MMR-BENCH V2: FROM VISUAL INFORMATION TO ROUTING DECISIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Multimodal large language models vary in capability and inference cost, motivating query-dependent routing. When do model diversity and visual information translate into better model choices? MMR-Bench V2 expands MMR-Bench to 44 models and 18 benchmarks, recording outcomes, execution status, and reference costs for every query–model pair. We examine model-pool composition, query information, and target-task adaptation. At the pool level, expansion increases answer coverage faster than routing quality. Compact global pools retain the quality of unrestricted task-conditioned policies within a predefined tolerance at medium and high validation budgets on the evaluated suite. At the information level, matched frozen-feature comparisons show that images can improve outcome prediction while worsening selection. Most prediction gains concern the mean success rate across candidates for a given query. Visual effects depend on task composition, and preferences for models that generally perform well on each task provide strong controls when those tasks are represented in training. For known-task adaptation with matched prior weights, concentrating target labels on candidates screened using other tasks can outperform tested broader uniform and sequential allocations at the same label budget. These findings favor learning task-level preferences and evaluating larger pools and richer inputs by their additional selection gains.

1 INTRODUCTION

Multimodal large language models (MLLMs) support document understanding, chart interpretation, and visual reasoning (Bai et al., 2025). These applications demand different combinations of perception, text recognition, and reasoning. Available models vary in both capability and inference cost (Ma et al., 2026; Huang et al., 2026). Multimodal model routing addresses this heterogeneity by selecting a suitable candidate for each query (Figure 1). The central question is when model diversity and visual information translate into better model choices.

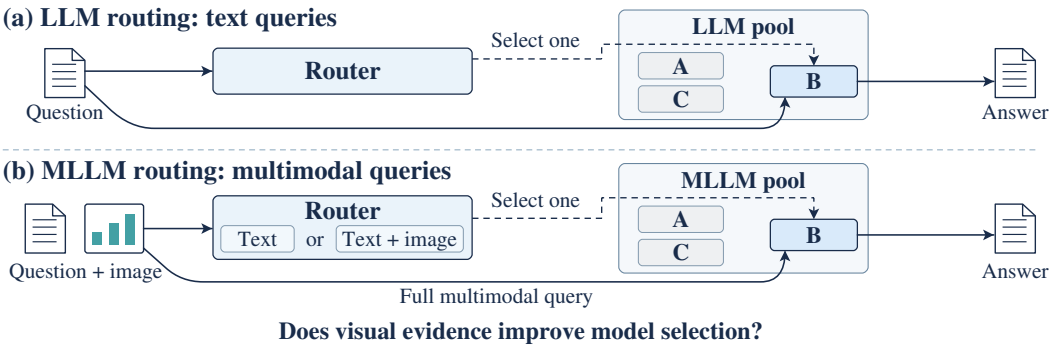

Figure 1: Model routing for text and multimodal queries. Dashed arrows select one candidate; solid arrows carry inputs and outputs. Our modality comparison is within (b): the router sees text or image–text inputs, while the selected MLLM receives the full multimodal query in both conditions.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Language-model routers learn query-dependent choices from prior performance data (Ong et al., 2025). MMR-Bench reported gains from multimodal fusion, cost-aware selection, and transfer (Ma et al., 2026). Yet an improvement over one globally fixed model can arise from recognizing a task and choosing a model that usually performs well on it. These task-level model preferences provide a stronger reference for testing query-specific selection. Testing visual information requires distinguishing this task recognition from better choices among queries within the same task. Likewise, adding candidates increases answer availability without ensuring that the router selects those answers (Shihab et al., 2026).

We examine three connected aspects of this problem: model-pool composition, query information, and target-task adaptation. Pool composition concerns whether adding candidates helps the router make better choices. Query information concerns whether images improve individual model choices beyond task-level preferences. Adaptation concerns how to learn those preferences from limited evaluations on a new task. Under a fixed label budget, evaluating more candidates trades off against estimating each candidate's performance from more queries.

To support these comparisons, we introduce MMR-Bench V2, covering 44 candidate models, 25,504 queries, and 18 benchmarks (Figure 2). Its complete query–model matrix holds candidate answers fixed across policies, permitting paired tests of pool composition, router inputs, and label allocation. Broader task coverage allows us to assess how findings depend on workload composition. Execution status separates answer-selection gains from service recovery, while reference costs support budget-controlled comparisons. V2 combines expanded coverage with task-preference controls and prediction–selection diagnostics, moving beyond aggregate routing gains to examine what improves model choice (Section 4).

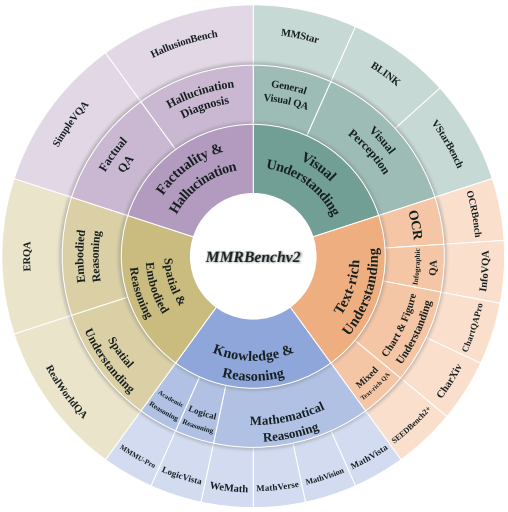

Figure 2: Task taxonomy of MMR-Bench V2. Inner to outer: capability families, task groups, and the 18 source benchmarks.

The analyses yield three findings. First, pool expansion adds answer coverage faster than routing quality, while compact global pools retain the quality of unrestricted task-conditioned policies within a predefined tolerance at medium and high validation budgets on the fixed suite. Second, matched frozen-feature heads and LatentRouter show that images can improve outcome prediction while worsening selection, with improvements concentrated in estimating the mean success rate across candidates for a given query. Visual effects depend on task composition, and task preferences remain strong controls when tasks are represented in training. Third, with target-task identity known, concentrating a fixed label budget on candidates screened using other tasks outperforms the tested 16- and 32-candidate uniform and sequential allocations at matched prior weight. Together, these findings favor evaluating model pools and visual inputs by gains in selection, using task preferences learned under matched supervision as a reference.

Our main contributions are as follows:

- We introduce **MMR-Bench V2**, an expanded multimodal routing resource with paired outcomes, execution status, and reference costs. Its task and model coverage supports controlled comparisons of candidate pools, visual inputs, and target-label allocations.

- We develop **decision-level comparisons** that distinguish answer availability, outcome prediction, and final model choice. Task-preference controls test the added value of query-specific information, while execution-aware and source-held-out analyses examine the scope of the resulting gains.

- We provide **evidence for pool selection and label allocation**: compact global pools can retain quality under the tested budgets, and source-informed target calibration can outperform broader evaluation. Controlled comparisons examine the roles of candidate screening, sample count, and source priors.

2 Related Work

2.1 Multimodal evaluation and routing benchmarks

Multimodal evaluation characterizes model capabilities across visual understanding, text recognition, and reasoning (Chen et al., 2024; Liu et al., 2024; Lu et al., 2024). Routing evaluates a selection policy, requiring paired query–model outcomes rather than aggregate rankings. Text-only resources support quality–cost evaluation with RouterBench (Hu et al., 2024) and model-pool scaling with RouterEval (Huang et al., 2025). LLMRouterBench finds diminishing returns from pool expansion, emphasizes candidate curation, and relates router gains to coarse-grained task structure (Li et al., 2026). ORBIT supplies shared implementations and budget-aware protocols (Lai et al., 2026). MMR-Bench studies modality-aware selection and transfer (Ma et al., 2026); VL-RouterBench evaluates visual routing through accuracy, cost, and throughput (Huang et al., 2026). MMR-Bench V2 expands multimodal task and model coverage to examine pool composition jointly with router-visible visual information and limited target supervision.

2.2 Model selection and routing diagnostics

Learned routers estimate candidate advantages from query features, preference data, or model embeddings (Ong et al., 2025; Zhuang et al., 2025). LatentRouter models interactions between image–question requirements and candidate capabilities (Cheng & Dong, 2026). EquiRouter addresses the mismatch between scalar-score prediction and routing decisions through ranking-based learning (Lai & Ye, 2026). Other diagnostics distinguish outcome-oracle opportunity from realizable routing gains (Shihab et al., 2026) and separate reproducible specialist advantages from single-draw label noise (Chen, 2026). Our matched text-only and joint-input comparisons decompose prediction error into candidate-shared and candidate-relative components, then relate visual changes to shortlist coverage and final selection against task-conditioned controls. Learning task preferences under scarce supervision also connects to budgeted model selection (Jamieson & Talwalkar, 2016). Whereas that work allocates training resources among hyperparameter configurations, our adaptation study allocates outcome labels across fixed pretrained models and target queries.

3 Preliminaries

3.1 Problem definition

For a multimodal query x_i, a score-based router selects a model from candidate pool $\mathcal{M}$ before any candidate response is generated:

$$\mathbf{s}_i = g_\theta(\phi(x_i)) \in \mathbb{R}^{|\mathcal{M}|}, \qquad \pi_\theta(x_i) = \arg\max_{m \in \mathcal{M}} s_{im}, \tag{1}$$

where $\phi(x_i)$ contains permitted features and s_{im} is a predicted decision score. Recorded outcomes provide strict correctness $y_{im} \in \{0, 1\}$ and reference cost c_{im}. Training and validation outcomes support fitting and policy selection; test outcomes are reserved for evaluation. Randomized policies assign probabilities p_{im} over single-model actions; deterministic policies put all mass on one candidate. Query-independent mixtures reuse the same probabilities across queries, calling one model rather than ensembling answers.

Query-only policies access the question and, in joint conditions, the visual input. Predicted-task policies select the fit-set best model for an inferred task; soft task priors weight fit-set task–model profiles by predicted task probabilities. Known-task policies additionally receive the true benchmark identity and are reported separately. Changing the router's input modalities leaves the candidate models' recorded multimodal inputs and answers fixed.

Target-task adaptation is a separate known-task setting. Source tasks screen candidates, and a bounded number of outcomes from disjoint target training queries calibrates a task-level choice before test-time deployment. This setting adds target supervision; its label budget counts query–candidate outcomes rather than queries or dollars.

3.2 Prediction and model selection

Selection depends on relative scores. Adding a common offset a_i to all candidates leaves the decision unchanged:

$$\arg\max_{m\in\mathcal{M}}(s_{im}+a_i)=\arg\max_{m\in\mathcal{M}}s_{im}. \tag{2}$$

For probability-prediction heads, a valid shared shift can reduce prediction error without changing the ranking. Changes to relative scores can instead alter selection even as aggregate error falls. We therefore distinguish visual improvements in overall prediction, candidate-relative advantages, and final decisions. Pool-wide and Top-k coverage measure correct-answer availability; final selection tests whether the router uses it. Task priors test gains beyond task-level structure, while global fixed mixtures provide query-independent controls. Section 4 defines quality and cost evaluation.

4 MMR-Bench V2: Construction and Evaluation

4.1 Task and model coverage

MMR-Bench V2 contains 25,504 multimodal queries from 18 benchmarks, with recorded outcomes for 44 candidate models, yielding 1,122,176 query–model records. Table 1 compares routing resources, including MixInstruct (Jiang et al., 2023) and the vision track of xRouteBench (Feng et al., 2026), by source modality and scale. Dataset counts refer to source suites, while model counts describe recorded pools whose coverage can vary across tasks and splits. Figure 2 organizes the benchmarks into five broad domains: visual understanding, text-rich understanding, knowledge and reasoning, spatial and embodied reasoning, and factuality and hallucination. Its middle ring gives the finer task groups; Appendix A specifies domain membership.

Table 1: Routing resources by source modality and scale.

Resource	Multimodal	Datasets	Models
RouterBench	×	8	14
MixInstruct	×	4	11
VL-RouterBench	✓	14	17
xRouteBench (vision)	✓	3	18
MMR-Bench	✓	7	10
MMR-Bench V2	✓	18	44

The model pool combines open-weight and API-served models, with recorded identifiers defining its columns. The complete matrix permits candidate-count and label-allocation comparisons on paired queries, while broader task coverage exposes heterogeneous gains. Outcomes can be replayed under different routing policies without regenerating answers. Appendix A lists task configurations, query counts, and candidate models.

From MMR-Bench to V2. MMR-Bench evaluated fixed-pool routing, multimodal fusion, and zero-shot transfer (Ma et al., 2026). V2 extends these questions through broader coverage and controlled comparisons. We examine how expansion affects selection and which candidates retain policy quality. For visual inputs, we separate outcome prediction from model selection using task-preference controls. For new tasks, we study how to allocate limited target labels across candidates and queries (Section 6).

4.2 Outcome records and reference costs

We evaluate each query–model outcome through both answer quality and execution status. Strict correctness requires full credit under the source benchmark's scoring rule, with partial native scores retained separately. Failed executions count as incorrect in the primary analysis. Their recorded status lets us distinguish execution-failure recovery from improvements over successfully returned but incorrect answers. We additionally analyze the subset where all 44 models return successfully. The recorded matrix reflects heterogeneous collection settings rather than a common hardware and decoding configuration. Appendix A.1 details the records and their availability.

