356)	6.9	6.5
	20	4.350	4.145 (0.229/13.880/4.131)	5.0	4.7
	30	6.515	6.294 (0.232/11.672/6.282)	3.6	3.4
	40	6.565	6.308 (0.234/10.643/6.298)	4.1	3.9

Total framework FLOPs decrease in all 24 model, class, and domain combinations. On generated-test images, the reduction ranges from 23.2% to 39.5%. The front-end stages change the VLM-only reduction by at most 0.05 percentage points because their operation counts are small relative to Stage 3. Absolute savings increase with model size. They range from 5.12–8.82 TFLOPs for 2B, 9.50–16.35 TFLOPs for 4B, and 16.90–29.09 TFLOPs for 9B per logical request.

On real-image-test images, total FLOPs also decrease in every setting, but the reduction is smaller at 3.1% to 6.5%. Stage 1 requires only 0.227–0.234 MFLOPs, while amortized Stage 2 requires 7.786–15.344 GFLOPs. Stage 3 remains dominant at 1.198–55.783 TFLOPs across the complete experiment. Stage 1 and Stage 2 together account for 0.014–0.083% of framework FLOPs on generated-test and 0.169–1.201% on real-image-test.

The operation-count result is uniformly favorable: the routed framework executes fewer profiled operations in every setting.

5.9 Ablation Study

The main ablation in this thesis is the Stage 2 comparison between LLM-only and paste-augmented detector training. As shown in Table 3, paste augmentation is not consistently beneficial. It improves the 30-class localization result but not the 10/20-class results, and it ties the 40-class gated IoU. Therefore, paste augmentation is not used as the final Stage 3 branch.

A second design choice is the fallback policy. The framework uses a crop only when both confidence gates pass; every rejected route uses the full image. A top-confidence crop from an arbitrary YOLO class is not used in the main results because it would add an additional heuristic and could route the VLM to an object category that does not match the EEG-predicted target. This keeps the reported `framework_accuracy` directly aligned with the method definition in Algorithm 1.

6 Conclusions and Discussion

This section interprets the final experimental results of the EEG-based ROI selection framework. The results support a conditional conclusion. EEG-guided ROI selection can improve VQA accuracy while reducing tokens and profiled operations in complex multi-object scenes. The benefit is smaller when the input already isolates the target, as in the real-image-test domain.

6.1 Discussion of Stage 1 EEG Classification

Stage 1 acts as the semantic entry point of the pipeline. Its role is to infer the target ImageNet WNID from EEG features and provide this predicted category to the downstream detector. The official 30/20 reproduction accuracy decreases from 0.7493 in the 10-class setting to 0.5151 in the 40-class setting. Under the downstream 30/5/15 protocol used by the final pipeline, accuracy decreases from 0.7311 to 0.4864. These results show that EEG provides a meaningful semantic cue, but the cue becomes less reliable as the number of candidate categories increases.

This trend has a direct effect on the rest of the system. Stage 2 can only search for the correct target object when Stage 1 predicts the correct WNID. Therefore, Stage 1 errors reduce gated IoU, pre-gate crop availability, and the accepted-crop coverage after confidence filtering. Improving the EEG classifier would likely improve the complete framework more than further optimizing the detector under oracle WNID input.

6.2 Discussion of Stage 2 ROI Localization

Stage 2 converts the semantic WNID cue into a spatial bounding box. The oracle localization results show that YOLO has strong spatial localization ability when the target WNID is correct. In the LLM-only branch used for final Stage 3, oracle IoU remains between 0.7236 and 0.7657 across 10/20/30/40 class settings. The detector is therefore not the primary bottleneck under correct semantic input.

The gated results are lower because they include Stage 1 error propagation. LLM-only gated IoU decreases from 0.5385 in the 10-class setting to 0.3689 in the 40-class setting, while pre-gate crop availability decreases from 0.6987 to 0.4471. The final confidence gates reduce the crop-used subset further. This explains why crop-detected VQA accuracy can be high while framework accuracy is lower: the crop is often helpful when accepted, but many logical trials follow the full-image fallback route.

