

ECVL-ROUTER: SCENARIO-AWARE ROUTING FOR VISION-LANGUAGE MODELS

Xin Tang^{*1,2} Youfang Han^{*2} Fangfei Gou¹ Wei Zhao² Xin Meng² Yang Yu²

Jinguo Zhang² Yuanchun Shi² Yuntao Wang^{†1} Tengxiang Zhang^{‡2}

¹Tsinghua University

²Goertek Inc.

ABSTRACT

Vision-Language Models (VLMs) excel in diverse multimodal tasks. However, user requirements vary across scenarios, which can be categorized into fast response, high-quality output, and low energy consumption. Relying solely on large models deployed in the cloud for all queries often leads to high latency and energy cost, while small models deployed on edge devices are capable of handling simpler tasks with low latency and energy cost. To fully leverage the strengths of both large and small models, we propose ECVL-ROUTER, the first scenario-aware routing framework for VLMs. Our approach introduces a new routing strategy and evaluation metrics that dynamically select the appropriate model for each query based on user requirements, maximizing overall utility. We also construct a multimodal response-quality dataset tailored for router training and validate the approach through extensive experiments. Results show that our approach successfully routes over 80% of queries to the small model while incurring less than 10% drop in problem solving probability.

1 INTRODUCTION

Vision-Language Models (VLMs), which integrate visual and textual understanding, have become crucial components in a wide range of AI applications, from robotics control to user interface navigation (Zhang et al., 2024; Shinde et al., 2025; Li et al., 2024). In practice, predominant deployment strategy relies heavily on powerful, cloud-hosted Large VLMs (LVLMs) to serve all user queries (Jang & Morabito, 2025; Navardi et al., 2025; Zheng et al., 2025), which excel at complex reasoning but incur latency and energy costs (Fernandez et al., 2025; Jegham et al., 2025; Charyyev et al., 2020) while underutilizing capable small VLMs (SVLMs) on edge devices (Sharshar et al., 2025; Belcak et al., 2025). Moreover, a one-size-fits-all deployment strategy is suboptimal, as users increasingly expect systems that not only deliver high-quality responses but also adapt to diverse real-world scenarios with varying demands for latency, cost, and privacy.

To effectively integrate the strengths of both LVLMs and SVLMs, edge-cloud collaborative routing (Yuan et al., 2025; Hao et al., 2024) is a natural fits. At its core is a lightweight model router (Ding et al., 2024; Ong et al., 2024) that inspects each query and selects an appropriate VLM. However, a general router is insufficient, routing must be **scenario-aware**: behaviors vary across diverse application contexts and can be configured by users or automatically inferred by scenario detection algorithms (Fifty et al., 2023; Someki et al., 2025). Existing routers are often text-centric and optimize a fixed trade-off between cost and quality, failing to adapt to multimodal, scenario-aware user needs. For example, real-time games interaction prioritizes low latency, medical diagnostics emphasizes answer quality, and mobile assistants require low energy use and strong privacy (Asgari et al., 2025). Therefore, an ideal model router should align with diverse user requirements and make the most appropriate routing decision in different scenarios.

In this work, we introduce **Edge-Cloud Vision-Language Router (ECVL-ROUTER)**, a novel, scenario-aware routing framework for VLMs. Motivated by heterogeneous real-world scenarios, we distill three primary user requirements: (1) fast response, (2) high-quality output, and (3) low energy consumption & data privacy. Accordingly, our routing objective is to maximize the use of small edge models while meeting user satisfaction. To achieve this, we introduce the *Minimal Expectation Score* (MES) to quantify the user’s acceptable quality threshold in different scenarios. Guided by MES, the router prefers an SVLM whenever its predicted output meets the MES threshold and escalates to a larger cloud model otherwise. To implement this strategy, we design a lightweight routing classifier and construct a response quality dataset using an MES-based annotation policy to support its training. For evaluation, we propose the *Routing Comprehensive Score* (RCS), integrating three key aspects: *Average Problem-Solving Probability* (APSP), *Cost Advantage* (CA), and *Average Inference Latency* (AIL), which correspond directly to three core user requirements. Ex-

tensive experiments demonstrate that our routing strategy and framework delivers favorable trade-offs across quality, latency, and cost while adhering to scenario-specific user requirements.

The main contributions of this paper can be summarized as follows:

1. We propose a novel, scenario-aware routing strategy for VLMs and a new set of model routing evaluation metrics (Section 3). This strategy moves beyond the traditional cost-quality trade-off by centering on the user’s dynamic requirements to maximize the system’s overall utility.
2. We design and implement ECVL-ROUTER, the first cloud-edge collaborative, scenario-aware routing framework for VLMs, which effectively combines the high performance of large cloud models with the low-cost advantages of small edge models (Section 4.2).
3. We construct the first response quality dataset for training and evaluating VLM routers (Section 4.1). Through extensive experiments on this dataset, we have thoroughly validated the effectiveness of our ECVL-ROUTER framework (Section 5). We also open source our dataset and framework for users to train their routers based on their specific requirements.¹

2 RELATED WORK

2.1 SVLM vs. LVLM

Vision-Language Models now underpin applications from accessibility assistance and UI navigation to robotics and scientific content creation (Zhang et al., 2024; Shinde et al., 2025). User requirements in these domains differ widely. Some applications, like real-time games or augmented reality, demand **fast response** where low latency is critical (Vasu et al., 2025; Liu et al., 2025). Others, such as medical diagnostics, insist on **high-quality answers** where accuracy is paramount (Singhal et al., 2025; 2023). For applications on mobile or IoT devices, **low energy consumption and privacy constraints** are decisive factors (Chu et al., 2023; Wang et al., 2025).

Cloud-hosted LVLMs (e.g., GPT-4o and Gemini 2.5 Pro) offer strong multimodal reasoning but impose round-trip latency and significant compute costs, making them ill-suited for latency-sensitive or privacy-constrained settings (Hurst et al., 2024; Team et al., 2023; Charyyev et al., 2020; Fernandez et al., 2025). In parallel, SVLMs have become increasingly capable on consumer hardware. For example, Google’s Gemma 3 270M can run on a mobile phone with very low power consumption; internal tests showed that 25 conversations on a Pixel 9 Pro consumed only 0.75% of the battery (Lacombe et al., 2025). Meanwhile, Microsoft’s Phi-4-multimodal, with only 5.6 billion parameters, runs on personal computers to efficiently handle tasks like OCR and chart understanding (Abouelenin et al., 2025). Despite these advances, production stacks still over-rely on cloud LVLMs. This gap motivates scenario-aware edge-cloud designs that preferentially serve queries on-device and escalate only when quality requirements exceed local capacity.

2.2 MODEL ROUTING

From objective functions and decision paradigms, prior work falls into three types:

(1) **Outcome-optimal, non-predictive cascades.** These methods prioritize the quality of the final answer by allowing multiple model calls until a target quality threshold is met. A common strategy is *cascading*: invoke models from low to high cost and stop once the response satisfies a predefined criterion. *FrugalGPT* follows this paradigm (Chen et al., 2023). *AutoMix* first lets a small model self-evaluate its draft and escalates only if quality is predicted to be insufficient (Aggarwal et al., 2023). While effective for quality, these approaches often incur significant extra latency due to repeated or parallel model evaluations.

(2) **Resource-optimal, predictive routing.** Unlike cascading methods, predictive routing seeks to choose the right model in a single shot, optimizing cost/latency by training a lightweight router that predicts the appropriate model before execution. *Hybrid-LLM* uses a difficulty-aware router to decide between small and large models (Ding et al., 2024). *RouterLLM* provides a unified training framework for learning performance-cost aware selectors (Ong et al., 2024). NVIDIA’s *LLM-Router* leverages pretrained classifiers to dispatch by task type and complexity (NVIDIA Corporation, 2024). Research also explores alternative designs, e.g., graph-based routing in *GraphRouter* (Feng et al., 2024) and tool-oriented modular routing in *TO-Router* (Stripelis et al., 2024). However, many methods frame routing as predicting whether a small model can *beat* a large model, rather than whether the small model is good enough for the scenario, a crucial distinction for practical deployments.

¹The code is available at <https://anonymous.4open.science/r/ECVL-Router-977D>

(3) **Compute-aware reasoning control in agentic systems.** Beyond model selection, routing principles appear in system architectures and agent workflows. Modern systems (e.g. GPT-5) use a real-time router to choose between instant response and deeper thinking for reasoning based on estimated task difficulty and explicit user intent (OpenAI, 2025). At the agent level, routing determines not only which LLM to use but also which tool or specialized sub-agent to activate within a complex system. (Wu et al., 2024; Yao et al., 2025).

Despite progress, three limitations remain. (i) **Modality limitation:** prior routers target only text inputs and do not account for the unique challenges of multimodal (image–text) inputs required by VLMs. (ii) **Oversimplified routing strategy:** many method such as RouterLLM and Hybrid-LLM are driven by routing to a binary “small-vs-large wins” signal, by contrast, our router explicitly considers “good enough” performance of small models for the target scenario, improving edge utilization without unnecessary escalation. (iii) **Lack of scenario-aware user needs:** existing evaluations largely collapse to a single cost–quality trade-off and rely on cost/latency alone, overlooking that user requirements over answer quality, responsiveness, and resource usage vary across applications.

3 SCENARIO-AWARE VLM ROUTING

This section formally defines our scenario-aware routing problem. We begin by defining the models and user-centric scenarios. Then we introduce the Minimal Expectation Score (MES) to quantify user satisfaction. we formally define the core routing problem and establish the decision rules. Finally, we propose a suite of evaluation metrics to assess the router’s effectiveness across three scenarios.

3.1 ROUTING PROBLEM DEFINITION

Model definitions. Our framework considers two types of VLMs: (1) M_{edge} , a lightweight SVLM deployed on consumer hardware such as laptops or mobile phones (typically < 10B parameters); and (2) M_{cloud} , a more powerful LVLM deployed on the cloud (typically > 10B parameters). Each model M maps a multimodal input—user query Q and input data I (text and images)—to a response:

$$M : \langle Q, I \rangle \mapsto \text{Response} \quad (1)$$

Scenario definitions & MES. We categorize user requirements into three primary scenarios: (1) **Fast Response** (Speed), (2) **High-Quality Output** (Quality), and (3) **Low Energy Consumption & Data Privacy** (Efficiency).

To quantify user requirements in our routing problem, we introduce the **Minimal Expectation Score (MES)**. The MES represents the lowest response-quality threshold a user is willing to accept in a given scenario. Let $\text{Score}_{\text{edge}} = \text{Score}_{M_{\text{edge}}}(Q, I)$ and $\text{Score}_{\text{cloud}} = \text{Score}_{M_{\text{cloud}}}(Q, I)$ denote response-quality scores (see Appendix A.2 for criteria). Any output with $\text{Score} \geq \text{MES}$ is deemed satisfactory; scores below MES are not. The MES effectively captures user needs across different scenarios; for instance, the Quality scenario typically demands a higher MES, while the Speed scenario may tolerate a lower one. Based on MES, we also construct a multimodal response-quality dataset for subsequent training and validation (see in Section 4.1).

Routing problem. The core of our work is the routing problem, which aims to maximize the use of M_{edge} while satisfying the user’s MES across different scenarios. Define a routing process R that selects $M_R \in \{M_{\text{edge}}, M_{\text{cloud}}\}$. We say M_{edge} is *competent* for (Q, I) if the binary label $L = 1$ under the following rule:

$$L = \mathbb{I} \left[\underbrace{\text{Score}_{\text{edge}} \geq \min\{\text{Score}_{\text{cloud}}, \text{MES}\}}_{\text{Case A: edge reaches the cloud-or-MES floor}} \text{ or } \underbrace{\text{Score}_{\text{cloud}} < \text{MES}}_{\text{Case B: cloud fails MES, avoid its cost}} \right]. \quad (2)$$

Case A routes to M_{edge} when it achieves at least the minimum of the cloud score and MES, which means M_{edge} is competent. *Case B* covers failure regimes where even M_{cloud} cannot satisfy MES, which means the problem can’t be solved. Using this rule, we construct training data $\mathcal{D}_{\text{train}} = \{(Q_t, I_t, L_t)\}_{t=1}^N$, where N is the total number of samples in the training set, (Q_t, I_t) is the t -th input query and data, L_t is the t -th routing result.

