Towards Fair and Comprehensive Evaluation of Routers in Collaborative LLM Systems

Anonymous ACL submission

Abstract

Large language models (LLMs) have achieved success, but cost and privacy constraints necessitate deploying smaller models locally while offloading complex queries to cloud-based models. Existing router evaluations are unsystematic, overlooking scenario-specific requirements and out-of-distribution robustness. We propose a principled evaluation framework with three dimensions: router ability, scenario alignment, and cross-domain robustness. Unlike prior work that relies on output probabilities or external embeddings, we utilize internal hidden states that capture model uncertainty before answer generation. We introduce **ProbeDirichlet**, a lightweight router that aggregates cross-layer hidden states via input-dependent Dirichlet distributions. Trained on multi-domain data, it generalizes robustly across in-domain and out-of-distribution scenarios. Our results show **ProbeDirichlet** outperforms the best baselines by 16.68% in router ability and 18.86% in high-accuracy scenarios, with strong generalization across heterogeneous tasks and agentic workflows.

1 Introduction

Large Language Models (LLMs) achieve remarkable performance across diverse tasks such as language understanding, creative writing, and code generation(Zhao et al., 2023; Matarazzo and Torlone, 2025), but balancing cost and accuracy under varying deployment constraints remains a key challenge. Routers address this by dynamically directing queries to different models: routing complex queries to powerful cloud models while processing simpler ones on local edge devices(Ding et al., 2024a; Zhang et al., 2025a; Barrak et al., 2025). This reduces computational cost, but may sacrifice some accuracy(Kassem et al., 2025; Shafran et al., 2025; Lin et al., 2025).

However, this trade-off is not equally acceptable across domains. Different domains have different tolerances: safety-critical applications like healthcare require high reliability (Busch et al., 2025), while customer support may tolerate accuracy drops for cost savings (Yu et al., 2025). Beyond domain-specific requirements, routers must also handle queries from unfamiliar distributions (out-of-distribution, OOD). Given these diverse requirements, a single metric cannot capture router quality. Fair evaluation requires assessing both deployment scenarios and cross-domain robustness.

Existing benchmarks fail to achieve this comprehensive assessment. Current evaluations rely on single metrics such as static thresholds (Chen et al., 2024b; Ding et al., 2024b; Stripelis et al., 2024; Aggarwal et al., 2024) or curve-based aggregate scores(Ramírez et al., 2024; Hu et al., 2024; Ong et al., 2025), which cannot capture the multi-faceted trade-offs required across diverse application scenarios (Subsection 3.2). Beyond metric limitations, many studies evaluate routing performance solely on in-distribution data without systematic out-of-distribution (OOD) assessment. However, real-world deployments face diverse, shifting query distributions, requiring comprehensive evaluation of both scenario-specific performance and cross-domain robustness.

Motivated by these gaps, we propose a systematic evaluation framework spanning three key dimensions: **(i) Router Ability**, measured by AUROC to capture a router's fundamental discrimination capability independent of deployment thresholds; **(ii) Scenario Alignment**, quantified by metrics tailored to low-cost, balanced, and high-accuracy deployment regimes (detailed in Section 3.3); and **(iii) Cross-Domain Robustness**, assessed across diverse in-distribution (ID) and out-of-distribution (OOD) tasks. By disentangling intrinsic routing ability from scenario-specific requirements, our framework enables more principled router comparison and guides our exploration of effective routing design.

We then focus on the core challenge: **How to construct routing that is both effective and generalizable?** We explore router design and training data composition, validated on our evaluation framework and agentic applications. Internal hidden states directly capture model uncertainty before answer generation, proving more reliable than output probabilities that suffer from softmax overconfidence (Guo et al., 2017). To adaptively aggregate cross-layer representations, we model layer importance as input-conditional Dirichlet distributions. These inject stochastic regularization during training while maintaining deterministic inference via expectation, acting as layer dropout to prevent overfitting specific depths. We show that diverse data mixtures improve cross-domain generalization while preserving in-distribution performance.

In summary, our contributions are threefold:

- **Identification of Critical Gaps.** We identify three fundamental gaps in routing evaluation: failure to disentangle intrinsic ability from scenario alignment, absence of metrics for diverse deployment regimes, and neglect of out-of-distribution robustness.
- **Systematic Evaluation Framework.** We propose a principled framework that systematically assesses router ability, scenario alignment, and cross-domain robustness, enabling fair comparison and guiding future router development.
- **Robust Routing Method.** We develop a lightweight probe using internal hidden states, achieving 16.68%(Router ability) and 18.86% (HCR) improvements with strong generalization across scenarios and agentic workflows.

2 Related Work

LLM Routing. Prior work explores several technical directions. Training-free approaches avoid labeled supervision by estimating model skill from relative performance (Zhao et al., 2024) or leveraging weak agreement signals (Guha et al., 2024; Aggarwal et al., 2024). Learning-based routing methods train models to predict which model should handle each query, including preference-based routers (Ong et al., 2025), contrastive query–model embedding alignment (Chen et al., 2024c), and instruction-level capability encoding (Zhang et al., 2025b). Adaptive routing formulates routing as sequential decision making, such as bandit-based selection (Li, 2025) or token-level deferral from small to large models (She et al., 2025). Quality- and compute-aware designs integrate routing with explicit test-time budget control, such as Hybrid LLM (Ding et al., 2024b) and BEST-Route (Ding et al., 2025). Beyond specific router designs, recent benchmarking efforts such as RouterEval (Huang et al., 2025) provide comprehensive frameworks to evaluate routing performance and explore the scaling effects of integrating multiple models of varying capacities.

LLM Collaboration. Collaboration strategies complement routing by coordinating multiple models or agents. Representative directions include speculative decoding, which accelerates inference using a draft–verifier pair (Chen et al., 2023; Cai et al., 2024; Li et al., 2024), and model cascades, which escalate queries through models of increasing capacity with calibrated deferral rules (Chen et al., 2024b; Gupta et al., 2024). More recent work explores multi-agent systems with specialized roles and coordination protocols (Wu et al., 2024; Li et al., 2023; Wang et al., 2025).

LLM Uncertainty Estimation. Uncertainty estimation provides key signals for routing. Existing methods include information-based scores such as perplexity or entropy (Fomicheva et al., 2020; Duan et al., 2024; Fadeeva et al., 2024), consistency-based signals from agreement across generations (Kuhn et al., 2023a; Lin et al., 2024b; Qiu and Miikkulainen, 2024), and introspective probes using hidden states or attention patterns (Chen et al., 2024a; Sriramanan et al., 2024; Lin et al., 2024a). These methods can be integrated into routers to improve decision reliability, though many were originally developed outside the routing context.

