Federate the Router: Learning Language Model Routers with Sparse and Decentralized Evaluations

Baris Askin[*]
Carnegie Mellon University

Shivam Patel[*]
Carnegie Mellon University

Anupam Nayak[*]
Carnegie Mellon University

Andrea Vigano
NVIDIA

Jiin Woo
Amazon

Gauri Joshi
Carnegie Mellon University

Carlee Joe-Wong
Carnegie Mellon University

Abstract

Large language models (LLMs) are increasingly accessed as remotely hosted services by edge and enterprise clients that cannot run frontier models locally. Since models vary widely in capability and price, routing queries to models that balance quality and inference cost is essential. Existing router approaches assume access to centralized query–model evaluation data. However, these data are often fragmented across clients, such as end users and organizations, and are privacy-sensitive, which makes centralizing data infeasible. Additionally, per-client router training is ineffective since local evaluation data is limited and covers only a restricted query distribution and a biased subset of model evaluations. We introduce the first federated framework for LLM routing, enabling clients to learn a shared routing policy from local offline query–model evaluation data. Our framework supports both parametric multilayer perceptron router and nonparametric K-means router under heterogeneous client query distributions and non-uniform model coverage. Across three benchmarks, federated collaboration improves the accuracy–cost frontier over client-local routers, both via increased effective model coverage and better query generalization. Our theoretical results also validate that federated training reduces routing suboptimality.

Code Repository

1 Introduction

Large language models (LLMs) have become general-purpose tools for language-centric tasks, spanning customer support, search, programming assistance, and scientific writing. Their prompt-conditioned flexibility [8] allows a single model to support a broad and continually expanding set of applications [49, 5, 60]. In many deployments, however, these capabilities are utilized by *edge* devices and in client-facing environments (e.g., mobile devices, on-premises applications, enterprise endpoints) which are limited in terms of compute, memory, and power and thus cannot host most large language models locally [81]. As a result, queries are served by *remotely hosted* models, which are accessed through application programming interfaces (APIs). Such remote hosting enables clients to select suitable models for individual queries by simply accessing different model APIs.

In practical scenarios, real-world queries are highly heterogeneous, spanning diverse tasks and varying substantially in difficulty. Some queries require sophisticated mathematical reasoning or long-context

[*]Equal contribution. Correspondence to: Baris Askin <baskin@andrew.cmu.edu>

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

understanding [2], while others may demand specialized domain knowledge [19]. Inference costs also vary widely across models, making the use of capable (and generally more expensive) models for simple queries inefficient [73]. Furthermore, the query inference cost, which depends on the number of tokens in the response, is unknown prior to generating the response. Consequently, no single model is uniformly best once both quality and cost are factored in, motivating mechanisms that utilize multiple models to deliver high-quality responses efficiently [68].

Language query routing aims to map each incoming query to a suitable model from the pool of models, typically to optimize an accuracy-cost trade-off [48]. Routers are trained on historical query–model evaluations, i.e., measured quality (e.g., correctness) and associated costs [32]. Given a query embedding (generated from a sentence encoder), *parametric* routers (e.g., Multilayer Perceptron (MLP) predictors) [22, 48] estimate per-model response accuracy and cost and then select the best model under their desired trade-off, while *nonparametric* routers (e.g., K-Means clustering router) [32, 35] summarize evaluations in the query embedding space and select models using local neighborhood statistics. Despite their algorithmic differences, both families of routers rely on a diverse dataset of queries that are evaluated on many language models, allowing them to learn a reliable mapping of queries to model capabilities.

In practice, constructing a centralized query-evaluation dataset for router training is challenging. Firstly, evaluating queries on all models is prohibitively expensive, and this cost is exacerbated by rapidly evolving model pools, requiring continuous evaluation[2]. Second, even defining a "broad" query set is nontrivial as public benchmarks and datasets are not representative of real usage, and practical query distributions are long-tailed and dynamic [3, 47]. Realistic and valuable evaluations come from individual users (*clients*) themselves, i.e., from their actual queries/prompts and observed outcomes. These evaluations are naturally fragmented across clients, ranging from individual end users to organizations deploying LLM-backed systems (e.g., customer-support agents, or internal coding copilots) [9, 45]. Third, each client typically possesses queries from its own domain and evaluates them on only a small subset of the model pool, and model usage among that subset can be highly skewed leaving many models severely under-sampled [80, 45]. As a result, individual clients have limited data for learning a router that generalizes to a wide distribution of queries. Privacy and regulatory constraints prevent centralizing raw query-evaluation data such as personal/proprietary data, customer conversations and internal code [59, 70].

