Attention, Please: Single-Head Cross-Attention for Unified LLM Routing

Anonymous Author(s)
Affiliation
Address
email

Abstract

The growing diversity of language models, ranging from lightweight (small), cost-efficient models to powerful (large) but expensive ones, has made dynamic model selection essential for scalable and cost-effective deployment. We propose a unified LLM routing framework that jointly models query and model embeddings using a single-head cross-attention mechanism. We evaluate router's decision-making capabilities on a large-scale, publicly available dataset called RouterBench [2], that enables evaluation across multiple LLM pools and domains. By capturing fine-grained query-model interactions, our router learns to predict both response quality and generation cost, outperforming existing predictive routers by up to 6.6% in Average Improvement in Quality (AIQ) and 2.9% in maximum performance. To better reflect the trade-off between performance and cost, we adopt a new exponential reward function with improved robustness. Our architecture is lightweight, generalizes well across various domains, and is more efficient than existing ones.

1 Introduction

The rise of large language models (LLM) has advanced reasoning, summarization, and code generation. Yet, the wide range of options, from lightweight, cost-efficient models (e.g., Mistral 7B [3]) to powerful but costly ones (e.g., GPT-4 [1]), creates a core challenge: selecting the right model per query to balance response quality and cost. This challenge is critical for hyperscalers, where both efficiency and user experience are paramount.

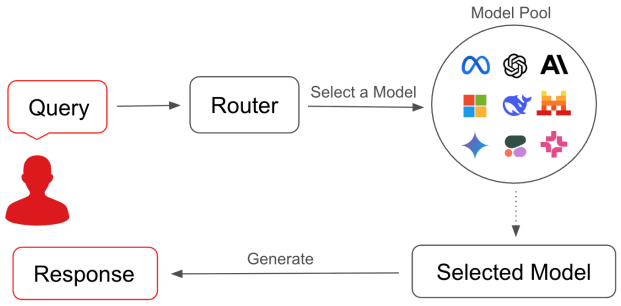


Figure 1: LLM Router selects an appropriate model for each query to route to

Recent work has proposed diverse strategies for LLM routing to balance quality and cost. Classification-based methods predict the best model from static query features [9, 8, 12], but assume fixed model sets and rely on pre-computed metrics, limiting adaptability. Reinforcement learning–based approaches learn dynamic policies [7, 11], yet require many interactions to converge and suffer in cold-start scenarios. Training-free and heuristic methods such as LLM-BLENDER[4] boost accuracy by combining outputs, but incur the overhead of querying all models. Similarity-based approaches [13, 10] leverage embeddings or domain classifiers, but often depend on static experts or domain-specific assumptions.

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

A common limitation of prior approaches is treating queries and models as independent, with routing based on query-only features or model-agnostic heuristics. Our method instead models query–model interactions via cross-attention, allowing the router to assess how a model will perform on a given query, enabling flexible, domain-agnostic routing. A detailed discussion is deferred to Appendix A.

Empirical results on RouterBench demonstrate that our attention-based router consistently achieves higher AIQ score across different LLM pools, outperforming traditional baselines- KNN, MLP and SVM routers by an average of 23.85%, 3.34% and 27.33% respectively. Ablation studies confirm that attention-based predictors consistently outperform regression and MLP variants, with up to 10.9% higher AIQ and 9.7% higher maximum performance. These findings highlight the importance of modeling query–model interactions for scalable and efficient LLM[1] routing.

2 Method

Problem Formulation Given a pool of LLMs $\mathcal{M} = \{M_1, M_2, ..M_K\}$ and user query space $\mathcal{Q}$, our goal is to design an LLM router (Figure 1) as a decision-making agent $\Pi : \mathcal{Q} \rightarrow \mathcal{M}$, mapping queries to models under response uncertainty. The router is designed to balance the trade-off between performance and cost, optimizing the competing objectives of maximizing response quality while minimizing resource usage. Our guiding principle is that a query should only be routed to an expensive model if all cheaper models fail to give a promising response and the user is willing to pay the additional cost.

Predictor-based Routing Framework In this work, we employ a predictor-based LLM routing framework and propose attention as an effective architecture to estimate the response quality or the generation cost of candidate models. Based on these estimates, the framework selects the most suitable model by incorporating user's willingness to pay in the reward function, thus decoupling predictors training from user's parameter. To ensure scalability across the model pool, we design a dual-predictor framework where one predictor estimates performance of all models, while the other estimates generation cost. Intuitively, as the user's willingness to pay increases, the framework places greater emphasis on response quality while discounting the cost factor. Later, we perform a systematic study on choosing an appropriate reward function for this framework 4.

Attention-Based Predictors. We propose similarity-based routing using a single-head cross-attention which encodes the incoming prompts as queries and LLM representations (See Appendix C) as keys and values. We presume query ($\vec{\mathbf{q}}$) captures level of prompt's complexity in multi-dimensions, while key ($\vec{\mathbf{k}}$) and value ($\vec{\mathbf{v}}$) express LLM's expertise in these dimensions. This predictor captures query–model interactions through attention, enabling it to estimate expected performance and generation cost of response from each model for a given prompt.

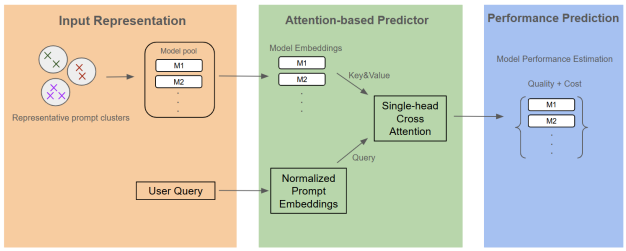


Figure 2: Single-head cross-attention block

$$\text{Attention}(\vec{\mathbf{q}}, \vec{\mathbf{k}}, \vec{\mathbf{v}}) = \text{softmax}\left(\frac{\vec{\mathbf{q}}.\vec{\mathbf{k}}^T}{\sqrt{d_v}}\right)\vec{\mathbf{v}}.$$

This similarity-based routing framework offers several advantages. By decoupling model representation creation from the training process, it remains adaptive to an evolving model pool with minimal retraining effort. The cross-attention module maps queries and model embeddings into a shared latent space, without restraining sizes of prompt and LLM representations. As a second-order similarity mechanism, attention could potentially capture richer and more nuanced interactions between queries and models than dot product or cosine similarity. Further, its inherent parallelism makes the router scalable to heavy query traffic, effectively handle larger LLM pools and batched queries.

