Optimizing Reasoning Efficiency through Prompt Difficulty Prediction

Bo Zhao[*] UC San Diego

Berkcan Kapusuzoglu Capital One

Kartik Balasubramaniam Capital One

Sambit Sahu Capital One

Supriyo Chakraborty Capital One

Genta Indra Winata Capital One

Abstract

Reasoning language models perform well on complex tasks but are costly to deploy due to their size and long reasoning traces. We propose a routing approach that assigns each problem to the smallest model likely to solve it, reducing compute without sacrificing accuracy. Using intermediate representations from s1.1-32B, we train lightweight predictors of problem difficulty or model correctness to guide routing across a pool of reasoning models. On diverse math benchmarks, routing improves efficiency over random assignment and matches s1.1-32B's performance while using significantly less compute. Our results demonstrate that difficulty-aware routing is effective for cost-efficient deployment of reasoning models.

1 Introduction

Recent advances in large language models (LLMs) have significantly improved reasoning across math, science, and general problem-solving tasks [Li et al., 2025]. However, these gains come with high computational costs, especially when large models are used uniformly across tasks of varying difficulty. Many problems can be solved by smaller models at a fraction of the cost, suggesting that adaptive model selection could greatly improve efficiency [OpenAI, 2025].

This paper investigates whether intermediate representations from LLMs can be used to predict problem difficulty and guide model selection. We train lightweight classifiers to predict either the difficulty of a problem or the likelihood that a model will answer it correctly. These predictors, trained on outputs from the s1.1-32B model, enable routing strategies that assign each problem to the smallest model expected to succeed. Prior work relied on problem-specific, hand-designed metrics (e.g., number of obstacles in a maze) to assess difficulty [Saha et al., 2025]. In contrast, we propose a learned, general-purpose classifier for labeling problem difficulty.

We evaluate this approach across multiple reasoning benchmarks, comparing against baselines that use a single model or assign models at random. Our results show that mid-layer LLM representations are highly informative for difficulty and correctness prediction, and that routing based on these signals can dramatically reduce inference cost while maintaining high accuracy of the large models.

2 Related Work

Improving efficiency using prompt attributes Prediction of expected performance [Shnitzer et al., 2023, Lu et al., 2023] or uncertainty level [Chuang et al., 2025a,b], often combined with the goal to minimize cost [Mohammadshahi et al., 2024, Hu et al., 2024, Ding et al., 2024, Šakota et al., 2024],

[*]Work done during an internship at Capital One. Contact: bozhao@ucsd.edu.

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

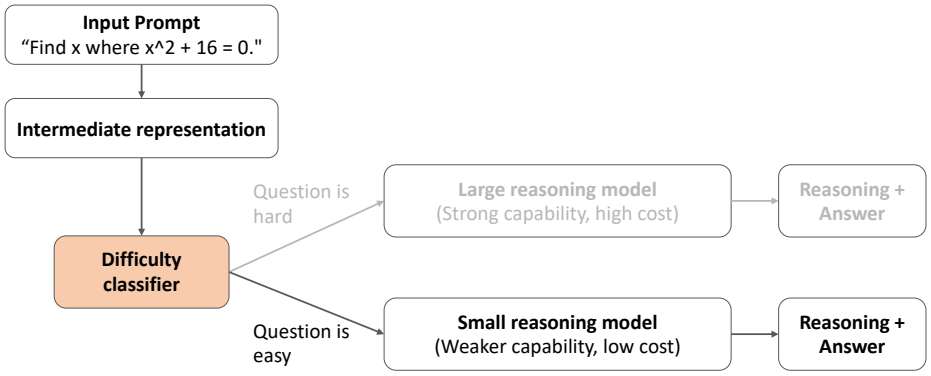

Figure 1: A classifier predicts problem difficulty from intermediate representations, and route each problem to the smallest reasoning model likely to solve it. This reduces inference cost while maintaining accuracy.

has been used to route query to LLMs with different expertise or size. As reasoning models emerge recently, people use similar query attributes to route to models with different reasoning capabilities to mitigate overthinking [Pu et al., 2025, Saha et al., 2025]. We contributing to this line through developing a prediction model for more fine-grained difficulty attributes of queries. We then use the learned query attributes to improve the same model, instead of to route among different models.