3 Evaluation Framework

3.1 Problem Setup

We consider routing between two models in an *edge–cloud collaboration* setting: a small model $\mathcal{M}_{\text{small}}$ deployed locally on edge devices for low latency and privacy, and a large model $\mathcal{M}_{\text{large}}$ deployed in the cloud for higher accuracy at greater cost. Given a query $q \in \mathcal{Q}$, the router decides which model to invoke. Let $\delta_{\text{small}}(q), \delta_{\text{large}}(q) \in [0, 1]$ denote the performance of the two models on q. The router computes a score $s(q) \in \mathbb{R}$, and the

[0]Code will be made publicly available.

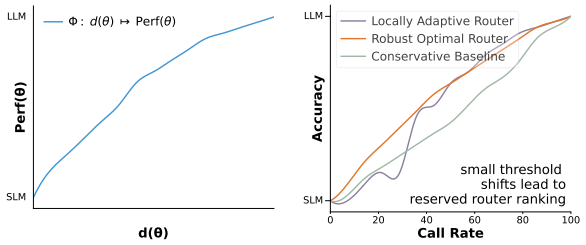

Figure 1: **Left:** Cost–performance mapping where $d(\theta)$ represents the call rate at threshold θ and $\mathrm{Perf}(\theta)$ denotes overall performance. By varying θ, this can be re-parameterized as call rate vs. performance (see §3.1). **Right:** An illustrative limitation of existing metrics.

decision is made by thresholding:

$$r(q;\theta) = \mathbf{1}\{s(q) \geq \theta\}, \tag{1}$$

where $r(q;\theta) = 1$ routes to the large model and $r(q;\theta) = 0$ uses the small model. The resulting system performance under threshold θ is

$$\delta(q;\theta) = (1 - r(q;\theta))\,\delta_{\text{small}}(q) + r(q;\theta)\,\delta_{\text{large}}(q). \tag{2}$$

For a given threshold, the large-model call rate is

$$d(\theta) \;=\; \frac{1}{|\mathcal{Q}|}\sum_{q\in\mathcal{Q}} r(q;\theta) \;\in [0,1], \tag{3}$$

and the corresponding overall performance is

$$\mathrm{Perf}(\theta) \;=\; \frac{1}{|\mathcal{Q}|}\sum_{q\in\mathcal{Q}} \delta(q;\theta). \tag{4}$$

Varying the threshold θ traces out the *cost–performance curve*:

$$\Phi :\; d(\theta) \;\mapsto\; \mathrm{Perf}(\theta). \tag{5}$$

Since $d(\theta)$ is monotonic, we re-parameterize this curve as a continuous function $\Phi(x)$ of the call rate $x \in [0,1]$ via linear interpolation, which serves as the basis for our integral metrics.

3.2 Limitations of Current Metrics

As shown in Figure 1(left), the cost–performance curve introduced above provides a unified view of router behavior. Existing metrics can be seen as different ways of extracting information from this curve, which broadly fall into two categories.

Static Metrics. These methods evaluate routers at fixed thresholds or compress performance into few indicators. A common approach is the cost–accuracy trade-off: FrugalGPT (Chen et al., 2024b) fixes accuracy and reports cost savings, while HybridLLM (Ding et al., 2024b) fixes cost and measures accuracy drop. Others use single or composite indicators. TO-Router (Stripelis et al., 2024) reports total inference cost, throughput, semantic similarity, and negative log-likelihood. AutoMix (Aggarwal et al., 2024) uses Incremental Benefit per Cost, normalizing accuracy improvement by cost into a single score.

Limitation. While static metrics are simple and interpretable, they provide only a fragmented view of router behavior. As illustrated in Figure 1 (right), router rankings can be highly sensitive to threshold choice: within the call-rate range 20% to 40% , even minor shifts can lead to opposite conclusions about the Locally Adaptive Router, indicating that static evaluations may capture incidental fluctuations rather than a router's consistent behavior.

Curve-based Metrics. These methods integrate performance over the entire cost–performance curve to avoid thresholds. Examples include the AUC (area under the accuracy–cost curve) (Ramírez et al., 2024), Average Improvement in Quality (Hu et al., 2024), and Average Performance Gap Recovered (Ong et al., 2025). By summarizing global trends, these metrics provide threshold-independent evaluations of the trade-off surface.

Limitation. Aggregation, however, is scenario-blind. The Figure 1(right) also shows the limitation. Locally Adaptive Router performs poorly in low call-rate regions, but AUC scores conceal this difference and limit interpretability.

More fundamentally, cost–accuracy metrics entangle two factors: *router ability*, referring to the correctness of judgments relative to the small model's capacity, and *scenario alignment*, concerning the leverage of the large model's performance. Since end-to-end accuracy at a given cost reflects both, high scores may stem from the large model's strength rather than the router's skill, preventing faithful assessment of intrinsic routing capability.

3.3 Triple-Perspective Framework

To address this conflation, we propose a triple-perspective framework (Figure 2) that independently evaluates three distinct dimensions of routing performance. AUROC captures intrinsic discriminative ability without considering deployment costs. LPM, HCR, and MPM assess scenario alignment by quantifying how well routing matches specific cost-quality constraints. Cross-domain robust-

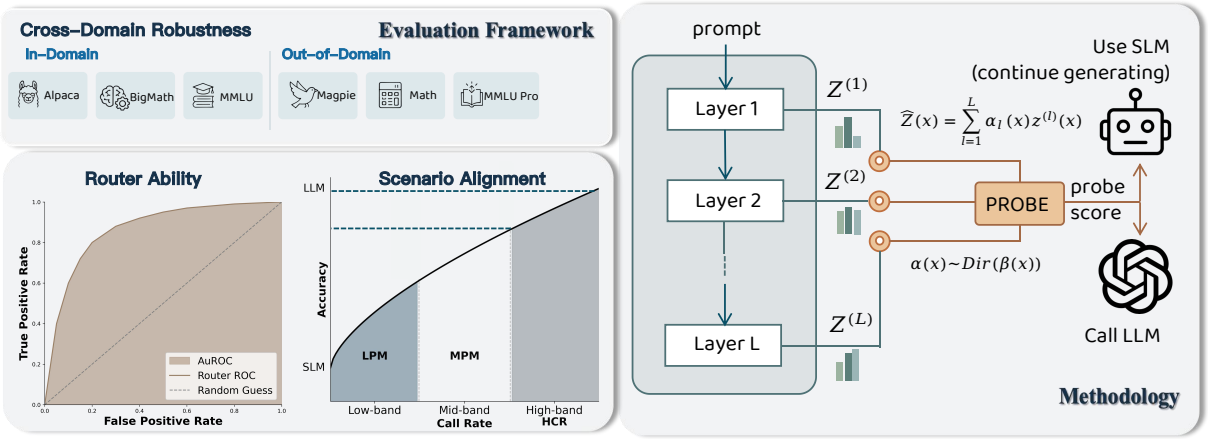

Figure 2: Overall collaboration architecture and evaluation protocol. Router ability is quantified using AUROC, measuring the router's accuracy in predicting whether the SLM can answer correctly. Scenario alignment is evaluated across three call-rate regimes: low band (Low-band Performance Mean, LPM), mid band (Mid-band Performance Mean, MPM), and high band (High-band Call-Rate, HCR).

ness examines performance stability across diverse task distributions to ensure reliable generalization.

