Dynaroute: Dynamic Model Routing via Task Profiling and Cost Tiers

Anonymous EMNLP submission

Abstract

The rapid advancements in Large Language Models (LLMs) have led to a diverse landscape of models with varying capabilities and associated costs. No single LLM is optimal for all tasks, necessitating intelligent routing systems that can dynamically select the most appropriate model for a given input to balance performance and operational expense. In this study, we propose a novel benchmark-driven LLM routing framework designed to achieve a practical balance between task-specific performance and cost. As opposed to previous studies' frameworks, such as HybridLLM, RouteLLM, and LLMProxy, which often focus on binary classifiers for query assessment, our multi-stage system employs explicit task profiling using a lightweight classifier LLM to determine not only the query's category but also a more granular, multi-level difficulty. A key differentiating aspect is our tiered cost-performance model selection strategy, which systematically buckets models into cost percentiles and then selects the best-performing model within the appropriate tier for the predicted task profile, offering a more structured approach to balancing cost and performance. We evaluate the framework using three routing configurations. The Optimum router consistently achieves performance comparable to or exceeding the best individual models on specific tasks, but at significantly lower total costs.

1 Introduction

Generative AI, particularly large language models (LLMs), has seen rapid advancements in recent years; good models are becoming cheaper, and cheap models are becoming good. Models such as GPT-4o (OpenAI, 2024a), o1 (OpenAI, 2024b), DeepSeek-R1 (DeepSeek-AI, 2025) have demonstrated remarkable capabilities in natural language processing tasks, including open-domain dialogue, question answering, and code generation.

Despite these advancements, a critical challenge persists: no single LLM is universally optimal across all tasks and applications, especially when balancing performance against operational costs and latency (Hu et al., 2024; Huang et al., 2024). This inherent limitation has spurred significant research into LLM routing systems. These systems dynamically select the most suitable model from a pool of candidates for a given input, aiming to optimize objectives such as accuracy, computational efficiency, or latency, thereby aligning model choice with the desired capability-resource trade-off (Varangot-Reille et al., 2025). Consequently, routing is now recognized as a promising paradigm for achieving model-level scaling up (Huang et al., 2024). Existing LLM routing methods can be broadly categorized. One key distinction is between non-predictive methods, which often involve generating responses from multiple models before selection, and predictive methods, which aim to select the optimal LLM before generation."

1.1 Non-Predictive and Cascading Approaches

These methods typically involve querying models sequentially or in parallel. Cascading approaches, like **FrugalGPT** (Chen et al., 2023; Ding et al.; Feng et al.), often start with smaller, cheaper models and escalate to more capable ones if the initial response quality (sometimes inferred by a "judger" regression model) is insufficient. **AutoMix** (Aggarwal et al., 2024) employs a self-verification step before potentially escalating. Other non-predictive strategies might generate responses from multiple LLMs and then select or combine the best one (Jiang et al., 2023). While these can be effective, the need to query multiple models can increase costs and latency (Tay et al., 2022; Wang et al., 2024). Our approach, on the other hand, clearly falls into the *predictive routing* category. It aims to select a single, most appropriate LLM based on prior classification and benchmark mapping, avoid-

ing the overhead of multiple generations inherent in many non-predictive and cascading systems.

1.2 Classification/Regression-Based:

HybridLLM (Ding et al.) uses a binary classifier for query difficulty. **RouteLLM** (Ong et al., 2024) trains various routers (similarity-weighted, matrix factorization, BERT, Causal LLM) on human preference data. **LLMProxy** (Martin et al., 2024) proposes model selection as one of its optimizations, potentially using LLM-powered heuristics. (Shekhar et al., 2024) propose quality prediction without invocation using LP-based routines. Domain-based routing (Jain et al., 2024; Liu et al., 2024) classifies queries into domains to match with expert models. Our method shares similarities with classification-based routing. We use Qwen 2.5 3B as a classifier to assign tasks to predefined categories (Law, Business, Computer Science, etc.) and one of three difficulty levels. This explicit, fine-grained categorization before model selection is a key aspect. Unlike systems that predict general query difficulty or directly regress performance, our classifier provides a structured task profile. The subsequent mapping of this profile to benchmark performance (MMLU-Pro, GPQA Diamond, HumanEval, etc.) to inform model choice is a data-driven, rule-based step that leverages empirical evidence.

Graph-Based Methods: **GraphRouter** (Feng et al.) uses a GNN to model interactions between tasks, queries, and LLMs, predicting optimal LLMs via edge predictions and offering generalization to new LLMs.

RL and Bandit-Based Methods: **PickLLM** (Sikeridis et al., 2024) uses RL for dynamic routing based on cost, latency, and accuracy. **MixLLM** (Wang et al., 2024) applies a contextual bandit framework with continual learning and tag-enhanced embeddings.

1.3 Cost Optimization Strategies

A central theme is optimizing cost. Many systems, like **RouterBench** (Hu et al., 2024), incorporate a "willingness-to-pay" (WTP) framework. **LLMProxy** (Martin et al., 2024) focuses on cost-saving through model selection, context management, and caching. (Shekhar et al., 2024) emphasize cost optimization through LP-based routines and token reduction. Our method integrates cost explicitly and structurally through a percentile-based bucketing of models. We divide our tested models (Qwen 2.5, gemma3, GPT 4.1 nano/mini, Gemini 2.0 Flash Lite, Llama 4 Scout, etc.) into three cost tiers. The routing logic then directly links these cost tiers to our predefined difficulty levels: the cheapest 33.3 percentile models are considered for Level 1 tasks (Simple/Factual), the next 33.3 percentile for Level 2 (Moderate/Standard), and the overall best-performing models (often the most expensive) for Level 3 (Complex/Advanced), always selecting the top performer within that tier for the given category. This explicit, tiered cost-performance mapping based on difficulty is a distinct and pragmatic strategy.

1.4 Benchmarking and Evaluation Frameworks

The field relies on standard LLM benchmarks (MMLU, HumanEval, etc.) and increasingly on dedicated routing benchmarks. **ROUTERBENCH** (Hu et al., 2024) and **RouterEval** (Huang et al., 2024) provide comprehensive datasets and frameworks to evaluate different routing strategies. RouterEval, for instance, highlights the "model-level scaling up" phenomenon where performance improves with more candidate LLMs and a capable router. **TaskEval** (Tambon et al.) focuses on assessing task difficulty itself using IRT, showing discrepancies between human and LLM perception of difficulty. **SLaM** (Irugalbandara et al.) provides tools for cost-benefit analysis when replacing proprietary LLMs with open-source SLMs. We leverage a wide array of benchmarks (MMLU-Pro, GPQA Diamond, LiveCodeBench, SciCode, HumanEval, MATH-500, AIME 2024, Multilingual Index, LegalBench) not just for final evaluation, but as an integral part of our routing logic. The performance of models on these benchmarks, within specific task categories, directly informs which model is chosen from the appropriate cost/difficulty tier.