Controllable fast and slow thinking Balancing between fast and slow thinking can be achieved by combining models with different capabilities or exploring shortcuts in the reasoning trace [Sui et al., 2025]. System-1.x Planner [Saha et al., 2025] decomposes problems and routes sub-tasks to fast or slow solvers based on a task-based hardness function. The Fast-Slow-Thinking framework [Sun et al., 2025] first simplifies a task by removing constraints for a quick solution, then refines it by reintroducing them. Dualformer [Su et al., 2024] trains a single model on randomized reasoning traces with different parts dropped, enabling it to learn both full reasoning and cognitive shortcuts. System-1.5 Reasoning [Wang et al., 2025] allocates computational effort by learning shortcuts along model depth in latent-space reasoning.

3 Methods and Experiment Settings

We train a model to predict properties of reasoning problems and route each to an appropriately sized reasoning model (Figure 1). We assume access to a pool of reasoning models with varying capacity and inference cost. We first use intermediate outputs from s1.1-32B to train a predictor of either problem difficulty or model correctness. This predictor then guides routing, assigning each question to the smallest model likely to solve it. We evaluate the router by measuring accuracy and inference time against a baseline of random assignment.

3.1 Difficulty Level Prediction

We train a model to predict question difficulty using the MATH dataset [Hendrycks et al., 2021], which includes 7,500 competition problems labeled from 1 (easiest) to 5 (hardest). We split the dataset into 6000 for training and 1500 for validation. For the difficulty level predictor, we train a 3-layer MLP with s1.1-32B's layer outputs as input (dimension 5120), hidden dimensions 256 and 64, and cross entropy loss. We train 20 epochs using batch size 32 and learning rate 10^{-5}.

3.2 Model Accuracy Prediction

To predict whether a given LLM can solve a problem, we construct the MathCombined dataset, which consists of 3136 reasoning tasks with ground truth solution but without difficulty levels. This dataset includes problems from AIME24 [Mathematical Association of America, 2024], AMC23 [Mathematical Association of America, 2023], GSM8k [Cobbe et al., 2021], Minerva [Lewkowycz et al., 2022], OlympiadBench [He et al., 2024], and TheoremQA [Chen et al., 2023]. We split the

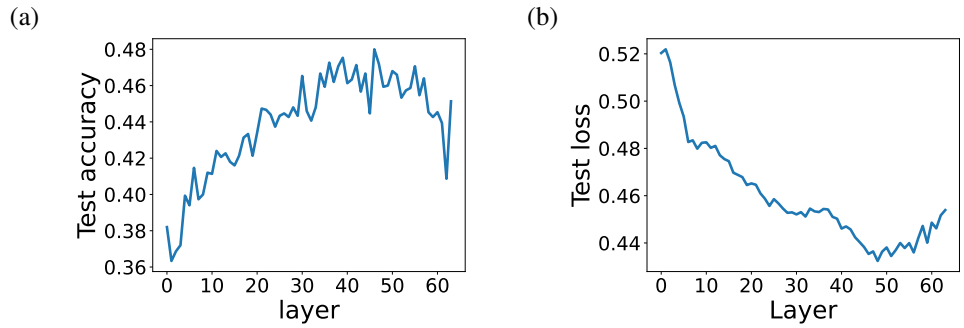

Figure 2: Prediction performance using outputs from different layers of s1.1-32B on (a) question difficulty level and (b) whether various language models can answer the given question correctly. Middle layers provide the most informative representations.

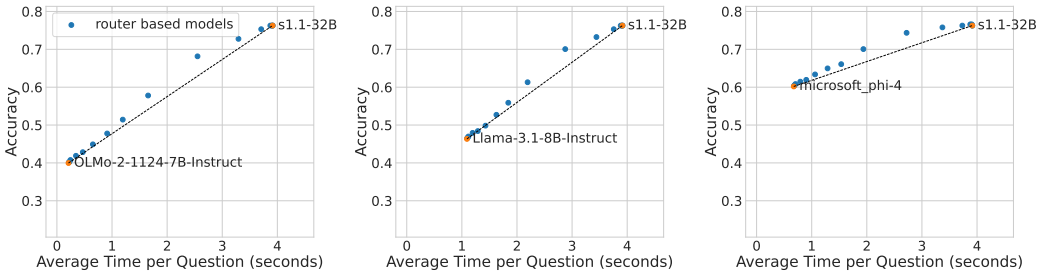

Figure 3: Performance of difficulty-based routing using s1.1-32B layer outputs. A problem is routed to a larger model if the predicted difficulty exceeds a threshold, and to a smaller model otherwise. Blue dots indicate router-based systems with thresholds between 2.1 and 2.9; orange dots show baseline models. Routers consistently outperform random assignment.

