



Efficient Training-Free Online Routing for High-Volume Multi-LLM Serving

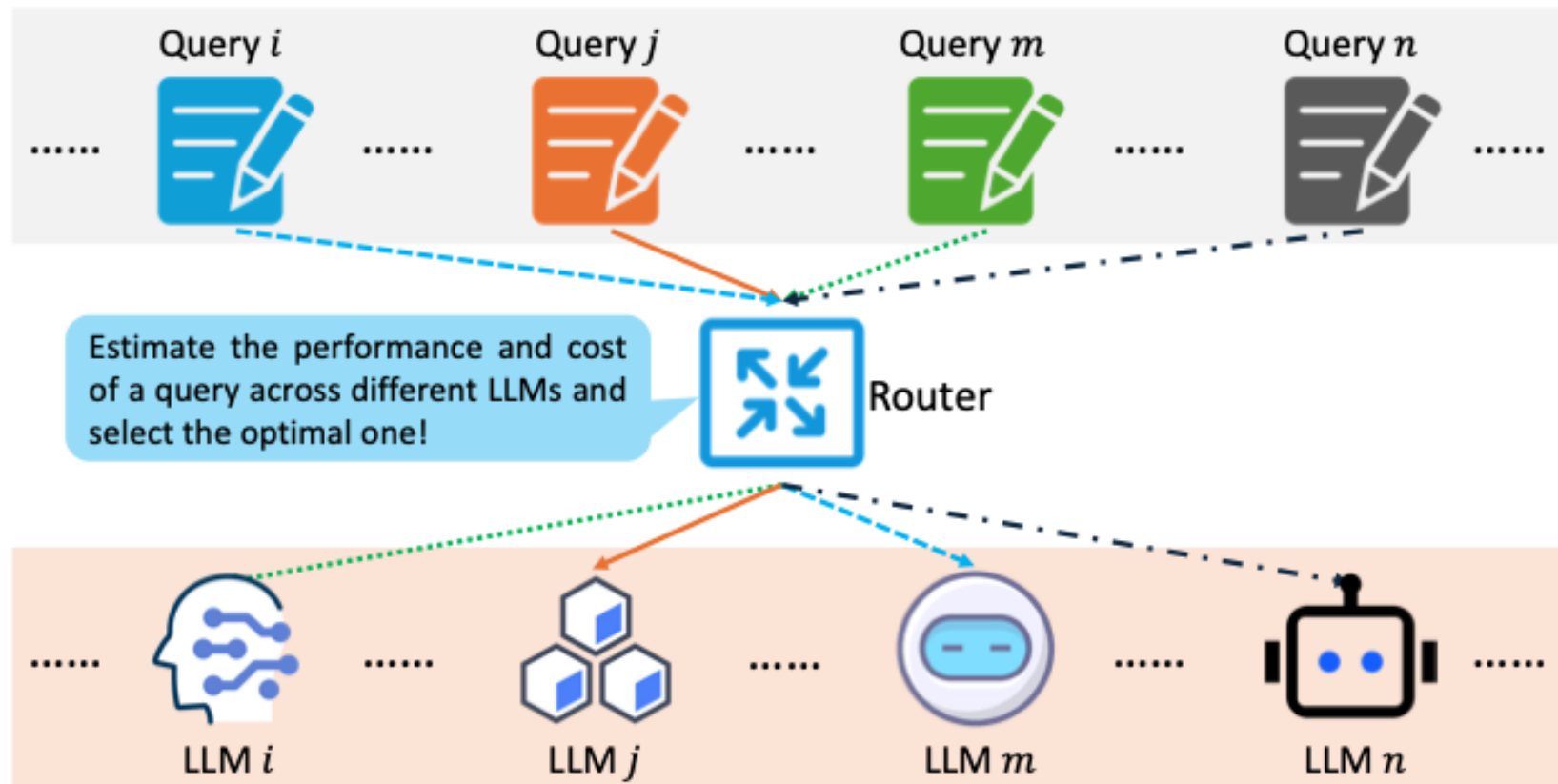
Fangzhou Wu Sandeep Silwal

University of Wisconsin–Madison



LLM Routing

- As Large Language Models (LLMs) become widely deployed, **managing rising query costs while ensuring high-quality service under tight token budgets** has become a key challenge for LLM-serving providers.
- **LLM routing** offers a cost-efficient solution by directing each query to the optimal LLM that best **balances response quality and inference cost**.



