NAACL 2025

NEC Laboratories America

Arizona State University

MixLLM: Dynamic Routing in Mixed Large Language Models

Xinyuan Wang[1], Yanchi Liu[2], Wei Cheng[2], Xujiang Zhao[2],
Zhengzhang Chen[2], Wenchao Yu[2], Yanjie Fu[1], Haifeng Chen[2]

[1]Arizona State University, [2]NEC Labs America

1. Motivation

Mistral 7B

Llama 3 70B

GPT-3.5 Turbo

GPT-4o

Response Quality

We want to select the model which can answer the query correctly.

Cost

At the set level, the comparable response quality with lower cost is possible.

Latency

We don't want the query to queue for a long time.

We aim to balance response quality, cost, and latency to achieve the trade-off.

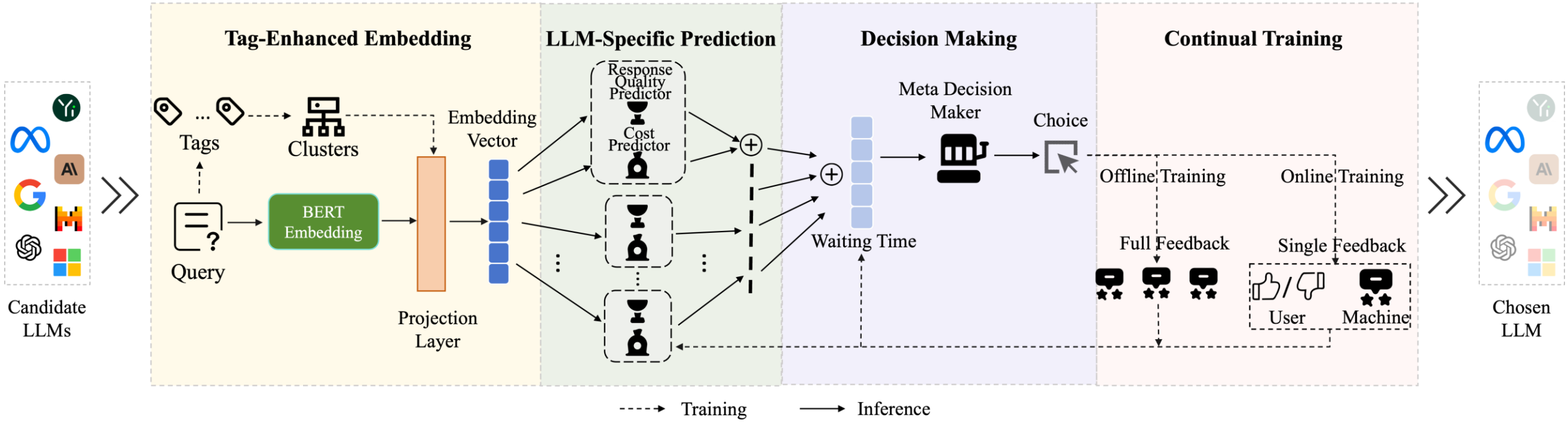
2. Key Challenges & Our Solutions

- **Challenge 1**: Dynamic trade-offs among quality, cost, and latency.
- **Insight**: Smart LLM selection reduces cost while maintaining response quality.
- **Solution**: Predict quality and cost and introduce the time penalty to perform query-specific LLM assignments.

- **Challenge 2**: Enabling continual learning in deployed systems.
- **Insight**: Using feedback improves performance on evolving queries.
- **Solution**: Real-time learning (user feedback) refines routing choices.

- **Challenge 3**: Navigating a varying set of LLM candidates over time (e.g., new LLM addition or old LLM removal).
- **Insight**: Dynamically add or remove LLMs without retraining the entire system.
- **Solution**: The LLM-specific prediction enables plug-and-play integration.

