MRG-ROUTER: PRE-RENDERING ROUTING VIA CAPABILITY PROBES FOR MULTI-REFERENCE IMAGE GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Multi-reference image generation (MRIG) requires preserving visual entities across multiple exemplars while following textual instructions, with applications in personalized advertising, e-commerce display, and creative editing. However, diverse generative models exhibit query-dependent complementarity alongside differences in inference costs, meaning that statically deploying a single model is either expensive or sub-optimal in quality. To address this issue, we study pre-rendering routing, selecting the optimal generator prior to rendering to maximize capability gains without candidate rendering overhead. Distinct from existing routers relying on static model identities or coarse global rankings, we propose MRG-Router to address the multimodal evidence mismatch prior to rendering by representing generators through empirical behavior rather than identity. Specifically, MRG-Router profiles candidate generators on a compact Capability Probe Bank and uses Query-Conditioned Probe Reweighting (QPR) to align probe-level evidence with incoming multimodal queries. For systematic evaluation, we construct MR-RouteBench, a dense benchmark systematically covering twelve representative compositional scenarios with associated compute traces. Evaluations across MR-RouteBench and two external benchmarks demonstrate that under quality-driven routing, MRG-Router outperforms the single strongest fixed generator. Under practical deployment constraints, it preserves visual quality comparable to the best fixed model while reducing generation costs by up to 96.0%, achieving superior quality-cost trade-offs over competing routing baselines. Furthermore, MRG-Router onboards unseen generators without parameter updates simply by appending probe profiles, demonstrating the potential of our routing framework to scale alongside advancements in generative modeling.

1 INTRODUCTION

Multi-reference image generation (MRIG) has emerged as a foundational interface for controllable visual synthesis. Practical domains such as e-commerce product display, targeted advertising, and creative image manipulation frequently require preserving identities, garments, objects, or styles across multiple visual exemplars while adhering to an arbitrary textual instruction. Although recent unified foundation models have advanced support for joint textual and visual conditioning (Xiao et al., 2025; Pan et al., 2024; Xie et al., 2025) alongside progress in in-context visual generation (Wu et al., 2025b; Chen et al., 2025; Fu et al., 2025), their comparative behaviors and relative failure modes across diverse multi-reference tasks remain insufficiently characterized.

Existing MRIG benchmarks highlight challenging reference configurations and compositional bottlenecks (Oshima et al., 2026; Wu et al., 2026b; Wei et al., 2026), yet they predominantly evaluate generators in isolation rather than analyzing comparative behavior across identical requests. This evaluation gap obscures a critical phenomenon in practice, where relative model competency is intensely query-dependent. As shown in Figure 1, the optimal generator for human-garment synthesis often performs poorly on multi-object arrangements, demonstrating distinct specializations within the candidate pool. This performance variance reveals substantial model routing potential, as individual requests naturally favor distinct generators. Such complementarity motivates pre-rendering

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.



Figure 1: Motivation and validation of pre-rendering routing in multi-reference visual synthesis. (a) Candidate generators exhibit query-dependent strengths across diverse composition requests. (b) The proposed MRG-Router achieves superior validation quality-cost trade-offs compared to fixed models and baseline routers. (c) Under pure quality routing ($\lambda = 0$), our approach closes 33.8% of the best-fixed-to-oracle gap prior to rendering. The "Nano Banana" represents fixed official outputs from the MICo benchmark.

routing, wherein the system designates an optimal generator from a heterogeneous candidate pool given an instruction and reference images before producing any visual output. Statically deploying the single best generator leaves query-specific capability gains untapped, whereas rendering outputs from all candidates introduces prohibitive latency and computational cost. Effective routing should capitalize on model complementarity solely through pre-generation evidence.

Conventional routing frameworks fail to address this objective. Language-model routers generally encode unimodal text requests and optimize policies over static candidate sets (Ong et al., 2025). Image-generation routers center primarily on prompt routing for text-to-image models or workflow-level agent tool invocation (Xin et al., 2025; Jiang et al., 2026; Chen et al., 2026). While recent multimodal routing studies explore visual question-answering settings (Cheng & Dong, 2026), MRIG requires jointly reasoning over a textual instruction and a variable-length sequence of visual references. Because generator suitability hinges on preserving visual entities across exemplars while satisfying compositional constraints, unimodal text encodings, aggregate benchmark averages, and static model identifier embeddings lack the expressiveness required for accurate capability matching.

These challenges reveal a fundamental evidence mismatch. The definitive indicator of model suitability, namely the rendered candidate output, is inherently inaccessible prior to generation. Static benchmark scores cannot compensate for this absence because model rankings fluctuate with compositional structures and input reference types. While empirical performance records offer grounded signals, the router must identify which historical behaviors align with the target multimodal request. Furthermore, indexing candidates via learned categorical identifiers restricts generalization when modifying the candidate pool. To resolve these issues, we represent generators through empirical behavior rather than identity. MRG-Router benchmarks each model on a compact Capability Probe Bank, projecting all candidates into a shared behavioral coordinate space. For an incoming query, the router retrieves semantically proximate probes and applies Query-Conditioned Probe Reweighting (QPR) to synthesize query-specific capability predictions from historical outcomes. This formulation allows unseen generators to be onboarded simply by profiling them once on the fixed probe set and appending their behavioral vectors, requiring no retraining of the router parameters. A de-

coupled cost formulation further facilitates operational quality-cost trade-offs across deployment budgets.

To systematically evaluate pre-rendering routing, we construct MR-RouteBench, a dense benchmark comprising 9,000 queries, ten fixed candidate models, 90,000 query-candidate outcomes, and full query-level computational traces. Under pure quality-driven routing, our framework consistently outperforms the strongest single fixed generator across evaluation benchmarks and demonstrating the capability gains achievable prior to rendering. To address practical deployment constraints, balancing generation quality against computational expense reveals that the proposed MRG-Router achieves superior quality-cost trade-offs over competing routing strategies across three diverse benchmarks. Compared to the single best fixed model, our framework dramatically reduces generation expenses while preserving comparable visual quality. Furthermore, the router onboards unseen generators without parameter updates, raising routed quality by up to 4.95 points over the base candidate pool. This retraining-free adaptation successfully harnesses the performance advantages of newer architectures while preventing sub-optimal degradation, highlighting the practical deployment utility of our framework and its capability to scale alongside ongoing advancements in generative modeling. Our contributions can be summarized as follows:

- We formulate pre-rendering model routing for multi-reference image generation to exploit model complementarity without candidate rendering overhead, and construct MR-RouteBench, a dedicated benchmark systematically covering 12 compositional structures and variable reference counts.
- We propose MRG-Router, which combines behavior-grounded generator profiles with query-conditioned probe reweighting for multimodal, open-pool, and cost-aware routing.
- Extensive experiments across three benchmarks demonstrate the superiority of our method over existing routing methods. MRG-Router matches the generation quality of the strongest fixed model while reducing generation cost by up to 96.0%, and supports retraining-free onboarding of unseen generators for continuous quality improvements.

2 Related Work

2.1 Multi-Reference Image Generation

Unified image-generation systems increasingly support instruction-based editing and generation (Xiao et al., 2025; Xie et al., 2025; Wu et al., 2025a), in-context generation and subject customization (Pan et al., 2024; Wu et al., 2025b; Wang et al., 2026c), and complex multi-reference conditioning (Wu et al., 2026a; Xia et al., 2026; Xu et al., 2026b). Recent benchmarks evaluate heterogeneous references and multi-image composition (Wei et al., 2026; Oshima et al., 2026; Wu et al., 2026b), as well as fine-grained compositional failures (Wang et al., 2026b; Huang et al., 2026a; Nguyen et al., 2026). However, they primarily assess individual generators in isolation within particular task configurations rather than characterizing the comparative behavior of a heterogeneous candidate pool on identical queries. This empirical gap makes query-level model rankings, generator complementarity, and routing potential difficult to study systematically. To address this limitation, we construct MR-RouteBench, a dense benchmark systematically covering twelve representative compositional scenarios across variable reference counts. By evaluating candidate models on shared requests with factorized quality metrics and recorded compute traces, MR-RouteBench establishes a foundation for pre-rendering selection and quality-cost routing.

2.2 Routing for Image Generation

Model routing selects generators on a per-query basis rather than deploying a single static model for all requests. In language-model serving, prior works learn routing policies from reward signals, human preferences, or task-specific representations (Ding et al., 2024; Lu et al., 2024; Ong et al., 2025). For image synthesis, DiffusionGPT routes across specialized diffusion experts (Qin et al., 2024), RouteT2I balances text prompts between cloud and edge models (Xin et al., 2025), and subsequent studies incorporate cost-sensitive or agentic workflow orchestration (Li et al., 2026; Jiang et al., 2026; Chen et al., 2026). Related benchmarks further investigate model routing under heterogeneous cost-quality constraints (Hu et al., 2024; Huang et al., 2026b; Ma et al., 2026). Nevertheless, these formulations predominantly focus on unimodal text requests, discriminative understand-

ing tasks, or macro tool invocations. In contrast, the proposed MRG-Router addresses pre-rendering routing for multi-reference synthesis by jointly conditioning on the instruction and visual exemplars. Rather than assessing models after rendering or relying on text-only heuristics, the router matches the incoming multimodal query against offline empirical capability evidence prior to generating any candidate output.

2.3 Open-Set Routing and Model Profiling

Prior routing methods characterize model capabilities using learned embeddings, query-model interactions, or historical performance on representative queries (Zhuang et al., 2025; Pulishetty et al., 2025; Wang et al., 2026a). Open-set routing investigates the incorporation of unseen models (Feng et al., 2025; Yu et al., 2026; Xu et al., 2026a), while cost-aware techniques balance heterogeneous inference expenditures (Shirkavand et al., 2025). In generation systems, candidate pools evolve rapidly and deployment budgets fluctuate with hardware configurations. Existing open-set approaches typically map model profiles through static dual towers or learned model identifiers, which can struggle to capture localized compositional competencies. To resolve this limitation, our method represents generators through empirical behavior rather than identity by projecting them into a shared capability probe bank. Unlike approaches that rely on monolithic profile matching, our framework decomposes performance into base capability and probe-level residuals, employing query-conditioned probe reweighting to synthesize query-tailored predictions. This formulation allows unseen models to be onboarded without parameter updates simply by profiling them once on the fixed probe set, while enabling independent cost calibration across diverse operating points.