We use a frozen, model-specific price schedule to compare routing policies under a fixed accounting convention. A response has reference cost $c_{im} = n_{im} r_m / 10^6$, where n_{im} is its recorded or retokenized output-token count and r_m is the model's rate in USD per million output tokens. The schedule contains 12 official, three hosted, and 29 estimated rates. Estimated rates are pricing proxies derived from model-size and provider information without correctness labels. Appendix A.2 details the pricing snapshot and calibration. These values provide a reference for output-token costs rather than end-to-end deployment bills: they exclude input and image usage, unrecorded reasoning, router encoding, network overhead, and latency.

4.3 Evaluation protocol

We evaluate routing policies by task-macro quality and query-micro reference cost. For N evaluated queries, let N_{t_i} be the size of query i's task and $w_i = 1/(18 N_{t_i})$, giving equal weight to each benchmark:

$$Q(\pi) = \sum_i w_i \sum_m p_{im} y_{im}, \qquad C(\pi) = \frac{1}{N} \sum_i \sum_m p_{im} c_{im}. \tag{3}$$

The hindsight oracle uses $\max_m y_{im}$ to measure correct-answer coverage. Top-k coverage restricts this maximum to the k highest-ranked candidates, distinguishing shortlist coverage from final selection. Supplementary micro-quality and macro-cost results appear in Appendix C.

Image-grouped, disjoint training, validation, and test partitions prevent shared-image leakage, with test outcomes reserved for evaluation. Visual comparisons match encoder family and prediction protocol across text-only and joint inputs, keeping candidate answers fixed. In target-task adaptation, source data additionally exclude all image groups occurring in the target task, and acquired labels come only from its training partition. Paired image-group resampling estimates uncertainty in policy differences; task resampling assesses sensitivity to benchmark composition, and adaptation comparisons also resample label acquisitions. Intervals condition on recorded outcomes and fitted policies. Appendices B and D.10 provide the implementation and statistical details.

5 Experiments

5.1 Experimental Setup

The initial comparison uses the full model pool and evaluation protocol in Section 4. We evaluate ORBIT implementations (Lai et al., 2026), mostly with shared frozen CLIP text–image features. Encoder-specific exceptions and training configurations are in Appendix B. Validation task-macro quality selects each learned router's operating point from its cost-threshold and unconstrained quality policies. Later controlled analyses specify when they instead fix the quality-argmax rule.

5.2 Baselines

We compare 20 non-oracle ORBIT implementations, simple policies, and matched-feature controls, grouped as follows. **(1) Simple.** Uniform random samples a candidate independently of the query; Cheapest and Global fixed select the lowest-cost or highest-quality model on training data. **(2) ML-based.** These use frozen query features, including kNN and ridge predictors, latent-factor RouteLLM-MF (Ong et al., 2025) and EmbedLLM (Zhuang et al., 2025), GraphRouter (Feng et al., 2025), clustering-based Avengers-Pro (Zhang et al., 2025b), and RM prediction heads. Matched text, image, and joint variants separate input choice from architecture. **(3) SLM-based.** The small language model (SLM) baseline, RouteLLM-BERT (Ong et al., 2025), fine-tunes a pretrained text encoder and classifier. **(4) References.** Known-task fixed uses supplied task identity to select a training-best candidate; Oracle accesses test outcome labels to give a hindsight quality upper bound. Table 2 reports representative implementations, with complete results in Appendix C.

5.3 Overall Results on MMR-Bench v2

Table 2 shows that average routing gains coexist with domain-level regressions. RMInterval leads the evaluated ORBIT implementations, improving task-macro quality over Global fixed by 3.4 percentage points, while ORBIT kNN and ridge remain within 0.6 points of the leader. Most of its

Table 2: Representative results on MMR-Bench V2. Strict domain quality (%) averages tasks within each domain in Figure 2; overall quality averages all 18 tasks equally. Cost is the query-micro output reference cost (μUSD). Each learned router uses one validation-selected policy across domains; costs are unmatched. Bold marks the best displayed learned router per quality column. Known-task fixed receives task identity; Oracle uses test outcomes.

	Domain quality (%) ↑					Overall	
Policy	Visual	Text-rich	Knowledge	Spatial	Factuality	Quality ↑	Cost ↓
Simple							
Global fixed	77.71	81.79	77.57	77.21	72.31	78.14	605.4
ML-based							
ORBIT kNN	81.29	**81.89**	84.69	73.39	**76.27**	81.15	1105.3
Ridge, text	**82.78**	81.31	84.47	72.57	75.37	80.97	1039.3
Ridge, joint	81.86	81.25	84.67	73.71	75.53	81.02	1158.1
RouteLLM-MF	78.39	76.23	79.73	70.78	72.31	76.71	1099.6
EmbedLLM	79.84	81.05	79.09	71.92	72.22	78.20	946.3
GraphRouter	77.57	81.27	76.42	**77.21**	72.34	77.59	800.2
Avengers-Pro	81.56	78.61	84.60	71.23	73.54	79.71	1080.7
RMInterval	82.48	81.66	**85.42**	74.44	75.43	**81.56**	1096.2
SLM-based							
RouteLLM-BERT	76.36	75.62	79.31	68.19	71.34	75.67	1076.0
References							
Known-task fixed	86.04	82.13	85.59	77.21	77.12	82.83	1139.2
Oracle	99.27	95.88	98.05	98.47	95.59	97.42	—

aggregate gain comes from the Knowledge domain, while performance declines in Spatial; kNN and both ridge variants exhibit the same contrast. Text and joint ridge also reverse their ordering across domains despite nearly identical overall quality. The aggregate ranking therefore combines task-dependent benefits and harms, motivating a closer examination of where routing and visual inputs improve selection.

Task-level allocation provides a stronger reference than one globally fixed model. Known-task fixed exceeds every learned router's overall quality without making query-specific choices within a task. Because benchmark identity is supplied, this is an information-rich reference rather than a like-for-like query-only comparison. A separate query-only task-prediction study also outperforms its matched ridge controls (Appendix C). Beating Global fixed thus leaves open whether a router improves on task-level model preferences. Oracle measures a different opportunity: correct answers available anywhere in the pool, not those identifiable before generation. These references distinguish task specialization from the finer-grained selection studied in Section 6.

Higher quality also carries a spending tradeoff: RMInterval uses 1.81 times the query-micro reference cost of Global fixed, so its quality gain alone does not establish cost efficiency. In separate budget-controlled comparisons, medium-budget routers exceed global mixtures but trail query-predicted task mixtures on average under shared validation spending caps. The latter require no supplied task identity, making task structure relevant even to query-only cost comparisons. This ordering depends on task composition, and validation caps do not enforce equal test spending (Appendix D.5). Assessing routing value therefore requires both domain-level quality and task-conditioned cost controls, beyond an overall improvement over a single model.

6 Analyses

Routing gains depend both on the candidates available and on the evidence for choosing among them. We first examine which benefits of pool expansion translate into better choices. We then trace visual information from outcome prediction to model choice, using task-level preferences as a reference. Finally, we study how to learn those preferences when a new task offers only a limited evaluation budget.

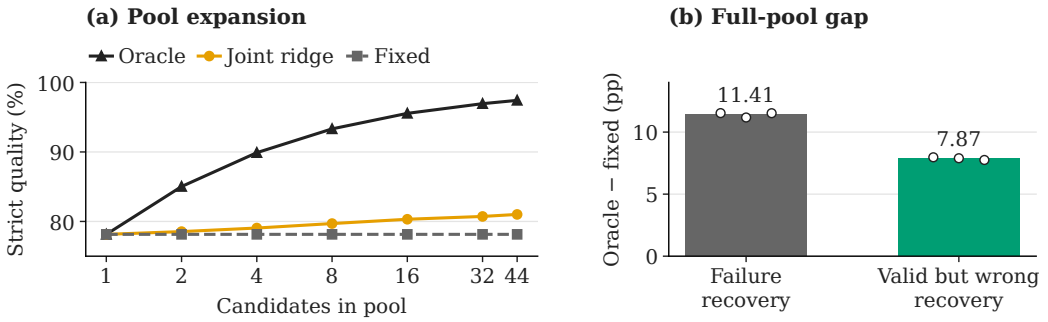

Figure 3: Coverage opportunity and realized quality. Left: fixed-model-anchored random nested pools, averaged over 50 paths per primary split. Right: the full-pool oracle gap above the training-selected fixed model separates execution-failure recovery from valid-but-wrong recovery. White points show the three overlapping splits, not independent replications.

(1) How does model-pool composition affect routing gains?

Coverage and selection. Pool expansion increases correct-answer availability faster than routing quality (Figure 3). Across random and fixed-model-anchored paths, added candidates provide answers that the router often fails to select. Starting from two strong training-selected anchors makes this distinction sharper: all eight frozen routers lose quality during expansion even as oracle coverage rises. For CLIP joint ridge, harmful switches outweigh beneficial ones. Additional candidates can therefore displace correct choices as well as supply new ones. The joint-minus-text gain also changes little with pool size, so broader coverage does not by itself produce a visual scaling benefit.

What constitutes complementarity? The oracle gap combines two benefits: recovering a failed execution and replacing a valid but incorrect answer. For the selected fixed reference and task weights, failure recovery accounts for the larger share. This makes execution status essential when interpreting coverage as evidence of complementary capabilities. Separately, matching pool size, best and mean capability, reference cost, and failure rate reveals a positive association between residual complementarity and small-pool routing quality, including on the all-model-success subset. The association weakens in larger pools, and residual diversity adds only modest predictive value beyond capability, cost, and failure descriptors (Appendix D.8). Composition matters, but diversity alone is an incomplete account of a pool's utility.

Compact global pools. Realized quality can nevertheless be retained with a compact pool. Predicted-task mixtures restricted to four globally shared candidates meet the prespecified 0.5 percentage-point non-inferiority margin against unrestricted mixtures at medium and high validation budgets in every fixed-suite comparison. This persists on all-model-success queries, although support is not uniform under task resampling. Native-score replay and validation reselection under price perturbations retain similar quality orderings. The unrestricted mixtures already use a subset of the library, so this compresses an active policy rather than reducing deployment from 44 models to four. Quality retention does not preserve full-pool coverage or guarantee test-budget feasibility (Appendix D.11). Pool design should account for the choices a policy can make, alongside the answers its candidates could provide.

(2) Does better visual prediction lead to better model selection?

What the prediction improvement captures. Images can improve outcome prediction while worsening model choice. With the same frozen Qwen2.5-VL-7B encoder and matched BCE heads, joint inputs lower Brier error but reduce routing quality from 81.30% to 80.08% (Figure 4). Text processing and pretraining are fixed; calibration affects the probability diagnostic, not selection. The released LatentRouter's query–model interaction architecture exhibits the same primary aggregate

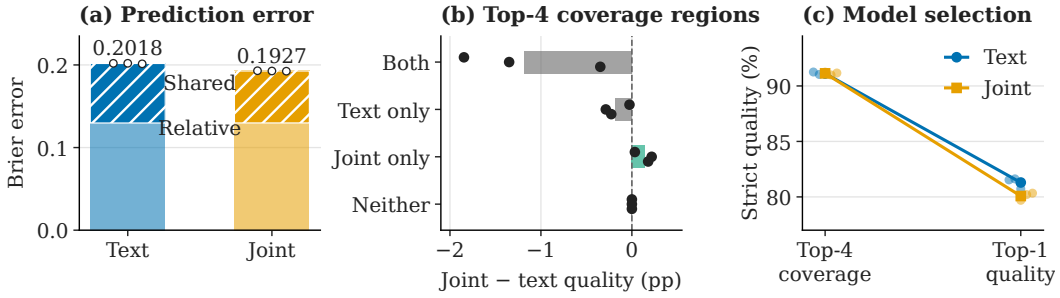

Figure 4: Visual prediction and selection with frozen VLM-BCE heads. Left: shared and relative Brier components. Middle: joint-minus-text quality contributions from the four Top-4 coverage regions; each uses the full population as denominator. Right: Top-4 coverage and Top-1 quality. Points show primary splits and bars their task-macro mean; uncertainty and task variation are discussed in the text.

ordering in a separate matched adaptation, extending the observation beyond independent prediction heads (Appendix D.12).

For query i and candidate m, let $\hat{s}_{im}$ be the predicted success probability, y_{im} its binary outcome, M the pool size, $e_{im} = \hat{s}_{im} - y_{im}$, and $\bar{e}_i = M^{-1} \sum_m e_{im}$. The usual mean-square identity separates candidate-shared and candidate-relative prediction error:

$$\underbrace{\frac{1}{M}\sum_m e_{im}^2}_{\text{Brier error}} = \underbrace{\bar{e}_i^2}_{\text{shared error}} + \underbrace{\frac{1}{M}\sum_m (e_{im} - \bar{e}_i)^2}_{\text{relative error}}. \tag{4}$$

For these VLM heads, the reduction is concentrated in shared error, while relative error slightly increases. The shared term measures the squared error in mean predicted success across candidates. Its reduction has adjusted support under both resampling schemes and persists without calibration (Appendix D.1). Thus, the improved probability estimates primarily describe how the candidate group performs. Whether the router identifies the right member of that group remains a separate empirical question.

From shortlist coverage to final choice. The quality decline from joint inputs is concentrated where correct candidates are already available. Most occurs where each condition's own Top-4 shortlist contains a correct answer. Aggregate coverage barely changes, yet the joint router less often selects a correct candidate. The lists can contain different models, so this partition localizes the loss without fixing membership. The pattern appears in primary means across both encoders and head objectives, including on all-model-success queries. Task composition matters: MathVerse drives much of the VLM-BCE decline, while HallusionBench and BLINK improve. LatentRouter improves on seven tasks, and excluding MathVerse changes its mean visual effect to slightly positive. Task-resampled intervals include zero. The evidence identifies a decision-level failure mode, not a uniform effect of visual inputs.