Paste augmentation remains a mixed ablation rather than a final method component. It improves the 30-class setting but not the 10/20-class settings, and it does not improve 40-class gated IoU. The final downstream choice of the LLM-only branch is therefore justified because it avoids adding an unstable augmentation factor to the main Stage 3 conclusion.

6.3 Discussion of VQA Accuracy Across Visual Settings

The generated-test results demonstrate the main advantage of the proposed framework. For all model sizes and class settings, framework accuracy is higher than full-all accuracy. Under the confidence-gated policy, the improvement ranges from +4.14 to +9.87 percentage points. This means that, in generated cluttered scenes, accepted crops provide cleaner and more question-relevant

visual evidence than the original full image. The detected subset confirms this interpretation: crop-detected accuracy remains high, ranging from 90.83% to 97.23%.

Real-image-test represents a different operating regime. Accepted-crop rates are only 8.44–13.20%, yet total FLOPs still decrease in all eight model–class settings, by 3.1–6.5%. The operation-count benefit is therefore consistent across both domains, although its magnitude is smaller for these already compact inputs.

The 4B model provides the strongest overall accuracy trade-off in these experiments. It obtains the highest mean framework accuracy on generated-test (85.23%) and real-image-test (95.14%). The 9B model does not consistently outperform 4B, which suggests that larger VLM size is not the only determinant of performance in this task. Input quality, accepted-crop coverage, and scene composition also play important roles.

6.4 Discussion of Token and FLOPs Efficiency

The primary efficiency results are consistent across token count and profiled computation. On generated-test images, tokens decrease by 23.2–39.3%, while total framework FLOPs decrease by 23.2–39.5%. These matched reductions show that crop routing reduces both the VLM input and the operations executed by the complete framework.

Real-image-test provides a stricter boundary condition. Accepted-crop rates are only 8.44–13.20%, and total FLOPs decrease by 3.1–6.5%. The smaller gain follows from the composition of this domain: its images are compact, target-centered, and often already resemble an ROI.

The operation-count benefit also scales in absolute terms with model size. For the 9B model on generated-test images, the framework saves 16.90–29.09 TFLOPs per logical request. This result strengthens the computational claim because it is obtained from actual GPU profiling rather than a theoretical estimator.

There is also an edge-cloud interpretation of these results. Stage 1 and Stage 2 together account for at most 1.201% of total framework FLOPs. A practical BCI system could place EEG decoding and ROI routing on an edge device, while the VLM runs remotely. This would shift preprocessing away from the centralized VLM service and reduce the transmitted visual input when a crop is accepted. Network transfer and distributed throughput were not measured, so this remains a deployment implication.

Observed wall-clock results are retained in Appendix A. They provide a stage-resolved record of the implemented system on the available shared cluster. Because elapsed time depends on concurrent workload, memory pressure, and device utilization, these measurements are treated as supplementary systems evidence rather than the primary basis for the efficiency claim.

The generated-test domain was introduced to approximate the intended everyday operating setting. In practical scenes, a relevant object may occupy a small image region and coexist with other objects and background context. The ImageNet-derived real-image-test subset contains genuine photographs, but its composition is strongly target-centered. The local generated-test and real-image-test collections average 6.74 MB and 0.13 MB per image, respectively.

These results refine the main claim of the thesis. The framework is valuable when the target is embedded in a broader scene rather than already isolated by image composition. In such everyday-like multi-object settings, the EEG-derived semantic cue can direct visual processing toward the intended object. The target-centered real-image-test subset offers less opportunity for this guidance because the full image already resembles an ROI.