3 Methodology

Figure 2 gives an overview of MRG-Router, which combines offline capability profiling with online, query-conditioned access to probe-level generator behavior. Pre-rendering routing presents an evidence gap: the router must choose a generator without seeing any candidate output for the current request. A global model ranking misses query-dependent complementarity, whereas learning a free embedding for each model ties the router to a fixed candidate pool. MRG-Router instead represents every generator by measured behavior on a shared Capability Probe Bank. An incoming query retrieves semantically related probes, and a shared matcher determines which capability evidence is useful for the current decision. Separating query matching from generator profiling supports both accurate local decisions and append-only model onboarding.

3.1 Problem Formulation

We study pre-rendering model routing for multi-reference image generation. A query is $x_i = (t_i, R_i)$, where t_i is an instruction and $R_i = (r_{i1}, \ldots, r_{iK_i})$ is its ordered reference-image sequence. Given a generator pool $\mathcal{M} = \{m_1, \ldots, m_M\}$, let $Q_{i,m}$ be the quality obtained by running generator m on x_i. The oracle selects $m_i^* = \arg\max_{m \in \mathcal{M}} Q_{i,m}$, but computing this decision requires rendering with every candidate. Our router instead predicts $\widehat{Q}_{i,m}$ from the query and offline capability evidence, then selects a generator before rendering. At inference time, it never observes current-query outputs, current-query evaluator scores, category labels, or target images.

We consider a closed pool for ordinary routing and an open pool in which a new generator must be incorporated without updating router parameters. Deployment may additionally specify a cost sensitivity λ, but cost is not used to construct profiles or train QPR.

3.2 Capability-Probe Routing

Multimodal Query Encoding. Frozen text and image encoders represent the instruction and each reference. We deliberately mean-pool the reference features, concatenate them with the text feature, and standardize the fused representation. This design choice ensures permutation invariance across input exemplars, prevents the router from overfitting to spurious reference ordering, and preserves minimal routing latency while capturing the macro-level visual semantics and compositional requirements of the task. Identical encoding pipelines process incoming queries and offline capability probes. Exact encoders and normalization are specified in Appendix A.3.

Capability Probe Bank and Generator Profiles. We construct a shared Capability Probe Bank $\mathcal{P} = \{p_s\}_{s=1}^{P}$ by stratified sampling over multi-reference query structures, without using generator

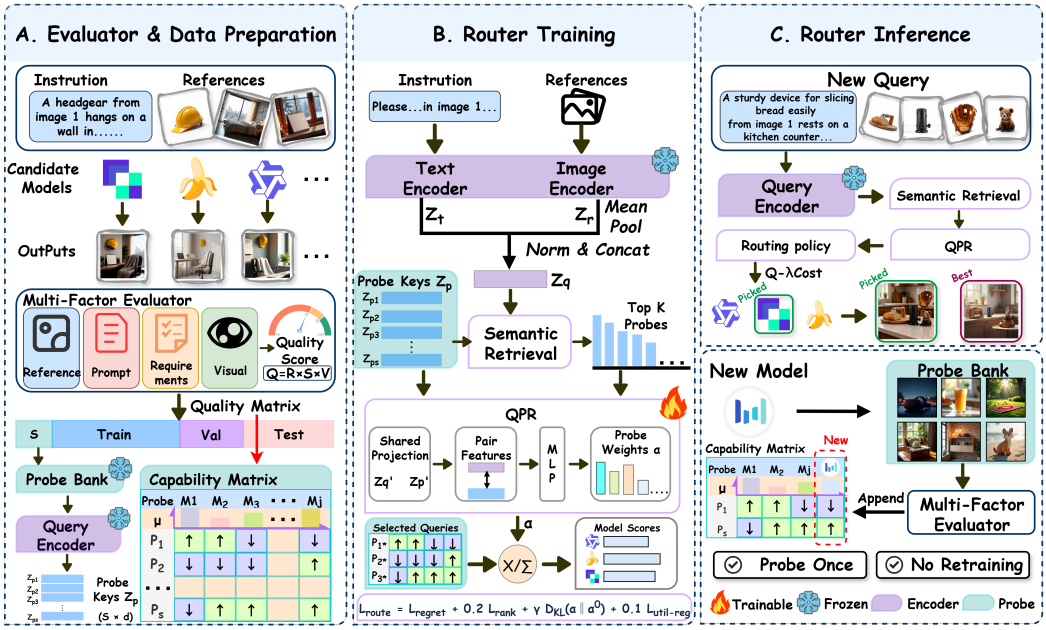

Figure 2: Overview of the MRG-Router framework. (a) Offline profiling benchmarks candidate generators on a fixed Capability Probe Bank to construct a shared behavioral coordinate space. (b) For each incoming multimodal query, the router retrieves relevant probes and utilizes Query-Conditioned Probe Reweighting (QPR) to predict generator capability. (c) Routing decisions balance predicted quality with inference costs, while new generators are integrated without parameter updates by evaluating them on the fixed probe set.

quality for probe selection. Every generator is evaluated on the same probes. We decompose its probe-level quality into a model mean μ_m, shared probe difficulty, and a centered generator–probe residual $A_{s,m}$. The resulting capability profile $c_m = (\mu_m, A_{:,m})$ therefore records measured generator behavior in a shared probe coordinate system rather than a model-ID embedding or textual description. Estimation details, including validity masking and post-stratification weighting, are provided in Appendix A.3.

Semantic Probe Retrieval and QPR. For an incoming query x_i, semantic retrieval first identifies a local probe neighborhood $\mathcal{N}(i)$. Semantic similarity determines where to seek capability evidence, but nearby queries need not induce identical generator rankings. We therefore introduce Query-Conditioned Probe Reweighting (QPR), which scores each retrieved query–probe pair and assigns normalized weights $\alpha_{i,s}$. In short, semantic retrieval determines where relevant evidence is likely to be found; QPR determines which retrieved evidence is useful for distinguishing generators on the current query. The same QPR is shared across all generators and receives neither model identity nor generator-specific logits. Exact pairwise features and network details are deferred to Appendix A.3.

Quality Prediction and Training. MRG-Router uses the query-conditioned weights to aggregate measured probe outcomes. In the centered parameterization, its decision is

$$\widehat{Q}_{i,m} = \mu_m + \sum_{s \in \mathcal{N}(i)} \alpha_{i,s} A_{s,m},$$
$$\widehat{m}_i = \arg\max_m \widehat{Q}_{i,m} = \arg\max_m \sum_{s \in \mathcal{N}(i)} \alpha_{i,s} Q^{\mathcal{P}}_{s,m}. \tag{1}$$

The shared probe-difficulty offset cancels when generators are ranked. Training follows two stages. A utility-based warm-up first encourages QPR to identify probes that provide useful generator-ranking evidence. We then optimize the routed decision using regret and ranking objectives. Probe-utility supervision and a prior regularizer stabilize the learned neighborhood weights. Full objectives and implementation details are given in Appendix A.3.

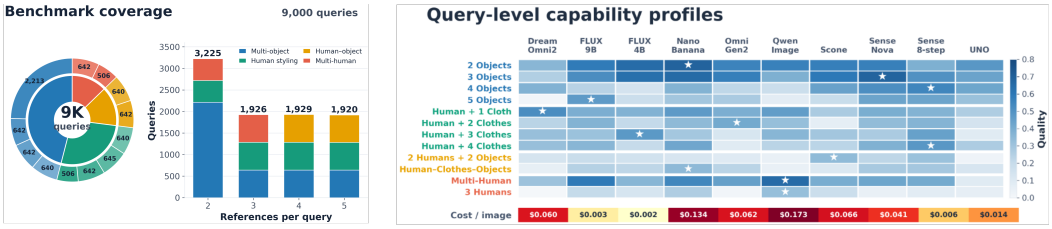

Figure 3: MR-RouteBench at a glance. **Left:** Coverage of 9,000 queries across reference counts and four compositional families derived from twelve MICo structures. **Right:** Illustrative query-level capability profiles across ten fixed candidate columns; each star marks the best generator for the displayed query. The bottom strip reports model-level cost estimates at four references. "Nano Banana" denotes the fixed MICo official-output column.

3.3 Cost-Aware and Open-Pool Routing

Cost-Aware Selection. Let $C_m(K_i)$ be the measured cost of generator m for a query with K_i references and $\widetilde{C}_m(K_i)$ its validation-scaled version. We select

$$\widehat{m}_i^{(\lambda)} = \arg\max_{m \in \mathcal{M}} \left[\widehat{Q}_{i,m} - \lambda \widetilde{C}_m(K_i) \right]. \tag{2}$$

Here $\lambda = 0$ gives quality-only routing, while larger λ favors lower generation cost. We select λ on the validation set for each deployment trade-off and keep it fixed for evaluation. The scaling and search rule are specified in Appendix A.3.

Probe-Based Model Onboarding. For an unseen generator m_{new}, we run the fixed probe bank once, compute $c_{m_{\text{new}}} = (\mu_{m_{\text{new}}}, A_{:,m_{\text{new}}})$, and append it as a new column of the Capability Profile Table. The query encoder and QPR remain unchanged ($\Delta\theta = 0$), so onboarding is append-only and requires only one generation-and-evaluation pass over the fixed probe bank. Probe-difficulty terms are estimated from the base pool and kept fixed, so existing profile columns are not recomputed.

4 MR-RouteBench and Quality Evaluation

4.1 Benchmark Construction

Existing multi-reference visual generation benchmarks primarily evaluate models in isolation within narrow task domains. Consequently, they lack dense, same-query generations across a heterogeneous candidate pool, obscuring query-level generator rankings, empirical headroom over fixed baselines, and open-pool model onboarding under standardized evidence. To bridge this empirical gap, we introduce MR-RouteBench, a dense benchmark comprising 9,000 multi-reference queries, nine frozen self-hosted candidate models, and one fixed proprietary baseline, equipped with factorized quality supervision and fine-grained computational traces. The benchmark is specifically designed for pre-rendering routing, where a selection policy must identify the winning generator solely from textual instructions, visual exemplars, and offline profiling evidence prior to candidate visual rendering. Figure 3 illustrates the workload coverage of MR-RouteBench and highlights the query-dependent capability profiles that justify pre-rendering selection.