Testing relative information against task priors. We next target candidate-relative information directly. Centered MSE improves mean quality over ordinary MSE, although task-resampled intervals do not establish the gains. Explicit task-profile residuals improve matched direct heads but remain below their priors, even with supplied task identity. Shrinkage and conditional use reduce harmful residuals without an adjusted positive gain over priors or activation-matched controls (Appendix D.9). A cluster-profile policy fitted without benchmark labels also approaches the predicted-task control, but its clusters remain strongly source-associated (Appendix D.13). These preferences are strong in-distribution controls. Under source holdout, coarse-domain and cluster profiles do not establish consistent gains over source-benchmark profiles (Appendix D.14). Visual information should be judged by additional model-selection gains over a control matched to the available task coverage and supervision.

(3) How should a limited label budget be allocated to adapt to a new task?

When a task is absent from source training, its candidate preferences must be estimated from limited target outcomes. All policies here know target-task identity, and source training excludes every image group appearing in that task. Equal-source-task macro quality screens candidates; disjoint target-training outcomes calibrate their preferences. These target-specific pools differ from the global pools above. We ask how to allocate additional supervision across candidates and queries.

We compare K candidates on n queries with $Kn = 128$ and source-prior weight one third. Allocating 32 queries to each of four candidates exceeds the 16- and 32-candidate allocations by 2.55 and 3.95 percentage points (Table 3). Both differences have positive adjusted intervals under fixed-task and task-resampled evaluation in all five splits. The advantage over eight candidates is not consistently supported under task resampling. This supports the joint breadth–depth allocation, rather than isolating sampling depth.

Table 3: Allocating 128 candidate-outcome labels per target task. Pools are source-ranked and target-specific; prior weight $\alpha/(n + \alpha) = 1/3$ is fixed. Quality is task-macro strict accuracy, averaged over ten acquisitions and the primary splits.

Candidates K	Queries n	Prior α	Quality (%) ↑
4	32	16	**82.15**
8	16	8	81.00
16	8	4	79.60
32	4	2	78.20

Crossed comparisons show that screening and calibration act together. Without shrinkage, more queries improve the four-candidate pool, while a smaller source-ranked pool helps at a fixed query count. A stronger prior nearly removes the pool-size gap when 32 queries are available. Random four-model pools perform worse, although candidate quality also changes. At the same 128-label budget, source-informed successive halving remains below the four-candidate allocation under both uncertainty tracks. The main allocation gains also persist on all-model-success queries. These results favor source-informed allocation without establishing an intrinsically optimal pool size. Source supervision is additional, and equal label counts do not equalize monetary costs. Appendix D.10 gives the sequential controls and the budget- and failure-dependent cost extension.

7 Discussion and Conclusion

MMR-Bench V2 separates the answers a model pool contains, task-level preferences, and the router's final choice. Compact task-conditioned pools retain quality within the prespecified tolerance at medium and high validation budgets on the fixed suite. Matched frozen-feature heads and LatentRouter expose a task-dependent prediction–selection mismatch: images can improve outcome estimates while worsening model choice. The value of visual information therefore requires decision-level evaluation beyond probability error.

Task preferences provide a strong control when the evaluated benchmarks are represented in training. Source-held-out comparisons do not establish a consistent advantage from coarse-domain profiles over source-benchmark profiles. With known target-task identity, however, source-screened four-candidate evaluation outperforms the tested 16- and 32-candidate uniform and successive-halving allocations at equal label budgets and matched prior weight. This supports learning useful preferences through source screening and target calibration, then measuring the incremental selection value of query-specific information.

The evidence is conditional on a repeatedly explored outcome matrix, heterogeneous collection settings, and the evaluated policy families. Overlapping splits and source holdouts do not supply independent confirmation on new data. Task composition affects the visual and compact-pool results, while reference costs and equal label counts do not establish equal deployment or adaptation spending. The resulting evaluation criterion is concrete: richer information should earn credit for better model choices beyond an appropriately supervised task-level control.

AI USE STATEMENT

Generative AI tools were used for language polishing and assistance with figure preparation. All technical content, experiments, and conclusions were developed and verified by the authors.

ETHICS STATEMENT

The benchmark aggregates existing evaluation questions and recorded model outputs. Source datasets retain their own terms; no blanket license is asserted for upstream questions or images, and the release does not redistribute image binaries. The current license inventory leaves redistribution terms unresolved for the individual sources. Reference costs are estimates for recorded outputs and must not be represented as actual customer bills. Execution failures and evaluator uncertainty are retained because suppressing them would distort conclusions about model reliability and routing benefit.

REPRODUCIBILITY STATEMENT

Section 3 defines routing decisions and information tracks; Section 4 specifies the frozen resource, outcomes, reference costs, and evaluation protocol. Appendix A lists task coverage and the candidate inventory; Appendix B details grouping, partitions, head configurations, and policy constraints. Appendices C and D provide per-task results and additional diagnostics. This submission includes the manuscript and its appendices, without code or dataset files.

REFERENCES

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

Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Jiaqi Wang, Yu Qiao, Dahua Lin, et al. Are we on the right way for evaluating large vision-language models? *Advances in Neural Information Processing Systems*, 37:27056–27087, 2024.

Teng-Ruei Chen. How much of the routing gap is real? decomposing the router-to-oracle gap into reproducible specialist advantage and single-draw label noise. *arXiv preprint arXiv:2607.03436*, 2026.

Xianfu Cheng, Wei Zhang, Shiwei Zhang, Jian Yang, Xiangyuan Guan, Xianjie Wu, Xiang Li, Ge Zhang, Jiaheng Liu, Yuying Mai, et al. Simplevqa: Multimodal factuality evaluation for multimodal large language models, 2025. *URL https://arxiv. org/abs/2502.13059*, 2025.

Xueqi Cheng and Yushun Dong. Latentrouter: Can we choose the right multimodal model before seeing its answer? *arXiv preprint arXiv:2605.11301*, 2026.

Tao Feng, Yanzhen Shen, and Jiaxuan You. Graphrouter: A graph-based router for llm selections. In *International Conference on Learning Representations*, volume 2025, pp. 26186–26203, 2025.

Tao Feng, Fangxu Yu, Haozhen Zhang, Zhongjie Dai, Liangqi Yuan, Zijie Lei, Weizhi Zhang, Kunlun Zhu, Haodong Yue, Keyang Xuan, et al. Llmrouter: Unified infrastructure for developing, evaluating, and deploying llm routers. *arXiv preprint arXiv:2608.06867*, 2026.

Xingyu Fu, Yushi Hu, Bangzheng Li, Yu Feng, Haoyu Wang, Xudong Lin, Dan Roth, Noah A Smith, Wei-Chiu Ma, and Ranjay Krishna. Blink: Multimodal large language models can see but not perceive. In *European Conference on Computer Vision*, pp. 148–166. Springer, 2024.

Tianrui Guan, Fuxiao Liu, Xiyang Wu, Ruiqi Xian, Zongxia Li, Xiaoyu Liu, Xijun Wang, Lichang Chen, Furong Huang, Yaser Yacoob, et al. Hallusionbench: an advanced diagnostic suite for entangled language hallucination and visual illusion in large vision-language models. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14375–14385. IEEE, 2024.

Qitian Jason Hu, Jacob Bieker, Xiuyu Li, Nan Jiang, Benjamin Keigwin, Gaurav Ranganath, Kurt Keutzer, and Shriyash Kaustubh Upadhyay. Routerbench: A benchmark for multi-llm routing system. *arXiv preprint arXiv:2403.12031*, 2024.

Zhehao Huang, Baijiong Lin, Jingyuan Zhang, Jingying Wang, Yuhang Liu, Ning Lu, Tao Li, and Xiaolin Huang. Vl-routerbench: A benchmark for vision-language model routing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9512–9523, 2026.

Zhongzhan Huang, Guoming Ling, Yupei Lin, Yandong Chen, Shanshan Zhong, Hefeng Wu, and Liang Lin. Routereval: A comprehensive benchmark for routing llms to explore model-level scaling up in llms. *arXiv preprint arXiv:2503.10657*, 2025.

Kevin Jamieson and Ameet Talwalkar. Non-stochastic best arm identification and hyperparameter optimization. In *Artificial intelligence and statistics*, pp. 240–248. PMLR, 2016.

Dongfu Jiang, Xiang Ren, and Bill Yuchen Lin. Llm-blender: Ensembling large language models with pairwise ranking and generative fusion. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14165–14178, 2023.

Guannan Lai and Han-Jia Ye. When routing collapses: On the degenerate convergence of llm routers. *arXiv preprint arXiv:2602.03478*, 2026.

Guannan Lai, Haoran Hu, Hao-Xuan Ma, and Han-Jia Ye. Orbit: An optimal routing and budgeted inference toolbox. *Frontiers of Computer Science*, 2026.

Bohao Li, Yuying Ge, Yi Chen, Yixiao Ge, Ruimao Zhang, and Ying Shan. Seed-bench-2-plus: Benchmarking multimodal large language models with text-rich visual comprehension. *arXiv preprint arXiv:2404.16790*, 2024.

Hao Li, Yiqun Zhang, Zhaoyan Guo, Chenxu Wang, Shengji Tang, Qiaosheng Zhang, Yang Chen, Biqing Qi, Peng Ye, Lei Bai, et al. Llmrouterbench: A massive benchmark and unified framework for llm routing. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 37733–37754, 2026.

Yuliang Liu, Zhang Li, Mingxin Huang, Biao Yang, Wenwen Yu, Chunyuan Li, Xu-Cheng Yin, Cheng-Lin Liu, Lianwen Jin, and Xiang Bai. Ocrbench: on the hidden mystery of ocr in large multimodal models. *Science China Information Sciences*, 67(12):220102, 2024.

Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. In *International Conference on Learning Representations*, volume 2024, pp. 23439–23554, 2024.

Haoxuan Ma, Guannan Lai, and Han-Jia Ye. Mmr-bench: A comprehensive benchmark for multimodal llm routing. *arXiv preprint arXiv:2601.17814*, 2026.

Ahmed Masry, Mohammed Saidul Islam, Mahir Ahmed, Aayush Bajaj, Firoz Kabir, Aaryaman Kartha, Md Tahmid Rahman Laskar, Mizanur Rahman, Shadikur Rahman, Mehrad Shahmohammadi, et al. Chartqapro: A more diverse and challenging benchmark for chart question answering. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 19123–19151, 2025.

Minesh Mathew, Viraj Bagal, Rubèn Tito, Dimosthenis Karatzas, Ernest Valveny, and CV Jawahar. Infographicvqa. In *2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 2582–2591. IEEE, 2022.

Isaac Ong, Amjad Almahairi, Vincent Wu, Wei-Lin Chiang, Tianhao Wu, Joseph E Gonzalez, Mohammed Kadous, and Ion Stoica. Routellm: Learning to route llms from preference data. In *International Conference on Learning Representations*, volume 2025, pp. 34433–34448, 2025.

Runqi Qiao, Qiuna Tan, Guanting Dong, MinhuiWu MinhuiWu, Chong Sun, Xiaoshuai Song, Jiapeng Wang, Zhuoma Gongque, Shanglin Lei, Yifan Zhang, et al. We-math: Does your large multimodal model achieve human-like mathematical reasoning? In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 20023–20070, 2025.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.

Ibne Farabi Shihab, Abu Sa-Adat Mohamed Moon-Im Al Ahsan, and Md Najmus Swaqeeb. Opportunity is not realizability: Selection-valid diagnostics for multi-llm routing. *arXiv preprint arXiv:2608.08265*, 2026.

Ke Wang, Junting Pan, Weikang Shi, Zimu Lu, Houxing Ren, Aojun Zhou, Mingjie Zhan, and Hongsheng Li. Measuring multimodal mathematical reasoning with math-vision dataset. *Advances in Neural Information Processing Systems*, 37:95095–95169, 2024a.

Zirui Wang, Mengzhou Xia, Luxi He, Howard Chen, Yitao Liu, Richard Zhu, Kaiqu Liang, Xindi Wu, Haotian Liu, Sadhika Malladi, et al. Charxiv: Charting gaps in realistic chart understanding in multimodal llms. *Advances in Neural Information Processing Systems*, 37:113569–113697, 2024b.

Penghao Wu and Saining Xie. V*: Guided visual search as a core mechanism in multimodal llms. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13084–13094. IEEE, 2024.

Yijia Xiao, Edward Sun, Tianyu Liu, and Wei Wang. Logicvista: Multimodal llm logical reasoning benchmark in visual contexts. *arXiv preprint arXiv:2407.04973*, 2024.

Xiang Yue, Tianyu Zheng, Yuansheng Ni, Yubo Wang, Kai Zhang, Shengbang Tong, Yuxuan Sun, Botao Yu, Ge Zhang, Huan Sun, et al. Mmmu-pro: A more robust multi-discipline multimodal understanding benchmark. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15134–15186, 2025.

Renrui Zhang, Dongzhi Jiang, Yichi Zhang, Haokun Lin, Ziyu Guo, Pengshuo Qiu, Aojun Zhou, Pan Lu, Kai-Wei Chang, Yu Qiao, et al. Mathverse: Does your multi-modal llm truly see the diagrams in visual math problems? In *European Conference on Computer Vision*, pp. 169–186. Springer, 2024.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, et al. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*, 2025a.