data into 1882 for training, 626 for validation, and 628 for router evaluation (Section 3.4). For the model accuracy predictor, we train a 4-layer MLP with s1.1-32B's layer outputs as inputs (dimension 5120), hidden dimensions 8192, 2048, and 128, and binary cross entropy loss. We train 20 epochs using batch size 32 and learning rate 5×10^{-6}.

We aim to predict correctedness for the following models: Mixtral-8x7B-instruct [Jiang et al., 2024], OLMo-2-1124-7B-Instruct [OLMo et al., 2024], Llama-3.1-8B-Instruct [Grattafiori et al., 2024], phi-4 [Abdin et al., 2024], Llama-3.1-Nemotron-Nano-8B [Bercovich et al., 2025], Llama-3.3-70B-Instruct [Grattafiori et al., 2024], Llama-3.3-Nemotron-Super-49B [Bercovich et al., 2025], and s1.1-32B [Muennighoff et al., 2025]. Additional dataset and model details can be found in Appendix A and B.

3.3 Router Based on Difficulty Prediction

We design a router using the difficulty level predictor from Section 3.1. A problem is assigned to a larger model if the predicted difficulty exceeds a threshold, and to a smaller model otherwise. The router is evaluated on the evaluation split of MathCombined.

Using the embeddings from a smaller model (e.g. Llama Nemotron 8B), one can still train a decent difficulty predictor and router. Similar predictor and router performance using Llama Nemotron 8B's layers can be found in Appendix C.

3.4 Router Based on Model Accuracy Prediction

We build another router using the model accuracy predictor from Section 3.2. Each problem is assigned to the weakest model whose predicted correctness exceeds a threshold. The router is evaluated on the evaluation split of MathCombined.

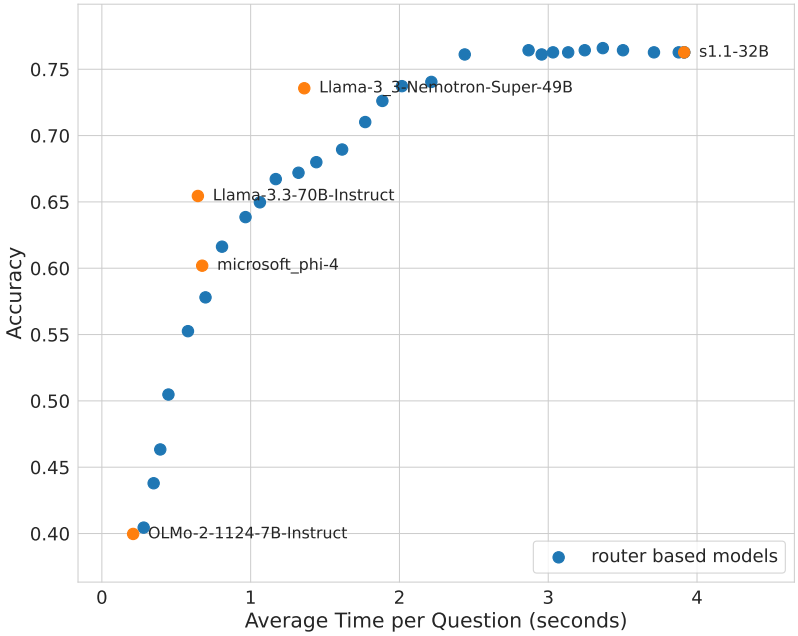

Figure 4: Performance of accuracy-based routing using s1.1-32B layer outputs. Each problem is routed to the weakest model with predicted correctness above a threshold. Blue dots correspond to thresholds between 0.05 and 0.9.