6.5 Error Propagation Across the Pipeline

Because the method is a sequential pipeline, errors can propagate from one stage to the next. Stage 1 errors affect Stage 2 because the predicted WNID determines the target category used for localization. Stage 2 errors affect Stage 3 because inaccurate boxes may produce crops that miss the target object or remove useful context. Stage 3 errors may still occur even when the crop is visually correct, because VLM answer generation and semantic answer scoring can introduce additional uncertainty.

Table 14: Major error sources in the EEG-to-ROI-to-VQA pipeline.

Stage	Error type	Downstream effect
Stage 1	Wrong WNID prediction	Detector may search for the wrong target
Stage 1	Low confidence prediction	Stage 2 is skipped; framework uses full-image fallback
Stage 2	Missed target detection	Framework falls back to full image
Stage 2	Wrong or poor bounding box	Crop may exclude target or needed context
Stage 3	VLM answer error	Accuracy decreases despite correct visual input
Scoring	Answer-matching ambiguity	Semantically correct answers may be miscounted

This decomposition motivates separate evaluation metrics. Oracle IoU isolates detector ability, while gated IoU captures semantic error propagation. Crop-detected and crop-gated accuracy assess accepted and missing crops, respectively. Framework accuracy evaluates the deployed policy.
Because the framework changes only the visual input, it remains potentially compatible with different VLM backbones and API-based systems.
ROI routing is most useful when the original image does not already isolate the target. In the real-image-test setting, the compact, target-centered input leaves less room for token and operation-count reduction. Deployment should therefore consider scene composition and available hardware.

6.6 Conclusion

This thesis proposed an EEG-based ROI selection framework for efficient VLM-based visual question answering. The framework predicts a target WNID from EEG, uses YOLO to localize the corresponding object, and applies a confidence-gated crop/fallback policy for downstream VQA. The method treats EEG as a semantic cue rather than a direct spatial coordinate predictor.
The experiments support three conclusions. First, Stage 1 provides useful EEG-derived semantic information, but its accuracy decreases as class count grows. Second, Stage 2 localizes targets well given the correct WNID, while Stage 1 errors and confidence gates limit crop use. Third, Stage 3 shows a scene-dependent accuracy-efficiency trade-off. On generated-test images, framework accuracy improves by 4.14–9.87 percentage points. Token use and total framework FLOPs decrease by 23.2–39.3% and 23.2–39.5%, respectively. Total framework FLOPs also decrease in every real-image-test setting, although by a smaller 3.1–6.5%.
Overall, EEG-based ROI selection is a promising preprocessing strategy for everyday-like scenes in which the intended object appears within a broader visual context. Its benefit depends on EEG classification, localization quality, crop acceptance, scene composition, and downstream

execution conditions. This thesis provides a proof of concept and identifies clear boundaries for neural-signal-guided visual input reduction.

6.7 Limitations

Several limitations should be considered when interpreting the results. First, the dataset remains relatively small. Although the thesis extends the original EEG-image setting with generated-test images, real-image-test photographs, ROI annotations, and VQA questions, each category still contains a limited number of base image-question pairs. Larger and more diverse datasets would provide stronger evidence of localization and VQA robustness.

Second, Stage 1 EEG classification remains a major bottleneck. Under the confidence-gated protocol, incorrect or low-confidence Stage 1 predictions reduce accepted-crop coverage and limit the benefit of ROI-based VQA. Future work could explore stronger EEG encoders, subject-adaptive models, temporal modeling, or multimodal priors to improve WNID prediction.

Third, the current framework is evaluated offline. The EEG signals, image data, ROI localization, and VQA evaluation are processed as an experimental pipeline rather than a real-time closed-loop brain-computer interface. Real-time deployment would require online EEG preprocessing, user calibration, latency control, and robustness to noisy neural signals.

Fourth, the supplementary wall-clock evaluation was conducted on shared cluster GPUs. The measurements document the behavior of the implemented pipeline, but elapsed time may vary with concurrent workload, memory pressure, and device utilization. The main efficiency conclusions therefore rely on matched token counts and GPU-profiled FLOPs. Future work should repeat the stage-resolved benchmark on dedicated hardware to obtain a controlled deployment comparison.