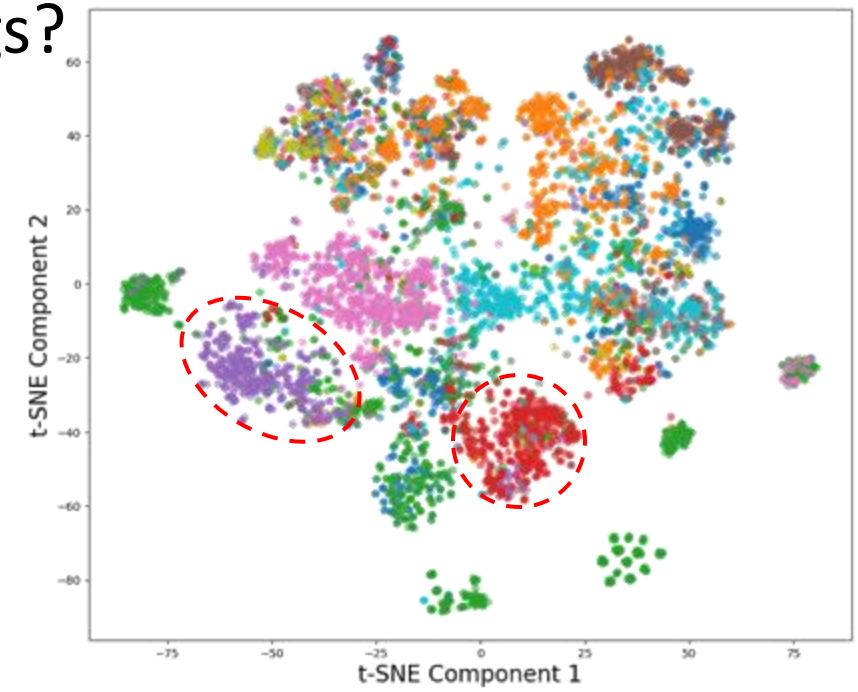
3. MixLLM: Key Components and Workflow

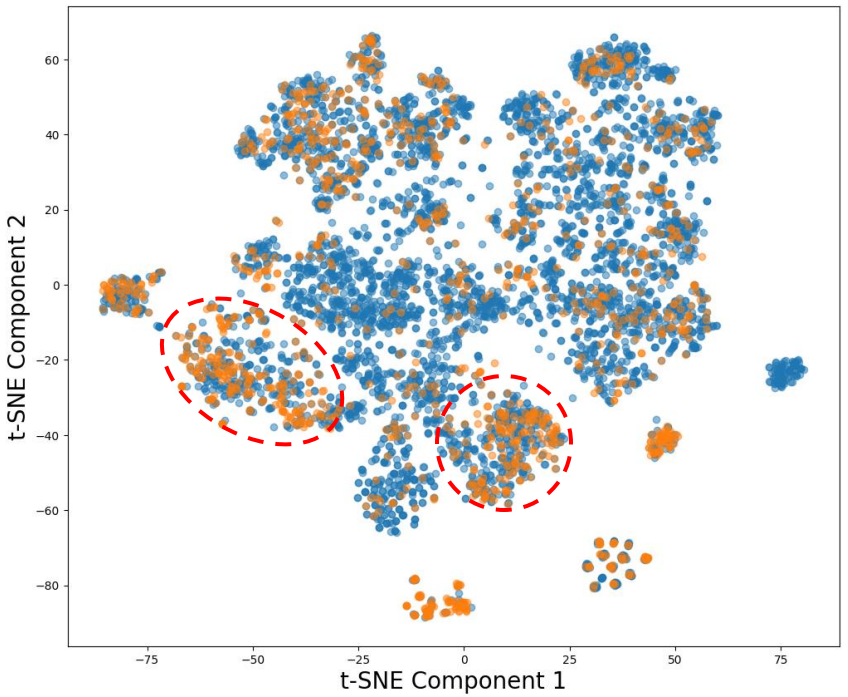


- Informative embedding
- Time penalty
- Individual prediction
- Feedback after depolyment

3.1. Tag-Enhanced Embedding

- Generate fine-grained query tags to train the encoder.
- Why Tags?





Tags have correlation with LLM response quality.

3.1. Tag-Enhanced Embedding

- Use BERT-based encoder for sentence embedding:

$$\mathbf{e}_n = \mathsf{Encoder}(q_n),$$

- Employ InsTag [1] to generate **query tags**, then cluster tags into relevant domains.

- Train encoder based on these **domain clusters**:

$$\mathcal{L}_{\mathsf{intra}} = -\frac{1}{|Q|}\sum_{i=1}^{|Q|}\log\frac{\exp(\mathbf{e}_i \cdot \boldsymbol{\mu}_i)}{\sum_{j=1}^{|D|}\exp(\mathbf{e}_i \cdot \boldsymbol{\mu}_j)}.$$

$$\mathcal{L}_{\mathsf{inter}} = \frac{1}{|D|}\sum_{j=1}^{|D|}\log\sum_{k\neq j}\exp(\boldsymbol{\mu}_j \cdot \boldsymbol{\mu}_k).$$

[1] Lu, Keming, et al. "# instag: Instruction tagging for analyzing supervised fine-tuning of large language models." *The Twelfth International Conference on Learning Representations*. 2023.

