MMR-Bench: A Comprehensive Benchmark for Multimodal LLM Routing

Haoxuan Ma[1,2†], Guannan Lai[1,2†], and Han-Jia Ye[1,2*]

[1] School of Artificial Intelligence, Nanjing University, Nanjing, China
[2] National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, China
{mahx,laign,yehj}@lamda.nju.edu.cn
[†]Equal contribution. [*]Corresponding author.

Datasets Source Code

Abstract. Multimodal large language models (MLLMs) have advanced rapidly, yet heterogeneity in architecture, alignment strategies, and efficiency means that no single model is uniformly superior across tasks. In practical deployments, workloads span lightweight OCR to complex multimodal reasoning; using one MLLM for all queries either over-provisions compute on easy instances or sacrifices accuracy on hard ones. Query-level model selection (routing) addresses this tension, but extending routing from text-only LLMs to MLLMs is nontrivial due to modality fusion, wide variation in computational cost across models, and the absence of a standardized, budget-aware evaluation.

We present **MMR-Bench**, a unified benchmark that isolates the multimodal routing problem and enables comparison under fixed candidate sets and cost models. MMR-Bench provides (i) a controlled environment with modality-aware inputs and variable compute budgets, (ii) a broad suite of vision–language tasks covering OCR, general VQA, and multimodal reasoning, and (iii) strong single-model baselines, oracle upper bounds, and representative routing policies. Using MMR-Bench, we show that incorporating multimodal signals improves routing quality. Empirically, these cues improve the cost–accuracy frontier and enable the routed system to exceed the strongest single model's accuracy at roughly 33% of its cost. Furthermore, policies trained on a subset of models and tasks generalize zero-shot to new datasets and text-only benchmarks without retuning, establishing MMR-Bench as a foundation for studying adaptive multimodal model selection and efficient MLLM deployment.

1 Introduction

Multimodal large language models (MLLMs) have witnessed rapid progress in recent years, driving remarkable advances across vision–language tasks spanning text recognition, image understanding, and other multimodal scenarios [3,23,33]. However, the MLLM landscape is highly heterogeneous: models vary substantially in architecture, training data, modality alignment, and computational efficiency [34,37]. We refer to this diversity as the **MLLM zoo**—a dynamic ecosystem of models with distinct strengths, limitations, and resource requirements.

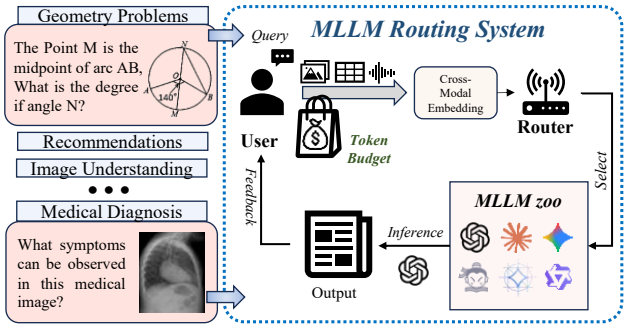

Fig. 1: Workflow of MLLM routing, which can be applied to tasks such as mathematical reasoning, multimodal recommendation and understanding, and medical diagnosis.

This diversity, while fueling innovation, also reveals a critical limitation: **no single MLLM is uniformly superior across tasks.** In real-world deployments, tasks vary widely, ranging from lightweight OCR to complex multimodal reasoning [21, 22, 38]. As a result, using a single MLLM for all queries either over-allocates compute on easy instances or sacrifices accuracy on difficult ones.

A natural remedy is routing, which has demonstrated its effectiveness for LLMs [5–7, 13]. Routing selects different models per query to balance efficiency and accuracy. To examine whether this idea transfers to MLLMs, we applied a **text-only** routing strategy to multimodal tasks. Surprisingly, as shown in Figure 2, text-only routing can **outperform** Qwen2.5-VL-7B and GPT-5-nano at equal cost, indicating the strong potential of model routing. Consistent with intuition, however, it does not surpass several stronger models such as Gemini and GPT-5 at the same cost; this suggests that text-only information is insufficient and **highlights the importance of incorporating multimodal signals** to further enhance routing performance.

Building on these findings, we introduce the concept of *multimodal LLM routing*. As illustrated in Figure 1, a user provides multimodal inputs along with a compute budget to the routing system, which selects an appropriate model from the MLLM zoo to generate the response and returns the output to the user. Compared with routing for text-only LLMs, this setting presents several unique challenges: (1) **Modality Gap**: MLLMs must integrate heterogeneous inputs across different modalities, introducing alignment and representation challenges absent in text-only scenarios [18]; (2) **Model Heterogeneity**: the MLLM zoo comprises models with diverse architectures, training data, modality alignments, and computational characteristics, which complicates routing decisions; (3) **Benchmarking Complexity**: routing for MLLMs couples multi-dimensional costs with modality-specific behavior, making it difficult to define unified cost models and candidate sets. This complexity has so far prevented the emergence of a stan-

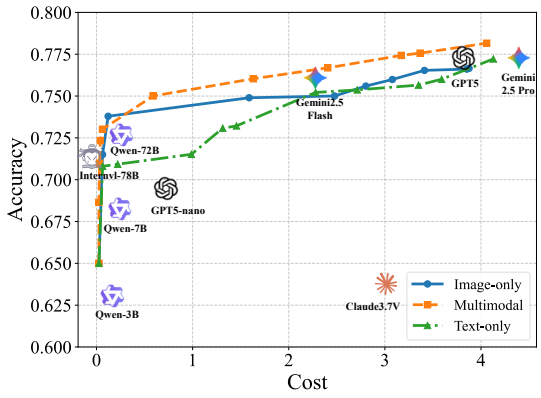

Fig. 2: Routing comparison across different modality signals, where Qwen denotes the Qwen2.5-VL series and InternVL denotes the InternVL3 series.

dardized, modality-aware benchmark for MLLM routing, even though analogous benchmarks already exist for text-only LLMs [8, 9, 24].