Fifth, VQA answer evaluation requires careful semantic scoring. Exact matching may miss semantically equivalent answers, while overly permissive judging may inflate accuracy. The scoring protocol should therefore remain pre-defined, persistent, auditable, and free from answer leakage during VQA generation.

Finally, cropped-ROI VQA depends on bounding-box quality. If the crop is too small, it may remove context needed for answering. If the crop is too large, it may retain background clutter and reduce the efficiency benefit. Future work could investigate adaptive crop margins, uncertainty-aware cropping, or multi-region selection to make ROI-based VQA more robust.

References

[1] Emily J. Allen, Ghislain St-Yves, Yihan Wu, Jesse L. Breedlove, Jacob S. Prince, Logan T. Dowdle, Matthias Nau, Brad Caron, Franco Pestilli, Ian Charest, et al. A massive 7t fmri dataset to bridge cognitive neuroscience and artificial intelligence. *Nature Neuroscience*, 25(1):116–126, 2022.

[2] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2425–2433, 2015.

[3] Nadine Chang, John A. Pyles, Austin Marcus, Abhinav Gupta, Michael J. Tarr, and Elissa M. Aminoff. Bold5000, a public fmri dataset while viewing 5000 visual images. *Scientific Data*, 6(1):49, 2019.

[4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[5] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

[6] Ruo-Nan Duan, Jia-Yi Zhu, and Bao-Liang Lu. Differential entropy feature for eeg-based emotion classification. In *2013 6th International IEEE/EMBS Conference on Neural Engineering*, pages 81–84. IEEE, 2013.

[7] Golnaz Ghiasi, Yin Cui, Aravind Srinivas, Rui Qian, Tsung-Yi Lin, Ekin D. Cubuk, Quoc V. Le, and Barret Zoph. Simple copy-paste is a strong data augmentation method for instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2918–2928, 2021.

[8] Alessandro T. Gifford, Kshitij Dwivedi, Gemma Roig, and Radoslaw M. Cichy. A large and rich eeg dataset for modeling human visual object recognition. *NeuroImage*, 264:119754, 2022.

[9] Tomoyasu Horikawa and Yukiyasu Kamitani. Generic decoding of seen and imagined objects using hierarchical visual features. *Nature Communications*, 8(1):15037, 2017.

[10] Ming Jiang, Shengsheng Huang, Juanyong Duan, and Qi Zhao. Salicon: Saliency in context. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1072–1080, 2015.

[11] Vernon J. Lawhern, Amelia J. Solon, Nicholas R. Waytowich, Stephen M. Gordon, Chou P. Hung, and Brent J. Lance. Eegnet: A compact convolutional neural network for eeg-based brain–computer interfaces. *Journal of Neural Engineering*, 15(5):056013, 2018.

[12] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International Conference on Machine Learning*, 2023.

[13] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International Conference on Machine Learning*, pages 12888–12900, 2022.

[14] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollar, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, pages 740–755, 2014.

[15] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *Advances in Neural Information Processing Systems*, 2023.

[16] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *Advances in Neural Information Processing Systems*, 2019.

[17] Michael I. Posner. Orienting of attention. *Quarterly Journal of Experimental Psychology*, 32(1):3–25, 1980.

[18] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763, 2021.

[19] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 779–788, 2016.

[20] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, 2015.

[21] Concetto Spampinato, Simone Palazzo, Isaak Kavasidis, Daniela Giordano, Nasim Souly, and Mubarak Shah. Deep learning human mind for automated visual classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6809–6817, 2017.

[22] Hao Tan and Mohit Bansal. Lxmert: Learning cross-modality encoder representations from transformers. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2019.

[23] Michal Teplan. Fundamentals of eeg measurement. *Measurement Science Review*, 2(2):1–11, 2002.

