LENS: Latent Precision Inference in Multi-LLM Routing

Juntao Liu[1,2] **Lixing Yu**[1,2] **Kun Yue**[1,2] **Zhiwen Tang**[*1,2]

[1]School of Information Science and Engineering, Yunnan University, Kunming,China
[2] Yunnan Key Laboratory of Intelligent Systems and Computing, Kunming, China

Abstract

Large language model (LLM) routing aims to select an appropriate model for each query under performance–cost trade-offs. A natural approach to improve adaptivity is to leverage interaction feedback to form behavioral signatures and continuously refine routing decisions as the environment changes. However, in realistic deployments, such feedback can have highly variable effective precision due to selective logging, imperfect evaluation signals, and temporal drift. As a result, behavioral signatures may be noisy or weakly informative, and treating them as uniformly precise can miscalibrate performance estimation and destabilize routing effectiveness. We propose **L**atent pr**E**cisio**N** inference **S**ystem (**LENS**), a probabilistic routing framework that explicitly models the latent precision of interaction-derived signals. LENS formulates routing as posterior utility maximization under imprecise supervision, and marginalizes over latent precision to adaptively control how strongly behavioral signatures influence model selection. We instantiate LENS with an efficient variational inference procedure and evaluate it on multi-LLM routing benchmarks across diverse tasks and distribution shifts. Experiments show that LENS consistently improves performance–cost trade-offs, with particularly strong gains under task or model shifts.

1 INTRODUCTION

The rapid expansion of large language models (LLMs), spanning proprietary systems such as GPT-4 and Claude and a rapidly growing ecosystem of open-source alternatives, has fundamentally reshaped the deployment landscape of intelligent services [Liu et al., 2025, Yang et al., 2025, Zhao et al., 2025, Grattafiori et al., 2024, Liu et al., 2024]. Modern AI platforms increasingly provide access to multiple models within a unified service interface, enabling flexible trade-offs among quality, latency, and cost. In enterprise copilots, retrieval-augmented generation pipelines, and API-based LLM services, each incoming query must be routed to an appropriate model under strict resource constraints. High-capacity models provide strong reasoning ability but incur substantial monetary and computational overhead, whereas lightweight models are economical yet prone to failure on complex or unfamiliar tasks. Effective routing has therefore become a foundational systems problem for scalable multi-LLM deployment.

Existing routing approaches typically rely on structured capability estimation. Methods such as IRT-Router [Song et al., 2025] quantify latent model ability and query difficulty using benchmark-derived statistics, while GraphRouter [Feng et al., 2024] leverages graph representations to predict query–model compatibility. These methods demonstrate that structured modeling improves routing over heuristic cascades. However, they predominantly assume that model competence can be characterized through static or pre-collected profiles. In practice, LLM ecosystems are highly dynamic. New models are introduced continuously, deployment contexts shift across domains, and task distributions evolve over time. Static profiling is often unavailable for newly released models and may become misaligned under distribution shift. Routing decisions based solely on static capability representations therefore lack robustness in non-stationary environments.

To compensate for the limitations of static profiling, modern systems incorporate runtime interaction signals, including historical response logs and automated evaluation metrics. Inspired by feedback-driven adaptation paradigms in related areas [Shinn et al., 2023, Madaan et al., 2023], we seek to endow routing mechanisms with the ability to adjust under environmental change. Such signals provide deployment-specific behavioral evidence, but their reliability is inherently heterogeneous. Interaction data are subject to selective

*Corresponding author.

Proceedings of the 42nd Conference on Uncertainty in Artificial Intelligence (UAI 2026), PMLR 337:3933–3956.

logging, imperfect or proxy evaluation, temporal drift, and changes in prompt distributions. Consequently, identical volumes of interaction records may carry substantially different predictive value across tasks and models. At the same time, routing systems must adapt to newly introduced models with limited history and to emerging task distributions unseen during profiling. These factors jointly create a structural challenge for routing in dynamic environments.

A robust routing mechanism must therefore regulate how behavioral evidence influences model selection. We use the term *precision* to denote the effective reliability of behavioral signals in predicting query-specific performance. Precision captures the degree to which interaction evidence should contribute to routing decisions relative to more stable capability information. When interaction patterns are stable and informative, behavioral signals exert stronger influence; when they are sparse or shifting, their contribution is moderated. This adaptive regulation enables routing decisions to remain calibrated under selective logging, imperfect evaluation, and distribution shift.

We propose **L**atent pr**E**cisio**N** inference **S**ystem (**LENS**), a precision-aware routing framework for adaptive multi-LLM systems. LENS constructs a latent-modulated representation that fuses static capability profiles with dynamic behavioral signatures in a query–model specific manner. A latent precision variable governs the contribution of behavioral evidence in the predictive representation, enabling routing decisions to interpolate smoothly between prior capability knowledge and recent interaction signals. This design supports stable model selection under selective logging and temporal drift, while also enabling rapid adaptation to new tasks and seamless integration of newly introduced models even with limited interaction history.

Formally, LENS formulates routing as posterior expected utility maximization under adaptive precision modulation. Posterior predictive performance is estimated through precision-aware representation fusion, and routing decisions balance predicted utility against resource cost. An amortized inference mechanism infers the latent precision variable from contextual features, providing scalable adaptation without per-query retraining. By explicitly modeling evidence reliability within the routing objective, LENS establishes a principled foundation for robust multi-model selection in evolving LLM ecosystems.

We evaluate LENS on large-scale multi-LLM routing benchmarks covering diverse tasks, model families, and distribution shifts. Experimental results demonstrate consistent improvements in routing accuracy and cost efficiency, with particularly strong gains under task shift and model shift scenarios. These findings indicate that adaptive precision modulation is essential for stable routing in dynamic and non-stationary environments.

Our contributions are summarized as follows.

- We formalize evidence precision as the effective reliability of behavioral signals in multi-LLM routing and identify its regulation as a core challenge under selective logging and distribution shift.
- We propose a precision-aware probabilistic routing framework that integrates static capability and dynamic behavior through latent precision inference and posterior utility maximization.
- We demonstrate strong generalization to new tasks and newly introduced models, enabling stable performance–cost trade-offs in dynamic and non-stationary deployment environments.

2 RELATED WORK

2.1 LLM ROUTING

Existing LLM routing methods primarily focus on structured capability estimation or cost-aware model selection. Early systems such as *FrugalGPT* [Chen et al., 2024], *HybridLLM* [Ding et al., 2024], and *RouteLLM* [Ong et al., 2025] leverage cached performance statistics or model identifiers to balance quality and cost. More recent approaches introduce transferable capability representations. *GraphRouter* [Feng et al., 2024] constructs heterogeneous graphs over queries and models, while *IRT-Router* [Song et al., 2025] models latent ability and difficulty through Item Response Theory. Additionally, reinforcement learning and online-learning routers (e.g., *Router-R1* [Zhang et al., 2025]) formulate routing as a multi-round sequential decision process, while OOD-oriented routers like *ProxRouter* [Patel et al., 2026] apply nonparametric aggregation to improve robustness against outlier queries. These methods demonstrate that structured modeling improves routing compared with heuristic cascades.

However, most existing routing frameworks rely predominantly on static or offline-aggregated capability profiles. In dynamic environments where interaction data are subject to selective logging, imperfect evaluation, and temporal drift [Zhu et al., 2024, Chen et al., 2023, Pu et al., 2025], static representations may become misaligned with online performance. Moreover, current methods lack an explicit mechanism to regulate the influence of dynamic behavioral signals under distribution shift or limited history. While RL-based methods[Zhang et al., 2025] excel in multi-round sequential settings and OOD-oriented designs enhance global out-of-distribution robustness[Feng et al., 2024, Patel et al., 2026], they do not explicitly model the uncertainty of local behavioral feedback for single-step cost-performance routing. LENS addresses this gap by introducing a precision-aware representation that adaptively integrates static capability and dynamic evidence in a query-dependent manner via latent-precision-based uncertainty modulation. This makes LENS complementary to RL and online routing methods.

2.2 LATENT VARIABLE MODELING

Latent variable models provide a principled way to handle heterogeneous or partially reliable observations by introducing hidden variables whose posterior inference regulates downstream prediction [Dawid and Skene, 1979, Raykar et al., 2010, Goldberger and Ben-Reuven, 2016]. Classical EM-style approaches and probabilistic graphical models estimate latent states jointly with model parameters, enabling observations to contribute according to inferred reliability [Raykar et al., 2010, Xiao et al., 2015]. Modern treatments generalize this idea under broader probabilistic learning settings [Song et al., 2023].

Variational inference enables scalable approximation of such latent posteriors through ELBO optimization and amortized inference networks. Representative examples such as Mult-VAE [Liang et al., 2018] illustrate how posterior regularization prevents overfitting to spurious patterns in heterogeneous observations. LENS adopts this principle in the routing setting by introducing a latent precision variable that modulates behavioral influence through posterior marginalization, rather than deterministically weighting interaction signals.

3 PRELIMINARY

3.1 PROBLEM SETTING

We consider cost-aware routing in a multi-LLM system [Chen et al., 2024, Ong et al., 2025]. Let $\mathcal{M} = \{M_j\}_{j=1}^{K}$ denote a set of candidate language models, and let q_i denote an incoming user query. Upon receiving q_i, the router selects a model M_j to generate a response.

Routing q_i to M_j yields a stochastic performance outcome $s_{ij} \in [0, 1]$, reflecting the quality of the generated response, and incurs a routing cost c_j. The cost term can aggregate different deployment resources, such as token price, inference latency, or computational overhead. For simplicity, we treat it as a known per-invocation cost that depends only on the selected model rather than on the input or output length. The routing objective is to maximize the expected utility

$$j^* = \arg\max_{j \in [K]} \ \mathbb{E}[U(s_{ij}, c_j) \mid q_i, \mathcal{O}_j], \tag{1}$$

where $\mathcal{O}_j$ denotes all information available about model M_j at routing time, and $U(s, c) = \alpha s - \beta c$ balances performance and cost with $\alpha \geq 0$ and $\beta \geq 0$.

In practical systems, the available observations $\mathcal{O}_j$ are derived from multiple information sources. Specifically, for each query–model pair (q_i, M_j), the router has access to (i) the semantic representation of the query q_i, (ii) a capability profile $\mathcal{P}_j$, obtained *offline* by probing M_j on a small set of representative benchmark queries and summarizing the resulting interaction traces into a compact descriptor via an auxiliary LLM-based summarization step [Feng et al., 2024, Song et al., 2025], and (iii) a behavioral signature $\mathcal{I}_j$, produced *online* by summarizing recent deployment interactions and evaluation signals into a query-conditioned descriptor using the same auxiliary LLM-based summarization step. $\mathcal{P}_j$ is query-independent and serves as a model-level prior descriptor, whereas $\mathcal{I}_j$ is query-dependent and reflects local behavioral patterns under the current query context and deployment state.