We establish the benchmark across four stages. First, source queries are curated from the MICo corpus (Wei et al., 2026) prior to any model execution, followed by a systematic coverage audit spanning twelve representative compositional structures with reference counts ranging from two to five. The resulting query distribution reflects practical workload frequencies rather than being biased by model outcomes or automated evaluator preferences. Second, we standardize and freeze nine self-hosted candidate models alongside their optimal sampling configurations, while incorporating the fixed official output column from MICo as an external commercial baseline. Third, every query is densely evaluated across all ten candidate columns, constructing a dense matrix of 90,000 query-candidate generation outcomes. Finally, a frozen factorized evaluator scores all synthesized outcomes, while automated system profiling captures query-level execution latencies for self-hosted models, supplemented by official pricing rates for the commercial column. By preserving continuous margins across multidimensional metrics alongside granular hardware costs rather

Table 1: Human alignment of within-query rankings over 512 queries and ten generators. Kendall's τ_b is the primary metric.

Evaluator	Kendall $\tau_b \uparrow$	Spearman $\rho \uparrow$	Top-1 Agree. $\uparrow$	Human Regret $\downarrow$
Traditional Composite	0.274	0.371	27.1%	0.78
MICo-original	0.472	0.612	51.8%	0.44
MICo w/o SR	0.386	0.515	42.6%	0.61
MultiBanana-Qwen	0.431	0.568	47.7%	0.52
MR-RouteBench Evaluator (Ours)	**0.528**	**0.667**	**58.4%**	**0.34**

than recording discrete winner labels, MR-ROUTEBENCH establishes a reproducible foundation for fixed-model benchmarking, oracle bound characterization, and cost-aware model routing.

The distribution of visual references across queries is naturally non-uniform, containing 3,225, 1,926, 1,929, and 1,920 requests for $K \in \{2, 3, 4, 5\}$ references respectively. We therefore report micro-averaged performance metrics while preserving structural stratification metadata for fine-grained diagnostic analysis. Appendix Table 8 summarizes the operational parameters of the ten candidate models alongside two distinct models reserved exclusively for open-pool onboarding experiments. To guarantee evaluation integrity and provide a competitive industry anchor, nine candidate columns were generated via our unified execution pipeline while the NanoBanana column adopts fixed official outputs from the MICo benchmark. Comprehensive execution environments, hardware specifications, and cost accounting protocols are documented in Appendix A.6.

4.2 FACTORIZED QUALITY EVALUATION

Each outcome is annotated for Reference Fidelity R, Prompt Following P, Atomic Requirement Satisfaction A, and Visual Quality V. The frozen evaluator combines them as

$$R = 0.75R_{30\mathrm{B}} + 0.25R_{8\mathrm{B}}, \quad \widetilde{A} = 0.1 + 0.9A, \quad S = \sqrt{P\widetilde{A}}, \quad V = \frac{\mathrm{PQ}}{10}, \quad Q = RSV. \tag{3}$$

The multiplicative form penalizes failure on any critical dimension. Atomic requirements are compiled once from the query, shared across ten candidates, and cannot inspect candidate generations. We retain every component and Final-Q; judge inputs and failure handling are detailed in the Appendix.

4.3 DATA ROLES AND HUMAN ALIGNMENT

We partition the 9,000 benchmark queries into a held-out test split of 1,000 instances and a validation split of 600 instances. Among the remaining 7,400 instances, 512 queries form the shared Capability Probe Bank and 6,888 queries serve as training data for the matching network. The Probe Bank contains 128 instances for each reference count $K \in \{2, 3, 4, 5\}$ and is curated via metadata stratification over task categories, strictly independent of generator performance and evaluation scores. During evaluation, ground-truth quality labels, candidate images, and category annotations remain strictly inaccessible to the router.

We validate evaluator alignment on 512 stratified queries: three blinded raters score the ten anonymized outputs per query in randomized order, and their mean defines the within-query human ranking. Table 1 shows the strongest alignment on all four routing-relevant measures: Kendall τ_b of 0.528, Spearman ρ of 0.667, 58.4% Top-1 Agreement, and the lowest Human Regret of 0.34 on the original 1–10 scale.

5 EXPERIMENTS

5.1 EXPERIMENTAL SETUP

We compare MRG-ROUTER with fixed-generator and oracle policies, query-only predictors, two retrieval-based baselines, KNN (Shnitzer et al., 2024) and PRkNN (Huang et al., 2026b), and MRIG adaptations of ZOOTER (Lu et al., 2024), RouterDC (Chen et al., 2024), and SCOPE-Router (Yu et al., 2026), denoted ZOOTER-Adapt, RouterDC-Adapt, and SCOPE-Adapt, respectively. Main MR-RouteBench results use a fixed train/probe/validation/test allocation, whereas auxiliary analyses use separate development allocations; MICoBench is an external test set, and MultiBanana uses

Table 2: Quality, cost (USD per 1K queries), and Q–C results on three benchmarks. Bold and underlined values mark the best and second-best cost-aware learned routers.

Method	MR-RouteBench			MICoBench			MultiBanana		
	Avg.Q ↑	Cost ↓	Q–C ↑	Avg.Q ↑	Cost ↓	Q–C ↑	Avg.Q ↑	Cost ↓	Q–C ↑
Non-learned anchors									
Quality Oracle	56.61	59.98	48.91	64.64	64.19	54.66	86.05	38.75	74.33
Best Fixed	47.87	134.00	23.49	56.80	134.00	32.34	75.94	6.10	75.41
Cheapest Fixed	37.67	1.38	39.94	47.17	1.51	49.55	59.02	1.51	61.30
Quality-only routing									
MRG-Router-Q (Ours)	49.30	56.78	44.26	57.45	62.04	50.20	76.83	9.29	75.15
Query-only predictors									
Direct Predictor	**47.86**	15.97	47.97	56.33	9.14	56.84	69.79	7.42	69.45
MLP	46.29	6.52	47.68	56.01	7.97	56.73	69.38	6.59	69.33
ZOOTER-Adapt	46.29	**4.54**	47.96	56.15	6.93	57.06	73.28	6.18	73.00
Query–model matching									
KNN	46.65	6.12	48.05	55.95	**5.09**	57.25	75.55	6.44	74.94
PRkNN	46.57	6.81	47.87	**56.70**	9.48	57.11	75.69	**6.03**	75.22
RouterDC-Adapt	46.78	6.59	48.11	56.56	9.12	57.07	74.80	6.18	74.36
SCOPE-Adapt	46.67	6.07	48.09	56.45	6.55	57.42	75.79	6.04	75.30
MRG-Router (Ours)	46.93	6.03	**48.33**	56.54	5.33	**57.75**	**75.91**	6.35	**75.30**

group-aware five-fold out-of-fold evaluation. The router sees neither category labels nor current-query generations. We report Average Quality, Average Cost, and $\mathrm{QCS}_\beta = (1+\beta)qc/(\beta q + c)$ with $\beta = 0.1$, where $q, c \in [0, 100]$ are normalized quality and inverse log-cost. Learned methods use three seeds. Table 9 and the Appendix provide the complete allocations, baseline adaptations, cost measurement, API price snapshots, and benchmark-specific quality scaling.

5.2 Main Results

Table 2 reports both the quality-only and cost-aware operating points on MR-RouteBench, MICoBench, and MultiBanana. When evaluated under a pure quality objective ($\lambda = 0$), the resulting MRG-Router-Q configuration consistently achieves higher average visual quality than the strongest single fixed generator across all three benchmarks. This uniform advantage confirms the core premise of pre-rendering selection, demonstrating that heterogeneous generators exhibit genuine query-dependent complementarity that can be captured prior to visual synthesis without generating candidate outputs. Appendix Figure 5 complements these aggregate results with held-out examples: MRG-Router selects different generators across compositional structures and matches the quality oracle in three of the four displayed queries.

Under cost-aware evaluation, MRG-Router achieves a more favorable balance between generation quality and computational expenditure than existing learned routing baselines. Compared to retrieval-based adapters and direct quality predictors, our approach attains higher composite scores while operating at lower average inference expenses. This advantage stems from the capability probe formulation, which allows the router to route straightforward requests to economical candidates and deploy resource-intensive models only when complex composition requires higher capacity.

Compared to deploying the single best fixed model, MRG-Router preserves comparable visual quality while substantially lowering total generation expenses on benchmarks that incorporate costly commercial models. On workloads where candidate pools consist primarily of lightweight open-weight models with minimal cost disparities, the router automatically prioritizes quality preservation to track the performance of the strongest candidate. As illustrated in Figure 1, adjusting the cost-sensitivity parameter provides flexible control across diverse deployment budgets without requiring updates to the routing parameters.

5.3 Ablation Study and Open-Pool Extension

Table 3 presents component ablations conducted on the MR-RouteBench validation split without access to task category annotations. Examining input representations indicates that joint multimodal conditioning is essential, as excluding visual references degrades generation quality more substantially than omitting textual instructions alone. In terms of core architecture, Query-Conditioned Probe Reweighting (QPR) provides the primary performance contribution, where reverting to unweighted uniform probe aggregation yields the lowest score across all settings. Integrating all

Table 3: Component ablations of MRG-Router on the validation split. Results report the mean and standard deviation across three random seeds.

Variant	Text	Refs	QPR	Utility Pretrain	Prior Reg.	Quality ↑
Text-only	✓		✓	✓	✓	51.08±0.04
Ref-only		✓	✓	✓	✓	51.43±0.01
Uniform aggregation (w/o QPR)	✓	✓				50.97±0.00
w/o Utility Pretraining	✓	✓	✓		✓	51.61±0.05
w/o Prior Regularization	✓	✓	✓	✓		51.51±0.01
MRG-Router	✓	✓	✓	✓	✓	51.64±0.05

Table 4: Retraining-free generator onboarding under leave-one-generator-out and real-model evaluation.

