VL-RouterBench: A Benchmark for Vision–Language Model Routing

Supplementary Material

A. Limitations

Here are some potential limitations of VL-RouterBench:

- **Single Image Input.** The current framework primarily considers single-image input scenarios for VLMs. This design limitation excludes more complex scenarios, such as multiple image inputs, which are relevant for many practical multimodal tasks like image captioning or image-based question answering that involve dynamic and large-scale images. Extending the benchmark to handle multi-image inputs could provide a more comprehensive evaluation of the routing system's capabilities
- **Binary Output Decision.** The routing decisions in the VL-RouterBench are based on a binary output model selection (correct or incorrect), while many real-world systems could benefit from more continuous score outputs. A more nuanced approach could involve probabilistic scoring, which would better capture the range of performance variation across models, improving the overall understanding of how well a model performs relative to another in a cost-sensitive manner.
- **Limited Evaluation of Cross-Modality.** Despite the extensive VLM dataset coverage across multiple domains, there is still a significant gap in evaluating cross-modal fusion mechanisms and multi-modal model capabilities. The benchmark primarily evaluates performance based on visual and textual data independently, but real-world applications often require complex fusion between multiple modalities, including audio, video, or even non-textual semantic information.
- **Robustness to Noisy or Imperfect Data.** VL-RouterBench is constructed from standardized logs with clean single-image prompts, and does not explicitly stress-test routers under noisy or imperfect inputs (e.g., typos, paraphrasing, or spurious visual artifacts). As a result, our evaluation primarily reflects routing performance under curated benchmark conditions and may overestimate robustness in real-world deployments where query quality is highly variable. Extending VL-RouterBench with perturbed or user-generated inputs and robustness-oriented metrics is an important direction for future work.

B. Implications

VL-RouterBench provides not only a standardized benchmark for VLM routing, but also several broader implications for the design of multimodal systems, routing algorithms, and future evaluation practice.

- **Implications for multimodal representation and router architecture.** Our results show that lightweight multimodal representations (frozen text and image encoders with simple fusion) already support competitive feature-level routers, while end-to-end VLC-style encoders further improve Rank Score at a modest throughput cost. This suggests that routing primarily relies on *coarse but discriminative* multimodal features, motivating router-specific encoders optimized for predicting which downstream model will succeed rather than for full answer generation.
- **Cost-aware training as a general recipe.** By casting routing as a multi-objective optimization over performance and cost and deriving a soft-label target that concentrates probability on correct yet cheaper models, VL-RouterBench provides a generic recipe for cost-aware training. The same Lagrangian-based construction can be reused in other multi-LLM or multi-VLM settings such as tool selection, CoT depth selection, or dynamic resolution control.
- **Towards more realistic multimodal routing workloads.** Built from large-scale VLMEvalKit logs over 14 datasets and 17 models, VL-RouterBench already approximates a realistic mixture of General, STEM, and Charts/OCR workloads, enabling offline prototyping of routing policies and their quality–cost–throughput trade-offs. It also encourages *workload-aware routing*, where router designs and model pools are tailored to domain-specific mixtures of difficulty and modality structure.
- **Bridging LLM routing benchmarks and multimodal routing.** VL-RouterBench extends existing LLM routing benchmarks into the multimodal regime, showing that ideas such as log-based evaluation, deferral curves, optimality analysis, and model-pool scaling transfer naturally but face new challenges from visual tokens and cross-modal alignment. This offers a shared conceptual and tooling foundation for unified routing frameworks across text-only LLMs, VLMs, and future audio or video models.

C. Ethics Statement

- **Data Sources and Privacy.** All data in VL-RouterBench originates from established vision-language benchmarks and the VLMEvalKit framework. We use only publicly available images, texts, and annotations under their respective licenses. The benchmark involves no human data collection, no user interaction, and no attempt to identify individuals. To our knowledge, the underlying datasets contain no personally identifiable information.
- **Methodological Scope.** Our routing approach operates purely at the model selection level and does not modify model internals or training data. While the benchmark itself involves no real-world deployment, we note that

selected models may inherit and potentially amplify biases, stereotypes, or harmful behaviors present in their original training corpora.

- **Use Restrictions.** We explicitly prohibit using VL-RouterBench or its components for optimizing routing in harmful applications, including discrimination, misinformation, or surveillance systems. Any practical deployment should incorporate independent safety mechanisms and domain-specific guardrails.

D. Proof of the Soft Label Strategy

We consider the routing framework for VLMs described in **Sec. 3.1**. Let the dataset be $\mathcal{D} = \{x_i = (I_i, T_i)\}_{i=1}^{N}$ and the candidate model set be $\mathcal{M} = \{m_j\}_{j=1}^{M}$. For each sample x_i, a router with parameters θ produces a distribution $\pi_\theta(\cdot \mid x_i)$ over models and selects

$$R_{\theta,i} = R(\theta, x_i) = \arg \max_{j \in \{1,\dots,M\}} \pi_\theta(m_j \mid x_i). \tag{A1}$$

For each sample–model pair (i, j), let $Y_{i,j}$ denote a performance indicator (*e.g.* correctness) and $C_{i,j}$ denote the inference cost. The global training objective can be written as the following multi-objective problem:

$$\min_{\theta} \underbrace{\mathbb{E}\left[-Y_{i,R_{\theta,i}}\right]}_{\text{performance risk}} + \lambda \underbrace{\mathbb{E}\left[C_{i,R_{\theta,i}}\right]}_{\text{expected cost}}, \tag{A2}$$

where $\lambda \geq 0$ controls the strength of the cost penalty.

To derive a cost-aware soft target, we first relax the hard decision $R_{\theta,i}$ to a distribution over models at each sample. For a fixed sample x_i, instead of choosing a single model, we consider a probability vector $q_i = (q_i(1), \dots, q_i(M)) \in \mathbb{R}^M, \sum_{j=1}^{M} q_i(j) = 1$. A natural sample-wise relaxation of **Eq. (A2)** is then

$$\min_{q_i \in \mathbb{R}^M} \sum_{j=1}^{M} q_i(j) \left(-Y_{i,j} + \lambda C_{i,j}\right), \tag{A3}$$

which trades off performance and cost in expectation at x_i.

To encourage non-degenerate, smooth distributions, we add an entropy regularizer with temperature $\tau > 0$:

$$\min_{q_i \in \mathbb{R}^M} \sum_{j=1}^{M} q_i(j) \left(-Y_{i,j} + \lambda C_{i,j}\right) \; - \; \tau H(q_i), \tag{A4}$$

where $H(q_i) = -\sum_{j=1}^{M} q_i(j) \log q_i(j)$ is the Shannon entropy. We now show that, under the assumption that incorrect models are forbidden, the unique minimizer of **Eq. (A4)** coincides with a cost-weighted soft label supported only on correct models.

Fix a sample x_i and let

$$\mathcal{S}_i = \{j \in \{1, \dots, M\} : Y_{i,j} = 1\} \tag{A5}$$

be the set of models that are correct on x_i. Assume $\mathcal{S}_i \neq \varnothing$ and that incorrect models are forbidden, *i.e.* $q_i(j) = 0$ whenever $Y_{i,j} = 0$. Consider the entropy-regularized problem

$$\min_{q_i \in \mathbb{R}^M} \sum_{j=1}^{M} q_i(j) \left(1 - Y_{i,j} + \lambda C_{i,j}\right) \; - \; \tau H(q_i), \tag{A6}$$

with $\tau > 0$. Then the unique minimizer $q_i^\star$ satisfies

$$q_i^\star(j) = \begin{cases} \dfrac{\exp(-\alpha C_{i,j})}{\sum_{k \in \mathcal{S}_i} \exp(-\alpha C_{i,k})}, & \text{if } j \in \mathcal{S}_i, \\ 0, & \text{otherwise}, \end{cases} \tag{A7}$$

where $\alpha = \lambda/\tau > 0$. In particular, $q_i^\star$ coincides with the **cost-aware soft label** $t_i^{(\lambda)}(j)$ under $\tau = 1$ defined by

$$t_i^{(\lambda)}(j) = \begin{cases} \dfrac{\exp(-\lambda C_{i,j})}{\sum_{k \in \mathcal{S}_i} \exp(-\lambda C_{i,k})}, & \text{if } Y_{i,j} = 1, \\ 0, & \text{otherwise}, \end{cases} \tag{A8}$$

since only correct models contribute to the denominator.

Proof. Because incorrect models are useless for training router, we may restrict q_i to be supported on $\mathcal{S}_i$. Define the restricted simplex

$$\Delta_i^+ = \{q_i : q_i(j) \geq 0,\ q_i(j) = 0 \; \forall j \notin \mathcal{S}_i\}. \tag{A9}$$

For $j \notin \mathcal{S}_i$ we have $Y_{i,j} = 0$, and under the constraint $q_i(j) = 0$ these terms do not affect the optimization. On $\mathcal{S}_i$, we have $Y_{i,j} = 1$ and therefore

$$1 - Y_{i,j} = 0, \qquad \forall j \in \mathcal{S}_i. \tag{A10}$$

Thus, on Δ_i^+ the objective in **Eq. (A6)** simplifies to

$$\tilde{\mathcal{L}}_i(q_i) = \sum_{j \in \mathcal{S}_i} q_i(j)\, \lambda C_{i,j} \; - \; \tau H(q_i), \tag{A11}$$

up to an additive constant that does not depend on q_i. Since $H(q_i)$ is strictly concave and the linear term in q_i is convex, the objective $\tilde{\mathcal{L}}_i$ is strictly convex on Δ_i^+, and hence has a unique minimizer.