We denote the conditioning set by $\mathcal{X}_{ij} \triangleq \{q_i, \mathcal{P}_j, \mathcal{I}_j\}$. Let $p(s \mid \mathcal{X}_{ij})$ be the posterior predictive distribution of the performance outcome. Since c_j is deterministic, the conditional expectation in Eq. (1) can be written as

$$\mathbb{E}[U(s_{ij}, c_j) \mid \mathcal{X}_{ij}] = \int U(s, c_j)\, p(s \mid \mathcal{X}_{ij})\, ds. \tag{2}$$

For the linear utility $U(s, c) = \alpha s - \beta c$, this reduces to

$$\mathbb{E}[U(s_{ij}, c_j) \mid \mathcal{X}_{ij}] = \alpha \mathbb{E}[s_{ij} \mid \mathcal{X}_{ij}] - \beta c_j. \tag{3}$$

Accordingly, routing amounts to estimating the posterior predictive mean $\mathbb{E}[s_{ij} \mid \mathcal{X}_{ij}]$ for each candidate model, followed by a cost-aware selection.

3.2 LATENT PRECISION MODELING

Although behavioral signatures provide informative signals for characterizing recent model behavior, their reliability varies substantially across queries and time. Such signatures are derived from online interaction records, and are therefore subject to temporal drift [Gama et al., 2014] and selective observation [Li et al., 2010]. As a result, the effective uncertainty associated with $\mathcal{I}_j$ may differ considerably across query–model contexts. In contrast, capability profiles are constructed offline through controlled probing and aggregated summarization over multiple benchmark instances, which yields more stable and query-independent descriptors. We therefore treat $\mathcal{P}_j$ as a relatively reliable contextual prior and focus on modeling uncertainty arising from behavioral signatures.

To account for this variability, we introduce a latent variable z_{ij} representing the precision of behavioral signatures when predicting the performance of model M_j on query q_i. Intuitively, z_{ij} quantifies the reliability of $\mathcal{I}_j$ under the context $\mathcal{X}_{ij}$. The latent variable z_{ij} serves as a context-dependent modulation factor that controls how strongly the behavioral signature $\mathcal{I}_j$ influences performance prediction under $\mathcal{X}_{ij}$.

Based on this formulation, the posterior predictive distribution is obtained by marginalizing over the latent precision variable:

$$p(s_{ij} \mid \mathcal{X}_{ij}) = \int p(s_{ij} \mid \mathcal{X}_{ij}, z_{ij})\, p(z_{ij} \mid \mathcal{X}_{ij})\, dz_{ij}. \tag{4}$$

This latent-variable decomposition separates performance modeling conditioned on a given reliability level from the

posterior belief over the quality of behavioral abstractions. By integrating over z_{ij}, the framework prevents unreliable or outdated signatures from being deterministically propagated into routing decisions.

In the following sections, we introduce a variational inference framework for approximating the intractable posterior predictive distribution in Eq. (4) and for performing efficient, precision-aware routing.

4 METHODOLOGY

4.1 FRAMEWORK OVERVIEW

LENS performs precision-aware routing in multi-LLM systems by constructing latent-modulated representations and maximizing posterior expected utility. Given an incoming query q_i and a candidate model M_j, LENS forms the contextual input $\mathcal{X}_{ij} = \{q_i, \mathcal{P}_j, \mathcal{I}_j\}$ from the query representation, the capability profile, and the behavioral signature, and estimates the posterior predictive performance $p(s_{ij} \mid \mathcal{X}_{ij})$ for cost-aware model selection.

At the core of LENS is a precision-modulated representation that integrates offline capability information and online behavioral signals. The capability profile and behavioral signature are fused with the query representation through a latent-controlled interaction mechanism, producing a context-dependent descriptor for performance prediction. This representation serves as the basis for modeling query–model compatibility and downstream utility estimation.

To support adaptive modulation under uncertainty, LENS introduces a latent precision variable z_{ij} that governs the contribution of behavioral information in representation fusion. The posterior distribution of z_{ij} is inferred from $\mathcal{X}_{ij}$ using amortized variational inference, enabling uncertainty-aware performance estimation without explicit supervision.

All components are trained jointly from historical interaction data and deployed for online routing in a unified inference–decision pipeline, without requiring per-query optimization.

4.2 PRECISION-MODULATED REPRESENTATION FUSION

The core of LENS is a precision-modulated representation that integrates offline capability information and online behavioral signals for routing decisions. For each model M_j, the capability profile $\mathcal{P}_j$ is constructed offline by randomly sampling k benchmark queries, collecting the model's responses, and summarizing the resulting traces into a structured descriptor. This yields a query-independent characterization of model competence based on stochastic probing, rather than relying on exhaustive evaluation.

The behavioral signature $\mathcal{I}_j$ is maintained via an online cache. For each model, a task-aware feedback buffer summarizes every $B_{\mathcal{I}}$ newly observed query-response pairs into a new signature, while retaining a sliding window of at most $W_{\mathcal{I}}$ recent signatures. At inference time, the router retrieves up to $R_{\mathcal{I}}$ cached signatures most relevant to q_i via task-cluster matching.

Both $\mathcal{P}_j$ and $\mathcal{I}_j$ are mapped into a shared embedding space together with the incoming query q_i through an encoder $E(\cdot)$, resulting in vector representations $\mathbf{q}_i$, $\mathbf{p}_j$, and $\mathbf{i}_j$. To integrate offline capability information with online behavioral signals, LENS constructs a latent-conditioned fused representation modulated by the inferred precision variable. Given a realization $\hat{z}_{ij}$, we define

$$\mathbf{m}_{ij} = \mathbf{p}_j + \hat{z}_{ij} \operatorname{softmax}\left(\frac{\mathbf{p}_j \mathbf{i}_j^\top}{\sqrt{d}}\right) \mathbf{i}_j, \tag{5}$$

where the capability profile embedding $\mathbf{p}_j$ serves as a baseline descriptor and the behavioral embedding $\mathbf{i}_j$ is incorporated through an attention-based interaction whose contribution is adaptively modulated by the latent precision.

The fused representation $\mathbf{m}_{ij}$ is then used to model query–model compatibility. We compute the token-level similarity matrix

$$\mathbf{S}_{ij} = \frac{\mathbf{q}_i \, \mathbf{m}_{ij}^\top}{\sqrt{d}}, \tag{6}$$

and aggregate maximal alignments across query tokens,

$$\hat{s}_{ij} = \frac{1}{L_q} \sum_{t=1}^{L_q} \max_{1 \le \ell \le L_m} \mathbf{S}_{ij}[t, \ell], \tag{7}$$

which serves as a point estimate of the conditional performance outcome. L_q and L_m denote the lengths of the query and fused representations, respectively.

4.3 VARIATIONAL PRECISION INFERENCE

To infer the latent precision variable governing representation fusion, LENS employs an amortized variational inference framework to approximate the intractable posterior distribution $p(z_{ij} \mid \mathcal{X}_{ij})$. Given the nonlinear parameterization of the conditional performance model, exact Bayesian inference over the latent precision variable is computationally infeasible in practice.

Specifically, LENS introduces a parameterized variational distribution $q_\phi(z_{ij} \mid \mathcal{X}_{ij})$ conditioned on the contextual input $\mathcal{X}_{ij}$. The parameters of q_ϕ are produced by a neural inference network. A standard Gaussian prior $p(z_{ij}) = \mathcal{N}(0, 1)$ is imposed on z_{ij} to regularize the latent space and encourage stable posterior estimates.

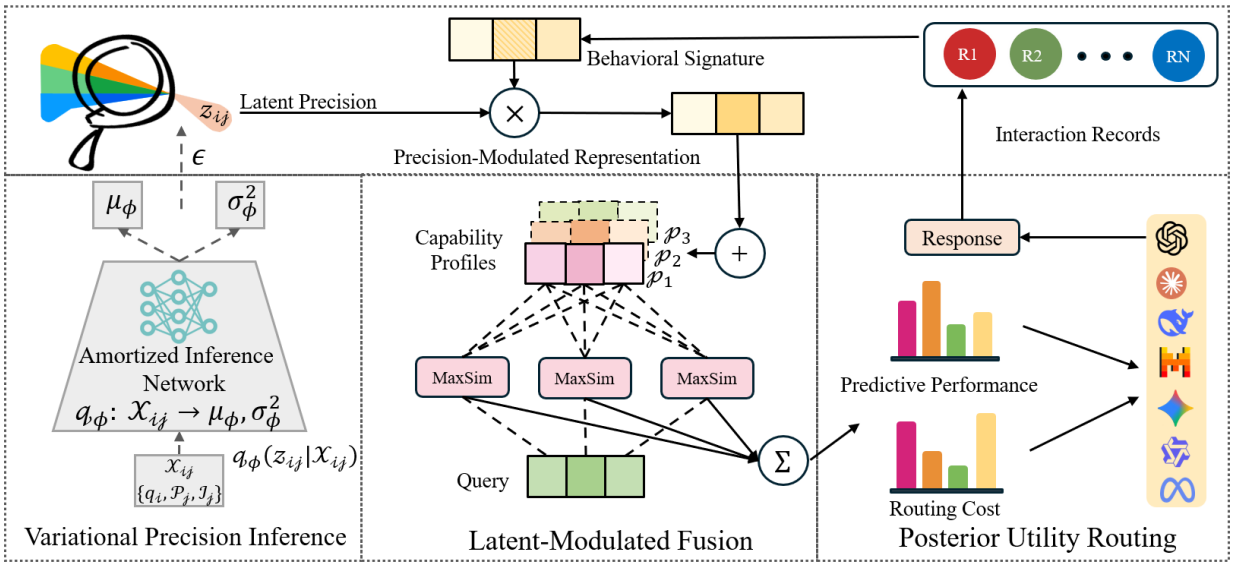


Figure 1: **Overview of the LENS framework.** LENS constructs precision-modulated representations by integrating capability profiles and behavioral signatures for performance prediction (middle), and performs posterior utility maximization for cost-aware model selection (right). A latent precision variable is inferred via amortized variational inference to support adaptive fusion and uncertainty-aware routing (left). Routing outcomes are incorporated into subsequent behavioral signatures for continuous adaptation.