Routing classifier and decision. The routing process is implemented by a **routing classifier**, which predicts the probability p that M_{edge} is competent for a task: $p = P_{\theta}(L=1 \mid Q, I)$, where θ denotes model parameters learned by minimizing cross-entropy loss on $\mathcal{D}_{\text{train}}$. We use a decision threshold τ to convert p into a routing decision:

$$R(Q, I) = \begin{cases} M_{\text{edge}}, & \text{if } p \geq \tau, \\ M_{\text{cloud}}, & \text{if } p < \tau. \end{cases} \quad (3)$$

Adjusting τ trades off quality, cost, and latency for different scenarios. (see in Section 4.2)

3.2 EVALUATION METRICS

To evaluate the scenario-aware router, we propose three metrics aligned with the core user requirements, along with a composite score. For the three, (N, Q_t, I_t) share the same meaning with Section 3.1 and M_{R_t} is the model selected by router R for the t -th sample.

Average Problem-Solving Probability (APSP). Rate at which routed responses meet MES (aligns with Quality):

$$\text{APSP}(R) = \frac{1}{N} \sum_{t=1}^N \mathbb{I}\{\text{Score}_{M_{R_t}}(Q_t, I_t) \geq \text{MES}\}. \quad (4)$$

A higher APSP indicates stronger problem-solving capability.

Cost Advantage (CA). Fraction of queries handled by M_{edge} (aligns with Efficiency):

$$\text{CA}(R) = \frac{1}{N} \sum_{t=1}^N \mathbb{I}\{M_{R_t} = M_{\text{edge}}\}. \quad (5)$$

A higher CA indicates better resource savings.

Average Inference Latency (AIL). Mean end-to-end latency (aligns with Speed):

$$\text{AIL}(R) = \frac{1}{N} \sum_{t=1}^N \text{latency}(M_{R_t}(Q_t, I_t)). \quad (6)$$

A lower AIL indicates faster responses.

Routing Comprehensive Score (RCS). A weighted aggregate (higher is better):

$$\text{RCS}(R) = \alpha \text{APSP}(R) + \beta \text{CA}(R) - \gamma \text{AIL}(R). \quad (7)$$

The hyperparameters α, β, γ can be tuned based on scenario priorities. A high APSP often leads to a lower CA and a higher AIL, as more tasks are routed to the LVLM. The RCS enables balanced evaluation. In our experiments (see in Sec 5), we use three parameterizations to simulate different user priorities:

$\text{RCS1}_{(\text{Quality})} : (\alpha, \beta, \gamma) = (1.2, 0.1, 0.001)$, $\text{RCS2}_{(\text{Efficiency})} : (1.0, 0.12, 0.001)$, $\text{RCS3}_{(\text{Speed})} : (1.0, 0.1, 0.0015)$.

4 METHODOLOGY

4.1 RESPONSE SCORE DATASET

Dataset construction. Guided by the scenario-aware MES defined in Sec.3, we construct the Response Score Dataset (RSD) tailored for VLM routing. Our goal is to efficiently estimate model response quality across diverse tasks via large-scale automated annotation. Concretely, we first select a representative set of **8** VLMs and prompt them to generate responses on **7** public benchmarks that cover a broad range of task types and difficulty levels (see in Appendix A.1). Next, we adopt the *LLM-as-a-Judge* way, which has shown to correlate strongly with human ratings in prior work (Zheng et al., 2023; Gu et al., 2024) and use GPT-4o as the *LLM Response Judge (LRJ)*. For each model response, the LRJ assigns a unified *score* $\in [1, 10]$ under a consistent rubric. The score evaluates the helpfulness, relevance, accuracy of the response (see in Appendix A.2 for details).

Human gold labels To ensure the reliability of automated scoring, we rigorously validated LRJ annotations against human gold labels. We recruited five human experts to independently score a stratified random sample of 200 instances. LRJ scores show strong alignment with the mean human rating (Pearson correlation $r > 0.85$), confirming its effectiveness and reliability as an automated annotator (see Appendix A.3 for details).

In summary, the Response Score Dataset (RSD) contains approximately **22k** image-text instances, each with response quality scores and inference times for **8** VLMs. Furthermore, we analyze the distribution of response quality and inference time, as well as the performance of different models across each benchmark (see Appendix A.4 for full analysis).

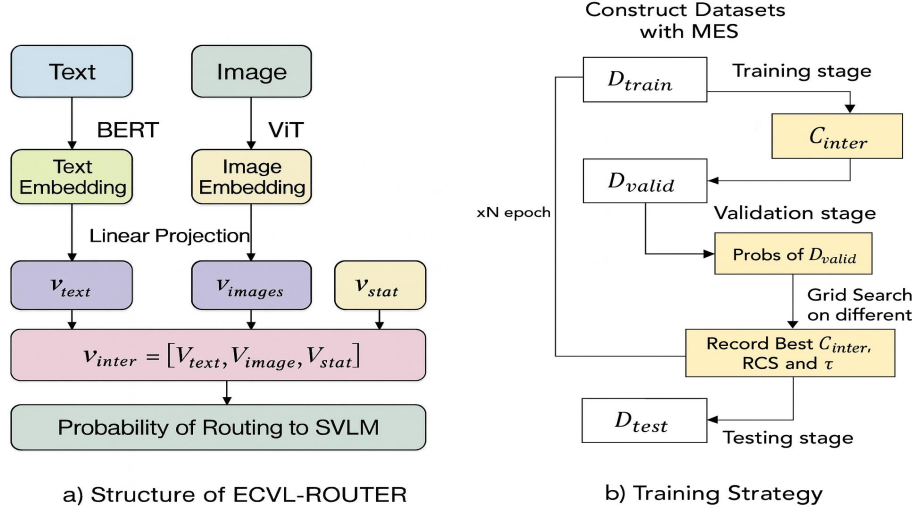


Figure 1: (a) Overall Structure and (b) Training Strategy of ECVL-ROUTER

This helps us define the difficulty of each benchmark and the capabilities of different models. The distribution result also supports the hypothesis that larger models typically take more time to provide higher-quality responses.²

4.2 ECVL-ROUTER

We propose ECVL-ROUTER, a transformer-based routing framework for Vision-language models. ECVL-ROUTER uses a Transformer classifier to process heterogeneous inputs (e.g. text and images). On the training side, ECVL-ROUTER selects the decision threshold τ via validation-set analysis, yielding a threshold tailored to each model pair, MES setting, and application scenario. The overall architecture and training pipeline of ECVL-ROUTER are shown in Fig 1.

4.2.1 MODEL ARCHITECTURE

Input encoding. We obtain modality-specific embeddings for text and images. For text, we use a pretrained encoder (e.g. BERT_BASE) to produce $\mathbf{e}_{\text{text}}$. For images, we adopt a standard vision encoder (e.g., ViT) to obtain $\mathbf{e}_{\text{image}}$. Each embedding $\mathbf{e}_m \in \mathbb{R}^{k_m}$ is projected into a shared d -dimensional space via a linear layer, i.e., $\mathbf{v}_m = \mathbf{W}_m \mathbf{e}_m + \mathbf{b}_m$, where $\mathbf{W}_m \in \mathbb{R}^{d \times k_m}$ and $\mathbf{b}_m \in \mathbb{R}^d$, with $m \in \{\text{text, image}\}$, and k_m is the dimensionality of the vector obtained by applying a linear projection to modality m .

Modalities fusion. To capture input complexity in addition to semantics, we compute lightweight statistics from raw inputs (e.g., word count, special-character and numeric-token counts for text; width/height and color-channel indicators for images). These statistics are linearly embedded as $\mathbf{v}_{\text{stat}} \in \mathbb{R}^d$. We then form the intermediate representation by concatenation: $\mathbf{v}_{\text{inter}} = [\mathbf{v}_{\text{text}}; \mathbf{v}_{\text{image}}; \mathbf{v}_{\text{stat}}]$.

Routing classifier. ECVL-ROUTER employs a lightweight Transformer encoder (2 layers, hidden size 256, 4 attention heads, FFN size 512, dropout 0.3) followed by a linear classifier. Given $\mathbf{v}_{\text{inter}}$ as input, the classifier outputs the probability that the edge-side SVLM can meet the scenario-dependent MES, modeled as $p_\theta(\text{SVLM} \mid q, \mathbf{x}) = \sigma(\mathbf{w}^\top \mathbf{h}_{\text{out}} + b)$, where θ denotes all model parameters, $\mathbf{h}_{\text{out}}$ is the final encoder representation, and $\sigma(\cdot)$ is the sigmoid function. At inference time, we route to the SVLM if $p_\theta \geq \tau$ and otherwise escalate to the cloud-side LVLM, with $\tau \in (0, 1)$ fixed by the validation procedure described above.

4.2.2 TRAINING STRATEGY

Data construction. We first structure the original corpora into triples $D = \{(Q, I, L)\}$ following Sec. 3. Here, Q denotes the user query, I denotes the associated input (images and text), and $L \in \{0, 1\}$ is the *edge-competency* label

²The construction of RSD—including model inference, LRJ, and human gold validation—incurred USD \$1000 cost.

computed by equation 2. The dataset is then randomly split into the training set D_{train} , validation set D_{valid} , and test set D_{test} with a ratio of 6:2:2, while preserving the distribution of sources and labels.

Training stage. At each epoch, an intermediate classifier C_{inter} is trained on D_{train} . The classifier outputs $p = P_{\theta}(L=1 \mid Q, I)$, the probability that M_{edge} is competent for (Q, I) as defined in Sec. 3. Given the ground-truth label $y = L \in \{0, 1\}$, parameters are updated by minimizing the binary cross-entropy $\mathcal{L} = -[y \log p + (1 - y) \log(1 - p)]$. We use Adam with an initial learning rate 10^{-3} and the OneCycleLR schedule (Smith & Topin, 2019) to leverage super-convergence. The model is trained for a total of 50 epochs with a batch size of 64.

Validation stage. During validation, we perform a grid search over the decision threshold $\tau \in [0, 1]$ with a step of 0.05 to maximize the RCS on D_{valid} . For each epoch, we evaluate $\text{RCS}(D_{\text{valid}}; \tau)$ for all candidate τ , record the best pair (τ^*, RCS^*) , and finally retain the C_{inter} achieving the highest RCS^* together with its τ^* . This procedure adapts the router to scenario-specific requirements and generalizes well to the test set (see in Fig.9).

5 EXPERIMENTS

5.1 EXPERIMENTAL SETUP

Base VLMs & Comparison Methods. We run all experiments offline on an NVIDIA A800 (80 GB) and an Intel[®] Xeon[®] Platinum 8250C (128 cores). To avoid API latency and reduce cost, the main study uses open source VLMs from OpenGVLab at three scales: InternVL3-38B, InternVL3-8B, and InternVL2.5-1B. These form three model pairs, and we include additional pairs for generalization tests (see in Appendix B.1). Due to there are no previous works in VLM router, following text-only router work (Ong et al., 2024; Ding et al., 2024), we compare against **All-at-Large**, **All-at-Small**: route all queries to large/small model deployed on cloud/edge, and Matrix Factorization (MF). We also evaluate Gradient Boosted Decision Trees (GBDT, 100 trees) and a multilayer perceptron (MLP) with three hidden layers. For the MLP, input/output dimensions and the optimizer learning rate match those of ECVL-ROUTER.

Dataset & Evaluation Metrics. We utilize the Response Score Dataset, where samples are labeled based on a MES of 6. A sample is positive when the SVLM reaches $\text{Score}_{\text{edge}} \geq 6$. The dataset is split into training, validation, and test sets at 60%, 20%, and 20%, respectively, ensuring stratified label distributions. During evaluation, we report the per-scenario metrics **APSP**, **CA** and **AIL**, along with their composite scores **RCS1**, **RCS2**, and **RCS3** for the three different scenarios.