Formal Problem Formulation

- Consider an LLM-serving system deployed with M LLMs with each LLM i is assigned a token budget B_i .
- Sending query j to LLM i yields performance score d_{ij} and cost g_{ij} .
- **Objective:** Route a set of Q queries with a routing strategy $x(\cdot)$ that maximizes the overall quality of responses.

$$\begin{aligned} & \max \sum_{j \in Q} \sum_{i \in [M]} d_{ij} x_{ij} \\ & \text{s.t. } \sum_j g_{ij} x_{ij} \leq B_i \text{ for all } i, \\ & \sum_i x_{ij} \leq 1 \text{ for all } j, \\ & x_{ij} \in \{0, 1\} \end{aligned} \tag{1}$$

Formal Problem Formulation

- Consider an LLM-serving system deployed with M LLMs with each LLM i is assigned a token budget B_i .
- Sending query j to LLM i yields performance score d_{ij} and cost g_{ij} .
- **Objective:** Route a set of Q queries with a routing strategy $x(\cdot)$ that maximizes the overall quality of responses.

$$\begin{aligned} \max \quad & \sum_{j \in Q} \sum_{i \in [M]} d_{ij} x_{ij} \\ \text{s.t.} \quad & \sum_j g_{ij} x_{ij} \leq B_i \text{ for all } i, \\ & \sum_i x_{ij} \leq 1 \text{ for all } j, \\ & x_{ij} \in \{0, 1\} \end{aligned} \tag{1}$$

Solving this MILP
online is non-trivial:
key challenges arise
in practice!

Challenges

- **Inaccessible ground truth performance and cost.** For any query j , the true performance score d_{ij} and cost g_{ij} are **unavailable** without accessing the actual LLMs.