New Generator(s)	Eval. N	Base Q	Onboarded Q	ΔQ ↑	Adoption (%)
Strict leave-one-generator-out					
10 held-out generators (macro avg.)	600 × 10	49.58	50.01	+0.43	9.52
Real new-generator onboarding					
Seedream-5.0	200	44.11±0.13	47.16±0.11	+3.05±0.15	54.5±1.1
Wan2.7-Image	200	44.11±0.13	49.06±0.10	+4.95±0.11	68.0±0.0
Seedream-5.0 + Wan2.7-Image	200	44.11±0.13	48.87±0.05	+4.75±0.08	75.5±0.4

proposed components yields the strongest overall performance, demonstrating that the individual modeling choices complement one another effectively. Furthermore, Appendix A.4 investigates the impact of probe bank capacity through controlled budget sweeps and resampling analysis, showing that routing performance improves steadily and gradually plateaus near 512 probes where estimation variance contracts and additional probes yield diminishing marginal returns, which leads us to adopt 512 probes as an effective operating point balancing profiling expenditure with routing accuracy.

Table 4 evaluates open-pool scalability, designed to isolate unseen-generator adaptation without altering router weights. In leave-one-generator-out validation episodes where held-out candidates are integrated purely through their offline probe bank profiles, routing quality improves consistently across all folds with zero parameter updates, verifying that the shared probe coordinate system reliably characterizes unseen model behaviors.

We further examine practical utility by onboarding two recent generators, Seedream-5.0 and Wan2.7-Image, into the deployed pool, where introducing either model independently or both concurrently delivers substantial performance improvements. Importantly, analyzing query adoption rates reveals that our method avoids trivially defaulting to newly introduced candidates. Instead, the router selectively dispatches requests where the newcomer demonstrates verified behavioral advantages on corresponding probes while continuing to allocate requests to incumbent models on tasks they handle reliably, confirming that the framework scales robustly as generative capabilities advance.

6 Conclusion

We propose MRG-Router, which matches multi-reference queries against empirical generator behaviors profiled in a shared Capability Probe Bank. Across three benchmarks, quality-driven routing consistently outperforms the strongest single fixed generator, while cost-aware selection establishes leading composite quality-cost performance among learned routers and preserves quality comparable to the best fixed model at substantially lower cost. Furthermore, appendable behavioral profiles integrate unseen generators without parameter updates, establishing pre-rendering routing as an extensible, practical complement to ongoing advances in individual generative models. Limitations and future directions are discussed in Appendix B.

AI USE STATEMENT

Generative AI tools were used for sentence-level language polishing, including improvements to grammar, clarity, readability, and presentation, and to support the retrieval and discovery of potentially relevant literature. All AI-assisted edits were reviewed by the authors, and candidate references and bibliographic metadata identified with AI assistance were checked against original papers, official proceedings, or author-maintained sources. Generative AI tools were not used for research ideation or execution, drafting substantive sections of the paper, generating synthetic datasets, or formulating or proving mathematical claims. The authors take full responsibility for the final content of this work.

ETHICS STATEMENT

This work studies routing among image-generation systems and therefore inherits risks associated with generative imagery, including identity misuse, unauthorized or misleading content, representational bias, and restrictions on the use or redistribution of reference images and model outputs. Any release or redistribution of benchmark artifacts is subject to the licenses and terms of the corresponding data and model sources. The human-alignment study was limited to blinded quality ratings of model-generated images by three raters; the paper reports only aggregate statistics and draws no conclusions about individual raters. The proposed router is intended for research evaluation, and practical deployment should incorporate appropriate consent, provenance, content-safety, and access-control safeguards.

REPRODUCIBILITY STATEMENT

The main paper and appendix document the information needed to reproduce the reported evaluation. Section 4 specifies the benchmark construction, data roles, generator operating points, evaluator aggregation, and human-alignment protocol, while Section 5 describes the experimental protocol and reported metrics. Appendix A.1, Appendix A.3, and Appendix A.6 provide the evaluator prompts and failure handling, router and baseline implementation details, training protocol and seeds, and cost assumptions, respectively. An anonymous repository containing benchmark schemas and query-level metadata, evaluation interfaces, model-runner scripts, and representative generated examples is available at `https://anonymous.4open.science/r/MRG-Router-EBE3`. Due to the scale of the full generated outputs, the repository provides representative examples rather than redistributing the complete set of generated images. Redistribution of third-party benchmark images and model assets remains subject to their original licenses.

REFERENCES

Harold Haodong Chen, Zhiyu Hou, Wen-Jie Shu, Weilin Ruan, Yingjie Xu, Litao Guo, and Ying-Cong Chen. Genrouter: Unified workflow routing for agentic image generation. *arXiv preprint arXiv:2608.16721*, 2026.

Shuhao Chen, Weisen Jiang, Baijiong Lin, James Kwok, and Yu Zhang. RouterDC: Query-based router by dual contrastive learning for assembling large language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL `https://openreview.net/forum?id=7RQvjayHrM`.

Xi Chen, Zhifei Zhang, He Zhang, Yuqian Zhou, Soo Ye Kim, Qing Liu, Yijun Li, Jianming Zhang, Nanxuan Zhao, Yilin Wang, et al. Unireal: Universal image generation and editing via learning real-world dynamics. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12501–12511. IEEE, 2025.

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

Dujian Ding, Ankur Mallick, Chi Wang, Robert Sim, Subhabrata Mukherjee, Victor Rühle, Laks Lakshmanan, and Ahmed H Awadallah. Hybrid llm: Cost-efficient and quality-aware query rout-

ing. In *International Conference on Learning Representations*, volume 2024, pp. 41348–41366, 2024.

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.

Tsu-Jui Fu, Yusu Qian, Chen Chen, Wenze Hu, Zhe Gan, and Yinfei Yang. Univg: A generalist diffusion model for unified image generation and editing. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 17160–17170. IEEE, 2025.

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. In *Agentic Markets Workshop at ICML 2024*, 2024. URL https://openreview.net/forum?id=IVXmV8Uxwh.

Wenwang Huang, Yusen Fu, Junjie Wang, Mengfei Huang, Yulin Li, Gan Liu, Jing Cai, Yancheng He, and Zhuotao Tian. Scaling multi-reference image generation with dynamic reward optimization. *arXiv preprint arXiv:2606.26947*, 2026a.

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, 2026b.

Xu Jiang, Bin Chen, Gehui Li, Yule Duan, Ronggang Wang, and Jian Zhang. Octot2i: A self-evolving agentic text-to-image router. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 31628–31638, June 2026.

Qinchan Li, Kenneth Chen, Changyue Su, Wittawat Jitkrittum, Qi Sun, and Patsorn Sangkloy. Cost-aware routing for efficient text-to-image generation. *Transactions on Machine Learning Research*, 2026. ISSN 2835-8856. URL https://openreview.net/forum?id=Jbe9AVsYS6.

Keming Lu, Hongyi Yuan, Runji Lin, Junyang Lin, Zheng Yuan, Chang Zhou, and Jingren Zhou. Routing to the expert: Efficient reward-guided ensemble of large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 1964–1974, 2024.

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

Long-Bao Nguyen, Quang-Khai Le, Tam V Nguyen, Minh-Triet Tran, and Trung-Nghia Le. Cogcanvas: A benchmark for evaluating multi-subject reference-based image generation. *arXiv preprint arXiv:2606.15867*, 2026.

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

Yuta Oshima, Daiki Miyake, Kohsei Matsutani, Yusuke Iwasawa, Masahiro Suzuki, Yutaka Matsuo, and Hiroki Furuta. Multibanana: A challenging benchmark for multi-reference text-to-image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 448–460, June 2026.

Xichen Pan, Li Dong, Shaohan Huang, Zhiliang Peng, Wenhu Chen, and Furu Wei. Kosmos-g: Generating images in context with multimodal large language models. In *International Conference on Learning Representations*, volume 2024, pp. 40959–40974, 2024.

Roshini Pulishetty, Mani Kishan Ghantasala, Keerthy Kaushik Dasoju, Niti Mangwani, Vishal Garimella, Aditya Mate, Somya Chatterjee, Yue Kang, Ehi Nosakhare, Sadid Hasan, et al. One head, many models: Cross-attention routing for cost-aware llm selection. *arXiv preprint arXiv:2509.09782*, 2025.

Jie Qin, Jie Wu, Weifeng Chen, Yuxi Ren, Huixia Li, Hefeng Wu, Xuefeng Xiao, Rui Wang, and Shilei Wen. Diffusiongpt: Llm-driven text-to-image generation system. *arXiv preprint arXiv:2401.10061*, 2024.

Reza Shirkavand, Shangqian Gao, Peiran Yu, and Heng Huang. Cost-aware contrastive routing for llms. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 155627–155659. Curran Associates, Inc., 2025. doi: 10.52202/085713-5202. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/e46eb6403af68506331f941282d838aa-Paper-Conference.pdf.

Tal Shnitzer, Anthony Ou, Mírian Silva, Kate Soule, Yuekai Sun, Justin Solomon, Neil Thompson, and Mikhail Yurochkin. Large language model routing with benchmark datasets. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=Zb0ajZ7vAt.

Chenxu Wang, Hao Li, Yiqun Zhang, Linyao Chen, Jianhao Chen, Ping Jian, Qiaosheng Zhang, and Shuyue Hu. Icl-router: In-context learned model representations for llm routing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 33413–33421, 2026a.

Haoran Wang, Chaofan Ma, Ran Yi, and Lizhuang Ma. Trace-bench: Decomposing and diagnosing multi-reference image generation. *arXiv preprint arXiv:2608.16765*, 2026b.

Yuran Wang, Bohan Zeng, Chengzhuo Tong, Wenxuan Liu, Yang Shi, Xiaochen Ma, Hao Liang, Yuanxing Zhang, and Wentao Zhang. Scone: Bridging composition and distinction in subject-driven image generation via unified understanding-generation modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7773–7783, 2026c.

Xinyu Wei, Kangrui Cen, Hongyang Wei, Zhen Guo, Bairui Li, Zeqing Wang, Jinrui Zhang, and Lei Zhang. Mico-150k: A comprehensive dataset advancing multi-image composition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 29695–29706, June 2026.

Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng-ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, et al. Qwen-image technical report. *arXiv preprint arXiv:2508.02324*, 2025a.