1. Router Ability. Since the router's primary role is to decide which model to invoke, end-to-end system accuracy may blur its individual contribution. To isolate the router's discriminative power from the large model's capabilities, we define ground truth labels based on the small model's performance. Varying the decision threshold traces an ROC curve, and the area under this curve (AUROC) provides a threshold-independent measure of discriminative ability. Unlike cost-accuracy metrics, AUROC focuses solely on the router's decision quality, and by aggregating over all thresholds, it avoids sensitivity to local fluctuations or opportunistic peaks.

2. Scenario Alignment. Routers with similar intrinsic ability can behave differently under deployment constraints. To reflect such differences, we partition the cost–performance curve into three regions: (i) low call-rate for budget-sensitive use, (ii) high accuracy for safety-critical domains, and (iii) a middle band for balanced deployment. For each region, we define a normalized mean metric: LPM, HCR, and MPM. As illustrated in Figure 2.

Low-band Performance Mean (LPM). For strict budget scenarios, let $d_1 \in (0, 1]$ denote the maximum allowable call rate. The average performance in this region is defined as:

$$\mathrm{LPM} = \frac{1}{d_1} \int_0^{d_1} \Phi(x)\, dx. \tag{6}$$

High-band Call Rate (HCR). For accuracy-critical applications, we target a specific Relative Performance (RP) range. Given an RP interval $[\rho_1, \rho_2]$, we map these to absolute performance thresholds $[\tau_1, \tau_2]$ via:

$$\tau_i = \mathrm{Perf}_S + \rho_i(\mathrm{Perf}_L - \mathrm{Perf}_S), \quad i \in \{1, 2\}. \tag{7}$$

We then identify the *feasible call-rate set* $\mathcal{D}$ where the router's performance curve $\Phi(x)$ falls within this absolute band:

$$\mathcal{D} = \{x \in [0, 1] : \tau_1 \le \Phi(x) \le \tau_2\}. \tag{8}$$

The HCR metric computes the complement of the average call rate within this feasible set:

$$\mathrm{HCR} = 1 - \frac{1}{|\mathcal{D}|} \int_{x \in \mathcal{D}} x\, dx. \tag{9}$$

A higher HCR indicates the router maintains high accuracy while relying more on the small model.

Mid-band Performance Mean (MPM). This metric evaluates the trade-off efficiency in the transition region between the strict budget constraint (d_1) and the accuracy-critical zone. Let d_2 be the minimum call rate required to satisfy the high-accuracy threshold τ_1:

$$d_2 = \min\{x \in [0, 1] : \Phi(x) \ge \tau_1\}. \tag{10}$$

The mid-band interval is defined as $(d_1, d_2]$. Provided that a valid transition region exists, the mean performance is:

$$\mathrm{MPM} = \frac{1}{d_2 - d_1} \int_{d_1}^{d_2} \Phi(x)\, dx. \tag{11}$$

3. Cross-Domain Robustness We assess cross-domain robustness by evaluating Router Ability across multiple in-distribution (ID) and out-of-distribution (OOD) pairs. This presentation highlights how routers generalize to diverse domains, with benchmarks fully described in Subsection 5.1.

4 Methodology

Guided by this framework, we explore three key aspects: routing on internal hidden states, cross-layer aggregation, and diverse training data.

Motivation. A key challenge in router design is achieving robust performance across both in-distribution and out-of-distribution scenarios. Recent studies reveal that existing routing systems suffer from notable performance degradation under distribution shifts (Ong et al., 2025; Huang et al., 2025). These approaches primarily rely on output-based features (Aggarwal et al., 2024; Zhang et al., 2025a) or external embedding models(Feng et al., 2025) to assess query difficulty. We argue for a fundamentally different approach: routing on **internal hidden states** from the model itself. Unlike output signals or external embeddings, internal representations directly capture the model's uncertainty and computational process before committing to answers. This enables robust routing with lightweight linear classifiers through diverse training data, achieving superior cross-domain generalization.

Cross-layer hidden states provide fine-grained discriminative information. External encoders lack model-internal access, while final output probabilities suffer from overconfidence due to softmax normalization (Guo et al., 2017). We instead route on cross-layer hidden states.

Different layers capture complementary information: early layers encode surface patterns, while deeper layers represent semantic understanding (Sun et al., 2025). Relying solely on the final layer discards intermediate uncertainty. Moreover, internal representations encode task difficulty before answer generation (Dong et al., 2025). We therefore extract and aggregate hidden states directly after the query prefix, combining cross-layer richness with computational efficiency.

Dynamic Dirichlet Aggregation: Probabilistic Training, Deterministic Inference. As shown in Figure 2, we first extract sentence-level representations by mean pooling over token-wise hidden states at each layer l:

$$z^{(l)}(x) = \frac{1}{T}\sum_{t=1}^{T} h_t^{(l)}. \tag{12}$$

The final representation aggregates across layers via a weighted combination:

$$\hat{z}(x) = \sum_{l=1}^{L} \alpha_l(x) z^{(l)}(x). \tag{13}$$

Why Dirichlet? Fixed layer weights (e.g., uniform averaging) cannot adapt to varying query complexity. Simple learned scalars α_l risk overfitting specific layers, especially under distribution shift. We instead model layer importance as an *input-conditional probability distribution*, introducing controlled stochasticity during training while maintaining efficient deterministic inference.

Concretely, a lightweight projection network $g_\theta(\cdot)$ predicts concentration parameters $\beta(x) = [\beta_1(x), \ldots, \beta_L(x)]$ from the pooled representations. During training, layer weights are sampled from a Dirichlet distribution:

$$\alpha(x) \sim \mathrm{Dir}(\beta(x)), \tag{14}$$

where larger $\beta_l(x)$ indicates higher confidence in layer l's relevance. This stochastic sampling acts as a form of *layer dropout*, forcing the router to learn robust features across the entire hidden hierarchy rather than relying on a narrow subset of layers.

During inference, to eliminate sampling overhead and ensure deterministic predictions, we replace the random weights with their expectation:

$$\bar{\alpha}_l(x) = \mathbb{E}[\alpha_l \mid x] = \frac{\beta_l(x)}{\sum_{j=1}^{L} \beta_j(x)}. \tag{15}$$

This formulation bridges probabilistic regularization during training with efficient deterministic aggregation at test time. Intuitively, $\beta_l(x)$ serves as a *layer-wise attention guide* that dynamically reweights layer contribution based on input characteristics. **Mean Pooling** variant emerges as a special case with uniform priors ($\beta_l \equiv c$ for all l).