We now compute this minimizer via Lagrange multipliers. Introduce a scalar multiplier μ for the normalization constraint $\sum_{j \in \mathcal{S}_i} q_i(j) = 1$. The Lagrangian is

$$\mathcal{J}(q_i, \mu) = \sum_{j \in \mathcal{S}_i} q_i(j)\, \lambda C_{i,j} - \tau H(q_i) + \mu\Big(\sum_{j \in \mathcal{S}_i} q_i(j) - 1\Big). \tag{A12}$$

Taking the derivative with respect to $q_i(j)$ for $j \in \mathcal{S}_i$ and setting it to zero yields

$$\frac{\partial \mathcal{J}}{\partial q_i(j)} = \lambda C_{i,j} - \tau(1 + \log q_i(j)) + \mu = 0. \tag{A13}$$

Rearranging gives

$$\log q_i(j) = \frac{\mu - \lambda C_{i,j}}{\tau} - 1, \tag{A14}$$

so that

$$q_i(j) = \exp\Big(\frac{\mu}{\tau} - 1\Big) \exp\Big(-\frac{\lambda}{\tau} C_{i,j}\Big) = Z_i^{-1} \exp(-\alpha C_{i,j}), \tag{A15}$$

where we set $\alpha = \lambda/\tau$ and $Z_i = \exp(1 - \mu/\tau)$ is a normalization constant independent of j. Imposing the simplex constraint $\sum_{j \in \mathcal{S}_i} q_i(j) = 1$ determines Z_i as

$$Z_i = \sum_{k \in \mathcal{S}_i} \exp(-\alpha C_{i,k}). \tag{A16}$$

Therefore the unique minimizer $q_i^\star$ is

$$q_i^\star(j) = \begin{cases} \dfrac{\exp(-\alpha C_{i,j})}{\sum_{k \in \mathcal{S}_i} \exp(-\alpha C_{i,k})}, & j \in \mathcal{S}_i, \\ 0, & j \notin \mathcal{S}_i, \end{cases} \tag{A17}$$

which establishes **Eq. (A7)**. Finally, note that

$$\sum_{k \in \mathcal{M}} e^{-\alpha C_{i,k}} = \sum_{k \in \mathcal{S}_i} e^{-\alpha C_{i,k}}, \tag{A18}$$

since all terms with $k \notin \mathcal{S}_i$ vanish after multiplying by the indicator $\mathbf{1}_{Y_{i,k}=1}$ in the definition of $t_i^{(\lambda)}(j)$. Hence $q_i^\star(j)$ coincides with $t_i^{(\lambda)}(j)$ under $\tau = 1$, completing the proof. □

E. Pareto Frontier Fitting

To rigorously evaluate the trade-off between inference cost and generation quality, we analyze the Pareto Frontier of the routing system. While RouterBench [8] proposes a geometric approach based on the Non-Decreasing Convex Hull (NDCH), we implement a parametric exponential saturation model to characterize the diminishing returns of computational investment and estimating the theoretical performance ceiling.

We approximate the Pareto frontier using a smooth exponential saturation function. Let x denote the average cost and y denote the average accuracy. We fit the empirical data to the following non-linear equation:

$$y(x) = a \cdot (1 - e^{-b \cdot x}) + c \tag{A19}$$

where c represents the floor performance approximated by the minimum performance in the Pareto set. $a + c$ represents the asymptotic maximum performance that the routing strategy could theoretically achieve given an infinite budget. b is the shape parameter and a larger b indicates high marginal utility in the low-cost regime, characterizing a highly efficient router.

We perform curve fitting using the Trust Region Reflective algorithm to minimize the least squares error. To ensure robust convergence, we initialize the parameters based on the empirical Pareto points, and the goodness of fit is validated using the coefficient of determination.

F. Detailed Information about the Datasets

F.1. General

- **MMBench** [14] provides a robust evaluation of 20 fine-grained skills using 3,217 meticulously balanced questions. It introduces a novel CircularEval strategy and uses GPT-4 for answer matching to ensure a comprehensive and robust assessment of multi-modal abilities.
- **MMStar** [15] is an elite "vision-indispensable" multi-modal benchmark comprising 1,500 meticulously human-selected samples, covering 6 core capabilities and 18 detailed axes. The dataset is designed to rigorously evaluate Large VLMs by eliminating samples that can be solved without visual input, thereby addressing data leakage and ensuring true multi-modal competency.
- **MMMU** [16] evaluates multimodal models on massive multi-discipline tasks demanding college-level knowledge and deliberate reasoning. It comprises 11.5K questions across six core disciplines and 30 heterogeneous image types, challenging models with expert-level content akin to professional exams.
- **RealWorldQA** [17] is a benchmark designed for evaluating basic real-world spatial understanding capabilities of multimodal models. The dataset comprises 765 images.
- **InfoVQA** [18] benchmarks the understanding of complex infographics, requiring joint reasoning over layout, textual content, and graphical elements. With 30,035 questions across 5,485 images, it specifically challenges models to perform elementary reasoning on diverse data visualizations, bridging the gap between computer vision and document understanding.
- **HallusionBench** [39] stands out as a diagnostic suite tailored to dissect "Language Hallucination × Visual Illusion" failure modes. Comprising 1,129 handcrafted VQA pairs, it utilizes a unique structure of original versus expert-modified images to test visual dependency rigorously. This approach moves beyond simple accuracy to quantitatively analyze specific failure dimensions.

F.2. STEM

- **MathVista** [19] combines diverse mathematical and visual challenges to evaluate fine-grained visual understanding and compositional reasoning. Integrating 28 existing and three new datasets, it covers 6,141 examples across varied visual contexts and seven distinct reasoning types.

- **MathVision** [20] offers 3,040 high-quality visual math problems meticulously selected from 19 math competitions across 12 grades. Strictly cross-validated by experts, it balances open-ended and multiple-choice formats, covering 16 mathematical disciplines to measure reasoning depth.
- **MathVerse** [21] assesses genuine visual diagram understanding by transforming problems into six versions with varying multi-modal information. It features 2,612 problems across geometry and functions, employing a novel Chain-of-Thought (CoT) evaluation strategy for fine-grained reasoning assessment.
- **AI2D** [22] addresses diagram interpretation using over 5,000 grade-school science diagrams. It uniquely employs Diagram Parse Graphs (DPG) to model syntactic structure and semantic relationships. With 150,000 rich annotations and 15,000 multiple-choice questions, it challenges models to decode visual symbolism—such as arrows indicating process flow versus consumption—beyond simple object recognition.

F.3. Chart OCR

- **TextVQA** [25] challenges models to read and reason about text embedded within images to answer questions. Comprising 45,336 human-generated questions across 28,408 text-rich images (e.g., billboards, signs) from the Open Images dataset, the benchmark requires integrating optical character recognition (OCR) with semantic reasoning to address complex queries involving scene text.
- **ChartQA** [23] addresses the limitations of synthetic datasets by evaluating visual and logical reasoning on 20,882 real-world charts collected from diverse online sources. It combines 9,608 human-written questions with 23,111 questions generated from summaries, overcoming issues of fixed vocabularies and template-based queries to support complex reasoning tasks involving aggregation and comparison.
- **DocVQA** [24] drives a "purpose-driven" approach to document analysis, featuring 50,000 questions across 12,767 diverse industry documents (e.g., forms, letters) from the UCSF Library. It challenges models to move beyond simple text extraction, requiring the interpretation of complex layouts, handwriting, and non-textual elements to retrieve information effectively.
- **OCRBench** [26] is a comprehensive benchmark tailored to evaluate the OCR capabilities of Large Multimodal Models. It comprises 1,000 manually verified question-answer pairs spanning five diverse components: text recognition, scene-text VQA, document VQA, key information extraction, and handwritten mathematical expressions. This design rigorously tests models on identifying and reasoning with text across varied visual contexts, from artistic fonts to complex documents.

G. Detailed Information about the Models

G.1. Open-source models

- **DeepSeek-VL2** [41] is a Mixture-of-Experts vision–language model featuring a total of 27B parameters, with roughly 4.5B parameters activated during inference. It offers a strong and competitive open-source baseline, serving as a higher-capacity counterpart to its Tiny variant.
  Official release: https://github.com/deepseek-ai/DeepSeek-VL2
- **DeepSeek-VL2-Tiny** [41] is a cost-efficient MoE variant with 27B total parameters but only 1B activated per token. It maintains reasonable multimodal performance while significantly reducing computational cost.
  Official release: https://github.com/deepseek-ai/DeepSeek-VL2
- **Gemma3-27B** [48] is a 27B dense VLM based on the Gemma3 architecture, offering strong general multimodal performance and serving as a competitive upper-tier open-source model.
  Official release: https://github.com/google-deepmind/gemma
- **InternVL2.5-78B** [49] is a 78B high-capacity VLM with advanced vision processing and strong reasoning ability. It is the largest model in our open-source pool and enriches the high-end of the Pareto frontier.
  Official release: https://github.com/OpenGVLab/InternVL
- **Janus-Pro-1B** [40] is a lightweight 1B VLM designed for cost-sensitive scenarios. Its compact transformer and ViT structure make it suitable as an extremely efficient routing candidate.
  Official release: https://github.com/deepseek-ai/Janus
- **Janus-Pro-7B** [40] is a 7B dense VLM equipped with a larger language backbone and a more capable vision encoder than the 1B variant, representing a standard mid-sized open-source model.
  Official release: https://github.com/deepseek-ai/Janus
- **Kimi-VL-A3B-Thinking-2506** [43] is a reasoning-optimized MoE VLM with 2.8B activated parameters. Although efficient, it often produces longer output sequences due to its "thinking-mode" tuning.
  Official release: https://github.com/MoonshotAI/Kimi-VL
- **LLaVA-Next-Vicuna-7B** [36] is a popular LLaVA-style baseline built on Vicuna-7B with a CLIP-like vision encoder. It performs well on standard VQA and everyday multimodal tasks.
  Official release: https://github.com/LLaVA-VL/LLaVA-NeXT
- **MiMo-VL-7B-RL** [45] is a 7B VLM improved with

reinforcement learning to encourage more deliberate reasoning. It provides an alternative training paradigm compared with purely supervised LLaVA models.
Official release: https://github.com/InternLM/MiMo

