Leveraging Uncertainty Estimation for Efficient LLM Routing

Tuo Zhang [*1] **Asal Mehradfar** [*1] **Dimitrios Dimitriadis** [2] **Salman Avestimehr** [1]

Abstract

Deploying large language models (LLMs) in edge-cloud environments requires an efficient routing strategy to balance cost and response quality. Traditional approaches prioritize either human-preference data or accuracy metrics from benchmark datasets as routing criteria, but these methods suffer from rigidity and subjectivity. Moreover, existing routing frameworks primarily focus on accuracy and cost, neglecting response quality from a human preference perspective. In this work, we propose the Confidence-Driven LLM Router, a novel framework that leverages uncertainty estimation to optimize routing decisions. To comprehensively assess routing performance, we evaluate both system cost efficiency and response quality. In particular, we introduce the novel use of LLM-as-a-Judge to simulate human rating preferences, providing the first systematic assessment of response quality across different routing strategies. Extensive experiments on MT-Bench, GSM8K, and MMLU demonstrate that our approach outperforms state-of-the-art routing methods, achieving superior response quality while maintaining cost efficiency.

1. Introduction

The deployment of AI models on edge devices is increasingly following a hybrid approach, where small language models (SLMs) run on-device (e.g., smartphones and IoT devices) while larger, more powerful models remain in the cloud (Apple, 2024). This setup provides a balance between efficiency and performance, allowing low-latency responses for simple queries while reserving cloud-based LLMs for more complex tasks. However, determining when to offload queries to the cloud is a crucial challenge: calling the cloud unnecessarily increases cost and latency, whereas over-relying on local SLMs risks suboptimal response quality. An effective routing strategy is essential to dynamically balance cost and performance.

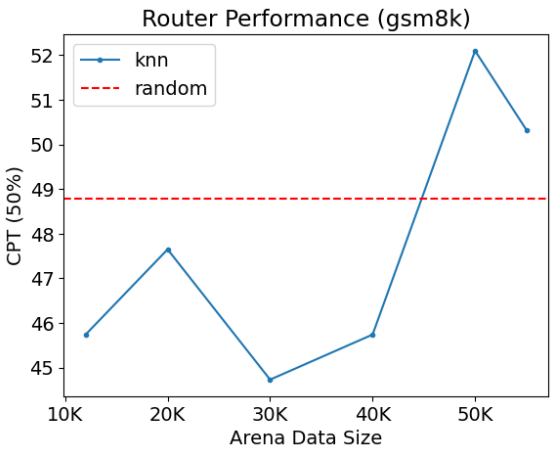

Figure 1. Performance of human preference-based router with varying training sample sizes. Routing efficiency even becomes worse as the number of training samples increases, indicating that additional data does not necessarily improve performance.

Traditional cascading routers, which sequentially query models until a satisfactory response is obtained (Chen et al., 2023), are inefficient for edge-cloud settings due to latency and redundant model calls. Recent predictive routing approaches aim to preemptively select the best model for a given query, with two leading solutions: TO-Router (Stripelis et al., 2024), which trains router on accuracy-based benchmarks and RouteLLM (Ong et al., 2024), which relies on human preference selection.

While human preference data and benchmark accuracy are commonly used as performance indicators in router training, our results reveal two major limitations that hinder data efficiency and system utilization. First, ***human judgment is not always reliable.*** User ratings are subjective and inconsistent, often failing to provide an accurate ranking of model performance. Also, collecting human-preference data is resource-intensive, requiring manual evaluation of each sample on a case-by-case basis. These issues are particularly evident in the arena dataset (Chiang et al., 2024), where the distribution of preference data across models is sparse

[*]Equal contribution [1]University of Southern California [2]Amazon. Correspondence to: Tuo Zhang <tuozhang@usc.edu>.

Accepted at the ICML 2025 Workshop on Collaborative and Federated Agentic Workflows (CFAgentic@ICML'25), Vancouver, Canada. July 19, 2025.

and uneven, complicating router training. Empirically, we demonstrate that arena data does not follow a scaling law to validate our argument. As shown in Figure 1, increasing the dataset size does not necessarily improve routing performance. Instead, adding more data can introduce noise and inconsistencies, potentially degrading routing accuracy rather than enhancing it.