These differences **motivate** the need for a controlled benchmark that isolates the multimodal routing problem and enables rigorous, directly comparable evaluations. We present **MMR-Bench**, designed to answer two key questions: (1) Are unimodal signals (text-only or image-only) sufficient, or are genuinely multimodal cues required? (2) Under fixed compute budgets, how much performance does multimodal routing actually deliver in practice?

Accordingly, MMR-Bench provides a unified framework for evaluating routing strategies across diverse MLLMs and tasks. It offers a controlled environment with standardized cost models, adjustable compute budgets, and modality-aware inputs; integrates a broad suite of vision–language benchmarks; and includes strong single-model baselines, oracle upper bounds, and representative routing methods. Together, these components enable consistent, reproducible comparisons and establish a solid foundation for studying adaptive multimodal model selection and efficient MLLM deployment.

Using MMR-Bench, we report three consistent findings:

- **Multimodal signals matter.** Unimodal routers are often miscalibrated on text-in-image inputs and in low-resolution or cluttered scenes, whereas fusing image and text modalities closes this gap at comparable cost.
- **Efficiency under budgets.** In certain scenarios, with only about 33% of the cost of the strongest single model, our multimodal routing matches or even surpasses its accuracy, achieving a strictly better cost–accuracy trade-off.
- **Robust generalization.** A routing policy trained on a subset of MLLMs and tasks generalizes to unseen data from different distributions without additional tuning, and can also transfer effectively to text-only LLM routing.

2 Related work

2.1 Multimodal Large Language Models

MLLMs couple language backbones with visual encoders through lightweight integration and instruction tuning, achieving broad vision–language competence [16, 20]. The ecosystem, however, spans markedly different cost profiles. **Commercial frontier models** typically provide stronger multi-step reasoning, long-context handling, and robust perception [1,29], while **open-weight models** are cheaper to deploy and remain competitive on routine recognition and short-form understanding tasks [3, 23, 33]. This heterogeneity makes any single model suboptimal for all inputs and budgets. These observations motivate *routing*, which decides at inference time which model to invoke for a given instance under explicit compute constraints [27, 40]. Our focus is on the selection problem rather than new architectures, quantifying when lightweight models suffice and when heavier models are warranted.

2.2 Routing and Model Selection for MLLMs

In text-only settings, model routing reduces cost without sacrificing quality by learning to dispatch queries among heterogeneous LLMs [14, 28, 36], and recent benchmarks [8,9] have standardized evaluation across routers and cost–accuracy trade-offs. Extending these ideas to multimodal settings introduces new challenges. Routing must exploit cross-modal signals and handle architecture-level heterogeneity, making both confidence estimation and cost modeling more complex. Early multimodal approaches embed routing within an MLLM via Mixture-of-Experts architectures [19] or retrofit existing MLLMs with dynamic expert paths [2, 26, 35], improving efficiency and accuracy within one model family but not addressing inter-model selection across the broader MLLM zoo. Our benchmark targets this gap by learning when to use which MLLM under a given budget.

3 Preliminaries

3.1 Problem Definition

Let M denote the number of modalities (e.g., text, image, audio, video). A *multimodal large language model* (MLLM) is a mapping

$$f : \mathcal{X}^1 \times \mathcal{X}^2 \times \cdots \times \mathcal{X}^M \to \mathcal{Y},$$

where $\mathcal{X}^r$ is the input space of modality r and $\mathcal{Y}$ is the shared output space.

We focus on a routing problem in the **two-modality case** (text and image). For each incoming multimodal input, the goal is to select one model from a candidate zoo so as to achieve a desirable trade-off between model utility and

cost. In practice, not all modalities are always available; to capture this, we introduce a *modality availability vector*

$$\mathbf{m}_i = (m_i^{\text{text}}, m_i^{\text{img}}), \qquad m_i^r \in \{0, 1\},$$

where $m_i^r = 1$ indicates modality r is present and $m_i^r = 0$ indicates it is missing. For example, $\mathbf{m}_i = (1, 0)$ denotes a text-only sample, while $\mathbf{m}_i = (1, 1)$ denotes a paired text–image input.

Definition 1 (MLLM routing). *Define a support set*

$$\mathcal{S} = \left\{(x_i, \mathbf{m}_i, \mathbf{u}_i, \mathbf{c}_i)\right\}_{i=1}^{n}, \tag{1}$$

where $x_i = (x_i^{text}, x_i^{img})$, $\mathbf{u}_i = (u_{i,1}, \ldots, u_{i,K})^\top$ *contains the utility of each candidate model on sample* i, *and* $\mathbf{c}_i = (c_{i,1}, \ldots, c_{i,K})^\top$ *contains their corresponding costs.*

For each sample the router selects the model index

$$j_i^\star = \underset{j \in \{1,\ldots,K\}}{\operatorname{argmax}} \; S_{i,j}, \tag{2}$$

using the overall score

$$S_{i,j} = u_{i,j} - \lambda \, c_{i,j}, \tag{3}$$

where $\lambda \geq 0$ *is a tunable parameter that controls the trade-off between performance and cost.*

During inference, the router uses only the modalities available for input i (specified by $\mathbf{m}_i$), enabling text-only, image-only, or multimodal baselines via the corresponding extractors/fusion. Thus, $\mathbf{m}_i$ selects the active router family and allowable signals, and performance differences reveal each modality's contribution.

3.2 Motivation

We conduct an empirical study using precomputed utilities $\{u_{i,j}\}$ and costs $\{c_{i,j}\}$ from a heterogeneous zoo of MLLMs (Section 4). The study addresses two questions: (i) whether a *text-only* router is effective when inputs are multimodal, and (ii) whether unimodal observables (text-only or image-only) are sufficient, or whether genuinely multimodal cues are required.