5.2 ROUTER PERFORMANCE RESULTS

Table 1 shows the performance of our ECVL-ROUTER against comparison methods across three model pairs. Based on this, we can draw the following observations:

Obs 1: ECVL-ROUTER achieves optimal performance across all scenarios. The experimental results show that our proposed ECVL-ROUTER consistently achieves the highest composite scores on all three model pairs across the three scenarios. Beyond these in-domain results, we also conduct the cross-domain experiments in Appendix B.2.

Obs 2: ECVL-ROUTER achieve high edge(small) model utilization with minimal quality loss and lower latency. While maintaining high-quality responses, ECVL-ROUTER substantially increases the utilization of the small model and significantly reduces response latency. For instance, compared to the All-at-Large baseline on the 38B/1B pair, ECVL-ROUTER routes 82% of queries to the SVLM. This is achieved with only a minor drop in the APSP of less than 8% ($0.549 \rightarrow 0.506$), while delivering a significant 39.1% reduction in AIL.

5.3 ANALYSIS OF SCENARIO-AWARE HYPERPARAMETERS (τ AND MES)

Our router is controlled by two hyperparameters: the decision threshold τ and MES. We pick τ via validation grid search for each scenario and validate its robustness in further experiments in Appendix B.2. MES is application-defined and proxies task difficulty (higher MES $\Rightarrow$ harder). We analyze the impact of varying τ and MES on the router’s performance. For all other experiments, MES is set to 6 by default. From figure 2 and 12, we can find that:

Obs 3: The decision threshold τ governs the cost-quality trade-off, with an optimal value τ^* that is scenario-dependent and lies within $[0.7, 0.9]$. In figure 2(a), as τ increases, fewer queries go to the SVLM: CA $\downarrow$, while

Table 1: Model router performance on three model pairs. APSP, CA, AIL are the metrics in RCS1. RCS1/2/3 are composite scores for *Quality*, *Efficiency*, and *Speed* scenarios.

Model Pair	Model Router	Evaluation metrics(RCS1)			Composite Scores		
		APSP $\uparrow$	CA $\uparrow$	AIL [s] $\downarrow$	RCS1 $\uparrow$	RCS2 $\uparrow$	RCS3 $\uparrow$
InternVL 38B/1B	ECVL-ROUTER	0.506	0.824	4.53	0.685	0.601	0.582
	GBDT	0.518	0.631	5.38	0.680	0.596	0.577
	MLP	0.515	0.645	4.41	0.678	0.594	0.575
	MF	0.503	0.439	4.49	0.643	0.551	0.540
	All-at-Large	0.549	0.000	7.44	0.652	0.542	0.538
	All-at-Small	0.456	1.000	0.94	0.646	0.575	0.554
InternVL 8B/1B	ECVL-ROUTER	0.483	0.910	1.34	0.669	0.591	0.572
	GBDT	0.478	0.941	1.29	0.666	0.589	0.570
	MLP	0.485	0.873	1.24	0.668	0.589	0.570
	MF	0.469	0.800	1.18	0.642	0.564	0.547
	All-at-Large	0.529	0.000	1.63	0.633	0.527	0.527
	All-at-Small	0.456	1.000	0.94	0.646	0.575	0.554
InternVL 38B/8B	ECVL-ROUTER	0.533	0.982	1.77	0.736	0.649	0.629
	GBDT	0.534	0.965	1.86	0.735	0.648	0.628
	MLP	0.534	0.887	2.30	0.727	0.638	0.619
	MF	0.529	1.000	1.63	0.733	0.647	0.627
	All-at-Large	0.549	0.000	7.44	0.652	0.542	0.538
	All-at-Small	0.529	1.000	1.63	0.733	0.647	0.627

escalation to the LVLM raises APSP $\uparrow$ and AIL $\uparrow$. This captures the core trade-off: higher quality costs more latency/energy; speed efficiency implies some quality drop. By finding the optimal τ on the validation set, we can identify the best value for a specific scenario. In figure 2(b), across RCS1/2/3, most methods (except MF) show unimodal curves peaking in 0.6–0.9 rather than at $\tau=0.5$, which means τ has optimal value and is differed by scenarios. As seen in the red annotations of figure 2(b), scenarios prioritizing quality (RCS1) favor a higher optimal τ^* , while those focused on low cost and speed (RCS2, RCS3) benefit from a lower optimal τ^* . Our router leads across most τ , with peak performance often at higher τ (0.7–0.9). In contrast, MLP and GBDT vary smoothly, whereas MF is limited by linear factorization and misses rich image–text interactions.

Obs 4: ECVL-ROUTER delivers the largest gains at medium difficulty task ($MES \in [5, 7]$). When $MES < 5$, tasks are easy and the SVLM already succeeds; when $MES > 8$, both models often fail, leaving little room for routing. Concretely, In $MES = 5\text{--}7$, ECVL-ROUTER shows the largest margin over alternatives (e.g., average RCS gain = +0.7% in figure 12(b)), because the small-large model capability gap is most informative: the router accurately escalates only those queries the SVLM cannot solve, maximizing edge usage without undue quality loss. And in 12(a), as MES increases, APSP decreases monotonically, while CA (first $\downarrow$, then $\uparrow$) and AIL (first $\uparrow$, then $\downarrow$) are non-monotonic because the rule defaults to the SVLM when neither model meets MES; thus at very high MES, the proportion of queries handled by the SVLM rises again. A detailed analysis of these cases is provided in Appendix B.4.

5.4 ROUTER LATENCY ANALYSIS & ABLATION EXPERIMENT

We measure end-to-end inference latency on the test set for the InternVL family and for each component of ECVL-ROUTER. From Table 2 and 3, we have the following observations:

Obs 5: Routing latency of ECVL-ROUTER is negligible for user experience. One ECVL-ROUTER pass takes 0.0159 s, i.e., 1.7% of InternVL2.5-1B, 0.97% of InternVL3-8B, and 0.21% of InternVL3-38B. Within the router, the ViT encoder dominates (93.7%), while fusion is near zero (0.06%). In practice, routing overhead is amortized by the subsequent VLM call and does not affect user-perceived latency.

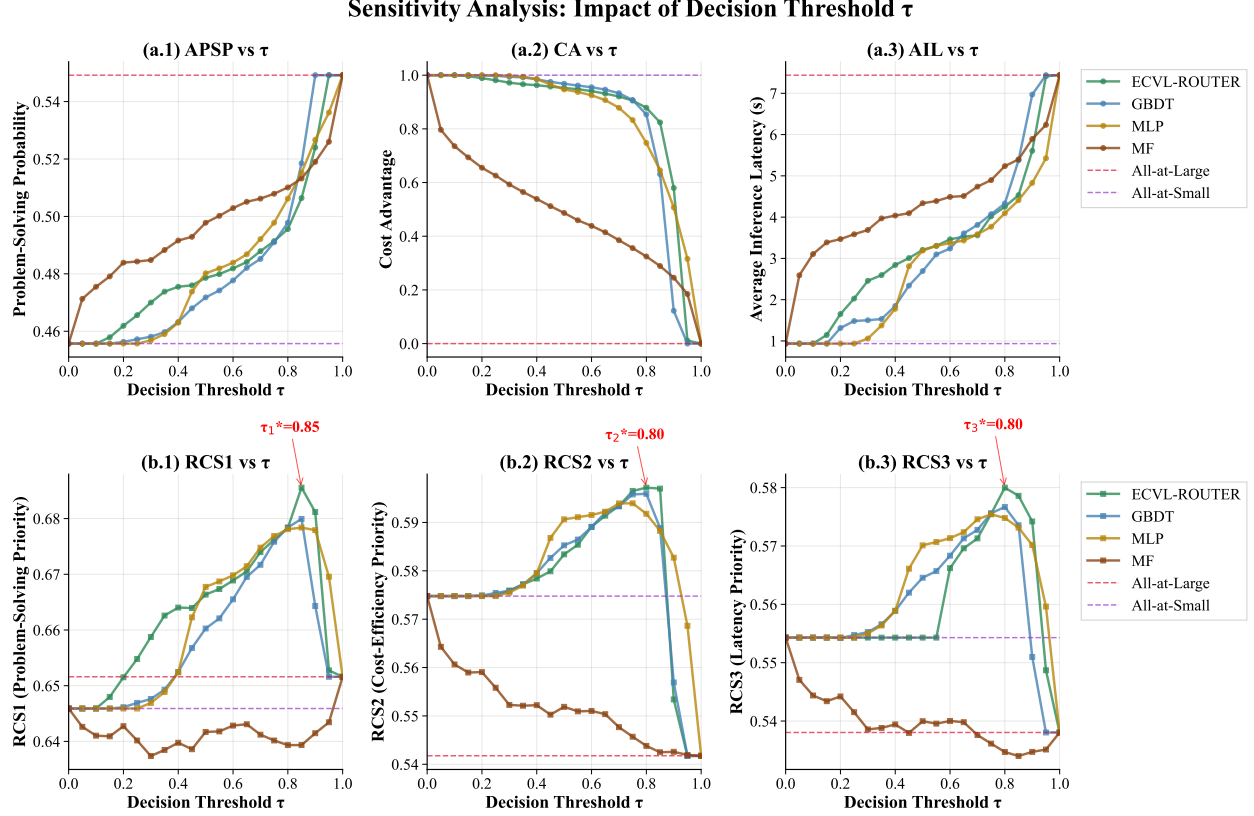


Figure 2: Impact of the decision threshold τ . It illustrates how the performance metrics for each model router (ECVL-ROUTER, MLP, GBDT, MF) change with different values of τ for the InternVL-38B/1B pair at MES=6.

Table 2: Comparison of Inference Latency Across Different Models.

	Router-Bert	Router-ViT	Router-Stat	Router-Fusion	Router-All	InternVL2.5-1B	InternVL3-8B	InternVL3-38B
Latency [s]	7.5×10^{-4}	0.0149	2.3×10^{-4}	8.8×10^{-6}	0.0159 (1.7% of 1B)	0.9359 (baseline)	1.6332 (1.75 $\times$ 1B)	7.4391 (7.95 $\times$ 1B)

Obs 6: Visual modality is the dominant driver in VLM routing. Our ablation studies, summarized in Table 3, reveal that visual features are not just beneficial but are the primary signal guiding the routing decision for VLM pairs. Removing any branch degrades all scenarios while dropping image hurts most (e.g., $\Delta\text{RCS2} = -0.022$, $\Delta\text{RCS3} = -0.017$); compared with removing text ($\Delta\text{RCS2} = -0.017$) and statistics yields smaller. Furthermore, among single-branch variants, image-only $>$ text-only on all composites and is competitive with statistics-only.

5.5 ALTERNATE ROUTING STRATEGY & METRICS

We reimplement two widely used routing paradigms: *win-hard* and *win-soft*, following the spirit of Hybrid LLM and RouteLLM.³ Table 4 reports per-scenario metrics (APSP/CA/AIL), the quality composite RCS1, router accuracy (ACC), the RouteLLM metric PGR, HybridLLM metric BARTScore, and aggregate Token/Time saving.

Obs 7: Scenario-Aware routing outperforms Single-Objective strategies. Our method ranks highest on the composite RCS1=0.6855 and on ACC (0.8408), and second on APSP/CA/AIL/PGR/BARTScore. It achieves the highest RCS1=0.6855 with APSP close to *win-hard* (0.5064 vs. 0.525), while routing more to the edge (CA +0.4233 $\uparrow$) and cutting latency (AIL -2.30s $\downarrow$). Against *win-soft*(1/2) (MES-shifted), it improves APSP/CA/AIL jointly (e.g.,

³All methods use our unified LRJ 1–10 scale for comparison

Table 3: Ablation of ECVL-ROUTER’s Components (Text, Image, Statistics) on the InternVL 38B/1B Model Pair.