The key insight of this paper is to propose federated learning (FL) [46] as a solution to these challenges. Rather than collecting client queries and their model evaluations in a central dataset, federated learning allows clients to collaborate and transfer supervision across semantically similar requests by sharing routing policies they locally train on private offline datasets. This enables training on diverse and privacy-sensitive query distributions, and additionally supports sparse and imbalanced client coverage over models.

We develop frameworks spanning both canonical router methods in the federated setting: parametric `MLP-Router` and nonparametric `K-Means-Router`. We then demonstrate that these federated frameworks solve the challenges of learning routing policies from sparse, fragmented client data. Empirically, federated routers consistently improve over routers trained on any single client's data: they generalize better on aggregate query distributions and also improve routing performance on individual clients by leveraging cross-client signal without exposing private client queries.

Contributions. After reviewing related work in FL and LLM routing (Section 2), we contribute the following:

• In Section 3, **we formulate LLM routing in a federated setting** that addresses practical problems: (i) prompts and responses are private and remain on clients, (ii) each client has limited query volume and a narrow, biased slice of the global prompt distribution, and (iii) query–model evaluations are sparse and imbalanced. To our knowledge, this is the first work to study LLM routing through a federated learning framework.

• In Section 4, **we propose federated training procedures for both `MLP-Router` and `K-Means-Router`**, enabling collaborative learning of routing-policy, which supports evolving model and client pools.

[2]Hugging Face had approximately 1M models by the end of 2024 and over 2M models by the end of 2025 [25, 63].

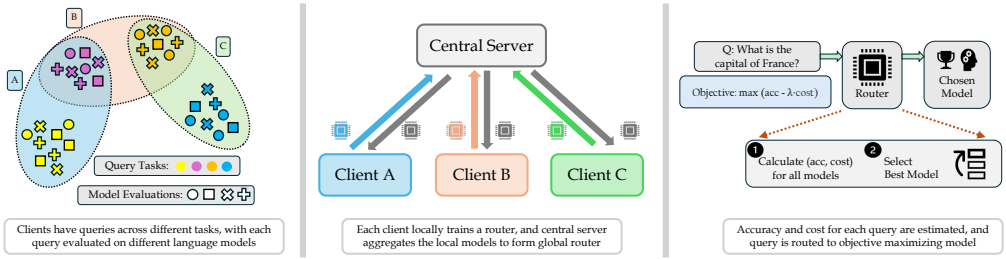


Figure 1: **Federated LLM Router System Description.** (i) Clients possess queries from different tasks, and each query is evaluated only on some LMs (in our experiments, we consider the extreme case of one LM per query, where the choice of LM is client-specific and can be highly non-uniform). (ii) Federated Learning setup where clients train their local router on local data, and central server aggregates these local models and returns global model to clients. (iii) Functioning of router, where modelwise accuracy and cost is calculated for the given query, which is routed to the most suitable model that maximizes the objective value.

- In Section 5, we establish convergence guarantees for federated optimization of `MLP-Router`, and **provide theoretical support** for why FL improves routing under sparse evaluations by increasing effective query and model coverage through cross-client aggregation for both routers.
- In Section 6, **we empirically evaluate both router families under heterogeneity and incomplete query–model evaluation data**. Our proposed federated training improves client-local routers for both out-of-distribution (better generalization) and in-distribution (increased effective model coverage) data. For high heterogeneity regime, we also propose an *adaptive personalization mechanism* that interpolates between the federated router and a client's local router, improving robustness when global collaboration can be misaligned with a client's own distribution. Section 7 concludes the paper, and additional discussion, experiments, and proofs are provided in the Appendix.

2 Related Work

Federated Learning and LLMs Federated learning (FL) trains shared models from decentralized data while keeping raw client data local [46]. In practice, FL must handle *statistical heterogeneity* as client data are non-iid, yet client distributions often share structure that collaborative training can exploit to improve generalization [36]. LLMs have advanced rapidly in scale and capability, making them the default backbone for many language-centric applications. However, adapting LLMs into FL is challenging because naïvely communicating or training billions of parameters is prohibitive for many clients. Accordingly, federated LLM research largely focuses on resource-aware adaptation, for example, parameter-efficient and quantization-aware fine-tuning, federated NLP/LLM benchmarks, federated instruction tuning under heterogeneous clients, and hybrid split paradigms that offload parts of the model [67, 52, 42, 74, 15, 11, 79]. In this work, we address a complementary problem: learning *federated routing policies* that enable model selection among multiple LLMs without centralizing client data. Routing requires query–model evaluations which are often expensive and distributed across clients. FL enables clients to collaboratively learn routers that generalize well across heterogeneous prompts and improve model coverage while keeping data local.