[1]As the definition of 'large' language models evolves, we use 'LLM' to refer to the models included in our experimental pool.

LLM Pool	AIQ ↑		$\lambda-$ sensitivity$_{\text{perf}}$ ↓		$\lambda-$ sensitivity$_{\text{cost}}$ ↓	
	R_1	R_2	R_1	R_2	R_1	R_2
Pool 1	0.85616	0.84221	0.0155	**0.0035**	2.66e-05	**1.55e-05**
Pool 2	0.83285	0.83366	0.0213	**0.0027**	8.15e-06	**5.24e-06**
Pool 3	0.87512	0.87362	0.0260	**0.0040**	1.88e-05	**1.29e-05**
Pool 4	0.80434	0.83626	0.0258	**0.0019**	4.74e-05	**2.30e-05**

Table 1: Comparison between R_1 and R_2 oracle routers on the basis of AIQ score, and performance and cost sensitivity with λ. Higher AIQ score means better cost-efficiency, while lower sensitivity indicates robustness of the oracle router to minor variations in λ, thereby the reward formulation.

3 Evaluation Methodology

Data. We evaluate the generalization of our proposed routing model across multiple domains on RouterBench [2]. It is a large-scale, public dataset designed to evaluate multi-LLM routing systems and contains responses from 11 LLMs on 8 benchmarks, including MMLU, GSM8K, HellaSwag, ARC Challenge, Winogrande, MBPP and MT-Bench datasets. (More details in Appendix B)

Baselines. We compare our proposed predictive router with established baseline routers [2] including multi-layer perceptron (MLP), support vector machines (SVM) and K nearest neighbors (KNN), as well as other proposed predictive routers. Our gold standard is the oracle router from RouterBench [2]. We draw parallels with oracle routers using different reward functions to identify the most appropriate formulation (see Section 4 and Table 1). Notably, the oracle router based on our exponential rewards achieves the best cost–performance trade-off, routes less queries to the expensive model, and is less sensitive to the user parameter.

Evaluation Metrics For evaluating cost-efficiency of the routers, we plot a pareto frontier on cost-performance plane [2] using the average cost vs performance points, obtained by varying user's willingness to pay (λ). The area under this convex hull divided by the cost range gives **Average Improvement over Quality (AIQ)**, thus aggregating the router's trade-off into a single metric. The AIQ is defined as $\text{AIQ}(R) = \text{Area}_{\text{cost-perf}}/(b-a)$, where $[a, b]$ represent the cost range. Intuitively, higher AIQ score represents better performance is achieved for most of the cost range and vice versa, lower generation cost for most of the performance range. It indicates how well the router trades-off conflicting goals - performance and generation cost, thus serving as our primary metric.

To identify an appropriate reward function for the predictive-routers framework, we employ λ-sensitivity, along with AIQ scores. **λ-sensitivity** (See Appendix E) measures the oracle router's sensitivity to changes in the user's willingness to pay (λ). We define performance sensitivity as the weighted average of the variation in the performance over a log scale difference of λ and cost sensitivity analogously.[2] Lower λ-sensitivity indicates the oracle router remains stable and consistent under small changes in λ, without severe degradation in performance/cost.

Along with AIQ score, we report **maximum performance** attained over the range of user's parameter.

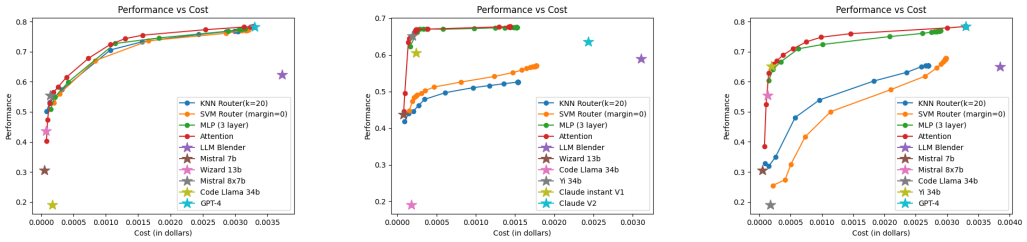
4 Results and Discussion

Reward functions. We determine an appropriate reward function in the predictive framework by evaluating oracle routers associated with them. We compare 2 reward functions, the traditionally used linear trade-off (R_1 in Eqn. (1)) and novel exponential trade-off (R_2 in Eqn. (1)), proposing the latter as an appropriate reward function. For a user prompt q with an LLM response r, the performance $s(q, r)$ and generation cost $c(q, r)$ are combined in reward functions as:

$$R_1 = s - \frac{1}{\lambda}c, \quad R_2 = s \times \exp\left(-\frac{1}{\lambda}c\right) \tag{1}$$

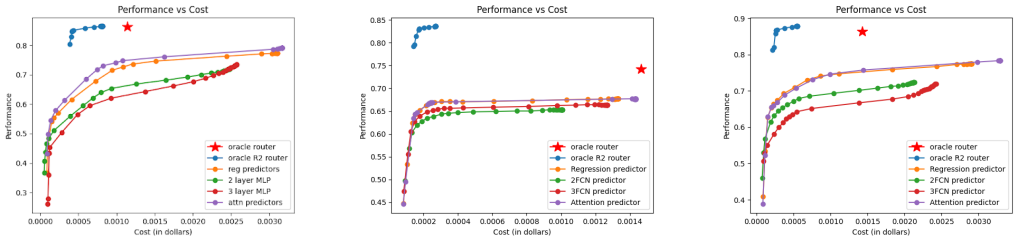
Though both the reward functions have similar AIQ scores (in Table 1) while routing at most 20% of user prompts to the expensive LLM, the λ-sensitivity of R_2 oracle router is drastically lesser than that

[2]We intentionally keep the performance and cost sensitivity distinct, for fine-grained results and their differing orders of magnitude.