Diverse Training Data for Cross-Domain Robustness. Beyond architecture design, training data composition critically impacts cross-domain robustness. Single-domain training encourages the router to exploit domain-specific patterns rather than generalizable difficulty signals, limiting transfer to unseen domains.

Table 1: Router ability (AUROC) comparison of routing strategies across multiple benchmarks.

Method	In Domain				Out of Domain						
	Alpaca	Big Math	MMLU	**AVG**	Magpie	MATH	STEM	Humanities	Social Sciences	Others	**AVG**
SelfAsk	49.03	47.20	53.75	49.99	37.09	49.29	53.74	55.86	56.06	50.91	50.49
SemanticEntropy	62.02	55.81	53.93	57.25	58.82	55.25	56.27	51.72	52.90	53.95	54.82
ConfidenceMargin	53.38	56.18	46.56	52.04	43.08	50.05	54.42	46.97	54.37	49.52	49.73
Entropy	46.24	51.41	49.26	48.97	52.62	55.30	49.70	52.36	48.54	49.23	51.29
MaxLogits	57.96	47.39	43.82	49.72	60.86	47.00	50.03	50.53	41.14	46.43	49.33
EmbeddingMLP	67.31	56.18	54.89	59.46	68.97	56.97	52.97	53.77	48.16	50.45	55.22
ProbeDirichlet	**72.02**	**66.18**	**67.88**	**68.70**	**74.08**	**73.90**	**65.32**	**57.84**	**58.82**	**62.77**	**65.46**

We therefore adopt a multi-domain training strategy, training across multiple domains simultaneously. This forces the router to learn cross-domain difficulty signals—such as reasoning depth or context length—rather than domain-specific artifacts, enabling robust transfer to unseen distributions.

5 Experiments

5.1 Experiments setup

Benchmarks. We evaluate routers on six representative benchmarks. For training and in-domain evaluation, we use *Alpaca* (Taori et al., 2023) (general tasks), *MMLU* (Hendrycks et al., 2021a) (knowledge), and *Big-Math* (Albalak et al., 2025) (math). For out-of-domain evaluation, we use *Magpie* (Xu et al., 2025) (general tasks), *MMLU Pro* (Wang et al., 2024) (knowledge, covering STEM, Humanities, Social Sciences, and Others), and *MATH* (Hendrycks et al., 2021b) (math). The benchmark design is guided by three principles. Task coverage is ensured by including general, knowledge, and math domains. The difficulty gradient is reflected in the progredanssion from simpler benchmarks such as *Alpaca*,*Magpie*, to more challenging ones like *MMLU*, *Big-Math*, and *MATH*. Detailed data preparation and specific evaluation protocols are provided in Appendix B.

For model selection, we use *GPT-5* as the large model and *Llama-3.1-8B Instruct* as the small model for evaluating router performance.

Baselines. We compare our hidden-state approach against three alternative signal modalities:

Verbose-based. Routers that depend on auxiliary generations, such as self-evaluation (Kadavath et al., 2022; Ding et al., 2025) or semantic entropy (Kuhn et al., 2023b; Zhang et al., 2025a), which are informative but incur prompt sensitivity.

Logit-based. Routers that only use the final-layer logits, such as entropy (Su et al., 2025), margin (Ramírez et al., 2024). These are efficient but brittle across domains.

Embedding-based. These routers use fixed pretrained encoders with lightweight classifiers for semantic representations (Feng et al., 2025). With comparable classifier sizes, this enables direct comparison of different routing signals.

By categorizing baselines via their signal sources, we can facilitate a systematic comparison of different signal modalities.

Training Setup. For all probe-based methods, we use a lightweight linear model with input dimension 4096, corresponding to the small model's hidden state size. All models are trained with a fixed random seed. Training proceeds for 50 epochs with a learning rate of 1×10^{-4}. The training data consists of 12K examples, combining MMLU, Big Math, and Alpaca with 4K samples each.

5.2 Main Results

Router Ability. Table 1 reports the overall routing accuracy across multiple benchmarks. Our hidden-state–based strategies achieve 16.68% relative improvement over the best baseline in both in-domain and out-of-distribution scenarios. Within our approaches, ProbeDirichlet achieves marginally higher performance than ProbeMean through learned distributional layer weights. However, both variants perform competitively, indicating that strong results stem primarily from the hidden-state signals themselves rather than the aggregation mechanism. These results demonstrate that signal provenance is crucial: internal representations encode task-model interactions that external features cannot capture.

Scenario Alignment. Our framework enables flexible scenario definition based on deployment needs. Table 2 demonstrates router performance across three scenarios: cost-sensitive (LPM at 25-30% call rate), balanced (MPM), and accuracy-critical (HCR at 85-95% relative performance).

Probe-based methods outperform all baselines, especially in accuracy-critical scenarios. In cost-

Table 2: Scenario alignment ability of routing strategies across multiple benchmarks.

Method	In Domain				Out of Domain						
	Alpaca	Big Math	MMLU	AVG	Magpie	MATH	STEM	Humanities	Social Sciences	Others	AVG
LPM (Low Performance Mean)											
SelfAsk	**76.52**	74.10	77.52	76.05	63.35	61.46	57.01	50.58	**59.20**	59.99	58.60
SemanticEntropy	76.49	74.82	75.90	75.74	63.08	61.63	57.15	49.42	57.40	59.85	58.09
ConfidenceMargin	76.37	76.18	75.70	76.08	62.60	62.72	56.64	49.50	58.81	58.60	58.15
Entropy	76.16	75.32	75.29	75.59	63.08	63.81	55.58	50.77	57.10	59.18	58.25
MaxLogits	75.99	74.88	75.03	75.30	63.13	61.16	56.07	51.19	55.24	58.48	57.55
EmbeddingMLP	76.16	75.25	75.90	75.77	62.66	63.95	56.78	50.26	56.38	59.01	58.17
ProbeDirichlet	76.50	**78.82**	**78.51**	**77.95**	**63.53**	**69.24**	**59.20**	**51.74**	59.12	**62.42**	**60.88**
MPM (Middle Performance Mean)											
SelfAsk	**82.04**	81.40	83.92	82.45	**71.91**	75.34	69.47	62.94	**69.66**	70.41	69.95
SemanticEntropy	81.88	82.07	82.44	82.13	70.84	76.24	69.64	61.64	67.71	70.10	69.36
ConfidenceMargin	81.84	83.01	82.34	82.39	71.34	77.26	69.47	61.87	68.36	69.20	69.58
Entropy	81.74	82.04	82.25	82.01	71.65	77.84	68.79	63.01	67.69	69.81	69.80
MaxLogits	81.60	82.12	81.61	81.78	71.63	76.63	68.43	62.15	66.13	69.11	69.01
EmbeddingMLP	81.93	82.51	82.61	82.35	71.63	78.18	69.29	62.53	67.15	69.60	69.73
ProbeDirichlet	81.96	**84.67**	**84.31**	**83.65**	71.77	**81.45**	**71.06**	**64.51**	69.16	**71.73**	**71.61**
HCR (High-band Call Rate)											
SelfAsk	10.50	6.00	12.50	9.67	13.50	11.50	13.64	10.75	11.00	11.83	12.04
SemanticEntropy	**14.00**	16.00	15.50	15.17	**14.50**	13.00	**16.00**	10.75	13.33	12.50	13.35
ConfidenceMargin	9.50	14.00	10.00	11.17	9.50	10.00	12.50	**12.25**	11.68	8.17	10.68
Entropy	11.50	8.50	9.00	9.67	11.00	12.50	8.83	9.23	10.50	10.50	10.43
MaxLogits	10.00	10.00	8.00	9.33	11.00	10.00	10.17	9.25	7.50	8.33	9.38
EmbeddingMLP	10.00	15.50	10.00	11.83	9.00	13.50	11.42	9.25	9.67	11.50	10.72
ProbeDirichlet	13.50	**21.00**	**21.00**	**18.50**	**14.50**	**21.00**	15.75	11.50	**14.83**	**14.83**	**15.40**