To enable gradient-based optimization, LENS employs the reparameterization technique [Kingma and Welling, 2014] to allow differentiable sampling. For Gaussian variational families, samples from $q_\phi(z_{ij} \mid \mathcal{X}_{ij})$ are generated as

$$z_{ij} = \mu_\phi(\mathcal{X}_{ij}) + \sigma_\phi(\mathcal{X}_{ij})\,\epsilon, \quad \epsilon \sim \mathcal{N}(0,1). \tag{8}$$

where $\epsilon \sim \mathcal{N}(0,1)$ is auxiliary noise, and μ_ϕ, σ_ϕ are deterministic outputs of the inference network. Conditioned on z_{ij} and $\mathcal{X}_{ij}$, the precision-modulated representation constructed in Section 4.2 parameterizes the conditional performance model. The predictor outputs

$$\pi_{ij} = \sigma\big(f_\theta(z_{ij}, \mathcal{X}_{ij})\big), \tag{9}$$

where $f_\theta(\cdot)$ denotes the precision-aware predictor and $\pi_{ij} \in (0,1)$ is interpreted as the predicted normalized performance, or equivalently the success probability under a binary outcome. Since the observed performance score s_{ij} may be continuous in $[0,1]$, we use a Bernoulli-style soft-label likelihood, equivalent to the binary cross-entropy objective:

$$\log p_\theta(s_{ij} \mid z_{ij}, \mathcal{X}_{ij}) = s_{ij} \log \pi_{ij} + (1 - s_{ij}) \log(1 - \pi_{ij}). \tag{10}$$

When $s_{ij} \in \{0,1\}$, this reduces to the standard Bernoulli log-likelihood. For continuous normalized scores, it provides a smooth surrogate for fitting performance outcomes. Since routing decisions depend only on posterior expected performance and relative utility, this relaxation does not alter the model selection objective.

Model parameters are learned by maximizing the evidence lower bound (ELBO),

$$\begin{aligned}\mathcal{L}_{\text{ELBO}}^{(ij)} = \mathbb{E}_{q_\phi(z_{ij}|\mathcal{X}_{ij})}\big[\log p(s_{ij} \mid z_{ij}, \mathcal{X}_{ij})\big] \\ - D_{\text{KL}}\big(q_\phi(z_{ij} \mid \mathcal{X}_{ij}) \,\|\, p(z_{ij})\big),\end{aligned} \tag{11}$$

where the first expectation fits the observed routing performance, and the second term regularizes the posterior toward the Gaussian prior to prevent degenerate latent representations. This bound serves as the training objective for joint optimization.

At inference time, LENS approximates the marginal posterior predictive distribution in Eq. (4) by replacing the intractable posterior $p(z_{ij} \mid \mathcal{X}_{ij})$ with its variational approximation $q_\phi(z_{ij} \mid \mathcal{X}_{ij})$ and performing Monte Carlo integration,

$$\begin{aligned} p(s_{ij} \mid \mathcal{X}_{ij}) &= \int p_\theta(s_{ij} \mid z_{ij}, \mathcal{X}_{ij}) p(z_{ij} \mid \mathcal{X}_{ij})\, dz_{ij} \\ &\approx \int p_\theta(s_{ij} \mid z_{ij}, \mathcal{X}_{ij}) q_\phi(z_{ij} \mid \mathcal{X}_{ij})\, dz_{ij} \\ &\approx \frac{1}{N} \sum_{n=1}^{N} \pi_{ij}\left(z_{ij}^{(n)}\right), \end{aligned} \tag{12}$$

where $z_{ij}^{(n)} \sim q_\phi(z_{ij} \mid \mathcal{X}_{ij})$.

4.4 POSTERIOR UTILITY ROUTING

Given the posterior predictive performance estimate $\hat{s}_{ij}$ obtained from the variational inference procedure in Section 4.3, LENS performs routing by maximizing the posterior expected utility. Specifically, for an incoming query q_i, the selected model is determined as

$$j^* = \arg\max_{j \in [K]} \left(\alpha \hat{s}_{ij} - \beta c_j \right), \tag{13}$$

where the coefficients α and β control the trade-off between predicted performance and the chosen cost measure. c_j denotes the known deployment cost of invoking model M_j. The cost term is a generic scalar that can represent token price, expected latency, computational overhead, or their weighted combination. In this work, we instantiate c_j as a model-level per-invocation cost because current routing benchmarks provide standardized cost information at this level, but do not offer unified latency measurements across heterogeneous APIs and deployment environments. Once such latency-aware benchmarks become available, LENS can incorporate latency directly through the same cost term without changing the routing objective.

LENS is trained on historical interaction records $\{(q_i, M_j, s_{ij})\}$, where $s_{ij} \in \{0, 1\}$ denotes the observed performance outcome. The parameters of the precision-aware predictor and the inference network, denoted by θ and ϕ, are learned jointly by minimizing the negative evidence lower bound,

$$\mathcal{L}(\theta, \phi) = -\mathbb{E}_{(i,j)} \Big[\mathbb{E}_{q_\phi(z_{ij} | \mathcal{X}_{ij})} \big[\log p_\theta(s_{ij} \mid z_{ij}, \mathcal{X}_{ij}) \big] - \lambda D_{\mathrm{KL}} \big(q_\phi(z_{ij} \mid \mathcal{X}_{ij}) \parallel p(z_{ij}) \big) \Big]. \tag{14}$$

where the expectation over (i, j) is taken with respect to the empirical training distribution. The KL divergence term admits a closed-form expression under the Gaussian variational family and regularizes the latent precision towards the standard normal prior. The resulting objective is optimized end-to-end using stochastic gradient descent.

5 EXPERIMENTS

5.1 EXPERIMENT SETUP

Datasets We evaluate LENS on a routing testbed comprising 20 LLMs and 12 benchmarks spanning reasoning, coding, and knowledge-intensive tasks. The models cover diverse architectures (e.g., GPT-4o, Llama-3.1, Qwen2.5). Detailed dataset statistics and pricing information are provided in Appendix Tables 3 and 4.

To assess generalization, we partition models and datasets into disjoint training and held-out splits (17/3 models and 8/4 datasets). This induces four evaluation regimes: **In-Distribution**, where both models and datasets belong to the training split; **OOD-Task**, where models come from the training split while datasets are held out; **OOD-Model**, where datasets belong to the training split while models are held out; and **OOD-Combined**, where both models and datasets are drawn from the held-out split.

Baselines We compare LENS against five routing strategies: *Always-Cheapest*, which defaults to a designated lightweight or resource-conservative model as an economical baseline; *Always-Strongest*, defaults to a designated high-capacity model; *FrugalGPT* [Chen et al., 2024], a cascade-based router; *Graph-Router* [Feng et al., 2024], a GNN-based routing method; and *IRT-Router* [Song et al., 2025], a state-of-the-art approach based on Item Response Theory. To isolate the effect of the external auditor model, we instantiate IRT-Router using different auditors, i.e. ChatGPT and DeepSeek-V3.2.

Metrics Following prior work [Song et al., 2025, Feng et al., 2024], we evaluate routing using three metrics.

Performance is the average normalized task score. **Total Cost** is computed from actual token usage and API pricing, thus accounting for token length at evaluation time; in contrast, the routing model assumes a fixed per-invocation cost for each LLM during training. **Reward** captures the performance–cost trade-off:

$$R = \alpha \cdot \text{Performance} - \beta \cdot \widetilde{\text{Cost}}, \tag{15}$$

where $\widetilde{\text{Cost}}$ denotes normalized total cost. We report results under three trade-off regimes: **Performance-First** ($\alpha = 0.8, \beta = 0.2$), **Balanced** ($\alpha = 0.5, \beta = 0.5$), and **Cost-First** ($\alpha = 0.2, \beta = 0.8$).

Implementation details are provided in Appendix A.

5.2 MAIN RESULTS

Table 1 reports results across four settings. LENS maintains top-tier reward performance, with advantages becoming more pronounced under distribution shifts. Notably, LENS outperforms both IRT-Router variants across most settings, demonstrating that the performance gains stem primarily from the latent precision architecture rather than the choice of auditor model. While baselines degrade under task or model changes, LENS preserves stable performance–cost trade-offs through precision-aware capability modeling.

In the *In-Distribution* setting, LENS performs comparably to IRT-Router, indicating that when sufficient data are available, capability estimation can be well calibrated. However, under cost-sensitive optimization, LENS provides finer discrimination among similarly capable models, yielding more efficient resource allocation.

Under *OOD-Task* shifts, LENS maintains superior performance, suggesting that behavioral profiling captures trans-

Table 1: **Experiment Results across different Distribution Setting and Trade-off Regimes.**Best **Reward** is bolded and the second-best is underlined in each configuration.

Setting	Method	Performance-First			Balanced			Cost-First		
		Perf. ↑	Cost ↓	Reward ↑	Perf. ↑	Cost ↓	Reward ↑	Perf. ↑	Cost ↓	Reward ↑
In-Distribution	Always-Cheapest	48.70%	0.31	38.48	48.70%	0.31	23.14	48.70%	0.31	7.81
	Always-Strongest	77.53%	12.93	42.02	77.53%	12.93	-11.24	77.53%	12.93	-64.49
	FrugalGPT	56.68%	0.69	44.28	50.29%	0.41	23.56	49.22%	0.34	7.75
	GraphRouter	72.17%	0.49	56.97	72.08%	0.47	34.21	72.10%	0.43	11.72
	IRT-Router (ChatGPT)	80.70%	0.53	**63.73**	80.33%	0.45	38.41	79.32%	0.42	<u>13.27</u>
	IRT-Router (DeepSeek)	80.63%	0.60	<u>63.57</u>	80.44%	0.46	<u>38.44</u>	78.33%	0.42	13.07
	LENS	80.34%	0.55	63.42	80.29%	0.43	**38.46**	78.53%	0.37	**13.40**
OOD-Task	Always-Cheapest	59.83%	0.11	47.69	59.83%	0.11	29.49	59.83%	0.11	11.29
	Always-Strongest	84.90%	5.30	59.73	84.90%	5.30	21.97	84.90%	5.30	-15.79
	FrugalGPT	70.37%	0.32	55.80	62.01%	0.16	30.39	59.82%	0.11	11.28
	GraphRouter	74.01%	0.19	58.91	73.73%	0.15	36.29	73.66%	0.16	13.75
	IRT-Router (ChatGPT)	86.55%	0.16	<u>68.99</u>	86.55%	0.14	42.73	86.70%	0.13	<u>16.51</u>
	IRT-Router (DeepSeek)	86.45%	0.18	68.89	86.64%	0.15	<u>42.75</u>	85.85%	0.13	16.34
	LENS	87.08%	0.14	**69.45**	86.83%	0.14	**42.88**	87.55%	0.13	**16.73**
OOD-Model	Always-Cheapest	59.73%	8.34	34.88	59.73%	8.34	-2.39	59.73%	8.34	-39.66
	Always-Strongest	77.54%	10.28	**46.13**	77.54%	10.28	<u>-0.98</u>	77.54%	10.28	-48.08
	FrugalGPT	64.03%	10.52	34.97	60.91%	8.83	-3.68	59.86%	8.38	-39.88
	GraphRouter	59.72%	8.34	34.88	59.72%	8.34	-2.39	59.72%	8.34	-39.65
	IRT-Router (ChatGPT)	59.73%	8.34	34.89	59.91%	8.24	-1.90	63.47%	6.33	-26.43
	IRT-Router (DeepSeek)	59.73%	8.34	34.89	59.89%	8.27	-2.02	66.62%	3.84	<u>-10.41</u>
	LENS	61.39%	8.15	<u>36.51</u>	60.63%	7.15	**2.66**	72.06%	0.96	**8.47**
OOD-Combined	Always-Cheapest	76.30 %	2.35	57.40	76.30 %	2.35	29.07	76.30 %	2.35	0.73
	Always-Strongest	82.06 %	2.90	**61.16**	82.06 %	2.90	<u>29.81</u>	82.06 %	2.90	-1.54
	FrugalGPT	75.47%	3.01	55.72	76.19%	2.38	28.88	76.32%	2.35	0.74
	GraphRouter	76.32%	2.35	57.43	76.32%	2.35	29.08	76.32%	2.34	0.74
	IRT-Router (ChatGPT)	76.32%	2.35	57.43	76.32%	2.35	29.08	76.67%	2.12	2.25
	IRT-Router (DeepSeek)	76.29%	2.35	57.40	76.29%	2.35	29.06	77.84%	1.40	<u>6.90</u>
	LENS	78.66%	2.24	<u>59.46</u>	76.35%	2.07	**30.16**	71.19%	0.27	**12.55**