- **Phi-3.5-Vision** [44] is a vision-augmented version of the Phi-3.5 language model, using a SigLIP-style vision encoder. It demonstrates strong efficiency and STEM-oriented reasoning performance.
Official release: https://github.com/microsoft/Phi-3
- **Pixtral-12B** [47] is a 12B dense VLM offering competitive performance in document understanding and visual reasoning. It provides a middle-high capacity option between 7B and 27B models.
Official release: https://huggingface.co/mistralai/Pixtral
- **Qianfan-VL-8B** [46] is an 8B instruction-tuned VLM from Baidu Qianfan, filling the performance gap between 7B and 12B models and enhancing diversity in the mid-range model pool.
Official release: https://qianfan.cloud.baidu.com
- **Qwen2.5-VL-32B-Instruct** [27] is a 32B VLM equipped with resolution-adaptive visual tokenization, making input cost sensitive to image size. It provides high accuracy and strong general reasoning.
Official release: https://github.com/QwenLM/Qwen2.5-VL
- **Qwen2.5-VL-72B-Instruct** [27] is a 72B flagship model offering state-of-the-art open-source multimodal reasoning. It is one of the most powerful open-source models in our evaluation.
Official release: https://github.com/QwenLM/Qwen2.5-VL
- **SmolVLM2** [42] is a 2.2B compact VLM combining a CLIP-based vision encoder with a lightweight language model. It strikes a balance between cost and performance and strengthens the low-to-mid capacity tier.
Official release: https://huggingface.co/SmolVLM

G.2. API models

- **Gemini-Flash-2.5** [50] is a closed-source multimodal model accessed through API, optimized for high throughput and low latency. Token usage and cost follow the official API billing.
Documentation: https://ai.google.dev/gemini-api
- **GPT-4o** [51] is a frontier multimodal model from OpenAI providing top-tier performance with relatively low input cost but high output-token cost. Token usage is taken directly from API response metadata.
Documentation: https://platform.openai.com/docs

H. Detailed Information about the Routing Methods

H.1. Router Strategy

- **K-Nearest Neighbors (KNN)** [52] We implement a similarity-based router using the KNN algorithm, extending the text-based routing approach to the multimodal domain. The core basis is that queries sharing similar visual and textual semantic features should be best served by the same underlying VLM.
- **Pairwise Preference Aggregated KNN (PRkNN)** [53] Unlike the standard KNN router which relies on a single "hard" label per training sample, discarding information about second-best options or close calls, PRkNN utilizes the full correctness matrix and cost matrix of the retrieved neighbors to perform a fine-grained pairwise comparison.
- **One-vs-Rest (OVR)** [54] We implement a decompositional routing strategy based on the One-vs-Rest (OVR) framework. OVR treats the routing problem as a set of K independent binary classification tasks, where K is the number of candidate models in the pool. It assumes that the "best" model is the one with the highest independent probability of correctness, effectively maximizing the expected accuracy.
- **K-means** [52] We implement a similarity-based routing approach using K class means, where K is the number of models and a nearest centroid classifier. Instead of storing all training samples, this method aggregates the multimodal features of queries and images best suited for a specific model into a single representative vector. This method significantly reduces inference latency, as it requires only K dot products, rather than searching the entire training database.
- **Linear** [52] We implement a parametric routing baseline, Linear, which treats the model selection task as a standard multi-class classification problem. Linear learns a global linear decision boundary in the multimodal feature space. Given the fused feature vector $\mathbf{x} \in \mathbb{R}^d$, the router computes a probability distribution $\mathbf{p}$ over the K candidate models via a linear transformation followed by the Softmax function.
- **Multi-Layer Perceptron (MLP)** [52] We implement a non-linear parametric router using a Multi-Layer Perceptron (MLP), which serves as a more expressive alternative to the linear baseline [52]. This method learns a non-linear mapping from the multimodal feature space to the model selection probability distribution, capturing complex interactions between visual and textual semantics.
- **Cosine Classifier (CosineCls)** [4] We implement a metric-based routing approach that learns a shared embedding

space for both queries and candidate models, leveraging the architecture and Sample-LLM contrastive learning objective proposed in the Cosine Classifier framework [4] to VLM Router. Cosine method fine-tunes a Transformer encoder, to align user queries directly with learnable model prototypes based on semantic suitability.

- **Dual-Contrastive Router(RouterDC)** [4] We implement the RouterDC framework to VLM Router, which extends the previously described Cosine Classifier by incorporating a dual-contrastive learning objective. While the Cosine Classifier focuses solely on aligning queries with model prototypes (Sample-LLM alignment), RouterDC introduces an auxiliary Sample-Sample objective to regularize the embedding space by preserving the semantic manifold structure of the queries. We implement the RouterDC framework to VLM Router, which extends the previously described Cosine Classifier by incorporating a dual-contrastive learning objective. While the Cosine Classifier focuses solely on aligning queries with model prototypes (Sample-LLM alignment), RouterDC introduces an auxiliary Sample-Sample objective to regularize the embedding space by preserving the semantic manifold structure of the queries.
- **Zooter** [33] We implement Zooter, an end-to-end routing framework that fine-tunes a pre-trained Transformer encoder to predict optimal model selection directly from raw text and vision queries, eliminating the need for fixed, pre-extracted embeddings. The router employs a standard Transformer encoder architecture followed by a classification head. Zooter supports both hard and soft label training regimes to balance performance and cost.

H.2. Fusion Method

Let the visual embedding be denoted as $\mathbf{v} \in \mathbb{R}^{d_v}$ and the textual embedding as $\mathbf{q} \in \mathbb{R}^{d_q}$. We implement the following fusion strategies to construct a unified representation $\mathbf{f}$ for the routing policy.

- **Concatenation** This method preserves the full information from both modalities by joining the vectors along the channel dimension. The fused representation is defined as:

$$\mathbf{f}_{\text{concat}} = [\mathbf{v}; \mathbf{q}] \in \mathbb{R}^{d_v + d_q}, \tag{A20}$$

  where $[\cdot;\cdot]$ represents the concatenation operation.
- **Normalized Concatenation** To mitigate scale discrepancies between the visual and textual encoders, we apply L_2 normalization to each embedding prior to concatenation:

$$\mathbf{f}_{\text{norm-concat}} = \left[\frac{\mathbf{v}}{\|\mathbf{v}\|_2} ; \frac{\mathbf{q}}{\|\mathbf{q}\|_2} \right]. \tag{A21}$$

- **Weighted Average** To allow the model to adaptively prioritize one modality (e.g., text-dominant vs. image-dominant queries), we introduce a learnable scalar coefficient $\alpha \in [0, 1]$:

$$\mathbf{f}_{\text{w-avg}} = \alpha \mathbf{v} + (1 - \alpha)\mathbf{q}. \tag{A22}$$

- **Gated Multimodal Unit (GMU)** [64] This approach uses a learnable gating mechanism to control the information flow from each modality per sample. We compute a gate vector $\mathbf{z}$ via a sigmoid activation σ:

$$\mathbf{z} = \sigma(W_z^v \mathbf{v} + W_z^q \mathbf{q} + \mathbf{b}_z). \tag{A23}$$

  The final fused representation combines the transformed features using $\mathbf{z}$ as a soft switch:

$$\mathbf{f}_{\text{GMU}} = \mathbf{z} \odot \tanh(W_v \mathbf{v}) + (1 - \mathbf{z}) \odot \tanh(W_q \mathbf{q}). \tag{A24}$$

  where $\odot$ denotes the element-wise product.
- **Multimodal Low-rank Bilinear (MLB) Fusion** [65] To capture multiplicative interactions between modalities efficiently, MLB projects the inputs into a common space and fuses them via element-wise multiplication (Hadamard product): To capture multiplicative interactions between modalities efficiently, MLB projects the inputs into a common space and fuses them via element-wise multiplication (Hadamard product):

$$\mathbf{h} = \sigma(W_v^T \mathbf{v}) \odot \sigma(W_q^T \mathbf{q}). \tag{A25}$$

  where σ is typically the hyperbolic tangent $(\tanh)$ function. The final routing prediction is obtained by passing this fused representation through a linear classifier:

$$\hat{y} = \text{softmax}(W_{\text{MLP}} \mathbf{h} + \mathbf{b}_{\text{MLP}}). \tag{A26}$$

H.3. Router Backbone

We use pretrained transformer models in multiple methods like Cosine Classifier [4], RouterDC [4], Zooter [33], etc. We select BERT [66], MobileNetv4 [67], VisualBERT [55], UNITER [56], LXMERT [57] and VILBERT [58] as different backbones for experiment and choose the best one to combine with these methods, which turns out to be LXMERT.

- **BERT** [66] We use BERT as the text-only backbone for our end-to-end router. BERT's key innovation is its ability to learn representations by conditioning on both left and right context through two pre-training objectives: Masked Language Modeling (MLM), which randomly masks 15% of input tokens and predicts the original vocabulary based on context, and Next Sentence Prediction (NSP), a binary classification task to predict whether sentence B follows sentence A. The special `[CLS]` token at the beginning of each sequence provides an aggregate representation, $\mathbf{h}_{[\text{CLS}]}$, which is used for classification to predict the optimal model index.
- **MobileNetV4** [67] We use MobileNetV4 as the image-only backbone for our end-to-end router. MobileNetV4's

key innovation is the Universal Inverted Bottleneck (UIB) block, which combines the classical Inverted Bottleneck (IB) and ConvNext-style block into a flexible component. The UIB adapts during architecture search, enabling it to function as an IB, ConvNext block, or Feed-Forward Network (FFN) to optimize performance across diverse hardware. To capture long-range dependencies without the overhead of traditional Self-Attention, MobileNetV4 introduces Mobile MQA (Multi-Query Attention), which shares keys and values across attention heads to reduce memory bandwidth demands while preserving global context modeling. This hybrid architecture strikes an excellent balance between latency and accuracy, making MobileNetV4 ideal for real-time routing with minimal inference overhead.