1.5 Summary and Positioning of Our Work

Our proposed method offers a structured, interpretable, and empirically-grounded approach to LLM routing. It stands out by:

1. **Explicit Task Profiling:** In contrast to studies like HybridLLM (Ding et al.), RouteLLM (Ong et al., 2024), or LLMProxy (Martin et al., 2024), which often employ binary classifiers, our approach uses a classifier LLM (Qwen 2.5 3B) for fine-grained task categorization and

multi-level difficulty assignment. This yields a more nuanced understanding of prompt characteristics beyond simple hard/easy distinctions.

2. **Benchmark-Driven Model-Task Matching:** Instead of relying solely on high-level heuristics, our system systematically maps detailed task profiles to empirical model performance across diverse, relevant benchmarks. This grounds model selection in observed capabilities for specific task types, facilitating the identification of the best-performing model from the candidate pool for a given profile.

3. **Tiered Cost-Performance Optimization:** While systems like HybridLLM (Ding et al.) and RouteLLM (Ong et al., 2024) route queries to different cost-level models based on difficulty, and cascading approaches like FrugalGPT (Chen et al., 2023) escalate through cost tiers, our method introduces distinct, preemptive cost-percentile bucketing. This explicitly links predefined model cost tiers to predicted task difficulty levels before invocation, systematically ensuring a practical performance-expenditure balance by selecting the optimal performer within an appropriate, pre-defined cost bracket for the task's profile.

4. **Transparency and Simplicity:** Unlike complex learned routers, such as graph-based methods (e.g., GraphRouter (Feng et al.)) or dynamic RL systems (e.g., PickLLM (Sikeridis et al., 2024)), which can be resource-intensive, our approach offers high interpretability. Its rule-based nature post-classification and benchmark mapping potentially simplifies deployment and allows for easier diagnosis and adjustment.

2 Methodology

The proposed system implements a multi-stage, benchmark-driven framework for intelligent LLM routing, prioritizing a balance between task-specific performance and operational cost. The overall architecture of this framework is depicted in Figure 1. The methodology is comprised of four key stages: (1) explicit task profiling using a classifier LLM, (2) offline benchmark-driven model recommendation generation, (3) dynamic online prompt routing, and (4) a comprehensive evaluation framework.

2.1 Explicit Task Profiling

At the core of the routing intelligence is a task profiling sub-system designed to understand the nature and complexity of incoming user prompts.

2.1.1 Classifier Model Selection

A lightweight yet effective LLM, Qwen 2.5 3B, was selected for task classification. This decision was informed by an empirical evaluation across multiple candidate models (including various sizes of Qwen, Gemma, and proprietary models like GPT 4.1 nano/mini and Gemini 2.0 Flash Lite) on a diverse corpus of 600 questions spanning GPQA, MMLU, MMLU-Pro, Math-500, Humanity's Last Exam, and LiveCodeBench. Qwen 2.5 3B demonstrated a leading category accuracy of 78.28 (Table 1) and a level accuracy of 65.03 (Table 2). While certain proprietary models such as GPT 4.1 mini and GPT 4.1 nano exhibited marginally higher category classification accuracy, Qwen 2.5 3B provided a more balanced performance, particularly in discerning difficulty levels, and was ultimately preferred due to its open-source nature.

2.1.2 Prompt Classification Process

The classifier model processes incoming raw user prompts by assigning them to a specific category from a predefined set, which includes domains like Professional, Science and Technology, Code-related, Language Tasks, and Humanities and Social Sciences, as detailed in Table **??**.

It also assigns the prompt to one of the difficulty levels as guided below.

- **Level 1 (Simple / Factual):** Basic recall, simple instructions, minimal reasoning.
- **Level 2 (Moderate / Standard):** Some reasoning, multi-step instructions, moderately complex generation.
- **Level 3 (Complex / Advanced):** Deep reasoning, synthesis, high creativity, complex problem-solving.

2.2 Offline Benchmark-Driven Model Recommendation Generation

To inform routing decisions, an offline process of recommended model generation is carried out each time the new model is added to the candidate model.

A comprehensive dataset has been compiled, detailing the performance scores of various large language models (LLMs) (Table 7) across a range of

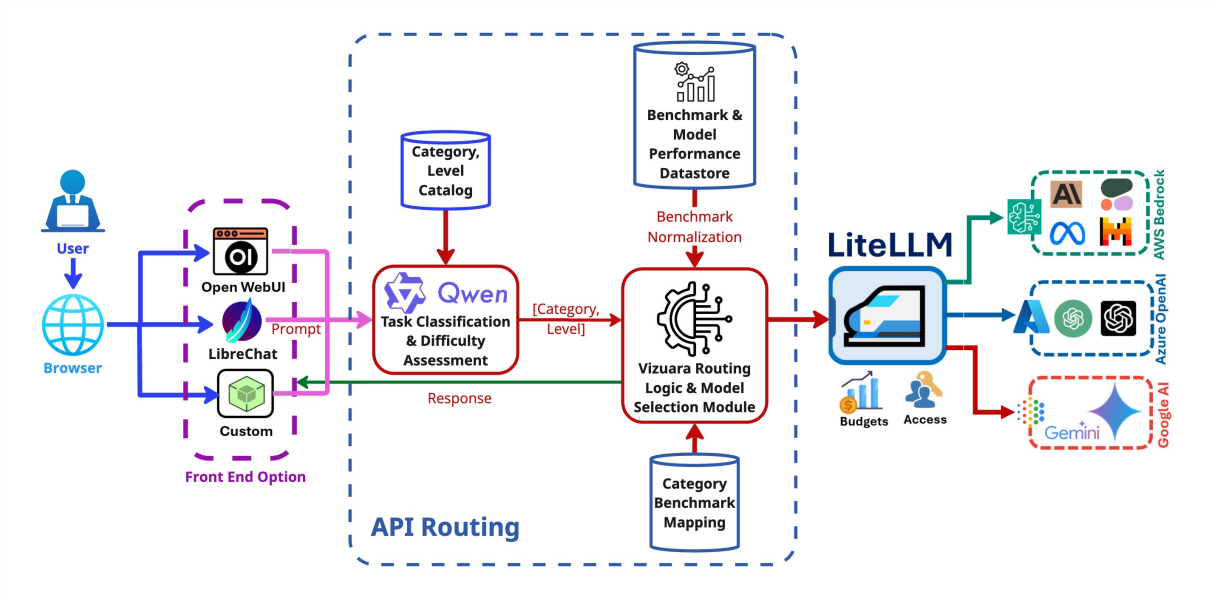

Figure 1: System Architecture of the Benchmark-Driven LLM Routing Framework.

Table 1: Model Category Classification Accuracy in Percentage Across Diverse Benchmarks. The final row reports overall accuracy averaged over 100 questions per benchmark (600 Total Questions).