ferable capability structures beyond task-specific patterns. In the *OOD-Model* setting, where interaction history is limited, LENS remains effective by constructing dynamic behavioral profiles and modulating them through inferred precision, enabling model integration without relying on prior task-specific interactions.

Finally, in the *OOD-Combined* scenario with simultaneous task and model shifts, LENS continues to outperform competing methods, which tend to revert to conservative defaults. By jointly modeling static capability profiles and dynamic interaction evidence within a precision-aware framework, LENS supports coherent routing under compounded distribution shifts.

5.3 ANALYSIS ON LATENT PRECISION INFERENCE

Ablation Study We conduct an ablation study to examine the role of each component in LENS. *w/o Capability Profile* removes static model information; *w/o Behavioral Signature* excludes interaction-derived signals; *w/o Latent Precision* discards uncertainty modeling and directly fuses representations; and *Mean-only Precision* replaces posterior marginalization with a mean approximation. Full LENS integrates all components.

Across all four settings, full LENS consistently lies on or near the non-dominated frontier in the performance–cost space. Each ablation shifts the operating point away from this frontier, either sacrificing performance for lower cost or incurring higher cost for comparable performance. Removing static capability information primarily degrades accuracy, while excluding behavioral signals or latent precision inflates cost under distribution shifts. The mean-only approximation further weakens the trade-off, underscoring the importance of uncertainty-aware inference. Overall, only the integrated model maintains a balanced performance–cost profile across ID and OOD scenarios.

Sensitivity of Latent Precision Inference We analyze the sensitivity of LENS to the KL coefficient λ governing latent

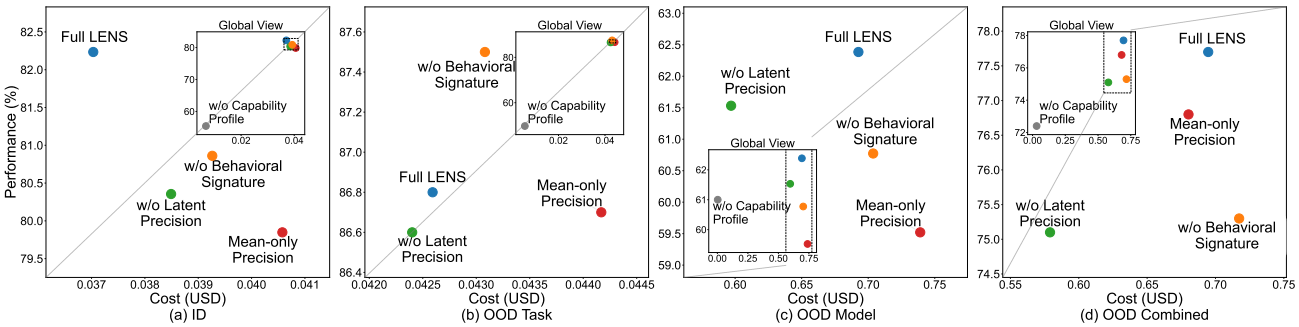


Figure 2: Ablation study on the core components of LENS.

Table 2: Case Study.

Task Query	Static Capability Profile	Behavioral Signature	Routing Decision
Math (Algebra) *"Express $0.1\overline{7}$ as a common fraction."*	**Llama-3.1-405B** "Strong STEM performer; fails on nuanced cultural tasks." **Result:** ✓ Success **Cost:** High ($0.0016)	*"Model reliably succeeds in tasks requiring direct, rule-based interpretation..."* → Suggests simpler model for rules.	**Gemini-1.5-Flash** (Selected via behavioral signature) **Result:** ✓ Success **Cost:** Low ($0.00007)
History (Knowledge) *"Based on the above material and historical context, ... Which of the following statements is the most appropriate?"*	**QwQ-32B** "Competent in logic, weak in factual recall." **Result:** ✗ **Failure** **Reward:** -5.47	*"Model reliably handles modern, unambiguous tasks with direct answers..."* → Corrects factual gap.	**Gemini-1.5-Flash** (Selected via behavioral signature) **Result:** ✓ **Success** **Reward:** +0.30

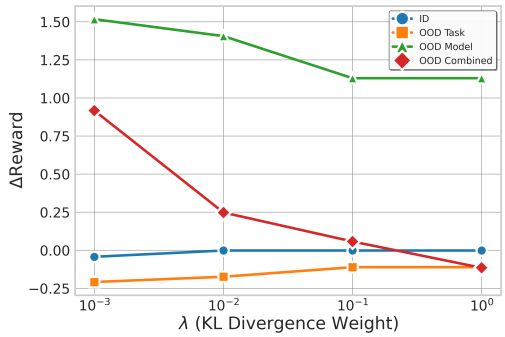


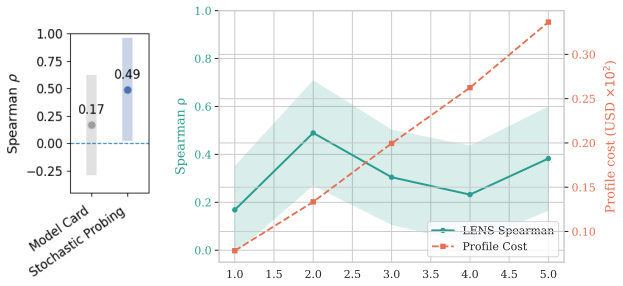
Figure 3: Sensitivity of λ on System Performance.

precision inference. The coefficient controls posterior contraction by weighting the divergence between the inferred precision distribution and its prior. We evaluate its impact by measuring Δ Reward relative to a base variant without behavioral signatures across a broad range of λ values.

Latent precision inference remains stable under substantial variation of λ. In ID, reward is largely insensitive to moderate changes, indicating well-conditioned posterior calibration. Under distribution shifts, particularly OOD-Model and OOD-Combined, smaller λ values tend to improve reward, suggesting that relaxed divergence weighting enhances adaptive uncertainty modeling. Overall, performance varies smoothly across orders of magnitude in λ, without abrupt degradation.

These results indicate that the latent precision mechanism is robust to coefficient selection and does not require delicate tuning to maintain competitive performance–cost trade-offs.

Interpretability of Latent Precision To examine the interpretability of the inferred precision z, we conducted a post-hoc analysis. The inferred z strongly correlates with heuristic reliability indicators, including interaction-history length ($\rho = 0.67$), signature length ($\rho = 0.66$), and profile-signature cosine similarity ($\rho = 0.66$), all with $p < 0.001$. Furthermore, perturbing z directly affects routing decisions: $\Delta z = \pm 0.2$ alters 8% to 9% of choices, while $\Delta z = \pm 0.8$ alters 17% to 25%. These results suggest that z successfully captures reliability-related variation and has a measurable, interpretable effect on routing without requiring explicit supervision.



(a) (b)

Figure 4: (a) **Spearman correlation between predicted and actual performance** using model cards versus stochastic probing ($k = 2$) for static profiling. Stochastic probing yields stronger rank alignment. (b) **Effect of sample size k in stochastic profiling.** The best trade-off is achieved at $k = 2$, balancing predictive accuracy and token usage.

5.4 ANALYSIS ON STATIC CAPABILITY PROFILE

We first compare two strategies for constructing static capability profiles: model-card–based descriptions and stochastic probing via random query sampling. Evaluating the Spearman correlation between predicted and actual performance shows that stochastic profiling produces substantially stronger rank alignment. This indicates that lightweight empirical probing provides a more faithful and discriminative characterization of model competence than static documentation alone.

We further analyze the effect of the probing sample size k. Varying k reveals that very small sample sizes are sufficient to capture stable capability signals. In particular, $k = 2$ achieves the best performance–cost trade-off, maintaining competitive predictive accuracy while minimizing token consumption. This suggests that stochastic capability profiling is both effective and economical, and does not require extensive probing to yield reliable model characterization.

5.5 CASE STUDY

As shown in Table 2, two examples illustrate how behavioral signatures complement static capability profiles in routing. In a procedural algebra query, the static capability profile favors a large STEM-oriented model, yielding correct results but at high cost. Incorporating behavioral signatures identifies the rule-based nature of the task and routes to a lightweight model, preserving accuracy while substantially reducing cost. In a historical knowledge query, the static profile selects a model characterized as logically competent but weak in factual recall, resulting in failure. The behavioral signature captures this mismatch and redirects to a more suitable model, correcting the error and improving reward. These cases demonstrate that integrating static capability profiles with context-specific behavioral signals enables more precise and interpretable routing decisions grounded in model–task interactions.

6 CONCLUSION

We introduce LENS, a precision-aware routing framework that combines static capability profiles with dynamic behavioral signatures via latent precision inference. By explicitly modeling evidence precision, LENS achieves stable and efficient performance–cost trade-offs without relying on task-specific heuristics. Experiments across in-distribution and distribution-shift settings demonstrate consistent robustness under task and model shifts.

