# When to Reason: Semantic Router for vLLM

**Chen Wang**
IBM Research
Yorktown Heights, NY, 10598
Chen.Wang1@ibm.com

**Xunzhuo Liu**
Tencent
bitliu@tencent.com

**Yuhan Liu**
University of Chicago
yuhanl@uchicago.edu

**Yue Zhu**
IBM Research
Yorktown Heights, NY, 10598
yue.zhu@ibm.com

**Xiangxi Mo**
UC Berkeley
xmo@berkeley.edu

**Junchen Jiang**
University of Chicago
junchenj@uchicago.edu

**Huamin Chen**
Red Hat
Boston, MA, 02210
hchen@redhat.com

## Abstract

Large Language Models (LLMs) demonstrate substantial accuracy gains when augmented with reasoning modes such as chain-of-thought and inference-time scaling. However, reasoning also incurs significant costs in inference latency and token usage, with environmental and financial impacts, which are unnecessary for many simple prompts. We present a semantic router that classifies queries based on their reasoning requirements and selectively applies reasoning only when beneficial. Our approach achieves a $10.2$ percentage point improvement in accuracy on the MMLU-Pro benchmark while reducing response latency by $47.1\%$ and token consumption by $48.5\%$ compared to direct inference with vLLM. These results demonstrate that semantic routing offers an effective mechanism for striking a balance between accuracy and efficiency in open-source LLM serving systems.

## 1 Introduction

Large Language Models (LLMs) achieve notable accuracy gains when augmented with advanced inference techniques such as chain-of-thought reasoning or inference-time scaling. Yet, these benefits come at substantial computational and energy costs, particularly when reasoning is applied indiscriminately. Prior studies [22] show that while reasoning improves performance on complex tasks, it is unnecessary for many straightforward queries. This tension makes selective reasoning a central challenge for practical LLM systems.

Recent frameworks such as LangChain/LangGraph [11] and DSPy [9] enable modular routing policies, but they require manual configuration and are tied to higher-level stacks. In contrast, open-source inference engines like vLLM [10]—the de facto standard for high-throughput LLM serving—deliver efficient inference but lack native semantic routing. Related systems (e.g., llm-d [12], Production Stack [16]) provide lightweight routing but do not support fine-grained control over reasoning. Consequently, developers using vLLM's APIs avoid vendor lock-in but remain without integrated mechanisms for adaptive reasoning.

To address this gap, we propose a semantic router for open-source inference engines. Our system integrates with vLLM and cloud-native routing frameworks (Envoy, ext_proc), classifies queries

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: MLForSys.

by intent, and selectively applies reasoning only when beneficial. Experiments on the MMLU-Pro benchmark across 14 domains show that our router achieves higher accuracy while reducing latency and token usage by nearly half.

Our contributions are as follows:

- We identify the need for semantic routing in open-source inference engines to enable reasoning-aware inference.
- We design, implement, and open-source [2] a high-performance and scalable semantic router that integrates with vLLM and Envoy/ext_proc for fine-grained reasoning control, accelerating Cloud Native ecosystem integration.
- We evaluate the semantic router on the MMLU-Pro benchmark and show that it improves accuracy by $10.2$ percentage points while reducing response latency by $47.1\%$ and token consumption by $48.5\%$ compared to direct vLLM inference.

## 2 Background

### 2.1 Routers in LLM Systems

Recent work has explored the use of routers to improve the efficiency and accuracy of LLM inference by dynamically deciding how queries should be handled. FrugalGPT [5] achieves up to 98% cost reduction by learning which combinations of LLMs to invoke for different queries, leveraging prompt adaptation, approximation, and cascaded model selection across commercial APIs. RouteLLM [15] similarly trains router models to choose between stronger and weaker LLMs during inference, guided by human preference data and augmentation, yielding substantial cost savings while maintaining accuracy across benchmarks such as MT Bench, MMLU, and GSM8K. These approaches highlight the promise of router-based techniques for improving inference performance, but they remain focused on model-level routing.