(a) LLM pool 1 (b) LLM pool 2 (c) LLM pool 3

Figure 3: Comparison of Attention Router with RouterBench Baseline Routers



(a) LLM pool 1 (b) LLM pool 2 (c) LLM pool 3

Figure 4: Cost-efficiency of predictors in predictor-based routing framework

of R_1 router, indicating stability of R_2 reward function over traditional linear trade-off R_1. This could be attributed to the boundedness of the exponential trade-off while linear trade-off is unbounded.

Cost-efficiency of LLM Routers Figure 3 and Table 2 show the performance of our attention-based router against KNN, SVM, and MLP baselines. The key metric is Average Improvement in Quality (AIQ), defined as the area under the cost–quality Pareto frontier. Across all LLM pools, the attention router achieves higher AIQ. In LLM pool 1, it improves AIQ by at least 3% over baselines. In other pools with similarly performing models, it outperforms KNN and SVM by at least 33.58% and 29.41%, respectively, along with higher maximum performance (Perf_{Max}). These results highlight the advantage of our similarity-based routing objective in both performance and cost-efficiency, and demonstrates that attention-based routing provides a robust and generalizable improvement.

Router	LLM Pool 1		LLM Pool 2		LLM Pool 3	
	AIQ ↑	Perf_{Max} ↑	AIQ ↑	Perf_{Max} ↑	AIQ ↑	Perf_{Max} ↑
KNN router (k=20)	*0.70608*	0.76912	0.49338	0.52573	0.55727	0.65385
MLP router	0.67598	0.73781	*0.66564*	*0.67551*	*0.72655*	*0.76975*
SVM router (margin=0)	0.70220	*0.77233*	0.51452	0.57024	0.49760	0.67767
Attention router (R_2)	**0.72737**	**0.78082**	**0.66586**	**0.67748**	**0.74439**	**0.78347**
LLM Blender	-	0.62314	-	0.58982	-	0.64905

Table 2: Comparison of router's performance and cost-efficiency with traditional routers

Ablation: Different Architectures in predictor-based LLM routing framework We ablate predictor architectures (regression, MLP, attention) and domains, finding attention-based router outperforms other predictive routers by up to 6.6% in AIQ and 2.9% in maximum performance. Detailed results are presented in Figure 4 and Tables 3–6 in Appendix H.

5 Conclusion

We propose a predictor-based LLM routing framework with dual predictors and a cross-attention similarity module, delivering strong cost–performance trade-offs and competitive AIQ. Its attention-based design enables scalable, plug-and-play routing for evolving LLM pools, and we also implement a new exponential reward formulation with improved robustness across multiple scenarios.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[2] 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.

[3] AQ Jiang, A Sablayrolles, A Mensch, C Bamford, DS Chaplot, D de Las Casas, F Bressand, G Lengyel, G Lample, L Saulnier, et al. Mistral 7b. corr, abs/2310.06825, 2023. doi: 10.48550. *arXiv preprint ARXIV.2310.06825*, 10, 2023.

[4] Dongfu Jiang, Xiang Ren, and Bill Yuchen Lin. Llm-blender: Ensembling large language models with pairwise comparison and generative fusion. In *Proceedings of the 61th Annual Meeting of the Association for Computational Linguistics (ACL 2023)*, 2023.

[5] Dongfu Jiang, Xiang Ren, and Bill Yuchen Lin. Llm-blender: Ensembling large language models with pairwise ranking and generative fusion. *arXiv preprint arXiv:2306.02561*, 2023.

[6] Wittawat Jitkrittum, Harikrishna Narasimhan, Ankit Singh Rawat, Jeevesh Juneja, Zifeng Wang, Chen-Yu Lee, Pradeep Shenoy, Rina Panigrahy, Aditya Krishna Menon, and Sanjiv Kumar. Universal model routing for efficient llm inference. *arXiv preprint arXiv:2502.08773*, 2025.

[7] Yang Li. Llm bandit: Cost-efficient llm generation via preference-conditioned dynamic routing. *arXiv preprint arXiv:2502.02743*, 2025.

[8] Yueyue Liu, Hongyu Zhang, Yuantian Miao, Van-Hoang Le, and Zhiqiang Li. Optllm: Optimal assignment of queries to large language models. In *2024 IEEE International Conference on Web Services (ICWS)*, pages 788–798. IEEE, 2024.

[9] 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*, 2024.

[10] Josef Pichlmeier, Philipp Ross, and Andre Luckow. Domain-aware llm routing during generation. In *2024 IEEE International Conference on Big Data (BigData)*, pages 8235–8237. IEEE, 2024.

[11] Dimitrios Sikeridis, Dennis Ramdass, and Pranay Pareek. Pickllm: Context-aware rl-assisted large language model routing. *arXiv preprint arXiv:2412.12170*, 2024.

[12] Seamus Somerstep, Felipe Maia Polo, Allysson Flavio Melo de Oliveira, Prattyush Mangal, Mírian Silva, Onkar Bhardwaj, Mikhail Yurochkin, and Subha Maity. Carrot: A cost aware rate optimal router. *arXiv preprint arXiv:2502.03261*, 2025.

[13] Dimitris Stripelis, Zijian Hu, Jipeng Zhang, Zhaozhuo Xu, Alay Dilipbhai Shah, Han Jin, Yuhang Yao, Salman Avestimehr, and Chaoyang He. Tensoropera router: A multi-model router for efficient llm inference. *arXiv preprint arXiv:2408.12320*, 2024.

[14] Asterios Tsiourvas, Wei Sun, and Georgia Perakis. Causal llm routing: End-to-end regret minimization from observational data. *arXiv preprint arXiv:2505.16037*, 2025.