Chenyuan Wu, Jiahao Wang, Pengfei Zheng, Ruiran Yan, Shitao Xiao, Xin Luo, Yueze Wang, Wanli Li, Xiyan Jiang, Yexin Liu, et al. Omnigen2: Towards instruction-aligned multimodal generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21964–21975, 2026a.

Mingrui Wu, Hang Liu, Jiayi Ji, Xiaoshuai Sun, and Rongrong Ji. Micon-bench: Benchmarking and enhancing multi-image context image generation in unified multimodal models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8227–8236, June 2026b.

Shaojin Wu, Mengqi Huang, Wenxu Wu, Yufeng Cheng, Fei Ding, and Qian He. Less-to-more generalization: Unlocking more controllability by in-context generation. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 18682–18692. IEEE, 2025b.

Bin Xia, Yuechen Zhang, Junjia Huang, Jiyang Liu, Jingyao Li, Haoru Tan, Sitong Wu, Chengyao Wang, Yitong Wang, Bei Yu, et al. Dreamomni2: Multimodal instruction-based generation and editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 29275–29284, 2026.

Shitao Xiao, Yueze Wang, Junjie Zhou, Huaying Yuan, Xingrun Xing, Ruiran Yan, Chaofan Li, Shuting Wang, Tiejun Huang, and Zheng Liu. Omnigen: Unified image generation. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13294–13304. IEEE, 2025.

Jinheng Xie, Weijia Mao, Zechen Bai, David Junhao Zhang, Weihao Wang, Kevin Qinghong Lin, Yuchao Gu, Zhijie Chen, Zhenheng Yang, and Mike Zheng Shou. Show-o: One single transformer to unify multimodal understanding and generation. In *International Conference on Learning Representations*, volume 2025, pp. 28240–28264, 2025.

Zewei Xin, Qinya Li, Chaoyue Niu, Fan Wu, and Guihai Chen. Adaptive routing of text-to-image generation requests between large cloud model and light-weight edge model. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 19482–19491. IEEE, 2025.

Jingjun Xu, Hongji Pu, Tao Feng, Haozhen Zhang, Jiaxuan You, and Ge Liu. Routeprofile: Elucidating the design space of llm profiles for routing. *arXiv preprint arXiv:2605.00180*, 2026a. URL https://arxiv.org/abs/2605.00180.

Yiyan Xu, Qiulin Wang, Wenjie Wang, Yunyao Mao, Xintao Wang, Pengfei Wan, Kun Gai, and Fuli Feng. Unicustom: Unified visual conditioning for multi-reference image generation. *arXiv preprint arXiv:2605.12088*, 2026b.

Tao Yu, Yifei Qu, Zhiqing Cui, Pengfei Zhou, Zhongtian Luo, Yujia Yang, Shenghua Chai, Haopeng Jin, Zhenghao Zhang, Xinming Wang, et al. Scope-router: Cost-aware open-set vlm routing for execution-oriented tasks. *arXiv preprint arXiv:2608.12127*, 2026.

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.

A Appendix

A.1 Evaluator and Benchmark Details

Evaluator Design. The evaluator separates three questions that are easily conflated by a single holistic judge: whether the supplied references are faithfully used, whether the requested composition is satisfied, and whether the output is visually well formed. Figure 4 distinguishes the query-level requirement-compilation stage from candidate-level scoring and shows how the latter decomposes into R, S, and V. The input visible to each branch is deliberately restricted so that one factor cannot compensate for another. In particular, Prompt Following does not receive the original reference images, and Visual Quality does not receive either the instruction or the references. Table 5 makes these component-wise information boundaries explicit.

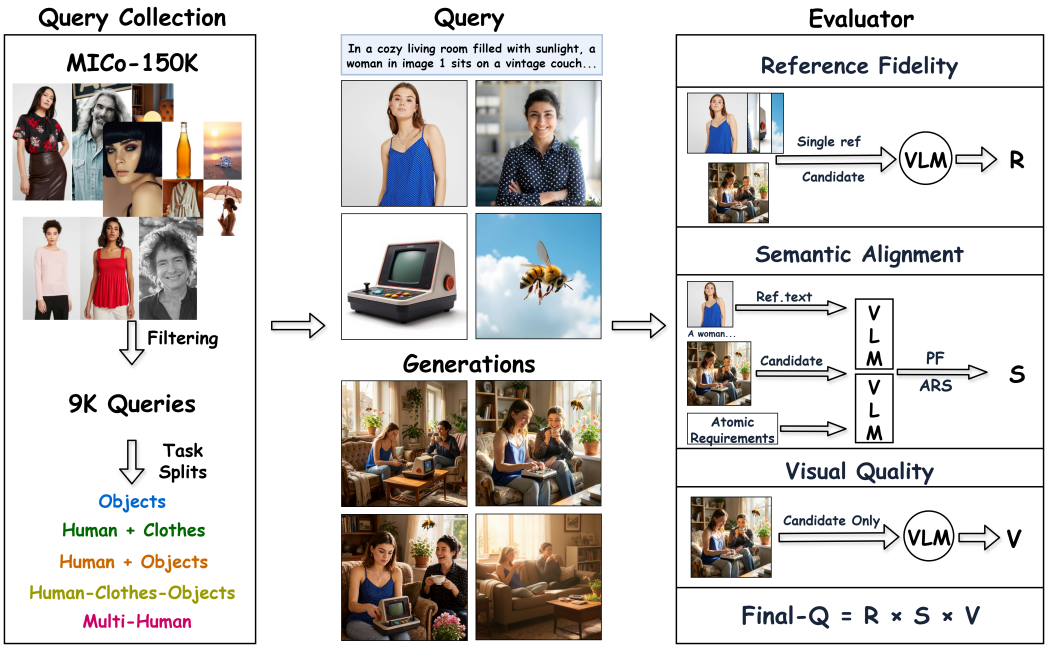

Figure 4: Factorized quality evaluation. Requirement compilation runs once per query from the instruction and ordered reference descriptions, producing frozen atomic checks shared by all candidate generators. At candidate-scoring time, Reference Fidelity R aggregates per-reference preservation evidence; Semantic Alignment S combines holistic Prompt Following P with Atomic Requirement Satisfaction A; and Visual Quality V evaluates the candidate without query conditioning. Their multiplicative composition gives Final-$Q = RSV$.

Table 5: Information boundary of each evaluator component. A checkmark denotes an input intentionally exposed to that component.

Component	Instruction	Ref. text	Ref. images	Candidate
Requirement compiler	✓	✓		
Reference Fidelity R			✓	✓
Prompt Following P	✓	✓		✓
Atomic Requirement Satisfaction A			Relevant only	✓
Visual Quality V				✓

Reference Fidelity and Visual Quality. Reference Fidelity is judged independently for every ordered reference and is then averaged with equal reference weights. For non-face references, Qwen3-VL-30B and Qwen3-VL-8B compare the candidate with one reference at a time; their aggregate is $R = 0.75R_{30B} + 0.25R_{8B}$. Face identity is handled by the ArcFace similarity branch before the

same per-reference aggregation. Visual Quality examines the candidate alone for naturalness, structural coherence, clarity, and visible artifacts. DeepSeek-V4.1-Flash (`deepseek-flash`) scores naturalness N and artifact freedom F on the same 0–10 scale; we set $\mathrm{PQ} = \sqrt{NF}$ and $V = \mathrm{PQ}/10$.

Atomic Requirement Satisfaction. ARS replaces an unconstrained semantic summary with a frozen set of visually verifiable facts. A query-level compiler receives only the instruction and the ordered textual reference descriptions; it never observes a candidate image or generator identity. DeepSeek-V4.1-Flash performs three independent text-only compilations. Schema-valid facts are aligned into consensus clusters, eligible singletons pass a constrained rescue stage, and sanitation removes nonvisual, duplicate, overlapping, or unscorable facts. The compiler binds one entity to each ordered reference and extracts atomic facts of the following types: existence, attribute, action, relation, spatial layout, count, scene, camera, style, and rendered text. Every fact must retain an exact supporting span from the instruction. Subjective intent, narrative implication, sound, and details copied only from a reference description are excluded from semantic scoring.

The tuple is accepted only if it satisfies a strict schema: reference IDs are one-to-one, entity links are valid, relation and spatial facts contain both arguments, source spans occur verbatim in the instruction, and duplicate facts are rejected. Deterministic rendering converts each accepted fact into one short visual question. For example, a relation tuple $(\text{person}, \texttt{wear}, \text{coat})$ becomes "Is the person wearing the coat?" Relevant reference images are attached only when identity or binding is needed. A deterministic Qwen3-VL-30B verifier returns exactly `YES` or `NO`; it is explicitly instructed not to judge aesthetics or unrelated requirements. If J_i is the answer for the i-th valid atomic check, then

$$A = \frac{1}{N_A} \sum_{i=1}^{N_A} \mathbf{1}[J_i = \texttt{YES}], \qquad \widetilde{A} = 0.1 + 0.9A. \tag{4}$$

The floor in $\widetilde{A}$ prevents one uncertain binary check from erasing all other evidence, while preserving monotonicity in Atomic Requirement Satisfaction.

Semantic Alignment and Final-Q. Prompt Following P is a holistic check over the instruction, candidate, and frozen textual reference descriptions. DeepSeek-V4.1-Flash scores five equally weighted aspects: entities and counts, attributes and roles, actions and interactions, spatial relations, and scene composition. Their normalized mean is P. It complements ARS: P captures global instruction coherence, whereas A tests explicit compositional facts. We use

$$S = \sqrt{P\widetilde{A}}, \qquad Q = R\,S\,V, \tag{5}$$

where all factors are normalized to $[0, 1]$. The geometric composition requires balanced performance and prevents a visually polished but semantically wrong image from receiving a high final score.

Freezing and Audit Trail. Requirement tuples, rendered questions, reference routing, and prompt hashes are frozen once per query and reused for every generator. Each judge must return a schema-valid bounded result. Parsing failure or incomplete component evidence invalidates the outcome rather than silently assigning a score. Released records retain component scores, atomic answers, judge identity, prompt hash, and provenance, allowing Final-Q to be reconstructed without rerunning the judges.