Second, ***accuracy is an incomplete indicator.*** Using accuracy as a performance indicator can miss essential nuances in model responses. For the same query, two models may provide correct answers labeled as a "tie" based solely on accuracy. However, one answer may be superior; for example, Response B is more precise and confident than Response A, even though both are technically correct. Additionally, there are cases where neither model provides an entirely correct answer, yet one response is closer to the desired output. Because accuracy-based routers rely strictly on binary correctness labels, they fail to capture these qualitative differences in response quality. In practice, we observe that models often either both perform well or both fail on a query. Consequently, routers trained on accuracy alone struggle to reliably distinguish between models when one significantly outperforms the other in non-binary ways.

To address the limitations of state-of-the-art methods, we propose **the Confidence-Driven LLM Router System**, which leverages Semantic Entropy (SE) as an uncertainty measure to guide routing decisions. Instead of relying on human preferences or accuracy-based thresholds, our system uses semantic entropy to measure model confidence. This enables the router to offload queries to cloud-based LLMs when higher certainty is needed, which keeps confident responses on-device to minimize cost. As the motivational example shown in Figure 2, by adopting uncertainty as a routing signal, our approach dynamically optimizes response quality and computational efficiency, making it better suited for real-world edge-cloud deployments compared to the state-of-the-art solutions, such as TO-Router and RouteLLM.

To validate our approach, we conduct comprehensive comparisons with state-of-the-art routing methods across diverse benchmark datasets, including MT-Bench (Zheng et al., 2023), GSM8K (Cobbe et al., 2021), and MMLU (Hendrycks et al., 2020). These evaluations demonstrate that the proposed Confidence-Driven LLM Router achieves superior trade-offs between response quality and cost efficiency. Furthermore, we introduce LLM-as-a-Judge as an evaluation protocol that simulates human preferences, offering a more human-centric assessment of response quality beyond traditional accuracy-based metrics. Our method consistently outperforms strong baselines such as RouteLLM and TO-Router, both in system efficiency and human-aligned response quality.

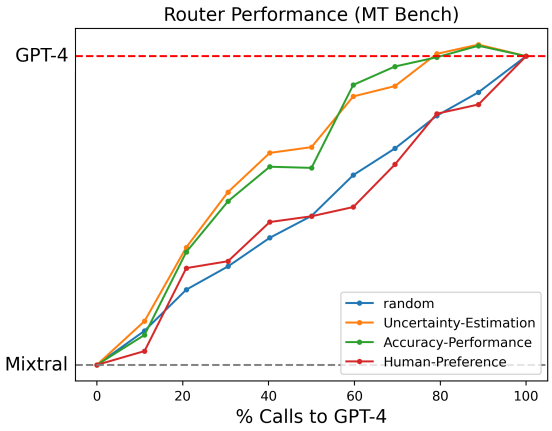

Figure 2. Routing performance/cost trade-off between strong model (GPT-4) and weak model (Mixtral-8x7B). All routers shown, except the random router, use the same kNN-based model architecture.

2. Related Works

Existing LLM routing methods could be categorized into two groups based on the type of data used to train the router: those based on accuracy labels and those based on human preference data. The first group relies on accuracy-based supervision, using correctness from benchmark datasets as routing signals. TO-Router (Stripelis et al., 2024) adopts a predictive router trained on model accuracy to select the most capable model per query. RouterDC (Chen et al., 2024) introduces a dual-contrastive loss to enhance query-model alignment under the same accuracy labels. GraphRouter (Feng et al., 2024) models the routing process as a heterogeneous graph and predicts routing decisions via edge classification. While these methods differ architecturally, they all operate under the same assumption: routing data is derived from benchmark accuracy. None of them revisits the supervisory signal used to define routing quality.

The second group uses human preference data. For instance, RouteLLM (Ong et al., 2024) trains the router on pairwise comparisons from human annotators, typically collected from chatbot arenas. While this improves alignment with user preferences, it introduces subjectivity and noise, and the scalability of human annotation remains limited. In contrast to prior works, we introduce a new training signal derived from model uncertainty, which provides a more informative and scalable alternative to accuracy or preference-based supervision.

To the best of our knowledge, Hybrid LLM (Ding et al.) is the only previous method that explicitly incorporates uncertainty into routing. However, it does not adopt standard uncertainty estimation techniques (Bakman et al., 2024; Yaldiz et al., 2024). Instead, it defines uncertainty as the response

quality gap between a strong and a weak model, making the signal model pair and prompt dependent, rather than an intrinsic model-specific confidence measure. In contrast, our method uses semantic entropy, providing a principled, model-aware, and query-specific uncertainty score that is directly usable for scalable, modular router training. Hybrid LLM also replaces hard routing labels with BART-based soft scores, which are sensitive to fluency and style rather than grounded semantic correctness. Our method relies on clustering and bidirectional entailment, yielding a more reliable uncertainty signal aligned with semantic fidelity.