[15] Zesen Zhao, Shuowei Jin, and Z Morley Mao. Eagle: Efficient training-free router for multi-llm inference. *arXiv preprint arXiv:2409.15518*, 2024.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and/or introduction should clearly state the main topics and contributions of this work.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Yes, limitations are discussed in Section K.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification: The paper does not include theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The main text along with the appendix mentions all the implementation details and experimental setup

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: While there is an open access to the data, we would like to disclose the code after the final decision about manuscript submission.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: The details are present in Appendix B, C.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: The results are reproduced for multiple sets of LLMs to establish robustness of results. However, the current version of the manuscript does not report any error bars or confidence intervals.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)

- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The details are included in Appendix J.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conforms with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Positive impact could be the society as well as service providers benefit from LLM inference cost savings, while achieving near-optimal performance. This high-level impact is mentioned in introduction and conclusion sections.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The router models do not have any direct high risks for misuse.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All the LLMs used are clearly attributed and original papers are cited whereever required.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: The paper leverages existing public datasets and existing LLMs / SLMs for the experiments.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: Our methodology is not developed using LLMs.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

A Related Work

As large language models (LLMs) proliferate across domains and deployment settings, the challenge of selecting the most appropriate model for a given query has become central to efficient and effective LLM usage. Prior work on LLM routing has largely focused on optimizing for cost, quality, or adaptability — but often treats these objectives in isolation or relies on rigid assumptions about model behavior. Our work builds on this foundation by proposing a unified, interaction-based routing framework that jointly models query and LLM characteristics to make informed, flexible routing decisions.

Early approaches to LLM routing framed the problem as a classification task, where a supervised model predicts the best LLM for a query. These methods, such as RouteLLM [9] and OptLLM [8], demonstrated that static query features could be used to reduce reliance on expensive models. Recent work, CARROT [12], predicts both cost and accuracy to select the most cost-effective model, and deduces minimax optimality under certain assumptions. However, they often assume a fixed set of models and rely on pre-computed performance metrics, limiting their ability to generalize to new domains or adapt to evolving model pools.

In parallel, reinforcement learning-based methods introduced dynamic routing policies that adapt over time. Methods like LLM Bandit [7] and PickLLM [11] use online feedback to refine model selection strategies. While these methods offer adaptability, they typically require many interactions to converge and struggle with cold-start scenarios — a critical limitation in real-world deployments where immediate routing decision is essential. Causal LLM Routing [14] learns routing policies from observational data via end-to-end regret minimization, avoiding costly full-feedback datasets, but still depends on rich historical logs and omits explicit query–model interaction modeling.

Immediate routing decisions can be achieved by exploring training-free or heuristic-based routing, aimed to reduce overhead by avoiding model training altogether. Eagle [15] and Universal Model Routing [6] use ranking systems or unsupervised clustering to guide routing decisions. These methods are appealing for their simplicity and low cost, but often lack the granularity needed to capture nuanced differences in model behavior, especially for complex or ambiguous queries. From another angle, LLM-BLENDER [4] ensembles multiple LLMs by ranking and fusing their generated outputs using a pairwise ranking module and a generative fusion module. While effective, this post-generation approach requires outputs from all candidate models, in contrast to our pre-generation routing perspective, which selects a single model before inference.

Across these diverse approaches, a common limitation emerges: most methods treat the query and model as independent entities, relying on either query-only features or model-agnostic heuristics. In contrast, our approach explicitly models the interaction between query and model embeddings, allowing the router to reason about how a specific query might perform on a specific model. By learning to predict both quality and cost in a unified framework, our method supports flexible, domain-agnostic routing that adapts to new models and tasks with minimal supervision.

B RouterBench Dataset

1. The dataset contains at most 1 response per model for each user prompt. However, the same model can answer the same question with multiple different responses.
2. Analysis on the dataset in the paper [2] shows that most of the answers that can be answered by an expensive model, can also be answered by smaller models as well. So, a cost-efficient router should learn to discern when a query can be routed to a smaller model.
3. For all proprietary models, we calculate the cost of input and output results based on their API pricing, and Together AI for open-source model.
4. Performance is not quite precise. Most of the queries have binary performance, since they have ground-truth and others have response quality in $\{0, 0.1, 0.25, 0.5, 0.75, 1\}$.
5. There are many multi-choice prompts (around 27k prompts out of 36k prompts are multiple choice.).
6. For the datasets MMLU, HellaSwag, GSM8K, ARC Challenge, and Winogrande, responses are evaluated using exact match method, while for MBPP, MT-Bench, and RAG, GPT evaluates responses, further are normalized to unit scale.

7. Cost is in the order of $10^{(-5)}$ mostly.

Our train-validation-test split is 75%:5%:20% on complete data and further, did analysis on domain-wise data.

C Method Details

Predictors. We primarily explored 2 architectural variants:

1. **Regression-Based Predictors:** Linear regressor learns the best fit regression line for each model with query as input and performance/cost as output. Although interpretable and computationally efficient, they are limited to capture linear relationships only.

$$QX_1 = S \in \mathcal{R}^{n \times q} \quad QX_2 = C \in \mathcal{R}^{n \times q}$$

2. **Neural Network Predictors:** Fully connected networks (2-layer and 3-layer MLP) learn mapping from query to all the model's performance/cost predictions. $f(q;\theta) = S \in \mathcal{R}^m$ and $g(q;\theta) = C \in \mathcal{R}^m$. Note that, unlike regression-based predictors, predictions for different language models share parameters here.

Model Representations Augmentation (represented as Reg-emb, 2FCN-emb and 3FCN-emb in H) In this improvement, we augment the respective model representation along with the query embeddings as context to the above predictors. De-coupling creation of model representations from training allows us to dynamically add/remove models from the pool during the inference time.

Given a query embedding $\mathbf{q} \in \mathbb{R}^{d_q}$ and a model embedding $\mathbf{m} \in \mathbb{R}^{d_m}$, the input is formed as $\mathbf{x} = [\mathbf{q}; \mathbf{m}]$ and outputs a scalar value $s/c \in \mathcal{R}$ of performance/cost for this model.

User prompt Embeddings. We employ DistilBERT embeddings (dim=768) of user prompts and further normalize them before attention computation.

LLM Representations. We compute these fixed-size model embeddings to best capture "latent expertise" of the models across different domains. We firstly cluster a large set of queries (training set) using K-Means and pick 20% of prompts as representative prompts from each cluster uniformly random. Given C clusters, a model embedding $\mathcal{I}_{\mathbf{m}} \in \mathbf{R}^C$ encodes the mean performance per cluster. This is a training-free approach, inspired by Universal Routing [6]. As mentioned in this paper, we take the large set of prompts for clustering, rather than using a small set of representative prompts is that it could lead to overfitting.