### 2.2 The Need for Selective Reasoning

While advanced reasoning strategies such as Chain-of-Thought (CoT) prompting can improve accuracy, recent studies highlight that reasoning is not universally beneficial and often incurs substantial computational overhead. Wilhelm et al. [5] demonstrate that CoT can increase energy costs by up to 150 times while offering little benefit for knowledge-based tasks. Similarly, Aggarwal et al. find that LLMs frequently "overthink" simple queries and "underthink" complex ones [1], leading to inefficiencies. Meta-analyses by Sprague et al. [17] and the original CoT work by Wei et al. [20] further establish that CoT primarily improves performance on math and logic tasks, with limited gains elsewhere and even degraded accuracy in smaller models. To mitigate these inefficiencies, recent frameworks [6, 24, 21] introduce adaptive reasoning strategies that dynamically regulate reasoning depth, reducing token usage while maintaining accuracy.

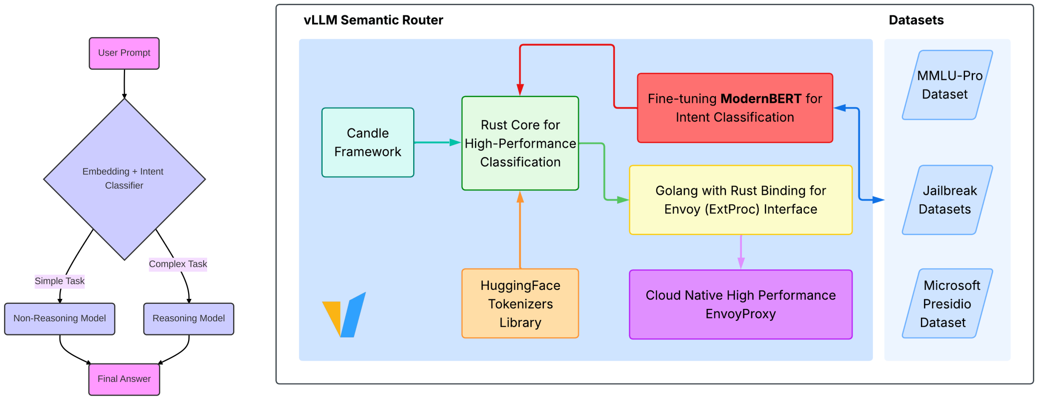
### 2.3 Semantic Routing

A semantic router refers to an emerging class of request forwarding systems for LLM inference, in which routing decisions are guided by the semantic meaning of the input rather than by explicit keywords or manually defined rules [13, 3]. The router operates by encoding both user queries and candidate routing utterances into high-dimensional embeddings [23] that capture contextual meaning, and then selecting the target pathway with the highest semantic similarity, typically measured using metrics such as cosine distance. Semantic routing provides a lightweight and efficient mechanism for query-level control, making it a promising foundation for reasoning-aware routing.

## 3 System Design

### 3.1 System Design

Our system integrates a semantic router with a reasoning mode selector to dynamically balance efficiency and accuracy in LLM inference. As shown in Figure 1a, the process begins by encoding the


User Prompt
Embedding + Intent Classifier
Simple Task
Complex Task
Non-Reasoning Model
Reasoning Model
Final Answer
vLLM Semantic Router
Datasets
Candle Framework
Rust Core for High-Performance Classification
Fine-tuning ModernBERT for Intent Classification
Golang with Rust Binding for Envoy (ExtProc) Interface
HuggingFace Tokenizers Library
Cloud Native High Performance EnvoyProxy
MMLU-Pro Dataset
Jailbreak Datasets
Microsoft Presidio Dataset

(a) Workflow

(b) System architecture

Figure 1: Overview of the proposed intent-aware semantic router. (a) Workflow of classification and routing; (b) system architecture.

user prompt into high-dimensional semantic embeddings, which capture the contextual meaning of the input. These embeddings are then processed by an intent classifier that determines whether the prompt corresponds to a simple factual query or a reasoning-intensive task. Based on this classification, the router directs the input to the most suitable inference pathway: lightweight inference with a non-reasoning model for simple tasks, or reasoning inference with a chain-of-thought–enabled model for complex queries. Finally, the outputs are unified into a final response. Unlike prior router approaches such as FrugalGPT and RouteLLM, which primarily operate at the model-selection level to trade off accuracy and cost, our design focuses on semantic intent–based routing and selectively invoking reasoning. This enables adaptive reasoning where costly step-by-step inference is applied only when beneficial, while maintaining low latency and efficiency for straightforward queries.

### 3.2 Implementation

The implementation of our intent-aware semantic router integrates three key modules—ModernBERT fine-tuning for intent classification, a Rust-based high-performance classification core, and Golang–Rust bindings for Envoy integration—into a unified architecture, as illustrated in Figure 1b.