- **VisualBERT** [55] We use VisualBERT as a text-image backbone, representing "single-stream" multimodal architectures. VisualBERT's key feature is its unified modeling approach, where visual tokens (derived from images) are directly concatenated with text embeddings and processed through a single stack of Transformer layers. This design enables early fusion, allowing deep, self-attention-based interaction between text tokens and image regions at every layer. Compared to late-fusion models, VisualBERT captures fine-grained grounding more effectively, while leveraging pre-trained BERT weights to accelerate convergence.
- **UNITER** [56] We adopt UNITER, a unified Transformer model for joint image-text representation learning. Through pre-training on tasks like Masked Language Modeling, Image-Text Matching, and Word-Region Alignment (via Optimal Transport), it achieves robust cross-modal alignment. UNITER's universal representations enable strong zero-shot or fine-tuned performance on diverse downstream tasks using a single set of weights.
- **LXMERT** [57] We employ LXMERT as a representative dual-stream architecture for multimodal fusion. It processes vision and language inputs separately before fusing them through a cross-modality encoder. This design enables strong unimodal representation learning and excels in complex reasoning tasks like VQA.
- **VILBERT** [58] We utilize ViLBERT, a dual-stream architecture that extends BERT to multimodal data. It processes vision and language in separate Transformer streams, interacting through co-attentional layers. These layers use cross-modal attention (e.g., vision queries attend to language keys/values) to enable context-aware fusion while preserving modality-specific representations.

I. Implementation Details

All trainable routers in our benchmark share the same data pipeline and optimization protocol, independent of the specific architecture (feature–level routers or end-to-end routers). We start from the VL-RouterBench logs, which provide for each sample $x_i = (I_i, T_i)$ a binary or real–valued quality vector $Y_i \in \mathbb{R}^M$ over the M candidate models and a corresponding cost vector $C_i \in \mathbb{R}^M$. The full quality and cost matrices are denoted by $Y \in \mathbb{R}^{N\times M}$ and $C \in \mathbb{R}^{N\times M}$, where N is the number of samples. We use the pre-defined `train`/`dev`/`test` splits, with the ratio $|\mathcal{D}_{\text{tr}}| : |\mathcal{D}_{\text{dev}}| : |\mathcal{D}_{\text{te}}| = 7 : 1 : 2$.

To supervise the router, we derive either hard or soft training targets over the M experts from (Y_i, C_i) via the multi–objective formulation in **Sec. 3.3**. Concretely, we compute a target distribution that concentrates probability mass on models with high quality and low cost, controlled by a trade-off parameter λ. We use a set of $\lambda = [0.0, 10.0, 100.0, 1000.0, 10000.0, +\infty]$ in the experiments to control the trade-off between accuracy and cost. For feature-level routers we first compute a fixed text embedding for T_i and then train a shallow classifier on top. For end-to-end routers we attach a task-specific classification head to the final `[CLS]` or pooled multimodal representation of the visual–language backbone. Unless otherwise specified, we train all routers with AdamW, a learning rate of 2×10^{-5}, batch size 16, and weight decay 0.01, using a cosine decay schedule thereafter. We train for at most 5 epochs.

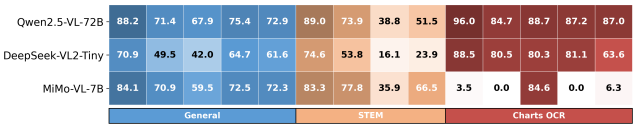

Figure A1. Accuracy heatmap on three representative models, Qwen2.5-VL-72B, DeepSeek-VL2-Tiny, and MiMo-VL-7B. The datasets are grouped into three task groups: General, STEM, and Charts OCR, shown in different colors.

J. Additional Experiments

J.1. Benchmark Dataset Analysis

We study how accuracy is distributed across task groups/datasets and models. As illustrated by **Fig. A1** on three representative models, VL-RouterBench contains both easy and hard samples rather than being saturated. Importantly, these models show clearly different strength profiles: Qwen2.5-VL-72B is consistently strong but expensive; DeepSeek-VL2-Tiny remains competitive on Charts OCR subsets despite weaker General and STEM; MiMo-VL-7B performs well on General and STEM but collapses on multiple Charts OCR subsets. These results demonstrate meaningful complementarity that routing can exploit.

J.2. Ablation Study on β in Rank Score

We visualize Rank Score under different β, including $\beta = 0.01, 0.1, 1.0, 10.0, 100.0$, which controls the trade-off between accuracy and cost. **Fig. A2** shows that Oracle con-

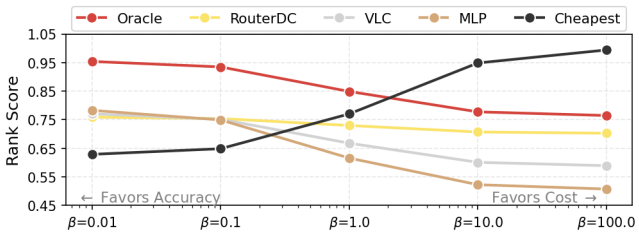

Figure A2. Rank Score comparison of different β values on Oracle, RouterDC, VLC, MLP, and Cheapest.

Table A1. Performance comparison of top routers on VL-RouterBench under multi-cost framework. Other settings are the same as in **Tab. 2**.

Router	Avg. Acc. ↑	Avg. Cost ↓	Avg. Mem. ↓	RS* ↑	Rank ↓
Oracle	95.47	0.41	12.20	93.69	0
RouterDC	70.40	0.72	16.35	70.48	1
VLC	65.62	0.35	11.32	68.09	2
MLP	65.54	0.49	12.72	67.38	3

sistently achieves the highest score, while learned routers outperform Cheapest in regimes where accuracy remains important. As β becomes large, scores shift predictably toward cost-minimizing routers, causing Cheapest to rise. We note that β serves as an application-oriented parameter: users with strict budget constraints may increase β, whereas those prioritizing quality may decrease it.

J.3. Extended Multi-cost Framework

To better reflect real deployment overhead, we extend VL-RouterBench with an additional cost metric: KV-cache memory (currently available for open-source models). Specifically, we construct a per-sample, per-model KV proxy from (i) logged token counts in inference records and (ii) model architectural KV parameters. Note that this metric is currently computed for open-source models where architectural details are accessible. Building on this, we formalize multi-cost routing as an optimization problem: maximizing expected correctness subject to both financial ($) and hardware (KV) resource constraints:

$$\min_{\theta} \ \mathbb{E}\big[-Y_{i,R_{\theta,i}}\big] + \lambda_C \, \mathbb{E}\big[C_{i,R_{\theta,i}}\big] + \lambda_M \, \mathbb{E}\big[M^{kv}_{i,R_{\theta,i}}\big],$$

which yields a Lagrangian with multiplier (λ_C, λ_M). Following the original soft-label derivation, we generalize the target distribution by replacing the single-cost energy with a multi-cost energy:

$$t_i^{(\lambda)}(j) = \frac{\mathbf{1}\{Y_{i,j}=1\}\exp(-E_{i,j}(\lambda))}{\sum_{m:Y_{i,m}=1}\exp(-E_{i,m}(\lambda))},$$
$$E_{i,j}(\lambda) = \lambda_C \tilde{C}_{i,j} + \lambda_M \tilde{M}^{kv}_{i,j},$$

where $\tilde{C}, \tilde{M}^{kv}$ are obtained via bounded log scaling for numerical stability and comparability across magnitudes. For

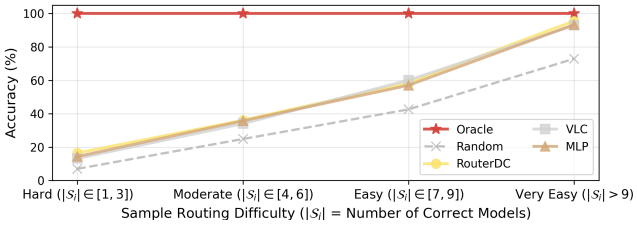

Figure A3. Accuracy distribution on different routing difficulty levels for top routers. $|\mathcal{S}_i|$ is the number of models correctly answering sample i.

Table A2. Performance comparison of learned routers on held-out datasets. Routers are trained on 11 seen datasets (**ID**) and evaluated on 3 held-out datasets (**OOD**). Other settings are the same as in **Tab. 2**.

Router	ID			OOD		
	Avg. Acc. ↑	Avg. Cost ↓	Rank Score ↑	Avg. Acc. ↑	Avg. Cost ↓	Rank Score ↑
Oracle	94.85	0.39	92.87	96.78	0.47	93.94
RouterDC	73.62	0.53	73.26	68.81	0.54	68.88
VLC	72.09	0.49	72.07	69.36	0.52	69.44
MLP	71.61	0.73	70.66	55.53	0.86	55.79

evaluation, we use a formal KV-cache memory metric, M^{kv} (MiB), which allows us to report Avg.Mem. alongside the existing Avg.Cost. We also define a Tri-RankScore RS* by combining $\bar{A}$, $\tilde{C}$, and $\tilde{M}^{kv}$ and using a weighted harmonic mean with (β_C, β_M)=$(0.1, 0.1)$:

$$\mathrm{RS}^*(\beta_C, \beta_M) = \frac{(1+\beta_C+\beta_M)\,\bar{A}\,C_{\text{norm}}\,M_{\text{norm}}}{C_{\text{norm}}M_{\text{norm}} + \beta_C\,\bar{A}M_{\text{norm}} + \beta_M\,\bar{A}C_{\text{norm}}}.$$

In experiments under this KV-aware protocol, results in **Tab. A1** show that router comparisons become more deployment-informative and RouterDC attains the best Tri-RankScore among learned routers. Importantly, any further cost metrics can be integrated into our multi-cost optimization framework through the same formalization process.

J.4. Oracle-Best Gap Study

To analyze the source of the Oracle-Best gap, we focus on routing-sensitive samples. We stratify instances by $|\mathcal{S}_i|$ (the number of models correctly answering sample i), which serves as a proxy for routing difficulty (smaller $|\mathcal{S}_i|$ means harder routing). As shown in **Fig. A3**, while top routers consistently outperform Random baseline, their performance drops sharply as difficulty increases, leaving substantial room for improvement relative to Oracle on hard-route samples. This indicates that the Oracle–Best gap is primarily driven by routers' limited ability to extract sufficiently discriminative multimodal signals when only a few specialized models can solve the instance. These hard-route cases therefore keep our benchmark challenging and motivate future work on stronger multimodal routing.