Our hypothesis is by incorporating model embeddings, the router gains a richer understanding of model-specific capabilities, enabling improved query-model matching and accurate quality prediction, especially for diverse and complex queries.

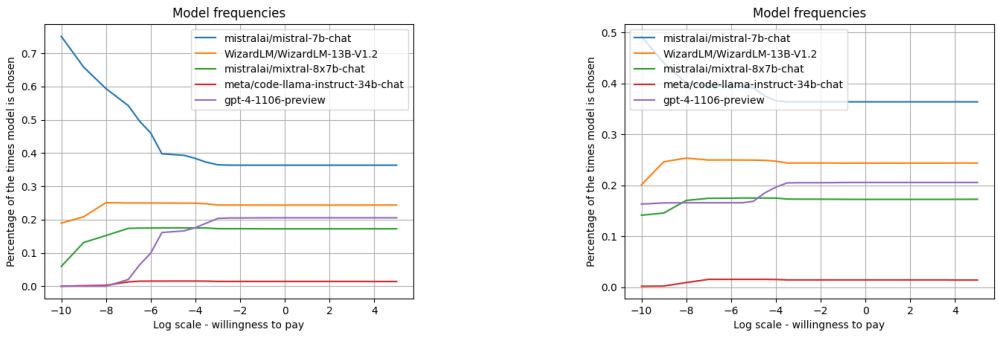
Training and Test Details. We trained all the predictors in the predictor-based routing framework with MSE (Mean Squared Error) loss, using Adam optimizer and CoineAnnealingLR scheduler. Particularly, we trained Attention-based performance predictor with $1e-3$ learning rate for 1000 epochs, 1024 batch size and $1e-5$ weight decay. Similarly, we trained attention-based cost predictor with $1e-4$ learning rate for 1000 epochs, 1024 batch size and $1e-7$ weight decay, while mapping the inputs to an internal dimension of 20. With train-validation-test splits being 75%, 5% and 20% respectively, we chose these hyper-parameters with the best train and validation loss.

For creating LLM representations, we divided the trainset user prompts into 20 clusters, obtained from elbow test. Thus, the LLM embeddings are 20-dimensional, while prompt embeddings are 768-dimensional.

LLM Blender Implementation We implemented the LLM Blender baseline using the open-source PairRM ranker provided by the LLM-Blender [5] framework. For each prompt in the RouterBench dataset, we collected responses from all candidate models in the pool and generated all possible response pairs. The PairRM ranker was used to perform pairwise comparisons of these responses,

assigning a win to the preferred response in each pair. The model with the highest total number of wins was selected as the routed model for that prompt. This method does not involve any additional training and was applied only to the test set for evaluation. Since all model outputs are required for comparison, the total cost for this method is computed as the sum of all candidate models' inference costs per sample.

D Proposed Reward Functions' Analysis



(a) $R1 = s(q,r) - \frac{1}{\lambda}c(q,r)$ (b) $R2 = s(q,r)exp(-\frac{1}{\lambda}c(q,r))$

Figure 5: Distribution of queries routed to each model in LLM pool 1 by oracle routers with our proposed reward functions

The oracle routers we defined in section 2 are ideal routers. As it can be seen in the above plot, these routers not only attain the best performance-cost trade-off, on par with oracle router in RouterBench but also route most of the queries to a lower cost model, while achieving this trade-off. Maximum number of queries routed to GPT-4 with either of the baselines is 20%, thus verifying that employing these reward functions is an appropriate approach for cost-efficient LLM router.

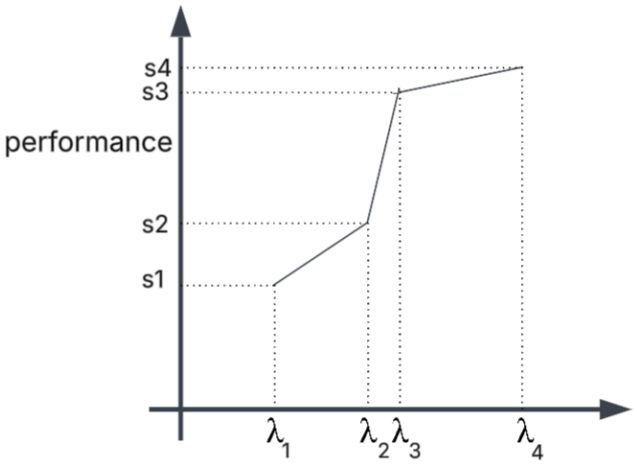


Figure 6: λ-sensitivity

E λ-Sensitivity of Reward functions

In order to gauge the robustness of reward functions, we perform sensitivity analysis of the reward functions with respect to user's willingness to pay (λ), that is how abruptly the performance/inference cost varies with λ. We define λ-sensitivity with respect to performance as the weighted average of the change in performance over the log scale difference in user parameter (λ), formulated as:

Similarly, with the cost. This metric expresses how fast the performance changes with minor variations in λ, indicating instability and inconsistency of the oracle router, in turn the instability of the reward function.

$$\lambda - \text{sensitivity}_{\text{perf}} = \frac{\log(\lambda_2/\lambda_1)(s_2 - s_1) + \log(\lambda_3/\lambda_2)(s_3 - s_2) + \log(\lambda_4/\lambda_3)(s_4 - s_3)}{(\log(\lambda_4/\lambda_1))}$$

F LLM Pools

We conducted experiments on the following LLM pools:
LLM pool 1: Mistral 7B Chat, WizardLM 13B V1.2, Mistral 8x7B Chat, Code Llama Instruct 34B Chat, GPT 4
LLM Pool 2: WizardLM 13B V1.2, Code Llama Instruct 34B Chat, Yi 34B Chat, Claude Instant V1, Claude V2
LLM Pool 3: Mistral 7B Chat, Mistral 8x7B Chat, Code Llama Instruct 34B Chat, Yi 34B Chat, GPT 4
LLM Pool 4: Llama 2 70B, Claude V1, Claude V2, GPT-4

G Detailed Comparison Tables: Across Predictor Models and Domains

H Predictor-based LLM Routing Framework

We conducted an ablation study by varying predictors in the predictor-based routing framework. We observe that the router with attention module as both performance and cost predictors yields the best AIQ score and maximum performance.