Questions	Qwen 2.5 0.5B	gemma3 1b	gemma3 4b	Qwen 2.5 1.5B	Qwen 2.5 3B	Qwen3 0.6B	Qwen3 4B	GPT 4.1 nano	GPT 4.1 mini	Gemini 2.0 Flash Lite	Llama 4 Scout
GPQA	46.94	72.73	94.00	85.71	87.50	88.89	98.00	98.00	99.00	97.00	99.00
MMLU	17.17	47.96	60.00	54.64	68.89	38.54	52.58	69.57	64.00	57.00	61.86
MMLU Pro	20.41	44.79	60.42	69.79	72.09	50.00	69.15	65.26	71.72	71.00	69.07
Math 500	100.00	77.00	99.00	100.00	100.00	97.00	100.00	97.98	99.00	98.00	93.00
Humanity Last Exam	46.88	48.48	69.39	63.92	70.79	59.38	72.45	75.00	72.00	69.00	74.23
Livebench Coding	0.00	0.00	81.00	0.00	71.00	0.00	0.00	0.00	89.00	87.00	20.00
Average accuracy	38.47	48.48	77.59	62.12	**78.28**	55.90	65.37	78.79	82.47	79.83	69.54

Table 2: Model Level Classification Accuracy for selected models on GPQA and Math-500 benchmarks. The last row shows total accuracy on 200 questions. Values are percentages.

Questions	Qwen 2.5 3B	Qwen2.5 1.5B	Qwen 2.5 0.5B	Gemma3 1B	Gemma3 4B	GPT 4.1 nano	GPT 4.1 Mini	Gemini 2.0 Flash Lite
GPQA	80.21	86.73	5.10	92.93	32.00	35.00	54.00	33.00
Math_500	48.28	41.84	49.49	32.00	56.00	37.37	35.00	47.00
Total - 200	**65.03**	64.29	27.41	62.31	44.00	36.18	44.5	40.00

benchmarks, including MMLU-Pro, GPQA Diamond, Humanity's Last Exam, LiveCodeBench, SciCode, HumanEval, MATH-500, AIME 2024, LegalBench, MedQA, MGSM, and Chatbot Arena, along with associated input and output cost metrics. This table needs to be updated when a new model is added to the dataset. A predefined category-benchmark mapping links each task category to a set of relevant benchmarks (e.g., "Mathematics" maps to "MMLU-Pro", "MATH-500", "AIME 2024", etc.).

In further step raw benchmark scores for all models are converted to Z-scores. This normalization standardizes performance across different benchmarks with varying scales and score distributions. If a benchmark models with a 0 Z-score indicates this benchmark does not contribute to differentiation for that category. An Avg Z-Score is computed for each model within a category by averaging its Z-scores across all relevant benchmarks for that

category. This Avg Z-Score serves as the primary performance metric for a model within a specific task category.

For each category, three model recommendations are generated as shown in table 6. Models with valid Avg Z-Score and Total Cost for the current category are considered. Total cost is computed as the sum of input cost and output cost for each model. Cost thresholds are determined using percentiles (1/3 and 2/3 quantiles) of these models' total cost.

- **Low Cost, Good Performance:** Models falling below or at the refined lower cost threshold (approx. bottom 33rd percentile for the category) are considered. The model from this tier with the highest average Z-score is selected.
- **Moderate Cost, Good Performance:** Models falling below the upper cost thresholds (approx. 66th percentile) are selected. The model from this tier with the highest average Z-score is chosen.
- **Best Performance:** Irrespective of total cost, the model is selected just based on the highest Avg Z-score.

2.3 Dynamic Online Prompt Routing

During inference, the prompt router receives input from a classifier that predicts both the category and difficulty level of the incoming prompt. In cases where the classifier is unable to determine a valid category, the system defaults to assigning the prompt to a miscellaneous category.

The routing mechanism is further classified into three types based on cost-performance trade-offs:

- **Optimum Router:** Selects the most suitable model from the tiered cost-performance dataset, balancing both cost and performance according to the predicted category and difficulty level.
- **Low-Cost Router:** Chooses the corresponding model based on the identified category, while defaulting to the lowest difficulty level (Level 1) to minimize computational cost.
- **Best-Performance Router:** Selects the model associated with the predicted category, defaulting to the highest difficulty level (Level 3) to maximize output quality regardless of computational expense.

2.3.1 Model Invocation via LiteLLM

The selected LiteLLM model is invoked with the user prompt through the LiteLLM chat completion endpoint. The system returns the LLM's response text to the user dashboard.

3 Experimental Results

This section presents the quantitative evaluation of the proposed benchmark-driven LLM routing framework. The primary objective is to demonstrate the system's capability to achieve significant cost efficiencies while maintaining or improving performance compared to using a single, fixed LLM across various task domains and difficulty levels. The evaluation focuses on the performance and total cost ($/1M tokens, input + output) of the three router configurations: Low-Cost, Optimum, and Best-Performance, as defined in Section Dynamic Online Prompt Routing.

3.1 Evaluation Benchmarks

To rigorously assess the router's effectiveness, we selected three diverse and challenging benchmarks representative of key task categories identified during explicit task profiling:

- **GPQA Diamond:** A graduate-level, Google-proof question-answering benchmark covering complex science domains (Biology, Chemistry, Physics). It comprises 198 questions designed to be difficult for LLMs, requiring deep factual knowledge and reasoning. (Source: (Austin et al., 2023))
- **MATH-500:** A subset of 500 problems from the challenging MATH benchmark, focusing on mathematical problem-solving. This requires strong logical deduction and symbolic manipulation capabilities. (Source: (Cobbe et al., 2021))
- **LiveCodeBench (Coding):** A benchmark designed to evaluate code generation and completion capabilities while mitigating test set contamination. We utilize the coding subset, consisting of 78 code generation and 50 code completion questions. (Source: (Kallas et al., 2023))

These benchmarks were chosen for their difficulty and relevance to categories frequently encountered in practical applications, allowing for a robust evaluation of the router's ability to select appropriate models based on task type and inferred complexity.

3.2 Router Performance Evaluation

Table 3 summarizes the performance scores (accuracy or completion rate) and total costs for each router configuration across the three selected benchmarks.

Table 3: Performance and Total Cost of LLM Router Configurations on Selected Benchmarks.

Benchmark	Router Type	Score (%)	Cost ($/1M)
GPQA Diamond	Low Cost	74.87	3.09
	Optimum	82.47	8.48
	Best Performance	78.28	11.13
MATH-500	Low Cost	98.20	0.75
	Optimum	98.00	1.13
	Best Performance	96.00	5.50
Live Code Bench Coding	Low Cost	76.60	0.75
	Optimum	84.38	6.44
	Best Performance	85.20	11.25

The results demonstrate clear trade-offs offered by the different router configurations, aligning with their design principles. To provide context, we compare these router results with the performance and cost of individual LLMs as depicted in the cost-performance scatter plots (Figures 2, 3, and 4).

