
A generalist Router for *Inspect*: Reasoning router demonstration

Gerard Boxó
UAB

Aman Neelappa
Independent

Ishan Garg
Independent

Alan McBeth
Independent

With
Martian & Apart Research

Abstract

The current paradigm for model serving embraced by most model providers consists on offering the user a small selection of models with distinct capabilities and prices. While this offers some flexibility to the user, manually choosing the model introduces some amount of friction to the overall experience. The field of model routing presents a compelling solution, by coordinating the inference between models with a diverse set of capabilities and costs. The introduction of reasoning models to the LLM stack, presents a situation in which cost can be clearly traded off for increase in performance. However, despite reasoning models excelling in some areas like math and code, they do not prove to be better in other areas like free-form writing and knowledge retrieval. Being able to route user queries to either reasoning and non-reasoning models is a clear use case of LLM routers, likely resulting in not just improved performance at a lower cost, but also an improved user experience via reductions in thinking time, latency and effective context length.

In this research hackathon we aim to make the evaluation of model routing more accessible by developing and integrating router tooling within the [Inspect Framework](#), an open-source platform for LLM evaluation from the UK AI Security Institute. As a motivating example we focused on building, training, and evaluating routers capable of classifying queries between ones that could benefit from reasoning and ones that would not benefit from reasoning.

Keywords: Model Routing, LLM Embeddings, Expert Orchestration

1. Introduction

1.1 Research Question & Motivation

Can a model router learn when a prompt requires a reasoning model vs. when the prompt can be accurately handled by a non-reasoning model? We hypothesize that a model router trained on the performance of a reasoning model and a non-reasoning model on a variety of prompts can achieve a higher task accuracy than the non-reasoning model alone and can achieve lower cost and latency than the reasoning model alone.

1.2 Challenge Track Focus

Track 1: Judge Model Development

We train a judge for use by the model router to determine whether a query is better suited for a reasoning model or a non-reasoning model.

1.3 Contribution to Expert Orchestration

Monolithic AI systems suffer from various deficiencies, including market concentration, high barriers to entry, limited user control, and an inefficient one-size-fits-all approach. Alternatively, an Expert Orchestration Architecture approach could democratize the AI ecosystem by making space for specialized model developers to excel in specific domains and could provide superior performance at lower cost.

Reason-based routing directly tackles the performance-cost dilemma by addressing one of the key factors in the performance-cost trade-off—reasoning can increase the cost of a query by an order of magnitude or more.

Reason-based routing fits well into the Expert Orchestration Architecture approach by considering whether a model that uses expensive and time-consuming reasoning is needed for a particular task. Not only does this directly speak to one of the main benefits of the Expert Orchestration Architecture approach, it also helps to realize its other principal benefits. Because reasoning is a crucial determinant of cost, failing to take reasoning into account in routing decisions may obscure the strengths of some specialized models, as they may be assessed inaccurately by treating them as equally suitable for reasoning and non-reasoning tasks. By helping to surface the most suitable specialized model across a variety of domains, reason-based routing may help to more fully realize the benefits of the Expert Orchestration Architecture approach, from democratization, to safety, to transparency, to efficiency.

2. Methods

2.1 Technical Approach

Our methodology builds on the matrix factorization-based router introduced by Hu et al. (2024), which has demonstrated state-of-the-art performance in multi-LLM routing tasks. We focused on mathematical reasoning problems spanning a range of difficulty levels, specifically drawing from the GSM8K and AIME datasets. These datasets were selected to ensure coverage of both elementary and advanced mathematical reasoning.

For the underlying language models, we selected two versions of the same model: Qwen 2.5 7B Instruct and a distilled version of Deepseek R1. Both models were chosen for their complementary strengths. Given that the reasoning model has the same architecture and parameter count as the non-reasoning counterpart, we can make an apples-to-apples comparison in which the cost differences are just a matter of the token use.

We implemented an efficient matrix factorization class, for efficient training with GPUs and minimal overhead at inference.

For embedding the queries, we experimented with two approaches: (1) using the OpenAI embed-small-3 model, and (2) extracting model-specific embeddings from each LLM. The router was trained to assign incoming queries to the model most likely to produce a correct answer, based on these embeddings.

The router was trained on the training set and evaluated on the held-out set to assess generalization.

2.2 Routing/Judge Architecture

The routing system follows the matrix factorization approach described in RouterBench (Hu et al., 2024). The router receives an embedded representation of each query and predicts which model—Qwen 2.5 7B Instruct or Deepseek R1 distilled—is more likely to answer correctly. Routing decisions are made by projecting the query embedding into a shared latent space and computing scores with each model’s profile, which is learned during training.

This approach leverages the verifiable nature of mathematical reasoning tasks, ensuring that routing decisions are transparent and interpretable. By focusing on binary correctness, we aim to extract a clear signal for routing, reducing opacity and increasing trust in the system’s decisions.