#### 3.2.1 ModernBERT Fine-tuning for Intent Classification

We fine-tune ModernBERT [19] —fast, memory-efficient, supports long contexts, and achieves high accuracy by incorporating modern LLM innovations like RoPE and FlashAttention—for multi-task intent classification. The training pipeline ingests three datasets: MMLU-Pro [18] (~12K academic samples across ~14 domains), Microsoft Presidio [14] (~50K token-level PII examples), and jailbreak security datasets [4]. The classification pipeline can use either CPU or GPU for real-time inline inference and simplifies the runtime environment resource requirements.

#### 3.2.2 Rust Core for High-Performance Classification

The classification engine is implemented in Rust using Hugging Face's Candle framework [8], which enables efficient, zero-copy tensor workflows, SIMD acceleration, and optimized memory usage. It runs multi-stage parallel inference—category classification, PII detection, and jailbreak detection—leveraging Rust's ownership model for thread safety. The pipeline batches requests and utilizes Hugging Face Tokenizers for fast tokenization, supports large context window, and chains multiple classification tasks, sustaining highly concurrent requests on commodity hardware without using expensive GPUs.

### 3.2.3 Golang + Rust (via CGO) for Cloud-Native Envoy Integration

We wrap the Rust-based classification core in a Golang layer using CGO bindings to support Envoy's External Processing (ext_proc) filter interface [7]. Envoy intercepts HTTP requests and forwards them via gRPC to the external processor, which applies real-time classification and routing decisions before responses reach backend services. The CGO layer is statically linked, minimizing runtime overhead while enabling seamless integration with Kubernetes, service meshes, and API gateway patterns. Such design pattern facilitates Cloud Native ecosystem adoption.

## 4 Evaluation

We evaluate our semantic router on an NVIDIA L4 GPU using the Qwen/Qwen3-30B-A3B model served by vLLM v0.10.1 with tensor parallelism degree 4. The evaluation is conducted on the MMLU-Pro benchmark across 14 domains, measuring accuracy, token usage, and latency. For direct vLLM comparison, we run the same model under six execution modes—neutral reasoning (NR) and explicit chain-of-thought (XC), each with reasoning enabled or disabled configurations.

Figure 2 breaks down accuracy by the 14 MMLU-Pro domains for all execution modes (NR/XC with `reason_on`, `reason_off`, and `base`), along with our semantic router. Across the majority of categories, the router leads in reasoning-heavy domains and remains competitive in knowledge-centric areas, indicating that selective reasoning does not sacrifice accuracy on fact-focused tasks while delivering benefits where structured reasoning is essential.

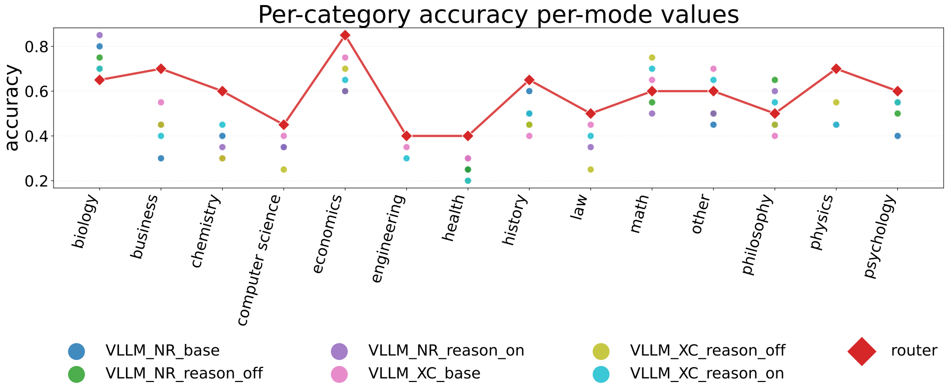


Figure 2: Per-category accuracy across 14 MMLU-Pro domains for direct vLLM modes and our semantic router.

Table 1 summarizes the aggregate performance metrics comparing our semantic router against direct vLLM inference. Overall, the semantic router improves accuracy by 10.24 points while cutting latency by 47.1% and token usage by 48.5%.