Yiqun Zhang, Hao Li, Jianhao Chen, Hangfan Zhang, Peng Ye, Lei Bai, and Shuyue Hu. 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, 2025b.

Richard Zhuang, Tianhao Wu, Zhaojin Wen, Andrew Li, Jiantao Jiao, and Kannan Ramchandran. Embedllm: Learning compact representations of large language models. In *International Conference on Learning Representations*, volume 2025, pp. 76913–76926, 2025.

APPENDIX OVERVIEW

The appendix is organized into four parts:

- **Appendix A: Benchmark Details** (p. 13). Documents the task and model inventories, domain assignments, outcome semantics, execution-failure handling, and reference-price construction.
- **Appendix B: Implementation and Statistical Protocol** (p. 15). Describes image-grouped partitions, router implementations and controlled prediction heads, uncertainty estimation, and cost-constrained policy selection.
- **Appendix C: Full Initial Baseline Results** (p. 16). Reports the initial baseline comparison, matched-feature controls, and per-task results for representative policies, and specifies the information available to the reference policies.
- **Appendix D: Additional Diagnostic Evidence** (p. 19). Examines visual prediction and selection, repeatability and scoring, transfer and label budgets, model-pool composition, task-prior controls, cost feasibility, sequential target-label allocation, global pool compression, LatentRouter, benchmark-label-free cluster profiles, and cross-source preference transfer.

A BENCHMARK DETAILS

Table 4 lists all task configurations used in the frozen release. The seven inherited tasks contain 10,370 queries; the eleven additions contain 15,134. The source benchmark names denote existing evaluation resources, not newly authored questions. Comparisons use the local configuration and counts below, which can differ from the complete upstream dataset.

For domain reporting, we follow the five broad domains in Figure 2: *Visual Understanding* contains MMStar, BLINK, and VStarBench; *Text-rich Understanding* contains OCRBench, InfoVQA, ChartQAPro, CharXiv, and SEEDBench2+; *Knowledge and Reasoning* contains MathVista, MathVision, MathVerse, WeMath, LogicVista, and MMMU-Pro; *Spatial and Embodied Reasoning* contains RealWorldQA and ERQA; and *Factuality and Hallucination* contains SimpleVQA and HallusionBench. Each domain score averages its constituent task scores equally. Overall task-macro quality retains equal weights over all 18 tasks, equivalent to weighting the five domain means by their task counts (3, 5, 6, 2, and 2). These reporting domains are separate from the eight category tags retained in the dataset metadata.

A.1 OUTCOME SEMANTICS AND PROVENANCE

Query records retain questions, visual input references, benchmark identities, and source identifiers. Model-specific records retain predictions, native scores, execution status, output-token counts, and reference costs. Available outcomes require successful execution and a finite native score. Strict correctness uses full-credit Boolean labels; fractional native scores do not count as strictly correct. The matrix contains 15,006 partial-credit cells. Thus, the strict measure is not a 0.5 threshold applied to arbitrary native metrics. Native-score sensitivity is evaluated separately, because partial credit from different tasks need not have the same interpretation.

The matrix includes 4,015 explicitly unavailable executions, so complete query–model coverage does not imply a successful response in every cell. The primary analysis counts failed executions as incorrect. The separate complete-case population contains 21,910 queries for which all 44 candidates return successfully. This restriction changes the evaluated query population as well as removing execution failures.

The collection retains task prompts, model identifiers, benchmark configurations, reference-price metadata, and outcome records. Historical per-row decoding and serving revisions are not uniformly recoverable from these records. In particular, the presence of a 4,096-token cap in current launch configurations does not certify the entire collected matrix as a uniform-cap comparison. The repeated-generation study below has its own explicitly controlled cap. The release preserves failure states instead of replacing them with hypothetical successful answers.

Table 4: Task configurations and full-release query counts. Inherited refers to the seven suites used by the original MMR-Bench. All listed configurations are included in the routing evaluation.

Benchmark configuration	Queries	Coverage
MMStar (Chen et al., 2024)	1,500	Inherited
RealWorldQA	765	Inherited
SEEDBench2+ (Li et al., 2024)	2,277	Inherited
OCRBench (Liu et al., 2024)	1,000	Inherited
MathVista (MINI) (Lu et al., 2024)	1,000	Inherited
MathVision (Wang et al., 2024a)	3,040	Inherited
MathVerse (MINI vision-only) (Zhang et al., 2024)	788	Inherited
SimpleVQA (Cheng et al., 2025)	2,025	Added
InfoVQA (val) (Mathew et al., 2022)	2,801	Added
ChartQAPro (Masry et al., 2025)	1,948	Added
CharXiv (reasoning val) (Wang et al., 2024b)	1,000	Added
LogicVista (Xiao et al., 2024)	447	Added
WeMath (Qiao et al., 2025)	1,740	Added
MMMU-Pro (10-choice) (Yue et al., 2025)	1,730	Added
BLINK (Fu et al., 2024)	1,901	Added
ERQA	400	Added
VStarBench (Wu & Xie, 2024)	191	Added
HallusionBench (Guan et al., 2024)	951	Added
Total	25,504	18 tasks

Table 5: Recorded candidate identifiers, grouped by the normalized families used in the experiments. IDs are preserved independently of reference-price mappings.

Family	Candidate identifiers
Claude	`claude-opus-4-7`
Gemini	`GeminiFlash2-5`; `GeminiPro2-5`
Gemma	`Gemma3-4B`; `Gemma4-12B-it`
GLM	`GLM-4.1V-9B-Thinking`
GPT	`gpt-5-2025-08-07`; `gpt-5.5-0424-global`
InternVL	`InternVL3-14B-Instruct`; `InternVL3-2B-Instruct`; `InternVL3-78B`; `InternVL3-8B-Instruct`; `InternVL3_5-14B-Instruct`; `InternVL3_5-2B-Instruct`; `InternVL3_5-30B-A3B`; `InternVL3_5-4B-Instruct`; `InternVL3_5-8B-Instruct`; `InternVL3_5-GPT-OSS-20B-A4B-Preview`
Keye	`Keye-VL-2.0-30B-A3B`
MiMo	`MiMo-VL-7B-RL`
MiniCPM	`MiniCPM-V-4.6`
Ovis	`Ovis2-16B`; `Ovis2-1B`; `Ovis2-2B`; `Ovis2-34B`; `Ovis2-4B`; `Ovis2-8B`; `Ovis2.5-2B`; `Ovis2.5-9B`; `Ovis2.6-30B-A3B`
Qwen	`Qwen2.5-VL-32B-Instruct`; `Qwen2.5-VL-3B-Instruct`; `Qwen2.5-VL-72B-Instruct`; `Qwen2.5-VL-7B-Instruct`; `Qwen3-VL-2B-Instruct`; `Qwen3-VL-30B-A3B-Instruct`; `Qwen3-VL-32B-Instruct`; `Qwen3-VL-4B-Instruct`; `Qwen3-VL-8B-Instruct`; `Qwen3.5-27B`; `Qwen3.5-9B`; `Qwen3.6-27B`; `Qwen3.6-35B-A3B`; `qwen3.8-max-preview`

The original MMR-Bench addresses modality use, cost–accuracy tradeoffs, and transfer (Ma et al., 2026). The present resource extends its task coverage from seven to eighteen and its recorded candidate pool to forty-four, with execution-aware outcomes and reference-price provenance supporting the diagnostic studies. The additional suites increase evaluated coverage; their incremental value for selecting future methods is tested rather than presumed (Appendix D).

A.2 Reference-price construction

The reference-price snapshot is dated September 15, 2026 and contains 12 official, three hosted, and 29 estimated rates. Estimates use model-size and provider information without correctness labels. Leave-family-out calibration yields a median multiplicative error of approximately $2.73\times$, so estimated rates should be interpreted as pricing proxies rather than measured serving prices. We retain rate categories and sensitivity ranges alongside the reference costs. Output-token counts are recorded or retokenized; input and image usage, unrecorded reasoning, router encoding, network overhead, and latency are excluded.

B Implementation and Statistical Protocol

B.1 Image grouping and partitions

Grouping is based on shared decoded RGB image content. Queries with multiple images connect their image components, so reuse across tasks cannot place a shared image in both fit and test sets. Exact prompt content is the fallback for queries without images. The audited data contain 21,111 distinct image contents and 18,901 connected query groups; the largest group has 35 queries.

Ten benchmark-stratified group folds allocate the first two folds to training, the next to validation, and the remaining seven to test, corresponding to approximately 20%, 10%, and 70% of queries. Primary summaries average seeds 42–44; sensitivity analyses use 45–46. These splits share queries and are not independent datasets. Primary partition sizes are:

Split	Training	Validation	Test	Neural fit	Inner tuning
42	5,146	2,543	17,815	3,399	1,747
43	5,049	2,544	17,911	3,365	1,684
44	5,036	2,566	17,902	3,317	1,719

In matched neural-head studies, neural fit and inner tuning partition the original training set, separating early stopping from final model selection. The original validation set selects among the early-stopped models. Task profiles and learned residual directions use their designated fit data; the fixed representation projections are label-independent.

B.2 Router implementations and controlled heads

The initial adapter uses the registered ORBIT training configurations with the benchmark's standardized outcomes and evaluation interface (Lai et al., 2026). Twenty non-oracle methods and one oracle entry are evaluated. Most receive shared CLIP joint features (Radford et al., 2021). Exceptions include Eagle's MiniLM text features, raw-text BERT for the corresponding RouteLLM implementation, and a local Qwen3-1.7B labeler for NIRT. MIRT and NIRT receive model-name descriptors. These choices define the initial comparison. A separate matched study adapts the released LatentRouter implementation to the same outcome matrix and grouped splits (Appendix D.12); its different representations and training protocol are reported separately.

The baseline groups refer to the implemented routing mechanism. RouteLLM-BERT updates both its pretrained text encoder and classification head. EmbedLLM learns query projections and candidate embeddings over fixed input features; GraphRouter uses the adapter's query–model bipartite graph. The RM variants train prediction heads on fixed embeddings. NIRT uses its language-model labeler during training, rather than querying that model for every routing decision. Thus, a language-model name or a pretrained feature extractor alone does not place a router in the SLM-based group.

The frozen-parameter pool analyses use kNN with 30 neighbors or ridge with regularization 10, alongside specified neural controls. Representation controls use CLIP text and image features, or the long-text representation combining Qwen3-Embedding text (Zhang et al., 2025a) with CLIP image features. These are separate from the Qwen2.5-VL-7B text-only and image–text representation study (Bai et al., 2025). The latter uses frozen last-token representations, fixed label-independent Gaussian projections to 512 dimensions, and a $512 \to 256 \to 256 \to 44$ head with 208,428 param-

eters. Visual comparisons match encoder family and prediction protocol. Scores from separately fitted stages do not measure single-factor effects.

The original representation study searches learning rates $3 \cdot 10^{-4}$, 10^{-3}, and $3 \cdot 10^{-3}$, with at most 100 epochs and early-stopping patience 15. Its Brier diagnostics use a positive global affine logit transform followed by a sigmoid, fitted on inner-tuning data. Candidate ranking and routing retain raw scores: calibration is diagnostic and does not determine the stored choices. The subsequent BCE, MSE, and centered-MSE comparison trains new matched heads with learning rates $3 \cdot 10^{-4}$ and 10^{-3}, at most 60 epochs, and patience 10. Shared initialization and batch-order prefixes control training comparisons. These new fits explain why their BCE numbers differ from the original representation study.

Task-prior experiments append 18 predicted task weights or known-task indicators, giving $530 \rightarrow 256 \rightarrow 256 \rightarrow 44$ heads with 213,036 parameters. Direct and residual heads use the same inputs and centered-MSE objective. Fit-set true-task profiles define training residual targets; inference adds back the profile available under the declared information track. Query-only task weights are clipped nonnegative and normalized, rather than asserted to be calibrated posterior probabilities. Post-training residual scaling reuses the original validation data previously used for learning-rate selection.

B.3 Uncertainty and selection

Image-group bootstrap draws apply shared weights to paired policies and across task boundaries. Fixed-task intervals describe the evaluated suite, while task-resampled intervals additionally vary task composition. Later studies use within-stage comparison-family corrections, with 4,000 draws in the prediction/objective and task-structure rounds, and 20,000 in the final policy and residual-use round. These corrections cover the stated comparison families, not the entire history of dataset exploration. Split-wise intervals remain separate; neither their endpoints nor their confidence levels are averaged.

Model pools, fit hyperparameters, cost penalties, and gates are selected using the training or validation partitions assigned to each study. Some later analyses reuse previously explored predictions and validation sets. Frozen configurations record the executed choices but do not create independent confirmation data. Bootstrap intervals condition on those choices, stored outcomes, and fitted predictors, without incorporating all retraining, generation, or judge uncertainty. Policy-point counts share data and are descriptive, not independent replication counts.

B.4 Cost constraints and mixtures

A fixed mixture maximizes validation task-macro quality over query-independent model probabilities, subject to a query-micro cost cap equal to the corresponding router's observed validation spend. Its held-out score and cost are exact weighted expectations over recorded single-model actions. This makes it a stronger control than a single inexpensive model without invoking answer ensembling.