3. Method

3.1. Overview

Inspired by prior work (Kuhn et al., 2023; Bakman et al., 2024), we quantify uncertainty in natural language generation based on semantic content rather than token-level variations, using it as a performance indicator for training the router.

SE (Kuhn et al., 2023) captures uncertainty by clustering generated outputs with equivalent meanings and computing entropy over their aggregated probabilities. Unlike traditional entropy, which treats all token sequences distinctly, SE accounts for semantic equivalence, ensuring paraphrases contribute to the same uncertainty estimate. A lower SE score indicates higher confidence, while a higher score signals greater uncertainty.

Formally, we define the probability of a meaning cluster c given an input prompt $\mathbf{x}$ as:

$$p(c|\mathbf{x}) = \sum_{s,\mathbf{x}\in c} p(s|\mathbf{x}). \tag{1}$$

The semantic entropy of $\mathbf{x}$ is then computed as:

$$SE(\mathbf{x}) = -\frac{1}{|C|}\sum_{i=1}^{|C|} \log p(C_i|\mathbf{x}), \tag{2}$$

where C represents the set of all clusters.

3.2. System Design of the Confidence-Driven LLM Router

The training and deployment of the Confidence-Driven LLM Router consist of three key phases:

Phase 1: Router Data Preparation. To create a training dataset that reflects real deployment scenarios, we select factual-related datasets, such as Natural QA and Trivia QA, to probe model confidence and knowledge capabilities. Additionally, domain-specific instruction datasets can be incorporated to tailor the router to specialized applications. Clustering generated outputs is a critical preprocessing step before computing SE. In this work, we adopt a bidirectional entailment mechanism following the previous work (Kuhn et al., 2023). The first generated response initializes a cluster. For each subsequent response, a semantic entailment classifier, fine-tuned on DeBERTa-large model (He et al., 2020), evaluates bidirectional entailment between the response and the current cluster representative. If both forward and backward entailments hold, the response is assigned to the existing cluster; otherwise, a new cluster is formed. This bidirectional criterion ensures that only semantically equivalent responses are grouped, allowing the number of clusters to be dynamically determined based on meaning variations among generated outputs.

Phase 2: Constructing Preference Data from Semantic Entropy Scores. In the second phase, we create preference data by comparing the SE scores across models for each unique prompt. Although no two SE values are exactly the same, some prompts yield similar performance between models, which we denote as a "tie" case. To identify ties, we treat cases with close uncertainty levels as equivalent, where neither model is a clear winner. To quantify the relative difference in uncertainty, we compute the normalized SE difference between the two models as:

$$\delta_{\text{SE}}(\mathbf{x}) = \left| \frac{SE_{\text{strong}}(\mathbf{x}) - SE_{\text{weak}}(\mathbf{x})}{SE_{\text{strong}}(\mathbf{x})} \right| \tag{3}$$

$SE_{\text{strong}}(\mathbf{x})$ represents the high-cost model, and $SE_{\text{weak}}(\mathbf{x})$ represents the low-cost model. Using this metric, we determine the preferred model as:

$$\text{Winner} = \begin{cases} \arg\min_M SE(M, \mathbf{x}) & \text{if } \delta_{\text{SE}}(\mathbf{x}) > \tau, \\ \text{Tie} & \text{otherwise.} \end{cases} \tag{4}$$

The predefined threshold τ controls sensitivity to uncertainty differences. If $\delta_{\text{SE}}(\mathbf{x})$ exceeds the predefined τ, the uncertainties are considered sufficiently distinct, and the model with the lower semantic entropy is designated as the preferred model. Otherwise, we consider both models equal in performance for the given prompt.

A higher τ enforces stricter distinctions, aligning the preference data more closely with traditional accuracy-based evaluations. Conversely, a lower τ increases sensitivity to subtle linguistic variations in model responses. By tuning τ, we balance robustness with linguistic granularity.

Phase 3: Training the Confidence-Driven Router. After generating the SE-based preference data in Phase 2, we format each training sample as follows: {id, model a, model b, prompt, response a, response b, winner model a, winner model b, winner tie}. In the last three columns, we use binary values (0 or 1) to represent the routing outcomes. For instance, if model_a is the preferred model, then `winner_model_a` is set to 1, while `winner_model_b`

and `winner_tie` are set to 0. The dataset used in this study is now publicly available on Huggingface [1]. Once the dataset is prepared, we transform the instruction records into vectorized representations using the pre-trained embedding model, which serves as inputs for training the router classifiers.