3.2.1 GPQA Diamond Results

On the challenging GPQA Diamond benchmark (Figure 2), the Optimum router achieved a significantly higher performance of 82.47% at a lower cost of $8.48. This performance level is very close to the peak observed performance of individual models on this benchmark (83.6% by Gemini 2.5 Pro) but at a notable cost reduction. This highlights the Optimum router's ability to find highly performant models without necessarily incurring the absolute highest costs. During the initial evaluation, it was observed that the model underperformed in the chemistry category. To enhance accuracy, an adjustment was implemented: all models assigned to the chemistry category were shifted up by one level. Specifically, the initial Gemini 2.5 Flash at Level 1 was replaced with the higher-performing o4 mini High. The Low-Cost router, operating at just $3.09 per million tokens, still delivered a respectable 74.87% accuracy. This performance is competitive with mid-range individual models (e.g., OpenAI o1-mini at $5.5 total cost for 60.3%, Llama 3.3 70b at $1.44 total cost for 50.0%) while offering substantial cost savings compared to the higher tiers and top individual models.

3.2.2 MATH-500 Results

The MATH-500 benchmark (Figure 3) evaluation revealed the router's exceptional efficiency in numerical and logical tasks. The Low-Cost router achieved an impressive 98.20% accuracy for a mere $0.75 total cost. This performance surpasses many individual models across the entire cost spectrum and is on par with or exceeds the accuracy of the most expensive models (e.g., Gemini 2.5 Pro at $11.25 for 98.0%). The Optimum router maintained this high accuracy at 98.00%, with a slightly higher cost of $1.13, still remarkably low compared to most high-performance individual models. The Best Performance router scored 96.00% at a cost of $5.50. While its performance was slightly lower than the other two tiers on this specific benchmark run. The performance on MATH-500 underscores the router's ability to identify highly efficient models for specific tasks, leading to dramatic cost reductions without sacrificing accuracy, particularly benefiting from the lower inference costs of certain models when selected optimally.

3.2.3 LiveCodeBench Coding Results

The LiveCodeBench Coding benchmark (Figure 4) demonstrates the router's strong capabilities in code-related tasks. The Best Performance router achieved the highest score among all tested models and router configurations on this specific benchmark set, reaching 85.20% completion rate at a cost of $11.25. This outperforms the highest individual model score observed (e.g., o4-mini (high) at 80.4% for $4.40, Gemini 2.5 Pro at 69.5% for $11.25). The Optimum router was close in performance at 84.38%, but at a significantly reduced cost of $6.44, offering an excellent balance for demanding coding tasks. The Low-Cost router again provided substantial cost savings ($0.75) while achieving a very strong 76.60% performance, competitive with many individual models costing significantly more (e.g., OpenAI o1 at $75 for 67.9%, DeepSeek-R1 at $6.75 for 61.7%).

3.3 Discussion

The experimental results clearly validate the effectiveness of our benchmark-driven LLM router, DynaRoute. Its three configurations successfully

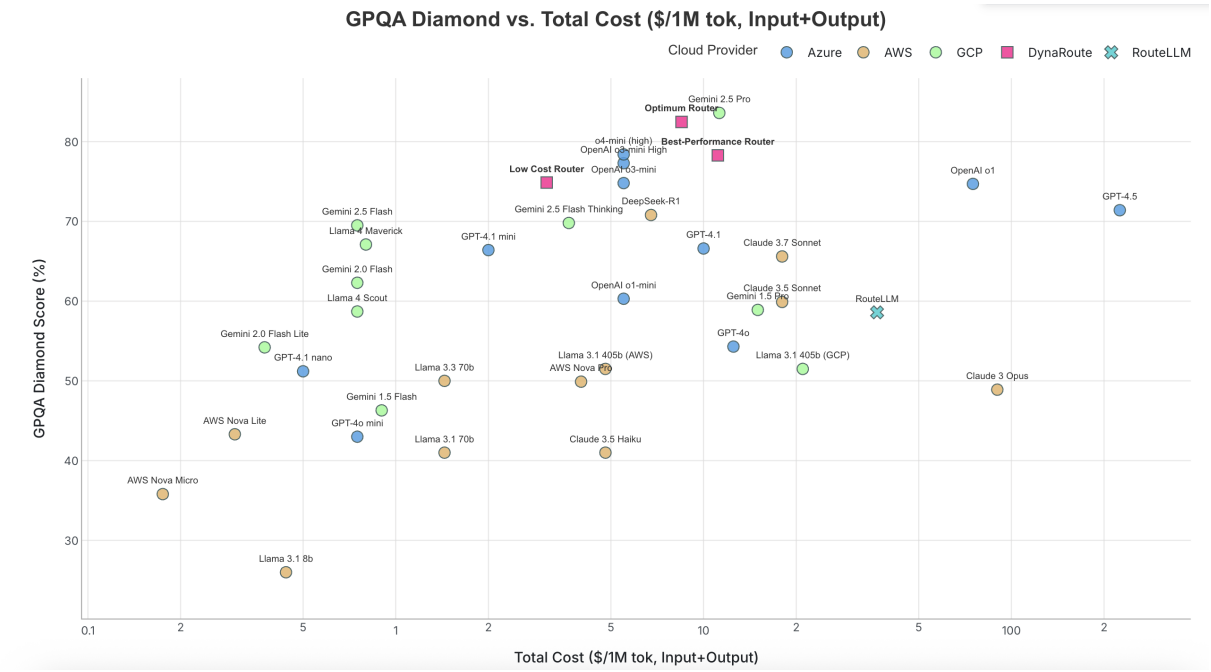

Figure 2: GPQA Diamond Score vs. Total Cost ($/1M tok, Input+Output). DynaRoute's Optimum router achieves a significantly higher performance of 82.47% at a cost of $8.48/1M tokens. In comparison, RouteLLM (Ong et al., 2024) between GPT-4.1 and OpenAI o1) scores 58.59% at a substantially higher cost of $36.62/1M tokens. Other individual models are also shown.