For the factorial and reserve studies, the validation partition is split into image-disjoint cost-calibration and policy-selection subsets. Quality heads, model library, and calibration data are held fixed across factorial cells. The small policy bank varies mean-cost penalties; the expanded bank also varies tail penalties. The optimizer mixes these policies and fixed-model actions. The mean limit is B and the excess limit is $0.1B$ at threshold $10B$. Randomized excess is computed over queries and actual selected actions, not by first averaging each query's cost and then clipping. A reserve m tightens the selection limits to $(1-m)B$ and $(1-m)0.1B$ without changing the original test criteria.

C Full Initial Baseline Results

Tables 6 and 7 report all 37 implementation–configuration combinations in the initial comparison: 20 non-oracle ORBIT implementations, one oracle, 12 matched-feature configurations, and four simple or known-task references. They are not 37 independent algorithms. Means summarize the three overlapping primary splits in Appendix B. Learned policies maximize validation task-macro strict quality within their policy family, breaking ties by lower validation task-macro cost; fixed

Table 6: All 21 registered ORBIT entries under the initial protocol, with implementation details in Appendix B. Oracle selects by true query-level quality and breaks ties by recorded cost; its cost is a hindsight quantity. Standard deviations, partial-credit quality, and complete-case metrics are retained in the experiment records.

	Quality (%) ↑		Cost (μUSD) ↓	
Registered method	Macro	Micro	Macro	Micro
ML-based				
RMInterval	81.56	82.07	818.1	1096.2
RMSoftmax	81.29	81.91	808.2	1078.9
knn	81.15	81.93	839.0	1105.3
OmniRouter	80.45	80.98	971.2	1232.6
AvengersPro	79.71	79.71	838.2	1080.7
EquiRouter	78.89	79.40	788.5	1029.3
Avengers	78.40	78.31	783.3	1046.0
EmbedLLM	78.20	78.82	726.8	946.3
mlp	78.15	77.51	654.4	789.9
GraphRouter	77.59	76.41	696.5	800.2
NIRT	77.02	77.32	1093.8	1385.8
RouteLLM_MF	76.71	77.75	805.4	1099.6
MIRT	76.65	77.52	1006.6	1301.3
svm	75.81	76.23	738.6	935.6
RouteLLM_SWRanking	75.73	77.00	778.1	1072.2
RMClassification	73.79	74.29	2219.8	2867.8
Eagle	70.67	70.94	543.7	741.0
RouterDC	69.80	68.88	382.1	449.9
HybridLLM	68.64	69.06	2200.9	2945.9
SLM-based				
RouteLLM_BERT	75.67	77.04	782.7	1076.0
References				
oracle	97.42	96.86	29.1	44.2

references are evaluated directly. Table 8 gives representative policies on all 18 tasks. These are held-out routing scores rather than candidate-model leaderboard averages.

Known-task fixed receives task identity; its training-best candidate can differ from the global candidate and can score lower on a particular test task. Oracle selects the highest-quality recorded outcome for each test query, breaking ties by the lowest recorded cost. If all candidates are incorrect, it selects the cheapest incorrect outcome. Its cost is therefore a hindsight quantity, not a realizable routing bill. Neither reference is a query-only learned method.

In the matched-feature controls, text, image, and joint denote CLIP representations with the same predictor family. Shuffled retains the text and permutes image features within each benchmark and partition before forming the joint representation. The registered ORBIT kNN and the separately configured matched-feature kNN have distinct training settings and are reported separately.

In a separate follow-up query-only comparison, predicting task identity yields 82.35% with CLIP text and 82.55% with joint features, versus 81.41% and 81.54% for the corresponding ridge policies. These fixed-quality comparisons provide additional evidence for strong task-level controls; they are not alternative scores for the initial validation-selected ridge rows.

Table 7: All simple references and matched-feature controls in the initial comparison. Quality is strict correctness (%); costs are output reference costs per query (μUSD). Learned policies use validation-quality selection. Shuffled permutes images within each task and partition while retaining text. Known-task fixed receives the true task identity.

	Quality (%) ↑		Cost (μUSD) ↓	
Policy	Macro	Micro	Macro	Micro
Simple				
Uniform random	59.76	58.61	421.6	502.5
Cheapest	42.10	40.10	2.6	2.6
Global fixed	78.14	77.02	551.1	605.4
ML-based: matched features				
kNN, text	79.22	79.46	718.5	852.9
kNN, image	79.05	79.06	965.9	1219.7
kNN, joint	79.88	80.28	848.9	1061.4
kNN, shuffled	79.45	79.77	825.0	997.7
Ridge, text	80.97	81.32	834.9	1039.3
Ridge, image	79.74	79.76	1058.1	1344.7
Ridge, joint	81.02	81.35	899.5	1158.1
Ridge, shuffled	80.71	80.92	776.2	962.1
MLP, text	80.09	80.08	798.9	1053.0
MLP, image	77.94	77.82	742.5	954.2
MLP, joint	78.35	78.63	730.9	935.7
MLP, shuffled	78.54	79.04	781.9	1049.4
References				
Known-task fixed	82.83	82.92	827.0	1139.2

Table 8: All eighteen held-out task results under the initial protocol, averaged descriptively over splits 42–44. Strict quality in percent. Fixed is the global training-best model; Task uses true task identity; RT/RJ are matched ridge text/joint. Task and oracle have extra information. These are not full-release candidate leaderboard scores.

Task	Fixed	Task	RMInterval	RT	RJ	Oracle
MMStar	83.19	83.19	80.94	81.19	80.33	98.12
RealWorldQA	87.25	87.25	84.27	81.47	81.90	99.50
SEEDBench2+	72.63	75.31	73.98	74.56	74.62	93.91
OCRBench	91.08	90.32	90.56	91.31	91.50	99.00
MathVista (MINI)	91.16	91.16	90.21	90.54	90.02	98.19
MathVision	85.73	89.88	89.52	88.14	88.92	98.09
MathVerse (MINI vision-only)	86.03	86.03	87.36	86.03	85.97	96.67
SimpleVQA	74.80	74.80	74.26	75.18	74.40	91.63
InfoVQA (val)	93.08	93.08	93.08	90.47	90.38	99.24
ChartQAPro	68.90	68.90	67.51	66.83	66.76	88.52
CharXiv (reasoning val)	83.28	83.04	83.18	83.37	82.99	98.71
LogicVista	77.91	77.91	78.34	77.58	77.58	99.57
WeMath	74.65	90.60	90.60	90.24	90.35	99.81
MMMU-Pro (10-choice)	49.94	77.97	76.49	74.26	75.14	95.98
BLINK	54.89	79.89	79.41	78.31	77.66	99.92
ERQA	67.17	67.17	64.60	63.67	65.53	97.43
VStarBench	95.04	95.04	87.10	88.83	87.60	99.75
HallusionBench	69.82	79.44	76.61	75.55	76.65	99.55
Macro	78.14	82.83	81.56	80.97	81.02	97.42
Micro	77.02	82.92	82.07	81.32	81.35	96.86

D Additional Diagnostic Evidence

D.1 Calibration and decision-local visual diagnostics

Table 9: Prediction and selection for the same VLM encoder and BCE-head protocol. Quality and coverage are task-macro percentages; selection loss is Top-4 coverage minus Top-1 quality, computed before rounding.

Router input	Brier ↓	Top-4 coverage ↑	Top-1 quality ↑	Selection loss (pp) ↓
Text	0.20177	91.12	81.30	9.83
Text + image	0.19269	91.13	80.08	11.05

Table 10 repeats the within-encoder VLM-BCE decomposition with raw probabilities. Both probability tracks retain the same frozen heads and raw-score choices. The lower total error with joint inputs, concentrated in the shared component, therefore does not depend on post-training calibration. The raw-probability sensitivity splits have the same qualitative pattern. Average pairwise ranking error also falls slightly, from 0.286617 to 0.286462, despite worse Top-1 quality. An average over candidate pairs is not a sufficient measure of the decisions at the top of the ranking.

Table 10: Joint minus text prediction error for frozen VLM-BCE heads. Descriptive means over the three primary, overlapping splits; these are not pooled confidence intervals. Top-1 quality decreases by 1.2178 pp in both probability tracks because selection uses the same raw scores.

Probability track	Total Brier	Shared error	Relative error
Calibrated	-0.009076	-0.009150	$+0.000074$
Raw BCE	-0.008331	-0.008459	$+0.000128$

For the paired shortlist analysis, let C_T and C_J indicate a correct candidate in the text and joint Top-4 lists, and let Y_T, Y_J indicate their selected candidates' strict correctness. The four combinations of (C_T, C_J) partition each query population. We evaluate $\mathbb{E}[(Y_J - Y_T)\mathbf{1}\{(C_T, C_J) = r\}]$ for each region r, with task-macro weights computed over the full population, not normalized inside the region. Their sum is exactly the overall quality change. Likewise, helpful minus harmful switches equals the quality change, and gained minus lost coverage equals the coverage change. Raw-score choices and the original aggregate switch/coverage statistics are replayed before computing this new partition. No model is retrained or selected using this diagnostic.

Table 11: Decision-local visual changes: primary descriptive means in percentage points. "Both" is the contribution from queries where each modality's own Top-4 list contains at least one correct candidate. Complete cases restrict to queries with all 44 executions successful and use that population's own denominators.

		All queries		Complete cases	
Encoder	Objective	Total change	Both	Total change	Both
CLIP	BCE	-1.577	-1.560	-1.536	-1.506
CLIP	Pairwise	-0.649	-0.601	-0.567	-0.547
VLM	BCE	-1.218	-1.181	-2.041	-1.996
VLM	Pairwise	-0.749	-0.608	-1.150	-1.034

For VLM-BCE, the both-covered region contains helpful and harmful switches contributing 3.322 and 4.503 percentage points of the full population, respectively. Its contribution has fixed-task, family-adjusted intervals of $[-2.645, -0.985]$, $[-2.264, -0.423]$, and $[-1.229, 0.563]$ pp in the three primary splits; all three task-resampled intervals include zero. These use 4,000 paired image-group draws and a 16-contrast family (four encoder/objective conditions times four diagnostics) per split and population. Sensitivity splits 45–46 average -1.495 pp total change and -1.425 pp both-covered contribution for VLM-BCE. MathVerse's primary quality change is -16.7434 pp: with equal weight over eighteen tasks, it contributes -0.9302 pp, approximately 76% of the net macro

decline. All tasks are retained; HallusionBench and BLINK improve by 4.8343 and 1.4354 pp. Individual task results, split-wise intervals, and exact arithmetic checks were recorded for this analysis. This is a post-hoc localization on an already explored test matrix, not an independent confirmation or a statement about all visual routers.

The VLM-BCE primary coverage partition has 1.052% newly covered queries, 1.041% no-longer-covered queries, and 90.082% with correct candidates in both shortlists. The both-covered contribution is -1.181 pp of the -1.218 pp total quality change. Text-only coverage contributes -0.180 pp and joint-only coverage $+0.143$ pp, using the full task-macro population as denominator. Six of the eighteen tasks improve. The shared Brier-error reduction has adjusted support in all three primary splits under both resampling schemes; CLIP-BCE additionally exhibits a small relative-error reduction.

D.2 Repetition and scoring

The repeatability study evaluates four local candidates, Gemma3-4B, InternVL3-8B-Instruct, Ovis2-16B, and Qwen2.5-VL-32B-Instruct, on 20 fixed queries from each of 18 tasks. Three greedy generations per query–model pair with a 4,096-token response cap produce 4,320 answers. The final uniform-v4 protocol evaluates them twice, producing 8,640 scoring records using objective rules where applicable and the same Qwen3.8-27B judge where required. This is the final repaired evaluation protocol, not an unchanged preregistered scoring procedure.

The training-selected fixed candidate scores 60.19%, the same-repeat oracle 78.06%, and selection based on the other two repeats' labels 77.13%. The cross-repeat loss is 0.926 pp, with a paired 95% interval of $[-1.577, -0.357]$ pp and a six-comparison-adjusted interval of $[-1.859, -0.231]$ pp for cross-repeat minus oracle. For 76.67% of queries, at least one candidate is judged correct on all three repeats; that candidate can differ across queries. Thus, most four-candidate coverage persists under this greedy repeat setting. Selection with additional labels from the same query is not a deployable query-only router; frozen query-only routers score 58.24–60.46%.

The two evaluations agree on the same recorded answers. This measures repeatability of this evaluation procedure, not independence of its errors or correctness against human judgment. Four models, 360 queries, and three greedy repeats do not establish the noise contribution for the full 44-model matrix or stochastic decoding (Chen, 2026).

D.3 Transfer, label budgets, and incremental task coverage

Task-held-out evaluation pairs target test queries while matching training size. Joint task-prediction quality falls from 82.54% with target-task coverage to 77.43% without it; joint ridge falls from 81.02% to 75.81%. The learned acquisition gate falls from 79.77% to 72.97%. These are within-resource task holdouts, not a new external dataset.

Candidate integration has a related limitation. With 16 observed labels for each held-out target, direct joint-kNN integration averages losses of 0.79 pp for single-model targets and 1.10 pp for family targets across five splits. At 1,024 labels, average losses shrink to 0.31 and 0.29 pp. More evidence can reduce integration harm without ensuring that a new candidate's potential coverage becomes useful.

The old-seven and new-eleven task rankings have mean Spearman correlation 0.914 across five splits, despite sizable changes for individual candidates. Under equal label budgets, adding new-task descriptors does not improve held-out-family prediction: MSE changes from 0.01031 to 0.01093. A separate matched-budget method-selection study obtains 78.27% on new tasks using the old panel, 78.86% using seven new tasks, and 78.34% using a mixed seven-task panel. The mixed-versus-old effect changes sign across the three primary splits. These tests support broader evaluation coverage without establishing stable incremental method-selection value.