2.3 Inspect Router Implementation

There are not many evaluations using routers, and LLM routing is a very young field that needs to demonstrate quantitative results. To address this, we implemented a routing method for the Inspect Framework that works with local models.

Our implementation involved creating a routing class within the Inspect Framework that supports routed inference for local models. We leveraged the VLLM engine to enable efficient inference and developed a Router Model API to allow for concurrent inference across multiple models. Routing decisions are made dynamically at

inference time, enabling the system to select the most appropriate model for each query based on its characteristics.

This approach enables local inference of models organized by the router and allows any of the Inspect Framework’s evaluation modules to be run with a routed system. As a result, we can benchmark and evaluate non-monolithic, multi-model systems, furthering the accessibility and transparency of model routing research.

2.4 Code

3. Results

Model Performance Analysis Tables

Table 1: Model Performance on GSM8K Dataset

Model	Accuracy	Total Tokens	Tokens per Problem*
DeepSeek-R1-Distill-Gwen-7B	0.908	2,868	36.4
Gwen 2.5-7B-Instruct	0.830	2,466	31.3

Table 1. Performance comparison of reasoning and non-reasoning models on the GSM8K mathematical reasoning dataset. DeepSeek-R1-Distill demonstrates superior accuracy with moderate token overhead. *Assuming ~79 problems in GSM8K test set.

Table 2: Model Performance on AIME Dataset

Model	Accuracy	Total Tokens	Tokens per Problem*
DeepSeek-R1-Distill-Qwen-7B	0.610	10,679	356.6
Qwen2-7B-Instruct	0.066	7,915	263.8

Table 2. Performance comparison on the AIME (American Invitational Mathematics Examination) dataset, representing high-difficulty mathematical reasoning problems. The specialized reasoning model shows dramatic improvement over the base instruction-tuned model, with a $9.2\times$ accuracy improvement despite 35% higher token usage. *Assuming ~ 30 problems in AIME test set.

Key Findings

Cost-Accuracy Trade-off Analysis:

- **GSM8K:** DeepSeek-R1 achieves 9.4% higher accuracy with 16.3% more tokens (cost ratio: $1.16\times$)
- **AIME:** DeepSeek-R1 achieves 824% higher accuracy with 34.9% more tokens (cost ratio: $1.35\times$)

Efficiency Metrics:

- **GSM8K Efficiency** (Accuracy/Token Ratio): DeepSeek-R1 = 24.9, Qwen2 = 26.5
- **AIME Efficiency** (Accuracy/Token Ratio): DeepSeek-R1 = 17.1, Qwen2 = 2.5

The results demonstrate that for high-difficulty reasoning tasks (AIME), the marginal cost of using reasoning models yields substantial accuracy gains, while for moderate-difficulty tasks (GSM8K), the efficiency trade-off is more balanced.

4. Discussion

4.1 Interpretation of Results

Current Challenges in AI Routing and Model Selection

- Our matrix factorization approach addresses several fundamental challenges in contemporary AI model routing systems. Traditional routing mechanisms often rely on heuristic rules or simple performance metrics that fail to capture the nuanced relationships between query characteristics and model capabilities. The matrix factorization framework we present fundamentally reframes this problem by learning compact representations that encode both model characteristics and question semantics in a unified latent space.
- The dramatic performance differences observed between GSM8K and AIME datasets illuminate a critical insight: **routing decisions must be context-aware rather than performance-agnostic**. While the reasoning model (DeepSeek-R1-Distill) showed only modest improvements on GSM8K (9.4% accuracy gain for 16.3% cost increase), it demonstrated transformative performance on AIME (824% accuracy improvement for 35% cost increase). This finding challenges the prevailing assumption in current routing systems that model selection should be based solely on aggregate performance metrics.
- Our model-specific question embedding innovation represents a significant departure from generic representation approaches. By allowing each model to "see" questions through its own representational lens, we move beyond the limitations of universal embeddings that may miss model-specific nuances. This approach aligns with recent work on mechanistic interpretability, suggesting that different models may indeed process identical inputs through fundamentally different computational pathways.

Broader Expert Orchestration Architecture Goals

The results strongly support the core tenets of Expert Orchestration Architecture in several key dimensions:

- **Specialization Over Generalization:** Our findings demonstrate that intelligent orchestration of specialized models can achieve superior cost-efficiency compared to monolithic approaches. The 53.8% cost reduction with minimal accuracy loss (4.3%) provides empirical evidence that the "right-tool-for-the-job" paradigm offers tangible benefits over "one-size-fits-all" architectures.
- **Democratic Access to Advanced Capabilities:** Perhaps most significantly, our orchestration framework demonstrates how advanced reasoning capabilities can be made accessible to cost-sensitive applications. The ability to achieve 95.7% of premium model performance at less than half the cost fundamentally alters the economics of AI deployment, potentially expanding access to sophisticated AI assistance across different economic contexts.