3.2. LLM-Specific Prediction

- For each candidate LLM:
  - Predict the **response quality** of this LLM on the current query:

$$\hat{p}_{n,l} = f_l^{\texttt{rq}}(\mathbf{e}_n; \boldsymbol{\theta}_l^{\texttt{rq}}),$$

  - Predict **response length** to estimate **total cost**:

$$\hat{\texttt{len}}_{n,l}^{\texttt{res}} = f_l^{\texttt{rl}}(\mathbf{e}_n; \boldsymbol{\theta}_l^{\texttt{rl}}),$$

$$\hat{c}_{n,l} = \underbrace{\texttt{len}_{n,l}^{\texttt{prm}} \cdot \texttt{price}_l^{\texttt{prm}}}_{\text{input cost}} + \underbrace{\hat{\texttt{len}}_{n,l}^{\texttt{res}} \cdot \texttt{price}_l^{\texttt{res}}}_{\text{output cost}},$$

3.3. Meta Decision Maker

- Select the **most suitable LLM** according to the score:

$$s_{n,l} = s_{n,l}^{\text{trade}} + \alpha \cdot s_{n,l}^{\text{unc}} - \beta \cdot s_{l}^{\text{pen}}.$$

  - **Quality vs. Cost Trade-off**: Finds the optimal balance.

$$s_{n,l}^{\text{trade}} = \frac{\lambda}{\lambda+1} \cdot \hat{p}_{n,l} - \frac{1}{\lambda+1} \cdot \hat{c}_{n,l},$$

  - **Uncertainty Correction**: Adjusts based on confidence in predictions.

$$s_{n,l}^{\text{unc}} = \mathbf{e}_n^T \cdot \mathbf{A}_l^{-1} \cdot \mathbf{e}_n,$$

  - **Time penalty**: Avoids excessive waiting time.

$$s_{l}^{\text{pen}} = e^{\gamma \cdot (w_l - \xi \cdot \tau)},$$

3.4. Continual Training

Offline Training

- Pre-deployment update
- **Full** feedback from **all** candidate LLMs

$$\boldsymbol{\theta}_l^{\mathtt{rq}} := \boldsymbol{\theta}_l^{\mathtt{rq}} - \eta_1 \cdot \nabla_{\boldsymbol{\theta}_l^{\mathtt{rq}}} \mathcal{L}(p_{n,l}, \hat{p}_{n,l}),$$

$$\boldsymbol{\theta}_l^{\mathtt{rl}} := \boldsymbol{\theta}_l^{\mathtt{rl}} - \eta_2 \cdot \nabla_{\boldsymbol{\theta}_l^{\mathtt{rl}}} \mathcal{L}(\mathtt{len}_{n,l}^{\mathtt{res}}, \hat{\mathtt{len}}_{n,l}^{\mathtt{res}}),$$