H.1 Rewards: R_1

		Cost Predictor							
		Oracle R1	Reg	2-FCN	3-FCN	Reg-emb	2-FCN-emb	3-FCN-emb	Attn
Quality Predictors	Oracle R1	0.85639	0.85442	0.85637	0.85638	0.85771	0.85572	0.85512	0.85830
	Reg	0.72045	0.71810	0.72090	**0.72122**	0.70643	0.72013	0.72007	0.71881
	2-FCN	0.66528	0.66505	0.66563	0.64657	0.65330	0.66604	**0.66637**	0.66628
	3-FCN	0.67217	0.67438	0.67415	0.65023	0.66863	0.67370	**0.68394**	0.67145
	Reg-emb	0.72144	**0.72313**	0.72225	0.72100	0.25780	0.72127	0.72099	0.72190
	2-FCN-emb	0.69368	0.69317	0.69382	0.69399	0.67951	**0.69501**	0.69417	0.69308
	3-FCN-emb	0.68270	0.68211	0.68247	0.68332	0.67491	**0.68427**	0.68359	0.68361
	Attn	0.72540	**0.72485**	**0.72485**	**0.72426**	**0.71490**	**0.72365**	**0.72396**	**0.72644**

Table 3: AIQ scores

		Cost Predictor							
		Oracle R1	Reg	2-FCN	3-FCN	Reg-emb	2-FCN-emb	3-FCN-emb	Attn
Quality Predictors	Oracle R1	0.86430	0.86430	0.86430	0.86430	0.86430	0.86430	0.86430	0.86430
	Reg	0.77338	0.77338	0.77338	0.77338	0.77338	0.77338	0.77338	0.77338
	2-FCN	0.72036	0.72036	0.72036	0.72039	0.72000	0.72036	0.72036	0.72050
	3-FCN	0.73552	0.73562	0.73564	0.73504	0.73581	0.73575	0.76401	0.73601
	Reg-emb	0.78337	**0.78337**	**0.78337**	**0.78337**	**0.78337**	**0.78337**	**0.78337**	**0.78337**
	2-FCN-emb	0.76811	0.76799	0.76799	0.76799	0.76787	0.76794	0.76812	0.76812
	3-FCN-emb	0.76389	0.76402	0.76402	0.76389	0.76420	0.76375	0.76415	0.76389
	Attn	0.78082	**0.78094**	**0.78082**	**0.78091**	**0.78082**	**0.78082**	**0.78082**	**0.78082**

Table 4: Maximum performance achieved

H.2 Rewards: R_2

		Cost Predictor							
		Oracle R2	Reg	2-FCN	3-FCN	Reg-emb	2-FCN-emb	3-FCN-enb	Attn
Quality Predictors	Oracle R2	0.84275	0.84518	0.84361	0.84318	0.85961	0.85378	0.85307	0.85564
	Reg	0.72122	0.71833	0.72129	0.72146	0.70525	0.72133	0.72127	**0.71949**
	2-FCN	0.66427	0.66404	0.66444	0.66348	0.65120	0.66484	0.66517	**0.66545**
	3-FCN	0.66970	0.67162	0.67050	0.67037	0.66983	**0.67180**	0.66887	0.66857
	Reg-emb	0.72229	0.72244	**0.72274**	0.72249	0.70136	0.72258	0.72249	0.71932
	2-FCN-emb	0.69053	0.68990	0.69038	0.69093	0.67823	**0.69197**	0.69011	0.68990
	3-FCN-emb	0.68282	0.68314	0.68347	0.68429	0.67701	**0.68465**	0.68447	0.68388
	Attn-eval	0.72433	**0.72340**	**0.72430**	**0.72476**	**0.71189**	**0.72307**	**0.72328**	**0.72737**

Table 5: AIQ scores

		Cost Predictor							
		Oracle R2	Reg	2-FCN	3-FCN	Reg-emb	2-FCN-emb	3-FCN-enb	Attn
Quality Predictors	Oracle R2	0.86430	0.86430	0.86430	0.86430	0.86430	0.86430	0.86430	0.86430
	Reg	0.77338	0.77338	0.77338	0.77338	0.77338	0.77338	0.77338	0.77338
	2-FCN	0.72036	0.72036	0.72036	0.72022	0.72000	0.72036	0.72036	0.72050
	3-FCN	0.73526	0.73565	0.73526	0.73578	0.73594	0.73604	0.73530	0.73604
	Reg-emb	0.78337	**0.78337**	**0.78337**	**0.78337**	**0.78337**	**0.78337**	**0.78337**	**0.78337**
	2-FCN-emb	0.76824	0.76812	0.76812	0.76812	0.76800	0.76807	0.76825	0.76825
	3-FCN-emb	0.76389	0.76402	0.76401	0.76402	0.76425	0.76376	0.76415	0.76402
	Attn-eval	0.78082	**0.78082**	**0.78082**	**0.78082**	**0.78082**	**0.78082**	**0.78082**	**0.78082**

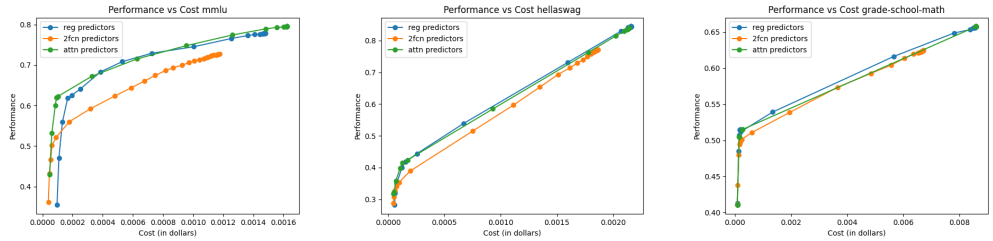
Table 6: Maximum performance achieved

I Domain-wise and Dataset-wise results

Figures 7–8 present cost-quality curves for each benchmark task and domain, including MMLU, HellaSwag, GSM8K, ARC Challenge, Winogrande, MBPP, and MT-Bench on LLM pool 1. Across most domains, Attention Router consistently matches or exceeds the performance of traditional predictors at lower cost, demonstrating robust generalization. For complex tasks such as ARC Challenge and MBPP, Attention Router achieves the highest performance at a fraction of the cost compared to baselines. In domains with high diversity (e.g., MMLU Professional Law, Moral Scenarios), our method maintains strong cost–quality trade-offs, validating its adaptability.