ACKNOWLEDGMENTS

This work was supported by the Joint Key Project of National Natural Science Foundation of China (U23A20298), the Central Government Fund for Guiding Local Science and Technology Development (202607AD040003), the Key Project of Fundamental Research of Yunnan Province (202401AS070138), the Program of Yunnan Key Laboratory of Intelligent Systems and Computing (202549CE340006), the Yunnan Fundamental Research Project (202501AT070231), the Yunnan University Medical Research Foundation (K204209250001), and Yunnan University School of ISE Graduate Student Scientific Research Project(ISEG2026B09).

REFERENCES

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and Charles Sutton. Program synthesis with large language models, 2021. URL https://arxiv.org/abs/2108.07732.

Lingjiao Chen, Matei Zaharia, and James Zou. How is chatgpt's behavior changing over time?, 2023. URL https://arxiv.org/abs/2307.09009.

Lingjiao Chen, Matei Zaharia, and James Zou. FrugalGPT: How to use large language models while reducing cost and improving performance. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=cSimKw5p6R. Featured Certification.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code, 2021. URL https://arxiv.org/abs/2107.03374.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge, 2018. URL https://arxiv.org/abs/1803.05457.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems, 2021. URL https://arxiv.org/abs/2110.14168.

A. P. Dawid and A. M. Skene. Maximum likelihood estimation of observer error-rates using the em algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1):20–28, 1979. ISSN 00359254, 14679876. URL http://www.jstor.org/stable/2346806.

Dujian Ding, Ankur Mallick, Chi Wang, Robert Sim, Subhabrata Mukherjee, Victor Rühle, Laks V. S. Lakshmanan, and Ahmed Hassan Awadallah. Hybrid LLM: Cost-efficient and quality-aware query routing. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=02f3mUtqnM.

Tao Feng, Yanzhen Shen, and Jiaxuan You. Graphrouter: A graph-based router for llm selections. In *The Thirteenth International Conference on Learning Representations*, 2024.

João Gama, Indrundefined Žliobaitundefined, Albert Bifet, Mykola Pechenizkiy, and Abdelhamid Bouchachia. A survey on concept drift adaptation. *ACM Comput. Surv.*, 46(4), March 2014. ISSN 0360-0300. doi: 10.1145/2523813. URL https://doi.org/10.1145/2523813.

Jacob Goldberger and Ehud Ben-Reuven. Training deep neural-networks using a noise adaptation layer. In *International Conference on Learning Representations*, 2016. URL https://api.semanticscholar.org/CorpusID:12190952.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021a. URL https://openreview.net/forum?id=d7KBjmI3GmQ.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. In Joaquin Vanschoren and Sai-Kit Yeung, editors, *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*, 2021b. URL https://datasets-benchmarks-proceedings.neurips.cc/paper/2021/hash/be83ab3ecd0db773eb2dc1b0a17836a1-Abstract-round2.html.

Yuzhen Huang, Yuzhuo Bai, Zhihao Zhu, Junlei Zhang, Jinghan Zhang, Tangjun Su, Junteng Liu, Chuancheng Lv, Yikai Zhang, Jiayi Lei, Yao Fu, Maosong Sun, and Junxian He. C-eval: A multi-level multi-discipline chinese evaluation suite for foundation models. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine, editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/c6ec1844bec96d6d32ae95ae694e23d8-Abstract-Datasets_and_Benchmarks.html.

Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In Yoshua Bengio and Yann LeCun, editors, *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014. URL http://arxiv.org/abs/1312.6114.

Haonan Li, Yixuan Zhang, Fajri Koto, Yifei Yang, Hai Zhao, Yeyun Gong, Nan Duan, and Timothy Baldwin. CMMLU: measuring massive multitask language understanding in chinese. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, volume ACL 2024 of *Findings of ACL*, pages 11260–11285. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.FINDINGS-ACL.671. URL https://doi.org/10.18653/v1/2024.findings-acl.671.

Lihong Li, Wei Chu, John Langford, and Robert E. Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, page 661–670, New York, NY, USA, 2010. Association for Computing Machinery. ISBN 9781605587998. doi: 10.1145/1772690.1772758. URL https://doi.org/10.1145/1772690.1772758.

Dawen Liang, Rahul G. Krishnan, Matthew D. Hoffman, and Tony Jebara. Variational autoencoders for collab-

orative filtering. In Pierre-Antoine Champin, Fabien Gandon, Mounia Lalmas, and Panagiotis G. Ipeirotis, editors, *Proceedings of the 2018 World Wide Web Conference on World Wide Web, WWW 2018, Lyon, France, April 23-27, 2018*, pages 689–698. ACM, 2018. doi: 10.1145/3178876.3186150. URL https://doi.org/10.1145/3178876.3186150.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report, 2025. URL https://arxiv.org/abs/2412.19437.

Jiayu Liu, Zhenya Huang, Tong Xiao, Jing Sha, Jinze Wu, Qi Liu, Shijin Wang, and Enhong Chen. Socraticlm: Exploring socratic personalized teaching with large language models. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 85693–85721. Curran Associates, Inc., 2024. doi: 10.52202/079017-2721. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/9bae399d1f34b8650351c1bd3692aeae-Paper-Conference.pdf.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. Self-refine: iterative refinement with self-feedback. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc.

Isaac Ong, Amjad Almahairi, Vincent Wu, Wei-Lin Chiang, Tianhao Wu, Joseph E. Gonzalez, M Waleed Kadous, and Ion Stoica. RouteLLM: Learning to route LLMs from preference data. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=8sSqNntaMr.

Shivam Patel, Neharika Jali, Ankur Mallick, and Gauri Joshi. Proxrouter: Proximity-weighted LLM query routing for improved robustness to outliers. In *The 29th International Conference on Artificial Intelligence and Statistics*, 2026. URL https://openreview.net/forum?id=GYWls3tyqM.

Sophia Xiao Pu, Sitao Cheng, Xin Eric Wang, and William Yang Wang. Dynamic evaluation for oversensitivity in LLMs. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng, editors, *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 2337–2344, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-335-7. doi: 10.18653/v1/2025.findings-emnlp.126. URL https://aclanthology.org/2025.findings-emnlp.126/.

Pranav Rajpurkar, Robin Jia, and Percy Liang. Know what you don't know: Unanswerable questions for SQuAD. In Iryna Gurevych and Yusuke Miyao, editors, *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-2124. URL https://aclanthology.org/P18-2124/.

Vikas C. Raykar, Shipeng Yu, Linda H. Zhao, Gerardo Hermosillo Valadez, Charles Florin, Luca Bogoni, and Linda Moy. Learning from crowds. *Journal of Machine Learning Research*, 11:1297—-1322, August 2010. ISSN 1532–4435.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: language agents with verbal reinforcement learning. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc.

Hwanjun Song, Minseok Kim, Dongmin Park, Yooju Shin, and Jae-Gil Lee. Learning from noisy labels with deep neural networks: A survey. *IEEE Trans. Neural Networks Learn. Syst.*, 34(11):8135–8153, 2023. doi: 10.1109/TNNLS.2022.3152527. URL https://doi.org/10.1109/TNNLS.2022.3152527.

Wei Song, Zhenya Huang, Cheng Cheng, Weibo Gao, Bihan Xu, GuanHao Zhao, Fei Wang, and Runze Wu. IRT-router: Effective and interpretable multi-LLM routing via item response theory. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15629–15644, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.761. URL https://aclanthology.org/2025.acl-long.761/.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4149–4158, Minneapolis, Minnesota, June 2019. Association for Computa-

tional Linguistics. doi: 10.18653/v1/N19-1421. URL https://aclanthology.org/N19-1421/.

Tong Xiao, Tian Xia, Yi Yang, Chang Huang, and Xiaogang Wang. Learning from massive noisy labeled data for image classification. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2691–2699, 2015. doi: 10.1109/CVPR.2015.7298885.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2.5 technical report, 2025. URL https://arxiv.org/abs/2412.15115.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1259. URL https://aclanthology.org/D18-1259/.

Haozhen Zhang, Tao Feng, and Jiaxuan You. Router-r1: Teaching llms multi-round routing and aggregation via reinforcement learning. NeurIPS 2025 Poster, San Diego, USA, 2025. URL https://neurips.cc/virtual/2025/loc/san-diego/poster/119214. Poster Session.

Yixuan Zhang and Haonan Li. Can large language model comprehend Ancient Chinese? a preliminary test on ACLUE. In Adam Anderson, Shai Gordin, Bin Li, Yudong Liu, and Marco C. Passarotti, editors, *Proceedings of the Ancient Language Processing Workshop*, pages 80–87, Varna, Bulgaria, September 2023. INCOMA Ltd., Shoumen, Bulgaria. URL https://aclanthology.org/2023.alp-1.9/.

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, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. A survey of large language models, 2025. URL https://arxiv.org/abs/2303.18223.

Kaijie Zhu, Jiaao Chen, Jindong Wang, Neil Gong, Diyi Yang, and Xing Xie. Dyval: Dynamic evaluation of large language models for reasoning tasks. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun, editors, *International Conference on Learning Representations*, volume 2024, pages 18091–18128, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/4ea4a1ea4d9ff273688c8e92bd087112-Paper-Conference.pdf.

Appendix
(Supplementary Material)

Juntao Liu[1,2] **Lixing Yu**[1,2] **Kun Yue**[1,2] **Zhiwen Tang**[*1,2]

[1]School of Information Science and Engineering, Yunnan University, Kunming,China
[2] Yunnan Key Laboratory of Intelligent Systems and Computing, Kunming, China

A IMPLEMENTATION DETAILS

We employ the `bert-base-uncased`[1]encoder for all text embeddings, providing a consistent 768-dimensional representation space for queries, capability profiles, and behavioral signatures. Capability profiles are constructed offline using 2-shot exemplars per dataset, selected to maximize the coverage of task-specific failure modes. For behavioral signature construction, each model maintains a task-aware online feedback buffer. We summarize every $B_{\mathcal{I}} = 10$ newly observed query-response pairs into a new signature, retain a sliding window of at most $W_{\mathcal{I}} = 20$ recent signatures, and retrieve up to $R_{\mathcal{I}} = 3$ most relevant cached signatures at inference time via task-cluster matching.

During training, we optimize the Evidence Lower Bound (ELBO). To ensure that the latent variable z captures meaningful reliability information rather than collapsing to the prior, a common issue known as posterior collapse, we weight the KL divergence term with a coefficient λ. We conduct a sensitivity analysis over $\lambda \in \{1.0, 0.1, 0.01, 0.001\}$, as shown in Figure 5. We observe that larger λ values lead to a vanishing KL divergence, indicating that the model ignores the latent variable. Therefore, we set $\lambda = 10^{-3}$, which encourages a structured latent space with non-vanishing KL divergence without degrading reconstruction accuracy. To approximate the expectations in the ELBO objective and posterior predictive distribution, we employ Monte Carlo sampling with the reparameterization trick. We set the number of Monte Carlo samples to $N = 5$ during training to balance gradient variance and computational efficiency, and increase it to $N = 10$ during inference to ensure robust marginalization, as analyzed in Appendix C . All experiments are conducted on NVIDIA A100 GPUs with mixed-precision training.