More labels generally help the frozen full-pool routers: increasing training rows from 5% to 100% changes joint kNN from 77.50% to 79.86%, ridge from 79.25% to 81.01%, and MLP from 76.94% to 78.48%. Validation labels are additional to these training fractions. At a 25% label budget, simple uncertainty-based acquisition is 2.54–9.57 pp below random complete-row acquisition across five splits. These full-pool query-level training layouts do not establish a reliable label-saving strategy.

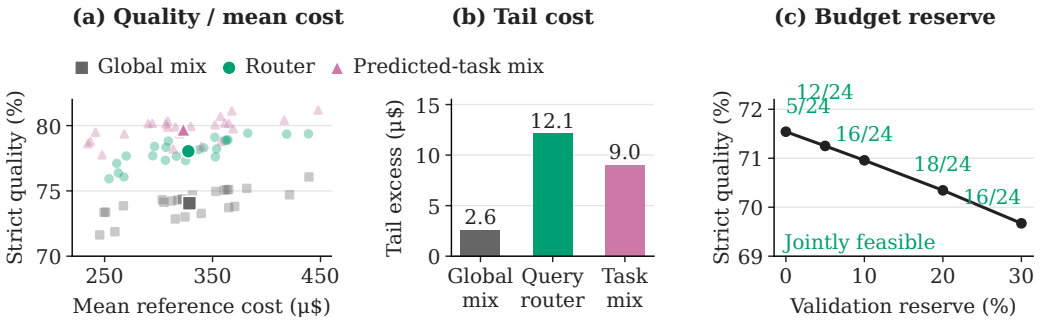

Figure 5: Quality and reference-cost boundaries. Left/middle: medium-budget routers, global mixtures, and predicted-task mixtures share each router's validation spend cap; quality is task-macro, cost and excess query-micro. Right: the tight-budget reserve path shows mean quality and joint feasibility out of 24 related policy points. These points share data and are not independent replications. Costs exclude input usage and router overhead.

The separate task-level adaptation study in Appendix D.10 instead allocates a fixed incremental target-label budget among source-screened candidates.

D.4 Low-rank residuals and paid probes

Projecting a frozen ridge residual onto four preselected fit-derived directions improves CLIP text and joint quality by about 1.01 and 1.09 pp over full residuals. Both remain below the zero-residual task prior. Four directions explain approximately 22.31% of the training residual spectrum, so this result is not evidence that nearly all candidate capability is four-dimensional. For VLM joint features, full residual prediction reduces MSE while quality falls from 81.818% to 81.441%. Training spectrum, predictable residuals, and useful decisions remain distinct objects.

A paid-probe study adds features from cached InternVL3-2B answers. It reaches 81.53% at reference cost \$0.001012 per query, compared with 82.55% and \$0.000847 for the query-only task-prediction control. Costs account for unique actual calls, avoiding duplicate charges when the probe is selected. This representation of paid information does not demonstrate a quality–cost advantage or an end-to-end latency improvement.

D.5 Cost-aware comparisons and feasibility

Cost-aware comparisons distinguish three questions: whether a policy improves quality under a validation spending cap, whether that cap transfers to test, and whether mean expenditure controls expensive individual actions. For a uniformly sampled query and an action drawn from policy π, let Z_π be its recorded reference cost. Tail expenditure is measured by excess cost, $\mathbb{E}[(Z_\pi - \tau)_+]$, above threshold τ, where $(z)_+ = \max(z, 0)$.

Figure 5 summarizes the main comparisons. At medium budgets, routing exceeds global mixtures but trails query-predicted task mixtures under shared validation spending caps. The latter improve mean quality and cost in the primary summary without attaining the global mixture's lower excess. Tail constraints and budget reserves reduce expenditure or improve held-out feasibility, but neither by itself establishes a quality gain over a strong control. The next two subsections report the complete comparisons; target-task adaptation under cost constraints is treated separately in Appendix D.10.

D.6 Task-conditioned cost controls

The conditional-mixture study freezes all 24 original routers per split: CLIP or long-text representations, text or joint inputs, kNN or ridge, and three nominal budgets. It reuses each representation's task predictions from the original train-only ridge classifier with regularization 10. Both the original quality/cost predictors and these task classifiers use the same full training partition, unlike the later

neural-head studies. Global mixtures replay the original spend-matched control. Predicted-task mixtures use query-only task predictions; known-task mixtures additionally receive actual benchmark identity.

For stratum g and candidate m, the quality coefficient is $\sum_{i:g(i)=g} y_{im}/(Tn_{t_i})$, where n_{t_i} is the validation count of query i's true task and $T = 18$. Its cost coefficient is $\sum_{i:g(i)=g} c_{im}/N_{\text{val}}$. A linear program optimizes one candidate-probability simplex per stratum under one overall cost ceiling, equal to the corresponding router's observed validation micro spend. Empty validation strata fall back to that router's global mixture. No test outcomes or test stratum counts enter selection. Expected quality, cost, and excess integrate single-model actions exactly. Conditional policy classes contain the global mixture, but their larger validation-fitting capacity does not guarantee a test improvement.

New paired intervals use 4,000 image-group draws with fixed-task and task-resampled tracks. Within each budget, eight routers, three metrics, and two new controls form a 48-comparison family. The global-mixture reproduction preserves its original 24-comparison family. This is a post-hoc extension of repeatedly explored data, not independent confirmation or causal attribution of all routing gains to task recognition.

Table 12: Cost-control progression on the three primary splits, averaged over eight matched router configurations per budget. Quality is task-macro; reference cost and excess are query-micro in microdollars. Nominal groups identify the original routers, while each comparison's actual cap is that router's validation spend. Known-task mixtures receive extra information.

Nominal budget	Policy	Quality (%)	Cost (μ\$)	Excess (μ\$)
\$0.0001	Router	70.724	69.571	34.626
	Global mixture	69.054	71.032	34.582
	Predicted-task mixture	73.617	73.294	28.124
	Known-task mixture	75.071	69.875	21.560
\$0.0005	Router	78.037	327.582	12.123
	Global mixture	74.055	328.587	2.598
	Predicted-task mixture	79.701	322.959	9.022
	Known-task mixture	80.080	324.090	1.894
\$0.001	Router	80.240	1183.501	105.690
	Global mixture	78.142	605.395	0.000
	Predicted-task mixture	81.066	1098.596	56.123
	Known-task mixture	81.635	1107.977	49.282

Router-minus-predicted-task quality averages -2.893, -1.664, and -0.826 pp across the tight, medium, and wide primary groups. This is not a uniform ranking: three wide-budget fixed-task adjusted intervals favor the router. Sensitivity splits 45–46 retain a negative medium-group quality difference (78.243% versus 78.986%), with six of sixteen adjusted fixed-task intervals favoring the predicted-task mixture and none favoring the router; all medium-group task-resampled intervals include zero. Unlike the primary medium group, its predicted-task mixture costs more on test (329.036 versus 322.096 microdollars). Thus, neither exact test-spend equality nor simultaneous quality/cost dominance holds in every setting. All 18 task results and every configuration were retained in the experiment records.

In the primary medium-budget group, the router gains 3.982 pp over the global mixture. All 24 adjusted fixed-task quality intervals and 12 task-resampling intervals support this difference. The predicted-task mixture instead exceeds the router by 1.664 pp: 20 of 24 adjusted fixed-task intervals favor the mixture and none favors the router, while all corresponding task-resampled intervals contain zero. Excess uses a shared threshold of ten times each router's validation micro-average spend. This pair-specific threshold differs from the nominal-budget threshold in the factorial study below.

D.7 Mean budgets, factorial controls, and reserve sensitivity

In the initial penalty-selection study at the tightest budget of \$0.0001, all 24 primary strategy points satisfy both test macro and micro mean-cost limits. Nevertheless, the 99th-percentile cost is 13.06–

33.20 times the query mean, and the most expensive 5% of queries contribute 78.13–87.29% of reference spending. No adjusted fixed-task quality interval supports a gain over the corresponding fixed model. Task-macro and query-micro cost constraints can diverge at wider budgets.

Table 13: Policy-bank and constraint controls at $B = \$0.0001$. Quality is task-macro, cost and excess query-micro. Joint feasibility counts the 24 primary strategy points satisfying both original test limits, even for mean-only selection. The candidate pool remains fixed at 44.

Policy bank	Constraints	Quality (%)	Cost (μ\$)	Excess (μ\$)	Joint feasible
Small	Mean	70.748	104.163	32.524	0/24
Expanded	Mean	71.571	101.713	19.579	0/24
Small	Mean + excess	71.401	103.125	8.862	8/24
Expanded	Mean + excess	71.542	102.527	9.358	5/24

Reserve (%)	Quality (%)	Cost (μUSD)	Excess (μUSD)	Joint feasible
0	71.542	102.527	9.358	5/24
5	71.252	97.726	8.792	12/24
10	70.958	92.916	8.244	16/24
20	70.345	83.543	7.961	18/24
30	69.671	74.192	8.303	16/24

The preselected primary reserve is 10%. The full path above keeps test limits unchanged at $B = \$0.0001$ and excess limit $0.1B$. Choosing the most favorable reserve after inspecting this path would use test information. A larger reserve lowers mean spend but can change the action mix enough to worsen tail excess.

The reported factorial quality contrasts are 0.823 pp for policy-bank expansion under the mean-only constraint, 0.652 pp for adding the tail constraint in the small bank, and -0.029 pp for adding it in the expanded bank. Their interaction is -0.681 pp. None of the primary adjusted task-resampling quality contrasts excludes zero. The preselected reserve reduces router quality by 0.585 pp and micro-average cost by 9.612 microdollars per query. The fixed mixture's joint feasibility also improves, from 1/3 to 2/3 splits. These are shared-split policy evaluations, not deployment success probabilities.

At the same 10% reserve, the fixed mixture reaches 70.854% quality against the router's 70.958%. The 0.104 pp difference has no positive adjusted task-resampling support. Router excess is higher, at 8.244 versus 7.311 microdollars. The fixed-mixture comparison and the router's reserve path use the same original test mean and excess limits.

D.8 Pool composition and expansion paths

For each split and representation, the expansion study evaluates 50 random and 50 fixed-model-anchored nested sequences, with family and training-complementarity controls. At 44 candidates, the training-selected fixed model achieves 78.14% quality, the oracle 97.42%, and CLIP joint ridge 81.01%. The router recovers 14.87% of the oracle's gain over that fixed model; this is gap recovery rather than accuracy. The oracle gap consists of 11.41 pp from recovering unsuccessful executions and 7.87 pp from replacing valid but wrong answers, totaling 19.28 pp. Sparse failures can contribute substantially when concentrated in the selected reference model and highly weighted tasks.

From 4 to 44 candidates in random sequences, oracle growth exceeds joint-router growth by 2.85 pp for CLIP kNN and 2.12 pp for CLIP ridge. The corresponding long-text differences are 2.33 and 2.32 pp. Split-wise adjusted fixed-task intervals support positive differences across all five splits. Changes in the joint-minus-text gain are $+0.02$ pp for CLIP kNN and -0.05 pp for CLIP ridge, compared with $+0.56$ and $+0.13$ pp for their long-text counterparts.

The composition study evaluates 4,946 unique pools per split and matches pool size, best and mean capability, reference cost, and failure rate. Residual diversity is one minus the mean pairwise correlation of doubly centered training outcomes, estimated on all-model-success training rows. Higher residual complementarity is associated with joint-kNN gains of 2.07, 1.68, and 0.51 pp at sizes

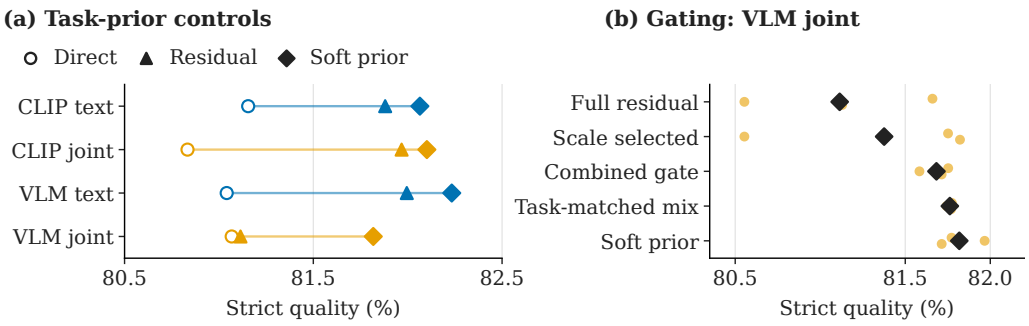

Figure 6: Task priors as a decision-level control. Left: matched direct and residual heads versus their query-predicted task prior. Right: post-training residual controls for the VLM-joint MLP; diamonds denote primary means and colored points the individual splits. Improving a direct head or repairing a harmful residual does not establish a gain over the prior.

2, 4, and 8, respectively. Predicting held-out joint-kNN pool utility from size alone gives RMSE 0.0490, averaged across five splits. Adding capability, cost, and failure descriptors reduces RMSE to 0.01268, and residual diversity further reduces it to 0.01235. The main matched gains average the three primary splits. These observational controls do not fully separate composition effects.