Per-Query Cost Traces. For the nine self-hosted generators, MR-RouteBench retains the runtime recorded when each benchmark image was generated on our RTX 4090 server. The proprietary column uses its fixed per-image API price. We release both raw latency and the derived query–generator cost matrix so that monetary cost can be recomputed under a declared GPU-hour rate. For the paper's cross-benchmark comparisons, we use the independent A100 cost registry described in Appendix A.6, avoiding confounding from historical server load while leaving the benchmark-level traces available for latency-aware research.

Human-Alignment Protocol. The human-alignment sample contains 512 distinct queries, stratified by task structure and reference count without consulting either automatic or human scores. It contains 128 queries for each reference count from two to five. Once annotation began, the query manifest, rating question, scale, evaluator definitions, and analysis metrics were frozen.

Table 6: Automatic evaluators compared in the human-alignment study. SR denotes subject resemblance, PF Prompt Following, and W Reference Fidelity.

Evaluator	Components	Judge / backbone	Aggregation
Traditional Composite	IQA, IAA, reference similarity, and presence	Q-Align; Grounding-DINO + DINOv2	Equal mean of normalized component scores
MICo-original	W, PF, SR, naturalness, artifact freedom	DeepSeek judge adaptation of the MICo protocol	$W\sqrt{\mathrm{PF}\,\mathrm{SR}}\sqrt{NF}/100$
MICo w/o SR	W, PF, naturalness, artifact freedom	Same as MICo-original	$W\,\mathrm{PF}\sqrt{NF}/100$
MultiBanana-Qwen	Instruction, reference, subject, realism, visual quality	Official Qwen3-VL-8B rubric	Equal mean of five 1–10 scores

Table 7: MR-RouteBench assets and main evaluation allocation.

Asset	Value	Asset	Value
Queries	9,000	Reference count K	2–5
Candidate columns	10	Dense outcomes	90,000
MICo structures	12	Capability probes	512
Router training	6,888	Validation / test	600 / 1,000

For each query, three raters independently viewed the instruction, all ordered references, and all ten anonymized generator outputs. Generator identities, task labels, automatic scores, and the other raters' responses were hidden; the ten outputs were independently randomized. Each rater assigned every candidate one integer Overall Quality score from 1 to 10. The equal-weight mean of the three ratings defines the human score for that candidate, and the ten human scores define the within-query Human GT ranking.

Every metric is first computed within a query and then macro-averaged over the 512 queries. Kendall τ_b is primary because it measures pairwise ordering while accounting for ties in the discrete human scores. Spearman ρ complements it with rank-displacement sensitivity. Top-1 Agreement records whether the evaluator-selected generator belongs to the human top-score tie set. Human Regret is the difference between the highest human score and the human score of the evaluator-selected generator, on the original 1–10 scale. Together, these measures evaluate generator-ranking accuracy and distinguish near-tie selection errors from consequential mistakes.

Table 6 specifies the comparison evaluators used in the human-alignment study. All variants are evaluated on the same 512 query-level candidate sets and compared against the same three-rater mean Overall Quality scores.

Benchmark Allocation. MR-RouteBench contains 9,000 queries and ten outcomes per query. The main protocol assigns 1,000 queries to testing and 600 to model selection. From the remaining queries, 512 form the Capability Probe Bank and 6,888 train the matcher. These roles are disjoint. MultiBanana is evaluated separately with group-aware five-fold out-of-fold splits. The component ablation and probe-budget analysis use a controlled allocation that assigns 600 of the complete 9,000 queries to *Analysis Validation* and divides the remaining 8,400 between probes and matcher training. *Retrieval Validation* denotes the 600-query validation subset of an earlier stratified development allocation comprising 512 probes, 6,688 matcher-training queries, and a disjoint 1,200-query internal holdout. It is used only for the neighborhood-size sweep in Table 11; it is neither Analysis Validation nor the 600-query Validation partition paired with the final 1,000-query Test split, which was established later. All main-table outcomes are computed exclusively on that final Test partition. The resulting benchmark assets and split allocation are reported in Table 7.

Generator Pool. Table 8 summarizes the fixed candidate pool and the two generators used exclusively in the open-pool onboarding experiments.

Table 8: MR-RouteBench candidate pool and onboarding-only generators. The dashed rule separates open-weight and proprietary models. Seedream-5.0 and Wan2.7-Image are excluded from the fixed ten-column pool.

Generator	Provider	Access	Scale	Cost source	Mean cost
DreamOmni2	DVLab	Open-weight	12B+7B	Measured GPU	\$0.04989
FLUX.2 [klein] 9B	BFL	Open-weight	9B	Measured GPU	\$0.00245
FLUX.2 [klein] 4B	BFL	Open-weight	4B	Measured GPU	\$0.00138
OmniGen2	VectorSpaceLab	Open-weight	4B	Measured GPU	\$0.05662
Qwen-Image-Edit-2511	Alibaba	Open-weight	20B	Measured GPU	\$0.14566
Scone	PKU/Kuaishou	Open-weight	15B	Measured GPU	\$0.06268
SenseNova-U1.5-8B-MoT	SenseTime	Open-weight	8B	Measured GPU	\$0.04060
SenseNova-U1.5-8B-MoT-LoRA-8step	SenseTime	Open-weight	8B	Measured GPU	\$0.00605
UNO	ByteDance	Open-weight	12B	Measured GPU	\$0.01393
NanoBanana (MICo)	Google	API	Undisc.	Price proxy	\$0.13400
Seedream-5.0	ByteDance	API	Undisc.	Relay API	\$0.03300
Wan2.7-Image	Alibaba	API	Undisc.	Relay API	$\approx$ \$0.028

Table 9: Evaluation protocols for the two external benchmarks. Split counts are reported per fold for MultiBanana; its test folds cover every query exactly once.

Benchmark	Queries	References	Candidate pool	Evaluator	Routing protocol
MICoBench	897	2–10	MR-RouteBench ten-column candidate pool	Official Weighted-Ref-VIEScore with a DeepSeek vision-judge adaptation	MR-trained router evaluated on the external query set; 97 structurally unavailable cells are masked from selection
MultiBanana	672	3–5	Nine self-hosted generators + NanoBanana Pro	Official MultiBanana rubric with Qwen3-VL-8B	Group-aware five-fold OOF; per fold: 96 probes, 374/375 train, 67 validation, and 134/135 test queries

Fixed-Policy Anchors. Best Fixed is the single generator with the highest mean development quality, selected once and then applied to every evaluation query. It is NanoBanana (MICo) on MR-RouteBench and MICoBench, and SenseNova-8step in every MultiBanana fold. The Quality Oracle instead selects the highest-quality available generator independently for each evaluation query and is therefore a hindsight ceiling, not a deployable fixed policy.

Table 10 reports the complete generator-level breakdown of the dense benchmark matrix. The component scores expose distinct model strengths rather than only a single overall ranking, while the cost column shows that quality and expense are not interchangeable. The two onboarding-only API generators are reported on the 200-query set and excluded from the closed-pool best and second-best markings.

A.2 Qualitative Examples from MR-RouteBench

Figure 5 shows four held-out test queries from the *Human + 3 Clothes*, *3 Objects*, *Human + Clothes + Objects*, and *2 Humans + 2 Objects* families. For each query, the ordered references are followed by five outcomes chosen to make the decisions of the compared routing policies visible. The number below an outcome is Final-Q from Equation 3; the adjacent dollar value is its generation cost for that reference count. A policy label is placed under the outcome it selects, and labels are grouped when policies agree.

The examples illustrate why routing is evaluated at the query level. Different generators lead on different reference configurations, and policies that agree on one query can diverge on another. MRG-Router matches the quality oracle in the second, third, and fourth rows. In the first row, its selected outcome is less expensive than the oracle under the reported deployment operating point, making the associated quality–cost trade-off explicit rather than presenting only favorable selections.

Table 10: Quality and cost of the generator pool. The first ten rows are the closed-pool candidate columns evaluated on all 9,000 MR-RouteBench queries; component and Final-Q values are reported in percentage points. Seedream-5.0 and Wan2.7-Image are evaluated only in the open-pool onboarding experiment and their values are means over the fixed, task-stratified 200-query onboarding set. They are excluded from the best and second-best markings, which are computed only within the fixed ten-column candidate pool. Cost is the reference-count-weighted mean USD per image defined in Equation 19.

Generator	$R\uparrow$	$S\uparrow$	$V\uparrow$	Final-$Q\uparrow$	Cost$\downarrow$
NanoBanana (MICo)	<u>83.51</u>	76.32	**72.57**	**47.19**	0.13400
SenseNova-8step	**86.11**	**82.05**	64.98	<u>46.23</u>	0.00605
SenseNova-Full	78.41	<u>79.90</u>	<u>70.40</u>	44.84	0.04060
FLUX.2 [klein] 9B	78.15	75.33	66.97	40.27	<u>0.00245</u>
FLUX.2 [klein] 4B	72.01	72.51	69.18	37.41	**0.00138**
Scone	71.65	72.46	63.10	33.92	0.06268
OmniGen2	69.92	65.27	67.47	32.26	0.05662
Qwen-Image-Edit-2511	64.48	62.68	69.17	30.67	0.14566
DreamOmni2	61.39	59.33	65.85	25.96	0.04989
UNO	56.98	58.54	64.52	23.71	0.01393
Seedream-5.0 (onboarding)	86.03	74.62	71.85	46.89	0.03300
Wan2.7-Image (onboarding)	93.62	72.86	71.06	48.87	≈0.02800

These examples complement the aggregate benchmark results by exposing the visual trade-offs behind individual routing decisions. Generators with similar Final-Q values can fail in different ways: one may preserve a salient reference while omitting another requested entity, whereas another may include all entities but weaken appearance fidelity or spatial coherence. The preferred generator therefore depends jointly on the instruction, the ordered reference set, and the quality–cost operating point, rather than on the query category or textual instruction alone.

A.3 Router and Baseline Implementation

Capability Profile Estimation. The default Capability Probe Bank contains $P = 512$ multi-reference queries selected by metadata-stratified sampling without using generator-quality labels. It allocates 128 probes to each reference count in $\{2, 3, 4, 5\}$ and distributes them across the benchmark's task structures, preventing the most frequent query form from dominating the profile. Post-stratification weights w_s restore the workload distribution after this balanced sampling. Every generator is evaluated on the same probe set, so profile columns differ by measured behavior rather than by calibration data.