sensitive and balanced regimes, performance differences remain modest because routers only need to escalate obviously difficult queries—a task most signal types handle adequately. However, accuracy-critical scenarios require precise identification of boundary cases where small models approach but do not meet requirements. Here, probe-based methods achieve 18.86% relative improvement, demonstrating that fine-grained difficulty discrimination requires richer internal signals.

5.3 Ablation Study

Table 3 compares three probe aggregation strategies: *Final* uses only the last layer, *Mean* uniformly averages all layers, and *Dirichlet* is our proposed method.Results show that our Dirichlet-based approach achieves the best average AUROC across all datasets.

Table 3: AUROC (%) of probe aggregation methods.

	Alpaca	Big-Math	MMLU	Average
Final Layer	61.97	50.33	49.45	53.91
Mean Pool	71.34	65.69	67.10	68.04
Dirichlet	72.02	66.18	67.88	68.70

Dirichlet achieves the best performance, and both aggregation methods significantly outperform the Final Layer baseline, confirming that cross-layer aggregation better captures task difficulty.

6 Analysis

Impact of Probe Architecture. To verify that lightweight architectures suffice, we compare a linear probe with a two-layer MLP under the mixed-dataset training setting.

Figure 3 compares one-hidden-layer MLPs with the linear baseline (dashed line). Introducing hidden layers provides almost no performance benefit but substantially increases overfitting, as evidenced by widening train-validation loss gaps. These results indicate that increasing model complexity is unnecessary for effective routing: a linear probe already achieves comparable or better performance, and introducing non-linearity or extra layers does not provide additional benefit.

Scaling Provides Diminishing Returns. We examine whether increasing training data improves probe performance by training on varying amounts of data from individual datasets. Table 4 shows that scaling from 1K to 10K samples yields maximum 3.89% improvement in AUROC, indicating that once probes capture sufficient signal, addi-

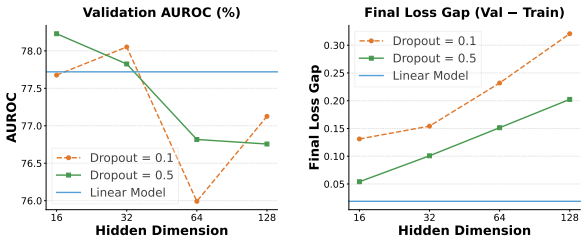

Figure 3: Effect of probe complexity on performance and generalization. The **horizontal line** represents the **Linear Probe baseline**, serving as a constant reference independent of the hidden dimension axis.

tional single-domain data provides minimal benefit. Given these diminishing returns, we ask: Does data diversity compensate for scale? The mixed-corpus probe matches specialized models, demonstrating diversity's effectiveness. To understand how diversity enables lightweight routing, we examine whether adding domains creates interference or yields additive gains.

Table 4: Validation AUROC (%) on respective held-out sets. Low (1K, Mixed 3K), Mid (4K, Mixed 12K), High (10K, Mixed 30K) denote training samples for MMLU/Big-Math/Alpaca.

Scale	MMLU	Big-Math	Alpaca	Mixed
Low	67.02	61.87	74.35	75.27
Mid	68.57	67.44	77.52	77.42
High	69.99	65.76	77.76	78.19

Data Diversity Yields Additive Gains Without Interference. We train on progressively larger data mixtures. Table 5 shows striking additive gains: existing performance is preserved (Alpaca: 71.85→71.96) while new domains contribute independently (BigMath: 49.19→66.49; MMLU: 49.35→66.00). This pattern explains why lightweight probes suffice. If domains conflicted, adding BigMath would degrade Alpaca. However, we observe no such interference; domains coexist harmoniously, suggesting hidden states encode a shared notion of difficulty that simple models can generalize across diverse tasks. Data diversity is additive, not competitive; diverse training improves robustness while preserving specialist capabilities.

Agent-based Inference Scenario. Beyond model collaboration, our router generalizes to agent-based inference, deciding when tool-augmented reasoning is needed. We evaluate it in

Table 5: Generalization Behavior under Different Dataset Compositions

Benchmark	Alpaca	Alpaca + BigMath	Mixed Training
In-domain			
Alpaca	71.85	71.63	**71.96**
BigMath	49.19	**66.49**	66.00
MMLU	49.35	51.06	**66.00**
Out-of-domain			
Magpie	72.80	**74.32**	72.49
MATH	57.97	**72.64**	71.80
MMLU-Pro	48.41	49.62	**59.66**

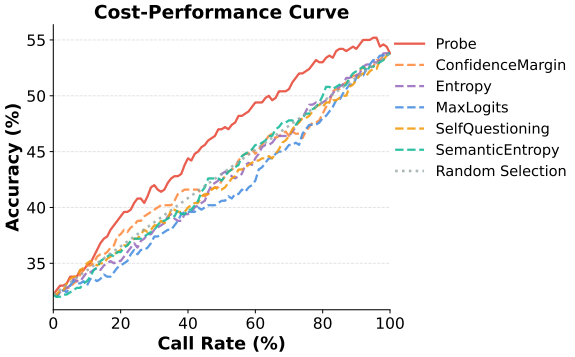

Figure 4: Cost-Performance curve under the agent-based inference scenario on HotpotQA.

HotpotQA, which requires multi-hop reasoning and iterative evidence retrieval. Figure 4 demonstrates robust generalization to agent scenarios. Our router shows a clear advantage across the entire cost-accuracy frontier.

7 Conclusion

We present a principled evaluation framework that disentangles intrinsic routing ability from scenario-specific requirements across three dimensions: router ability (AUROC), scenario alignment (LPM, MPM, HCR), and cross-domain robustness. This enables fair router comparison under diverse deployment constraints.