Method	APSP $\uparrow$	CA $\uparrow$	AIL[s] $\downarrow$	RCS1 $\uparrow$	RCS2 $\uparrow$	RCS3 $\uparrow$
ECVL-ROUTER (Full)	0.5064	0.8241	4.5331	0.6855	0.6008	0.5820
w.o. Text	0.5079	0.6720	4.9529	0.6718	0.5836	0.5767
w.o. Image	0.5174	0.5496	4.7378	0.6711	0.5786	0.5653
w.o. Statistic	0.5150	0.6567	5.1550	0.6785	0.5886	0.5729
only Text	0.5152	0.5710	5.0925	0.6702	0.5786	0.5647
only Image	0.5174	0.5807	4.9666	0.6740	0.5821	0.5680
only Statistic	0.4894	0.9032	4.4539	0.6732	0.5933	0.5730
Random	0.4980	0.5000	4.1653	0.6434	0.5538	0.5418
All-at-Large	0.5492	0.0000	7.4391	0.6516	0.5418	0.5380
All-at-Small	0.4557	1.0000	0.9359	0.6459	0.5748	0.5543

Table 4: Offline comparison of routing strategies under a unified label/metric protocol. **Bold**=best; underline=second-best. *win-hard*: direct comparison: $\text{Score}_{\text{edge}} \geq \text{Score}_{\text{cloud}}$. *win-soft*(k): Score-offset variant favoring SVLM by $k \in 1, 2$; during labeling and decision the SVLM is judged against $\text{Score}_{\text{edge}} + k \geq \text{Score}_{\text{cloud}}$.

Metric	Proposed	win-hard	win-soft (1)	win-soft (2)
APSP $\uparrow$	0.5064	0.5250	0.5012	0.4784
CA $\uparrow$	<u>0.8241</u>	0.4008	0.7601	0.8609
AIL [s] $\downarrow$	<u>4.5331</u>	6.8286	5.1018	3.5520
RCS1 $\uparrow$	0.6855	0.6633	<u>0.6723</u>	0.6566
ACC $\uparrow$	0.8408	<u>0.8159</u>	0.8113	0.8011
PGR $\uparrow$	<u>0.7720</u>	0.9486	0.6001	0.3645
BARTScore $\uparrow$	<u>-4.8154</u>	-4.6750	-4.8869	-5.0040
Token saving $\uparrow$	28,269	9,341	25,292	33,136
Time saving $\uparrow$	<u>12,544</u>	4,858	10,324	14,717

vs. soft(1): APSP +0.0052 $\uparrow$, CA +0.0640 $\uparrow$, AIL −0.57s $\downarrow$). Notably, PGR is highest for *win-hard*, consistent with RouteLLM’s cost–performance emphasis, while our RCS objective aligns better with scenario-aware utility that balances quality, edge utilization, and latency. Hybrid LLM reports up to 40% fewer large-model calls at no quality drop under its thresholds; under the same unified LRJ regime, ECVL-ROUTER achieves stronger composite utility while maintaining high edge coverage. Furthermore, to assess real-world adaptability, we run an online study with $n=5$ participants over 200 image–text queries under two scenarios—Quality-first and Speed-first—using the same router trained offline. Full online study task analyses appear in Appendix B.3.

6 DISCUSSION

Point 1: Scenario-aware routing objective to align with user needs. We redefine routing as *satisficing* against a Minimal Expectation Score (MES): a response is “good enough” if it meets the user-specified minimum. This aligns routing with real application needs and maximizes the utility of on-device SVLMs by sending easy queries locally while escalating only when necessary. The approach is most effective when there is a clear capability gap between the SVLM and LVLM; when the SVLM is too weak (e.g., < 500M parameters) or the two models are similar in ability, gains from routing are limited. These findings offer actionable guidance for edge–cloud collaboration: set acceptance thresholds by user needs to minimize compute while maintaining satisfaction.

Point 2: Router flexibility and deployment practice. ECVL-ROUTER instantiates the above strategy with user-definable *MES* and threshold τ . Practitioners can construct training data with LRJ labels at a chosen *MES* and train with the composite metric RCS using weights (α, β, γ) over APSP/CA/AIL (see Sec. 3.2) to encode scenario preferences; the optimal τ is then selected on a validation set. In practice, we recommend curating in-domain data and tuning (α, β, γ) to the target scenario, as cross-domain generalization is modest. Relative to All-at-Large processing, our approach substantially increases edge model utilization with only small satisfaction drops, cutting both latency and cloud cost.

7 CONCLUSION

We introduced ECVL-ROUTER, a scenario-aware routing framework for vision–language models that reframes routing as meeting a user-defined Minimal Expectation Score (MES). By optimizing the Routing Comprehensive Score (RCS)—which balances quality (APSP), efficiency (CA), and latency (AIL), our method maximizes on-device small model usage while preserving response quality. Trained on the RSD dataset (~ 22 k image–text instances), ECVL-ROUTER consistently outperforms baselines across multiple model pairs, routing over 80% of queries to the edge with less than 10% drop in APSP and substantial end-to-end latency reductions. The decision threshold τ enables scenario-specific trade-offs; ablations show visual features dominate routing efficacy, and the router’s overhead (≈ 0.016 s) is negligible. The approach works best when edge and cloud capabilities differ markedly. Future work will focus on enhancing cross-domain generalization and extending the framework to other modalities.

ETHICS STATEMENT

This work develops a scenario-aware routing method for vision–language systems using publicly available datasets and open-source/base models. No personally identifiable information was collected or released; all data use complies with original licenses/terms. We label responses via an LLM-as-a-Judge rubric and perform human sanity checks to assess agreement and common biases (e.g., over-verbosity, modality imbalance), applying mitigation such as a dimensioned rubric, separation of judge/model, and conservative decision thresholds. Potential risks include unsafe escalation policies, amplification of dataset bias, and privacy leakage in cloud calls. We recommend deployment safeguards: scenario-appropriate MES settings, content-safety filters, rate/permission controls for remote inference, and clear documentation of intended use and known limitations. Code, configs, and evaluation scripts are released to support auditing and community oversight.

REPRODUCIBILITY STATEMENT

We release an anonymous repository containing code, configs, and scripts for dataset preparation, training, and evaluation, together with a readme that enumerates dependencies and exact commands. We document model pairs, routing architecture, hyperparameters, random seeds, and the 60/20/20 stratified splits; evaluation follows APSP/CA/AIL with composite RCS. Threshold selection uses a validation grid over τ ; we provide the sweep script and the best setting per scenario. Hardware/software details (e.g., NVIDIA A800 80 GB; Intel Xeon) and environment files are included. All reported tables/figures can be reproduced by running the provided pipelines; ablations and latency measurement scripts are also supplied. For reviewers, we include an anonymized code link and archival artifact with cached intermediate results to reduce compute and facilitate verification.

LLM USAGE

We used ChatGPT (OpenAI; Aug–Sep 2025 access window) for grammar checking and minor phrasing, and AI-assisted coding tools (e.g., Cursor) for refactoring boilerplate and editor suggestions during system development. All LLM outputs were reviewed by authors. We verified factual claims, math, code logic, citations, and figures; any errors were corrected by humans. We did not upload non-public data or PII to third-party services. Prompts contained only de-identified text or synthetic/task data.

REFERENCES

- Abdelrahman Abouelenin, Atabak Ashfaq, Adam Atkinson, Hany Awadalla, Nguyen Bach, Jianmin Bao, Alon Ben-haim, Martin Cai, Vishrav Chaudhary, Congcong Chen, et al. Phi-4-mini technical report: Compact yet powerful multimodal language models via mixture-of-loras. *arXiv preprint arXiv:2503.01743*, 2025.
- Pranjal Aggarwal, Aman Madaan, Ankit Anand, Srividya Pranavi Potharaju, Swaroop Mishra, Pei Zhou, Aditya Gupta, Dheeraj Rajagopal, Karthik Kappaganthu, Yiming Yang, et al. Automix: Automatically mixing language models. *arXiv preprint arXiv:2310.12963*, 2023.
- Elham Asgari, Nina Montaña-Brown, Magda Dubois, Saleh Khalil, Jasmine Balloch, Joshua Au Yeung, and Dominic Pimenta. A framework to assess clinical safety and hallucination rates of llms for medical text summarisation. *npj Digital Medicine*, 8(1):274, 2025.
- Peter Belcak, Greg Heinrich, Shizhe Diao, Yonggan Fu, Xin Dong, Saurav Muralidharan, Yingyan Celine Lin, and Pavlo Molchanov. Small language models are the future of agentic ai. *arXiv preprint arXiv:2506.02153*, 2025.
- Batyr Charyyev, Engin Arslan, and Mehmet Hadi Gunes. Latency comparison of cloud datacenters and edge servers. In *GLOBECOM 2020-2020 IEEE Global Communications Conference*, pp. 1–6. IEEE, 2020.
- Lianmin Chen, Matei Zaharia, and James Zou. Frugalgpt: How to use large language models while reducing cost and improving performance. *arXiv preprint arXiv:2305.05176*, 2023.
- Xiangxiang Chu, Limeng Qiao, Xinyang Lin, Shuang Xu, Yang Yang, Yiming Hu, Fei Wei, Xinyu Zhang, Bo Zhang, Xiaolin Wei, et al. Mobilevlm: A fast, strong and open vision language assistant for mobile devices. *arXiv preprint arXiv:2312.16886*, 2023.

-
- Dujian Ding, Ankur Mallick, Chi Wang, Robert Sim, Subhabrata Mukherjee, Victor Ruhle, Laks VS Lakshmanan, and Ahmed Hassan Awadallah. Hybrid llm: Cost-efficient and quality-aware query routing. *arXiv preprint arXiv:2404.14618*, 2024.
- Tao Feng, Yanzhen Shen, and Jiaxuan You. Graphrouter: A graph-based router for llm selections. *arXiv preprint arXiv:2410.03834*, 2024.
- Jared Fernandez, Clara Na, Vashisth Tiwari, Yonatan Bisk, Sasha Luccioni, and Emma Strubell. Energy considerations of large language model inference and efficiency optimizations. *arXiv preprint arXiv:2504.17674*, 2025.
- Christopher Fifty, Dennis Duan, Ronald G Junkins, Ehsan Amid, Jure Leskovec, Christopher Re, and Sebastian Thrun. Context-aware meta-learning. *arXiv preprint arXiv:2310.10971*, 2023.
- Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, et al. A survey on llm-as-a-judge. *arXiv preprint arXiv:2411.15594*, 2024.
- Zixu Hao, Huiqiang Jiang, Shiqi Jiang, Ju Ren, and Ting Cao. Hybrid slm and llm for edge-cloud collaborative inference. In *Proceedings of the Workshop on Edge and Mobile Foundation Models*, pp. 36–41, 2024.
- Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.
- SiYoung Jang and Roberto Morabito. Edge-first language model inference: Models, metrics, and tradeoffs. *arXiv preprint arXiv:2505.16508*, 2025.
- Nidhal Jegham, Marwan Abdelatti, Lassad Elmoubarki, and Abdeltawab Hendawi. How hungry is ai? benchmarking energy, water, and carbon footprint of llm inference. *arXiv preprint arXiv:2505.09598*, 2025.
- Olivier Lacombe, Kathleen Kenealy, Kat Black, Ravin Kumar, Francesco Visin, and Jiageng Zhang. Introducing gemma 3 270m: The compact model for hyper-efficient ai. Google Developers Blog, August 2025. URL <https://developers.googleblog.com/en/introducing-gemma-3-270m/>. Internal tests on a Pixel 9 Pro SoC show the INT4-quantized model used just 0.75% of the battery for 25 conversations.
- Xiang Li, Congcong Wen, Yuan Hu, Zhenghang Yuan, and Xiao Xiang Zhu. Vision-language models in remote sensing: Current progress and future trends. *IEEE Geoscience and Remote Sensing Magazine*, 12(2):32–66, 2024.
- Tianen Liu, Shuai Wang, Zheng Dong, Borui Li, and Tian He. From perception to computation: Revisiting delay optimization for connected autonomous vehicles. *ACM Computing Surveys*, 57(8):1–45, 2025.
- Mozhgan Navardi, Romina Aalishah, Yuzhe Fu, Yueqian Lin, Hai Li, Yiran Chen, and Tinoosh Mohsenin. Genai at the edge: Comprehensive survey on empowering edge devices. In *Proceedings of the AAAI Symposium Series*, volume 5, pp. 180–187, 2025.
- NVIDIA Corporation. Prompt task and complexity classifier. NGC model card, 2024. URL <https://catalog.ngc.nvidia.com/orgs/nvidia/teams/nemo/models/prompt-task-and-complexity-classifier>. Multi-head classifier for 11 task types and 6 complexity dimensions; used for prompt routing.
- Isaac Ong, Amjad Almahairi, Vincent Wu, Wei-Lin Chiang, Tianhao Wu, Joseph E Gonzalez, M Waleed Kadous, and Ion Stoica. Routellm: Learning to route llms with preference data. *arXiv preprint arXiv:2406.18665*, 2024.
- OpenAI. Gpt-5 system card. System card, OpenAI, August 2025. URL <https://cdn.openai.com/gpt-5-system-card.pdf>. Canonical PDF; see also the HTML overview at <https://openai.com/index/gpt-5-system-card/>.
- Ahmed Sharshar, Latif U Khan, Waseem Ullah, and Mohsen Guizani. Vision-language models for edge networks: A comprehensive survey. *IEEE Internet of Things Journal*, 2025.
- Gaurav Shinde, Anuradha Ravi, Emon Dey, Shadman Sakib, Milind Rampure, and Nirmalya Roy. A survey on efficient vision-language models. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 15(3): e70036, 2025.
- Karan Singhal, Shekoofeh Azizi, Tao Tu, S Sara Mahdavi, Jason Wei, Hyung Won Chung, Nathan Scales, Ajay Tanwani, Heather Cole-Lewis, Stephen Pfohl, et al. Large language models encode clinical knowledge. *Nature*, 620(7972):172–180, 2023.