Setting. We consider a fixed set of candidate models $\{f_j\}_{j=1}^K$, along with their utilities $\{u_{i,j}\}$ and costs $\{c_{i,j}\}$ for each query. The only difference between routers lies in the observable features $\phi(x_i)$ they use: (a) *text-only* features $\phi_{\text{text}}(x_i)$; (b) *image-only* features $\phi_{\text{img}}(x_i)$; and (c) *multimodal* features $\phi_{\text{mm}}(x_i)$ that jointly encode text and image signals. Each router defines a policy $\pi(x_i) \in \{1, \ldots, K\}$ and is evaluated with the score defined by Equation (3), where $\lambda \geq 0$ is varied to trace the cost–accuracy trade-off.

Protocol. To avoid confounding effects from powerful learned routers, we employ a lightweight procedure shared across all routing families. Given a trade-off parameter $\lambda \geq 0$ and a specified number of clusters k, the protocol is as follows: (i) compute features $\phi(x_i)$ on the training and validation split; (ii) apply k-means to the training features to obtain clusters $\{h\}_{h=1}^{k}$; (iii) for each cluster h, select a single model by maximizing the average validation score:

$$j_h^\star(\lambda) = \underset{j \in \{1,\ldots,K\}}{\operatorname{argmax}} \frac{1}{|\mathcal{S}_{\mathrm{val}}(h)|} \sum_{i \in \mathcal{S}_{\mathrm{val}}(h)} S_{i,j}(\lambda), \tag{4}$$

where $\mathcal{S}_{\mathrm{val}}(h)$ denotes validation samples in cluster h, and $S_{i,j}(\lambda)$ is defined in Equation (3). At test time, each sample x is assigned to its nearest cluster $h(x)$ and routed to $j_{h(x)}^\star$. Sweeping the number of clusters k and cost weight λ yields a family of policies covering the cost–accuracy trade-off, whose frontiers (Figure 2) enable matched-cost and matched-accuracy comparisons.

Q1: Can text-only routing methods be used in multimodal scenarios?

Yes, but only partially. Across mixed multimodal workloads, the text-only router improves over several fixed models such as Qwen2.5-VL-7B and GPT-5-nano at comparable normalized cost, as illustrated in Figure 2; this shows that instance-wise selection can help even without visual features. However, the improvements are uneven. The router consistently underperforms on subsets where visual evidence determines difficulty, including text-in-image queries, dense OCR, charts, spatial reasoning, and low-resolution crops. Error analysis indicates that these failures arise from incorrect assignments: the router tends to overestimate the adequacy of cheaper models when textual cues provide an unreliable proxy for visual complexity. Overall, text-only routing fails to achieve optimal performance on vision-governed cases.

Q2: Are unimodal signals sufficient, or are multimodal cues required?

Multimodal cues are essential for strong routing. Figure 2 compares routing frontiers under identical model sets and cost accounting. The *multimodal* router consistently outperforms both *text-only* and *image-only* variants: at fixed cost it achieves higher accuracy, and at fixed accuracy it operates at lower cost. Ablations over the number of clusters k and the cost weight λ show that this advantage is stable, and the gap widens in settings where visual clutter or linguistic complexity dominate difficulty. Category-level evaluations show the same trend. Text-only routing falls behind on vision-driven inputs, image-only routing falls behind on language-driven inputs, and multimodal features reduce incorrect assignments in both regimes.

Takeaways. (1) Routing over heterogeneous MLLMs improves the cost–accuracy trade-off even with simple, unsupervised policies. (2) Unimodal observables are insufficient to realize the full gains; access to *joint* text–image cues is necessary to approach the empirical frontier on mixed multimodal workloads.

Table 1: Comparison across routing benchmarks along four dimensions: supported modality, reported evaluation metrics, and the availability (or count) of router baselines. MMR-Bench reports three cost-aware metrics in an offline setting.

Benchmark	Text	Image	Metrics	Baselines
RouterBench [8]	✓	✗	1	5
RouterEval [9]	✓	✗	1	8
EmbedLLM [42]	✓	✗	1	4
MixInstruct [10]	✓	✗	3	6
LLMRouterBench [17]	✓	✗	4	10
MMR-Bench	✓	✓	3	10

4 MMR-Bench: Benchmark Construction

We introduce **MMR-Bench**, a budget-aware benchmark for routing across the heterogeneous MLLM zoo of candidate models. MMR-Bench is designed as an *offline environment*: for each multimodal query and each candidate model, we provide the raw model output, a task-specific utility score, and a normalized inference cost derived under a unified pricing scheme. This outcome table (over **100k** pairs of instances and models across multiple candidates and tasks), together with frozen splits and deterministic scoring scripts, enables systematic and reproducible evaluation of cost-aware routing strategies without re-running any MLLM, as summarized in Table 1.

4.1 Design Principles

MMR-Bench is built specifically for routing across a heterogeneous MLLM zoo. Its construction follows three principles:

- **Extensive coverage.** The dataset selection covers most typical multimodal scenarios and difficulty levels, with varied resolutions and scene complexity.
- **Representative models.** The candidate pool includes widely used MLLMs with diverse capacities and training regimes, ensuring practical relevance.
- **Extensibility.** The benchmark is offline and modular; new models or datasets can be added incrementally, while frozen splits and deterministic scoring preserve comparability.

4.2 Datasets and Model Pool

Datasets. We benchmark routing on three scenarios: visual OCR (OCRBench [21], SEED-Bench v2 Plus [15]), general VQA (MMStar [4], RealWorldQA [25]), and multimodal reasoning (MathVerse [39], MathVista [22], MathVision [32]). This mix prevents a single MLLM from dominating and motivates instance-wise, budget-aware routing (Figure 3).