Challenges

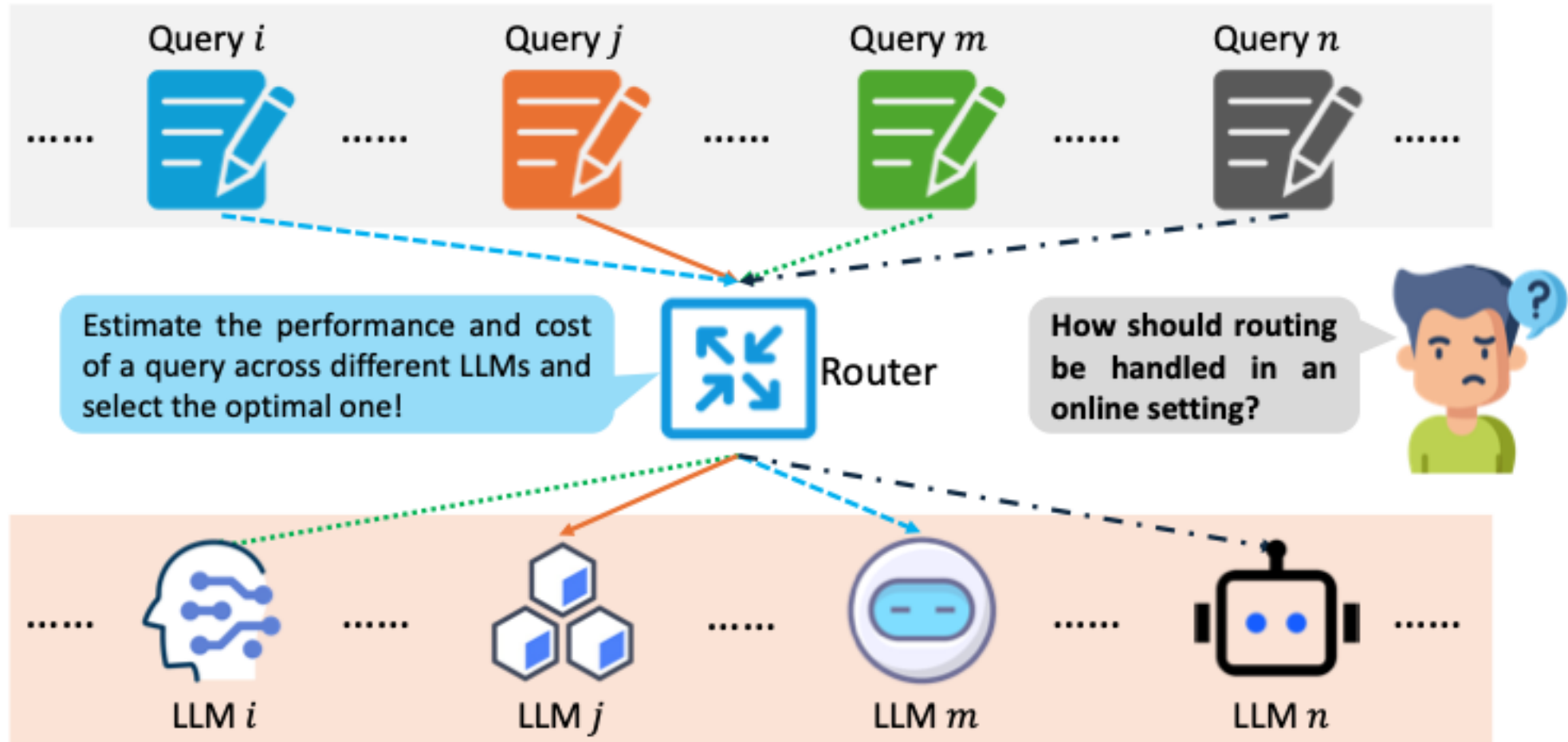
- **Inaccessible ground truth performance and cost.** For any query j , the true performance score d_{ij} and cost g_{ij} are **unavailable** without accessing the actual LLMs.
- **Deployment scalability.** LLM deployment configurations may vary across different environments. Routing algorithm must be adaptive to these variations while minimizing adaptation overhead.

Challenges

- **Inaccessible ground truth performance and cost.** For any query j , the true performance score d_{ij} and cost g_{ij} are **unavailable** without accessing the actual LLMs.
- **Deployment scalability.** LLM deployment configurations may vary across different environments. Routing algorithm must be adaptive to these variations while minimizing adaptation overhead.
- **Sequential query arrival under uncertainty.** In practice, queries arrive sequentially rather than simultaneously and must be routed without knowledge of future queries.

Motivation Question

- Can we design a **training-free online routing algorithm** that still achieves a near-optimal cumulative performance?



Efficient Performance and Cost Estimation

- For each query, we employ Approximate Nearest Neighbor Search (ANNS) to efficiently estimate its performance and cost for each deployed LLM using a historical dataset D .
- Key advantages:
 - **Deployment scalability:** Directly operate on D **without requiring model training.**
 - **Computational scalability:** ANNS algorithms are **much more efficient** than traditional KNN.