LLM Query Routing Query routing focuses on assigning a given query to the most suitable model to optimize a chosen tradeoff between response quality and inference cost. Most practical routers utilize a fixed dimensional query representation, generally obtained from pre-trained sentence encoders [10, 53]. Given such representations for queries, one direction of routers estimates modelwise correctness and cost on queries via small neural predictors or other scoring models [32, 22, 23, 82, 54, 33, 44, 76]. A complementary direction uses nonparametric designs that rely on the geometry of the embedding space and empirical performance statistics, e.g., clustering or nearest-neighbor selection [27, 31, 32, 35, 58, 41, 78, 77, 50]. Prior works utilize full evaluation data where all queries are evaluated on all models, and discussion on router design using sparse and decentralized evaluation data is largely missing. We address this practical problem in our work across both parametric and nonparametric router paradigms.

3 Problem Formulation

We consider a pool of M language models, denoted by $\mathcal{M}$. Each query is a text prompt s that we embed via a fixed, pretrained sentence encoder to obtain a representation[3] $\mathbf{x} \in \mathbb{R}^{d_{\text{emb}}}$, i.e., $\mathbf{x} = \text{Enc}(s)$ [10]. For each model $m \in \mathcal{M}$ and query embedding $\mathbf{x}$, let $\text{acc}(\mathbf{x}, m) \in [0, 1]$ denote the (expected) accuracy/performance score of the model m's response on that query, and let $\text{cost}(\mathbf{x}, m) \in [0, c_{\max}]$ denote the expected cost of responding to query, including input and output sequence costs [4].

Following the standard formulation [32, 61], we define the utility of prompt $\mathbf{x}$ for model m as its expected accuracy performance minus a penalty proportional to its expected cost (expectation taken over randomness of LLM responses):

$$U_\lambda(\mathbf{x}, m) := \Big(\text{acc}(\mathbf{x}, m) \; - \; \lambda \, \text{cost}(\mathbf{x}, m)\Big), \tag{1}$$

where the parameter $\lambda \geq 0$ controls the trade-off between cost and accuracy. Smaller λ prioritizes accuracy, while larger λ encourages cheaper models. A routing policy is a function that takes a prompt embedding as input and selects a corresponding model, written as $\pi : \mathbb{R}^{d_{\text{emb}}} \to \mathcal{M}$. Our objective is to learn routing policies that maximize the utility defined in eq. (1) given a query $\mathbf{x}$ and trade-off parameter λ.

In practice, $\text{acc}(\mathbf{x}, m)$ and $\text{cost}(\mathbf{x}, m)$ are unknown quantities that need to be estimated. While one can directly train a classifier that learns the routing policy (utility U_λ maximizing model) for a fixed λ [48, 33], we focus on modeling accuracy and cost functions via estimators $\mathsf{A}(\mathbf{x}, m) \approx \text{acc}(\mathbf{x}, m)$ and $\mathsf{C}(\mathbf{x}, m) \approx \text{cost}(\mathbf{x}, m)$ for all $m \in \mathcal{M}$, and then instantiate the router by plugging these estimates into Eq. (1). We then choose the model that maximizes the estimated utility for a given query $\mathbf{x}$ as the routing policy. This design is particularly natural in federated settings since different clients may have different budget/quality preferences, and can therefore choose λ at inference time without retraining the router.

In the federated setting, we consider N clients whose raw queries and embeddings are not shared, and only a ***single*** language model is prompted for each query. Each client i holds a local offline dataset $\mathcal{D}_i$ with $|\mathcal{D}_i| = D_i$:

$$\mathcal{D}_i \;=\; \left\{ \left(\mathbf{x}_i^{(j)},\, m_i^{(j)},\, \widehat{\text{acc}}_i^{(j)},\, \widehat{\text{cost}}_i^{(j)}\right) \right\}_{j=1}^{D_i}, \tag{2}$$

where $\mathbf{x}_i^{(j)} \in \mathbb{R}^{d_{\text{emb}}}$ is the embedding of the j-th local query and $m_i^{(j)} \in \mathcal{M}$ denotes the (single) model that is evaluated for query j at client i. $\widehat{\text{acc}}_i^{(j)}$ and $\widehat{\text{cost}}_i^{(j)}$ are observed accuracy and cost samples corresponding to the evaluation, where $\mathbb{E}\Big[\widehat{\text{acc}}_i^{(j)} \mid \mathbf{x}_i^{(j)}, m_i^{(j)}\Big] \;=\; \text{acc}\left(\mathbf{x}_i^{(j)}, m_i^{(j)}\right)$ and $\mathbb{E}\Big[\widehat{\text{cost}}_i^{(j)} \mid \mathbf{x}_i^{(j)}, m_i^{(j)}\Big] \;=\; \text{cost}\left(\mathbf{x}_i^{(j)}, m_i^{(j)}\right)$. We also denote the union of all local datasets by $\mathcal{D} = \bigcup_i \mathcal{D}_i$ with $|\mathcal{D}| = D = \sum_i D_i$.