$$\mathbf{A}_l := \mathbf{A}_l + \mathbf{e}_n^T \cdot \mathbf{e}_n.$$

Online Training

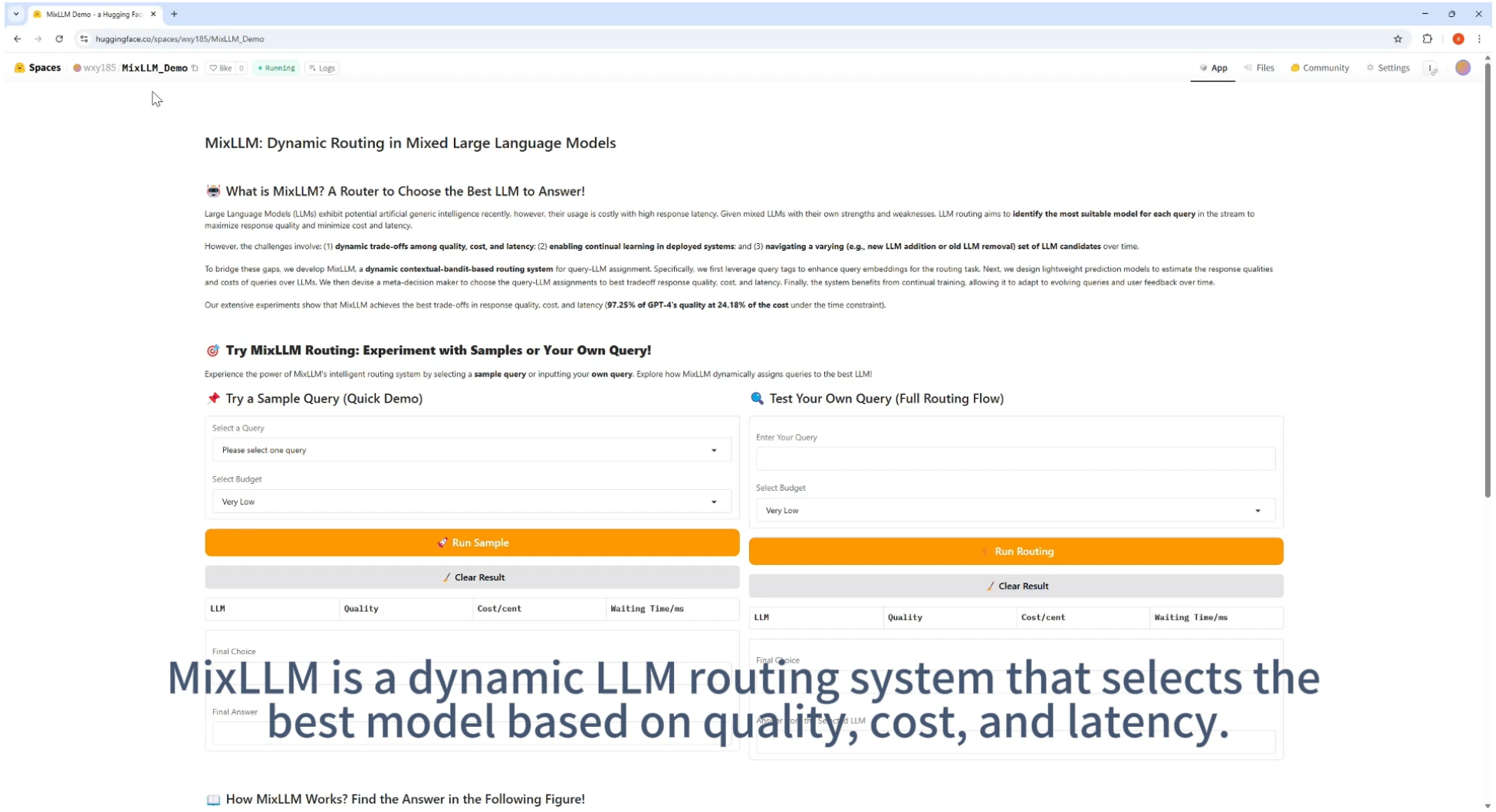
- Post-deployment update
- **Partial** feedback from the **selected** LLM
- Binary → Dynamic Feedback Score

$$s'_{n,l} = s_{n,l} + \kappa_{n,l} \cdot s_{n,l}^{\mathtt{df}},$$

$$\left[s_{n,1}^{\mathtt{df}}, s_{n,2}^{\mathtt{df}}, \ldots, s_{n,|M|}^{\mathtt{df}}\right] = f^{\mathtt{df}}(\mathbf{e}_n; \boldsymbol{\theta}^{\mathtt{df}}).$$

- Apply the Policy Gradient method to update the parameters.

4. MixLLM in Action: Live Demonstration



MixLLM is a dynamic LLM routing system that selects the best model based on quality, cost, and latency.

5. Evaluating MixLLM: Performance & Insights

- Dataset:
  - RouterBench: Consists of **36,497** queries from **8** NLP datasets. Each query is answered by **11** different LLMs.
  - Data Split: **80%** Training (**Offline** Training: Pre-train on all LLM responses), **20%** Testing (**Online** Training: Adapt using binary feedback)