J.5. Generalization on Held-out Dataset

To investigate generalization of learned routers, we introduce a held-out-dataset protocol: we select one unseen dataset

per task group (default: MMMU for General, MathVision for STEM, and ChartQA for Charts/OCR), train routers on the remaining 11 datasets, and evaluate on ID (test splits of the 11 seen datasets) versus OOD (test split of the 3 held-out datasets only). We then report the ID and OOD Avg. Acc., Avg. Cost, and Rank Score. Under this stricter setting, learned routers remain competitive on OOD and preserve meaningful behavior rather than collapsing, e.g., RouterDC achieves 68.88 OOD Rank Score generalized from 73.26 ID Rank Score. Results are shown in **Tab.** A2.

J.6. Results on Oracle and Models

We provide the detailed results of all candidate models and the Oracle router on VL-RouterBench in **Fig.** A4. The Oracle represents an ideal routing policy and achieves the highest average accuracy at very low average cost, which highlights the substantial potential of routing systems for improving the cost effectiveness of VLM deployments. It is also noteworthy that the two API models, despite their significantly higher prices, do not obtain better average accuracy than several cheaper open source candidates. This pattern poses a challenge for practical router design, since an effective routing strategy must explicitly avoid high cost options whose accuracy does not justify their price and instead maintain a favorable balance between accuracy and cost.

J.7. Detailed Results on Each Dataset

We further report per-dataset detailed results under different values of λ, as shown in **Tab.** A3-**Tab.** A8. These tables characterize the accuracy-cost trade-offs achieved by different routing methods across a wide range of λ, providing a fine-grained view of how each method adapts under varying cost preferences.

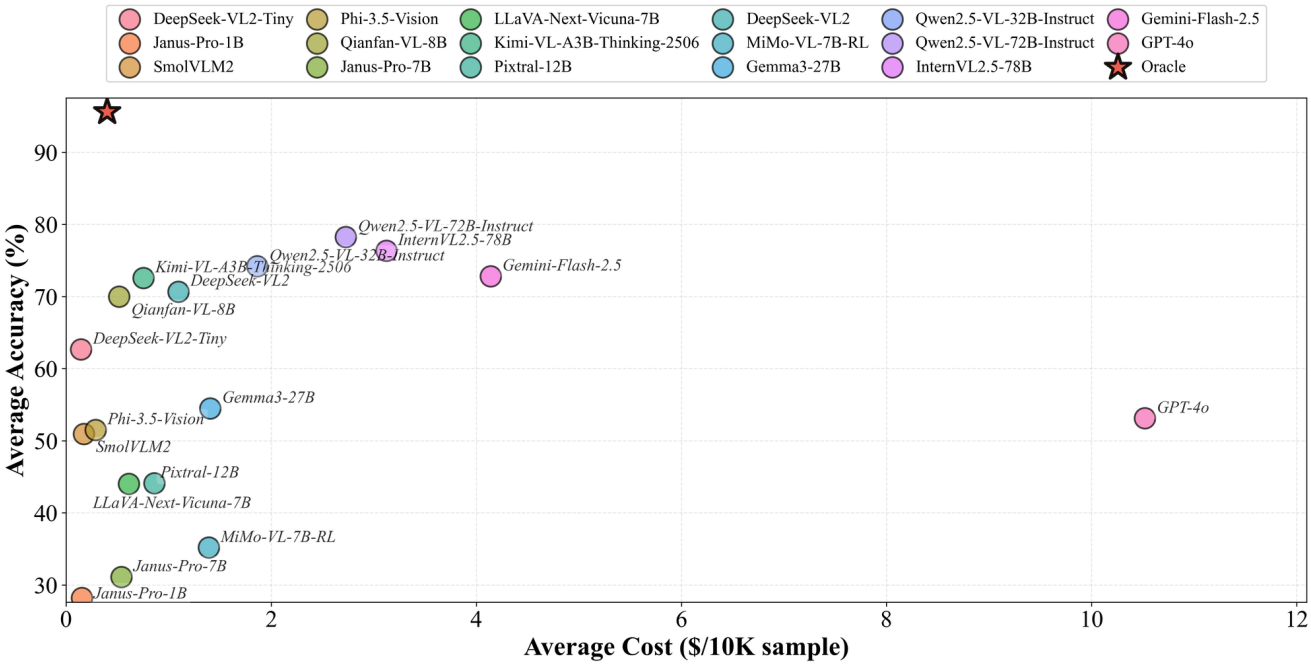

Figure A4. Accuracy-Cost Curve of Oracle and Models. The closer to the upper left corner, the better the accuracy-cost trade-off.

Table A3. Performance comparison of router methods on VL-RouterBench across datasets for $\lambda = 0.0$. **Avg. Acc.** is average accuracy (%), **Avg. Cost** is average cost ($/10K samples). The best and second-best results among learned routers are highlighted in **bold** and underlined, respectively (excluding Oracle and baseline methods).

Group	Dataset	Metric	Baselines			Hard Label				Soft Label ($\lambda = 0.0$)					
			Oracle	Strongest	Cheapest	KNN	PRkNN	OVR	K-means	Linear	MLP	CosineCls	RouterDC	VLC	ZOOTER
General	HallusionBench	Avg. Acc. ↑	99.46	71.89	57.84	40.54	43.24	66.49	70.27	68.65	68.65	79.46	**80.00**	67.03	71.89
		Avg. Cost ↓	0.23	2.97	0.14	**0.18**	0.21	1.67	2.89	2.17	3.02	4.18	2.82	3.03	2.16
	InfoVQA	Avg. Acc. ↑	96.49	83.28	53.68	58.36	61.87	81.10	80.60	77.93	81.10	80.60	81.10	**82.61**	80.77
		Avg. Cost ↓	0.37	2.19	0.14	**0.24**	0.29	1.49	2.09	1.66	1.73	0.99	1.12	1.20	1.61
	MMBench	Avg. Acc. ↑	98.16	89.86	69.59	69.59	81.11	85.25	70.97	82.95	87.56	**90.32**	90.32	87.10	84.79
		Avg. Cost ↓	0.24	2.22	0.13	**0.19**	0.27	2.26	1.17	1.55	1.90	5.33	2.52	1.84	1.57
	MMMU	Avg. Acc. ↑	98.46	74.36	45.64	42.05	53.33	60.00	65.64	67.69	76.92	**77.44**	76.41	75.38	72.31
		Avg. Cost ↓	0.64	3.24	0.17	**0.41**	0.43	6.95	10.59	11.33	13.22	17.54	7.06	12.91	10.65
	MMStar	Avg. Acc. ↑	96.68	72.76	49.17	54.48	59.80	68.44	66.11	63.46	70.76	**75.08**	72.76	69.44	74.75
		Avg. Cost ↓	0.36	2.29	0.14	**0.25**	0.29	2.85	1.68	1.76	2.78	7.98	8.26	2.58	2.78
	RealWorldQA	Avg. Acc. ↑	99.32	77.40	67.12	55.48	63.70	67.81	60.27	69.86	69.86	78.08	77.40	74.66	**80.14**
		Avg. Cost ↓	0.22	2.20	0.13	**0.14**	0.18	1.46	1.17	1.15	1.84	3.55	3.77	2.27	1.78
STEM	AI2D	Avg. Acc. ↑	99.69	88.21	72.33	67.45	73.58	80.66	76.42	76.10	83.80	**87.11**	86.95	82.70	84.43
		Avg. Cost ↓	0.24	2.19	0.13	**0.16**	0.19	2.05	2.56	1.20	2.94	7.17	2.10	2.02	2.40
	MathVerse	Avg. Acc. ↑	90.36	49.74	23.05	59.77	67.97	68.75	66.93	69.53	**71.09**	62.37	66.15	69.92	52.99
		Avg. Cost ↓	1.06	4.67	0.18	**1.64**	1.70	4.01	3.19	3.99	4.04	7.85	2.59	3.87	2.55
	MathVision	Avg. Acc. ↑	89.55	38.81	13.43	28.36	23.88	35.82	46.27	49.25	50.75	**61.19**	61.19	50.75	61.19
		Avg. Cost ↓	1.77	8.16	0.17	1.23	**0.72**	13.10	5.64	9.92	15.50	26.73	2.73	19.30	24.39
	MathVista	Avg. Acc. ↑	95.85	75.13	51.30	53.89	70.98	77.20	73.58	72.54	79.27	76.68	77.20	76.17	**79.79**
		Avg. Cost ↓	0.36	5.00	0.16	**0.50**	0.54	3.46	2.81	3.79	4.82	9.52	6.67	3.54	4.45
Chart & Doc	ChartQA	Avg. Acc. ↑	97.80	86.20	81.00	81.40	80.80	81.00	40.60	70.20	**83.80**	70.60	76.40	83.20	82.60
		Avg. Cost ↓	0.21	2.19	0.14	**0.14**	0.17	1.09	0.69	0.38	0.98	1.74	0.86	1.30	1.34
	DocVQA	Avg. Acc. ↑	98.56	91.48	78.95	78.66	80.48	85.55	82.30	80.19	88.13	87.85	88.04	87.37	**89.57**
		Avg. Cost ↓	0.18	2.18	0.14	**0.15**	0.18	1.22	1.25	0.58	1.34	0.98	0.21	1.34	1.41
	OCRBench	Avg. Acc. ↑	99.06	89.20	80.28	77.00	81.22	85.92	79.34	81.22	86.85	93.90	**94.37**	89.67	87.32
		Avg. Cost ↓	0.23	2.41	0.14	**0.16**	0.19	1.44	0.96	0.71	1.61	1.60	0.68	1.77	1.70
	TextVQA	Avg. Acc. ↑	89.46	73.85	72.13	70.98	72.13	67.05	66.09	68.29	71.26	67.62	68.49	**73.56**	69.73
		Avg. Cost ↓	0.18	2.15	0.13	**0.14**	0.16	1.12	1.81	0.33	1.86	1.31	0.84	1.41	0.78
Average	All Datasets	Avg. Acc. ↑	95.60	78.01	62.43	66.26	70.68	75.49	70.01	73.08	78.62	77.19	78.24	**78.65**	76.70
		Avg. Cost ↓	0.37	2.72	0.14	**0.38**	0.41	2.18	2.21	1.82	2.75	3.37	1.14	2.53	2.32
		Rank Score ↑	93.68	68.88	64.63	67.13	71.09	68.92	64.62	68.23	69.19	65.90	**74.81**	70.04	69.36

Table A4. Performance comparison of router methods on VL-RouterBench across datasets for $\lambda = 10.0$. **Avg. Acc.** is average accuracy (%), **Avg. Cost** is average cost ($/10K samples). The best and second-best results among learned routers are highlighted in **bold** and underlined, respectively (excluding Oracle and baseline methods).