Online Routing from Observed Queries

- **Approximate LP with Control Parameter.**

- We approximate the original MILP by Equation (2) using the estimated \hat{d}_{ij} and cost \hat{g}_{ij} , and introduce **a control parameter α** .

$$\begin{aligned} \max \quad & \sum_{j \in Q} \sum_{i \in [M]} \alpha \hat{d}_{ij} x_{ij} \\ \text{s.t.} \quad & \sum_j \hat{g}_{ij} x_{ij} \leq B_i, \quad \forall i, \\ & \sum_i x_{ij} \leq 1, \quad \forall j, \\ & x_{ij} \in \{0, 1\}, \quad \forall i, j \end{aligned} \tag{2}$$

Online Routing from Observed Queries

- **Approximate LP with Control Parameter.**

- We approximate the original MILP by Equation (2) using the estimated \hat{d}_{ij} and cost \hat{g}_{ij} , and introduce a **control parameter α** .


- **Routing via Learned Weights.**

- We further relax Equation (2) to Equation (3) and derive its dual in Equation (4).


$$\begin{aligned} \max \quad & \sum_{j \in Q} \sum_{i \in [M]} \alpha \hat{d}_{ij} x_{ij} \\ \text{s.t.} \quad & \sum_j \hat{g}_{ij} x_{ij} \leq B_i, \forall i, \\ & \sum_i x_{ij} \leq 1, \forall j, \\ & x_{ij} \in \{0, 1\}, \forall i, j \end{aligned} \quad (2) \quad \xrightarrow{\text{orange arrow}} \quad \begin{aligned} \max \quad & \sum_{j \in Q} \sum_{i \in [M]} \alpha \hat{d}_{ij} x_{ij} \\ \text{s.t.} \quad & \sum_j \hat{g}_{ij} x_{ij} \leq B_i, \forall i, \\ & \sum_i x_{ij} \leq 1, \forall j, \\ & x_{ij} \in [0, 1], \forall i, j \end{aligned} \quad (3) \quad \xrightarrow{\text{orange arrow}} \quad \begin{aligned} \min \quad & \sum_{i \in [M]} \gamma_i B_i + \sum_{j \in Q} \beta_j \\ \text{s.t.} \quad & \beta_j \geq \alpha \hat{d}_{ij} - \hat{g}_{ij} \gamma_i, \forall i, j, \\ & \gamma_i \geq 0, \quad \beta_j \geq 0, \forall i, j \end{aligned} \quad (4)$$

Online Routing from Observed Queries

$$\begin{aligned}
 & \max \sum_{j \in Q} \sum_{i \in [M]} \alpha \hat{d}_{ij} x_{ij} \\
 & \text{s.t. } \sum_j \hat{g}_{ij} x_{ij} \leq B_i, \forall i, \\
 & \sum_i x_{ij} \leq 1, \forall j, \\
 & x_{ij} \in \{0, 1\}, \forall i, j
 \end{aligned} \tag{2}$$



$$\begin{aligned}
 & \max \sum_{j \in Q} \sum_{i \in [M]} \alpha \hat{d}_{ij} x_{ij} \\
 & \text{s.t. } \sum_j \hat{g}_{ij} x_{ij} \leq B_i, \forall i, \\
 & \sum_i x_{ij} \leq 1, \forall j, \\
 & x_{ij} \in [0, 1], \forall i, j
 \end{aligned} \tag{3}$$



$$\begin{aligned}
 & \min \sum_{i \in [M]} \gamma_i B_i + \sum_{j \in Q} \beta_j \\
 & \text{s.t. } \beta_j \geq \alpha \hat{d}_{ij} - \hat{g}_{ij} \gamma_i, \forall i, j, \\
 & \gamma_i \geq 0, \quad \beta_j \geq 0, \forall i, j
 \end{aligned} \tag{4}$$

- By complementary slackness, we have:

$$x_{ij} > 0 \Leftrightarrow \beta_j = \max_i (\alpha \hat{d}_{ij} - \hat{g}_{ij} \gamma_i)$$