4 Proposed Federated Routing Methods

In this section, we develop federated training and deployment pipelines for two common router families, parametric `MLP-Router` and nonparametric `K-Means-Router`. For both designs, we study how federated collaboration (Figure 1) mitigates (i) heterogeneous client query distributions and (ii) sparse and imbalanced query–model evaluations, while supporting personalization and dynamic participation.

4.1 Federated `MLP-Router`

The goal of the federated setting is to learn global estimators that predict the expected accuracy and cost for all $m \in \mathcal{M}$ given a prompt $\mathbf{x}$ and generalize across clients. We employ accuracy and cost estimators A_θ and C_θ, parametrized by $\theta \in \mathbf{\Theta}$. Specifically, we use a single `MLP-Router` consisting of a shared MLP trunk $h_\theta : \mathbb{R}^{d_{\text{emb}}} \to \mathbb{R}^{d_h}$ that maps a query embedding $\mathbf{x}$ to a hidden representation $h_\theta(\mathbf{x})$, followed by model-specific linear heads which output $\mathsf{A}_\theta(\mathbf{x}, m)$ and $\mathsf{C}_\theta(\mathbf{x}, m)$

[3] Throughout the text, we use terms prompt and query interchangeably to denote $\mathbf{x}$.

for each $m \in \mathcal{M}$ from $h_\theta(\mathbf{x})$. We denote the collection of all trunk and head parameters by θ and define the global empirical loss function $\mathcal{L}(\theta)$ given by,

$$\mathcal{L}(\theta) = \frac{1}{|\mathcal{D}|} \sum_{i=1}^{N} \sum_{j=1}^{|\mathcal{D}_i|} \Big[\ell\left(\widehat{\mathsf{acc}}(\mathbf{x}_i^{(j)}, m_i^{(j)}), \mathsf{A}_\theta(\mathbf{x}_i^{(j)}, m_i^{(j)})\right) + \ell\left(\widehat{\mathsf{cost}}(\mathbf{x}_i^{(j)}, m_i^{(j)}), \mathsf{C}_\theta(\mathbf{x}_i^{(j)}, m_i^{(j)})\right) \Big]. \tag{3}$$

We train `MLP-Router` using Federated Averaging (FedAvg) [46] as depicted in Algorithm 1. At each communication round, the server broadcasts the current global weights θ^t to the active clients. Each client i minimizes its empirical objective $\mathcal{L}_i(\theta) = \frac{1}{|\mathcal{D}_i|} \sum_{(\mathbf{x},m)\in\mathcal{D}_i} \big[\ell(\widehat{\mathsf{acc}}(\mathbf{x}, m), \mathsf{A}_\theta(\mathbf{x}, m)) + \ell(\widehat{\mathsf{cost}}(\mathbf{x}, m), \mathsf{C}_\theta(\mathbf{x}, m))\big]$ using τ steps of local stochastic gradient descent on its private dataset $\mathcal{D}_i$, and returns the updated parameters to the server. To quantify sample-wise discrepancies, we adopt Mean Squared Error (MSE) as the loss function ℓ. The server then aggregates the client updates via weighted averaging to produce the next global model. Finally, the `MLP-Router` induced by a parameterization θ is defined as the routing policy that maximizes the utility approximated by the estimators

$$\pi_{\theta,\lambda}(\mathbf{x}) = \arg\max_{m\in\mathcal{M}} U_\lambda(\mathbf{x}, m; \theta), \tag{4}$$

where $U_\lambda(\mathbf{x}, m; \theta) = \mathsf{A}_\theta(\mathbf{x}, m) - \lambda \mathsf{C}_\theta(\mathbf{x}, m)$.