Then we introduce a lightweight hidden-state router that achieves strong generalization through multi-domain training. Our method outperforms baselines by 16.68% in router ability and 18.86% in high-accuracy scenarios, with consistent improvements across benchmarks and agentic workflows. Our analysis reveals that router robustness stems from training data diversity rather than architectural complexity, providing guidance for collaborative LLM systems.

Limitations

Our routing framework assumes the large model's capability exceeds the small model's; however, both models may perform similarly or converge on the same incorrect answer in certain domains (Appendix D.2), limiting routing effectiveness. Our experiments focus on a single small-large model pair and report single-run results due to computational constraints; broader validation across diverse architectures, multiple seeds, and more complex OOD conditions would further strengthen the conclusions.

References

Pranjal Aggarwal, Aman Madaan, Ankit Anand, Srividya Pranavi Potharaju, Swaroop Mishra, Pei Zhou, Aditya Gupta, Dheeraj Rajagopal, Karthik Kappaganthu, Yiming Yang, Shyam Upadhyay, Manaal Faruqui, and Mausam. 2024. AutoMix: Automatically Mixing Language Models.

Alon Albalak, Duy Phung, Nathan Lile, Rafael Rafailov, Kanishk Gandhi, Louis Castricato, Anikait Singh, Chase Blagden, Violet Xiang, Dakota Mahan, and Nick Haber. 2025. Big-math: A large-scale, high-quality math dataset for reinforcement learning in language models. *Preprint*, arXiv:2502.17387.

Amine Barrak, Yosr Fourati, Michael Olchawa, Emna Ksontini, and Khalil Zoghlami. 2025. Cargo: A framework for confidence-aware routing of large language models. *Preprint*, arXiv:2509.14899.

Felix Busch, Lena Hoffmann, Christopher Rueger, Elon H. C. van Dijk, Rawen Kader, Esteban Ortiz-Prado, Marcus R. Makowski, Luca Saba, Martin Hadamitzky, Jakob Nikolas Kather, Daniel Truhn, Renato Cuocolo, Lisa C. Adams, and Keno K. Bressem. 2025. Current applications and challenges in large language models for patient care: A systematic review. *Communications Medicine*, 5(1):26.

Tianle Cai, Yuhong Li, Zhengyang Geng, Hongwu Peng, Jason D. Lee, Deming Chen, and Tri Dao. 2024. Medusa: Simple llm inference acceleration framework with multiple decoding heads. *Preprint*, arXiv:2401.10774.

Chao Chen, Kai Liu, Ze Chen, Yi Gu, Yue Wu, Mingyuan Tao, Zhihang Fu, and Jieping Ye. 2024a. INSIDE: LLMs' internal states retain the power of hallucination detection. In *The Twelfth International Conference on Learning Representations*.

Charlie Chen, Sebastian Borgeaud, Geoffrey Irving, Jean-Baptiste Lespiau, Laurent Sifre, and John Jumper. 2023. Accelerating large language model decoding with speculative sampling. *Preprint*, arXiv:2302.01318.

Lingjiao Chen, Matei Zaharia, and James Zou. 2024b. FrugalGPT: How to use large language models while reducing cost and improving performance. *Transactions on Machine Learning Research*.

Shuhao Chen, Weisen Jiang, Baijiong Lin, James Kwok, and Yu Zhang. 2024c. RouterDC: Query-based router by dual contrastive learning for assembling large language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Dujian Ding, Ankur Mallick, Chi Wang, Robert Sim, Subhabrata Mukherjee, Victor Ruhle, Laks V. S. Lakshmanan, and Ahmed Hassan Awadallah. 2024a. Hybrid LLM: Cost-Efficient and Quality-Aware Query Routing. *Preprint*, arXiv:2404.14618.

Dujian Ding, Ankur Mallick, Chi Wang, Robert Sim, Subhabrata Mukherjee, Victor Ruhle, Laks V. S. Lakshmanan, and Ahmed Hassan Awadallah. 2024b. Hybrid llm: Cost-efficient and quality-aware query routing. *Preprint*, arXiv:2404.14618.

Dujian Ding, Ankur Mallick, Shaokun Zhang, Chi Wang, Daniel Madrigal, Mirian Del Carmen Hipolito Garcia, Menglin Xia, Laks V. S. Lakshmanan, Qingyun Wu, and Victor Rühle. 2025. BEST-route: Adaptive LLM routing with test-time optimal compute. In *Forty-second International Conference on Machine Learning*.

Zhichen Dong, Zhanhui Zhou, Zhixuan Liu, Chao Yang, and Chaochao Lu. 2025. Emergent response planning in LLMs. In *Proceedings of the 42nd International Conference on Machine Learning*. PMLR.

Jinhao Duan, Hao Cheng, Shiqi Wang, Alex Zavalny, Chenan Wang, Renjing Xu, Bhavya Kailkhura, and Kaidi Xu. 2024. Shifting attention to relevance: Towards the uncertainty estimation of large language models.

Ekaterina Fadeeva, Aleksandr Rubashevskii, Artem Shelmanov, Sergey Petrakov, Haonan Li, Hamdy Mubarak, Evgenii Tsymbalov, Gleb Kuzmin, Alexander Panchenko, Timothy Baldwin, Preslav Nakov, and Maxim Panov. 2024. Fact-checking the output of large language models via token-level uncertainty quantification. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 9367–9385, Bangkok, Thailand. Association for Computational Linguistics.

Tao Feng, Haozhen Zhang, Zijie Lei, Pengrui Han, Mostofa Patwary, Mohammad Shoeybi, Bryan Catanzaro, and Jiaxuan You. 2025. Fusionfactory: Fusing LLM capabilities with multi-LLM log data. *Preprint*, arXiv:2507.10540.

Marina Fomicheva, Shuo Sun, Lisa Yankovskaya, Frédéric Blain, Francisco Guzmán, Mark Fishel, Nikolaos Aletras, Vishrav Chaudhary, and Lucia Specia. 2020. Unsupervised quality estimation for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:539–555.

Neel Guha, Mayee F Chen, Trevor Chow, Ishan S. Khare, and Christopher Re. 2024. Smoothie: Label free language model routing. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. 2017. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1321–1330. PMLR.

Neha Gupta, Harikrishna Narasimhan, Wittawat Jitkrittum, Ankit Singh Rawat, Aditya Krishna Menon, and Sanjiv Kumar. 2024. Language model cascades: Token-level uncertainty and beyond. In *The Twelfth International Conference on Learning Representations*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021a. Measuring massive multitask language understanding. In *International Conference on Learning Representations (ICLR)*.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021b. Measuring mathematical problem solving with the math dataset. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Qitian Jason Hu, Jacob Bieker, Xiuyu Li, Nan Jiang, Benjamin Keigwin, Gaurav Ranganath, Kurt Keutzer, and Shriyash Kaustubh Upadhyay. 2024. Routerbench: A benchmark for multi-LLM routing system. In *Agentic Markets Workshop at ICML 2024*.