Finally, while ideally we would leverage intermediate representations from all candidate models to capture model-specific interpretations of problem difficulty, we adopt a pragmatic approach using embeddings from a single representative model (S1.1-32B). This approach assumes that difficulty patterns captured by one sufficiently capable model can generalize across similar model architectures.

4 Results

Figure 2 shows prediction performance when using outputs from different layers of s1.1-32B to estimate either problem difficulty or model correctness. We find that middle layers yields the best prediction models, which suggests that middle layers contain more information relevant to problem difficulty. This aligns with previous findings that intermediate layers often have better performance in downstream tasks than the final layer [Skean et al., 2025]. Accordingly, we use layer 45 and the prediction models trained on it for the routers.

The router based on these prediction models yields reasoning models with higher overall efficiency. Figure 3 shows performance of router based on difficulty prediction. With a router, we are able to obtain models that outperforms randomly assigning between the two given models. Figure 4 shows performance of router based on model accuracy prediction. With the appropriate thresholds, our proposed model achieves comparable and even slightly better performance as s1.1-32B, while requiring only about two-thirds of the inference compute.

5 Discussion

We proposed a difficulty classifier to improve inference efficiency on reasoning tasks by routing problems to the smallest model likely to solve them. Beyond efficiency, such classifiers have broader utility. They can provide automatic difficulty annotations for datasets, support curriculum learning and evaluation, and enable selective abstention when a problem is predicted to be too hard. Future work include improving the classifiers by analyzing the embedding space. For example, one could explore routing based on similarity in this space, such as assigning a problem to the model that has successfully solved its nearest neighbors. More generally, integrating richer routing strategies and extending beyond a single representative model may further enhance both efficiency and robustness.

References

Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J. Hewett, Mojan Javaheripi, Piero Kauffmann, James R. Lee, Yin Tat Li, Weishung Liu, Caio C. T. Mendes, Anh Nguyen, Eric Price, Gustavo de Rosa, Olli Saarikivi, Adil Salim, Shital Shah, Xin Wang, Rachel Ward, Yue Wu, Dingli Yu, Cyril Zhang, and Yi Zhang. Phi-4 technical report. Technical Report MSR-TR-2024-57, Microsoft Research, December 2024. URL `https://www.microsoft.com/en-us/research/wp-content/uploads/2024/12/P4TechReport.pdf`.

Akhiad Bercovich, Itay Levy, Izik Golan, Mohammad Dabbah, Ran El-Yaniv, Omri Puny, Ido Galil, Zach Moshe, Tomer Ronen, Najeeb Nabwani, et al. Llama-nemotron: Efficient reasoning models. *arXiv preprint arXiv:2505.00949*, 2025.

Wenhu Chen, Ming Yin, Max Ku, Pan Lu, Yixin Wan, Xueguang Ma, Jianyu Xu, Xinyi Wang, and Tony Xia. Theoremqa: A theorem-driven question answering dataset. *arXiv preprint arXiv:2305.12524*, 2023.

Yu-Neng Chuang, Prathusha Kameswara Sarma, Parikshit Gopalan, John Boccio, Sara Bolouki, Xia Hu, and Helen Zhou. Learning to route llms with confidence tokens. In *Forty-second International Conference on Machine Learning*, 2025a.

Yu-Neng Chuang, Leisheng Yu, Guanchu Wang, Lizhe Zhang, Zirui Liu, Xuanting Cai, Yang Sui, Vladimir Braverman, and Xia Hu. Confident or seek stronger: Exploring uncertainty-based on-device llm routing from benchmarking to generalization. *arXiv preprint arXiv:2502.04428*, 2025b.

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. *arXiv preprint arXiv:2110.14168*, 2021.

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

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. *arXiv preprint arXiv:2407.21783*, 2024.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Leng Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, Jie Liu, Lei Qi, Zhiyuan Liu, and Maosong Sun. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems, 2024.

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. *arXiv preprint arXiv:2103.03874*, 2021.

Qitian Jason Hu, Jacob Bieker, Xiuyu Li, Nan Jiang, Benjamin Keigwin, Gaurav Ranganath, Kurt Keutzer, and Shriyash Kaustubh Upadhyay. Routerbench: A benchmark for multi-llm routing system. *arXiv preprint arXiv:2403.12031*, 2024.

Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. Mixtral of experts. *arXiv preprint arXiv:2401.04088*, 2024.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, et al. Solving quantitative reasoning problems with language models. *Advances in neural information processing systems*, 35:3843–3857, 2022.

Zhong-Zhi Li, Duzhen Zhang, Ming-Liang Zhang, Jiaxin Zhang, Zengyan Liu, Yuxuan Yao, Haotian Xu, Junhao Zheng, Pei-Jie Wang, Xiuyi Chen, et al. From system 1 to system 2: A survey of reasoning large language models. *arXiv preprint arXiv:2502.17419*, 2025.

Keming Lu, Hongyi Yuan, Runji Lin, Junyang Lin, Zheng Yuan, Chang Zhou, and Jingren Zhou. Routing to the expert: Efficient reward-guided ensemble of large language models. *arXiv preprint arXiv:2311.08692*, 2023.

Mathematical Association of America. American mathematics competitions (amc) 2023. `https://maa.org/math-competitions`, 2023.

Mathematical Association of America. American invitational mathematics examination (aime) 2024. `https://maa.org/math-competitions`, 2024.

Alireza Mohammadshahi, Arshad Rafiq Shaikh, and Majid Yazdani. Routoo: Learning to route to large language models effectively. *arXiv preprint arXiv:2401.13979*, 2024.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. s1: Simple test-time scaling. *arXiv preprint arXiv:2501.19393*, 2025.

Team OLMo, Pete Walsh, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Shane Arora, Akshita Bhagia, Yuling Gu, Shengyi Huang, Matt Jordan, et al. 2 olmo 2 furious. *arXiv preprint arXiv:2501.00656*, 2024.

OpenAI. Introducing gpt-5. `https://openai.com/index/introducing-gpt-5/`, August 2025. OpenAI blog post announcing GPT-5; accessed 24 August 2025.

Xiao Pu, Michael Saxon, Wenyue Hua, and William Yang Wang. Thoughtterminator: Benchmarking, calibrating, and mitigating overthinking in reasoning models. *arXiv preprint arXiv:2504.13367*, 2025.

Swarnadeep Saha, Archiki Prasad, Justin Chen, Peter Hase, Elias Stengel-Eskin, and Mohit Bansal. System 1.x: Learning to balance fast and slow planning with language models. In *The Thirteenth International Conference on Learning Representations*, 2025.

Marija Šakota, Maxime Peyrard, and Robert West. Fly-swat or cannon? cost-effective language model choice via meta-modeling. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, pages 606–615, 2024.

Tal Shnitzer, Anthony Ou, Mírian Silva, Kate Soule, Yuekai Sun, Justin Solomon, Neil Thompson, and Mikhail Yurochkin. Large language model routing with benchmark datasets. *arXiv preprint arXiv:2309.15789*, 2023.

Oscar Skean, Md Rifat Arefin, Dan Zhao, Niket Patel, Jalal Naghiyev, Yann LeCun, and Ravid Shwartz-Ziv. Layer by layer: Uncovering hidden representations in language models. *arXiv preprint arXiv:2502.02013*, 2025.

DiJia Su, Sainbayar Sukhbaatar, Michael Rabbat, Yuandong Tian, and Qinqing Zheng. Dualformer: Controllable fast and slow thinking by learning with randomized reasoning traces. *arXiv preprint arXiv:2410.09918*, 2024.

Yang Sui, Yu-Neng Chuang, Guanchu Wang, Jiamu Zhang, Tianyi Zhang, Jiayi Yuan, Hongyi Liu, Andrew Wen, Shaochen Zhong, Hanjie Chen, et al. Stop overthinking: A survey on efficient reasoning for large language models. *arXiv preprint arXiv:2503.16419*, 2025.

Yiliu Sun, Yanfang Zhang, Zicheng Zhao, Sheng Wan, Dacheng Tao, and Chen Gong. Fast-slow-thinking: Complex task solving with large language models. *arXiv preprint arXiv:2504.08690*, 2025.

Xiaoqiang Wang, Suyuchen Wang, Yun Zhu, and Bang Liu. System-1.5 reasoning: Traversal in language and latent spaces with dynamic shortcuts. *arXiv preprint arXiv:2505.18962*, 2025.