Group	Dataset	Metric	Baselines			Hard Label				Soft Label ($\lambda = 10.0$)					
			Oracle	Strongest	Cheapest	KNN	PRkNN	OVR	K-means	Linear	MLP	CosineCls	RouterDC	VLC	ZOOTER
General	HallusionBench	Avg. Acc. ↑	99.46	71.89	57.84	40.54	43.24	66.49	70.27	**73.51**	65.95	61.08	58.38	70.81	65.41
		Avg. Cost ↓	0.23	2.97	0.14	0.18	0.21	1.67	2.89	3.13	0.98	**0.17**	0.19	0.90	1.49
	InfoVQA	Avg. Acc. ↑	96.49	83.28	53.68	58.36	61.87	81.10	80.60	72.41	78.60	57.19	65.05	**83.11**	80.43
		Avg. Cost ↓	0.37	2.19	0.14	**0.24**	0.29	1.49	2.09	0.87	1.31	0.24	0.43	2.00	1.72
	MMBench	Avg. Acc. ↑	98.16	89.86	69.59	69.59	81.11	85.25	70.97	82.95	**89.86**	77.42	79.72	88.02	89.86
		Avg. Cost ↓	0.24	2.22	0.13	**0.19**	0.27	2.26	1.17	2.44	1.08	0.24	0.28	0.93	1.62
	MMMU	Avg. Acc. ↑	98.46	74.36	45.64	42.05	53.33	60.00	65.64	**70.77**	58.46	48.21	51.79	62.05	60.00
		Avg. Cost ↓	0.64	3.24	0.17	0.41	0.43	6.95	10.59	11.64	1.59	0.33	**0.31**	1.08	1.16
	MMStar	Avg. Acc. ↑	96.68	72.76	49.17	54.48	59.80	68.44	66.11	67.44	69.10	56.81	59.14	69.77	**70.43**
		Avg. Cost ↓	0.36	2.29	0.14	0.25	0.29	2.85	1.68	2.33	1.05	**0.24**	0.24	0.83	1.33
	RealWorldQA	Avg. Acc. ↑	99.32	77.40	67.12	55.48	63.70	67.81	60.27	67.12	69.86	63.70	61.64	71.92	**79.45**
		Avg. Cost ↓	0.22	2.20	0.13	**0.14**	0.18	1.46	1.17	1.91	0.70	0.16	0.31	1.27	1.29
STEM	AI2D	Avg. Acc. ↑	99.69	88.21	72.33	67.45	73.58	80.66	76.42	79.56	83.02	73.58	72.80	**86.16**	84.28
		Avg. Cost ↓	0.24	2.19	0.13	**0.16**	0.19	2.05	2.56	2.20	1.10	0.17	0.21	1.50	1.37
	MathVerse	Avg. Acc. ↑	90.36	49.74	23.05	59.77	67.97	68.75	66.93	67.71	**69.92**	64.71	57.68	62.11	64.32
		Avg. Cost ↓	1.06	4.67	0.18	1.64	1.70	4.01	3.19	3.33	3.28	3.92	2.39	**1.52**	2.18
	MathVision	Avg. Acc. ↑	89.55	38.81	13.43	28.36	23.88	35.82	46.27	46.27	49.25	49.25	37.31	**52.24**	52.24
		Avg. Cost ↓	1.77	8.16	0.17	1.23	**0.72**	13.10	5.64	7.07	2.55	9.16	1.17	1.78	1.78
	MathVista	Avg. Acc. ↑	95.85	75.13	51.30	53.89	70.98	77.20	73.58	76.17	**81.87**	62.69	62.18	77.72	77.72
		Avg. Cost ↓	0.36	5.00	0.16	0.50	0.54	3.46	2.81	3.72	1.90	0.77	**0.44**	1.14	1.17
Chart & Doc	ChartQA	Avg. Acc. ↑	97.80	86.20	81.00	81.40	80.80	81.00	40.60	79.40	84.00	82.40	83.40	**87.00**	81.20
		Avg. Cost ↓	0.21	2.19	0.14	**0.14**	0.17	1.09	0.69	1.36	0.58	0.17	0.22	0.85	1.58
	DocVQA	Avg. Acc. ↑	98.56	91.48	78.95	78.66	80.48	85.55	82.30	82.78	85.07	79.14	81.24	**90.14**	88.52
		Avg. Cost ↓	0.18	2.18	0.14	**0.15**	0.18	1.22	1.25	0.62	0.65	0.15	0.21	1.59	1.20
	OCRBench	Avg. Acc. ↑	99.06	89.20	80.28	77.00	81.22	85.92	79.34	80.28	**88.26**	81.22	85.45	88.26	85.45
		Avg. Cost ↓	0.23	2.41	0.14	**0.16**	0.19	1.44	0.96	1.18	1.06	0.16	0.21	0.57	0.97
	TextVQA	Avg. Acc. ↑	89.46	73.85	72.13	70.98	72.13	67.05	66.09	66.38	**72.70**	72.41	70.59	70.88	71.74
		Avg. Cost ↓	0.18	2.15	0.13	**0.14**	0.16	1.12	1.81	1.05	0.57	0.14	0.17	0.62	0.31
Average	All Datasets	Avg. Acc. ↑	95.60	78.01	62.43	66.26	70.68	75.49	70.01	73.97	77.32	69.88	70.99	**78.09**	77.26
		Avg. Cost ↓	0.37	2.72	0.14	**0.38**	0.41	2.18	2.21	2.08	1.22	0.77	0.53	1.23	1.30
		Rank Score ↑	93.68	68.88	64.63	67.13	71.09	68.92	64.62	68.10	73.73	68.98	70.89	**74.33**	73.40

Table A5. Performance comparison of router methods on VL-RouterBench across datasets for $\lambda = 100.0$. **Avg. Acc.** is average accuracy (%), **Avg. Cost** is average cost ($/10K samples). The best and second-best results among learned routers are highlighted in **bold** and underlined, respectively (excluding Oracle and baseline methods).

Group	Dataset	Metric	Baselines			Hard Label				Soft Label ($\lambda = 100.0$)					
			Oracle	Strongest	Cheapest	KNN	PRkNN	OVR	K-means	Linear	MLP	CosineCls	RouterDC	VLC	ZOOTER
General	HallusionBench	Avg. Acc. ↑	99.46	71.89	57.84	40.54	43.24	66.49	**70.27**	67.03	58.92	56.22	59.46	59.46	58.38
		Avg. Cost ↓	0.23	2.97	0.14	**0.18**	0.21	1.67	2.89	1.56	0.33	0.20	0.21	0.23	0.19
	InfoVQA	Avg. Acc. ↑	96.49	83.28	53.68	58.36	61.87	**81.10**	80.60	79.77	74.41	65.05	68.06	67.56	56.69
		Avg. Cost ↓	0.37	2.19	0.14	0.24	0.29	1.49	2.09	1.71	0.56	0.36	0.44	0.60	**0.21**
	MMBench	Avg. Acc. ↑	98.16	89.86	69.59	69.59	81.11	**85.25**	70.97	83.41	77.42	80.18	81.57	82.49	68.66
		Avg. Cost ↓	0.24	2.22	0.13	**0.19**	0.27	2.26	1.17	1.32	0.30	0.23	0.44	0.34	0.20
	MMMU	Avg. Acc. ↑	98.46	74.36	45.64	42.05	53.33	60.00	65.64	**68.72**	55.90	51.28	53.85	53.33	54.87
		Avg. Cost ↓	0.64	3.24	0.17	**0.41**	0.43	6.95	10.59	10.21	0.52	0.41	0.45	0.50	0.49
	MMStar	Avg. Acc. ↑	96.68	72.76	49.17	54.48	59.80	**68.44**	66.11	62.13	64.45	60.47	58.80	60.47	61.13
		Avg. Cost ↓	0.36	2.29	0.14	**0.25**	0.29	2.85	1.68	2.66	0.36	0.30	0.30	0.32	0.30
	RealWorldQA	Avg. Acc. ↑	99.32	77.40	67.12	55.48	63.70	67.81	60.27	65.07	67.12	65.07	65.75	**70.55**	66.44
		Avg. Cost ↓	0.22	2.20	0.13	**0.14**	0.18	1.46	1.17	0.80	0.17	0.23	0.21	0.27	0.15
STEM	AI2D	Avg. Acc. ↑	99.69	88.21	72.33	67.45	73.58	80.66	76.42	78.14	**80.82**	74.53	72.01	74.53	72.96
		Avg. Cost ↓	0.24	2.19	0.13	**0.16**	0.19	2.05	2.56	0.91	0.36	0.25	0.21	0.22	0.23
	MathVerse	Avg. Acc. ↑	90.36	49.74	23.05	59.77	67.97	**68.75**	66.93	67.32	61.98	64.84	64.06	60.29	63.02
		Avg. Cost ↓	1.06	4.67	0.18	1.64	1.70	4.01	3.19	3.55	1.60	3.86	3.69	**1.37**	2.31
	MathVision	Avg. Acc. ↑	89.55	38.81	13.43	28.36	23.88	35.82	46.27	41.79	49.25	**52.24**	50.75	50.75	52.24
		Avg. Cost ↓	1.77	8.16	0.17	1.23	**0.72**	13.10	5.64	10.53	1.80	3.29	2.49	1.76	1.78
	MathVista	Avg. Acc. ↑	95.85	75.13	51.30	53.89	70.98	77.20	73.58	71.50	75.65	65.29	68.39	73.06	**77.72**
		Avg. Cost ↓	0.36	5.00	0.16	**0.50**	0.54	3.46	2.81	4.93	1.03	0.85	0.79	0.89	1.11
Chart & Doc	ChartQA	Avg. Acc. ↑	97.80	86.20	81.00	81.40	80.80	81.00	40.60	80.60	81.40	83.00	82.00	**83.80**	81.80
		Avg. Cost ↓	0.21	2.19	0.14	**0.14**	0.17	1.09	0.69	0.91	0.16	0.21	0.25	0.25	0.16
	DocVQA	Avg. Acc. ↑	98.56	91.48	78.95	78.66	80.48	**85.55**	82.30	81.44	83.06	80.48	79.52	82.58	79.04
		Avg. Cost ↓	0.18	2.18	0.14	**0.15**	0.18	1.22	1.25	0.74	0.36	0.20	0.21	0.29	0.16
	OCRBench	Avg. Acc. ↑	99.06	89.20	80.28	77.00	81.22	85.92	79.34	78.40	83.10	84.04	83.57	**87.32**	79.34
		Avg. Cost ↓	0.23	2.41	0.14	**0.16**	0.19	1.44	0.96	0.78	0.28	0.20	0.22	0.32	0.17
	TextVQA	Avg. Acc. ↑	89.46	73.85	72.13	70.98	**72.13**	67.05	66.09	64.37	70.31	71.46	70.31	70.59	72.13
		Avg. Cost ↓	0.18	2.15	0.13	0.14	0.16	1.12	1.81	0.39	0.20	0.17	0.15	0.25	**0.13**
Average	All Datasets	Avg. Acc. ↑	95.60	78.01	62.43	66.26	70.68	**75.49**	70.01	73.20	73.31	71.35	72.07	72.02	69.97
		Avg. Cost ↓	0.37	2.72	0.14	**0.38**	0.41	2.18	2.21	1.85	0.52	0.74	0.73	0.48	0.50
		Rank Score ↑	93.68	68.88	64.63	67.13	71.09	68.92	64.62	68.26	**73.01**	70.38	71.07	72.01	70.06