-
- Karan Singhal, Tao Tu, Juraj Gottweis, Rory Sayres, Ellery Wulczyn, Mohamed Amin, Le Hou, Kevin Clark, Stephen R Pfohl, Heather Cole-Lewis, et al. Toward expert-level medical question answering with large language models. *Nature Medicine*, 31(3):943–950, 2025.
- Leslie N Smith and Nicholay Topin. Super-convergence: Very fast training of neural networks using large learning rates. In *Artificial intelligence and machine learning for multi-domain operations applications*, volume 11006, pp. 369–386. SPIE, 2019.
- Masao Someki, Yifan Peng, Siddhant Arora, Shinji Watanabe, Markus Müller, Thanasis Mouchtaris, Grant Strimel, and Jing Liu. Context-aware dynamic pruning for speech foundation models. 2025.
- Dimitris Stripelis, Zijian Hu, Jipeng Zhang, Zhaozhuo Xu, Alay Dilipbhai Shah, Han Jin, Yuhang Yao, Salman Avestimehr, and Chaoyang He. Tensoropera router: A multi-model router for efficient llm inference. *arXiv preprint arXiv:2408.12320*, 2024.
- Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.
- Pavan Kumar Anasosalu Vasu, Fartash Faghri, Chun-Liang Li, Cem Koc, Nate True, Albert Antony, Gokula Santhanam, James Gabriel, Peter Grasch, Oncel Tuzel, et al. Fastvlm: Efficient vision encoding for vision language models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 19769–19780, 2025.
- Xubin Wang, Zhiqing Tang, Jianxiong Guo, Tianhui Meng, Chenhao Wang, Tian Wang, and Weijia Jia. Empowering edge intelligence: A comprehensive survey on on-device ai models. *ACM Computing Surveys*, 57(9):1–39, 2025.
- Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, et al. Autogen: Enabling next-gen llm applications via multi-agent conversations. In *First Conference on Language Modeling*, 2024.
- Yuhang Yao, Haixin Wang, Yibo Chen, Jiawen Wang, Min Chang Jordan Ren, Bosheng Ding, Salman Avestimehr, and Chaoyang He. Toward super agent system with hybrid ai routers. *arXiv preprint arXiv:2504.10519*, 2025.
- Liangqi Yuan, Dong-Jun Han, Shiqiang Wang, and Christopher G Brinton. Local-cloud inference offloading for llms in multi-modal, multi-task, multi-dialogue settings. *arXiv preprint arXiv:2502.11007*, 2025.
- Jingyi Zhang, Jiaying Huang, Sheng Jin, and Shijian Lu. Vision-language models for vision tasks: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 46(8):5625–5644, 2024.
- Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36:46595–46623, 2023.
- Yue Zheng, Yuhao Chen, Bin Qian, Xiufang Shi, Yuanchao Shu, and Jiming Chen. A review on edge large language models: Design, execution, and applications. *ACM Computing Surveys*, 57(8):1–35, 2025.
- LLMs are not authors. The authors take full responsibility for all content, results, and any errors.

A RESPONSE SCORE DATASET

A.1 MODELS AND DATASET COMPOSITION

Models. We evaluate a pool of VLMs that spans both large cloud models M_{cloud} (LVLM) and small edge models M_{edge} (SVLM), following the definitions in §3.

Table 5: Model roster by category.

Category	Models
Large (LVLM)	Gemma 3-27B; InternVL3-38B
Small (SVLM)	InternVL2.5-2B; InternVL2.5-1B; SmolVLM-256M; InternVL3-8B; Phi-4-Multimodal-5.6B; Qwen2.5-VL-7B

Dataset composition. RSD covers diverse tasks and difficulty levels to support MES-based scenario-aware routing. Table 6 reports *Core Task Type*, *Key Abilities*, dataset *Volume (k)*, and *Difficulty* (three levels: Easy/Medium/Hard). The total annotated instances are about $\sim 22\text{k}$ model–instance pairs, aligned with the discussion in the main text.

Table 6: Task coverage of RSD training/eval data. Volumes (in thousands) sum to $\sim 22\text{k}$ overall. Difficulty uses three discrete levels (Easy/Medium/Hard) to coarsely stratify instance hardness for MES analysis.

Dataset	Core Task Type	Key Abilities	Volume (k)	Difficulty
WildVision	Real-world VQA	open-ended reasoning, contextual understanding	0.5	Easy
ChartQA	Chart QA	Structured extraction, logic/arithmetic	2.5	Easy
GQA	Compositional VQA	Spatial reasoning, multi-step inference	12.0	Medium
VizWiz	Blind-assistance VQA	Noise robustness, answerability	4.3	Medium
MMVet	Composite benchmark	Recognition/OCR/knowledge/spatial/math	0.22	Medium
MMMU-Pro	Professional hard VQA	Domain knowledge, deep reasoning	1.73	Hard
MMStar	Leak-resistant eval	Fine-grained/counterfactual, vision reliance	1.5	Hard
Total			22.7	—

A.2 LLM-AS-A-JUDGE PROMPT & BIASES CONTROL

Scoring Prompt. We adopt an *LLM-as-a-Judge* (LRJ) procedure to assign a unified *score* $\in [1, 10]$ per model–instance pair. The LRJ (GPT-4/4o in our setup) evaluates *accuracy*, *completeness*, *relevance*, and *level of detail* under a consistent rubric; the resulting 1–10 scale aligns with the scenario-specific MES threshold τ used by the router in §3. Human–LRJ agreement on a stratified sample demonstrates high correlation. We use the following template to elicit a scalar score followed by a short rationale:

SCORING_TEMPLATE_1_10 = """You are a helpful and precise assistant for checking the quality of multimodal AI responses.

[Question]
{question}

[Reference Answer]
{reference}

[Model Answer]
{prediction}

[System]
We would like to request your feedback on the performance of the AI assistant in response to the user question displayed above.

Please rate the helpfulness, relevance, accuracy, level of details of the response. The assistant receives an overall score on a scale of 1 to 10, where a higher score indicates better overall performance.

Scoring Guidelines:

- 1: Completely unable to answer the task, content is completely unrelated to the question, or refuses to answer
- 2: Attempts to answer but severely deviates from the topic, contains obvious understanding errors, completely incorrect information
- 3: Partially understands the question but answer quality is extremely poor, contains multiple major errors, illogical
- 4: Basically understands the question but answer is inaccurate, contains some important errors, lacks key information
- 5: Understands the question and provides relevant answer, but insufficient accuracy, obvious defects
- 6: Answer is basically correct, can execute the task but with average effect, passing line, some minor errors
- 7: Answer is accurate and has certain logic, can execute the task well, good quality
- 8: Answer is accurate, detailed and logically clear, almost no errors, excellent quality
- 9: Answer is complete, basically consistent with reference answer, high accuracy, clear and complete expression
- 10: Answer not only meets reference answer requirements, but also considers more aspects, more comprehensive, exceeds expectations

Please first output a single line containing only one value indicating the score for the model. In the subsequent line, please provide a comprehensive explanation of your evaluation, focusing on the accuracy, completeness, logic, and relevance of the response.

Score: ""

Bias Control Our scoring design explicitly addresses common biases:

- **Self-enhancement bias.** *Risk:* a model favors its own outputs. *Mitigation:* the evaluator (LRJ) is strictly disjoint from the evaluated models; we use GPT-4/4o solely as the judge.
- **Knowledge limitation bias.** *Risk:* judge hallucination or gaps hurt accuracy. *Mitigation:* provide a *Reference Answer* (ground truth) and a structured rubric focusing on correctness/relevance, reducing reliance on the judge’s prior.
- **Multimodal-specific biases.** *Risks:* (i) over-reliance on salient visual cues; (ii) inconsistency handling when text and image conflict. *Mitigation:* use a multimodal-capable judge (GPT-4o) and require dimension-wise assessment (accuracy, completeness, logic, relevance) to encourage cross-modal consistency checks.

A.3 GOLD-STANDARD VALIDATION (HUMAN EXPERT)

We randomly sample 200 items from RSD and collect eight-model responses per item ($200 \times 8 = 1,600$ LRJ–human pairs). Five anonymized experts (Expert1–5) independently score each item; their mean serves as the gold standard. LRJ aligns closely with humans: Pearson $r = 0.8805$, Spearman $\rho = 0.8349$ (both $p < 10^{-4}$). Agreement is high (79.83% within 1 point; 90.75% within 2 points), and mean inter-rater correlation is 0.8615. A paired t -test shows a small bias (human–LRJ $\Delta \approx -0.375$; $t = -23.93$; $p < 10^{-4}$). Per-evaluator correlations are reported in Table 7.

Table 7: Per-evaluator correlations between LRJ and individual experts on the 200-sample validation (1,600 pairs). Experts are anonymized as *Expert1–5*.

Evaluator	Pearson r
Expert1	0.883
Expert2	0.930
Expert3	0.894
Expert4	0.866
Expert5	0.903

A.4 SCORE AND LATENCY DISTRIBUTIONS

A.4.1 SCORE DISTRIBUTION

RSD aggregates $\sim 22k$ model-instance pairs spanning seven VLMs (SVLM $\leftrightarrow$ LVL) and eight benchmarks (Appendix A.1, Table 6). Figure 3 shows an approximately spread 1–10 score distribution with a mean of ≈ 5.6 and median ≈ 6.0 , consistent with our MES-aligned rubric. Figure 4 further decomposes scores: (a) dataset-level means reveal a clear difficulty gradient (MMStar hardest; ChartQA easiest), and (b) per-model boxplots indicate overlapping performance across scales, with several SVLMs competitive with LVLs. These statistics ground our scenario-aware routing analysis; latency distributions are reported in the next subsection.