4. Evaluation

4.1. Experimental Methodology

4.1.1. Tasks and Datasets

We use GPT-4 as the strong model and Mixtral-8x7B as the weaker model to ensure consistency and fair comparison with previous works (Ong et al., 2024). We also provide an additional experiment that takes GPT-4 and Qwen2-1.5B-Instruct (Yang et al., 2024) as a model pair to simulate a realistic edge-cloud deployment scenario.

The Confidence-Driven Router is trained with a combination of Natural QA (Kwiatkowski et al., 2019; Lee et al., 2019), Trivia QA (Joshi et al., 2017), PopQA (Mallen et al., 2022), and MAWPS (Koncel-Kedziorski et al., 2016) datasets to ensure a knowledge space. We choose them to reflect the types of queries that appear in real-world, open-ended generation settings. We want to use this combination to compare the human preference data that RouteLLM collected for their training. We randomly selected 3,610 samples from each QA dataset and 1,418 samples from the MAWPS dataset, resulting in 12,247 samples, matching the quantity as Chatbot Arena dataset (Chiang et al., 2024) for RouteLLM training.

To comprehensively evaluate the routing systems, we select a diverse set of benchmark datasets: the MMLU (Hendrycks et al., 2020) dataset, consisting of 14,042 questions across 57 subjects; the MT-Bench dataset (Zheng et al., 2023), which includes 160 open-ended questions assessed using the LLM-as-a-judge approach; and the GSM8K dataset (Cobbe et al., 2021), containing over 1,000 grade-school math problems. These datasets provide a broad evaluation across varied question types and subject domains. More details related to the datasets are listed in the Appendix A.1.

4.1.2. Routing Architectures

We select four different predictive routing methods in our evaluation. To match the hardware constraints on edge computing, we purposely select the lightweight routing architectures in our experiments. Now, we describe our approach for learning the win prediction model $P(\text{win}_{\mathcal{M}_{\text{strong}}} \mid q)$. We represent a sample from our dataset $\mathcal{D}$ as (q, M_w, M_l), where M_w and M_l denote the winning and losing models, respectively.

[1]The training dataset would be available when the paper is public.

Similarity-Weighted (SW) Ranking. Same as RouteLLM (Ong et al., 2024), we adopt a Bradley-Terry (BT) model (Bradley & Terry, 1952) for this routing task. Given an input query q, we compute a weight ω_i for each query q_i in the training set based on its similarity to q, as follows:

$$\omega_i = \gamma^{1+S(q,q_i)}, \tag{5}$$

where γ is a scaling factor which is 10 in our case, and $S(q, q_i)$ represents the similarity between queries q and q_i, defined as:

$$S(q, q_i) = \frac{\epsilon \cdot \epsilon_i}{\|\epsilon\|\|\epsilon_i\| \cdot \max_{1\le s\le |D|}\left(\frac{\epsilon_i \cdot \epsilon_s}{\|\epsilon_i\|\|\epsilon_s\|}\right)}, \tag{6}$$

with ϵ denoting a query embedding. The BT coefficients ξ are then learned by solving:

$$\arg\min_{\xi} \sum_{i=1}^{|D|} \omega_i \cdot \ell\left(l_i, \frac{1}{1+e^{\xi_{w_i}-\xi_{l_i}}}\right), \tag{7}$$

where ℓ is the binary cross-entropy loss.

The learned BT coefficients allow us to estimate the win probability given query q as:

$$P(\text{win}_{M_w} \mid q) = \frac{1}{1+e^{\xi_w-\xi_l}}. \tag{8}$$

This routing model does not require additional training, and the optimization is performed at inference time.

Matrix Factorization. Inspired by the RouteLLM approach (Ong et al., 2024; Koren et al., 2009), we leverage matrix factorization as one of our routing models. The objective is to uncover a latent scoring function $s : \mathcal{M} \times \mathcal{Q} \to \mathbb{R}$ that assesses the quality of the model M_w's response to a given query q. Specifically, if model M_w performs better than M_l on query q, then $s(M_w, q) > s(M_l, q)$. We encode this preference by modeling the win probability using a Bradley-Terry (BT) relationship (Bradley & Terry, 1952):

$$P(\text{win}_{M_w} \mid q) = \sigma(s(M_w, q) - s(M_l, q)), \tag{9}$$

where σ is the sigmoid function, and s is a bilinear function of the model and query embeddings. This approach effectively performs matrix factorization over the score matrix on the set $\mathcal{Q} \times \mathcal{M}$.