Zhongzhan Huang, Guoming Ling, Yupei Lin, Yandong Chen, Shanshan Zhong, Hefeng Wu, and Liang Lin. 2025. Routereval: A comprehensive benchmark for routing llms to explore model-level scaling up in llms. In *Findings of the Association for Computational Linguistics: EMNLP 2025*. Association for Computational Linguistics.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, Scott Johnston, Sheer El-Showk, Andy Jones, Nelson Elhage, Tristan Hume, Anna Chen, Yuntao Bai, Sam Bowman, Stanislav Fort, and 17 others. 2022. Language models (mostly) know what they know. *Preprint*, arXiv:2207.05221.

Aly M. Kassem, Bernhard Schölkopf, and Zhijing Jin. 2025. How robust are router-llms? analysis of the fragility of llm routing capabilities. *Preprint*, arXiv:2504.07113.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. 2023a. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. In *The Eleventh International Conference on Learning Representations*.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. 2023b. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. *Preprint*, arXiv:2302.09664.

Guohao Li, Hasan Abed Al Kader Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. 2023. CAMEL: Communicative agents for "mind" exploration of large language model society. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Yang Li. 2025. LLM bandit: Cost-efficient LLM generation via preference-conditioned dynamic routing.

Yuhui Li, Fangyun Wei, Chao Zhang, and Hongyang Zhang. 2024. Eagle-2: Faster inference of language models with dynamic draft trees. *Preprint*, arXiv:2406.16858.

Qiqi Lin, Xiaoyang Ji, Shengfang Zhai, Qingni Shen, Zhi Zhang, Yuejian Fang, and Yansong Gao. 2025. Life-cycle routing vulnerabilities of llm router. *Preprint*, arXiv:2503.08704.

Zhen Lin, Shubhendu Trivedi, and Jimeng Sun. 2024a. Contextualized sequence likelihood: Enhanced confidence scores for natural language generation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 10351–10368, Miami, Florida, USA. Association for Computational Linguistics.

Zhen Lin, Shubhendu Trivedi, and Jimeng Sun. 2024b. Generating with confidence: Uncertainty quantification for black-box large language models. *Transactions on Machine Learning Research*.

Andrea Matarazzo and Riccardo Torlone. 2025. A survey on large language models with some insights on their capabilities and limitations. *arXiv preprint arXiv:2501.04040*. Version 2.

Isaac Ong, Amjad Almahairi, Vincent Wu, Wei-Lin Chiang, Tianhao Wu, Joseph E. Gonzalez, M Waleed Kadous, and Ion Stoica. 2025. RouteLLM: Learning to route LLMs from preference data. In *The Thirteenth International Conference on Learning Representations*.

Xin Qiu and Risto Miikkulainen. 2024. Semantic density: Uncertainty quantification for large language models through confidence measurement in semantic space. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Guillem Ramírez, Alexandra Birch, and Ivan Titov. 2024. Optimising calls to large language models with uncertainty-based two-tier selection. In *First Conference on Language Modeling*.

Avital Shafran, Roei Schuster, Thomas Ristenpart, and Vitaly Shmatikov. 2025. Rerouting llm routers. *Preprint*, arXiv:2501.01818.

Jianshu She, Wenhao Zheng, Zhengzhong Liu, Hongyi Wang, Eric Xing, Huaxiu Yao, and Qirong Ho. 2025. Token level routing inference system for edge devices. *Preprint*, arXiv:2504.07878.

Gaurang Sriramanan, Siddhant Bharti, Vinu Sankar Sadasivan, Shoumik Saha, Priyatham Kattakinda, and Soheil Feizi. 2024. LLM-check: Investigating detection of hallucinations in large language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Dimitris Stripelis, Zijian Hu, Jipeng Zhang, Zhaozhuo Xu, Alay Dilipbhai Shah, Han Jin, Yuhang Yao, Salman Avestimehr, and Chaoyang He. 2024. Tensoropera router: A multi-model router for efficient llm inference. *Preprint*, arXiv:2408.12320.

Jiayuan Su, Fulin Lin, Zhaopeng Feng, Han Zheng, Teng Wang, Zhenyu Xiao, Xinlong Zhao, Zuozhu Liu, Lu Cheng, and Hongwei Wang. 2025. Cp-router: An uncertainty-aware router between llm and lrm. *Preprint*, arXiv:2505.19970.

Qi Sun, Marc Pickett, Ashish Kumar Nain, and Luke Jones. 2025. Transformer layers as painters. https://arxiv.org/abs/2407.09298.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca.

Song Wang, Zhen Tan, Zihan Chen, Shuang Zhou, Tianlong Chen, and Jundong Li. 2025. Anymac: Cascading flexible multi-agent collaboration via next-agent prediction. *Preprint*, arXiv:2506.17784.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, and 1 others. 2024. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, Ahmed Hassan Awadallah, Ryen W White, Doug Burger, and Chi Wang. 2024. Autogen: Enabling next-gen LLM applications via multi-agent conversations. In *First Conference on Language Modeling*.

Zhangchen Xu, Fengqing Jiang, Luyao Niu, Yuntian Deng, Radha Poovendran, Yejin Choi, and Bill Yuchen Lin. 2025. Magpie: Alignment data synthesis from scratch by prompting aligned llms with nothing. In *International Conference on Learning Representations (ICLR)*.

Shibo Yu, Mohammad Goudarzi, and Adel Nadjaran Toosi. 2025. Efficient routing of inference requests across llm instances in cloud-edge computing. *Preprint*, arXiv:2507.15553.

Tuo Zhang, Asal Mehradfar, Dimitrios Dimitriadis, and Salman Avestimehr. 2025a. Leveraging uncertainty estimation for efficient llm routing. *Preprint*, arXiv:2502.11021.

Yi-Kai Zhang, De-Chuan Zhan, and Han-Jia Ye. 2025b. Capability instruction tuning: A new paradigm for dynamic llm routing. *Preprint*, arXiv:2502.17282.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, and 3 others. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*. Version 16, last revised March 2025.

Zesen Zhao, Shuowei Jin, and Z. Morley Mao. 2024. Eagle: Efficient training-free router for multi-llm inference. *Preprint*, arXiv:2409.15518.

A Implementation Details

All experiments are conducted with a fixed random seed (seed=42) to ensure reproducibility. Due to computational constraints, we report single-run results for all experiments.

B Benchmark Datasets

We utilize six datasets spanning general instruction following, reasoning, and domain-specific knowledge. Table 6 summarizes the statistics of each dataset.