Table A6. Performance comparison of router methods on VL-RouterBench across datasets for $\lambda = 1000.0$. **Avg. Acc.** is average accuracy (%), **Avg. Cost** is average cost ($/10K samples). The best and second-best results among learned routers are highlighted in **bold** and underlined, respectively (excluding Oracle and baseline methods).

Group	Dataset	Metric	Baselines			Hard Label				Soft Label ($\lambda = 1000.0$)					
			Oracle	Strongest	Cheapest	KNN	PRkNN	OVR	K-means	Linear	MLP	CosineCls	RouterDC	VLC	ZOOTER
General	HallusionBench	Avg. Acc. ↑	99.46	71.89	57.84	40.54	43.24	66.49	**70.27**	68.11	46.49	57.84	56.22	57.30	57.30
		Avg. Cost ↓	0.23	2.97	0.14	0.18	0.21	1.67	2.89	1.62	0.18	**0.14**	0.20	0.14	0.14
	InfoVQA	Avg. Acc. ↑	96.49	83.28	53.68	58.36	61.87	81.10	80.60	**81.27**	59.36	55.18	65.05	54.01	53.68
		Avg. Cost ↓	0.37	2.19	0.14	0.24	0.29	1.49	2.09	1.17	0.22	0.16	0.45	0.15	**0.14**
	MMBench	Avg. Acc. ↑	98.16	89.86	69.59	69.59	81.11	**85.25**	70.97	80.18	76.96	73.27	77.88	72.35	70.97
		Avg. Cost ↓	0.24	2.22	0.13	0.19	0.27	2.26	1.17	1.43	0.21	0.19	0.23	0.17	**0.13**
	MMMU	Avg. Acc. ↑	98.46	74.36	45.64	42.05	53.33	60.00	**65.64**	61.03	47.69	49.74	54.36	45.13	43.08
		Avg. Cost ↓	0.64	3.24	0.17	0.41	0.43	6.95	10.59	4.69	0.26	0.33	0.52	0.18	**0.17**
	MMStar	Avg. Acc. ↑	96.68	72.76	49.17	54.48	59.80	**68.44**	66.11	64.45	54.82	51.50	57.81	49.50	49.17
		Avg. Cost ↓	0.36	2.29	0.14	0.25	0.29	2.85	1.68	1.79	0.22	0.19	0.32	0.15	**0.14**
	RealWorldQA	Avg. Acc. ↑	99.32	77.40	67.12	55.48	63.70	**67.81**	60.27	65.75	61.64	67.12	65.07	65.75	67.12
		Avg. Cost ↓	0.22	2.20	0.13	0.14	0.18	1.46	1.17	1.06	0.14	0.14	0.28	**0.13**	0.13
STEM	AI2D	Avg. Acc. ↑	99.69	88.21	72.33	67.45	73.58	**80.66**	76.42	76.10	71.86	72.33	75.31	72.96	72.01
		Avg. Cost ↓	0.24	2.19	0.13	0.16	0.19	2.05	2.56	1.51	0.15	0.14	0.24	**0.13**	0.13
	MathVerse	Avg. Acc. ↑	90.36	49.74	23.05	59.77	67.97	**68.75**	66.93	66.93	56.90	60.29	64.19	40.89	23.05
		Avg. Cost ↓	1.06	4.67	0.18	1.64	1.70	4.01	3.19	3.19	1.35	1.37	3.75	0.85	**0.18**
	MathVision	Avg. Acc. ↑	89.55	38.81	13.43	28.36	23.88	35.82	46.27	47.76	31.34	**52.24**	43.28	31.34	13.43
		Avg. Cost ↓	1.77	8.16	0.17	1.23	0.72	13.10	5.64	5.56	1.03	1.78	1.47	1.01	**0.17**
	MathVista	Avg. Acc. ↑	95.85	75.13	51.30	53.89	70.98	**77.20**	73.58	72.02	56.99	67.88	62.18	54.92	51.30
		Avg. Cost ↓	0.36	5.00	0.16	0.50	0.54	3.46	2.81	2.75	0.37	0.80	0.52	0.28	**0.16**
Chart & Doc	ChartQA	Avg. Acc. ↑	97.80	86.20	81.00	81.40	80.80	81.00	40.60	80.20	81.40	81.60	**84.20**	81.00	81.00
		Avg. Cost ↓	0.21	2.19	0.14	**0.14**	0.17	1.09	0.69	0.60	0.18	0.15	0.24	0.14	0.14
	DocVQA	Avg. Acc. ↑	98.56	91.48	78.95	78.66	80.48	**85.55**	82.30	81.05	79.43	78.95	80.67	79.23	78.85
		Avg. Cost ↓	0.18	2.18	0.14	0.15	0.18	1.22	1.25	0.70	0.15	**0.14**	0.22	0.14	0.14
	OCRBench	Avg. Acc. ↑	99.06	89.20	80.28	77.00	81.22	**85.92**	79.34	80.75	80.75	80.75	83.57	80.75	80.28
		Avg. Cost ↓	0.23	2.41	0.14	0.16	0.19	1.44	0.96	0.79	0.20	0.15	0.19	**0.14**	0.14
	TextVQA	Avg. Acc. ↑	89.46	73.85	72.13	70.98	72.13	67.05	66.09	66.95	71.65	72.13	71.55	**72.22**	72.13
		Avg. Cost ↓	0.18	2.15	0.13	0.14	0.16	1.12	1.81	1.14	0.14	**0.13**	0.16	0.13	0.13
Average	All Datasets	Avg. Acc. ↑	95.60	78.01	62.43	66.26	70.68	**75.49**	70.01	73.40	67.75	68.65	71.17	65.21	62.33
		Avg. Cost ↓	0.37	2.72	0.14	0.38	0.41	2.18	2.21	1.57	0.33	0.35	0.72	0.24	**0.14**
		Rank Score ↑	93.68	68.88	64.63	67.13	**71.09**	68.92	64.62	69.29	68.73	69.53	70.31	66.78	64.54

Table A7. Performance comparison of router methods on VL-RouterBench across datasets for $\lambda = 10000.0$. **Avg. Acc.** is average accuracy (%), **Avg. Cost** is average cost ($/10K samples). The best and second-best results among learned routers are highlighted in **bold** and underlined, respectively (excluding Oracle and baseline methods).