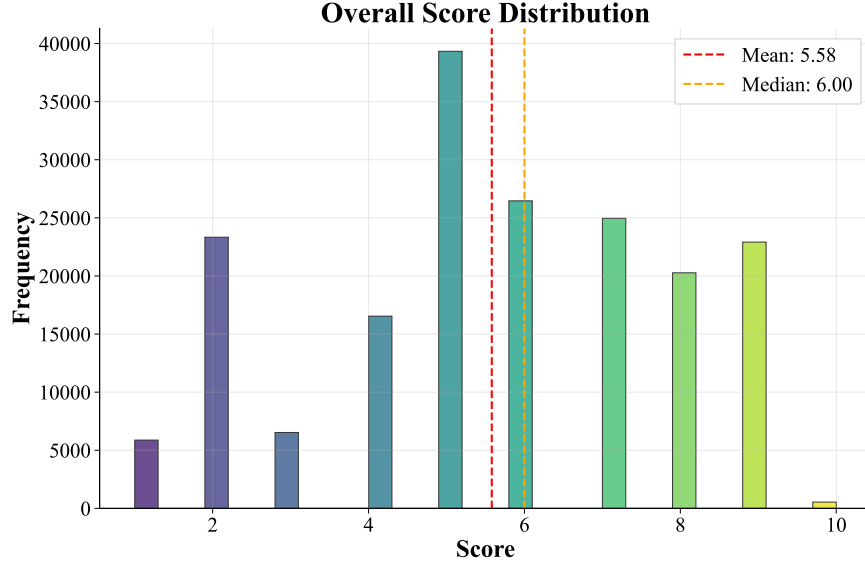
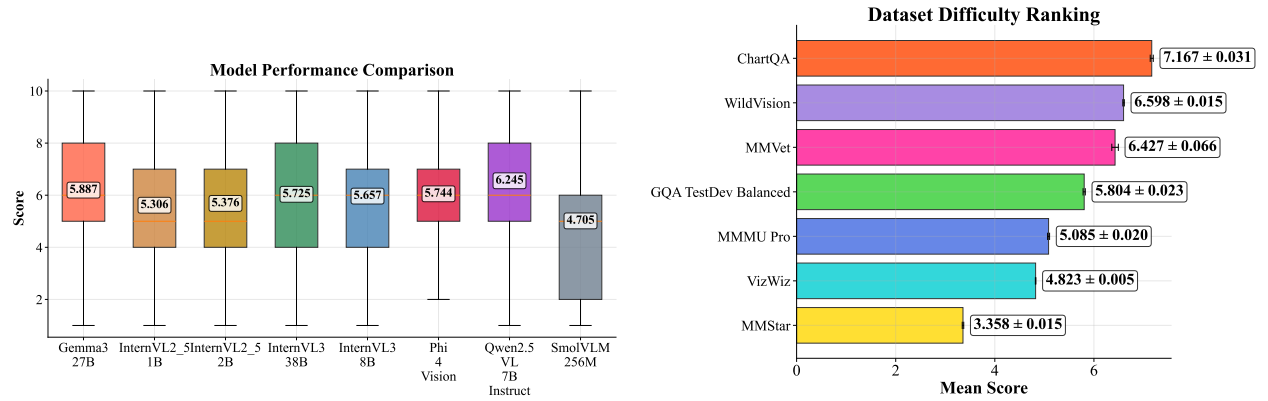


Figure 3: Overall score histogram across all models and datasets in RSD ($\sim 22k$ pairs). Dashed lines mark the mean (≈ 5.58) and median (≈ 6.00).



(a) Dataset difficulty ranking (mean $\pm$ SE). Higher mean $\Rightarrow$ easier. (b) Per-model score distributions (boxplots with annotated means).

Figure 4: Score breakdown in RSD: (a) difficulty by dataset; (b) performance by model across the same instances.

A.4.2 LATENCY DISTRIBUTION

Latency in RSD is markedly right-skewed (Figure 5). The median end-to-end time is $\approx 0.60s$ while the mean is $\approx 1.31s$, indicating a long tail: $P75 \approx 1.17s$, $P90 \approx 2.52s$, and $P99 \approx 5.45s$. Such headroom at the tail motivates scenario-aware routing that prefers M_{edge} when MES is met. Figure 6 decomposes latency by dataset and model. Datasets with

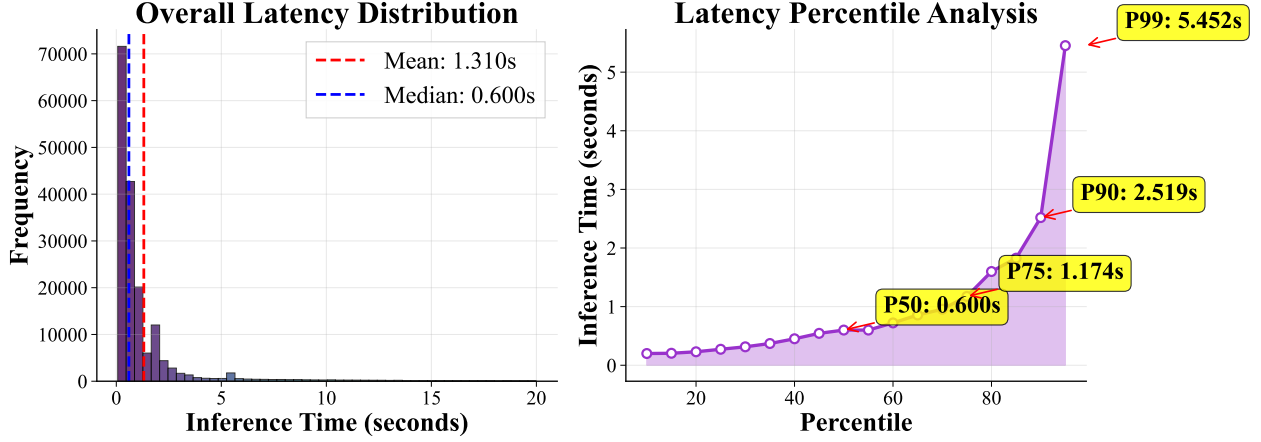
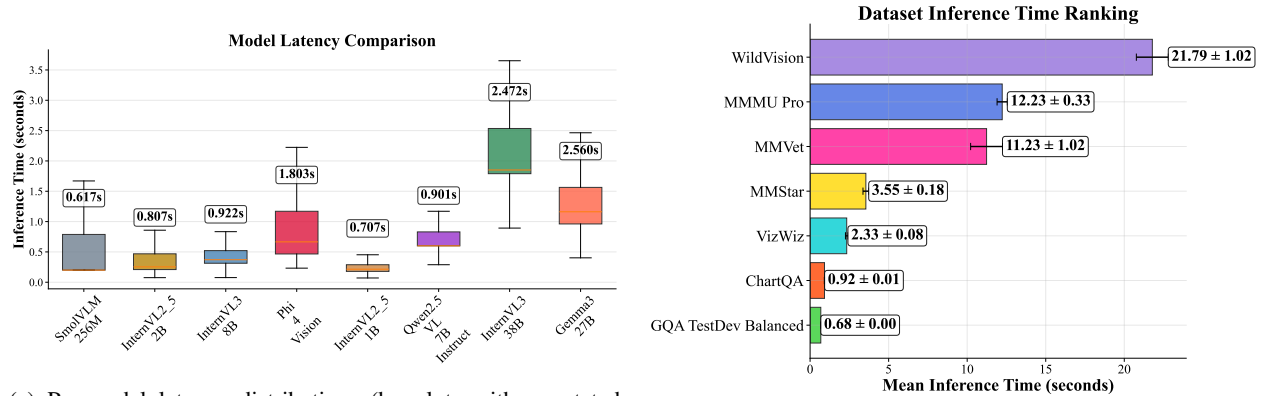


Figure 5: Overall latency histogram (left) and percentile curve (right) across all $\sim 22k$ model-instance pairs. Dashed lines mark the mean ($\approx 1.31s$) and median ($\approx 0.60s$); tail extends beyond 5s at P99.

heavy OCR or open-world reasoning (WildVision, MMVet, MMMU-Pro) are slowest, whereas structured QA (GQA, ChartQA) is fast. Across models, SVLMs achieve sub-second medians (e.g., SmolVLM-256M 0.62s; InternVL2.5-1B 0.71s; 2B 0.81s), while LVLMs are slower (InternVL3-38B 2.47s; Gemma3-27B 2.56s). These trends, aligned with Table 6, provide the latency side of our quality-speed trade-off for routing.



(a) Per-model latency distributions (boxplots with annotated means).

(b) Dataset mean latency (error bars: standard error).

Figure 6: Latency breakdown in RSD: (a) model-level comparisons on the same instances; (b) dataset-level inference time ranking.

A.4.3 ANALYSIS HEATMAP

To examine the accuracy-speed-stability trade-off on the same RSD instances, we summarize per-(model, dataset) statistics in a four-panel heatmap (Figure 7). Panel (a) shows mean scores (higher is better); (b) shows median latency in seconds (lower is better); (c) reports efficiency as *score/time* (higher is better); and (d) reports score standard deviation (lower is better). Overall, SVLMs attain strong efficiency on structured sets (ChartQA, GQA), while LVLMs deliver higher absolute scores on harder sets (MMVet, MMStar) at a latency cost. WildVision and MMMU-Pro display both large latency and higher variability, matching the difficulty/latency trends reported above.

Model-Dataset Performance Heatmaps Analysis

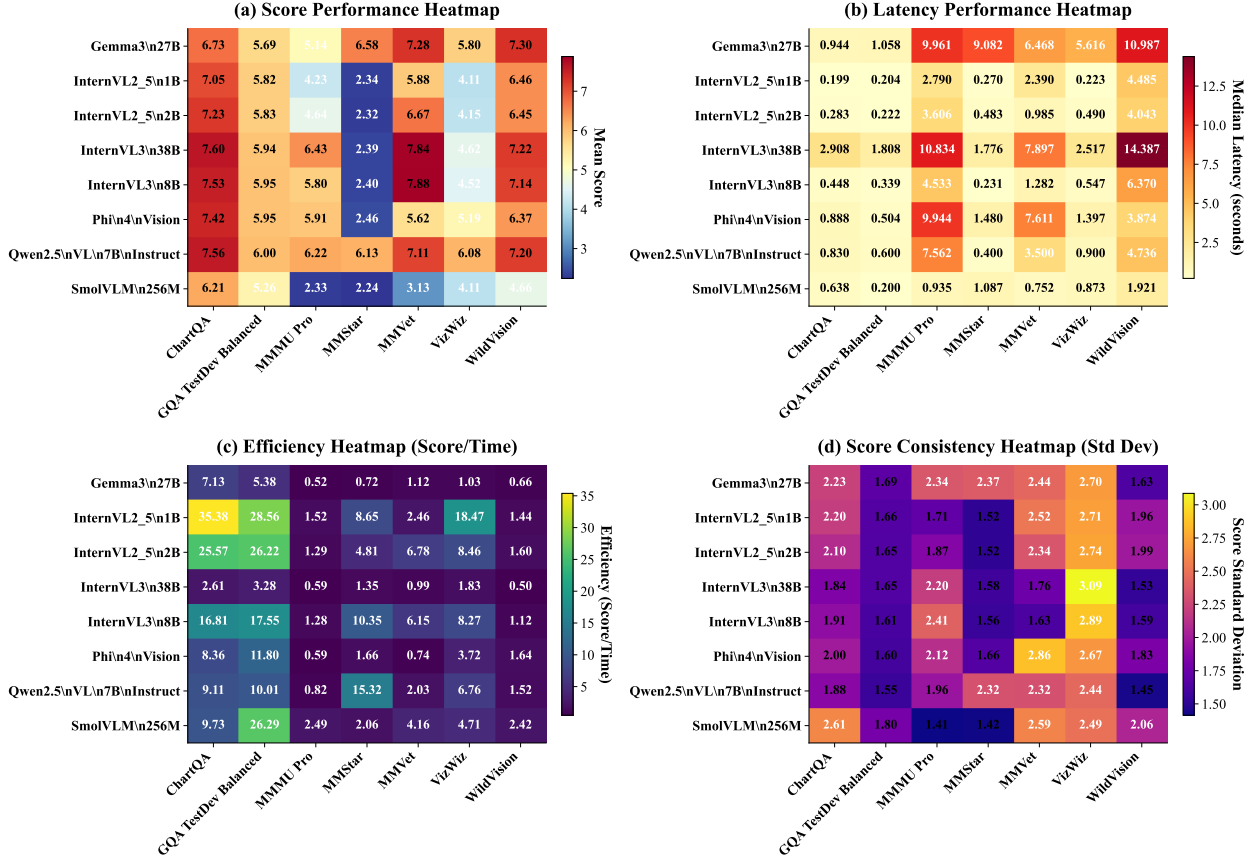


Figure 7: **Model-Dataset Heatmaps.** (a) Score performance (mean score). (b) Latency performance (median seconds). (c) Efficiency (*score/time*). (d) Score consistency (standard deviation): a “hotter” plasma color (toward bright/purple) indicates larger fluctuation; the cell value is the exact std. dev., and white numbers highlight cells with variance *above the global average*. *How to read:* scan a *row* to compare one model’s stability across datasets; scan a *column* to compare how challenging a dataset is across models.