I.1 Dataset-wise results

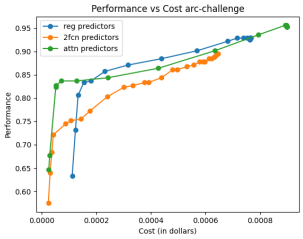
I.1.1 Rewards: R1



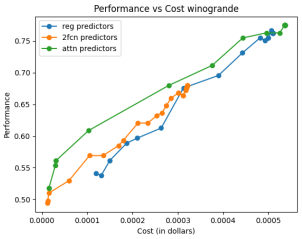
(a) MMLU

(b) Hellaswag

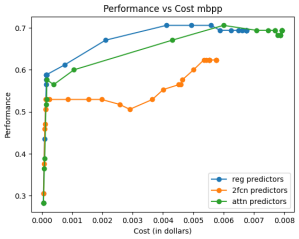
(c) GSM8K



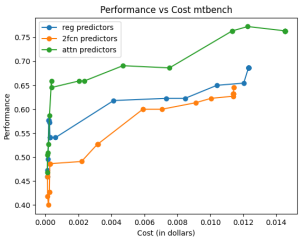
(d) Arc Challenge



(e) Winograde



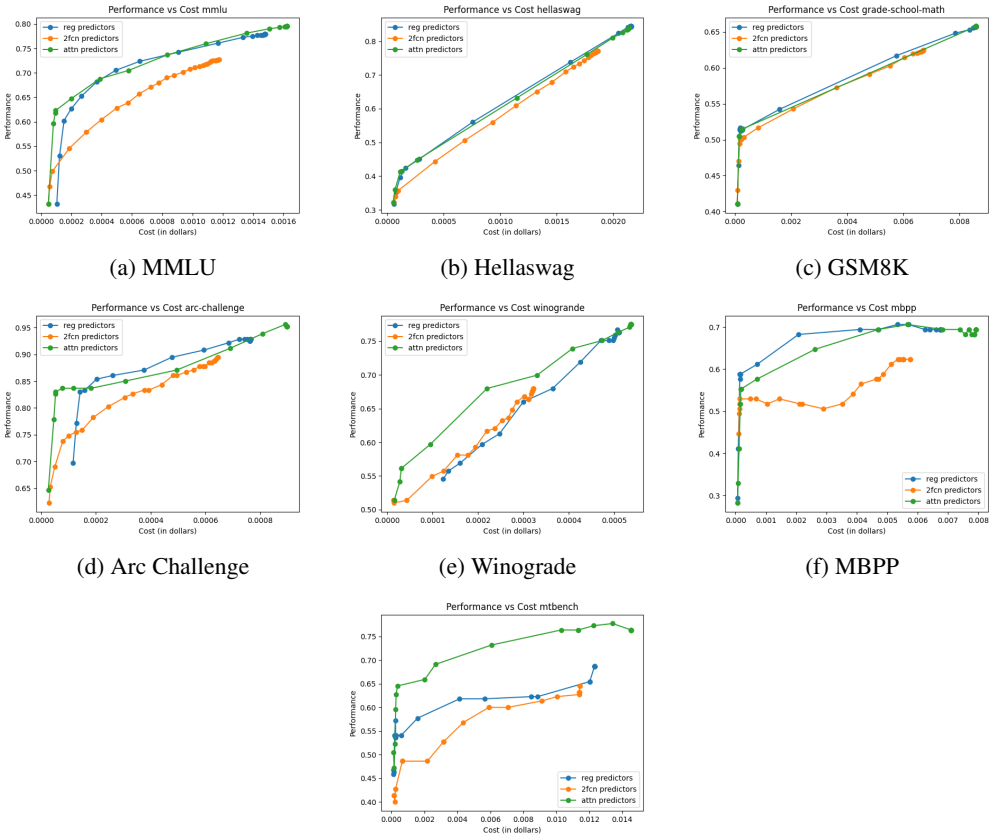
(f) MBPP



(g) MT Bench

Figure 7: Dataset-wise results of the predictor-based routers using $R1 = s(q, r) - \frac{1}{\lambda} c(q, r)$ rewards