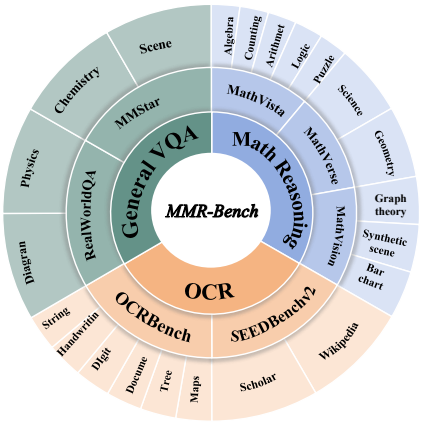

Fig. 3: Composition of **MMR-Bench** across scenarios and datasets.

Model zoo. The model set $\mathcal{F} = \{f_j\}_{j=1}^{K}$ includes both closed-source and open-weight MLLMs with heterogeneous architectures, capacities, and cost profiles:

- **Commercial MLLMs**, including GPT-5 series [1], Gemini 2.5 series [29], and Claude models, which offer strong general-purpose multimodal reasoning at higher monetary cost.
- **Open-weight MLLMs**, spanning 3B–72B parameters and including Gemma [30], InternVL3 [33], and Qwen2.5-VL families [3]. They are cheaper to deploy yet exhibit diverse OCR, grounding, and reasoning behavior.

All models share fixed prompts and decoding settings within each scenario. For every instance–model pair, we provide a task-specific utility $u_{i,j} \in [0, 1]$ and a normalized inference cost $c_{i,j} \geq 0$, forming the offline outcome table used throughout our routing evaluation.

Overall, MMR-Bench contains 11,000 instances across 10 models, 8 datasets, and 3 scenarios.

4.3 Cost-aware Routing Evaluation Protocol

MMR-Bench evaluates routing policies entirely in an *offline* fashion using the precomputed instance–model outcomes from Equation (1). Following that formulation, we use the support set

$$\mathcal{S} = \left\{ ((x_i^{\text{text}}, x_i^{\text{img}}), \mathbf{u}_i, \mathbf{c}_i) \right\}_{i=1}^{n}. \tag{5}$$

Routing behavior. For an input x_i (optionally with a budget signal b_i), a routing policy R_θ outputs a distribution over models

$$R_\theta(x_i, b_i) \in \Delta^{K-1}, \quad \pi_\theta(j \mid x_i, b_i) = [R_\theta(x_i, b_i)]_j. \tag{6}$$

The per-instance expected utility and cost are obtained by weighting the precomputed $\{u_{i,j}, c_{i,j}\}$ with the router's distribution $\pi_\theta(\cdot \mid x_i, b_i)$; dataset-level means summarize each method as $\mathrm{Perf}(R_\theta)$ and $\mathrm{Cost}(R_\theta)$. Varying operating points produces $(\mathrm{Cost}, \mathrm{Perf})$ pairs whose Pareto upper envelope defines the performance–cost curve $p(c)$, evaluated over the shared range $[c_{\min}, c_{\max}]$ set by always choosing the cheapest and most expensive single-model baselines.

Routing constraints. Routers may use only inference-time information: (i) the query $(x_i^{\text{text}}, x_i^{\text{img}})$ and optional budget b_i; (ii) benchmark lightweight features (e.g., embeddings, token counts, scenario tags); (iii) static model metadata (e.g., normalized cost, context length, supported modalities). They may not access ground-truth labels or $\{u_{i,j}\}$ on the evaluation split, nor issue multiple adaptive model calls per instance, since MMR-Bench fixes one outcome per instance–model pair. Oracle selectors (e.g., $\arg\max_j u_{i,j}$) are reported only as analysis upper bounds.

Metrics. All utilities are normalized to $[0, 1]$ to enable aggregation across tasks and models. Following standard practice in routing and model selection [11, 12], we evaluate routing performance by tracing a performance–cost curve obtained by varying the user cost budget. Prior work typically summarizes this curve with three metrics: **(1) nAUC** (Normalized AUC), the normalized area under the performance–cost curve measuring overall efficiency across budgets; **(2)** P_s (Peak Score), the best performance achieved on the curve together with its corresponding cost; and **(3) QNC** (Quality–Neutral Cost), the relative cost required to match the performance of the most accurate standalone model. Together, these metrics summarize average routing quality, best attainable utility, and sensitivity to budget constraints.

5 Experiments

5.1 Experimental Setup

Datasets and splits. We use the three scenarios and eight datasets in Sec. 4.2 with frozen 2:8 train/test splits. Routers are trained on the training split, and test-time evaluation is purely offline: we apply the learned router to benchmark features and index into the fixed $\{u_{i,j}, c_{i,j}\}$ (Sec. 4.3).

Models and cost. We route over the K=10 candidate MLLMs in Sec. 4.2. All per-instance utilities $u_{i,j} \in [0, 1]$ and normalized monetary costs $c_{i,j} \in \mathbb{R}_+$ are precomputed under a unified cost scheme and shared by all methods. The shared cost range $[c_{\min}, c_{\max}]$ is defined by always selecting the cheapest versus the most expensive single model.

Metrics and aggregation. Following Sec. 4.3, we report the performance–cost curve $p(c)$ and summarize methods with nAUC and P_s, with QNC included where relevant. Metrics are computed per dataset and then macro-averaged within each scenario and across all scenarios. Unless otherwise noted, results for stochastic routers are averaged over multiple runs.

Table 2: Performance of different routing methods across datasets and settings. **Bold numbers** indicate the best performance (excluding Oracle and Best Single), and underlined numbers indicate the second-best (excluding Oracle and Best Single).