B DATASETS DETAILS

Following the evaluation protocol established by IRT-Router[Song et al., 2025], our benchmark suite comprises 12 distinct datasets, strategically partitioned into In-Distribution (ID) and Out-of-Distribution (OOD) sets. The precise Train and Test size splits for the ID datasets, as well as the evaluation metrics utilized, strictly mirror the IRT-Router methodology to ensure a rigorous and fair baseline comparison. The in-distribution (ID) datasets are used to train the router and construct the initial profiles. These datasets are selected to ensure comprehensive, balanced coverage of core LLM capabilities, including general knowledge, logical reasoning, coding, and mathematics. To rigorously evaluate the router's OOD generalization ability to unseen scenarios, we exclusively reserve four held-out datasets for out-of-distribution (OOD) testing.

*Corresponding author.

[1]`https://huggingface.co/google-bert/bert-base-uncased`

[2]For closed-source models (e.g., GPT-4o), pricing is based on official API rates. For open-source models, we refer to third-party providers such as Together AI `https://www.together.ai/`

Table 3: Benchmark datasets used in evaluation. **In-Distribution** datasets are used for training the router, while **Out-of-Distribution** datasets are reserved strictly for evaluation to unseen scenarios.

Dataset Used for In-Distribution				
Dataset	**Type**	**Evaluation Metric**	**Train Size**	**Test Size**
ACLUE [Zhang and Li, 2023]	Ancient Chinese	accuracy	1400	600
ARC_C [Clark et al., 2018]	Reasoning	accuracy	1400	600
CMMLU [Li et al., 2024]	Chinese Multitask	accuracy	7000	3000
Hotpot_QA [Yang et al., 2018]	Multi-Hop	EM	1400	600
MATH [Hendrycks et al., 2021b]	Math	accuracy	1400	600
MBPP [Austin et al., 2021]	Code	pass@1	630	270
MMLU [Hendrycks et al., 2021a]	Multitask	accuracy	9800	4200
SQuAD [Rajpurkar et al., 2018]	Reading Comprehension	f1	1400	600
Dataset Used for Out-of-Distribution				
Dataset	**Task type**	**Evaluation Metric**	**Train Size**	**Test Size**
C-Eval [Huang et al., 2023]	Chinese Multitask	accuracy	–	1000
Commonsense_QA [Talmor et al., 2019]	Commonsense Reasoning	accuracy	–	1000
GSM8K [Cobbe et al., 2021]	Math	accuracy	–	1000
HumanEval [Chen et al., 2021]	Code	pass@1	–	160

Table 4: Candidate LLMs and API pricing (USD per 1M tokens) [2]. Models are categorized into the standard training pool and the unseen pool used for OOD evaluation. DeepSeek-V3.2 is utilized solely as the Auditor for both capability profile and behavioral signature.

LLM	**Input $/1M**	**Output $/1M**
DeepSeek-Chat	0.14	0.28
DeepSeek-Coder	0.14	0.28
GLM-4-Air	0.137	0.137
GLM-4-Flash	0.0137	0.0137
GLM-4-Plus	6.85	6.85
GPT-4o	2.5	10
GPT-4o Mini	0.15	0.6
GPT-4o Mini+COT	0.15	0.6
Llama3.1-8B-Instruct	0.1	0.2
Llama3.1-70B-Instruct	0.792	0.792
Mistral-8B-Instruct-2410	0.1	0.2
Mistral-7B-Instruct-v0.2	0.1	0.2
Mixtral-8x7B-Instruct	0.54	0.54
Qwen2.5-32B-Instruct-GPTQ-Int4	0.1	0.2
Qwen2.5-7B-Instruct	0.1	0.2
Qwen2.5-72B-Instruct	1.08	1.08
Qwen2.5-Math-7B-Instruct	0.1	0.2
Unseen Test Models (OOD-Model)		
Gemini-1.5-Flash	0.075	0.30
QwQ-32B-Preview	1.20	1.20
Llama-3.1-405B-Instruct	3.15	3.15
Auditor Model		
DeepSeek-V3.2	0.27	0.42

C SENSITIVITY TO MONTE CARLO SAMPLES AND SIGNATURE CONSTRUCTION

Sensitivity to Monte Carlo Samples (N). To evaluate the impact of the Monte Carlo integration scale, we report the routing rewards across varying sample sizes $N \in \{1, 5, 10, 20\}$ during inference in Table 5. We observe that the performance–cost trade-offs remain remarkably stable, with marginal variations across all evaluation splits. This demonstrates that LENS is robust to the choice of N, and $N = 10$ provides a reliable empirical approximation for marginalization without incurring significant computational latency.

Table 5: Routing reward across different numbers of Monte Carlo samples (N) during inference ($\alpha = 0.5, \beta = 0.5$).

Evaluation Split	$N = 1$	$N = 5$	$N = 10$	$N = 20$
In-Distribution (ID)	38.15	38.17	38.46	38.16
OOD-Task	42.90	42.87	42.88	42.90
OOD-Model	2.73	2.62	2.66	2.63
OOD-Combined	30.11	30.10	30.16	30.13

Behavioral Signature Construction Details. To ensure online efficiency, behavioral signatures are cached and maintained non-parametrically rather than recomputed per query. Each candidate model is allocated a task-aware feedback buffer. Upon accumulating every 10 new query-response interactions, the auxiliary auditor (DeepSeek-V3.2) condenses these localized traces into a new behavioral signature. We maintain a sliding window of at most 20 historical signatures per model to eliminate outdated feedback and bound memory overhead. At inference time, the router performs task-cluster matching based on the embedding similarity of the incoming query to adaptively retrieve up to 3 most relevant cached signatures for precision-modulated fusion.

D SENSITIVITY ANALYSIS OF KL REGULARIZATION

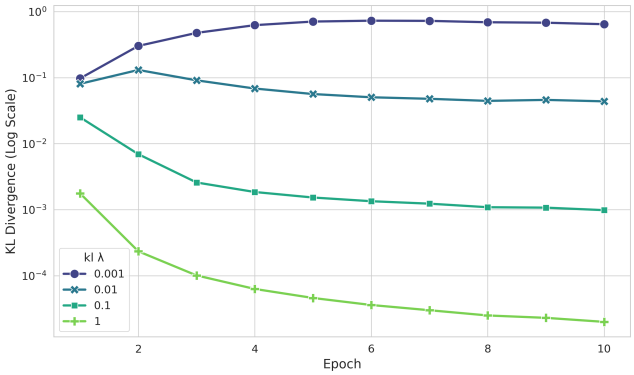


Figure 5: **Sensitivity Analysis of KL Regularization Weight λ.**

Figure 5 shows the validation KL divergence (log scale) over training epochs for different λ values. We observe that standard VAE settings ($\lambda = 1.0, 0.1$) result in posterior collapse ($KL \approx 0$), where the latent variable becomes uninformative. In contrast, $\lambda = 10^{-3}$ (our choice) maintains a healthy latent structure ($KL \approx 0.77$) while achieving comparable validation accuracy to other settings.

E COST-BENEFIT BREAKDOWN

Figure 6 shows the Cost-Benefit breakdown in the OOD Model scenario ($\alpha = 0.5$). The behavioral signature generation incurs a marginal overhead (**0.92%** of total cost) yet drives a substantial reward improvement (**+123.53%**) and a net cost reduction (**-4.77%**). This highlights the high return on investment of the reflection mechanism, which effectively leverages latent reliability to correct expensive errors with minimal computational expense.

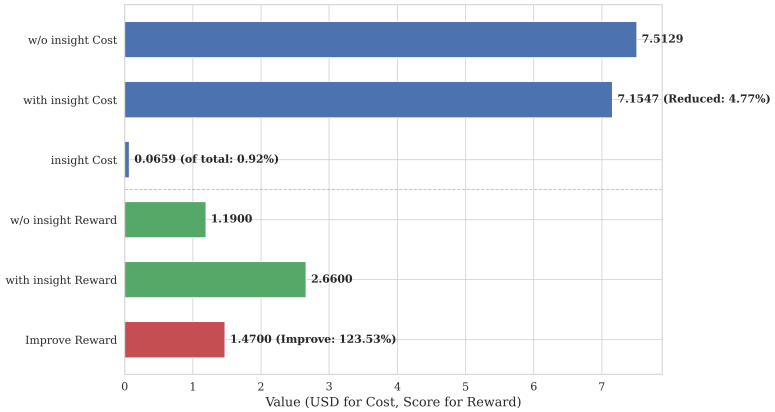


Figure 6: Cost-Benefit breakdown in the OOD Model scenario ($\alpha = 0.5$).

F PROMPTS

In Section 4, we mentioned that we construct profiles through k-shot (each dataset) and generate behavioral signature based on the responses of each model. Specifically, we use DeepSeek V3.2 to accomplish this task. The prompt words are shown in the following two subsections.

F.1 CAPABILITY PROFILE GENERALIZATION PROMPT

System Prompt for Profile Generation

You are a Model Auditor. Analyze the behavior of the LLM '[Model Name]' based on the logs below. Your goal is to identify SPECIFIC limitations and distinct behavioral patterns.
=== INTERACTION LOGS ===
[Insert List of Input-Output Pairs grouped by Category]
=== END LOGS ===
=== TASK & CONSTRAINTS ===
Generate a JSON profile. **CRITICAL CONSTRAINTS:**

1. **NO FLUFF**: Do not use complete sentences. Use keywords, short phrases, or bullet points.
2. **MAX LENGTH**: 'failure_analysis' must be under 40 words. 'overall_summary' under 30 words.
3. **FOCUS ON ERRORS**: We need to know where the model FAILS, not just where it succeeds.