I.1.2 Rewards: R2



(a) MMLU (b) Hellaswag (c) GSM8K

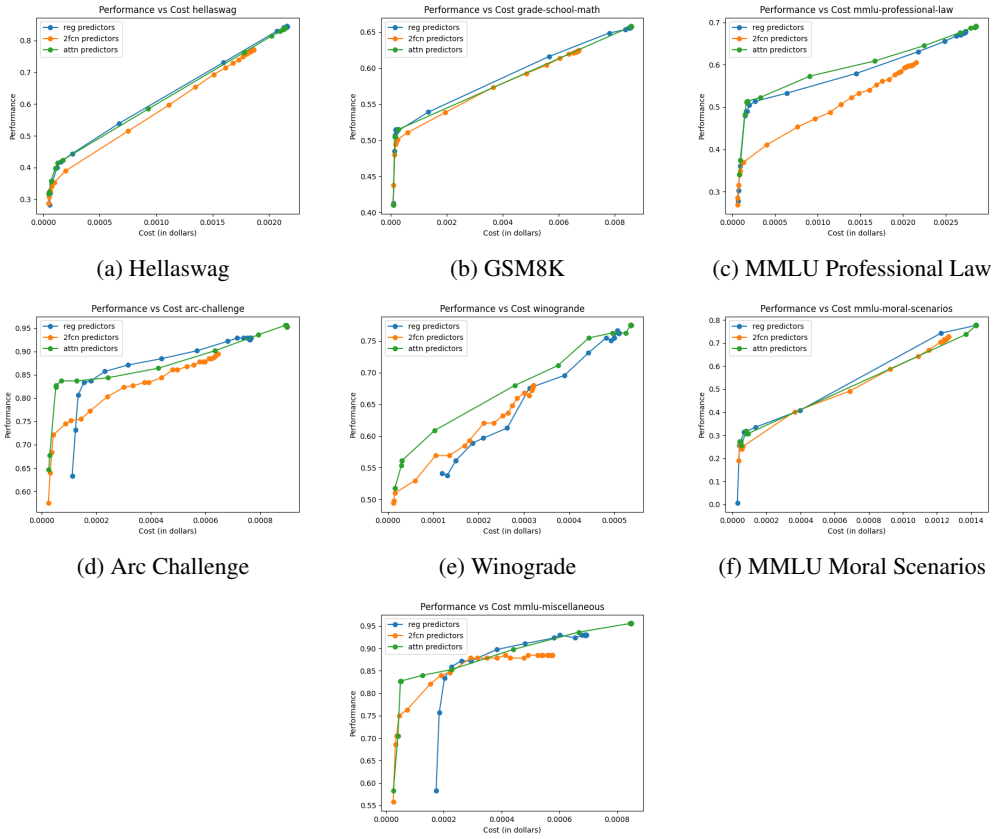
(d) Arc Challenge (e) Winograde (f) MBPP

(g) MT Bench

Figure 8: Dataset-wise results of the predictor-based routers using $R2 = s(q,r)\exp\left(-\frac{1}{\lambda}c(q,r)\right)$ rewards

I.2 Domain-wise results

I.2.1 Rewards: R1



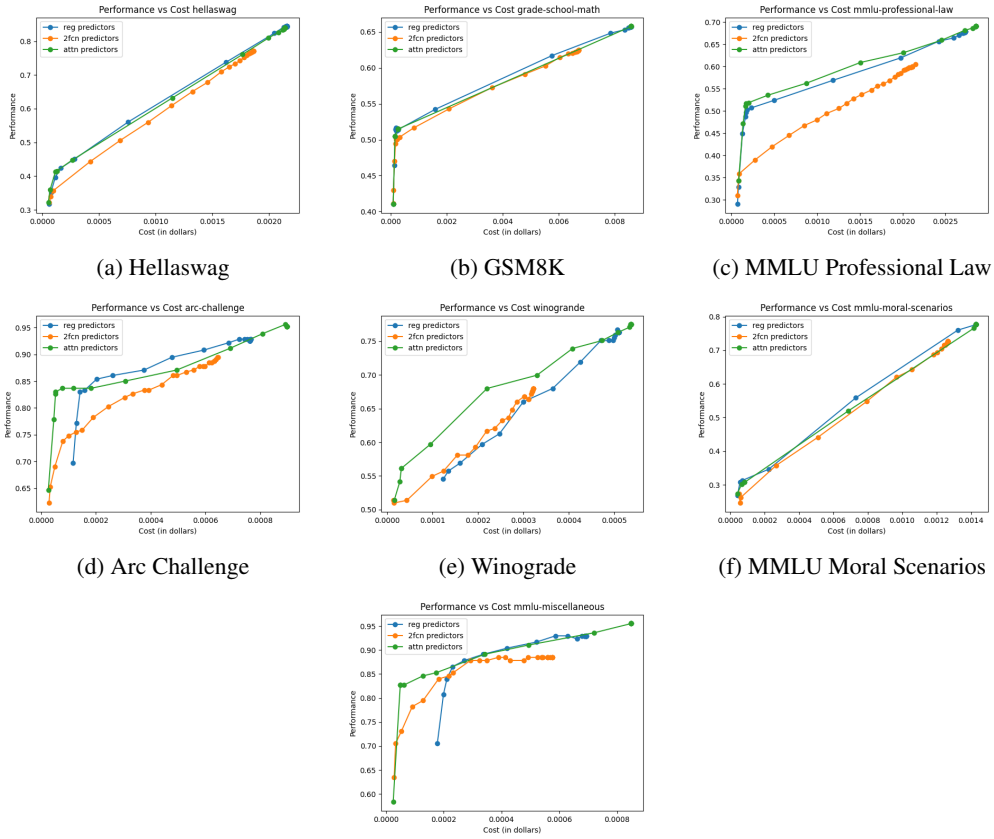
(a) Hellaswag (b) GSM8K (c) MMLU Professional Law

(d) Arc Challenge (e) Winogrande (f) MMLU Moral Scenarios

(g) MMLU Miscellaneous

Figure 9: Domain-wise results of the predictor-based routers using $R1 = s(q, r) - \frac{1}{\lambda} c(q, r)$ rewards

I.2.2 Rewards: R2



(a) Hellaswag (b) GSM8K (c) MMLU Professional Law

(d) Arc Challenge (e) Winogrande (f) MMLU Moral Scenarios

(g) MMLU Miscellaneous

Figure 10: Domain-wise results of the predictor-based routers using $R2 = s(q,r)\exp\left(-\frac{1}{\lambda}c(q,r)\right)$ rewards

J Compute Resources

We implemented our proposed method in PyTorch and conducted experiments on either a single NVIDIA A40 or a single NVIDIA A100 Tensor Core GPU. Our models consume at most 1GB memory. While the training period of predictors for the framework is upto 30 minutes, inference time for predictors is around 5-10 minutes, varying with batch size, trainset size and the architecture.

K Limitations

Our work does not include direct experimental comparisons with recent multi-LLM routing methods on RouterBench, such as Universal Model Routing. Future work should benchmark our approach against these contemporary baselines for a more robust evaluation. Moreover, our experiments are conducted on a static dataset with fixed LLM pool and tasks from RouterBench. There is a degree of uncertainty with the same LLM responses for the same query, so there is a scope of making it dynamic, and modeling performance and inference cost with a degree of uncertainty. Finally, the method relies on LLM representations, and results vary based on the quality and quantity of the data.