Let $I_{s,m}$ indicate whether generator m has a valid outcome for probe s, and let $\mathcal{M}_s$ denote the generators with valid outcomes on that probe. The model mean, shared probe difficulty, and centered residual are estimated as

$$\mu_m = \frac{\sum_s w_s I_{s,m} Q^{\mathcal{P}}_{s,m}}{\sum_s w_s I_{s,m}}, \qquad d_s = \frac{1}{|\mathcal{M}_s|} \sum_{m \in \mathcal{M}_s} \left(Q^{\mathcal{P}}_{s,m} - \mu_m \right), \qquad A_{s,m} = Q^{\mathcal{P}}_{s,m} - \mu_m - d_s. \tag{6}$$

The resulting profile $c_m = (\mu_m, A_{:,m})$ records measured generator behavior in a common probe coordinate system. With normalized probe weights, its routing score satisfies

$$\mu_m + \sum_{s \in \mathcal{N}_K(i)} \alpha_{i,s} A_{s,m} = \sum_{s \in \mathcal{N}_K(i)} \alpha_{i,s} Q^{\mathcal{P}}_{s,m} - \sum_{s \in \mathcal{N}_K(i)} \alpha_{i,s} d_s. \tag{7}$$

The final term is shared by all generators and therefore does not affect their ranking. The decomposition is a centered parameterization of the profile, while the decision is determined by query-weighted probe outcomes. Invalid outcomes are masked throughout profile estimation and policy evaluation.

Multimodal Representation and Retrieval. We use frozen BGE-M3 and SigLIP2 encoders for text and reference images, respectively, and mean-pool features across references. Training-set standardization statistics are estimated once on Router Training and shared by ordinary queries and capability probes. The standardized text and image vectors are concatenated and ℓ_2-normalized to



Figure 5: Held-out qualitative examples from MR-RouteBench. Each row contains the instruction, ordered references, and five outcomes from the frozen generator pool. The values below each image are Final-Q and generation cost in USD for that query. Method labels mark the selected outcome; red denotes the quality oracle and purple denotes MRG-ROUTER.

form z_i. Semantic Probe Retrieval then selects

$$\mathcal{N}_K(i) = \mathrm{TopK}_{s\in\mathcal{P}} \cos(z_i, z_s), \qquad \alpha^0_{i,s} = 1/K, \tag{8}$$

with $K = 32$ in the final model.

Retrieval Neighborhood Selection. The neighborhood size is selected on Retrieval Validation before test evaluation. Dense retrieval distributes weight across most of the 512 probes and smooths away local capability differences. Restricting retrieval to semantically related probes improves quality across three seeds. Top-32 and Top-64 fall on the same performance plateau, while Top-32 has a slightly higher mean, lower cross-seed variation, and half as many retrieved probes. We therefore use Top-32 in all reported experiments.

Table 11: Retrieval-neighborhood selection on Retrieval Validation. Values are mean quality and standard deviation over three seeds.

Retrieval	Final-$Q\uparrow$	Retrieved probes
Dense	0.4837 ± 0.0016	512
Top-64	0.4902 ± 0.0023	64
Top-32	$\mathbf{0.4915 \pm 0.0016}$	32

Query-Conditioned Probe Reweighting. For each retrieved query–probe pair, QPR first constructs

$$q_i = W_q z_i, \qquad k_s = W_k z_s, \qquad \phi_{i,s} = [q_i \odot k_s; |q_i - k_s|; \cos(q_i, k_s); \cos(z_i, z_s)]\,. \tag{9}$$

It then predicts a bounded residual correction,

$$\delta_{i,s} = \tanh g_\theta(\phi_{i,s}), \qquad \alpha_{i,:} = \mathrm{softmax}\left(\log \alpha^0_{i,:} + \delta_{i,:}\right). \tag{10}$$

The QPR MLP contains 525,313 trainable parameters and uses a zero-initialized final layer, so optimization starts exactly from the Uniform Probe Router. The $\tanh$ bound and probe prior constrain QPR to learn a local correction rather than unconstrained probe weights. QPR parameters are shared across all generators and receive neither model identity nor generator-specific logits.

Training Objectives. Training proceeds in two stages. For probe s and training query i, the Stage-I warm-up defines the probe-induced generator policy

$$\pi^s_m = \mathrm{softmax}_m\left(\frac{Q^{\mathcal{P}}_{s,m}}{\tau_p}\right), \qquad y_{i,s} = \exp\left[-\frac{\max_m Q_{i,m} - \sum_m \pi^s_m Q_{i,m}}{\tau_u}\right]. \tag{11}$$

$\mathcal{L}_{\text{utility}}$ is the Smooth-L1 loss between the predicted probe utility and $y_{i,s}$; the pairwise probe-ranking term ignores near-tied targets. Stage I optimizes

$$\mathcal{L}_{\text{pre}} = \mathcal{L}_{\text{utility}} + 0.2\mathcal{L}_{\text{probe-rank}}. \tag{12}$$

Stage II directly optimizes the routed decision. From the predicted generator qualities, we form

$$\rho_{i,m} = \mathrm{softmax}_m\left(\widehat{Q}_{i,m}/\tau_m\right), \qquad \mathcal{L}_{\text{regret}} = \frac{1}{N}\sum_i \left[\max_m Q_{i,m} - \sum_m \rho_{i,m} Q_{i,m}\right]. \tag{13}$$

The final routing objective is

$$\mathcal{L}_{\text{route}} = \mathcal{L}_{\text{regret}} + 0.2\mathcal{L}_{\text{model-rank}} + 0.1\mathcal{L}_{\text{utility}} + 0.02\mathcal{L}_{\text{trust}}, \qquad \mathcal{L}_{\text{trust}} = \mathrm{KL}(\alpha_i \| \alpha^0_i). \tag{14}$$

The final term is the Probe-Prior Regularization, which discourages QPR from departing excessively from the retrieved semantic neighborhood. The remaining terms align routed quality, generator ordering, and local probe utility.

Optimization Details. Stage I and Stage II run for 5 and 20 epochs with AdamW, using learning rates 3×10^{-4} and 1.5×10^{-4}, respectively, weight decay 10^{-3}, batch size 192, and gradient clipping at 1.0. We use $\tau_p = \tau_m = 0.03$ and $\tau_u = 0.05$, select checkpoints on validation quality, and aggregate learned methods over three random seeds. Encoder weights and training-set standardization statistics remain fixed throughout training.

Inference and Policy Selection. At inference time, the router encodes the query once, retrieves its Top-32 neighborhood, predicts $\alpha_{i,s}$, and scores each generator as

$$\widehat{Q}_{i,m} = \mu_m + \sum_{s \in \mathcal{N}_K(i)} \alpha_{i,s} A_{s,m}. \tag{15}$$

By Equation 7, pure-quality routing is equivalent to ranking the query-weighted raw probe outcomes. A deployment may instead apply a cost policy to the same predictions, for example choosing among near-best generators by monetary cost. Cost is not an input to QPR and does not alter the learned capability profile.

Table 12: Adaptation of routing baselines to multi-reference image generation.

Method	Adaptation used in our setting
KNN	Retrieves queries in the shared multimodal feature space and aggregates their observed generator qualities.
PRkNN	Combines neighborhood evidence with a probe-estimated model mean.
Direct Predictor	Regresses all generator qualities directly from the fused instruction–reference representation.
ZOOTER-Adapt	Distills the dense quality vector into a distribution over generators using KL divergence.
RouterDC-Adapt	Learns query and generator embeddings with a quality- distribution objective and query–query preference geometry.
SCOPE-Adapt	Uses a query/profile dual tower; generator profiles contain probe behavior, cost, and summary statistics, and training samples cost preferences.

Baselines. All methods receive the same frozen multimodal query features and candidate pool. Their adaptations are summarized in Table 12. The goal is to preserve each method's defining learning signal while replacing task-specific inputs with the common MRIG representation and dense quality matrix. No baseline sees category labels, current-query generations, or external-test quality at inference. Learned methods use the same development allocation, checkpoint rule, and three random seeds.

Evaluation Protocol. Each run uses its designated training and validation roles for parameter fitting and checkpoint selection. For learned methods, we report the mean and standard deviation over three predetermined seeds; deterministic methods are run once. The candidate pool, evaluator scores, feature caches, cost registry, and split IDs are identical across methods. MultiBanana follows grouped out-of-fold evaluation so that each reported query is routed by a model trained without that query's fold. Full external-benchmark allocations are listed in Table 9.

Quality–Cost Reporting. For selection, the reference-count-dependent cost is scaled to the typical range of validation predictions. If $\mathcal{V}$ is the validation set, we use

$$s = \frac{\mathrm{median}_{i\in\mathcal{V}}[\max_m \widehat{Q}_{i,m} - \min_m \widehat{Q}_{i,m}]}{\mathrm{median}_{i\in\mathcal{V}}[\max_m C_m(K_i) - \min_m C_m(K_i)] + 10^{-12}}, \qquad \widetilde{C}_m(K_i) = sC_m(K_i). \quad (16)$$

For score-minus-cost policies, we search $\lambda \in \{0, .01, .03, .1, .3, 1, 3, 10, 30\}$ on validation Q–C score, breaking ties by validation quality, and freeze the selected value for evaluation. MultiBanana performs this selection independently inside each fold.

For reporting, quality is mapped to $[0, 100]$ without pooling benchmarks: $q = 100\,\overline{Q}$ for MRRouteBench, $q = 100\,\overline{Q}_{\mathrm{VIE}}$ for MICoBench, and $q = 10\,\overline{Q}_{\mathrm{MB}}$ for MultiBanana's native 1–10 score. Let C be average USD/query. We define $C_{\min}$ and $C_{\max}$ as the query-averaged minimum and maximum cost among the available candidates in the same benchmark, and compute

$$c = 100\,\mathrm{clip}\left(\frac{\log C_{\max} - \log C}{\log C_{\max} - \log C_{\min}}, 0, 1\right), \qquad \mathrm{QCS}_\beta = \frac{(1+\beta)qc}{\beta q + c}, \quad \beta = 0.1. \quad (17)$$

The Quality Oracle chooses the highest-Q available generator independently for each query. It is a quality ceiling rather than a Q–C upper bound because its selected generators may be expensive. Best Fixed is defined in Appendix A.1 and never varies across evaluation queries.