Multilayer Perceptron (MLP). For the MLP routing, we utilize a 2-layer multilayer perceptron (MLP) architecture. The output y_k for the MLP-Router is given by:

$$P(\text{win}_{M_w} \mid q) = \varphi\left(\sum_{j=1}^{m} w_{jk}^{(2)}\,\text{ReLU}\Big(\sum_{i=1}^{n} w_{ij}^{(1)}\epsilon_i + b_j^{(1)}\Big) + b_k^{(2)}\right) \tag{10}$$

where φ represents the softmax activation function in the output layer and ϵ denoting a query embedding.

Table 1. System Performance Comparison of Routing Systems on test datasets with GPT-4 and Mixtral-8x7B as model pair. A low CPT value indicates a cost-effective routing strategy. **Bold** highlight the best performance, and underlined denote the second-best.

		MT-Bench		GSM8K		MMLU	
Routing	**Method**	**CPT(50%)**	**CPT(80%)**	**CPT(50%)**	**CPT(80%)**	**CPT(50%)**	**CPT(80%)**
	Random	51.29	78.55	48.79	80.16	50.04	79.32
TO-Router	kNN	59.72	90.39	47.93	79.03	43.73	77.74
	MLP	49.15	86.67	51.03	77.77	44.01	77.43
RouteLLM	SW	56.08	78.37	46.03	79.58	47.41	74.23
	MF	55.59	84.12	49.07	80.09	58.55	83.68
Confidence-Driven LLM Router	SW	**27.31**	**55.61**	48.03	80.41	**37.96**	**73.85**
	MF	42.94	63.53	**41.89**	**75.34**	50.06	78.38
	kNN	60.84	81.50	44.08	76.32	42.70	75.28
	MLP	35.54	74.92	50.04	79.79	57.07	83.27

k-Nearest Neighbors (kNN). The k-Nearest Neighbors router represents all training queries q_i with an embedding ϵ_i. For each test query q, with embedding ϵ, we identify the closest training query q' by finding the query in the training set with the highest cosine similarity to ϵ:

$$i = \arg\min_{1 \le i \le |D|} \left(\frac{\epsilon_i \cdot \epsilon}{\|\epsilon_i\| \|\epsilon\|} \right). \tag{11}$$

$$q' = q_i \tag{12}$$

After identifying the nearest query q', the router decides on the winner model based on the performance of the winner model associated with q'. This method leverages the similarity between the test query and training queries to select the most suitable expert dynamically.

4.1.3. Baseline Selection

We select RouteLLM (Ong et al., 2024) and TO-Router (Stripelis et al., 2024), two state-of-the-art predictive routing systems. Following their original configurations, we evaluate TO-Router using kNN and MLP architectures, while RouteLLM is assessed using SW ranking and MF models. We also include a random router without any training as a baseline for comparison.

4.1.4. Evaluation Criteria

We evaluate performance based on two key criteria: system cost and response quality.

To evaluate system cost, we adopt the Call-Performance Threshold (CPT) metric from prior work (Ong et al., 2024). CPT(x%) represents the minimum fraction of queries that must be routed to the stronger model to achieve an x% improvement over the baseline accuracy of the weaker model. For example, CPT(50%) specifies the proportion of calls required to the strong model to attain a 50% improvement over the weak model's baseline accuracy. A lower CPT value indicates a more cost-effective routing strategy, achieving performance gains with fewer calls to the stronger model.

To evaluate response quality, we use LLM-as-a-Judge to simulate human ratings. We employ an independent LLM (i.e., GPT-o1) to choose the most preferable responses from the routed model alongside ground-truth answers. The judge is instructed to select based on correctness and precision of reasoning, as the detailed system prompt listed in the Appendix A.2. We design the score as $\text{Score}(i) = \left(\frac{S_i}{T}\right) \times 100$, where S_i be the number of times router i is selected, and T be the total number of queries. Unlike traditional accuracy-based evaluations using golden labels, this approach simulates human judgment by considering not only the correctness but also the interpretability and coherence of model outputs, better aligning with human preference selection rather than relying solely on objective correctness.

4.2. Models and Test Environment

We implemented the experiments using PyTorch (Paszke et al., 2019), and conducted our experiments on two NVIDIA A100 GPUs. For GPT-4 model, we use `gpt-4-0613` API. For GPT-o1 model, we use `o1-2024-12-17` API.