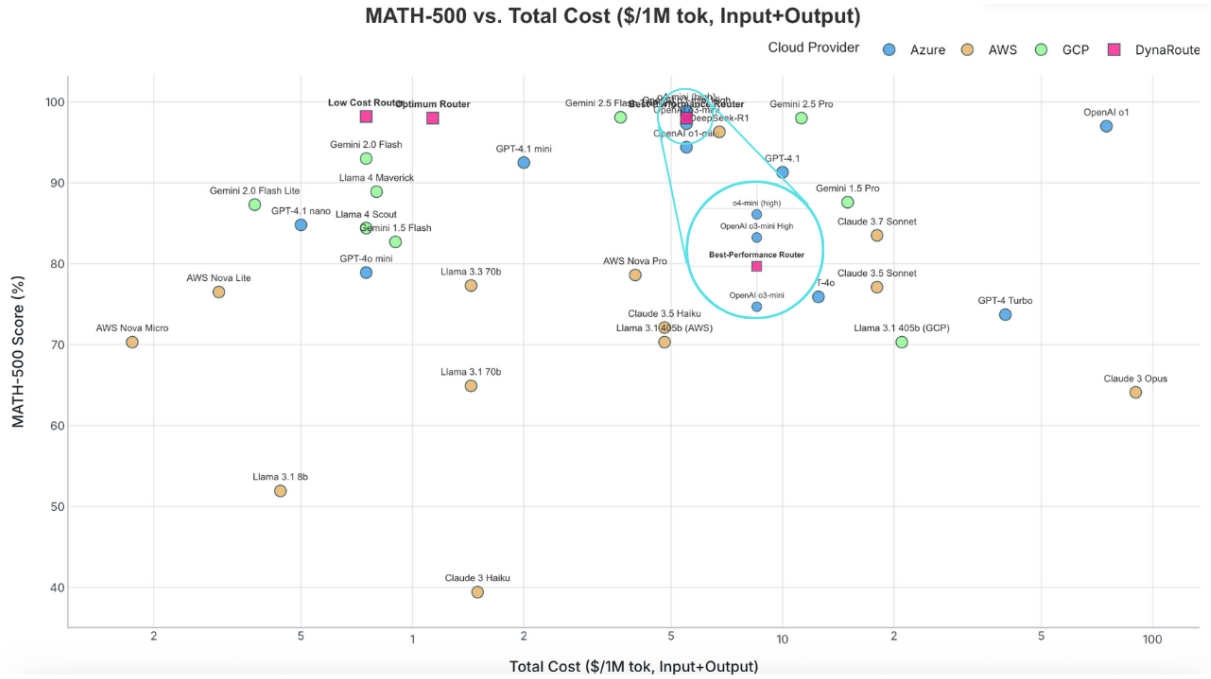

Figure 3: MATH-500 Score vs. Total Cost ($/1M tok, Input+Output). DynaRoute's Optimum router achieves a high accuracy of 98.00% at a remarkably low cost of $1.13/1M tokens. Even its Low-Cost router configuration attains 98.20% accuracy for only $0.75/1M tokens, outperforming or matching many significantly more expensive individual models. Router configurations are shown alongside individual models.

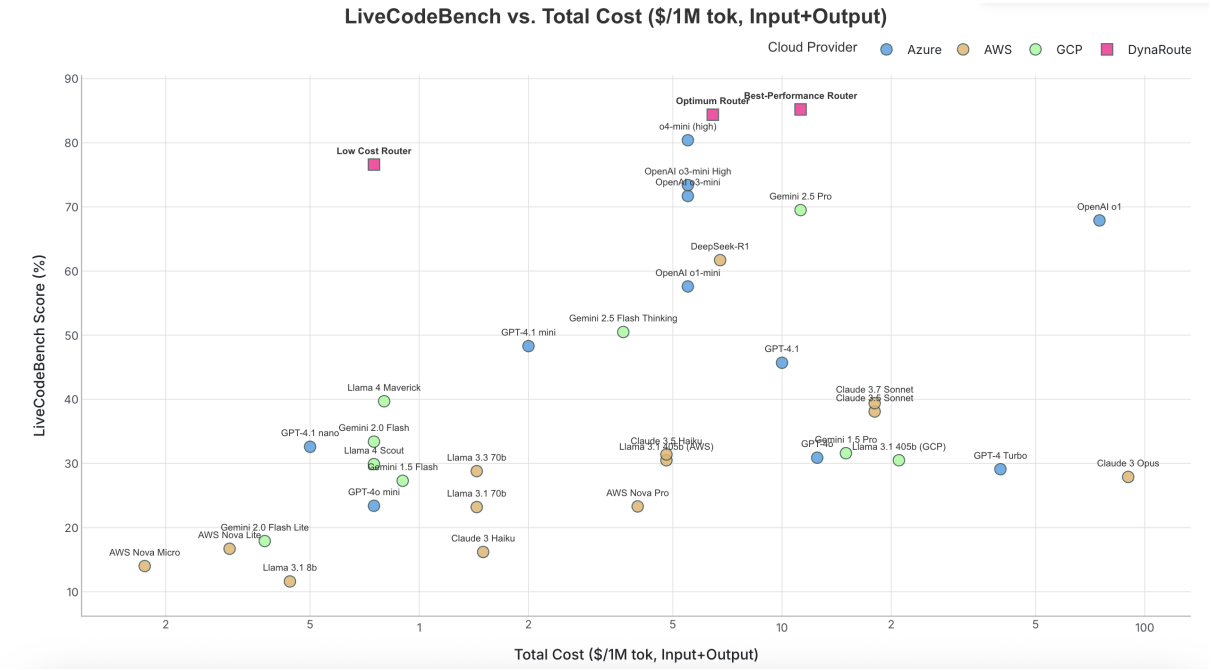

Figure 4: LiveCodeBench Coding Score vs. Total Cost ($/1M tok, Input+Output). DynaRoute's Best-Performance router achieves a leading score of 85.20% at $11.25/1M tokens, while its Optimum router delivers a strong 84.38% at a reduced cost of $6.44/1M tokens. Router configurations are shown alongside individual models.

embody distinct cost-performance trade-offs, offering flexibility based on application requirements.

To contextualize DynaRoute's performance, RouteLLM (Ong et al., 2024) was selected for comparison on the GPQA benchmark. Configured with OpenAI's o1 (strong) and GPT-4.1 nano (weak) as per its design, RouteLLM achieved 58.59% accuracy at $36.62/1M tokens. In contrast, DynaRoute's Optimum router delivered significantly higher 82.47% accuracy at only $8.48/1M tokens, demonstrating superior cost-efficiency.

The **Optimum router** epitomizes the system's value by achieving performance comparable to or exceeding top individual models on challenging benchmarks, yet at significantly lower total costs, optimizing the cost-performance frontier.

The **Low-Cost router** demonstrated dramatic cost reductions while maintaining surprisingly high performance across diverse tasks like MATH-500 and LiveCodeBench. This tier is ideal for cost-constrained scenarios or high-volume simple tasks, as not all prompts demand the most expensive models.

The **Best-Performance router** acts as an upper bound, showcasing the system's ability to identify and utilize the most capable models when maximum accuracy is paramount. In practical applications with mixed prompt complexities, routing simpler requests to low-cost models will yield substantial overall cost savings.

4 Conclusion

In this work, we introduced a benchmark-driven LLM routing framework, DynaRoute, that intelligently matches tasks to models based on domain, difficulty, and cost-performance trade-offs. Our tiered selection strategy, Low-Cost, Optimum and Best-Performance, demonstrated strong results across diverse benchmarks, offering a practical and interpretable alternative to black-box systems and achieving superior cost-performance compared to other routing approaches as RouteLLM (Ong et al., 2024) , on specific tasks. While reliant on classifier accuracy and up-to-date benchmarks, our approach lays a solid foundation for cost-aware, performance-optimized LLM deployment. Future directions include incorporating real-time feedback and finer-grained profiling.

5 Acknowledgements

In the preparation of this manuscript, we utilized AI-powered writing assistance tools. The scope of this assistance was primarily focused on improving the linguistic quality of the text, including sentence restructuring, grammar correction, and sentence completion to enhance clarity and flow.

6 Limitations

The efficacy of the routing system is heavily dependent on the initial task profiling accuracy of the classifier LLM (Qwen 2.5 3B). While selected for its balance of performance and efficiency, Tables 1 and 2 show that classification is not perfect (78.28% category accuracy, 65.03% level accuracy). Misclassifying the category or difficulty level of a prompt can lead to suboptimal model selection, potentially routing a complex query to a low-cost, less capable model or a simple query to an expensive, high-performance one unnecessarily.