On the 21,910-query subset where all 44 candidates return successfully, matched small-pool gains remain 2.42 pp at size 2 and 1.91 pp at size 4. This restriction changes the query population, so its score difference from the full dataset does not isolate a failure effect. Along expansion paths starting from two training-selected anchors, oracle coverage rises from 87.15% to 97.42% while all eight frozen routers lose quality. CLIP joint ridge falls from 82.03% to 81.01%, with cumulative beneficial switches contributing 2.15 pp and harmful switches removing 3.17 pp. Prior shrinkage recovers quality to 82.00%, below the earlier 82.55% task-prediction control.

D.9 Relative objectives, task priors, and conditional routing

Task-level visual gains depend on the representation and predictor. In the CLIP-kNN comparison, joint inputs gain 4.16 pp on SimpleVQA and 3.06 pp on HallusionBench, while losing 2.12 pp on MathVerse and 1.90 pp on InfoVQA. In the BCE representation study, joint VLM features outperform joint CLIP features, but this cross-encoder comparison also changes text processing and pretraining and does not isolate image information.

The matched objective study trains new heads with BCE, ordinary MSE, and centered MSE. Centering removes each query's shared component during training rather than transforming already-trained scores. Relative to ordinary MSE, centered MSE improves primary mean quality by 0.701 pp for CLIP text and 1.310 pp for CLIP joint, and by 0.828 pp for both VLM inputs. CLIP joint quality is 80.443% against 82.032% for its task-prediction control; VLM joint quality is 81.037% against 81.762%. Both encoders retain negative mean joint-minus-text gains. All 30 primary adjusted task-resampling intervals contain zero.

In the matched direct-versus-residual study, CLIP residual heads improve on their direct heads by 0.726 pp for text and 1.134 pp for joint inputs, but remain below their soft task priors by 0.184 and 0.134 pp, respectively. VLM residual heads improve on direct heads by 0.954 pp for text and 0.047 pp for joint inputs, while remaining below their priors by 0.238 and 0.703 pp (Figure 6). The four residual means in the known-task track also remain below the fixed task prior. Both heads receive the same input and centered-MSE objective. Query-only inference uses task weights predicted from the input; the known-task track supplies actual task identity.

The conditional-residual study holds fitted heads fixed. Validation selects $\alpha \in \{0, 0.1, 0.25, 0.5, 1\}$ in a prior-plus-α-residual score. Gates use task-prediction entropy and the prior's top-two margin, with global and predicted-task-stratified mixtures matched to validation activation rates. For the VLM joint MLP, quality is 81.818% for the prior, 81.114% for the full residual, 81.376% with selected amplitude, and 81.683% with the combined gate. The task-stratified mixture obtains 81.762%.

No adjusted query-only comparison establishes positive gate gains over the prior or either matched control in primary or sensitivity splits. Intervals containing zero do not establish equivalence. These MLP results are distinct from the ridge residuals in the low-rank study.

The separate CLIP-kNN image-acquisition gate uses images for 51.05% of queries and scores 80.26%, against 79.86% for always-joint routing. Its non-inferiority test with a 0.3 pp margin passes the primary splits but fails both sensitivity splits. Unlike image acquisition, post-training residual gating acts on features that have already been computed and provides no evidence of saved encoding cost.

D.10 Target-task adaptation with limited supervision

Source and target information. For each target task, source training removes every image group occurring anywhere in that task, including groups shared with other benchmarks. The target test queries retain the original partitions in Appendix B. Calibration draws from the target's training partition: sample distinct image groups uniformly without replacement, then one query uniformly from each selected group. Ten acquisitions per task and split provide nested query sets. All policies know which target task is being evaluated. Source-fixed chooses the best source candidate, while the frozen joint ridge router predicts query-level candidate quality from source training. Both use zero new target labels. Calibration instead acquires labeled candidate outcomes and selects one fixed model for each target task and acquisition.

We rank candidates by source-task macro strict quality, breaking ties by source-task macro reference cost and then model index. For a top-K pool, the calibrated estimate combines n acquired target labels with the source profile:

$$\widehat{q}_{t,m} = \frac{\sum_{i \in \mathcal{A}_t} y_{i,m} + \alpha s_m^{(-t)}}{n + \alpha}, \qquad |\mathcal{A}_t| = n, \tag{5}$$

where $s_m^{(-t)}$ averages correctness across source tasks equally and $y_{i,m}$ is strict correctness. The chosen model maximizes this estimate with the same source-cost tie-break. Our initial calibration sweep uses $K \in \{4, 8, 44\}$, $n \in \{4, 8, 16, 32\}$, and $\alpha \in \{0, 16, 64\}$. Four candidates and 32 queries with $\alpha = 16$ reach 82.15% task-macro quality, against 76.52% for source-fixed and 75.81% for source-trained joint ridge. This establishes adaptation with additional target supervision, not an equal-supervision comparison.

Separating pool size and sample count. Without shrinkage ($\alpha = 0$), increasing n from 4 to 32 at $K = 4$ raises primary quality from 80.236% to 82.137%. Holding $n = 32$, restricting the source-ranked pool from 44 to four candidates raises quality from 79.869% to 82.137%. With $\alpha = 64$ and $n = 32$, the four-, eight-, and 44-candidate pools nearly tie at 82.231%, 82.211%, and 82.211%. These descriptive comparisons share source rankings and nested acquisitions but vary total label counts. They support roles for both screening and estimation, without identifying estimation variance as the cause. The random-pool control also changes candidate quality. A fixed policy selected using all target-training queries and all 44 outcomes reaches 82.83%. This extra-supervision reference is not an upper bound, and non-inferiority against it was not tested.

Allocating an equal label budget. The matched study fixes $Kn = 128$ candidate-outcome evaluations per target task, or 2,304 across the 18 tasks per acquisition. It reuses the same acquisitions and source rankings. Table 14 reports every calibration configuration, separating the zero-target-label references. Fixed $\alpha = 16$ changes the prior weight $\alpha/(n + \alpha)$ from $1/3$ at $n = 32$ to $4/5$ at $n = 4$. We therefore include both the full $\alpha = 0$ grid and $\alpha = n/2$, which keeps prior weight at $1/3$. Random-pool controls sample four of the 44 candidates uniformly for each task, split, and acquisition; both prior settings share that pool and query set. Source screening uses pre-existing labeled source outcomes, whose acquisition is additional to the matched target-label budget.

Uncertainty and interpretation. The primary four-candidate policy is compared with eight pre-specified controls: three broader pools at $\alpha = 16$, the same pool at $\alpha = 0$, the random pool at $\alpha = 16$, and three broader pools at equal prior weight. Within each split, 10,000 paired global image-group bootstrap draws are crossed with resampling of the ten acquisition vectors. Random-pool and query-acquisition variation are resampled together. A second track also resamples the

Table 14: Complete equal-label calibration comparison. Each calibration row uses 128 target labels per task. Quality is task-macro strict accuracy (%), averaged over ten acquisitions and the indicated overlapping splits. The two references use zero new target labels. Primary splits are 42–44 and sensitivity splits are 45–46.

Pool or reference	K	n	α	Primary quality	Sensitivity quality
Source-fixed	44	0	–	76.52	76.95
Source-trained joint ridge	44	0	–	75.81	75.78
Source-ranked	4	32	0	82.14	82.44
Source-ranked	4	32	16	**82.15**	**82.46**
Source-ranked	8	16	0	80.12	80.48
Source-ranked	8	16	16	81.17	81.48
Source-ranked	16	8	0	75.36	76.04
Source-ranked	16	8	16	80.47	80.16
Source-ranked	32	4	0	67.04	68.13
Source-ranked	32	4	16	79.42	79.51
Random	4	32	0	70.02	70.22
Random	4	32	16	70.87	71.18
Source-ranked, equal prior weight	8	16	8	81.00	81.17
Source-ranked, equal prior weight	16	8	4	79.60	79.38
Source-ranked, equal prior weight	32	4	2	78.20	78.30

18 task identities. Two-sided percentile endpoints at $0.05/16$ and $1 - 0.05/16$ apply the eight-comparison Bonferroni correction. The five splits overlap, and intervals condition on the observed training data, source rankings, and fitted source router. The studies reuse previously explored outcomes, with all tested settings retained rather than selected on test.

At equal prior weight, the four-candidate policy exceeds the 16- and 32-candidate policies by 2.55 and 3.95 pp on primary splits, and by 3.09 and 4.16 pp on sensitivity splits. Both adjusted interval tracks support these differences in every split. The four-versus-eight task-resampled comparison is positive in one of five splits. Removing the prior at four candidates changes primary quality by 0.016 pp, with no adjusted support, whereas removing it from broader, shallower allocations causes much larger declines. On the common population where all 44 outcomes are available, the four-candidate gains over equal-prior-weight 16- and 32-candidate policies remain 2.42 and 3.00 pp descriptively. These comparisons evaluate joint breadth–depth allocations under source screening. They do not isolate sampling depth or establish a universally optimal pool size. Matching labels also leaves inference costs unequal: the four-candidate policy costs 1,056.98 and 1,334.35 microdollars per query on primary and sensitivity splits, respectively.

Sequential allocation at the same label budget. We additionally compare source-informed successive halving with the same four static allocations and ten paired acquisitions. Starting with the top 16 source-ranked candidates, the three rounds allocate $16 \times 4 + 8 \times 4 + 4 \times 8 = 128$ distinct query–model labels. Starting with 32 candidates instead allocates $32 \times 2 + 16 \times 2 + 8 \times 4 = 128$. Each round observes the next unused queries in the shared acquisition plan for surviving candidates, then ranks them by Equation 5 with $\alpha = n/2$, where n is each survivor's cumulative label count. This preserves prior weight $1/3$. Eliminated candidates receive no later labels. Both schedules reuse exactly the source profiles and image-group exclusions of the static comparison, with no target-test access during selection.

The 16- and 32-candidate schedules reach 80.44% and 78.94% primary quality, compared with 79.60% and 78.20% for their same-starting-pool static allocations. Those improvements lack adjusted task-resampling support. Both schedules remain below the four-candidate static allocation's 82.15%, with negative fixed-task and task-resampled adjusted intervals in all five splits. The primary gaps remain 1.88 and 3.21 pp on all-model-success queries. Four planned contrasts per split and scoring/population track use 10,000 image-group–acquisition bootstrap draws. These fixed-budget schedules test adaptive elimination without establishing best-arm optimality. Their comparison with four candidates jointly changes initial screening breadth, final estimation depth, and allocation strategy.

Table 15: Cost-constrained adaptation without a reserve, averaged over the three primary splits and ten acquisitions. Quality is task-macro strict accuracy. Budget and query-micro expected inference cost are in microdollars (10^{-6} USD). "Source" uses the full 44-model source-only LP; "Calibrated" uses the cost-aware four-model pool and $\alpha = 16$.

Budget	Source quality (%)	Calibrated quality (%)	Source cost	Calibrated cost
100	70.88	69.42	112.13	71.87
500	76.74	77.66	533.35	315.15
1000	76.93	79.53	596.96	467.23

Adding an inference-cost constraint. The cost-constrained extension constructs a different target-specific four-model pool using source data alone. It includes the cheapest source model and the highest-quality source model, then greedily adds candidates maximizing average source LP quality over six fixed caps. These are B and $0.8B$ for $B \in \{100, 500, 1000\}$ microdollars per query. Source quality remains task-macro, but source cost is query-micro. The selected pool is shared across budgets and acquisitions. The same 32 acquired queries update both quality and cost using Equation 5, with $\alpha = 16$ or 0. Each task then solves a simplex LP maximizing estimated quality subject to its estimated mean-cost cap. Inference samples one model from this mixture. Source-only LPs over the same four candidates and all 44 candidates provide matched-cap controls with zero new target labels.

With $\alpha = 16$, all 40 medium- and high-budget comparisons against the two controls have positive fixed-task adjusted intervals, using a twelve-comparison correction per split. The same 10,000-draw crossed bootstrap is used, now with twelve contrasts. All 60 task-resampled intervals include zero. The 20 tight-budget point differences are negative, including those with the fixed 20% reserve. Thus, the gain in Table 15 is specific to the evaluated task composition and budget range. On the common all-model-success population, the primary medium/high differences against the full-pool control become $-2.425/-1.480$ pp without a reserve. Concurrent failure-rate reductions and this population dependence preclude interpreting the result as a general improvement in selecting among successful answers.

All 30 $\alpha = 16$ split/budget/reserve configurations have overall expected test cost below their cap, including each of the ten acquisitions. Per-task violations remain: at the medium budget, a 20% reserve raises the primary task/acquisition fraction within the nominal budget from 69.07% to 93.15%, while quality falls from 77.66% to 76.75%. Neither this reserve nor aggregate expected-cost feasibility guarantees every task or a finite draw of randomized actions stays within budget. Costs use query means, and tail metrics threshold each model action's cost before averaging, rather than clipping each query's mean cost. The recorded calibration-output reference cost averages $\$0.377$ across all 18 tasks on primary splits, separately from inference costs and label counts. It excludes judging, human annotation, input usage, router encoding, and historical source supervision.

D.11 Global compact-pool control

A separate compression study asks whether a single global library can retain a full-pool budget frontier. Its four-model restriction is shared across all predicted tasks, unlike the target-specific source-ranked pools above or the distinct cost-aware adaptation pools. Frozen CLIP text and joint task classifiers use training-only ridge fits with regularization 10. Validation data alone fit task-conditioned mixtures with global cardinality $K \in \{1, 2, 4, 8, 44\}$, maximizing true-task macro quality under the same query-micro budgets of 100, 500, and 1,000 microdollars. Binary model-admission variables enforce the global support constraint. The unrestricted $K = 44$ LP and a query-independent global mixture use identical caps. The resulting policies describe expected single-model calls, not answer ensembles.