4.3. Evaluation on System Costs

We first evaluate the system costs of various routing strategies on the test datasets, as summarized in Table 1. The Confidence-Driven LLM Router consistently achieves strong performance in CPT(50%) and CPT(80%) across datasets. To provide a clearer comparison, we report the actual OpenAI API cost (in USD) for each routing system on the MT-Bench dataset under CPT(80%): Random router costs $4.06; TO-Router costs $3.88; RouteLLM costs $4.04; and our proposed method costs lowest with $3.74. We also provide an additional experiment that takes GPT-4 and Qwen2-1.5B-Instruct as a model pair. As shown in Table 2,

Table 2. System Performance Comparison of Routing Systems on test datasets with GPT-4 and Qwen2-1.5B-Instruct as model pair.

		GSM8K	
Routing	**Method**	**CPT(50%)**	**CPT(80%)**
	Random	48.79	80.16
TO-Router	kNN	49.44	79.98
	MLP	51.75	80.80
RouteLLM	SW	<u>48.74</u>	79.86
	MF	51.13	80.09
Confidence-Driven LLM Router	SW	49.17	79.08
	MF	**48.57**	**78.87**
	kNN	48.82	<u>78.54</u>
	MLP	52.52	81.36

Table 3. Response quality comparison of routing systems on the GSM8K dataset based on LLM-as-a-Judge. Higher LLM-judge scores reflect better response quality.

	TO-Router	RouteLLM	Confidence Router
CPT(50%)	78.88	79.72	**79.95**
CPT(80%)	85.97	88.88	**89.21**

with the new model pairs, our proposed router still outperforms other baselines in CPT, which validates our results and arguments in our paper.

Compared to the proposed Confidence-Driven LLM Router, TO-Router and RouteLLM show less consistent and often suboptimal performance across the datasets. While TO-Router achieves reasonable results on GSM8K, it struggles to maintain comparable performance on MT-Bench and MMLU. Similarly, RouteLLM demonstrates competitive results in isolated metrics but fails to match the overall effectiveness of the Confidence-Driven LLM Router. These inconsistencies suggest that traditional routing indicators, such as accuracy-based metrics or human preference data, lack the adaptability provided by uncertainty-based routing, which enables the proposed system to deliver superior end-to-end performance across varied tasks and domains.

4.4. Evaluation on Response Quality

To better understand the advantage of our proposed method, we evaluate the response quality of each routing system under the same accuracy with GSM8K dataset. We select the best routing models under each routing system in Table 1. As summarized in Table 3, the proposed method achieves the highest LLM-as-a-Judge rating, indicating that its responses are the most human-preferable among all baselines. Our uncertainty-aware training strategy optimizes routing decisions based on a direct measure of model confidence. By minimizing uncertainty in routing, our approach ensures that queries are directed to models that can generate the most confident and reliable responses, leading to outputs that better align with human preferences.

5. Conclusion

In this paper, we introduced the Confidence-Driven LLM Router, a novel framework that leverages uncertainty estimation to optimize LLM deployment in edge-cloud environments. By using semantic entropy as a performance indicator, our approach addresses the limitations of existing methods, such as the subjectivity of human preference data and the rigidity of accuracy-based metrics. Extensive experiments on benchmark datasets demonstrate that our method outperforms state-of-the-art routing systems, achieving a better trade-off between response quality and efficiency. Future work will explore multi-modal query integration and further latency reduction in distributed systems.

6. Limitations

We have two major limitations which we aim to address in future works. First, our evaluation is limited to text-based queries, and we do not extend our analysis to multi-modal routing scenarios. In real-world applications, queries may include image-text pairs or other modalities, especially in Vision-Language Models (VLMs). Future work should investigate how uncertainty estimation-based routing generalizes to multi-modal inputs and whether SE remains an effective performance indicator in VLM settings.

Second, we do not analyze the computational overhead of different routing architectures. Our study primarily evaluates routing effectiveness, but in practice, the choice of router architecture can significantly impact system efficiency. Future work should explore the trade-offs between neural network-based routers and statistical methods, assessing their cost, scalability, and real-time deployment feasibility in edge-cloud environments.

7. Impact Statement

This paper advances the field of machine learning by introducing a novel uncertainty-aware routing framework for efficient deployment of LLMs in edge-cloud environments. By enabling cost-effective and confidence-aware routing, our approach supports the development of intelligent systems that can dynamically distribute tasks between local and remote agents, improving responsiveness, adaptability, and interpretability in real-world applications such as mobile assistants, IoT systems, and autonomous platforms. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Apple. Apple intelligence foundation language models. *ArXiv*, abs/2407.21075, 2024. URL https://api.semanticscholar.org/CorpusID:271546289.