Scenario	**OCR**				**General VQA**			
Dataset	OCRBench		SeedBenchV2Plus		MMStar		RealWorldQA	
Method \ Metric	nAUC (↑)	P_s (↑)	nAUC (↑)	P_s (↑)	nAUC (↑)	P_s (↑)	nAUC (↑)	P_s (↑)
Random	–	0.8075	–	0.7102	–	0.6942	–	0.7108
Linear	0.8789	0.9088	0.7376	0.7458	0.7496	0.7683	0.8036	0.8252
EmbedLLM	0.8952	0.9088	**0.7443**	**0.7508**	0.7615	**0.7817**	0.8097	0.8268
k-NNRouter	0.8913	**0.9150**	0.7375	0.7425	0.7560	0.7767	0.8009	0.8252
AvengersPro	**0.9126**	0.9138	0.7382	0.7475	0.7564	0.7733	0.8072	0.8235
GraphRouter	0.8874	0.8975	0.7175	0.7136	0.7546	0.7786	0.7942	0.8076
RM-Softmax	0.9060	0.9077	0.7341	0.7482	**0.7625**	0.7714	**0.8123**	**0.8301**
Oracle	0.9663	0.9825	0.8854	0.8913	0.9563	0.9650	0.9773	0.9853
Best Single	1.0000	0.9050	1.0000	0.7502	1.0000	0.7716	1.0000	0.8235

Scenario	**Reasoning**						**Avg**	
Dataset	MathVista		MathVerse		MathVision		Full Dataset	
Method \ Metric	nAUC (↑)	P_s (↑)	nAUC (↑)	P_s (↑)	nAUC (↑)	P_s (↑)	nAUC (↑)	P_s (↑)
Random	–	0.6688	–	0.4800	–	0.3927	–	0.6135
Linear	0.7724	0.8038	0.6837	0.7258	0.5565	0.6826	**0.7042**	**0.7533**
EmbedLLM	0.7751	0.8038	0.6784	**0.7337**	0.5494	0.6937	0.6913	0.7494
k-NNRouter	**0.7807**	**0.8075**	**0.6881**	0.7194	0.5610	0.6867	0.6950	0.7457
AvengersPro	0.7763	0.8038	0.6835	0.7194	0.5639	0.6933	0.6885	0.7496
GraphRouter	0.7423	0.7998	0.6734	0.7117	**0.5734**	0.6942	0.6754	0.7420
RM-Softmax	0.7794	0.8055	0.6845	0.7281	0.5724	**0.6944**	0.6927	0.7479
Oracle	0.9310	0.9475	0.8806	0.8954	0.8031	0.8741	0.8897	0.9188
Best Single	1.0000	0.8037	1.0000	0.7194	1.0000	0.6932	1.0000	0.7412

5.2 Baselines

Feature fusion. We first extract frozen embeddings offline: images are encoded with a ViT-based encoder and text with a pretrained language encoder. We then construct a per-instance feature z_i from these embeddings as follows:

- **Equal-weight mean.** Align the text and image embedding dimensions; if a modality is missing, fill with a zero vector. Fuse by averaging the two embeddings with equal weight to obtain a single joint representation.
- **Adaptive weights.** We estimate per-modality confidence (cosine-to-mean and a norm-based sigmoid), convert it to softmax weights, and fuse features via weighted sum, product, and absolute difference, followed by a linear mix and ℓ_2 normalization.

We analyze fusion choices later in Sec. 6.

Routing methods. All methods take an instance representation z_i (optionally with a budget b_i) and output a routing decision over K candidate models: (1) **Random (lower bound)** randomly selects a feasible model under the cost constraint; (2) **Oracle (upper bound)** selects the best feasible model

using per-instance ground-truth outcomes; (3) **Linear** learns a linear scoring function over z_i (and b_i) for each model and chooses the highest-scoring feasible one; (4) **EmbedLLM** [42] routes by using compact learned representations of LLMs and a lightweight predictor to select the most suitable model; (5) **k-NNRouter** retrieves the k nearest instances to z_i and selects the model with the best aggregated neighbor performance under budget; (6) **AvengersPro** [41] performs test-time routing that selects exactly one model per query to optimize the performance–efficiency trade-off of an LLM ensemble; (7) **GraphRouter** [7] constructs a graph over queries/tasks and candidate models and selects a model by predicting query–model compatibility on the graph; (8) **RM-Softmax** [31] learns an end-to-end routing policy from logged observational data by minimizing regret with a softmax-based objective.

Training and evaluation. Learned routers are trained on the training split, and test-time evaluation is purely offline by indexing into the fixed $\{u_{i,j}, c_{i,j}\}$ under the normalized cost scheme (Sec. 4.3). All methods use the same z_i for a controlled comparison.

5.3 Overall Results on MMR-Bench

Comparative Analysis. Table 2 shows that learned routers consistently outperform the heuristic lower bound, with complementary strengths across workloads. On the full-dataset average, **Linear** performs best, achieving the highest nAUC and peak score (**0.7042**/**0.7533**), followed by **k-NNRouter**. Instance-based methods excel on specific datasets: **k-NNRouter** leads P_s on *OCRBench* and nAUC/P_s on *MathVista*; **AvengersPro** leads nAUC on *OCRBench*; and **GraphRouter** leads nAUC on *MathVision*. Policy/representation-based routers are stronger on general VQA: **RM-Softmax** is best on *MMStar* and *RealWorldQA*, while **EmbedLLM** achieves the best P_s on *MMStar*. Overall, **Linear** is the most stable on aggregate, while retrieval, graph, and regret-based methods peak on particular distributions.

Cost efficiency and stability. The results point to a general conclusion: routers that *explicitly model per-model outcomes in a low-rank space* produce smoother, budget-robust frontiers. Their higher macro nAUC and competitive P_s indicate sustained quality across budgets rather than spikes at isolated operating points, suggesting they capture transferable, modality-aware difficulty signals rather than dataset-specific quirks. In contrast, purely instance-based or clustering dispatchers can peak on narrow distributions but are brittle when budgets or domains shift. *Practical takeaway:* under variable or uncertain budgets, performance-modeling routers that learn smooth global frontiers are the safer default; purely instance-based dispatchers are better reserved for stable, well-characterized workloads.