Table 1: Overall performance of semantic router versus direct vLLM inference on MMLU-Pro.

| **Method** | **Avg. Accuracy** | **Avg. Latency (s)** | **Avg. Tokens** |
|---|---|---|---|
| Semantic Router | **58.57%** | **13.09** | **887.5** |
| Direct vLLM | 48.33% | 24.76 | 1,722.1 |
| **Improvement** | **+10.24pp** | **-47.1%** | **-48.5%** |

Our evaluation shows that the semantic router delivers substantial efficiency gains while improving overall accuracy, achieving a statistically significant 10.24 percentage point increase ($p < 0.01$) with 48.5% fewer tokens and 47.1% lower latency. The router is particularly effective in knowledge-intensive domains such as business and economics, where accuracy improvements exceed 20 percentage points, while performance in technical areas like engineering and computer science remains more challenging. Mixed results in reasoning-heavy domains (e.g., mathematics and biology) highlight opportunities for refining routing strategies. Overall, the router demonstrates robust improvements across 11 of 14 domains, underscoring its ability to match queries to appropriate reasoning strategies. These results suggest that semantic routing offers a practical path toward more accurate and cost-efficient LLM inference in production settings.

## 5 Conclusion

This paper presented a semantic router that dynamically selects between reasoning and non-reasoning strategies to optimize large language model inference. Evaluation on MMLU-Pro shows that the router improves accuracy by more than 10 percentage points while reducing token usage and latency by nearly 50%. The approach is particularly effective in knowledge-intensive domains such as business, economics, and physics, though challenges remain in technical and reasoning-heavy areas. Integrated with vLLM, the router demonstrates that semantic routing is a practical and efficient solution for real-world inference serving.

## References

[1] Pranjal Aggarwal, Seungone Kim, Jack Lanchantin, Sean Welleck, Jason Weston, Ilia Kulikov, and Swarnadeep Saha. Optimalthinkingbench: Evaluating over and underthinking in llms. *arXiv preprint arXiv:2508.13141*, 2025.

[2] Anonymous. vLLM Semantic Router. `https://vllm-semantic-router.netlify.app/`. Accessed: 2025-08-29.

[3] Aurelio.ai. Semantic router. `https://www.aurelio.ai/semantic-router`, 2025. Accessed: 2025-08-29.

[4] Patrick Chao, Edoardo Debenedetti, Alexander Robey, Maksym Andriushchenko, Francesco Croce, Vikash Sehwag, Edgar Dobriban, Nicolas Flammarion, George J Pappas, Florian Tramer, et al. Jailbreakbench: An open robustness benchmark for jailbreaking large language models. *Advances in Neural Information Processing Systems*, 37:55005–55029, 2024.

[5] Lingjiao Chen, Matei Zaharia, and James Zou. Frugalgpt: How to use large language models while reducing cost and improving performance. *arXiv preprint arXiv:2305.05176*, 2023.

[6] Qiguang Chen, Dengyun Peng, Jinhao Liu, HuiKang Su, Jiannan Guan, Libo Qin, and Wanxiang Che. Aware first, think less: Dynamic boundary self-awareness drives extreme reasoning efficiency in large language models. *arXiv preprint arXiv:2508.11582*, 2025.

[7] Envoy Proxy Contributors. External processing filter (ext_proc). `https://www.envoyproxy.io/docs/envoy/latest/configuration/http/http_filters/ext_proc_filter`, 2025. Accessed: 2025-08-29.

[8] Hugging Face. Candle: A minimalist machine learning framework for Rust. `https://github.com/huggingface/candle`, 2023. Accessed: 2025-08-29.

[9] Omar Khattab, Arnav Singhvi, Paridhi Maheshwari, Zhiyuan Zhang, Keshav Santhanam, Sri Vardhamanan, Saiful Haq, Ashutosh Sharma, Thomas T Joshi, Hanna Moazam, et al. Dspy: Compiling declarative language model calls into state-of-the-art pipelines. In *ICLR*, 2024.

[10] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th symposium on operating systems principles*, pages 611–626, 2023.

[11] LangChain Documentation. How to route between sub-chains. `https://js.langchain.com/docs/how_to/routing`, 2025. [Online; accessed DD-Month-YYYY].

[12] llm-d Contributors (including Red Hat, Google, IBM Research, CoreWeave, NVIDIA, AMD, and others). llm-d: A kubernetes-native high-performance distributed llm inference framework. `https://llm-d.ai/`, 2025. Open-source project: Kubernetes-native distributed LLM inference stack built on vLLM with intelligent scheduling and prompt-aware routing.