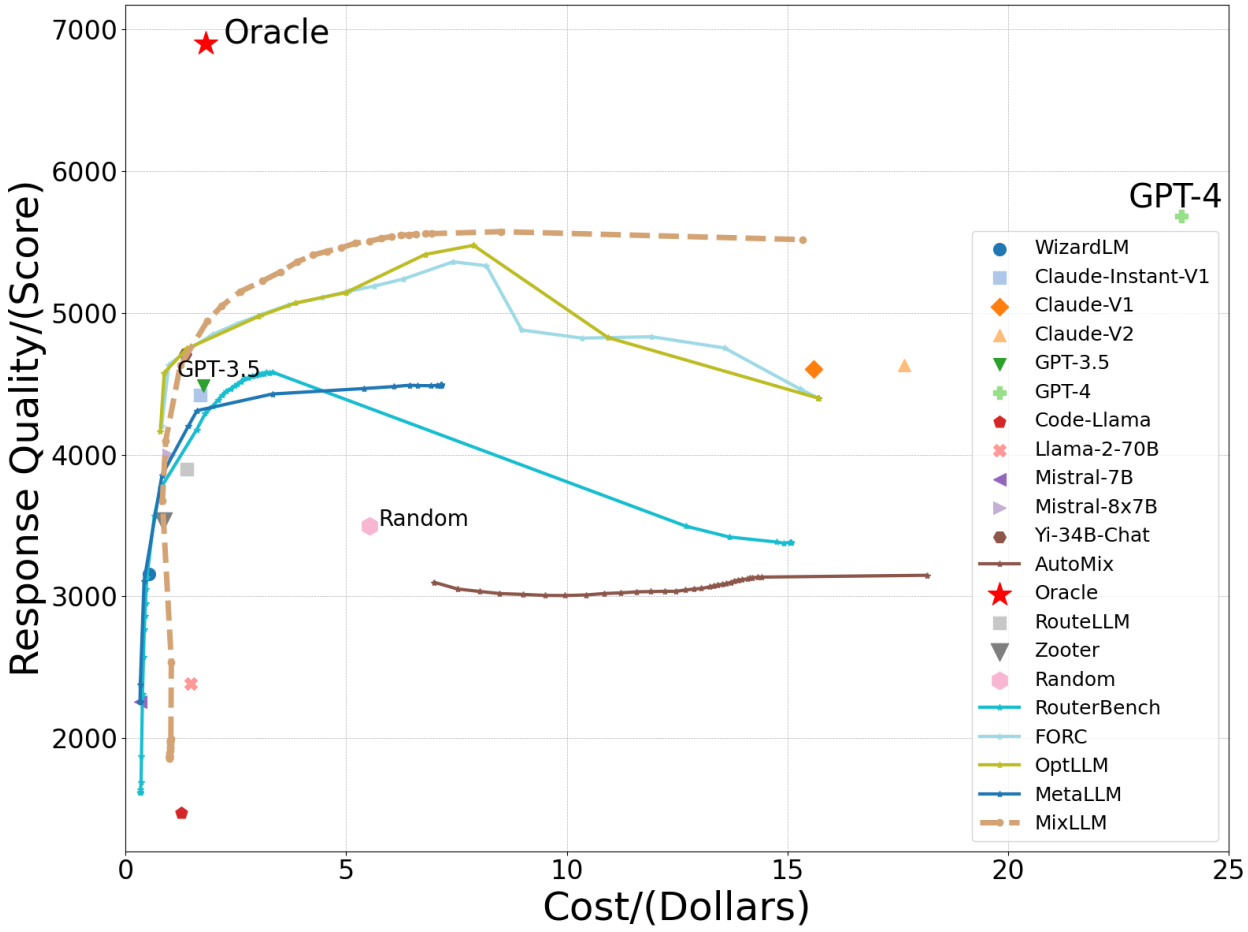
- Baselines:
  - AutoMix, RouteLLM, Zooter, RouterBench, FORC, OptLLM, MetaLLM.

- Metric (LLMs cost & latency) source:
  - https://artificialanalysis.ai/

5.1. Overall Routing Performance

- MixLLM:
  - outperforms baselines;
  - achieves **97.25%** of GPT-4's quality at **24.18%** of the cost under the time constraint;
  - remains stable when the budget is high.

- Why can response quality decline even with a high budget?
  - Higher budgets encourage using powerful LLM, where many queries exceed the waiting time tolerance.



5.2. Study on Continual Training

- Continual training offers improved performance.

Setting	Offline : Online		
	80:20	**50:50**	**30:70**
Without Online Training	75.54%	71.98%	69.74%
With Refined Feedback	76.45%	72.99%	71.29%
Improvement	**1.21%**	**1.39%**	**2.22%**
With Binary Feedback	75.93%	72.37%	70.65%
Improvement	**0.52%**	**0.53%**	**1.31%**