4.2 Limitations & Future Work

Technical Limitations of Current Approach

- **Fixed Model Pool Constraint:** The current matrix factorization architecture requires pre-defining the set of available models during training. Adding new models necessitates complete retraining of the embedding system, limiting the framework's adaptability to the rapidly evolving model landscape. This represents a significant practical limitation as new models are released frequently and existing models are updated or deprecated.
- **Binary Correctness Oversimplification:** The reconstruction objective focuses solely on binary correctness predictions, potentially missing nuanced aspects of response quality such as explanation clarity, reasoning process quality, or stylistic preferences. This limitation may result in suboptimal routing decisions for applications where response characteristics beyond correctness matter.
- **Question Embedding Scalability:** The model-specific question embedding approach, while innovative, faces scalability challenges. Storing separate embeddings for each model-question pair scales quadratically with both model count and question diversity, potentially becoming prohibitive for large-scale deployments.

Scalability Concerns

- **Computational Overhead:** While the routing decision itself is efficient (milliseconds), the training process requires substantial computational resources, particularly when computing model-specific embeddings. The quadratic scaling with model count poses challenges for ecosystems with hundreds of available models.
- **Real-time Adaptation:** The current approach cannot adapt to changing model performance characteristics without retraining. Models may improve or degrade over time due to updates, fine-tuning, or infrastructure changes, but the routing system remains static until explicitly retrained.

Areas for Improvement

- **Dynamic Embedding Updates:** Future work should explore mechanisms for updating model embeddings without full retraining, potentially through incremental learning approaches or transfer learning techniques that can adapt to new models based on limited evaluation data.
- **Multi-Objective Optimization:** The current system optimizes primarily for accuracy, but real-world deployments require consideration of multiple objectives including latency, cost, energy consumption, and user satisfaction. Developing multi-objective routing frameworks represents a critical area for improvement.
- **Uncertainty Quantification:** Improving confidence calibration through better uncertainty quantification methods could significantly enhance routing reliability. Techniques from ensemble methods or **conformal prediction** may provide more reliable confidence estimates.
- **Adversarial Robustness:** The routing system's vulnerability to adversarial inputs that manipulate confidence scores represents a significant security concern requiring investigation and mitigation strategies.

5. Conclusion

The Expert Orchestration Architecture vision promises increased democratization, transparency, safety, and efficiency for AI models. These promises all hinge on the ability of a model router to find the best model for a given query. To the extent that model routers unnecessarily route queries that require reasoning to non-reasoning models or queries that do not require reasoning to expensive reasoning models, specialized models that would have been accurate and cost-effective for a task may find it more difficult to stand out. Furthermore, improving the cost-performance trade-off within an Expert Orchestration Architecture solution will accelerate adoption of the Expert Orchestration Architecture solution, thereby accelerating its transparency, safety, and democratization benefits.

Reason-based routing helps to ensure that specialized reasoning models are used when their reasoning capabilities are called for, and that specialized non-reasoning models are given tasks for which more costly reasoning models (including, e.g., more general-purpose models) are not needed.

Avoiding unnecessarily routing queries to reasoning models may also improve safety outcomes by avoiding the use of capacities that are unneeded.

References

He, Yang, et al. "Self-Route: Automatic Mode Switching via Capability Estimation for Efficient Reasoning." *arXiv preprint arXiv:2505.20664* (2025).

<https://arxiv.org/pdf/2505.20664>

Ong, Isaac, et al. "RouteLLM: Learning to Route LLMs from Preference Data." *The Thirteenth International Conference on Learning Representations*. 2024.

<https://openreview.net/pdf?id=8sSqNntaMr>

Yang, An, et al. "Qwen3 technical report." *arXiv preprint arXiv:2505.09388* (2025).

<https://arxiv.org/pdf/2505.09388>

Pan, Zhihong, et al. "Route to Reason: Adaptive Routing for LLM and Reasoning Strategy Selection." *arXiv preprint arXiv:2505.19435* (2025).

<https://arxiv.org/abs/2505.19435v1>

Manvi, Rohin, Anikait Singh, and Stefano Ermon. "Adaptive inference-time compute: LLMs can predict if they can do better, even mid-generation." *arXiv preprint arXiv:2410.02725* (2024).

<https://arxiv.org/abs/2410.02725>

Hu, Qitian Jason, et al. "Routerbench: A benchmark for multi-llm routing system." *arXiv preprint arXiv:2403.12031* (2024).

<https://arxiv.org/abs/2403.12031>

<https://learn.microsoft.com/en-us/azure/ai-services/openai/concepts/model-router>