[13] Dimitrios Michael Manias, Ali Chouman, and Abdallah Shami. Semantic routing for enhanced performance of llm-assisted intent-based 5g core network management and orchestration. In *GLOBECOM 2024-2024 IEEE Global Communications Conference*, pages 2924–2929. IEEE, 2024.

[14] Microsoft. Presidio research: Data science utilities, evaluation tools and synthetic data generation for presidio. `https://github.com/microsoft/presidio-research`, 2023. Accessed: 2025-08-29.

[15] 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.

[16] Production Stack Contributors. Production stack: Scalable inference infrastructure with vllm. `https://docs.vllm.ai/projects/production-stack/en/latest/`, 2025. Open-source project documentation, vLLM Production Stack.

[17] Zayne Sprague, Fangcong Yin, Juan Diego Rodriguez, Dongwei Jiang, Manya Wadhwa, Prasann Singhal, Xinyu Zhao, Xi Ye, Kyle Mahowald, and Greg Durrett. To cot or not to cot? chain-of-thought helps mainly on math and symbolic reasoning. *arXiv preprint arXiv:2409.12183*, 2024.

[18] Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, et al. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *Advances in Neural Information Processing Systems*, 37:95266–95290, 2024.

[19] Benjamin Warner, Antoine Chaffin, Benjamin Clavié, Orion Weller, Oskar Hallström, Said Taghadouini, Alexis Gallagher, Raja Biswas, Faisal Ladhak, Tom Aarsen, Nathan Cooper, Griffin Adams, Jeremy Howard, and Iacopo Poli. Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference, 2024.

[20] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

[21] Zihao Wei, Liang Pang, Jiahao Liu, Jingcheng Deng, Shicheng Xu, Zenghao Duan, Jingang Wang, Fei Sun, Xunliang Cai, Huawei Shen, et al. Stop spinning wheels: Mitigating llm overthinking via mining patterns for early reasoning exit. *arXiv preprint arXiv:2508.17627*, 2025.

[22] Patrick Wilhelm, Thorsten Wittkopp, and Odej Kao. Beyond test-time compute strategies: Advocating energy-per-token in llm inference. In *Proceedings of the 5th Workshop on Machine Learning and Systems*, pages 208–215, 2025.

[23] Jiarui Zhang, Xiangyu Liu, Yong Hu, Chaoyue Niu, Fan Wu, and Guihai Chen. Query routing for retrieval-augmented language models. *arXiv preprint arXiv:2505.23052*, 2025.

[24] Yekun Zhu, Guang Chen, and Chengjun Mao. Think in blocks: Adaptive reasoning from direct response to deep reasoning. *arXiv preprint arXiv:2508.15507*, 2025.

## Appendix A. Additional Per-Category Results

In addition to the per-category accuracy results reported in Figure 3, we include two supplementary breakdowns that highlight the efficiency benefits of semantic routing.

The per-category breakdowns in Figures 4 and 5 confirm that the semantic router consistently improves efficiency across domains. In terms of token usage, the router reduces average consumption by nearly half relative to direct vLLM execution modes, with especially pronounced savings in knowledge-intensive subjects such as history, law, and health, where reasoning is rarely required. Similarly, the latency results show that the router sustains substantially faster response times across most categories, cutting delays by over 40% even in reasoning-sensitive areas like mathematics and physics. These results demonstrate that semantic routing not only improves aggregate efficiency but also achieves robust per-domain benefits, delivering faster and cheaper inference without sacrificing accuracy.