Algorithm 1 Federated `MLP-Router` training

Input: clients $\{\mathcal{D}_i\}_{i=1}^N$, rounds T, local steps τ, LR η, init global model. θ^0
for $t = 0$ **to** $T-1$ **do**
  Send θ^t to active clients $\mathcal{S}^{(t)}$
  for client $i \in \mathcal{S}^{(t)}$ **in parallel do**
    Set local model $\theta_i \leftarrow \theta^t$
    for $s = 1$ **to** τ **do**
      $\theta_i \leftarrow \theta_i - \eta \widetilde{\nabla} \mathcal{L}_i(\theta_i)$
    end for
    Send updated model θ_i to server
  end for
  $\theta^{t+1} \leftarrow \frac{\sum_{i\in\mathcal{S}^{(t)}} |\mathcal{D}_i| \theta_i}{\sum_{i\in\mathcal{S}^{(t)}} |\mathcal{D}_i|}$
end for
Output: parameters θ^T

4.2 Federated `K-Means-Router`.

`K-Means-Router` provides a simple, training-free estimator for routing utilities by exploiting locality in the query-embedding space. Concretely, the method partitions embeddings into a finite set of regions (Voronoi cells) and treats the per-model accuracy and cost as approximately constant within each region. This yields a piecewise-constant nonparametric estimator. For a new query, we map it to its nearest cluster center and then use the cluster empirical averages of estimated accuracy and cost for each model. We use a text encoder function $\mathrm{Enc}(\cdot)$ to group semantically similar prompts [10, 53].

In this section, we adopt slightly different notation, since the accuracy ($\mathsf{A}_k^{(m)}$) and cost estimators ($\mathsf{C}_k^{(m)}$) operate on clusters rather than directly on prompts. Our federated training of `K-Means-Router` is elaborated in Algorithm 2. First, each client runs local K-means [43] with K_{local} centers on its own embeddings (line 3) and transmits the resulting centroids along with cluster sizes (line 4). The server performs weighted K-means (i.e., each centroid is weighted in the server's objective by its local cluster size, so the global centers minimize a size-weighted squared distance) over those centroids to obtain K_{global} global centers (line 6), which are broadcast back to clients (line 7). Using

Algorithm 2 Federated `K-Means-Router` training

1: **Input:** clients $\{\mathcal{D}_i\}_{i=1}^N$, nb. of local clusters K_{local}, nb. of global clusters K_{global}, distance metric $d(\cdot,\cdot)$
2: **for** each client $i \in \{1,\dots,N\}$ **in parallel do**
3: Run k-means on local embeddings $\{\mathbf{x}\}_{\mathbf{x}\in\mathcal{D}_i}$
4: Send $\{(\mu_{i,j}, n_{i,j})\}_{j=1}^{K_{\text{local}}}$ obtained in line 3 to server
5: **end for**
6: Server runs weighted k-means on $\bigcup_i \{(\mu_{i,j}, n_{i,j})\}_{j=1}^{K_{\text{local}}}$
7: Broadcast $\{\mu_k\}_{k=1}^{K_{\text{global}}}$ obtained in line 6 to all clients
8: **for** each client $i \in \{1,\dots,N\}$ **in parallel do**
9: **for** each cluster $k \in [K_{\text{global}}]$ and model $m \in \mathcal{M}$ **do**
10: Compute local counts $n_{i,k}^{(m)}$ and local averages: $\bar{a}_{i,k}^{(m)} = \frac{1}{n_{i,k}^{(m)}} \sum_{\substack{\mathbf{x}\in\mathcal{D}_i \\ k_\mu(\mathbf{x})=k}} \widehat{\mathsf{acc}}(\mathbf{x}, m)$, $\bar{c}_{i,k}^{(m)} = \frac{1}{n_{i,k}^{(m)}} \sum_{\substack{\mathbf{x}\in\mathcal{D}_i \\ k_\mu(\mathbf{x})=k}} \widehat{\mathsf{cost}}(\mathbf{x}, m)$
11: **end for**
12: Send $\{(\bar{a}_{i,k}^{(m)}, \bar{c}_{i,k}^{(m)}, n_{i,k}^{(m)})\}_{m,k}$ to server
13: **end for**
14: Server aggregates for all m, k:
$$\mathsf{A}_k^{(m)} \leftarrow \frac{\sum_i n_{i,k}^{(m)} \bar{a}_{i,k}^{(m)}}{\sum_i n_{i,k}^{(m)}}, \quad \mathsf{C}_k^{(m)} \leftarrow \frac{\sum_i n_{i,k}^{(m)} \bar{c}_{i,k}^{(m)}}{\sum_i n_{i,k}^{(m)}}.$$
15: Output $\theta = \left\{ \{\mu_k\}_{k=1}^{K_{\text{global}}}, \left\{\mathsf{A}_k^{(m)}, \mathsf{C}_k^{(m)}\right\}_{\substack{m\in\mathcal{M} \\ k\in[K_{\text{global}}]}} \right\}$