A.4 Probe Bank Analysis

We vary the probe budget within a controlled allocation of 8,400 queries. Increasing the Probe Bank therefore reduces the number of queries available for matcher training. Table 13 reports the exact budget sweep, while Figure 6 summarizes both quality and resampling stability. Analysis Validation quality peaks at $P = 512$, and cross-bank variance decreases as the bank grows.

Independent probe resampling further measures sensitivity to the particular Probe Bank. The standard deviation of MICoBench quality decreases from 0.27 points at $P = 64$ to 0.16 at $P = 128$ and 0.10 at $P = 512$. At $P = 512$, diversity-stratified sampling also reduces variance relative to uniform sampling, indicating that the fixed 512-probe operating point is stable under resampling.

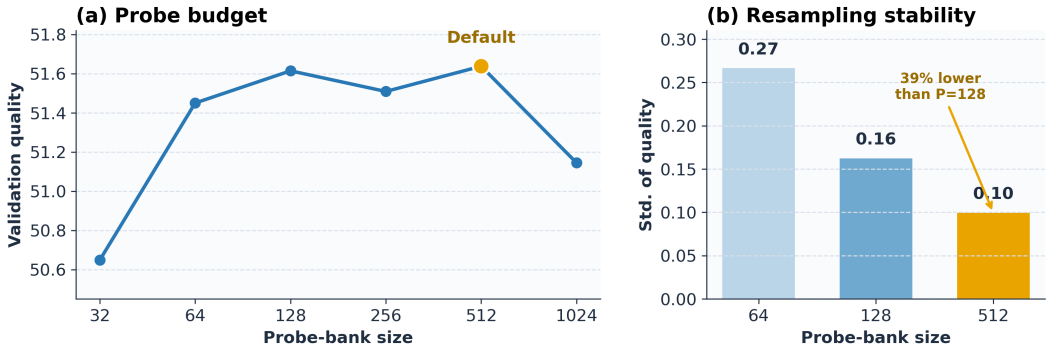

Figure 6: Probe Bank allocation and resampling stability under a controlled 8,400-query allocation. The 512-probe setting combines the highest Analysis Validation quality with low variance across independently sampled banks.

Table 13: Probe budget on Analysis Validation (600 queries).

P	32	64	128	256	512	1,024
Quality	50.65	51.45	51.61	51.51	**51.64**	51.14

This post-hoc sensitivity analysis is not used to select the reported probe budget, model parameters, or checkpoint.

A.5 Open-Pool Protocol

In leave-one-generator-out evaluation, all profile evidence from the held-out generator is excluded during router training. At onboarding time, the unseen generator is evaluated once on the fixed 512-query Capability Probe Bank. We compute $c_{m_{\text{new}}} = (\mu_{m_{\text{new}}}, A_{:,m_{\text{new}}})$ and append it as a new column of the Capability Profile Table. The query encoder and QPR parameters remain bitwise unchanged throughout onboarding. The shared probe-difficulty terms are estimated from the base pool and frozen; a new column is added without recomputing existing profiles.

Leave-one-generator-out evaluation uses the same 600-query development validation set in each of ten held-out-generator episodes, giving 600×10 routing decisions in Table 4. Real-model onboarding draws a fixed 200-query non-probe set from a 300-query generation set. After requiring complete outputs from both new generators, queries are selected without reading their scores, using proportional stratification over task structure and reference count. Adoption is the fraction of these 200 queries routed to an onboarded generator. We report both adoption and quality gain because high adoption alone does not establish that the newly introduced generator improves routing quality.

A.6 Cost and Reproducibility Notes

CostBench-300 Composition. GPU cost is measured on CostBench-300, a timing set sampled from a 1,000-query MICo-150K timing pool using seed 20260912. The set contains 25 queries from each of the twelve MR-RouteBench task structures. It is also exactly balanced by reference count, with 75 queries for each $K \in \{2, 3, 4, 5\}$. A separate five-query-per-structure reserve set (60 queries) and ten warm-up queries are not included in the reported measurements. This design separates timing from the frequency of any one benchmark structure and provides equal support for the four reference-count conditions.

Each self-hosted generator processes the same 300-query manifest with serial batch-one inference on one NVIDIA A100-SXM4-40GB. The timer covers generation for one query and excludes model loading and prompt-rewrite preprocessing. Output resolution, inference steps, guidance, precision, and all other generator settings are frozen before timing. Table 14 reports the observed arithmetic mean and median for the endpoint conditions $K = 2$ and $K = 5$; measurements for all four reference counts remain part of the cost registry. Table 15 records the corresponding operating points. Scale labels in Table 8 follow the released checkpoints and documented backbones: DreamOmni2 combines a 12B FLUX.1-Kontext generator with a separate 7B Qwen2.5-VL module (Xia et al.,

Table 14: Measured CostBench-300 inference latency for the nine self-hosted generators at the endpoint reference counts $K \in \{2, 5\}$. Each entry reports mean/median seconds per query over 75 successful queries; the complete $K \in \{2, 3, 4, 5\}$ measurements are used for cost conversion, with the two intermediate columns omitted here for compactness. Runs use serial batch-one inference on one NVIDIA A100-SXM4-40GB. Model loading and external prompt rewriting are excluded; peak memory is the maximum observed over the full 300-query run.

Generator	$K = 2$	$K = 5$	Peak VRAM (MiB)
DreamOmni2	56.127 / 56.13	139.188 / 139.21	36,145
FLUX.2 [klein] 9B	3.568 / 3.62	5.578 / 5.53	35,716
FLUX.2 [klein] 4B	1.983 / 2.00	3.108 / 3.15	17,748
OmniGen2	81.085 / 82.75	130.431 / 128.66	18,134
Qwen-Image-Edit-2511	167.568 / 166.88	403.945 / 403.07	22,718
Scone	102.070 / 102.87	124.937 / 126.35	31,934
SenseNova-U1.5-8B-MoT	73.325 / 73.21	73.568 / 73.36	30,247
SenseNova-U1.5-8B-MoT-LoRA-8step	10.658 / 10.77	10.933 / 11.05	29,907
UNO	24.497 / 23.50	28.400 / 27.76	14,069

Table 15: Frozen operating points used for self-hosted generation and timing. Resolution denotes the requested output size; SenseNova uses its native approximately four-megapixel target. Guidance lists text/image scales when the implementation exposes them separately.

Generator	Resolution	Steps	Guidance	Precision / execution mode
DreamOmni2	1024^2	30	3.5	BF16
FLUX.2 [klein] 9B	1024^2	4	1.0	BF16
FLUX.2 [klein] 4B	1024^2	4	1.0	BF16
OmniGen2	1024^2	50	text 5.0 / image 3.0	BF16
Qwen-Image-Edit-2511	896×1152	40	CFG 4.0	BF16, layerwise FP8
Scone	1024^2	50	text 4.0 / image 2.0	official inference
SenseNova-U1.5-8B-MoT	$\sim$4 MP	50	text 4.0 / image 1.0	BF16, balanced memory
SenseNova-U1.5-8B-MoT-LoRA-8step	$\sim$4 MP	8	text 1.0 / image 1.0	BF16, balanced memory
UNO	704^2	25	4.0	FP8 checkpoint, offload

2026), Scone contains 15B parameters (Wang et al., 2026c), and UNO uses the 12B FLUX.1-dev backbone (Wu et al., 2025b).

Conversion to Monetary Cost. The cost registry uses the within-K sample median $\widetilde{t}_{m,K}$ to reduce sensitivity to transient system jitter, while the displayed columns in Table 14 also report arithmetic means to expose the observed runtime directly. For a self-hosted generator m using g_m GPUs, the per-query cost is

$$C_m(K) = \frac{g_m \widetilde{t}_{m,K}}{3600} p_{\mathrm{GPU}}, \qquad p_{\mathrm{GPU}} = \$1.99/\text{GPU-hour}. \tag{18}$$

All nine measured operating points use $g_m = 1$. The model-level mean reported in Tables 8 and 10 applies the MR-RouteBench reference-count distribution,

$$\overline{C}_m = \sum_{K=2}^{5} \frac{n_K}{9000} C_m(K), \qquad (n_2, n_3, n_4, n_5) = (3225, 1926, 1929, 1920). \tag{19}$$

Thus a router is charged according to the selected generator and the query's actual reference count rather than a pool-wide constant.

For MICoBench queries with more than five references, self-hosted costs extend the final measured segment,

$$C_m(K) = C_m(5) + (K - 5)\max\{C_m(5) - C_m(4), 0\}, \qquad K > 5, \tag{20}$$

while API models retain their fixed per-image price.

NanoBanana (MICo) uses a fixed \$0.134/image price proxy (snapshot: 2026-09-20). Here the name identifies MICo's fixed official-output column; because those outputs mix Nano Banana variants without per-image provenance, we use the conservative Pro price rather than infer a lower cost.

MultiBanana instead uses the NanoBanana Pro operating point. Seedream-5.0 and Wan2.7-Image use their recorded relay-API prices only in the onboarding study; Wan's CNY 0.20/image tariff is displayed using CNY 7.2/USD. API prices are fixed snapshots for reproducibility and are not converted from GPU latency.

Cost is maintained as a deployment attribute rather than as part of the Capability Profile or QPR parameters. Consequently, changes in hardware, inference configuration, or API pricing can be reflected by updating the cost registry without reconstructing generator capability profiles or retraining the router.

The supplementary artifacts record split IDs, probe IDs and weights, frozen evaluator components, capability profiles, model-configuration hashes, and the complete cost registry. Image redistribution remains subject to the corresponding source licenses.

B Limitations and Future Work

While MRG-Router provides an effective pre-rendering routing framework, its empirical accuracy depends on the coverage of the Capability Probe Bank, and onboarding an unseen generator currently requires a one-time profiling pass over the fixed probe set. Future research could explore adaptive probe selection and sample-efficient calibration to lower profiling overhead. In addition, the present formulation designates a single generator for the entire request prior to rendering. A natural extension is to integrate behavioral capability profiles into agentic model orchestration, where a coordinating agent decomposes complex generation requests into sub-goals, dynamically dispatches specialized subtasks to complementary generators using probe evidence, and verifies intermediate outcomes with uncertainty-guided fallback.