[24] Pavan Kumar Anasosalu Vasu, Fartash Faghri, Chun-Liang Li, Cem Koc, Nate True, Albert Antony, Gokul Santhanam, James Gabriel Peter Grasch, Oncel Tuzel, and Hadi Pouransari. Fastvlm: Efficient vision encoding for vision language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

[25] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7464–7475, 2023.

[26] JinYi Yoon, Kazi Hasan Ibn Arif, and Bo Ji. Gazevlm: Gaze-guided vision-language models for efficient and robust inference. *OpenReview*, 2025.

[27] Sangdoo Yun, Dongyun Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6023–6032, 2019.

[28] Yuan Zhang, Chun-Kai Fan, Junpeng Ma, Wenzhao Zheng, Tao Huang, Kuan Cheng, Denis Gudovskiy, Tomoyuki Okuno, Yohei Nakata, Kurt Keutzer, et al. Sparsevlm: Visual token sparsification for efficient vision-language model inference. *arXiv preprint arXiv:2410.04417*, 2024.

[29] Shuqi Zhu, Ziyi Ye, Qingyao Ai, and Yiqun Liu. Eeg-imagenet: An electroencephalogram dataset and benchmarks with image visual stimuli of multi-granularity labels. *arXiv preprint arXiv:2406.07151*, 2024.

A Supplementary Wall-Clock Latency Evaluation

A.1 Scope and Measurement Protocol

Wall-clock latency remains relevant to practical deployment, so the complete measurements are retained in this appendix. They provide a transparent record of the implemented confidence-gated pipeline and show how execution time is distributed across its three stages. The experiments were conducted on shared cluster GPUs. Elapsed time can therefore vary with concurrent workload, memory pressure, GPU utilization, and multi-device execution. For this reason, latency is treated as supplementary systems evidence rather than a hardware-normalized comparison or the primary basis for the efficiency claim. The main analysis uses matched token counts and actual GPU-profiled FLOPs, which are less sensitive to concurrent cluster activity.

Latency was measured with batch size one. Each logical request followed the fixed confidence-gated route with $\tau_{\text{eeg}} = 0.20$ and $\tau_{\text{roi}} = 0.05$. The direct baseline measured full-image VQA. The framework measurement included Stage 1 EEG classification, conditionally executed Stage 2 localization, and routed Stage 3 VQA. Offline training, model loading, data transfer, semantic scoring, file I/O, and cache-hit time were excluded.

The complete framework latency is

$$L_{\text{framework}} = L_{\text{Stage 1}} + L_{\text{Stage 2,amortized}} + L_{\text{Stage 3}}. \tag{21}$$

Let $e_i \in \{0, 1\}$ indicate whether Stage 2 executes for logical request i. Stage 2 latency is amortized over all N logical requests:

$$L_{\text{Stage 2,amortized}} = \frac{1}{N} \sum_{i=1}^{N} e_i L_{\text{Stage 2}}(i). \tag{22}$$

A request rejected by the Stage 1 confidence gate has $e_i = 0$ and therefore contributes zero Stage 2 latency. This convention follows the actual runtime route and does not charge a skipped request for localization work. The reported observed latency reduction is

$$\text{LatencyReduction} = \frac{L_{\text{full}} - L_{\text{framework}}}{L_{\text{full}}} \times 100\%. \tag{23}$$

A.2 Stage-Resolved Results

Tables 15–17 report the observed wall-clock measurements. Each framework entry gives the total first, followed in parentheses by Stage 1, amortized Stage 2, and Stage 3.

Table 15: Observed wall-clock latency for Qwen3.5-2B under the confidence-gated framework policy ($\tau_{\text{eeg}} = 0.20$, $\tau_{\text{roi}} = 0.05$). The framework column reports total mean latency, followed by Stage 1, amortized Stage 2, and Stage 3 mean latency.