the received global centers, each client assigns its local samples to clusters (line 9), computes the average observed accuracy and cost for every (cluster, model) pair, and sends them to the server along with the number of evaluated samples (lines 10-13). No information is sent for (cluster, model) pairs with no samples. The server then takes weighted average of these statistics to obtain global estimators (line 14). At inference time, accuracy and cost statistics of closest cluster center to a query embedding $\mathbf{x}$ are used in Eq. (1) for routing. Both local and global K-means use Euclidean distance. K_{local} and K_{global} are chosen with validation experiments.

Finally, given the $\theta = \left\{\mu_{1:K_{\text{global}}}, \left\{\mathsf{A}_k^{(m)}, \mathsf{C}_k^{(m)}\right\}_{m\in\mathcal{M},\, k\in[K_{\text{global}}]}\right\}$[4] cluster centers and cost accuracy estimates, define the assigned clusters $k_\mu(\mathbf{x}) = \arg\min_{k\in[K_{\text{global}}]} \|\mathbf{x} - \mu_k\|_2$, and then the induced router is given as

$$\pi_{\theta,\lambda}(\mathbf{x}) := \arg\max_{m\in\mathcal{M}} \left(\mathsf{A}^{(m)}_{k_\mu(\mathbf{x})} - \lambda \mathsf{C}^{(m)}_{k_\mu(\mathbf{x})}\right).$$

Comparison of both methods. Both parametric and nonparametric routers have been examined in literature, showing complementary strengths and weaknesses. Parametric routers (like learned MLPs) can capture complex model behavior over a broad range of queries, improving generalization to unseen queries. In federated settings, we expect them to perform better due to the increased "collective coverage" by diverse clients. Alternatively, nonparametric routers (like clustering based routers) [35, 32, 50] are naturally incremental by design, eliminating the need for adapting or retraining router to accommodate new language models or queries. New language models can be onboarded by evaluating on the existing pool of queries and calculating cluster level averages, new queries can be incorporated through re-clustering and updating query embeddings. In contrast, a parametric router needs new head training or tuning parameters to adapt.

5 Theoretical Results

In this section, we provide informal statements of theoretical results to support our claims concerning the effectiveness of federated routing.

5.1 Federated `MLP-Router`

Theorem 5.1 (Convergence: Algorithm 1 - Informal)**.** *Under standard bounded heterogeneity, unbiased gradient, and smoothness assumptions [65, 37], with step size $\eta = \Theta\big(\sqrt{N/(\tau T)}\big)$, the empirical risk objective in Eq.* (3) *converges at the rate*

$$\min_{t\in\{1,\ldots,T\}} \mathbb{E}\Big[\big\|\nabla\mathcal{L}(\theta^{(t)})\big\|^2\Big] \le \tilde{\mathcal{O}}\left(\frac{1 + A\sigma^2}{\sqrt{N\tau T}}\right).$$

Here, $\tilde{\mathcal{O}}$ hides the faster decaying $1/T$ terms and σ^2 denotes the upper bound on heterogeneity in mini-batch gradients at a client, and the constant $A := N\sum_{i=1}^N (D_i/D)^2$.

The analysis is similar to Wang et al. [65]. When all clients have the same number of data points, i.e. $D_i/D = 1/N\ \forall i \in [N]$, we obtain a linear speedup of N (the number of clients). Moreover, the convergence rate also scales with the number of local steps τ.

Definition 5.2 (Suboptimality)**.** We use $\pi^\star : \mathcal{X} \to \mathcal{M}$ to denote the optimal router policy defined as

$$\pi^\star(\mathbf{x}) = \arg\max_{m\in\mathcal{M}} U_\lambda(\mathbf{x}, m), \tag{5}$$

define the suboptimality of a routing policy $\widehat{\pi}$ on a query distribution $\mathbb{D}^{\text{test}}$ and tradeoff parameter λ,

$$\text{Subopt}(\widehat{\pi}) := \mathbb{E}_{\mathbf{x}\sim\mathbb{D}^{\text{test}}}\left[U_\lambda\big(\mathbf{x}, \pi^\star(\mathbf{x})\big) - U_\lambda\big(\mathbf{x}, \widehat{\pi}(\mathbf{x})\big)\right].$$