- **In-Domain:** We use *Alpaca* (Taori et al., 2023) for general instruction tuning. For knowledge-intensive tasks, we incorporate *MMLU* (Hendrycks et al., 2021a). Mathematical reasoning capabilities are represented by *Big-Math* (Albalak et al., 2025) .
- **Out-of-Domain:** To evaluate generalization, we employ *Magpie* (Xu et al., 2025) for aligned dialogue scenarios. For complex knowledge evaluation, we use *MMLU Pro* (Wang et al., 2024), which extends MMLU with harder distractors and broader subject coverage. *MATH* (Hendrycks et al., 2021b) is used to assess advanced problem-solving skills not covered in the training distribution.

Table 6: Benchmark statistics for router training and evaluation.

Dataset	Domain	Train/Val	Test
Alpaca	General	3.2K/0.8K	1K
MMLU	Knowledge	3.2K/0.8K	10K
Big Math	Math	3.2K/0.8K	1K
Magpie	General	—	10K
MMLU-Pro	Knowledge	—	12K
MATH	Math	—	5K

B.1 Ground Truth Label Construction

Exact Reasoning Tasks. For tasks requiring precise reasoning or factual correctness, rule-based string matching is often brittle due to format variations. To ensure robust evaluation, we leverage **xVerify**,[1] a specialized open-source verification framework, specifically the `xVerify-9B-C` model. Given the query and the small model's response, xVerify performs semantic parsing and verification against the ground truth, outputting a hard binary correctness label:

$$y = \text{xVerify}(q, r_{\text{small}}, a_{\text{gold}}), \tag{16}$$

where $y = 1$ indicates correctness (no routing needed) and $y = 0$ indicates failure (route to large model).

Open-ended Generation Tasks. For instruction-following tasks without unique answers, we use GPT-5 as an LLM-as-a-Judge evaluator[2] to score responses from 0 to 10. For each query q, we compare the small model's score S_{small} against the SOTA score S_{sota} (prompt in Figure 5):

$$y = \mathbb{1}(S_{\text{small}} \geq S_{\text{sota}}). \tag{17}$$

This yields $y = 1$ (no routing needed) when the small model performs comparably, and $y = 0$ (route to large model) otherwise.

Example Query

System Prompt: You are a helpful assistant.
Instruction: Please act as an impartial judge and evaluate the quality of the response provided by an AI assistant to the user question displayed below. Your evaluation should consider factors such as the helpfulness, relevance, accuracy, depth, creativity, and level of detail of the response. Begin your evaluation by providing a short explanation. Be as objective as possible. After providing your explanation, you must rate the response on a scale of 1 to 10 by strictly following this format: "[[rating]]", for example: "Rating: [[5]]".

[Question]
{question}
[The Start of Assistant's Answer]
{answer}
[The End of Assistant's Answer]

Figure 5: The prompt template used for LLM-as-a-Judge evaluation on open-ended generation tasks (e.g., AlpacaEval, Magpie). Both the small model and the SOTA proxy model responses are scored using this template to construct the relative ground truth labels.

[1] https://github.com/IAAR-Shanghai/xVerify

[2] As a proxy for SOTA performance. GPT-5 also serves as our large model; as judge, it blindly scores all responses without knowledge of their source.

C Pseudocode for ProbeDirichlet

Algorithm 1 ProbeDirichlet

procedure FORWARD($H \in \mathbb{R}^{B\times L\times D}$, return_uncertainty)
  if probe_type = "softmax" **then**
    $w = \text{softmax}(\theta_w)$ ▷ Fixed layer weights
    $F = \sum_{l=1}^{L} H[:, l, :] \cdot w[l]$
    return Linear(F), None
  else if probe_type = "dirichlet" **then**
    $\alpha = e^{\beta_0} \cdot \text{softmax}(\theta_\alpha)$ ▷ Concentration params
    if training **then**
      $w \sim \text{Dirichlet}(\alpha)$ ▷ Sample weights
      $u = -\sum_l w_l \log w_l$ ▷ Entropy uncertainty
    else
      $w = \alpha / \sum_l \alpha_l$ ▷ Expected weights
      $u = \log(\sum_l \alpha_l)$ ▷ Total concentration
    end if
    $F = \sum_{l=1}^{L} H[:, l, :] \cdot w[:, l, :]$
    return Linear(F), u
  end if
end procedure

Mean Pooling:

$$\hat{z}(x) = \frac{1}{L}\sum_{l=1}^{L} z^{(l)}(x)$$

D Supplemental Experiments

D.1 Layer Importance Analysis

To understand how training data affects layer importance, we visualize the normalized layer concentration for Llama-3.1-8b-Instruct in Figure 6. Across all training datasets, deeper layers show higher concentration, with the mixed dataset exhibiting the most pronounced pattern. Combined with our earlier analysis on data diversity, this suggests that deeper layers encode stronger signals about the model's capability to answer a given query, making them particularly informative for routing decisions.

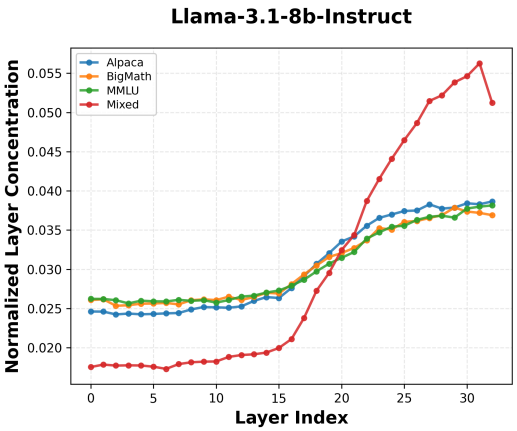

Figure 6: Normalized layer concentration across different training datasets. Deeper layers show higher importance, especially for mixed data.

D.2 When Routing is Not Enough: A Case Study

To illustrate both the effectiveness and limitations of routing systems, we analyze queries where our router correctly identified difficulty but the strong model still failed. Consider the following example:

Example Query

Query: This biome has cold winters and is known for its pine forests.
Options: A. Tundra B. Rainforest C. Grassland D. Chaparral E. Savanna F. Alpine G. Wetland H. Deciduous forests I. Desert J. Taiga
Small Model: J
Large Model: J
Ground Truth: H

In such cases, routing becomes ineffective: both models converge on the same incorrect answer, making it futile whether the system routes to save cost or to seek quality.This reveals critical gaps in current routing frameworks. When both models fail on the same query, the system faces a fundamental choice: it can route to the small model to save cost, but this delivers incorrect results that may mislead users; or route to the large model, which wastes resources without improving quality.

Addressing this requires two complementary strategies. The model pool should include more capable or specialized alternatives to handle queries where current models fail. Equally important, routing frameworks must incorporate uncertainty-aware mechanisms to detect when no available model is confident in these cases, the system should explicitly communicate uncertainty to users, rather than defaulting to the small model to save cost while silently delivering incorrect results.