A Additional Dataset Details

The MathCombined dataset, which consists of 3136 reasoning tasks with ground truth solution but without difficulty levels, includes the following data:

- AIME24 [Mathematical Association of America, 2024] (30 problems from the 2024 AIME I and AIME II tests)
- AMC23 [Mathematical Association of America, 2023] (40 problems from the 2023 AMC 12)
- GSM8k [Cobbe et al., 2021] (1319 grade school math problems)
- Minerva [Lewkowycz et al., 2022] (272 STEM problems at the undergraduate level)
- OlympiadBench [He et al., 2024] (675 challenging math problems)
- TheoremQA [Chen et al., 2023] (800 question-answering covering theorems from math, physics, EE&CS, and finance)

The MATH dataset [Hendrycks et al., 2021] consists of 7500 problems from various mathematics competitions, together with the difficulty level of each problem. Figure 5 shows the distribution of difficulty levels in the MATH dataset.

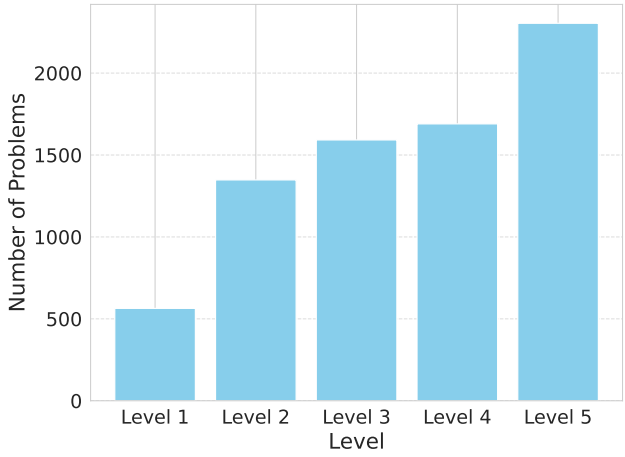

Figure 5: Difficulty level distribution of the MATH dataset.

B Additional Model Details

Figure 6 shows the percentage of questions answered correctly versus the average inference time per problem, for all models on the MathCombined dataset. As a high level trend, models achieving higher accuracy tend to need longer inference time. Notable exceptions are Mixtral-8x7B-instruct and Llama-3.1-8B-Instruct, both of which are not reasoning models, as well as Llama-3.1-Nemotron-Nano-8B. We therefore remove them from the set of models available to the routers.

Figure 7 provides a more fine grained comparison between all models. For each pair of models i, j, we examine the number of questions model i answers correctly but model j does not. Interestingly, for all pairs of models, there are a nonzero number of questions that the first model can answer but the second cannot. Even the strongest model (s1.1-32B) fails at problems that the weakest model (Mixtral-8x7B-instruct) answers correctly. This justifies the possibility that our router based model can achieve higher accuracy than always using the strongest model.

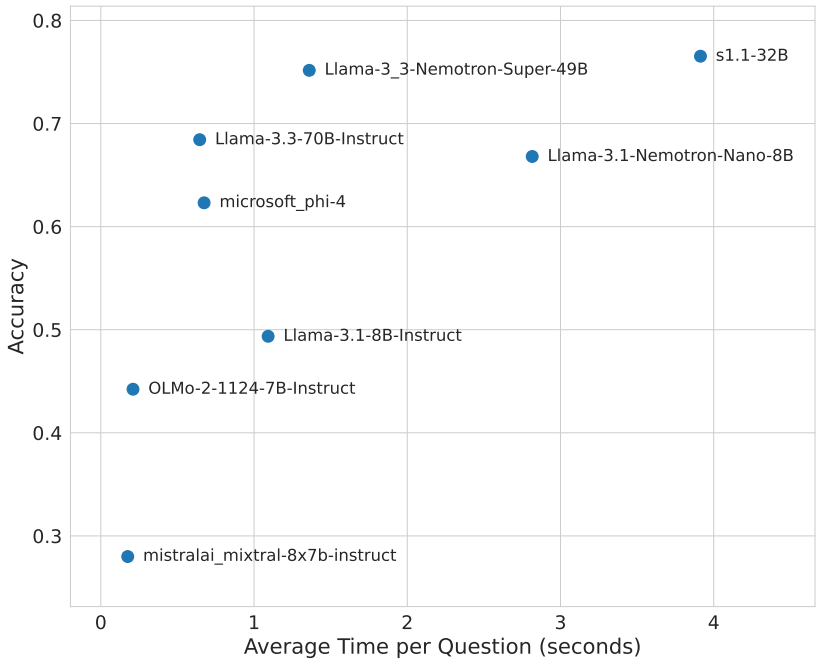

Figure 6: Comparison of models' performance on MathCombined.