Theorem 5.3 (Suboptimality - Informal)**.** *Let $\widehat{\theta}_i$ be the parameters of router learnt on the local dataset $\mathcal{D}_i$ and $\widehat{\theta}_{\text{fed}}$ be the parameters of the federated router on $\mathcal{D} = \bigcup_{j=1}^N \mathcal{D}_j$. Let $\pi_{\theta,\lambda}$ denote*

[4] We denote the output as a parameterization using θ.

router induced by a parameterization θ for tradeoff parameter λ (Eq. (4)) Then, under standard realizability assumptions [75, 28] we have with probability $1-\delta$,

$$\text{Subopt}\left(\pi_{\widehat{\theta}_i,\lambda}\right) \leq C'\,\Gamma(\mathcal{D}_i)\sqrt{\log\left(\frac{2\,\mathcal{N}_{\mathcal{F}}(1/D_i)}{\delta}\right)},$$

$$\text{Subopt}\left(\pi_{\widehat{\theta}_{\text{fed}},\lambda_i}\right) \leq C'\,\Gamma(\mathcal{D})\sqrt{\log\left(\frac{2\,\mathcal{N}_{\mathcal{F}}(1/D)}{\delta}\right)}.$$

Where $C' = c\max\{1,\lambda\}c_{\max}$ for some universal constant c, $\mathcal{N}_{\mathcal{F}}$ is the covering number [64] of the estimator class and $\Gamma(\mathcal{D}_{in})$ denotes the coverage coefficient of the dataset $\mathcal{D}_{in}$ for the test distribution at client i.

A formal statement of the theorem, along with the relevant definitions and an interpretation of the coverage coefficient, is provided in Appendix J, Theorem J.12. The coverage coefficients of a dataset of size D' decays roughly as $1/\sqrt{D'}$ and the covering number grows as $\log(D')$. Since the federated router optimizes on the dataset $\mathcal{D} = \bigcup_{i=1}^{N}\mathcal{D}_i$ we get a lower suboptimality against local routing $\tilde{\mathcal{O}}(1/\sqrt{D_i})$ vs $\tilde{\mathcal{O}}(1/\sqrt{D})$.

Remark 5.4. Theorem 5.3 extends to each client's test distributions $\mathbb{D}_i^{\text{test}}$ when routers are induced using learned parameters and local tradeoff coefficients λ_i. A stronger version of it is in Appendix J.

5.2 Federated `K-Means-Router`

Theorem 5.5 (Suboptimality-Informal)**.** *Let $\widehat{\theta}$ be the output of Algorithm 2. Under mild smoothness of utility over the embedding space and uniform model logging assumptions, the suboptimality of the router $\pi_{\widehat{\theta},\lambda}$ induced by the output from Algorithm 2 on the global test distribution $\mathbb{D}^{test}$ is upper bounded as*

$$\text{Subopt}\left(\pi_{\widehat{\theta},\lambda}\right) \;\leq\; 2L_\lambda\,\mathbb{E}_{\mathbf{x}\sim\mathbb{D}^{\text{test}}}\big[\|\mathbf{x}-\mu_{\widehat{\theta}}(\mathbf{x})\|_2\big] \;+\; 2L_\lambda\,\Delta_{\max} \;+\; 2B_\lambda\sqrt{\frac{\log\big(\frac{2K_{global}|\mathcal{M}|}{\delta}\big)}{2\,n_{\min}}},$$

where $B_\lambda = 1+\lambda\,c_{\max}$, and the term $\mu_{\widehat{\theta}}(\mathbf{x})$ denotes the embedding of the global cluster center associated with $\mathbf{x}$ using $\widehat{\theta}$. The constant L_λ is the smoothness parameter of the utility in the embedding space and $n_{\min} = \min_{k,m}\ n_{k,m}$, where $n_{k,m}$ is the number of points in global cluster k evaluated on model m in the training dataset. Finally, $\Delta_{\max}$ quantifies the maximum discrepancy across clusters between the training and test distributions.

The formal version of the theorem and its proof are presented in Appendix J, Theorem J.21. Federated training will generally result in a higher $n_{\min}$ as a result of more datapoints $n_{i,k,m} = \sum_i n_{k,m}$. Moreover, better "coverage" of the input embedding space in the federated dataset leads to smaller first term $\mathbb{E}_{\mathbf{x}\sim\mathbb{D}^{\text{test}}}\big[\|\mathbf{x}-\mu_{\widehat{\theta}}(\mathbf{x})\|_2$ in federated routing. Hence, the federated methods incur a lower suboptimality compared to a local `K-Means-Router`.