Online Routing from Observed Queries

$$\begin{aligned}
 & \max \sum_{j \in Q} \sum_{i \in [M]} \alpha \hat{d}_{ij} x_{ij} \\
 & \text{s.t. } \sum_j \hat{g}_{ij} x_{ij} \leq B_i, \forall i, \\
 & \sum_i x_{ij} \leq 1, \forall j, \\
 & x_{ij} \in \{0, 1\}, \forall i, j
 \end{aligned} \quad (2) \quad \xrightarrow{\text{orange arrow}} \quad
 \begin{aligned}
 & \max \sum_{j \in Q} \sum_{i \in [M]} \alpha \hat{d}_{ij} x_{ij} \\
 & \text{s.t. } \sum_j \hat{g}_{ij} x_{ij} \leq B_i, \forall i, \\
 & \sum_i x_{ij} \leq 1, \forall j, \\
 & x_{ij} \in [0, 1], \forall i, j
 \end{aligned} \quad (3) \quad \xrightarrow{\text{orange arrow}} \quad
 \begin{aligned}
 & \min \sum_{i \in [M]} \gamma_i B_i + \sum_{j \in Q} \beta_j \\
 & \text{s.t. } \beta_j \geq \alpha \hat{d}_{ij} - \hat{g}_{ij} \gamma_i, \forall i, j, \\
 & \gamma_i \geq 0, \quad \beta_j \geq 0, \forall i, j
 \end{aligned} \quad (4)$$



- By complementary slackness, we have:

$$\begin{aligned}
 & x_{ij} > 0 \Leftrightarrow \beta_j = \max_i (\alpha \hat{d}_{ij} - \hat{g}_{ij} \gamma_i) \\
 & \min \sum_{i \in [M]} \gamma_i B_i + \sum_{j \in Q} \beta_j \xrightarrow{\text{orange arrow}} F(\gamma) = \sum_i \gamma_i B_i + \sum_j \max_i (\alpha \hat{d}_{ij} - \hat{g}_{ij} \gamma_i) \quad (5)
 \end{aligned}$$

This motivates treating γ as a set of learnable weights across LLMs to aid the query routing.

Online Routing from Observed Queries

$$\begin{aligned}
 & \max \sum_{j \in Q} \sum_{i \in [M]} \alpha \hat{d}_{ij} x_{ij} \\
 & \text{s.t. } \sum_j \hat{g}_{ij} x_{ij} \leq B_i, \forall i, \\
 & \sum_i x_{ij} \leq 1, \forall j, \\
 & x_{ij} \in \{0, 1\}, \forall i, j
 \end{aligned} \tag{2}$$

$$\begin{aligned}
 & \max \sum_{j \in Q} \sum_{i \in [M]} \alpha \hat{d}_{ij} x_{ij} \\
 & \text{s.t. } \sum_j \hat{g}_{ij} x_{ij} \leq B_i, \forall i, \\
 & \sum_i x_{ij} \leq 1, \forall j, \\
 & x_{ij} \in [0, 1], \forall i, j
 \end{aligned} \tag{3}$$

$$\begin{aligned}
 & \min \sum_{i \in [M]} \gamma_i B_i + \sum_{j \in Q} \beta_j \\
 & \text{s.t. } \beta_j \geq \alpha \hat{d}_{ij} - \hat{g}_{ij} \gamma_i, \forall i, j, \\
 & \gamma_i \geq 0, \quad \beta_j \geq 0, \forall i, j
 \end{aligned} \tag{4}$$

$$F(\gamma) = \sum_i \gamma_i B_i + \sum_j \max_i (\alpha \hat{d}_{ij} - \hat{g}_{ij} \gamma_i) \tag{5}$$

- Since computing the global optimal γ for all Q is infeasible online, we estimate it with γ^* from the first $P = \epsilon |Q|$ queries by solving Equation (6):

$$F(\gamma, P) = \epsilon \sum_i \gamma_i B_i + \sum_{j \in P} \max_i (\alpha \hat{d}_{ij} - \gamma_i \hat{g}_{ij}) \tag{6}$$