B ADDITIONAL EXPERIMENTS

B.1 ADAPTABILITY ACROSS MODELS

We further supplemented our study with additional model-pair experiments to demonstrate the generality of ECVL-Router. As shown in Figure 8, when the small model is reasonably strong, ECVL-Router yields a clear improvement in RCS scores—for example, in the case of the Gemma3-27B / InternVL2.5-1B pair. We also experimented with a significantly smaller model, SmolVLM-0.2B, and found that when the performance of the small model is too weak, the model router routes the vast majority of queries to the large model. In such cases, the routing strategy offers little practical benefit in cost savings. Therefore, the smaller model must have sufficient problem-solving capability for the model router to be effective when selecting large-small model pairs.

B.2 ROUTER GENERALIZATION

In this work, the value of τ is selected via a grid search on the validation set, and the chosen τ is subsequently used together with the ECVL-Router for the model routing task. The figure 9 illustrates how the RCS on both the validation and test sets vary as τ changes. It can be observed that the trends of RCS variations on the validation and test sets are largely consistent. The Pearson correlation coefficient between the RCS and the τ values on the validation and test

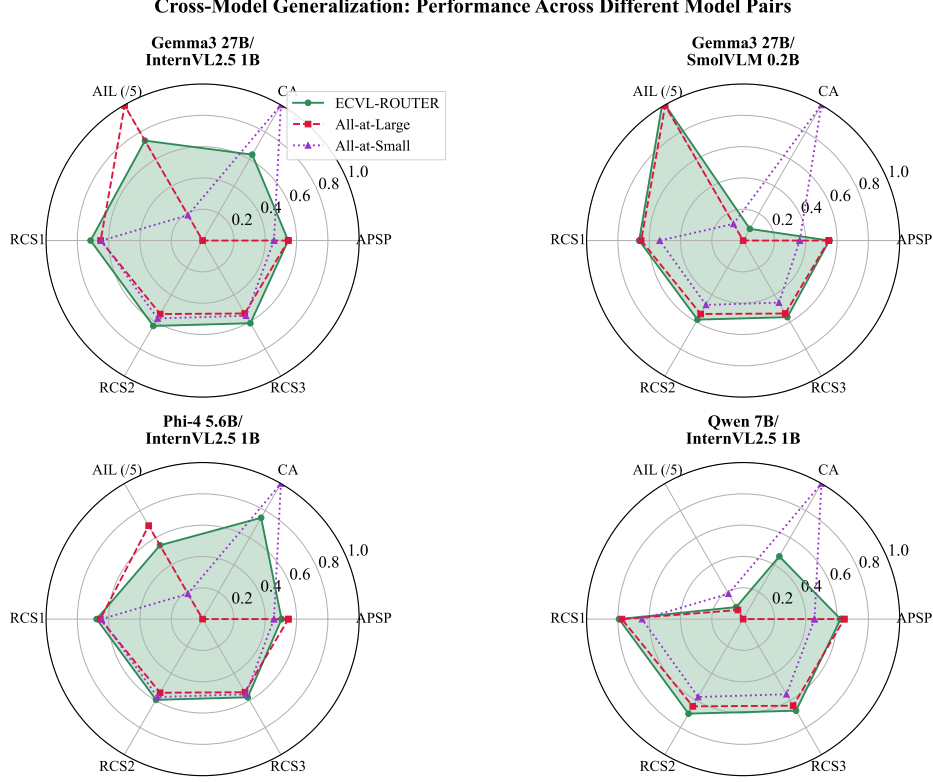


Figure 8: **Cross-model generalization across four model pairs.** Radar plots compare *ECVL-ROUTER* (green, solid), *All-at-Large* (red, dashed), and *All-at-Small* (purple, dotted) on **APSP**($\uparrow$), **RCS1/2/3**($\uparrow$), **CA**($\uparrow$, share routed to Small), and **AIL/5**($\downarrow$; latency divided by 5 for scale). Larger radii indicate better performance for all axes *except* AIL, where larger means slower. When the Small model is reasonably capable (e.g., pairs with **InternVL2.5-1B**), *ECVL-ROUTER* attains higher RCS while keeping latency moderate and CA non-trivial. With a very weak Small model (e.g., the **SmolVLM-0.2B** pair), the router intentionally lowers CA (sending most traffic to the large model), yielding limited cost benefit but preserving quality.

sets is approximately 0.98, indicating a strong correlation. This strongly supports the validity of using the validation set to determine the optimal τ value.

Table 8: Results on cross-domain generalization.

Model Router	APSP	CA	AIL	RCS1	RCS2	RCS3
ECVL-ROUTER	0.8982	0.6316	18.02408891	1.123	0.9547	0.9343
GBDT	0.8380	0.9582	6.100296302	1.0953	0.9469	0.9247
MLP	0.8380	0.9758	5.119651497	1.0981	0.9500	0.9279
MF	0.8558	0.7702	7.893076085	1.0961	0.9403	0.9210
All-at-Large	0.9432	0.0000	24.44174668	1.1074	0.9188	0.9065
All-at-Small	0.8332	1.0000	4.235969758	1.0956	0.9490	0.9268

To evaluate cross-domain generalization, we use InternVL-38B/1B as the model pair. ChartQA, WildVision, MMMU, and MMVet (about 5k samples) serve as the *test* sets, while the rest of RSD is used for training/validation. As summarized in Table 8, *ECVL-ROUTER* attains higher RCS than either *All-at-Large* or *All-at-Small* across all three scenarios, demonstrating cross-domain transfer. Compared with in-domain results (Section 5.2), the gains are smaller (roughly 0.5–1.5% vs. 5–6%), suggesting routing effectiveness is domain-sensitive; thus, routing strategies and training data should align closely with the intended application domain.

Generalization Capability: Validation vs Test Performance

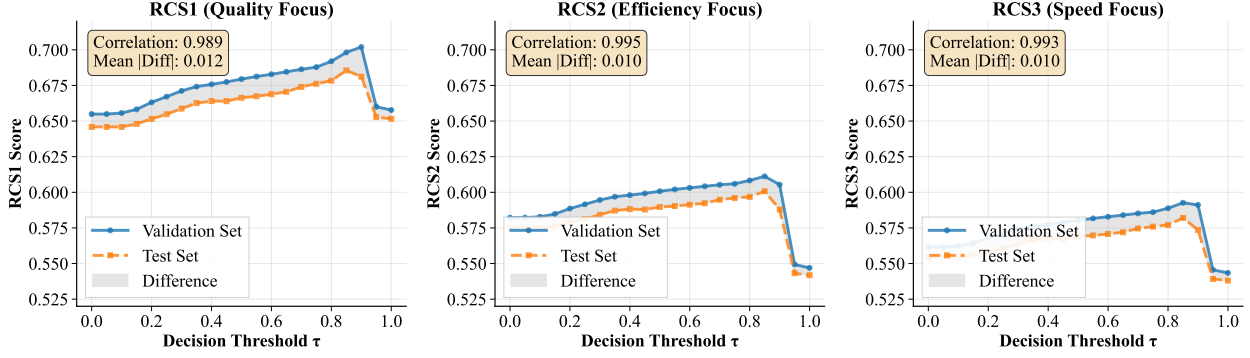


Figure 9: **Validation–test alignment for selecting the decision threshold τ .** RCS1/2/3 vs. τ on the validation set (solid blue) and test set (dashed orange). Pearson correlations between the two curves are 0.989 (RCS1), 0.995 (RCS2), and 0.993 (RCS3); mean absolute differences are 0.012/0.010/0.010. Performance increases with τ and peaks near $\tau \approx 0.85$ –0.90, then drops sharply as τ approaches 1.0, indicating over-conservative routing and validating grid-search selection on the validation set.

B.3 ONLINE USER STUDY

To evaluate the adaptability of our method across different real-world application scenarios, we conducted an online user study involving five participants who interacted with the model router on a dataset of 200 image–text pairs. Two experimental scenarios were designed:

Scenario 1: Users prioritize response quality, with the MES set to 7, and RCS1 used as the metric in validation stage.

Scenario 2: Users prioritize response speed, with MES set to 5, and RCS3 used as the metric in validation stage.

In each scenario, we trained an ECVL-Router using our routing strategy for online evaluation. For comparison, we also included three baselines: routing all queries to the large model, routing all queries to the small model, and training an ECVL-Router using the win-soft(1) routing strategy.

As shown in Figure 10, the average APSP and AIL results recorded by the five participants across the two scenarios. In Scenario 1, where response quality is emphasized, our routing strategy achieves higher APSP than win-soft(1), while its AIL is slightly higher. In Scenario 2, where response speed is prioritized, our method outperforms win-soft(1) on both APSP and AIL. This is because our training strategy is designed to maximize user satisfaction rather than to simply favor the stronger model. When MES is set to 5, a larger proportion of queries can already be satisfactorily handled by the small model; as a result, our method successfully routes more queries to the small model while maintaining a high level of user satisfaction.

These online experiments demonstrate that our routing strategy is highly flexible in practice, allowing users to balance response quality and response speed according to the requirements of different application scenarios.

B.4 UNEXPECTED QUERY ANALYSIS

Our router tends to favor the small model (SVLM) partly due to the dataset labeling rule: when both the large and small models fail to reach the user’s Minimal Expectation Score (MES), the sample is still labeled as “route to SVLM.” This yields a subset of inherently unsolvable queries in the SVLM bucket.

As shown in Figure 11, **Cost Advantage (CA)** is displayed as bars on the left y-axis, while the **failure rate** is the red line on the right y-axis. In the *safe zone* (MES = 1–4; green band, < 20% failures), most routed queries are solved and CA remains high (1.000/1.000/0.998/0.903). Entering the *caution zone* (MES = 5–6; yellow band), failures rise sharply (26.5% $\rightarrow$ 51.1%) and CA drops (0.844 $\rightarrow$ 0.792). In the *high-risk zone* (MES $\geq$ 7; red band), failures dominate (63.7%, 78.0%, 88.7%), with CA reaching its minimum at MES = 7 (0.744) and then partially recovering at MES = 8–9 (0.858/0.894).

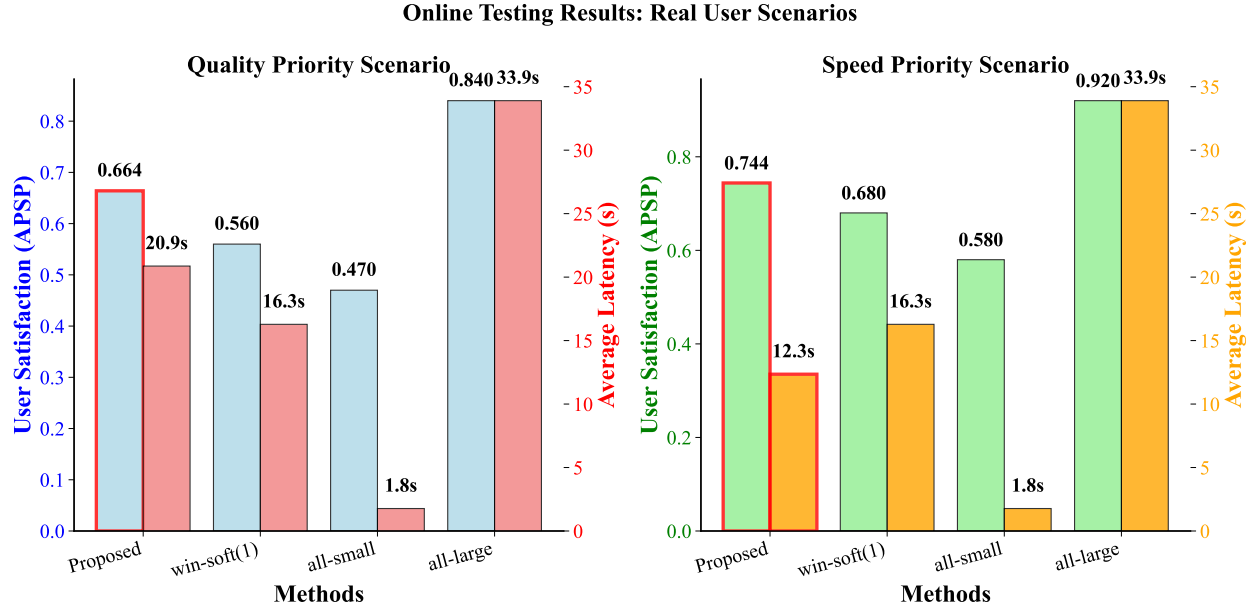


Figure 10: **Online user study across two scenarios.** APSP (left) and latency (s; right) for four routing policies. Our router adapts to goals: Quality-priority (MES=7/RCS1)—higher APSP than win-soft(1) at moderate latency; Speed-priority (MES=5/RCS3)—higher APSP with lower latency. All-large is highest quality but slowest; all-small fastest yet least satisfying.