Analyze these dimensions:

- Instruction Following: (e.g., 'Ignores word count', 'Follows formatting well')
- Reasoning: (e.g., 'Hallucinates in calculus', 'Logic sound but verbose')
- Failure Modes: (e.g., 'Fails nested JSON', 'Cannot handle negation')

=== OUTPUT FORMAT ===
Output strictly valid JSON:

```
{
  "characteristics": {
    "instruction_following": "Key phrase...",
    "reasoning_depth": "Key phrase...",
    "tone_style": "Key phrase...",
    "safety_robustness": "Key phrase..."
  },
  "failure_analysis": "Dense bullet points of errors...",
  "weakness_tags": ["tag1", "tag2"],
  "overall_summary": "High-level summary..."
}
```

F.2 BEHAVIORAL SIGNATURE GENERALIZATION PROMPT

System Prompt for Dynamic Behavioral Signature Generation

You are a Model Performance Auditor. Your task is to analyze a single interaction of the LLM '[Model Name]' and determine if its performance contradicts or updates its known Static Profile.
=== STATIC PROFILE SUMMARY ===
[Insert Static Profile Summary Here]
=== INTERACTION LOG ===
User Query: *[Insert Question]*
Model Response: *[Insert Model Response]*
Ground Truth/Reference: *[Insert Ground Truth]*
Evaluated Score (0-1): *[Insert Score]*
=== TASK ===

1. Compare the Score against the Static Profile. Is this result expected?
   - If the score is LOW but the profile claims the model is 'expert' in this domain, this is a **Negative Deviation**.
   - If the score is HIGH but the profile says the model is 'weak', this is a **Positive Deviation**.
   - If the result matches the profile (e.g., weak model fails hard question), it is **Expected**.
2. Identify the specific skill or constraint involved (e.g., 'JSON Formatting', 'Calculus', 'Safety Refusal').

=== OUTPUT FORMAT ===
Provide a JSON object with the following keys. Use double quotes for all keys and string values, and do not include comments:

```
{
  "is_deviation": true or false,
  "deviation_type": "positive" or "negative" or "none",
  "reasoning": "brief explanation of the discrepancy",
  "impacted_skills": ["skill1", "skill2"],
  "behavioral_signature_text": "concise sentence to add to the dynamic memory"
}
```

G PER-DATASET PERFORMANCE BREAKDOWN

Table 6 details the per-dataset performance breakdown across all distribution settings, evaluated under Performance-First ($\alpha = 0.8$), Balanced ($\alpha = 0.5$), and Cost-First ($\alpha = 0.2$) trade-off regimes. Due to space constraints, we report the dataset-level results for LENS and the strongest baseline, IRT-Router (ChatGPT). The evaluation is conducted on the full test split of each respective dataset. Minor variations between the weighted averages of these dataset-level metrics and the globally aggregated scores in Table 1 are expected due to split-specific cost normalization and slight stochasticity during reproducibility runs. Crucially, the dataset-level breakdown confirms that LENS maintains a consistent advantage in reward across the majority of individual tasks, particularly under out-of-distribution scenarios.

Table 6: **Per-Dataset Results.** Performance (%), Total Cost (USD), and Reward for LENS and IRT-Router across distribution settings and trade-off regimes evaluated on full datasets.

Setting	Dataset	Method	Performance-First			Balanced			Cost-First		
			Perf. ↑	Cost ↓	Reward ↑	Perf. ↑	Cost ↓	Reward ↑	Perf. ↑	Cost ↓	Reward ↑
In-Distribution	ACLUE	LENS	66.50%	0.0200	53.10	65.33%	0.0197	32.41	65.17%	0.0150	12.72
		IRT-Router	66.67%	0.3417	51.54	65.67%	0.2736	29.24	65.67%	0.2329	8.24
	ARC-C	LENS	95.33%	0.0149	76.18	96.49%	0.0145	48.06	95.49%	0.0143	18.80
		IRT-Router	95.83%	0.1491	75.88	95.83%	0.1395	46.08	94.99%	0.1079	16.73
	CMMLU	LENS	83.13%	0.0946	66.01	83.17%	0.0875	40.43	81.37%	0.0722	14.75
		IRT-Router	84.07%	1.9335	57.09	83.80%	1.7851	18.45	83.20%	1.4707	-14.27
	HotpotQA	LENS	36.33%	0.0423	28.84	37.33%	0.0206	18.40	36.50%	0.0245	6.79
		IRT-Router	41.50%	0.1335	32.50	41.83%	0.1310	19.20	41.33%	0.1157	5.84
	MATH	LENS	80.67%	0.0862	64.08	82.67%	0.0870	40.19	82.50%	0.0859	14.70
		IRT-Router	83.83%	0.0861	66.61	82.67%	0.0866	40.20	83.00%	0.0876	14.76
	MBPP	LENS	85.43%	0.0320	68.18	87.16%	0.0314	43.17	82.72%	0.0244	16.03
		IRT-Router	85.56%	0.1187	67.82	86.30%	0.0944	41.91	86.79%	0.0360	16.60
	MMLU	LENS	83.73%	0.1826	66.03	83.40%	0.1385	39.88	81.35%	0.1248	13.65
		IRT-Router	84.64%	1.9213	57.61	84.97%	1.7975	18.87	84.61%	1.5662	-16.00
	SQuAD	LENS	82.08%	0.0274	65.52	81.64%	0.0208	40.54	80.90%	0.0160	15.84
		IRT-Router	78.82%	0.1714	62.15	78.58%	0.1677	37.09	79.53%	0.1247	13.28
OOD-Task	C-Eval	LENS	82.98%	0.0331	66.21	82.98%	0.0330	41.06	82.58%	0.0293	15.90
		IRT-Router	82.08%	0.6818	62.08	82.58%	0.6311	33.00	81.38%	0.5165	5.42
	CSQA	LENS	84.40%	0.0250	67.39	84.70%	0.0237	42.04	86.70%	0.0194	16.93
		IRT-Router	86.20%	0.2875	67.45	85.20%	0.2716	39.03	85.00%	0.2230	12.31
	GSM8K	LENS	92.80%	0.0673	73.89	93.10%	0.0666	45.67	92.50%	0.0623	17.19
		IRT-Router	93.20%	0.0898	74.09	93.40%	0.0866	45.56	93.20%	0.0893	16.76
	HumanEval	LENS	90.62%	0.0166	72.41	89.38%	0.0199	44.43	86.25%	0.0127	16.98
		IRT-Router	91.87%	0.0593	73.19	90.62%	0.0470	44.70	91.25%	0.0215	17.80
OOD-Model	ACLUE	LENS	50.83%	0.6635	37.18	48.00%	0.6457	15.52	55.50%	0.0925	9.16
		IRT-Router	48.33%	0.6414	35.30	48.33%	0.6414	15.74	48.33%	0.6414	-3.81
	ARC-C	LENS	86.14%	0.1647	68.05	86.64%	0.1467	41.40	93.99%	0.0093	18.60
		IRT-Router	86.14%	0.1647	68.05	86.14%	0.1647	40.91	86.14%	0.1647	13.77
	CMMLU	LENS	63.37%	2.5955	37.06	59.60%	2.4166	-1.94	69.83%	0.4967	3.53
		IRT-Router	59.00%	2.4697	34.22	59.00%	2.4697	-2.94	59.00%	2.4697	-40.11
	HotpotQA	LENS	7.00%	0.6283	2.30	8.17%	0.5731	-3.44	30.83%	0.0225	5.69
		IRT-Router	6.83%	0.6321	2.15	6.67%	0.6349	-5.01	6.83%	0.6229	-11.73
	MATH	LENS	82.83%	1.0393	60.81	81.33%	0.4545	34.70	77.17%	0.0789	13.78
		IRT-Router	83.17%	1.1370	60.56	83.17%	1.1370	26.65	83.17%	1.1370	-7.26
	MBPP	LENS	71.48%	0.5899	54.09	75.19%	0.4137	32.16	81.98%	0.0139	16.10
		IRT-Router	67.90%	0.7481	50.39	67.90%	0.7455	24.16	67.90%	0.7481	-2.14
	MMLU	LENS	65.05%	2.2073	40.45	65.89%	2.0666	5.80	76.80%	0.0842	13.59
		IRT-Router	64.72%	2.2922	39.73	64.72%	2.2922	2.25	64.72%	2.2922	-35.23
	SQuAD	LENS	40.90%	0.2495	31.41	41.40%	0.2421	17.52	79.48%	0.0151	15.58
		IRT-Router	39.34%	0.2566	30.12	39.34%	0.2566	16.30	39.34%	0.2566	2.47
OOD-Combined	C-Eval	LENS	61.16%	0.9641	43.86	54.65%	0.8841	15.71	68.47%	0.2345	8.76
		IRT-Router	52.35%	0.9608	36.83	52.35%	0.9608	13.55	52.35%	0.9608	-9.72
	CSQA	LENS	82.80%	0.2347	65.01	82.70%	0.2074	38.63	82.70%	0.0135	16.26
		IRT-Router	82.90%	0.2374	65.07	82.90%	0.2374	38.33	82.90%	0.2374	11.59
	GSM8K	LENS	92.90%	0.7997	70.12	91.40%	0.7749	35.52	59.60%	0.0300	11.29
		IRT-Router	93.00%	0.8064	70.16	93.00%	0.8064	35.91	93.00%	0.8064	1.65
	HumanEval	LENS	85.00%	0.2007	66.95	86.88%	0.1565	41.38	83.75%	0.0132	16.47
		IRT-Router	80.63%	0.3444	62.69	80.63%	0.3444	35.79	80.63%	0.3444	8.89

Table 7: LLM Static Capability Profile generated via k-shot Probing.

LLM	Profile
glm_4_air	**Overall:** Strong in Chinese humanities, weak in STEM and factual recall; prone to confident errors. **Characteristics:** instruction_following: Follows formatting, sometimes misses nuance; reasoning_depth: Variable; strong in humanities, weak in STEM; tone_style: Explanatory, confident. **Failure analysis:** Fails physics (gravity, momentum). Hallucinates facts (music, biology). Math errors (algebra, precalculus). Misinterprets literary analysis. **Weakness tags:** factual_hallucination, stem_reasoning, numerical_calculation, textual_interpretation
glm_4_flash	**Overall:** Competent on simple tasks; fails on factual accuracy, complex reasoning, and precise calculations. **Characteristics:** instruction_following: Follows formatting, sometimes ignores content accuracy; reasoning_depth: Variable; basic logic okay, fails on complex or factual reasoning; tone_style: Concise, analytical, but can be overconfident. **Failure analysis:** Factual errors in trivia (e.g., music, botany). Math errors in simplification. Misinterprets historical/religious questions. Overconfident in wrong answers. **Weakness tags:** factual_inaccuracy, mathematical_error, context_misinterpretation, overconfidence
glm_4_plus	**Overall:** Competent in structured QA and coding, but fails on precise technical calculations and nuanced comprehension. **Characteristics:** instruction_following: Generally follows explicit instructions, but can misinterpret problem details.; reasoning_depth: Variable; strong in humanities and basic math, but fails in precise algebraic manipulation and physics reasoning.; tone_style: Explanatory and structured, often provides step-by-step analysis.. **Failure analysis:** Algebra error: miscombined like terms. Physics error: misapplied inverse-square law for gravity. Misinterpreted literature question central thesis. **Weakness tags:** algebraic_manipulation, physics_reasoning, textual_interpretation
gpt_4o	**Overall:** Strong in STEM/logic, weak in cultural/linguistic tasks, overconfident in errors. **Characteristics:** instruction_following: Follows formatting well, adheres to output structure; reasoning_depth: Sound in STEM, inconsistent in cultural/linguistic nuance; tone_style: Explanatory, sometimes overconfident in incorrect answers. **Failure analysis:** Fails Chinese couplet matching; misinterprets classical Chinese text context; errors in literary analysis (e.g., misidentifying central thesis). **Weakness tags:** cultural_context, linguistic_nuance, literary_analysis
gpt_4o_mini	**Overall:** Reliable on STEM/facts; weak on cultural/linguistic tasks; occasional code logic errors. **Characteristics:** instruction_following: Follows formatting, sometimes misses implicit context; reasoning_depth: Strong in STEM, inconsistent in cultural/linguistic nuance; tone_style: Formal, explanatory. **Failure analysis:** Fails Chinese couplet matching; misinterprets polysemy; code generation errors (incorrect cone formula). **Weakness tags:** cultural_context, linguistic_nuance, implicit_knowledge, code_accuracy