Online Routing from Observed Queries

$$F(\gamma, P) = \epsilon \sum_i \gamma_i B_i + \sum_{j \in P} \max_i (\alpha \hat{d}_{ij} - \gamma_i \hat{g}_{ij}) \quad (6)$$

- **Routing Rule.** The estimated γ^* is then used to route future queries by assigning each query j to the LLM i that maximizes:

$$i = \arg \max_i (\alpha \hat{d}_{ij} - \hat{g}_{ij} \gamma_i^*)$$

- **Theoretical Guarantee.** We provide formal theoretical guarantees demonstrating that our algorithm achieves a competitive ratio of $1 - o(1)$ under mild assumptions

Main Results

- Extensive experiments on 3 benchmarks demonstrate that our algorithm consistently **outperforms all 8 baselines across performance, cost efficiency, and throughput**.
- As shown in the table below, our algorithm outperforms all baselines on average by **3.55×** in performance, **1.85×** in cost efficiency, and nearly **4.25×** in throughput.

Algorithm	RouterBench					SPROUT					Open LLM Leaderboard v2				
	Performance	Cost	Cost Efficiency	Throughput	Relative Performance	Performance	Cost	Cost Efficiency	Throughput	Relative Performance	Performance	Cost	Cost Efficiency	Throughput	Relative Performance
Random	1384.25	0.427	3243.25	3276	43.10%	2827.6	0.72	3927.29	4742	47.61%	953.0	0.741	1284.37	2877	49.89%
Greedy-Perf	1012.1	0.27	3742.379	1687	31.52%	764.9	0.406	1881.742	1083	12.88%	553.0	0.499	1107.91	1189	28.95%
Greedy-Cost	1626.25	0.46	3534.46	4061	50.64%	3934.7	0.849	4630.41	6789	66.25%	1051.0	0.766	1371.30	3164	55.02%
KNN-Perf	1005.1	0.27	3720.58	1677	31.3%	769.6	0.407	1888.46	1084	12.96%	556.0	0.498	1114.29	1194	29.11%
KNN-Cost	1592.05	0.46	3454.04	4027	49.58%	3905.1	0.85	4593.37	6709	65.75%	991.0	0.766	1293.07	3172	51.88%
BatchSplit	1838.05	0.458	4005.93	3903	57.24%	3975.5	0.83	4784.49	6221	66.94%	1059.0	0.76	1392.07	3099	55.44%
Roberta-Perf	154.5	0.077	2019.00	190	4.81%	458.9	0.283	1621.64	536	7.73%	153.0	0.207	738.21	283	8.01%
Roberta-Cost	481.4	0.129	3738.88	1292	14.99%	3996.2	0.848	4709.22	6765	67.29%	1044.0	0.766	1362.53	3173	54.66%
Ours	2718.6	0.447	6075.58	5195	84.66%	4513.05	0.815	5536.74	7475	75.99%	1465.0	0.711	2060.3	3692	76.7%
<i>Offline Oracle (Algorithm Upper Bounds Reference)</i>															
Approx Optimum(\hat{C}_{opt})	3211.35	0.46	6975.16	6225	100%	5938.99	0.85	6986.45	8781	100%	1910.0	0.765	2493.66	4319	100%
Optimum (C_{opt})	6376.9	0.46	13865.62	6436	198.57%	11934.4	0.848	14060.34	12336	200.94%	4688.0	0.763	6143.64	4688	245.44%

Computational Scalability

- We vary query volume from 4000 to 12000 and observe that our algorithm **consistently outperforms all baselines, maintaining top performance and robustness as load increases.**