Pareto-front comparison. To make the budget–quality trade-off visually clear, we plot accuracy against normalized cost and report the Pareto envelope of each router, with single-model baselines shown as isolated points (Figure 4). We rescale the x-axis to magnify the low-cost region where most practical operating points lie. The resulting frontiers show that instance-wise routing consistently

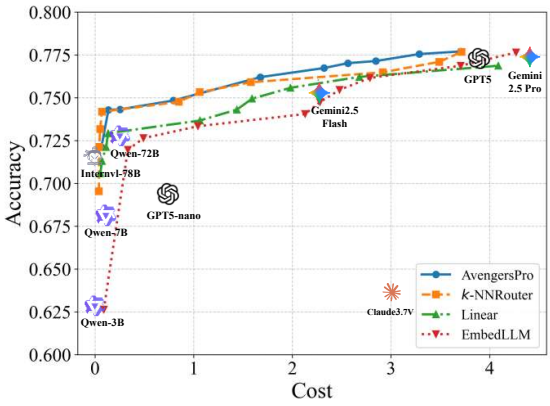

Fig. 4: Cost-accuracy Pareto frontiers on MMR-Bench. Lines of different styles represent different routing strategies, while single-model positions are indicated by their logos. For readability, the x-axis is log-scaled to emphasize the low-cost region. Compared with any fixed model, routing shifts the frontier upward and leftward, indicating higher accuracy at the same cost and lower cost at the same accuracy.

dominates fixed-model choices: with only **33**% of the strongest single model's cost, our router already matches its accuracy, and at higher budgets it *surpasses* that model, yielding a uniformly stronger cost–accuracy trade-off. This demonstrates that adaptive selection over a heterogeneous MLLM pool is effective in practice, especially under tight or variable budget constraints.

6 Analyses

Building on the overall routing results in Sec. 5.3, this section briefly analyzes the sources of routing gains, focusing on modality gaps revealed by adaptive fusion, robustness to intra-scenario distribution shift, and cross-modal transfer from multimodal to unimodal tasks.

(1) Adaptive fusion reveals a clear modality gap between image and text signals.

Table 3 contrasts adaptive fusion with equal-weight averaging under fixed router families. Adaptive yields the largest lift for **AvengersPro** (ΔnAUC $=$ $+0.3403$; QNC $+\infty \to 1.0585$), small but consistent gains for **EmbedLLM** (QNC $+\infty \to$ **0.9947** < 1), modest average improvements for **Linear** (ΔnAUC $=$ $+0.0124$) with a slightly higher QNC ($0.9055 \to 0.9701$), and mild regressions for k**-NNRouter** (ΔnAUC $= -0.0074$; QNC remains $+\infty$).

These outcomes provide direct evidence of a *modality gap* in multimodal routing: the information carried by image and text is *not* uniformly balanced

Table 3: Adaptive minus equal fusion. Positive Δ is better for nAUC/P_s; lower QNC is better.

Method	Δ nAUC	ΔP_s	**QNC change**
AvengersPro	+0.3403	+0.1275	$+\infty \rightarrow 1.0585$
EmbedLLM	+0.0033	+0.0020	$+\infty \rightarrow 0.9947$
Linear	+0.0124	+0.0036	$0.9055 \rightarrow 0.9701$
k-NNRouter	−0.0074	−0.0014	$+\infty \rightarrow +\infty$
GraphRouter	+0.0025	+0.0020	$+\infty \rightarrow 0.9998$
RM-Softmax	+0.0117	+0.0125	$1.1747 \rightarrow 0.9814$

Table 4: Within-scenario cross-dataset robustness measured by peak score P_s (↑ higher is better). Each column reports a scenario; rows compare the best single model on the target dataset B with the cross-dataset router $A \rightarrow B$ (**Shift**)

Method	**OCR**	**VQA**	**Math**
Best model	0.7062	0.7936	0.7592
Shift	**0.7234**	**0.8012**	**0.7914**

across tasks or instances. Equal-weight fusion imposes an implicit parity assumption between modalities; when one modality dominates or the modalities disagree, this assumption is violated, yielding miscalibrated scores and suboptimal model choices. Adaptive fusion explicitly *reweights* modalities at the instance level and exposes *agreement/mismatch* via multiplicative and difference interactions. Overall, these fusion results substantiate the modality-gap hypothesis: because modality importance varies across workloads, routers that sense and exploit this variation dominate naive equal averaging on the cost–performance frontier.

(2) Routing policies can remain robust under intra-scenario distribution shifts.

To test whether routing captures *scenario-consistent*, modality-aware difficulty cues rather than overfitting to dataset-specific biases, we run a within-scenario cross-dataset evaluation: for each scenario, the router is evaluated on a different dataset from the same scenario. As shown in Table 4, the cross-dataset router consistently achieves a higher peak score P_s than the best single model and remains close to the in-domain router.

The cross-dataset results demonstrate robustness to within-scenario distribution shift: the router is not memorizing dataset quirks but capturing scenario-consistent, modality-aware difficulty signals. From a multimodal perspective, the transferable signals include instance-level modality salience (relative importance of image versus text), cross-modal agreement or mismatch, and structural cues such as OCR density, layout complexity, and question length. Our adaptive, interaction-augmented fusion exposes these cues through reweighting, product,

Table 5: Cross-modality transfer to *text-only* benchmarks. The router is trained on *multimodal* workloads and evaluated under text-only inference by masking the image channel. We report peak score P_s (↑ higher is better). Rows compare (i) the best single-model baseline on each text benchmark and (ii) our multimodal-trained router evaluated in the text-only setting (MM↛Text).

Method	GSM8K	MMLU	ARC
Best single model	94.5	91.2	65.7
MM↛Text (ours)	**96.7**	**92.4**	**66.7**

and difference terms, which remain stable at the scenario level and thus support transfer. The small residual gap to the in-domain router is attributable to dataset-specific biases rather than a failure to generalize.