Group	Dataset	Metric	Baselines			Hard Label				Soft Label ($\lambda = 10000.0$)					
			Oracle	Strongest	Cheapest	KNN	PRkNN	OVR	K-means	Linear	MLP	CosineCls	RouterDC	VLC	ZOOTER
General	HallusionBench	Avg. Acc. ↑	99.46	71.89	57.84	40.54	43.24	66.49	**70.27**	62.70	48.11	51.89	58.38	41.62	56.76
		Avg. Cost ↓	0.23	2.97	0.14	0.18	0.21	1.67	2.89	0.77	**0.14**	0.30	0.17	0.16	0.14
	InfoVQA	Avg. Acc. ↑	96.49	83.28	53.68	58.36	61.87	**81.10**	80.60	79.10	58.19	62.54	61.71	60.87	53.68
		Avg. Cost ↓	0.37	2.19	0.14	0.24	0.29	1.49	2.09	0.80	0.20	0.46	0.33	0.29	**0.14**
	MMBench	Avg. Acc. ↑	98.16	89.86	69.59	69.59	81.11	**85.25**	70.97	71.43	74.19	76.50	82.03	77.42	69.59
		Avg. Cost ↓	0.24	2.22	0.13	0.19	0.27	2.26	1.17	0.43	0.16	0.61	0.47	0.19	**0.13**
	MMMU	Avg. Acc. ↑	98.46	74.36	45.64	42.05	53.33	60.00	**65.64**	55.38	44.10	53.33	45.13	44.10	45.13
		Avg. Cost ↓	0.64	3.24	0.17	0.41	0.43	6.95	10.59	1.20	0.28	8.07	6.11	0.30	**0.17**
	MMStar	Avg. Acc. ↑	96.68	72.76	49.17	54.48	59.80	**68.44**	66.11	51.83	53.82	60.13	55.81	54.82	49.17
		Avg. Cost ↓	0.36	2.29	0.14	0.25	0.29	2.85	1.68	0.49	0.18	3.12	1.77	0.20	**0.14**
	RealWorldQA	Avg. Acc. ↑	99.32	77.40	67.12	55.48	63.70	**67.81**	60.27	60.96	65.75	59.59	62.33	62.33	67.12
		Avg. Cost ↓	0.22	2.20	0.13	0.14	0.18	1.46	1.17	0.46	**0.13**	0.51	0.19	0.16	0.13
STEM	AI2D	Avg. Acc. ↑	99.69	88.21	72.33	67.45	73.58	**80.66**	76.42	70.44	70.91	73.58	73.43	72.17	72.33
		Avg. Cost ↓	0.24	2.19	0.13	0.16	0.19	2.05	2.56	0.51	0.14	0.38	0.27	0.15	**0.13**
	MathVerse	Avg. Acc. ↑	90.36	49.74	23.05	59.77	67.97	**68.75**	66.93	53.39	52.47	45.83	50.39	43.36	23.05
		Avg. Cost ↓	1.06	4.67	0.18	1.64	1.70	4.01	3.19	1.56	1.73	9.42	7.30	0.79	**0.18**
	MathVision	Avg. Acc. ↑	89.55	38.81	13.43	28.36	23.88	35.82	**46.27**	34.33	31.34	23.88	22.39	22.39	13.43
		Avg. Cost ↓	1.77	8.16	0.17	1.23	0.72	13.10	5.64	1.48	1.48	40.17	40.70	0.98	**0.17**
	MathVista	Avg. Acc. ↑	95.85	75.13	51.30	53.89	70.98	**77.20**	73.58	63.21	55.44	55.96	59.07	54.92	51.30
		Avg. Cost ↓	0.36	5.00	0.16	0.50	0.54	3.46	2.81	1.00	0.38	6.74	3.64	0.31	**0.16**
Chart & Doc	ChartQA	Avg. Acc. ↑	97.80	86.20	81.00	81.40	80.80	81.00	40.60	76.60	81.40	81.40	**83.00**	81.20	81.00
		Avg. Cost ↓	0.21	2.19	0.14	**0.14**	0.17	1.09	0.69	0.33	0.15	0.24	0.22	0.17	0.14
	DocVQA	Avg. Acc. ↑	98.56	91.48	78.95	78.66	80.48	**85.55**	82.30	80.10	78.56	79.90	79.71	78.95	78.85
		Avg. Cost ↓	0.18	2.18	0.14	0.15	0.18	1.22	1.25	0.61	**0.14**	0.19	0.18	0.16	0.14
	OCRBench	Avg. Acc. ↑	99.06	89.20	80.28	77.00	81.22	**85.92**	79.34	80.28	79.34	83.10	83.57	81.69	80.28
		Avg. Cost ↓	0.23	2.41	0.14	0.16	0.19	1.44	0.96	0.50	0.17	0.22	0.21	0.16	**0.14**
	TextVQA	Avg. Acc. ↑	89.46	73.85	72.13	70.98	**72.13**	67.05	66.09	66.67	71.55	70.11	70.59	69.92	72.13
		Avg. Cost ↓	0.18	2.15	0.13	0.14	0.16	1.12	1.81	0.43	**0.13**	0.27	0.17	0.14	0.13
Average	All Datasets	Avg. Acc. ↑	95.60	78.01	62.43	66.26	70.68	**75.49**	70.01	68.55	66.60	76.17	76.91	65.47	62.36
		Avg. Cost ↓	0.37	2.72	0.14	0.38	0.41	2.18	2.21	0.71	0.38	2.48	1.93	0.27	**0.14**
		Rank Score ↑	93.68	68.88	64.63	67.13	**71.09**	68.92	64.62	68.01	67.48	68.41	70.86	66.90	64.57

Table A8. Performance comparison of router methods on VL-RouterBench across datasets for $\lambda = \infty$. **Avg. Acc.** is average accuracy (%), **Avg. Cost** is average cost ($/10K samples). The best and second-best results among learned routers are highlighted in **bold** and underlined, respectively (excluding Oracle and baseline methods).

Group	Dataset	Metric	Baselines			Hard Label				Soft Label ($\lambda = \infty$)					
			Oracle	Strongest	Cheapest	KNN	PRkNN	OVR	K-means	Linear	MLP	CosineCls	RouterDC	VLC	ZOOTER
General	HallusionBench	Avg. Acc. ↑	99.46	71.89	57.84	40.54	43.24	66.49	70.27	57.84	47.57	78.38	**80.00**	45.41	49.73
		Avg. Cost ↓	0.23	2.97	0.14	0.18	0.21	1.67	2.89	**0.14**	0.16	5.18	4.66	0.15	0.14
	InfoVQA	Avg. Acc. ↑	96.49	83.28	53.68	58.36	61.87	81.10	80.60	51.17	58.03	81.77	**81.94**	59.20	52.34
		Avg. Cost ↓	0.37	2.19	0.14	0.24	0.29	1.49	2.09	**0.14**	0.23	1.34	1.47	0.26	0.14
	MMBench	Avg. Acc. ↑	98.16	89.86	69.59	69.59	81.11	85.25	70.97	62.67	69.59	**90.32**	90.32	67.74	64.52
		Avg. Cost ↓	0.24	2.22	0.13	0.19	0.27	2.26	1.17	0.15	0.16	5.49	5.49	0.17	**0.14**
	MMMU	Avg. Acc. ↑	98.46	74.36	45.64	42.05	53.33	60.00	65.64	37.95	39.49	76.92	**77.44**	41.03	37.44
		Avg. Cost ↓	0.64	3.24	0.17	0.41	0.43	6.95	10.59	0.19	0.28	17.30	17.60	0.24	**0.17**
	MMStar	Avg. Acc. ↑	96.68	72.76	49.17	54.48	59.80	68.44	66.11	46.51	55.15	73.42	**74.09**	51.16	46.84
		Avg. Cost ↓	0.36	2.29	0.14	0.25	0.29	2.85	1.68	0.20	0.20	8.08	7.93	0.17	**0.15**
	RealWorldQA	Avg. Acc. ↑	99.32	77.40	67.12	55.48	63.70	67.81	60.27	57.53	60.27	**78.08**	75.34	59.59	62.33
		Avg. Cost ↓	0.22	2.20	0.13	0.14	0.18	1.46	1.17	**0.13**	0.14	3.42	3.53	0.14	0.13
STEM	AI2D	Avg. Acc. ↑	99.69	88.21	72.33	67.45	73.58	80.66	76.42	66.20	67.30	**87.11**	86.95	68.08	67.45
		Avg. Cost ↓	0.24	2.19	0.13	0.16	0.19	2.05	2.56	**0.13**	0.14	7.02	7.09	0.14	0.13
	MathVerse	Avg. Acc. ↑	90.36	49.74	23.05	59.77	67.97	**68.75**	66.93	46.22	56.25	65.62	64.19	43.49	25.13
		Avg. Cost ↓	1.06	4.67	0.18	1.64	1.70	4.01	3.19	1.03	1.35	7.56	6.75	0.96	**0.28**
	MathVision	Avg. Acc. ↑	89.55	38.81	13.43	28.36	23.88	35.82	46.27	31.34	29.85	**61.19**	61.19	22.39	22.39
		Avg. Cost ↓	1.77	8.16	0.17	1.23	0.72	13.10	5.64	1.00	1.04	26.73	26.73	0.84	**0.54**
	MathVista	Avg. Acc. ↑	95.85	75.13	51.30	53.89	70.98	**77.20**	73.58	57.51	55.44	77.20	77.20	49.22	50.78
		Avg. Cost ↓	0.36	5.00	0.16	0.50	0.54	3.46	2.81	0.67	0.42	9.63	9.63	0.30	**0.19**
Chart & Doc	ChartQA	Avg. Acc. ↑	97.80	86.20	81.00	**81.40**	80.80	81.00	40.60	80.00	80.00	79.20	80.60	80.40	80.60
		Avg. Cost ↓	0.21	2.19	0.14	**0.14**	0.17	1.09	0.69	0.16	0.16	2.08	2.04	0.16	0.14
	DocVQA	Avg. Acc. ↑	98.56	91.48	78.95	78.66	80.48	85.55	82.30	75.50	78.47	88.23	**89.76**	78.76	77.61
		Avg. Cost ↓	0.18	2.18	0.14	0.15	0.18	1.22	1.25	**0.14**	0.15	1.12	1.29	0.16	0.14
	OCRBench	Avg. Acc. ↑	99.06	89.20	80.28	77.00	81.22	85.92	79.34	75.59	75.12	**92.49**	92.49	77.00	76.53
		Avg. Cost ↓	0.23	2.41	0.14	0.16	0.19	1.44	0.96	0.15	0.17	1.58	1.68	0.15	**0.14**
	TextVQA	Avg. Acc. ↑	89.46	73.85	72.13	70.98	**72.13**	67.05	66.09	70.69	71.17	68.58	68.68	69.83	69.83
		Avg. Cost ↓	0.18	2.15	0.13	0.14	0.16	1.12	1.81	**0.13**	0.13	1.66	1.81	0.14	0.13
Average	All Datasets	Avg. Acc. ↑	95.60	78.01	62.43	66.26	70.68	75.49	70.01	62.92	65.93	77.49	**78.23**	63.87	60.43
		Avg. Cost ↓	0.37	2.72	0.14	0.38	0.41	2.18	2.21	0.28	0.33	3.49	3.46	0.28	**0.16**
		Rank Score ↑	93.68	68.88	64.63	67.13	**71.09**	68.92	64.62	64.41	67.04	65.61	66.56	65.33	62.58