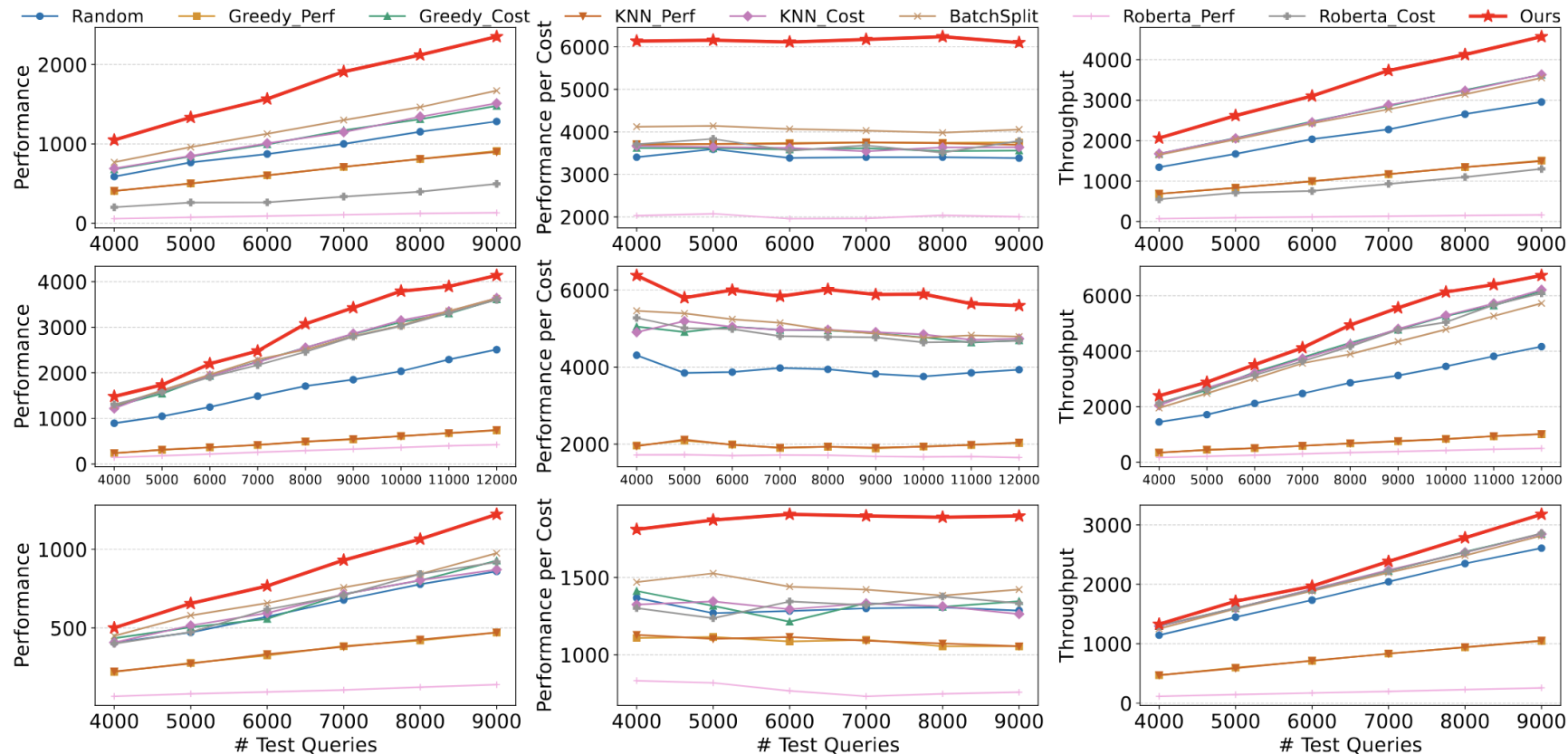
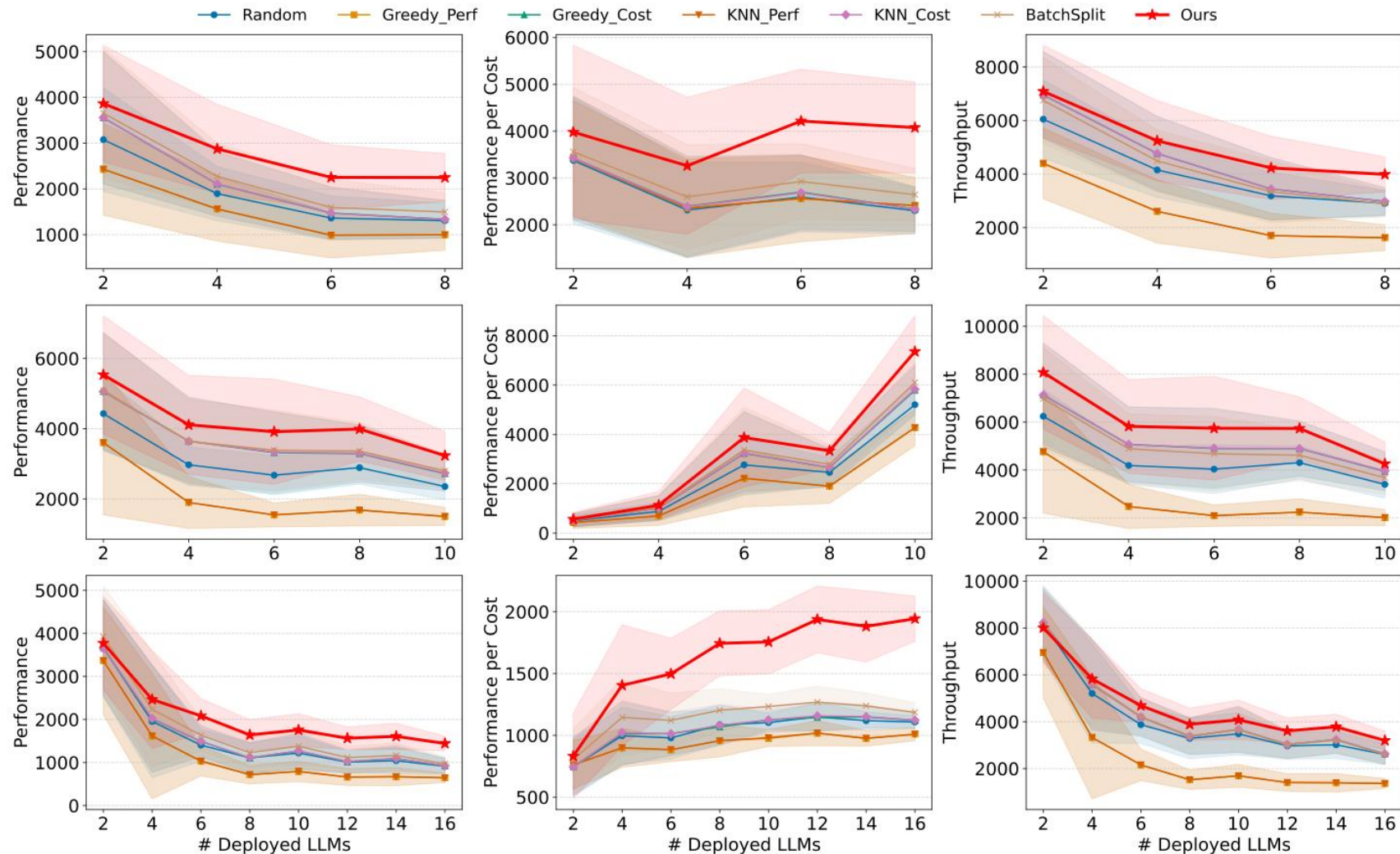


Figure 1: Results with test query volume varying from 4000 to 9000 (12000). Rows correspond to different datasets: RouterBench (top), SPROUT (middle), and Open LLM Leaderboard v2 (bottom).

Scalability Across LLM Deployments

- Our method consistently **outperforms training-free baselines under diverse LLM deployment configurations, highlighting its scalability.**



Thanks!