(3) Multimodal routers can transfer across modalities and generalize to unimodal tasks.

Building on the previous robustness analysis, we test robustness across *modalities*. We train routers on multimodal workloads using the fusion features in Sec. 5.2, then evaluate zero-shot on text-only benchmarks by freezing the router and masking the image embedding at inference. We compare against the best single-model baseline and the multimodal-trained router under text-only inference.

Table 5 shows that the multimodal-trained router remains effective without images: its P_s exceeds the best single model on GSM8K/MMLU/ARC. We attribute this to (i) modality-agnostic difficulty cues captured from text, (ii) fusion-induced regularization from sum/product/difference interactions that degrades gracefully when images are masked, and (iii) cost-aware calibration that promotes conservative selection under uncertainty. Overall, the router is robust to modality drop and transfers to text-only workloads without retraining.

7 Conclusion

We present **MMR-Bench**, an offline, cost-aware benchmark for routing over a heterogeneous collection of multimodal large language models. By providing precomputed utilities and normalized costs for each instance–model pair, along with unified evaluation metrics, MMR-Bench enables controlled, reproducible comparison of routing policies without rerunning any model. Our experiments demonstrate that routing significantly improves the cost–accuracy trade-off over the best single model, sometimes matching or exceeding its accuracy at roughly one-third of the cost, with matrix-factorization-based routers achieving the most robust performance across heterogeneous workloads. Further analyses show that multimodal observables with adaptive fusion are critical for strong routing, closing the gap left by unimodal policies, and that the learned routers remain sta-

ble under intra-scenario distribution shifts and can transfer to text-only benchmarks. We hope that MMR-Bench will serve as a standardized testbed for future research on cost-aware multimodal routing and the practical deployment of MLLMs under realistic budget constraints.

Acknowledgements

This work was supported in part by the National Key Research and Development Program of China (No. 2024YFE0202800), the Basic Research Program of Jiangsu under Grant No. BK20253021, the National Natural Science Foundation of China (NSFC) under Grants No. 62522605 and No. 62376118, the Fundamental and Interdisciplinary Disciplines Breakthrough Plan of the Ministry of Education of China (No. JYB2025XDXM118), the "111 Center" (No. B26023), and the Collaborative Innovation Center of Novel Software Technology and Industrialization.

References

1. Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F.L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al.: Gpt-4 technical report. arXiv preprint arXiv:2303.08774 (2023)
2. Aggarwal, P., Madaan, A., Anand, A., Potharaju, S.P., Mishra, S., Zhou, P., Gupta, A., Rajagopal, D., Kappaganthu, K., Yang, Y., et al.: Automix: Automatically mixing language models. In: NeurIPS (2024)
3. Bai, S., Chen, K., Liu, X., Wang, J., Ge, W., Song, S., Dang, K., Wang, P., Wang, S., Tang, J., et al.: Qwen2. 5-vl technical report. arXiv preprint arXiv:2502.13923 (2025)
4. Chen, L., Li, J., Dong, X., Zhang, P., Zang, Y., Chen, Z., Duan, H., Wang, J., Qiao, Y., Lin, D., et al.: Are we on the right way for evaluating large vision-language models? In: NeurIPS (2024)
5. Chen, L., Zaharia, M., Zou, J.: Frugalgpt: How to use large language models while reducing cost and improving performance. arXiv preprint arXiv:2305.05176 (2023)
6. Chen, S., Jiang, W., Lin, B., Kwok, J., Zhang, Y.: Routerdc: Query-based router by dual contrastive learning for assembling large language models. In: NeurIPS (2024)
7. Feng, T., Shen, Y., You, J.: Graphrouter: A graph-based router for llm selections. arXiv preprint arXiv:2410.03834 (2024)
8. Hu, Q.J., Bieker, J., Li, X., Jiang, N., Keigwin, B., Ranganath, G., Keutzer, K., Upadhyay, S.K.: Routerbench: A benchmark for multi-llm routing system. arXiv preprint arXiv:2403.12031 (2024)
9. Huang, Z., Ling, G., Lin, Y., Chen, Y., Zhong, S., Wu, H., Lin, L.: Routereval: A comprehensive benchmark for routing llms to explore model-level scaling up in llms. arXiv preprint arXiv:2503.10657 (2025)
10. Jiang, D., Ren, X., Lin, B.Y.: Llm-blender: Ensembling large language models with pairwise ranking and generative fusion. arXiv preprint arXiv:2306.02561 (2023)
11. Jitkrittum, W., Narasimhan, H., Rawat, A.S., Juneja, J., Wang, Z., Lee, C.Y., Shenoy, P., Panigrahy, R., Menon, A.K., Kumar, S.: Universal llm routing with correctness-based representation. In: ICLR workshop (2025)