Domain	Classes	Full VQA latency (ms)	Framework latency (ms) Total (Stage 1/2/3)	Observed reduction (%)
generated-test	10	1870.7	1601.9 (65.2/16.5/1520.2)	14.4
	20	1851.5	1491.0 (65.2/16.3/1409.5)	19.5
	30	1812.3	1566.2 (63.9/15.6/1486.8)	13.6
	40	1818.7	1634.2 (66.0/15.8/1552.5)	10.1
real-image-test	10	359.8	406.8 (64.8/10.0/332.0)	-13.1
	20	334.3	422.3 (62.6/9.7/350.1)	-26.3
	30	357.2	427.8 (62.6/18.7/346.5)	-19.8
	40	362.8	425.9 (64.2/10.6/351.1)	-17.4

Table 16: Observed wall-clock latency for Qwen3.5-4B under the confidence-gated framework policy ($\tau_{\text{eeg}} = 0.20$, $\tau_{\text{roi}} = 0.05$). The framework column reports total mean latency, followed by Stage 1, amortized Stage 2, and Stage 3 mean latency.

Domain	Classes	Full VQA latency (ms)	Framework latency (ms) Total (Stage 1/2/3)	Observed reduction (%)
generated-test	10	2636.3	2333.9 (65.5/15.6/2252.7)	11.5
	20	2637.3	2085.5 (64.6/16.3/2004.6)	20.9
	30	2573.5	2210.6 (63.9/15.6/2131.1)	14.1
	40	2555.4	2183.9 (66.0/15.8/2102.2)	14.5
real-image-test	10	495.7	519.6 (64.0/9.9/445.6)	-4.8
	20	453.3	538.1 (64.8/9.7/463.6)	-18.7
	30	475.5	536.7 (62.6/18.7/455.3)	-12.9
	40	486.3	537.6 (64.2/10.6/462.8)	-10.5

Table 17: Observed wall-clock latency for Qwen3.5-9B under the confidence-gated framework policy ($\tau_{\text{eeg}} = 0.20$, $\tau_{\text{roi}} = 0.05$). The framework column reports total mean latency, followed by Stage 1, amortized Stage 2, and Stage 3 mean latency.

Domain	Classes	Full VQA latency (ms)	Framework latency (ms) Total (Stage 1/2/3)	Observed reduction (%)
generated-test	10	3591.7	4437.3 (65.5/15.6/4356.2)	-23.5
	20	3570.4	4039.9 (64.6/16.3/3959.1)	-13.2
	30	3043.7	4278.6 (65.9/15.8/4196.9)	-40.6
	40	3073.2	4264.3 (61.2/14.4/4188.6)	-38.8
real-image-test	10	466.0	731.6 (64.0/9.9/657.7)	-57.0
	20	443.0	758.7 (64.8/9.7/684.2)	-71.2
	30	539.9	780.9 (63.3/17.7/699.9)	-44.6
	40	528.8	787.3 (58.8/9.0/719.4)	-48.9

A.3 Interpretation and Measurement Boundary

Under the measured shared-cluster conditions, the 2B and 4B models show lower total latency on generated-test images, with observed reductions of 10.1–20.9%. Their Stage 1 and amortized Stage 2 components remain small relative to Stage 3. This result demonstrates that the implemented route can reduce elapsed time in some operating conditions.

The 9B generated-test runs and all real-image-test runs show higher observed framework latency. The stage breakdown places most of the 9B difference in Stage 3 rather than in EEG classification or localization. For real-image-test, the full-image VQA baseline is already short because the images are compact and target-centered, leaving less opportunity to recover the serial front-end time.

These measurements complement, but do not replace, the main computational analysis. They document the behavior of the complete implementation on the available system and reveal where runtime is spent. However, they should not be interpreted as a controlled cross-hardware ranking because concurrent jobs and device conditions were not isolated. A dedicated-hardware benchmark with controlled GPU utilization and paired repetitions would provide a more direct deployment-level latency comparison.