Bakman, Y. F., Yaldiz, D. N., Buyukates, B., Tao, C., Dimitriadis, D., and Avestimehr, A. S. Mars: Meaning-aware response scoring for uncertainty estimation in generative llms. *ArXiv*, abs/2402.11756, 2024. URL https://api.semanticscholar.org/CorpusID:267750426.

Bradley, R. A. and Terry, M. E. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39:324, 1952. URL https://api.semanticscholar.org/CorpusID:125209808.

Chen, L., Zaharia, M. A., and Zou, J. Y. Frugalgpt: How to use large language models while reducing cost and improving performance. *ArXiv*, abs/2305.05176, 2023. URL https://api.semanticscholar.org/CorpusID:258564349.

Chen, S., Jiang, W., Lin, B., Kwok, J., and Zhang, Y. Routerdc: Query-based router by dual contrastive learning for assembling large language models. *Advances in Neural Information Processing Systems*, 37:66305–66328, 2024.

Chiang, W.-L., Zheng, L., Sheng, Y., Angelopoulos, A. N., Li, T., Li, D., Zhang, H., Zhu, B., Jordan, M., Gonzalez, J. E., and Stoica, I. Chatbot arena: An open platform for evaluating llms by human preference, 2024.

Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun, H., Kaiser, L., Plappert, M., Tworek, J., Hilton, J., Nakano, R., Hesse, C., and Schulman, J. Training verifiers to solve math word problems. *ArXiv*, abs/2110.14168, 2021. URL https://api.semanticscholar.org/CorpusID:239998651.

Ding, D., Mallick, A., Wang, C., Sim, R., Mukherjee, S., Rühle, V., Lakshmanan, L. V., and Awadallah, A. H. Hybrid llm: Cost-efficient and quality-aware query routing. In *The Twelfth International Conference on Learning Representations*.

Feng, T., Shen, Y., and You, J. Graphrouter: A graph-based router for llm selections. *arXiv preprint arXiv:2410.03834*, 2024.

He, P., Liu, X., Gao, J., and Chen, W. Deberta: Decoding-enhanced bert with disentangled attention. *ArXiv*, abs/2006.03654, 2020. URL https://api.semanticscholar.org/CorpusID:219531210.

Hendrycks, D., Burns, C., Basart, S., Zou, A., Mazeika, M., Song, D. X., and Steinhardt, J. Measuring massive multitask language understanding. *ArXiv*, abs/2009.03300, 2020. URL https://api.semanticscholar.org/CorpusID:221516475.

Joshi, M., Choi, E., Weld, D., and Zettlemoyer, L. triviaqa: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension. *arXiv e-prints*, art. arXiv:1705.03551, 2017.

Koncel-Kedziorski, R., Roy, S., Amini, A., Kushman, N., and Hajishirzi, H. Mawps: A math word problem repository. In *North American Chapter of the Association for Computational Linguistics*, 2016. URL https://api.semanticscholar.org/CorpusID:2228719.

Koren, Y., Bell, R., and Volinsky, C. Matrix factorization techniques for recommender systems. *Computer*, 42(8): 30–37, 2009. doi: 10.1109/MC.2009.263.

Kuhn, L., Gal, Y., and Farquhar, S. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=VD-AYtP0dve.

Kwiatkowski, T., Palomaki, J., Redfield, O., Collins, M., Parikh, A., Alberti, C., Epstein, D., Polosukhin, I., Devlin, J., Lee, K., Toutanova, K., Jones, L., Kelcey, M., Chang, M.-W., Dai, A. M., Uszkoreit, J., Le, Q., and Petrov, S. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466, 2019. doi: 10.1162/tacl_a_00276. URL https://doi.org/10.1162/tacl_a_00276.

Lee, K., Chang, M.-W., and Toutanova, K. Latent retrieval for weakly supervised open domain question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 6086–6096, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1612. URL https://www.aclweb.org/anthology/P19-1612.

Mallen, A., Asai, A., Zhong, V., Das, R., Hajishirzi, H., and Khashabi, D. When not to trust language models: Investigating effectiveness and limitations of parametric and non-parametric memories. *arXiv preprint*, 2022.