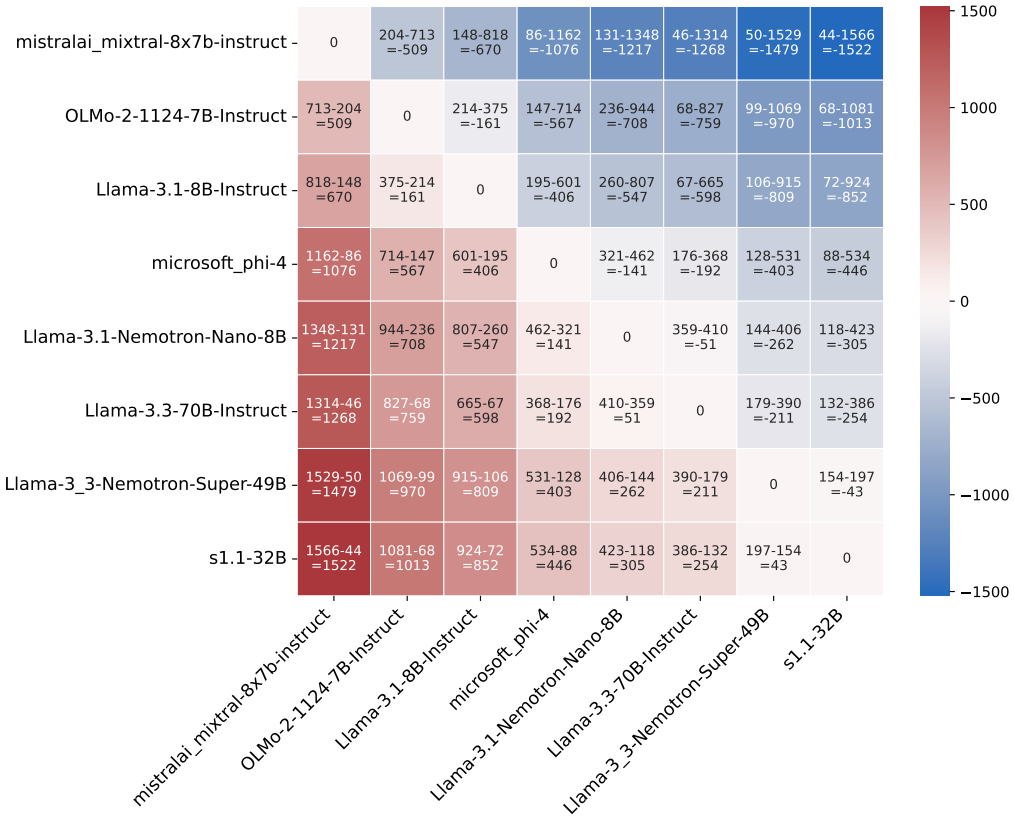

Figure 7: Heatmap comparing models' performance on MathCombined. Label[i][j]: total number of questions model i answers correctly but model j does not − total number of questions model j answers correctly but but model i does not = difference.

C Llama Nemotron 8B Results

In the main paper, we used the embeddings from s1.1-32B to train the difficulty predictor. We show below that using the embeddings from a smaller model, one can still train a decent difficulty predictor and router. Below, we show predictor and router performance using Llama Nemotron 8B. The routers use the prediction models trained on embeddings from layer 20, which has one of the strongest performance when used to predict problem difficulty.