The unrestricted LP assigns positive weight to 4–11 models across the 30 configurations. The four-model restriction therefore compresses an already sparse policy, rather than reducing actual deployment from 44 models to four. Only six of the 30 four-model policies satisfy their nominal test-cost cap: none at the tight budget, one at the medium budget, and five at the high budget.

The prespecified $K = 4$ versus $K = 44$ test uses a 0.5 pp non-inferiority margin, 10,000 paired image-group bootstrap draws, and a six-comparison family per split covering two representations and three budgets. Its one-sided Bonferroni lower endpoint must exceed -0.5 pp. At medium/high budgets, all 20 fixed-task comparisons pass, while 14 of 20 pass with task resampling; the tight budget passes 8 of 10 fixed-task comparisons. These counts summarize overlapping evaluations rather than independent replications. Validation feasibility does not establish a hard test-cost guarantee, and quality non-inferiority does not assert preserved oracle coverage. The global compression result and the target-label allocation result therefore concern different uses of a compact pool.

Execution and scoring sensitivity. Replaying these frozen policies on the 21,910-query all-model-success population preserves the medium/high-budget non-inferiority counts: 20 of 20 fixed-task comparisons and 14 of 20 task-resampled comparisons pass. At the tight budget, nine of ten fixed-task comparisons pass. Table 16 reports primary complete-case differences. Replacing strict correctness with native scores changes these point differences little. Native scores rescore the original policies, rather than fitting new native-utility policies, and have no assigned non-inferiority margin. The complete-case restriction changes the query population, so agreement across populations is a robustness observation rather than a causal estimate of execution-failure effects. Bootstrap families are defined separately within each population and scoring track.

Table 16: Compact-pool sensitivity on queries with all 44 executions successful. Values are primary-split mean $K = 4$ minus $K = 44$ task-macro differences in percentage points, with both policies unchanged.

Representation	Budget (μ\$)	Strict difference	Native difference
Text	100	$+0.513$	$+0.535$
Text	500	$+0.403$	$+0.406$
Text	1,000	$+0.424$	$+0.424$
Joint	100	$+0.382$	$+0.382$
Joint	500	$+0.302$	$+0.302$
Joint	1,000	$+0.750$	$+0.750$

Price perturbations with policy reselection. We further reoptimize the compact, unrestricted, and global mixtures after perturbing the 29 estimated output rates, keeping the 12 official and three hosted rates unchanged. Four scenarios multiply every estimated rate by 0.5, 2, $1/2.73$, or 2.73; five additional scenarios draw independent family-level log-uniform factors in $[1/2.73, 2.73]$. Each scenario refits the original validation LP/MILP at the same three nominal caps, then evaluates its selected weights on test. The unperturbed scenario exactly reproduces the original policies. The multiplicative range is a stress test, not a calibrated price-confidence interval.

Across the nine perturbed scenarios and five splits, all 180 medium/high-budget $K = 4$ minus $K = 44$ point differences remain above -0.5 pp, ranging from -0.163 to $+1.530$ pp. At the tight budget, 89 of 90 do so. These are point-estimate tolerance counts, not new non-inferiority tests or independent replications. Only 64 of the 270 perturbed four-model policies meet their nominal test mean-cost cap. Thus, this output-rate sensitivity supports the stability of the compact-pool quality comparison more strongly than budget feasibility. It does not evaluate input/image charges, router overhead, latency, or the rate sensitivity of every other cost experiment.

D.12 Latent communication and task-dependent visual effects

We extend the matched comparison to the released latent-communication implementation of LatentRouter (Cheng & Dong, 2026). Its shared query–model interaction network retains query tokens, seven routing capsules, two four-head communication layers, a distributional outcome head, bounded correction, and all six training-loss terms. We use the public benchmark configuration: hidden width 48, four query tokens, batch size 64, ten maximum epochs, patience three, and AdamW with learning rate 10^{-3} and weight decay 10^{-4}. Training-only capability profiles use the official 35% calibration fraction, capped at 512 queries. Those profiles can describe training benchmark slices, but query features contain neither benchmark identities nor group identifiers.

The adaptation replaces the default hashing features with the same real frozen VLM text/joint representations used in our controlled study, normalized and projected to 512 dimensions. The upstream scaler is fit on the original training partition. All 44 candidates remain eligible, failed outcomes are zero targets, and cost inputs are broadcast training-only candidate means rather than realized test costs. Quality-only routing uses zero cost penalty. The original image-grouped splits replace the upstream default split, and validation task-macro quality selects the training checkpoint. Validation, test selection, and Top-4 ranking break score ties by canonical model index. The matched soft task prior uses the same projected features and training queries, a task-balanced ridge classifier with regularization ten, and training task–model profiles. This evaluates the released architecture under our protocol, rather than reproducing the original paper's numbers or fine-tuning a VLM end to end.

Table 17: LatentRouter adaptation and its matched task prior. Primary task-macro means; quality and Top-4 coverage are percentages. Native and all-success columns rescore the same strict-trained policies. All-success restricts to queries with every candidate execution successful.

Policy	Input	Strict	Native	All-success	Brier	Top-4
LatentRouter	Text	79.91	80.19	87.34	0.20748	91.10
LatentRouter	Joint	78.53	78.82	84.37	0.19868	91.21
Task prior	Text	82.54	82.80	87.60	0.21574	91.13
Task prior	Joint	81.92	82.18	86.93	0.21473	91.25

The primary aggregate reproduces prediction–selection separation with query–model interactions (Table 17). Joint inputs reduce shared squared error from 0.07402 to 0.06464 while relative error increases from 0.13346 to 0.13404. Top-4 coverage changes little, whereas Top-1 quality decreases. Among Top-4 pairs with unequal correctness, the per-query inversion fraction increases from 0.414 to 0.421, averaged over eligible queries within tasks. This local ranking diagnostic complements the uniformly candidate-averaged error decomposition. The task prior yields higher routing quality for both modalities despite its higher Brier error.

Task composition remains decisive. Seven of eighteen tasks improve with joint inputs (Table 18). MathVerse declines by 26.08 pp, and excluding it descriptively changes the mean joint-minus-text effect from -1.373 to $+0.080$ pp. Across all eighteen leave-one-task-out summaries, effects range from -1.854 to $+0.080$ pp. All three primary fixed-task adjusted intervals are negative, but all five task-resampled intervals include zero. Sensitivity splits have opposite directions: $+0.720$ and -1.769 pp. The complete-case and native-score results retain the negative primary mean. These observations extend the architecture coverage of the failure mode while preserving its dependence on the evaluated task distribution. Intervals use 4,000 paired image-group draws and twenty comparisons per split, covering five contrasts across two populations and two scoring tracks.

D.13 Cluster preferences without benchmark-label supervision

Benchmark-prediction controls learn the identities of the 18 source suites. To test whether explicit identity supervision is necessary for fitting a competitive preference policy, we instead cluster query representations and estimate a candidate-success profile within each cluster. Clustering and profile estimation use training queries without benchmark labels. Validation task-macro quality selects the configuration. For the cluster policy, test benchmark identities are used only for reporting. This separates explicit identity supervision from query features that may themselves reveal the source distribution.

Both modalities use the cached Qwen2.5-VL-7B representations, the same label-independent 512-dimensional Gaussian projection, training-only feature standardization, and row normalization. Mini-batch k-means uses $k \in \{8, 18, 32, 64\}$, ten initializations, batch size 1,024, and at most 200 iterations. Each hard-cluster outcome profile is shrunk toward the global training-query mean with a weight equivalent to 20 observations. Inference uses either the nearest cluster or a distance-softmax mixture at temperature 0.25 or 1, scaled by the median training nearest-center squared distance. Validation selects among these twelve configurations. Matched direct ridge and predicted-task controls use the same features and uniform query weights, with regularization in $\{1, 10, 100\}$. Among these query-only controls, the predicted-task policy alone uses training benchmark labels for fitting

Table 18: Complete task-level joint-minus-text strict-quality distribution (pp), averaged over the three primary splits. The frozen BCE heads and LatentRouter have separate matched text/joint protocols. These descriptive effects do not constitute independent task-level significance tests.

Benchmark	VLM-BCE heads	LatentRouter
BLINK	$+1.44$	$+5.84$
CharXiv	-0.95	-0.76
ChartQAPro	-0.58	-0.17
ERQA	$+1.37$	-2.75
HallusionBench	$+4.83$	$+3.92$
InfoVQA	-0.10	-0.25
LogicVista	$+0.61$	$+1.06$
MMMU-Pro	-1.46	$+6.81$
MMStar	-1.68	-1.43
MathVerse	-16.74	-26.08
MathVision	-0.53	$+1.02$
MathVista	-0.28	-0.33
OCRBench	-0.62	-2.25
RealWorldQA	-2.92	-1.13
SEEDBench2+	$+0.67$	$+1.12$
SimpleVQA	-1.67	-0.82
VStarBench	-3.47	-9.18
WeMath	$+0.16$	$+0.66$
Task macro	-1.22	-1.37

its classifier and task profiles. All policies retain the original image-grouped splits and 44 eligible candidates.

Table 19: Benchmark-label-free cluster preferences and matched controls. Task-macro strict quality (%) averages the three primary splits. Known-task profiles additionally receive the true benchmark identity at inference. These controls have their own matched preprocessing and fitting weights.

Policy	Text	Joint
Direct ridge	81.60	80.76
Cluster profile	81.70	81.72
Predicted-task profile	81.97	82.29
Known-task profile	82.83	82.83

Cluster preferences retain much of the task-profile quality without explicit benchmark supervision in fitting (Table 19). The primary joint cluster policy exceeds direct ridge by 0.96 pp, but trails the predicted-task profile by 0.57 pp. The joint-minus-text cluster difference is just 0.02 pp. All corresponding task-resampled adjusted intervals include zero. On sensitivity splits, the joint cluster advantage over direct ridge disappears, averaging -0.05 pp. Moreover, cluster assignments remain strongly associated with source identity: test normalized mutual information ranges from 0.67 to 0.70 across the five splits and two modalities. Thus, explicit source labels are unnecessary for fitting a competitive in-distribution profile policy, while semantic task structure and source-specific feature cues remain unresolved. Intervals use 10,000 paired global image-group draws and eight planned comparisons per split; the original frozen matrix remains the evaluation population.

D.14 Preference transfer across benchmark sources

We test whether coarser preference groups transfer beyond an observed benchmark source. Each of the 18 benchmarks is held out in turn under all five splits, yielding 90 target folds. Every image-connected group appearing anywhere in the target benchmark is removed from both source training and source validation. The remaining source partitions retain the original split assignments. All target queries are evaluated, so the aggregated target population contains all 25,504 queries rather than the earlier 70% test population; absolute scores are not directly comparable across these settings. Both modalities use the same frozen VLM features, 512-dimensional projection, source-training

standardization, row normalization, and fitting grids as Appendix D.13. Hyperparameters maximize source-benchmark-macro validation quality on held-out queries from the observed sources, rather than through nested source holdout. All 44 candidates remain eligible.

The six policies compare different structures for transferring source outcomes. A global fixed policy selects the candidate with the highest equal-source-benchmark mean. Domain profiles instead average source benchmark profiles equally within the five domains defined in Section 4. Predicted-domain and predicted-benchmark policies mix the corresponding source profiles using query-based ridge classifiers. The known-domain reference receives the true coarse domain at inference, but no target outcomes. Cluster profiles and direct ridge retain their matched definitions from Appendix D.13; clustering uses no benchmark labels during fitting. Spatial reasoning and factuality each contain two benchmarks, leaving only one same-domain source when a target is held out. Thus, the domain comparison tests this existing coarse taxonomy with limited within-domain source coverage.

Table 20: Preference transfer to held-out benchmark sources. Strict task-macro quality (%) averages the three primary splits. Coarse domain profiles do not establish an advantage over source-benchmark profiles. The known-domain reference receives additional domain information; it and the global fixed policy do not use query representations.

Policy	Text	Joint
Global source fixed	76.64	76.64
Known-domain source profile	74.43	74.43
Predicted-domain profile	75.06	75.35
Predicted-source benchmark profile	75.80	76.34
Cluster profile	76.78	75.07
Direct ridge	76.49	74.17

Coarser grouping does not establish a stable transfer advantage (Table 20). Predicted-domain profiles trail predicted-benchmark profiles in the primary means for both modalities, whereas the cluster comparison changes direction between text and joint inputs. All task-resampled adjusted intervals for these four contrasts include zero across the five splits. Visual effects also depend on the policy: the primary joint-minus-text mean is positive for predicted-domain profiles and negative for direct ridge. The in-distribution strength of task preferences therefore does not establish that this coarse taxonomy transfers to unseen sources. Intervals use 10,000 paired global image-group draws, with fixed-task and task-resampled tracks and Bonferroni adjustment over six prespecified full-population strict-score contrasts per split. They condition on the stored outcomes and fitted policies, excluding retraining uncertainty from the highly overlapping source folds. This is a source-held-out diagnostic within the previously explored matrix, not independent confirmation. The benchmark-level domain mapping and unchanged query features also leave source-specific artifacts unresolved.