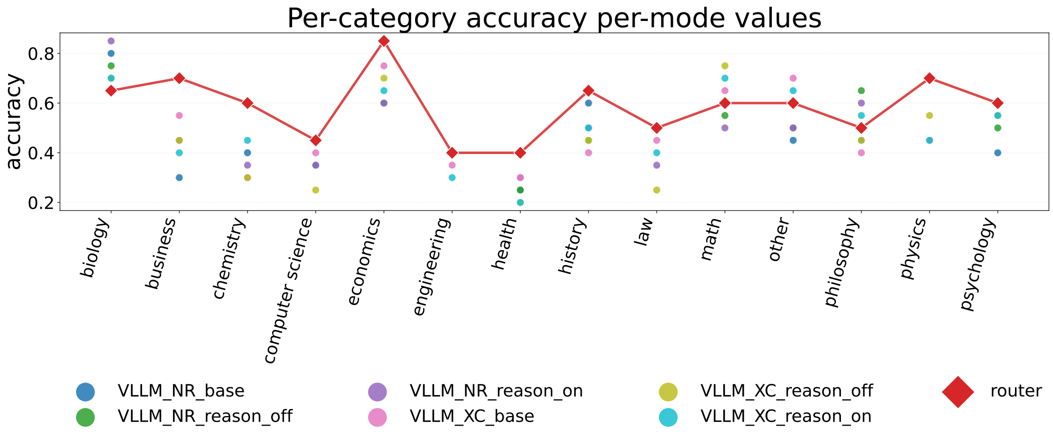


Figure 3: Per-category accuracy across all inference modes on MMLU-Pro.

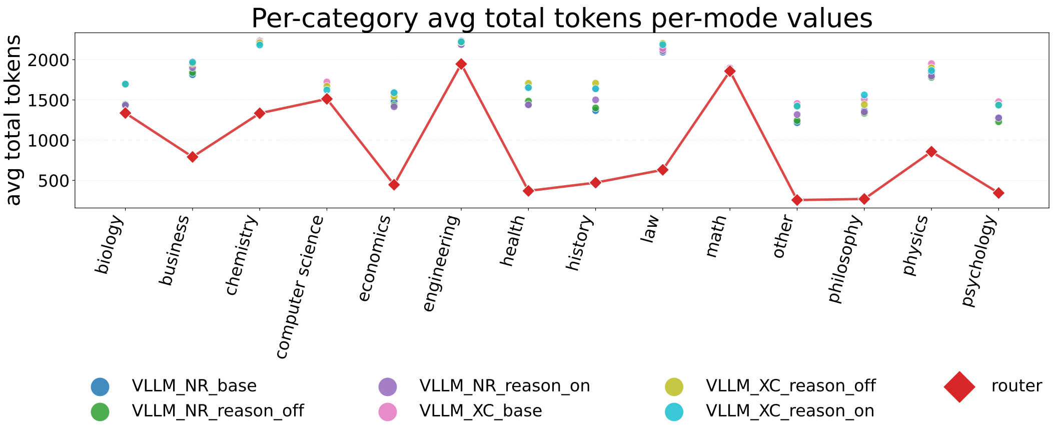


Figure 4: Per-category average total tokens across all inference modes on MMLU-Pro. The semantic router consistently achieves the lowest token usage, reducing overhead in knowledge-centric domains (e.g., history, law, health) while remaining competitive in reasoning-heavy areas such as math and physics.

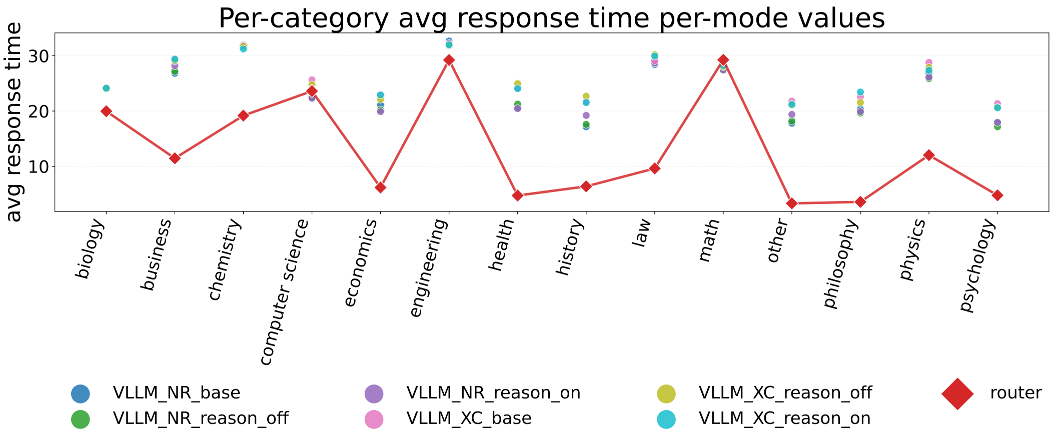


Figure 5: Per-category average response latency across all inference modes on MMLU-Pro. The semantic router reduces latency substantially compared to direct vLLM modes, particularly in domains with shorter factual queries (e.g., history, philosophy). Even in complex reasoning categories, the router sustains lower response times by avoiding unnecessary reasoning overhead.