Ong, I., Almahairi, A., Wu, V., Chiang, W.-L., Wu, T., Gonzalez, J. E., Kadous, M. W., and Stoica, I. Routellm: Learning to route llms with preference data. *ArXiv*, abs/2406.18665, 2024. URL https://api.semanticscholar.org/CorpusID:270764307.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmasion, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. Pytorch: An imperative style, high performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pp. 8024–8035, 2019.

Stripelis, D., Hu, Z., Zhang, J., Xu, Z., Shah, A., Jin, H., Yao, Y., Avestimehr, S., and He, C. Tensoropera router: A multi-model router for efficient llm inference. In *Conference on Empirical Methods in Natural Language Processing*, 2024. URL https://api.semanticscholar.org/CorpusID:271923913.

Yaldiz, D. N., Bakman, Y. F., Buyukates, B., Tao, C., Ramakrishna, A., Dimitriadis, D., Zhao, J., and Avestimehr, S. Do not design, learn: A trainable scoring function for uncertainty estimation in generative llms. *arXiv preprint arXiv:2406.11278*, 2024.

Yang, A., Yang, B., Hui, B., Zheng, B., Yu, B., Zhou, C., Li, C., Li, C., Liu, D., Huang, F., Dong, G., Wei, H., Lin, H., Tang, J., Wang, J., Yang, J., Tu, J., Zhang, J., Ma, J., Xu, J., Zhou, J., Bai, J., He, J., Lin, J., Dang, K., Lu, K., Chen, K., Yang, K., Li, M., Xue, M., Ni, N., Zhang, P., Wang, P., Peng, R., Men, R., Gao, R., Lin, R., Wang, S., Bai, S., Tan, S., Zhu, T., Li, T., Liu, T., Ge, W., Deng, X., Zhou, X., Ren, X., Zhang, X., Wei, X., Ren, X., Fan, Y., Yao, Y., Zhang, Y., Wan, Y., Chu, Y., Liu, Y., Cui, Z., Zhang, Z., and Fan, Z. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2024.

Zheng, L., Chiang, W.-L., Sheng, Y., Zhuang, S., Wu, Z., Zhuang, Y., Lin, Z., Li, Z., Li, D., Xing, E. P., Zhang, H., Gonzalez, J. E., and Stoica, I. Judging llm-as-a-judge with mt-bench and chatbot arena. *ArXiv*, abs/2306.05685, 2023. URL https://api.semanticscholar.org/CorpusID:259129398.

A. Appendix

A.1. Details about Datasets

To train the RouteLLM router, we randomly sample 12,247 data points from the Chatbot Arena dataset (Chiang et al., 2024). In contrast, both the Confidence-Driven Router and TO-Router are trained using a combination of Natural QA (Kwiatkowski et al., 2019; Lee et al., 2019), Trivia QA (Joshi et al., 2017), PopQA (Mallen et al., 2022), and MAWPS (Koncel-Kedziorski et al., 2016) datasets to ensure a comparable data volume. Specifically, we randomly selected 3,610 samples from each QA dataset and 1,418 samples from the MAWPS dataset, resulting in 12,247 samples, matching the quantity used for RouteLLM.

To comprehensively evaluate the routing systems, we select a diverse set of benchmark datasets: the MMLU (Hendrycks et al., 2020) dataset, consisting of 14,042 questions across 57 subjects; the MT-Bench dataset (Zheng et al., 2023), which includes 160 open-ended questions assessed using the LLM-as-a-judge approach; and the GSM8K dataset (Cobbe et al., 2021), containing over 1,000 grade-school math problems. These datasets provide a broad evaluation across varied question types and subject domains. For all the data listed above, we properly use them under the propose of research by following their license.

A.2. System Prompt Design for LLM-as-a-Judge

To evaluate response quality, we use LLM-as-a-Judge to simulate human ratings. We employ an independent LLM (i.e., GPT-o1) to choose the most preferable responses from the routed model alongside ground-truth answers. The judge is instructed to select based on correctness and precision of reasoning. We designed and implemented the following system prompt:

```
You are an evaluator for math problem solutions. You will receive:
1. A question.
2. A ground truth answer.
3. Three LLM-generated responses.
Your task is to select which response(s) is/are best, based on whether
   the answer is correct and the reasoning is precise.
Follow these rules:
* DO NOT provide any explanation or reasoning in your answer-only
   state which LLM(s) you judge as having the best response.
* If more than one response is equally best, name each of them.
Question: {}
Ground Truth Answer:{}
LLM 1 Response: {}
LLM 2 Response: {}
LLM 3 Response: {}
Your output must ONLY indicate the selected LLM(s). For example, 'LLM
   1' or 'LLM 1 and LLM 3'.
```