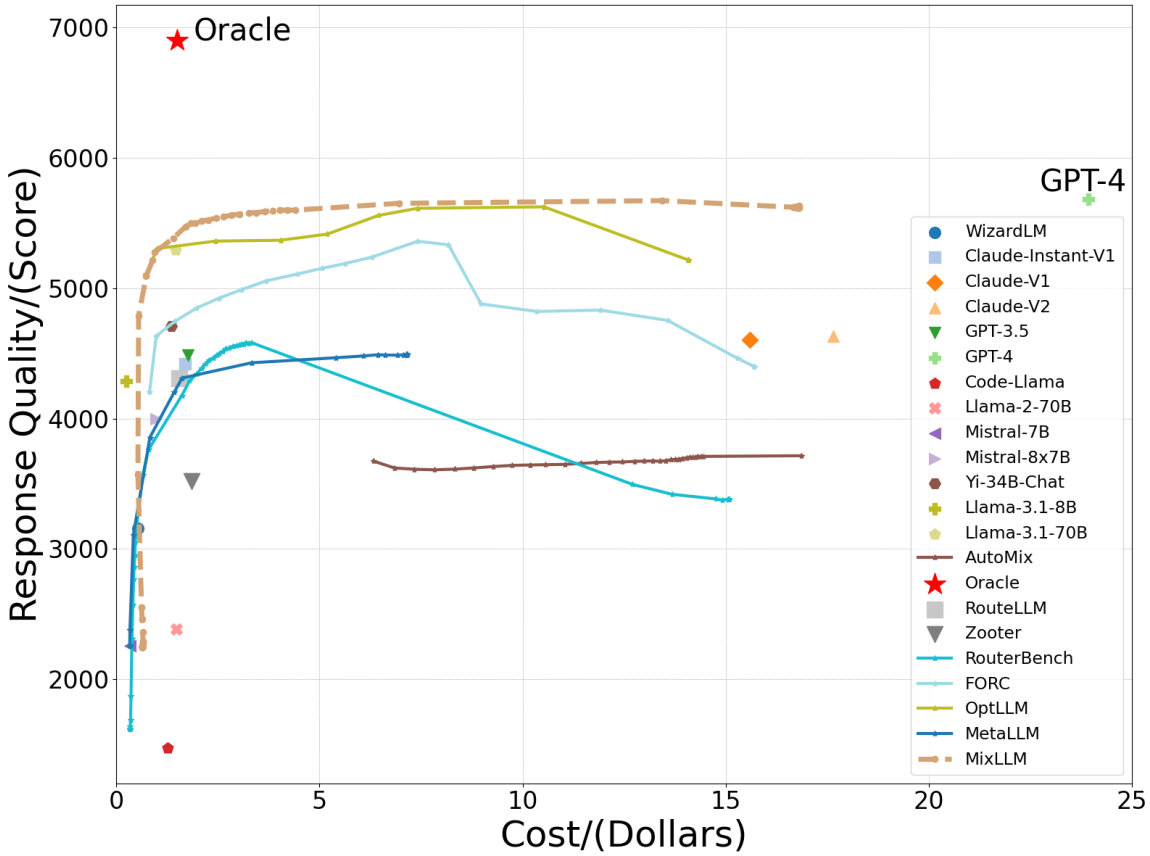
- In real-world applications, collecting full feedback is **difficult** and **expensive**.
- The responses to queries can serve as **partial feedback**.
- The amount of data during **inference** will far **exceed** that during training.

5.3. Study on Adaptive Training

- We add 2 new models:
  - Llama 3.1 8B;
  - Llama 3.1 70B.

- MixLLM achieves **98.55%** of GPT-4's response quality while reducing the cost to just **18.36%**.

- The original parameters remain **unchanged**. We only train **2 new** sets of prediction models.



5.4. Out-of-Domain Generalization

- Real-world queries often originate from new or unseen domains.
- OOD splitting: the test set contains non-overlapping domains not in the training set

Splitting Policy	Offline Only	Offline + Online
Normal 80:20 Splitting	75.54%	76.45%
OOD 80:20 Splitting	71.43%	73.89%
Decrease	5.44%	**3.35%**

- The offline-online training strategy effectively enhances domain generalization and adaptation.

- **How to solve the OOD routing task?**

6. Takeaways & Future Directions

- MixLLM **dynamically** routes queries to **the most suitable** LLM while maintaining a **balance** between response quality, cost, and latency.
- Extensive experiments confirm MixLLM's **effectiveness**: it achieves **97.25%** of GPT-4's quality at only **24.18%** of the cost.
- MixLLM includes **continual training**: it learns from large-scale **post-deployment** data and improves performance over time.
- MixLLM is highly **flexible**: it can add or remove LLM candidates **without requiring full retraining**.
- Future work will focus on improving **out-of-domain (OOD) generalization** and refining LLM **selection policies** for better performance.

NAACL 2025

NEC Laboratories America

Arizona State University

Q & A

Thank you for listening.

Looking forward to collaboration!

xwang735@asu.edu