References

Pranjal Aggarwal, Aman Madaan, Ankit Anand, Srividya Pranavi Potharaju, Swaroop Mishra, Pei Zhou, Aditya Gupta, Dheeraj Rajagopal, Karthik Kappaganthu, Yiming Yang, et al. 2024. AutoMix: Automatically Mixing Language Models. *arXiv preprint arXiv:2310.12963*.

Jacob Austin, Yi Lee, Kevin Wu, Oleg Popkov, Taro Koreeda, Justin Shleifer, Jonathan Lightman, Jyothi Kudugunta, Ethan Mu, David Xu, Maria Lan, Lisa Tang, Kevin Dohan, Long Ouyang Chan, ChatGPT team, Will Chen, Jeffrey Huang, Josh Achiam, Greg Mishkin, Jack Clark, Alec Radford, Ilya Sutskever, Dario Amodei, Tom Brown, Paul Christiano, Jan Leike, Ryan Lowe, Shane McAuliffe, and Ethan Mollick. 2023. GPQA: A Graduate-Level Algorithmic Question-Answering Benchmark. *arXiv preprint arXiv:2311.12022*.

Lingjiao Chen, Matei Zaharia, and James Zou. 2023. FrugalGPT: How to Use Large Language Models While Reducing Cost and Improving Performance. *arXiv preprint arXiv:2305.05176*.

Karl Cobbe, Prafulla Kosaraju, Mohammad Hilton, William Peng, Hengyuan Chen, and John Schulman. 2021. Training Verifiers to Solve Math Word Problems. *arXiv preprint arXiv:2110.14168*.

DeepSeek-AI. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning.

Dujian Ding, Ankur Mallick, Chi Wang, Robert Sim, Subhabrata Mukherjee, Victor Rühle, Laks VS Lakshmanan, and Ahmed Hassan Awadallah. HybridLLM: Cost-Efficient and Quality-Aware Query Routing for Large Language Models. *arXiv preprint arXiv:2404.14618*.

Tao Feng, Yanzhen Shen, and Jiaxuan You. GraphRouter: A Graph-based Router for LLM Selections. *arXiv preprint arXiv:2410.03834*.

Qitian Jason Hu, Jacob Bieker, Xiuyu Li, Nan Jiang, Benjamin Keigwin, Gaurav Ranganath, Kurt Keutzer, and Shriyash Kaustubh Upadhyay. 2024. RouterBench: A Benchmark for Multi-LLM Routing System. In *Proceedings of the International Conference on Machine Learning (ICML) Agentic AI Workshop*.

Zhongzhan Huang, Guoming Ling, Vincent S. Liang, Yupei Lin, Yandong Chen, Shanshan Zhong, Hefeng Wu, and Liang Lin. 2024. RouterEval: A Comprehensive Benchmark for Routing LLMs to Explore Model-level Scaling Up in LLMs. *arXiv preprint arXiv:2503.10657*.

Chandra Irugalbandara, Ashish Mahendra, Roland Daynauth, Tharuka Kasthuri Arachchige, Jayanaka Dantanarayana, Krisztian Flautner, Lingjia Tang, Yiping Kang, and Jason Mars. Scaling Down to Scale Up: A Cost-Benefit Analysis of Replacing OpenAI's LLM with Open Source SLMs in Production. *arXiv preprint arXiv:2312.14972*.

Swayambhoo Jain, Ravi Raju, Bo Li, Zoltan Csaki, Jonathan Li, Kaizhao Liang, Guoyao Feng, Urmish Thakkar, Anand Sampat, Raghu Prabhakar, and Sumati Jairath. 2024. Composition of experts: A modular compound ai system leveraging large language models.

Dongfu Jiang, Xiang Ren, and Bill Yuchen Lin. 2023. LLM-Blender: Ensembling Large Language Models with Pairwise Ranking and Generative Fusion. *arXiv preprint arXiv:2306.02561*.

Aari Kallas, Tarmo Kallas, Simo Lill, Siim Kännaste, Andreas Saarepera, Karl-Hendrik Piik, Kevin Sild, Kaspar Köiv, Sven Sutt, Kaur Parts, Robert Padari, Robin Kärner, and Toomas Aasamets. 2023. LiveBench: Towards Live and Contamination-Free Benchmarking of Code Large Language Models. *arXiv preprint arXiv:2310.05311*.

Hongwei Liu, Zilong Zheng, Yuxuan Qiao, Haodong Duan, Zhiwei Fei, Fengzhe Zhou, Wenwei Zhang, Songyang Zhang, Dahua Lin, and Kai Chen. 2024. Mathbench: Evaluating the theory and application proficiency of llms with a hierarchical mathematics benchmark.

Noah Martin, Abdullah Bin Faisal, Hiba Eltigani, Rukhshan Haroon, Swaminathan Lamelas, and Fahad Dogar. 2024. LLMProxy: Reducing Cost to Access Large Language Models. *arXiv preprint arXiv:2410.11857*.

Isaac Ong, Amjad Almahairi, Vincent Wu, Wei-Lin Chiang, Tianhao Wu, Joseph E. Gonzalez, M Waleed Kadous, and Ion Stoica. 2024. RouteLLM: Learning to Route LLMs with Preference Data. *arXiv preprint arXiv:2406.18665*.

OpenAI. 2024a. Gpt-4o system card.

OpenAI. 2024b. Openai o1 system card.

Shivanshu Shekhar, Tanishq Dubey, Koyel Mukherjee, Apoorv Saxena, Atharv Tyagi, and Nishanth Kotla. 2024. Towards Optimizing the Costs of LLM Usage. *arXiv preprint arXiv:2402.01742*.

Dimitrios Sikeridis, Dennis Ramdass, and Pranay Pareek. 2024. PickLLM: Context-Aware RL-Assisted Large Language Model Routing. In *arXiv preprint arXiv:2412.12170*.

Florian Tambon, Amin Nikanjam, Cyrine Zid, Foutse Khomh, and Giuliano Antoniol. TaskEval: Assessing Difficulty of Code Generation Tasks for Large Language Models. *arXiv preprint arXiv:2407.21227*.

Yi Tay, Mostafa Dehghani, Vinh Q. Tran, Xavier Garcia, Jason Wei, Xuezhi Wang, Hyung Won Chung, Siamak Shakeri, Dara Bahri, Tal Schuster, et al. 2022. UL2: Unifying Language Learning Paradigms. *arXiv preprint arXiv:2205.05131*.

Clovis Varangot-Reille, Christophe Bouvard, Antoine Gourru, Mathieu Ciancone, Marion Schaeffer, and François Jacquenet. 2025. Doing More with Less – Implementing Routing Strategies in Large Language Model-Based Systems: An Extended Survey. *arXiv preprint arXiv:2502.00409*.

Xinyuan Wang, Yanchi Liu, Wei Cheng, Xujiang Zhao, Zhengzhang Chen, Wenchao Yu, Yanjie Fu, and Haifeng Chen. 2024. MixLLM: Dynamic Routing in Mixed Large Language Models. *arXiv preprint arXiv:2502.18482*.