12. Lai, G., Hu, H., Chen, L., Li, Z., Ye, H.J.: From sampled outcomes to capability distributions: Rethinking supervision for llm routing. arXiv preprint arXiv:2606.06924 (2026)
13. Lai, G., Hu, H., Ye, H.J.: Routejudge: Preference-based evaluation of llm routers under pluralistic user preferences. In: Pluralistic Alignment Workshop at ICML 2026 (2026)
14. Lai, G., Ye, H.J.: When routing collapses: On the degenerate convergence of llm routers. arXiv preprint arXiv:2602.03478 (2026)
15. Li, B., Wang, R., Wang, G., Ge, Y., Ge, Y., Shan, Y.: Seed-bench: Benchmarking multimodal llms with generative comprehension. arXiv preprint arXiv:2307.16125 (2023)
16. Li, F., Zhang, R., Zhang, H., Zhang, Y., Li, B., Li, W., Ma, Z., Li, C.: Llava-next-interleave: Tackling multi-image, video, and 3d in large multimodal models. arXiv preprint arXiv:2407.07895 (2024)
17. Li, H., Zhang, Y., Guo, Z., Wang, C., Tang, S., Zhang, Q., Chen, Y., Qi, B., Ye, P., Bai, L., et al.: Llmrouterbench: A massive benchmark and unified framework for llm routing. arXiv preprint arXiv:2601.07206 (2026)
18. Liang, V.W., Zhang, Y., Kwon, Y., Yeung, S., Zou, J.Y.: Mind the gap: Understanding the modality gap in multi-modal contrastive representation learning. In: NeurIPS (2022)
19. Lin, B., Tang, Z., Ye, Y., Cui, J., Zhu, B., Jin, P., Huang, J., Zhang, J., Pang, Y., Ning, M., et al.: Moe-llava: Mixture of experts for large vision-language models. arXiv preprint arXiv:2401.15947 (2024)
20. Liu, H., Li, C., Wu, Q., Lee, Y.J.: Visual instruction tuning. In: NeurIPS (2023)
21. Liu, Y., Li, Z., Huang, M., Yang, B., Yu, W., Li, C., Yin, X.C., Liu, C.L., Jin, L., Bai, X.: Ocrbench: on the hidden mystery of ocr in large multimodal models. SCIS (2024)
22. Lu, P., Bansal, H., Xia, T., Liu, J., Li, C., Hajishirzi, H., Cheng, H., Chang, K.W., Galley, M., Gao, J.: Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. arXiv preprint arXiv:2310.02255 (2023)
23. Lu, S., Li, Y., Xia, Y., Hu, Y., Zhao, S., Ma, Y., Wei, Z., Li, Y., Duan, L., Zhao, J., et al.: Ovis2. 5 technical report. arXiv preprint arXiv:2508.11737 (2025)
24. Lu, Y., Liu, R., Yuan, J., Cui, X., Zhang, S., Liu, H., Xing, J.: Routerarena: An open platform for comprehensive comparison of llm routers. arXiv preprint arXiv:2510.00202 (2025)
25. Malinowski, M., Fritz, M.: A multi-world approach to question answering about real-world scenes based on uncertain input. In: NeurIPS (2014)
26. Šakota, M., Peyrard, M., West, R.: Fly-swat or cannon? cost-effective language model choice via meta-modeling. In: ICDE (2024)
27. Shnitzer, T., Ou, A., Silva, M., Soule, K., Sun, Y., Solomon, J., Thompson, N., Yurochkin, M.: Large language model routing with benchmark datasets. arXiv preprint arXiv:2309.15789 (2023)
28. Song, W., Huang, Z., Cheng, C., Gao, W., Xu, B., Zhao, G., Wang, F., Wu, R.: Irt-router: Effective and interpretable multi-llm routing via item response theory. arXiv preprint arXiv:2506.01048 (2025)
29. Team, G., Anil, R., Borgeaud, S., Alayrac, J.B., Yu, J., Soricut, R., Schalkwyk, J., Dai, A.M., Hauth, A., Millican, K., et al.: Gemini: a family of highly capable multimodal models. arXiv preprint arXiv:2312.11805 (2023)
30. Team, G., Kamath, A., Ferret, J., Pathak, S., Vieillard, N., Merhej, R., Perrin, S., Matejovicova, T., Ramé, A., Rivière, M., et al.: Gemma 3 technical report. arXiv preprint arXiv:2503.19786 (2025)

31. Tsiourvas, A., Sun, W., Perakis, G.: Causal LLM routing: End-to-end regret minimization from observational data. In: NeurIPS (2025)
32. Wang, K., Pan, J., Shi, W., Lu, Z., Ren, H., Zhou, A., Zhan, M., Li, H.: Measuring multimodal mathematical reasoning with math-vision dataset. In: NeurIPS (2024)
33. Wang, W., Gao, Z., Gu, L., Pu, H., Cui, L., Wei, X., Liu, Z., Jing, L., Ye, S., Shao, J., et al.: Internvl3. 5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. arXiv preprint arXiv:2508.18265 (2025)
34. Wu, J., Gan, W., Chen, Z., Wan, S., Yu, P.S.: Multimodal large language models: A survey. In: IEEE BigData (2023)
35. Wu, Q., Ke, Z., Zhou, Y., Sun, X., Ji, R.: Routing experts: Learning to route dynamic experts in multi-modal large language models. arXiv preprint arXiv:2407.14093 (2024)
36. Yang, J., Wu, Q., Feng, Z., Zhou, Z., Guo, D., Chen, X.: Quality-of-service aware llm routing for edge computing with multiple experts. IEEE TMC (2025)
37. Yin, S., Fu, C., Zhao, S., Li, K., Sun, X., Xu, T., Chen, E.: A survey on multimodal large language models. National Science Review (2024)
38. Yue, X., Ni, Y., Zhang, K., Zheng, T., Liu, R., Zhang, G., Stevens, S., Jiang, D., Ren, W., Sun, Y., et al.: Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In: CVPR (2024)
39. Zhang, R., Jiang, D., Zhang, Y., Lin, H., Guo, Z., Qiu, P., Zhou, A., Lu, P., Chang, K.W., Qiao, Y., et al.: Mathverse: Does your multi-modal llm truly see the diagrams in visual math problems? In: ECCV (2024)
40. Zhang, Y.K., Zhan, D.C., Ye, H.J.: Capability instruction tuning: A new paradigm for dynamic llm routing. In: AAAI (2025)
41. Zhang, Y., Li, H., Chen, J., Zhang, H., Ye, P., Bai, L., Hu, S.: Beyond gpt-5: Making llms cheaper and better via performance-efficiency optimized routing. In: Proceedings of the 2025 7th International Conference on Distributed Artificial Intelligence. pp. 122–129 (2025)
42. Zhuang, R., Wu, T., Wen, Z., Li, A., Jiao, J., Ramchandran, K.: Embedllm: Learning compact representations of large language models. arXiv preprint arXiv:2410.02223 (2024)