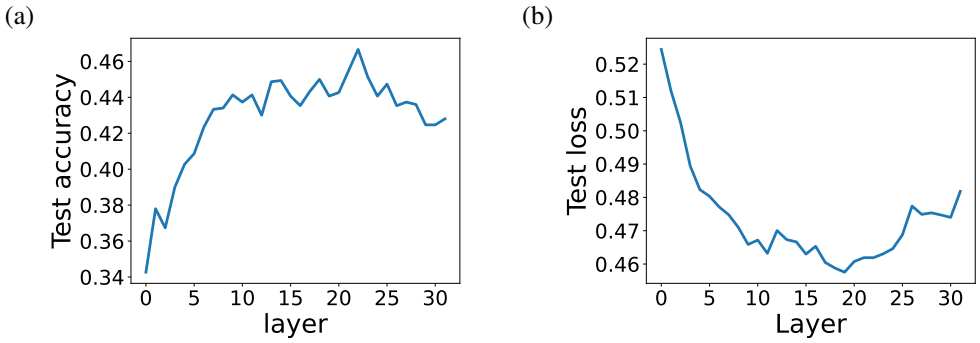

Figure 8: Prediction performance using outputs from different layers of Llama-3.1-Nemotron-Nano-8B-v1 on (a) question difficulty level and (b) whether various language models can answer the given question correctly. Middle layers provide the most informative representations.

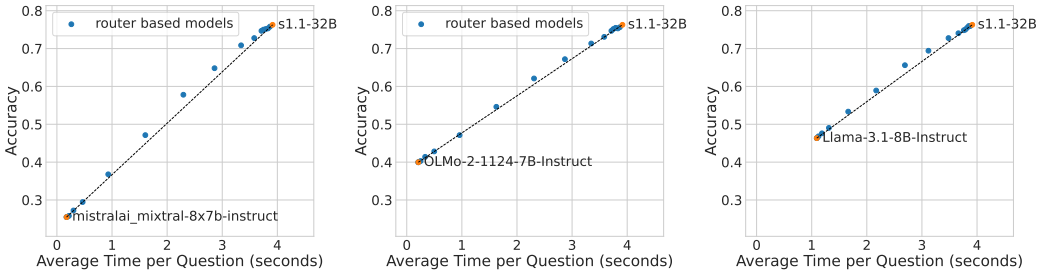

Figure 9: Performance of difficulty-based routing using Llama-3.1-Nemotron-Nano-8B-v1 layer outputs. A problem is routed to a larger model if the predicted difficulty exceeds a threshold, and to a smaller model otherwise. Blue dots indicate router-based systems with thresholds between 2.1 and 2.9; orange dots show baseline models. Routers consistently outperform random assignment.

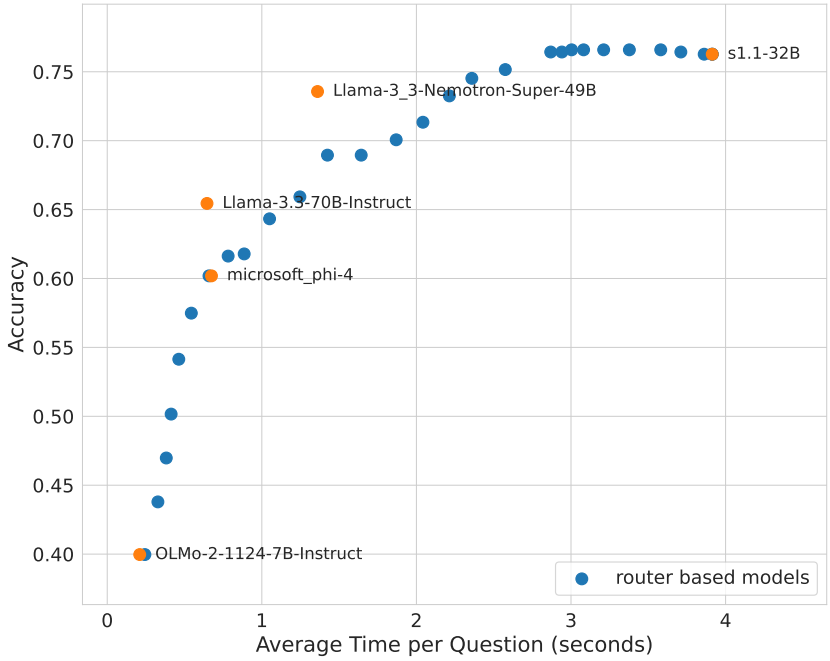

Figure 10: Performance of accuracy-based routing using Llama-3.1-Nemotron-Nano-8B-v1 layer outputs. Each problem is routed to the weakest model with predicted correctness above a threshold. Blue dots correspond to thresholds between 0.05 and 0.9.