A Appendix

This appendix provides supplementary details regarding the task-to-benchmark mapping used in the offline recommendation generation process and the resulting model recommendations per category.

Table 4: Task Categories for Prompt Classification

Main Category	**Specific Domains / Tasks**
Professional Domains	Law, Business, Economics
Science and Technology	Biology, Chemistry, Engineering, Physics, Computer Science, Mathematics
Code-related	Syntax Check, Code Generation, Code Explanation, Debugging Assistance, Algorithm, System Design
Language Tasks	Language Translation, Summarization, Creative Writing, Text Formatting and Editing, General Knowledge
Humanities and Social Sciences	History, Geography, Philosophy, Sociology, Astrology
Other	Health, Facts, Logic, Miscellaneous

Table 5: Mapping of Task Categories to Relevant Benchmarks for Offline Performance Evaluation.

Category	**List of Associated Benchmark**
Law	MMLU-Pro, Humanity's Last Exam, Multilingual Index
Business	MMLU-Pro, Humanity's Last Exam
Economics	MMLU-Pro, Humanity's Last Exam
Biology	MMLU-Pro, GPQA Diamond, Humanity's Last Exam, SciCode
Chemistry	MMLU-Pro, GPQA Diamond, Humanity's Last Exam, SciCode
Engineering	MMLU-Pro, Humanity's Last Exam
Physics	MMLU-Pro, GPQA Diamond, Humanity's Last Exam, SciCode
Computer Science	MMLU-Pro, Humanity's Last Exam, LiveCodeBench, SciCode, HumanEval
Mathematics	MMLU-Pro, Humanity's Last Exam, SciCode, HumanEval, MATH-500, AIME 2024
Syntax Check	HumanEval, LiveCodeBench, SciCode
Code Generation	LiveCodeBench, SciCode, HumanEval, Multilingual Index
Code Explanation	(None listed)
Debugging Assistance	LiveCodeBench
Algorithm	LiveCodeBench, SciCode, HumanEval
System Design	LiveCodeBench, SciCode
Language Translation	Multilingual Index
Summarization	Multilingual Index
Creative Writing	Multilingual Index
Text Formatting	(None listed)
General Knowledge	MMLU-Pro, GPQA Diamond, Humanity's Last Exam, Multilingual Index
History	MMLU-Pro, Humanity's Last Exam
Geography	Humanity's Last Exam
Philosophy	MMLU-Pro, Humanity's Last Exam
Sociology	Humanity's Last Exam
Astrology	(None listed)
Health	MMLU-Pro, Humanity's Last Exam
Facts	MMLU-Pro, GPQA Diamond, Humanity's Last Exam
Logic	MMLU-Pro, GPQA Diamond, Humanity's Last Exam, LiveCodeBench, HumanEval, MATH-500, AIME 2024
Miscellaneous	MMLU-Pro

Table 6: Tiered Model Recommendations by Task Category (Model, Avg Z-Score, Total Cost $/1M tok).

Category	Low Cost	Moderate Cost	Best Performance
Law	Gemini 2.5 Flash (Score: 80.20, Cost: 0.75)	Claude 3 Haiku (Score: 68.30, Cost: 1.50)	Gemini 2.5 Flash (Score: 80.20, Cost: 0.75)
Business	Gemini 2.5 Flash (Score: 80.20, Cost: 0.75)	o4-mini (high) (Score: 50.35, Cost: 5.50)	Gemini 2.5 Flash (Score: 80.20, Cost: 0.75)
Economics	Gemini 2.5 Flash (Score: 80.20, Cost: 0.75)	o4-mini (high) (Score: 50.35, Cost: 5.50)	Gemini 2.5 Flash (Score: 80.20, Cost: 0.75)
Biology	Gemini 2.5 Flash (Score: 57.07, Cost: 0.75)	o4-mini (high) (Score: 56.40, Cost: 5.50)	GPT-4.5 (Score: 71.40, Cost: 225.0)
Chemistry	Gemini 2.5 Flash (Score: 57.07, Cost: 0.75)	o4-mini (high) (Score: 56.40, Cost: 5.50)	GPT-4.5 (Score: 71.40, Cost: 225.0)
Engineering	Gemini 2.5 Flash (Score: 80.20, Cost: 0.75)	o4-mini (high) (Score: 50.35, Cost: 5.50)	Gemini 2.5 Flash (Score: 80.20, Cost: 0.75)
Physics	Gemini 2.5 Flash (Score: 57.07, Cost: 0.75)	o4-mini (high) (Score: 56.40, Cost: 5.50)	GPT-4.5 (Score: 71.40, Cost: 225.0)
Computer Science	Gemini 2.5 Flash (Score: 50.85, Cost: 0.75)	o4-mini (high) (Score: 65.32, Cost: 5.50)	GPT-4 (Score: 88.40, Cost: 90.0)
Mathematics	Llama 4 Maverick (Score: 55.77, Cost: 0.80)	o4-mini (high) (Score: 73.18, Cost: 5.50)	GPT-4 (Score: 88.40, Cost: 90.0)
Syntax Check	Gemini 1.5 Flash (Score: 55.55, Cost: 0.90)	o4-mini (high) (Score: 75.30, Cost: 5.50)	GPT-4 (Score: 88.40, Cost: 90.0)
Code Generation	Gemini 1.5 Flash (Score: 63.93, Cost: 0.90)	o4-mini (high) (Score: 75.30, Cost: 5.50)	GPT-4 (Score: 88.40, Cost: 90.0)
Debugging Assistance	Llama 4 Maverick (Score: 39.70, Cost: 0.80)	o4-mini (high) (Score: 80.40, Cost: 5.50)	o4-mini (high) (Score: 80.40, Cost: 5.5)
Algorithm	Gemini 1.5 Flash (Score: 55.55, Cost: 0.90)	o4-mini (high) (Score: 75.30, Cost: 5.50)	GPT-4 (Score: 88.40, Cost: 90.0)
System Design	Llama 4 Maverick (Score: 36.40, Cost: 0.80)	o4-mini (high) (Score: 63.45, Cost: 5.50)	o4-mini (high) (Score: 63.45, Cost: 5.5)
Language Translation	Gemini 1.5 Flash (Score: 80.70, Cost: 0.90)	Llama 3.3 70b (Score: 83.90, Cost: 1.44)	Claude 3.5 Sonnet (Score: 88.40, Cost: 18.0)
Summarization	Gemini 1.5 Flash (Score: 80.70, Cost: 0.90)	Llama 3.3 70b (Score: 83.90, Cost: 1.44)	Claude 3.5 Sonnet (Score: 88.40, Cost: 18.0)
Creative Writing	Gemini 1.5 Flash (Score: 80.70, Cost: 0.90)	Llama 3.3 70b (Score: 83.90, Cost: 1.44)	Claude 3.5 Sonnet (Score: 88.40, Cost: 18.0)
General Knowledge	Gemini 2.5 Flash (Score: 74.85, Cost: 0.75)	Claude 3 Haiku (Score: 68.30, Cost: 1.50)	Gemini 2.5 Flash (Score: 74.85, Cost: 0.75)
History	Gemini 2.5 Flash (Score: 80.20, Cost: 0.75)	o4-mini (high) (Score: 50.35, Cost: 5.50)	Gemini 2.5 Flash (Score: 80.20, Cost: 0.75)
Geography	Gemini 2.0 Flash (Score: 5.30, Cost: 0.75)	o4-mini (high) (Score: 17.50, Cost: 5.50)	o4-mini (high) (Score: 17.50, Cost: 5.5)
Philosophy	Gemini 2.5 Flash (Score: 80.20, Cost: 0.75)	o4-mini (high) (Score: 50.35, Cost: 5.50)	Gemini 2.5 Flash (Score: 80.20, Cost: 0.75)
Sociology	Gemini 2.0 Flash (Score: 5.30, Cost: 0.75)	o4-mini (high) (Score: 17.50, Cost: 5.50)	o4-mini (high) (Score: 17.50, Cost: 5.5)
Health	Gemini 2.5 Flash (Score: 80.20, Cost: 0.75)	o4-mini (high) (Score: 50.35, Cost: 5.50)	Gemini 2.5 Flash (Score: 80.20, Cost: 0.75)
Facts	Gemini 2.5 Flash (Score: 74.85, Cost: 0.75)	o4-mini (high) (Score: 59.70, Cost: 5.50)	Gemini 2.5 Flash (Score: 74.85, Cost: 0.75)
Logic	Gemini 2.5 Flash (Score: 64.33, Cost: 0.75)	o4-mini (high) (Score: 78.77, Cost: 5.50)	GPT-4 (Score: 88.40, Cost: 90.0)
Miscellaneous	Llama 4 Maverick (Score: 80.90, Cost: 0.80)	o4-mini (high) (Score: 83.20, Cost: 5.50)	Gemini 2.5 Pro (Score: 85.80, Cost: 11.25)