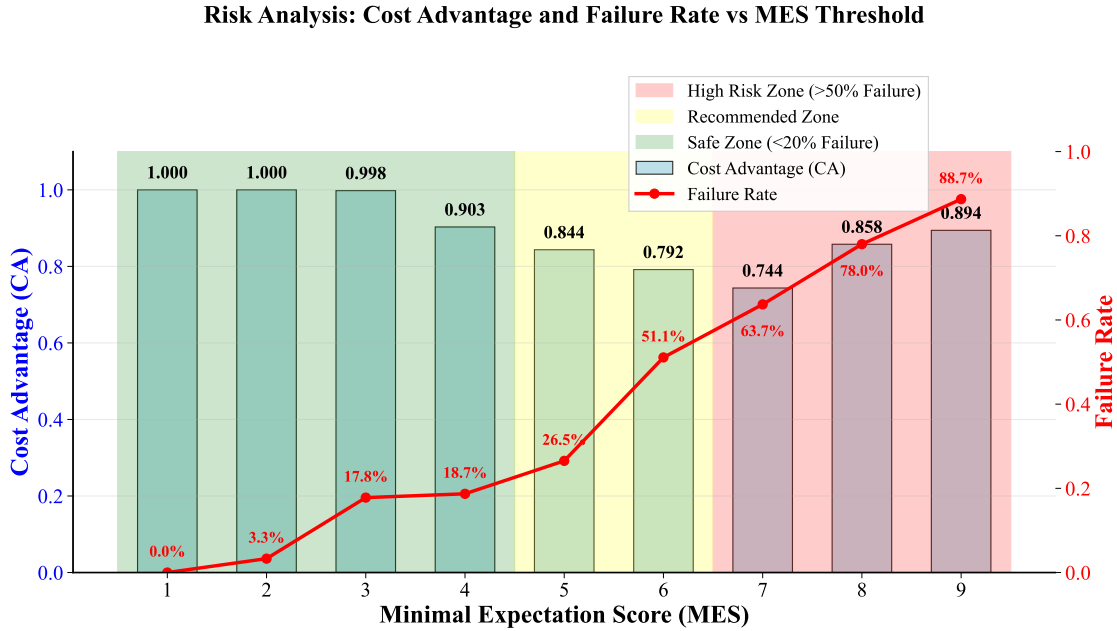


Figure 11: **MES risk map.** Failure rate (red line, right axis) climbs steeply as MES increases, defining a *safe* region (MES 1–4, < 20% failures), a *caution* region (MES 5–6), and a *high-risk* region (MES ≥ 7, > 50% failures). Cost Advantage (bars, left axis) stays near 1.0 at low MES, bottoms out at MES=7 (0.744), and partially recovers at MES=8–9, illustrating the trade-off between user expectation and routing cost.

B.5 ADDITIONAL PLOT

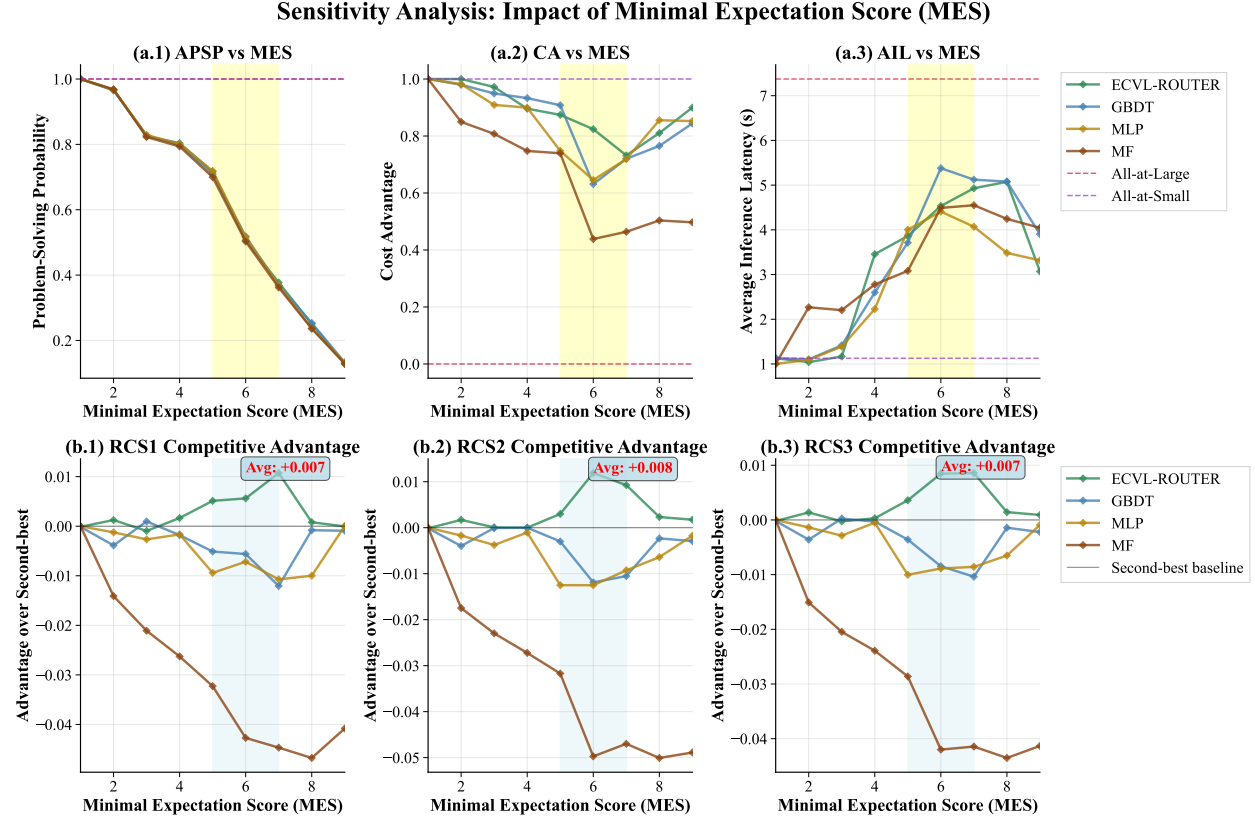


Figure 12: This figure shows the performance trends for all models and routing strategies as the MES value changes. The (a) plots the three metrics performance across different MES, and the (b) panel plots the performance gain of each router relative to the second-best method, as the baseline.

C EXAMPLE OF QUERY HARDNESS AND LLM JUDGE SCORE

C.1 EASY QUERY EXAMPLE

Input Query: (Fig 13 (a)) "What is the spatial relation between the frisbee and the man?"

InternVL3-38B:

VLM Response: "The frisbee is in the air, positioned above and slightly in front of the man. The man is crouched on the ground, reaching out towards the frisbee, indicating that he has likely thrown it or is preparing to catch it. The dog is jumping over the man, aiming to catch the frisbee mid-air."

LRJ Reason: "The model answer accurately describes the spatial relation between the frisbee and the man, stating that the frisbee is "in the air, positioned above and slightly in front of the man." This aligns well with the reference answer ("above the man (OR) top left") and provides additional context about the scene, such as the man's posture and the involvement of a dog. The extra details about the man's action and the dog's movement are relevant to understanding the scene but are not strictly necessary for answering the original question, which specifically asked about the spatial relation between the frisbee and the man. The answer is clear, logically structured, and free from errors, but it could be more concise and focused on the direct spatial relationship. Thus, it earns a score of 8 for being accurate, detailed, and well-expressed, though slightly more verbose than required."

LRJ Score: 8

InternVL3-8B:



(a) Image of easy query



(b) Image of hard query

Figure 13: An example of an easy query figure and a hard query figure.

VLM Response: "The frisbee is in front of the man and slightly above his head, moving in the air towards the left side of the image. The man appears to be throwing it or preparing to catch it. The dog is jumping in the air, aiming to catch the frisbee."

LRJ Reason: "The model answer is mostly accurate and relevant, correctly identifying that the frisbee is above the man and slightly in front of him, which aligns with the reference answer (above the man (OR) top left). The model also adds contextual details about the man possibly throwing or catching the frisbee and mentions a dog jumping to catch it. While this extra context is not required for the specific spatial relation question, it does not detract from the main answer. However, the answer could be more concise and focused on the direct spatial relation, as the reference answer is. The inclusion of "in front of the man" is a minor deviation, as the primary spatial relation should be "above" or "top left." Overall, the response is accurate and logical but could be improved by focusing more precisely on the spatial relationship asked."

LRJ Score: 7

InternVL2.5-1B:

VLM Response: "The frisbee is in the air, and the man is jumping to catch it."

LRJ Reason: "The model answer is relevant and generally correct in describing the scene: it states that the frisbee is in the air and the man is jumping to catch it. However, it does not directly answer the specific spatial relation question as clearly as the reference answer ("above the man" or "top left"). The response is descriptive but lacks the precise spatial terminology requested. While it gives context, it misses the concise spatial relationship, resulting in a passing but not high score."

LRJ Score: 6

C.2 HARD QUERY EXAMPLE

Input Query: (Fig 13 (b)) "Should I add sugar when mixing egg and milk?"

InternVL3-38B:

VLM Response: "No, according to the image, you should add sugar in the second step when you mix it with cocoa powder and other dry ingredients."

LRJ Reason: "The model answer is accurate, relevant, and provides additional helpful context beyond the simple "no" in the reference answer. It correctly states that sugar should not be added when mixing egg and milk, and further

clarifies that sugar is added in the second step with cocoa powder and other dry ingredients, presumably based on the referenced image. The explanation is logical and clear, enhancing the user's understanding of the process. However, since the answer relies on information from an image (not shown here), it assumes the user is following a specific recipe, which may not universally apply. For this reason, while the answer is excellent and detailed, it is not fully comprehensive for all contexts, so it does not receive a perfect score."

LRJ Score: 8

InternVL3-8B:

VLM Response: "No, according to the recipe, you should add the sugar during mixing, as shown in step 2."

LRJ Reason: "The model answer is inaccurate and contradicts the reference answer. The user asks whether sugar should be added when mixing egg and milk, and the reference answer is "no." However, the model answer incorrectly states that sugar should be added during mixing, referencing a "step 2" that is not present in the context. This introduces confusion and provides incorrect guidance. While the answer is somewhat relevant to the topic, it lacks accuracy and misleads the user, resulting in a low score. The response does not meet the requirements for correctness or clarity."

LRJ Score: 4

InternVL2.5-1B:

VLM Response: "Yes, you should add sugar when mixing egg and milk. It helps to create a smooth and creamy texture in the chocolate cake."

LRJ Reason: "The model answer is inaccurate and misleading. The user asked whether sugar should be added when mixing egg and milk, and the reference answer is "no." The model incorrectly asserts that sugar should be added and justifies this by mentioning the creation of a smooth and creamy texture in chocolate cake, which is irrelevant to the general question about mixing egg and milk. The response assumes a specific context (chocolate cake) that was not provided by the user and gives advice contrary to the reference answer. The answer lacks accuracy, is not relevant to the general question, and contains significant errors in logic and content."

LRJ Score: 2