(Continued on next page)

LLM	Profile
gpt_4o_mini_cot	**Overall:** Structured reasoning but fails on nuanced tasks like poetry and physics. **Characteristics:** instruction_following: Follows formatting, sometimes verbose; reasoning_depth: Step-by-step but can misapply logic; tone_style: Formal, explanatory. **Failure analysis:** Misinterprets Chinese couplet structure; misapplies physics formula; incorrectly assumes cone input parameters. **Weakness tags:** cultural_context, formula_misapplication, assumption_errors
deepseek_coder	**Overall:** Strong in math and factual QA, fails in nuanced language and applied physics. **Characteristics:** instruction_following: Follows formatting well, provides structured outputs; reasoning_depth: Sound in math and factual recall, inconsistent in polysemy and physics; tone_style: Formal, explanatory, sometimes verbose. **Failure analysis:** Polysemy resolution error (misinterprets). Physics reasoning error (miscalculates momentum). Code misunderstanding (cone lateral area uses slant height directly, not derived from height). **Weakness tags:** polysemy_resolution, physics_reasoning, code_interpretation
deepseek_chat	**Overall:** Reliable on structured tasks, fails on nuanced language and some applied reasoning. **Characteristics:** instruction_following: Follows formatting well, provides structured outputs; reasoning_depth: Strong in math and factual recall, inconsistent in polysemy and physics; tone_style: Formal, explanatory, sometimes verbose. **Failure analysis:** Polysemy resolution error (misinterprets), physics reasoning error (momentum miscalculation), code misunderstanding), code misunderstanding (cone lateral area formula). **Weakness tags:** polysemy_ambiguity, physics_misapplication, code_requirement_misinterpretation
qwen25_32b_int4	**Overall:** Reliable on structured tasks, fails on specific factual and quantitative problems. **Characteristics:** instruction_following: Follows formatting, sometimes verbose; reasoning_depth: Strong in structured domains, inconsistent in factual recall; tone_style: Analytical, confident even when incorrect. **Failure analysis:** Factual errors in pop culture and taxonomy. Incorrect momentum calculation. Overconfident in wrong answers. **Weakness tags:** factual_recall, numeric_calculation, overconfidence
qwen25_7b_instruct	**Overall:** Inconsistent accuracy; strong in basic QA, weak in nuanced reasoning and precise tasks. **Characteristics:** instruction_following: Follows formatting, but misinterprets problem details; reasoning_depth: Inconsistent; logical in some domains, flawed in others; tone_style: Explanatory, sometimes overconfident. **Failure analysis:** Fails polysemy/cultural context (couplet, religion). Math errors (algebra, trig). Hallucinates facts (music trivia). Misreads instructions (cone surface area). **Weakness tags:** cultural_knowledge, mathematical_calculation, factual_hallucination, instruction_misinterpretation

(Continued on next page)

LLM	Profile
qwen25_72b_instruct	**Overall:** Strong in structured math/code, weak in nuanced language, factual accuracy, and some science. **Characteristics:** instruction_following: Follows formatting, sometimes verbose; reasoning_depth: Inconsistent; strong in math, weak in polysemy and factual recall; tone_style: Analytical, explanatory. **Failure analysis:** Polysemy resolution errors (e.g., misinterpreted). Factual recall errors (e.g., Lithuanian producer, genus species count). Physics reasoning errors (e.g., gravitational force, momentum). Chinese literature interpretation errors. **Weakness tags:** polysemy_resolution, factual_recall, physics_reasoning, chinese_literature
qwq_32b_preview	**Overall:** Competent in math/logic, weak in factual recall and nuanced cultural tasks. **Characteristics:** instruction_following: Follows formatting, but verbose explanations; reasoning_depth: Logical in math, inconsistent in factual recall; tone_style: Explanatory, sometimes overly detailed. **Failure analysis:** Factual errors in music/biology; misinterprets Chinese couplet structure; overconfident in incorrect answers; fails to verify niche knowledge. **Weakness tags:** factual_inaccuracy, cultural_context_misunderstanding, overconfidence, specialized_knowledge_gaps
qwen25_math_7b_instruct	**Overall:** Unreliable: produces nonsense under moderate complexity, especially in humanities and non-English contexts. **Characteristics:** instruction_following: Often ignores output constraints, verbose, inconsistent formatting; reasoning_depth: Structured on math/science, chaotic on humanities, hallucinates facts; tone_style: Repetitive, incoherent segments, mixes languages, abrupt truncation; safety_robustness: Refuses some queries, but generates gibberish or nonsense in others. **Failure analysis:** Severe output degradation: gibberish, repetition, truncation. Fails Chinese cultural tasks (couplets, polysemy). Hallucinates in biology/history. Code logic errors. Ignores word limits. **Weakness tags:** output_degradation, cultural_knowledge, factual_hallucination, repetition, truncation, code_errors
llama31_8b_instruct	**Overall:** Prone to confident, incorrect answers via shallow pattern matching over deep reasoning. **Characteristics:** instruction_following: Follows formatting, but ignores implicit constraints (e.g., answer selection).; reasoning_depth: Superficial pattern matching; fails on nuanced or multi-step logic.; tone_style: Overconfident, declarative; uses 'analysis' framing incorrectly.. **Failure analysis:** Chooses plausible-sounding but incorrect options. Misinterprets context in polysemy, literature, and history. Fails basic physics calculations. Hallucinates facts in open-domain queries. Makes arithmetic errors in algebra. **Weakness tags:** context_misinterpretation, factual_hallucination, arithmetic_error, superficial_reasoning, overconfidence

(Continued on next page)

LLM	Profile
llama31_70b_instruct	**Overall:** Competent on straightforward factual and basic reasoning tasks; fails on nuanced cultural, historical, and precise mathematical applications. **Characteristics:** instruction_following: Generally adheres to format, but sometimes misinterprets task specifics; reasoning_depth: Variable; strong in basic math and factual recall, weak in nuanced cultural/historical contexts; tone_style: Analytical and explanatory, often provides step-by-step reasoning. **Failure analysis:** Fails Chinese couplet prediction (misinterprets cultural context). Incorrect on Chinese literature analysis (misidentifies central thesis). Makes critical math error in trigonometry (tan double angle). Misinterprets cone surface area problem (ignores given parameters). **Weakness tags:** cultural_context, historical_analysis, advanced_math, problem_misinterpretation
llama31_405b_instruct	**Overall:** Strong STEM performer; fails on nuanced cultural/literary tasks; occasional math errors. **Characteristics:** instruction_following: Follows formatting, sometimes verbose; reasoning_depth: Variable; strong in STEM, weak in cultural context; tone_style: Analytical, explanatory. **Failure analysis:** Fails Chinese literary tasks (couplet, classical text analysis). Algebra error: incorrect like-term combination. Over-explains simple questions. **Weakness tags:** cultural_context, chinese_literature, algebraic_manipulation, verbosity
mixtral_8x7b_instruct	**Overall:** Competent on straightforward tasks but unreliable for precise calculations, factual accuracy, and nuanced reasoning. **Characteristics:** instruction_following: Generally follows instructions, but sometimes ignores explicit constraints or adds unsolicited explanations.; reasoning_depth: Variable; can perform basic calculations and logic, but struggles with complex or multi-step reasoning, especially in math and factual recall.; tone_style: Formal, explanatory, and sometimes overly verbose.. **Failure analysis:** Math errors (algebra, trigonometry). Factual hallucinations (music, history). Misinterprets Chinese idioms. Overcomplicates simple tasks. Fails to select correct option when answer is present. **Weakness tags:** mathematical_calculation, factual_hallucination, cultural_context, multi_step_reasoning, overcomplication
mistral_7b_instruct_v02	**Overall:** Model frequently produces incorrect answers with confident but flawed reasoning across diverse topics. **Characteristics:** instruction_following: Inconsistent; sometimes follows format, often ignores reasoning steps or provides incorrect answers.; reasoning_depth: Shallow; frequent logical errors, misapplies formulas, and fails to verify answers.; tone_style: Confident but often incorrect; uses explanatory language even when wrong.. **Failure analysis:** Math errors (algebra, precalculus). Misinterprets questions (couplet, polysemy). Factual inaccuracies (history, science). Overconfident incorrect explanations. Code correct but reasoning flawed. **Weakness tags:** mathematical_calculation, logical_reasoning, factual_accuracy, context_interpretation, overconfidence

(Continued on next page)

LLM	Profile
ministral_8b_instruct_2410	**Overall:** Inconsistent performance; strong in math, weak in language nuance, factual knowledge, and applied reasoning. **Characteristics:** instruction_following: Follows formatting, sometimes misinterprets task specifics; reasoning_depth: Variable; strong on algebra, weak on physics and polysemy; tone_style: Formal, explanatory. **Failure analysis:** Fails polysemy resolution. Fails basic physics (gravity, momentum). Fails factual recall (music, history). Misinterprets code function specs (cone lateral area). **Weakness tags:** polysemy_resolution, physics_reasoning, factual_recall, specification_misinterpretation
gemini15_flash	**Overall:** Struggles with nuanced language, precise calculation, and integrating multiple constraints. **Characteristics:** instruction_following: Follows formatting, sometimes misinterprets task core; reasoning_depth: Inconsistent; fails on multi-step logic and context nuance; tone_style: Explanatory, confident even when wrong. **Failure analysis:** Fails polysemy/cultural context (couplets, classical Chinese). Misapplies physics formulas. Incorrect literary/historical analysis. Basic algebra errors (sign mistakes). **Weakness tags:** contextual_disambiguation, multi-step_reasoning, cultural_knowledge, arithmetic_accuracy