Table 7: Full Model Benchmark Performance and Cost Data.

Model	MMLU-Pro	GPQA	Humanity's Last Exam	Live Code	Sci Code	Human Eval	MATH-500	AIME 2024	Multi Ling.	Input Cost	Output Cost
OpenAI o1	84.10	74.70	7.70	67.90	35.80	97.40	97.00	72.30	87.60	15.00	60.00
Claude 3.5 Sonnet	77.20	59.90	3.90	38.10	35.10	93.00	77.10	15.70	88.40	3.00	15.00
GPT-4o	74.80	54.30	3.30	30.90	33.30	93.00	75.90	15.00	83.80	2.50	10.00
Llama 3.1 405b	73.20	51.50	4.20	30.50	29.90	85.40	70.30	21.30	76.50	2.40	2.40
Llama 3.1 405b	73.20	51.50	4.20	30.50	29.90	85.40	70.30	21.30	76.50	5.00	16.00
OpenAI o1-mini	74.20	60.30	4.90	57.60	32.30	97.20	94.40	60.30	83.30	1.10	4.40
GPT-4 Turbo	69.40	—	3.30	29.10	29.50	91.80	73.70	9.70	—	10.00	30.00
Claude 3 Opus	69.60	48.90	3.10	27.90	23.30	84.80	64.10	3.30	—	15.00	75.00
DeepSeek V3	75.70	55.70	3.60	35.90	35.40	90.60	88.70	25.30	86.40	—	—
GPT-4	—	—	—	—	—	88.40	—	—	—	30.00	60.00
Llama 3.1 70b	67.60	41.00	4.60	23.20	26.70	81.20	64.90	17.30	—	0.72	0.72
Llama 3.3 70b	71.30	50.00	4.00	28.80	26.00	86.00	77.30	30.00	83.90	0.72	0.72
Gemini 1.5 Pro	75.00	58.90	4.90	31.60	29.50	89.80	87.60	23.00	85.00	5.00	10.00
Claude 3.5 Haiku	63.40	41.00	3.50	31.40	26.00	85.90	72.10	3.30	78.50	0.80	4.00
Gemini 1.5 Flash	67.80	46.30	3.50	27.30	—	83.80	82.70	18.00	80.70	0.30	0.60
Claude 3 Haiku	—	—	—	16.20	17.70	70.60	39.40	—	68.30	0.25	1.25
Llama 3.1 8b	47.60	26.00	5.10	11.60	13.20	66.50	51.90	7.70	61.00	0.22	0.22
GPT-3.5 Turbo	—	—	—	—	—	—	—	—	—	0.50	1.50
Gemini 2.0 Flash	77.90	62.30	5.30	33.40	31.20	90.40	93.00	33.00	—	0.15	0.60
AWS Nova Micro	53.10	35.80	3.40	14.00	9.40	79.90	70.30	8.00	71.10	0.035	0.14
AWS Nova Lite	59.00	43.30	4.60	16.70	13.80	82.80	76.50	10.70	76.10	0.06	0.24
AWS Nova Pro	69.10	49.90	4.70	23.30	20.80	84.10	78.60	10.70	83.40	0.80	3.20
GPT-4o mini	64.80	43.00	4.00	23.40	22.90	87.60	78.90	11.70	80.50	0.15	0.60
OpenAI o3-mini	79.10	74.80	8.70	71.70	39.80	97.20	97.30	77.00	—	1.10	4.40
OpenAI o3-mini High	80.20	77.30	12.30	73.40	39.90	—	98.50	86.00	—	1.10	4.40
DeepSeek-R1	84.40	70.80	9.30	61.70	35.70	97.70	96.30	68.30	—	1.35	5.40
GPT-4.5	—	71.40	—	—	—	—	—	36.70	—	75.00	150.00
Claude 3.7 Sonnet	80.30	65.60	4.80	39.40	37.50	92.20	83.50	24.30	—	3.00	15.00
Gemini 2.0 Flash Lite	72.30	54.20	4.40	17.90	27.70	89.60	87.30	30.30	—	0.075	0.30
GPT-4.1	80.60	66.60	4.60	45.70	38.10	95.60	91.30	43.70	—	2.00	8.00
GPT-4.1 mini	78.10	66.40	4.60	48.30	40.40	95.00	92.50	43.00	—	0.40	1.60
GPT-4.1 nano	65.70	51.20	3.90	32.60	25.90	87.70	84.80	23.70	—	0.10	0.40
Llama 4 Maverick	80.90	67.10	4.80	39.70	33.10	87.90	88.90	39.00	—	0.20	0.60
Llama 4 Scout	75.20	58.70	4.30	29.90	17.00	82.60	84.40	28.30	—	0.15	0.60
o4-mini (high)	83.20	78.40	17.50	80.40	46.50	99.00	98.90	94.00	—	1.10	4.40
Gemini 2.5 Flash	80.20	69.50	—	—	21.50	—	—	43.30	—	0.15	0.60
Gemini 2.5 Flash Thinking	80.00	69.80	11.60	50.50	35.90	—	98.10	84.30	—	0.15	3.50
Gemini 2.5 Pro	85.80	83.60	17.10	69.50	39.50	98.50	98.00	87.00	—	1.25	10.00