6 Experiments

We evaluate our federated routers on RouterBench-Data [32] (results are in the main text), ProxRouter-Data [50] and SPROUT dataset [55] (results are in the appendix). RouterBench-Data contains evaluations of 11 public and proprietary LLMs over 8 datasets, ProxRouter-Data contains 14 public LLMs over 12 datasets, and SPROUT dataset contains evaluations of 13 public and proprietary LLMs over 6 datasets. In our experiments, we use all-mpnet-base-v2 by [56] as the query encoder $\text{Enc}(\cdot)$. We also experiment with two other encoder models in Appendix E, and report no significant performance difference in centralized experiments. We detail the experimental settings in Appendix C.

Federated simulation protocol. We simulate a FL setting with $N = 10$ clients and a partial participation rate of 0.6 fraction of clients per round. To induce statistical heterogeneity, we partition RouterBench-Data across clients using Dirichlet distribution with concentration parameter $\alpha = 0.6$ over task labels, following [71]. To reflect realistic client behavior, we assume that each query is evaluated by a single model, i.e., each training sample contains feedback for one $(\text{query}, \text{model})$ pair.

The models observed in client datasets are non-uniformly distributed, so some models receive fewer supervision signals than others within same $\mathcal{D}_i$. Details on the FL setup, a t-SNE visualization for query heterogeneity and a bubble plot for model heterogeneity are in Appendix B.

In the results, we report *accuracy–cost trade-off* curves from sweeping the routing trade-off parameter λ, which controls the relative preference between selecting cheaper models versus higher-quality models. Larger λ prioritizes cost reduction while smaller λ prioritizes accuracy by routing more queries to typically stronger and more expensive models. The cost is reported as the average price across all test queries based on the chosen routed models. To summarize each curve with a scalar, we additionally report normalized area under the curve (AUC), by integrating accuracy as a function of cost over the sweep and normalizing by the cost range. A higher AUC indicates a better frontier.

6.1 Federated Learning Improves Generalization with Better Query Distribution Coverage

In practice, each client typically observes a small number of training queries, and its local query distribution may cover a narrow region of the global query space. As a result, routers trained purely on local data can fail to generalize to the broader population distribution. To quantify this effect, we compare a federated router trained collaboratively across clients against *client-local* (no-FL) routers trained independently on each client, where both are evaluated on a global test set. Figure 2 shows that federated training consistently yields a better accuracy–cost trade-off for both `MLP-Router` and `K-Means-Router`. The gains are particularly pronounced for `K-Means-Router`, since clustering with limited and heterogeneous local samples produces unstable centers and poor global generalization, whereas federated collaboration effectively increases query coverage without requiring central access to raw queries. Across random seeds, the global-test AUCs (mean $\pm$ standard error) are 0.750 ± 0.003 versus 0.683 ± 0.003 for federated versus client-local `MLP-Router`, and 0.747 ± 0.003 versus 0.655 ± 0.011 for `K-Means-Router`, supporting the consistency of these gains (Appendix I). We also train the same router families on the centralized union of client training datasets for comparison. Figure 4 shows closely overlapping frontiers, indicating federated router matches the performance of centralized routers.

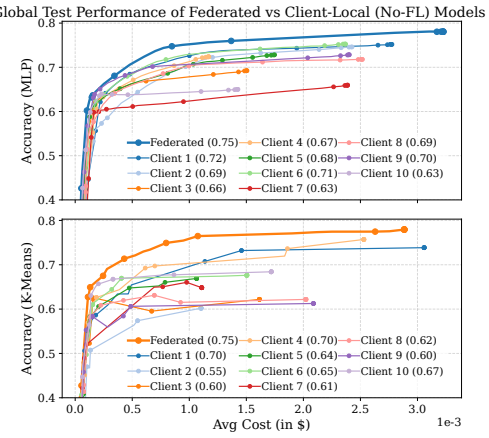


Figure 2: **Federated vs. client-local routers on the global test distribution.** Accuracy–cost curves for `MLP-Router` and `K-Means-Router` by sweeping λ. AUC scores are shown in parentheses.

6.2 FL Even Improves In-distribution Local Performance via Better Model Coverage

To reflect practice, we assume that each training query is evaluated by one model, rather than exhaustively by all models, and the model distribution in a client training set is non-uniform. This sparsity and imbalance implies that a single client may have limited model coverage even for queries drawn from its own distribution, making local routers noisy and biased. Federated learning mitigates this by learning more accurate estimators for models with collaboration across clients. As shown in Figure 3, the federated router improves the accuracy–cost trade-off *even when evaluated on clients' local test set*, indicating that collaboration helps clients learn better routing in-distribution via more effective model coverage. The results for all clients are in Appendix D.3.

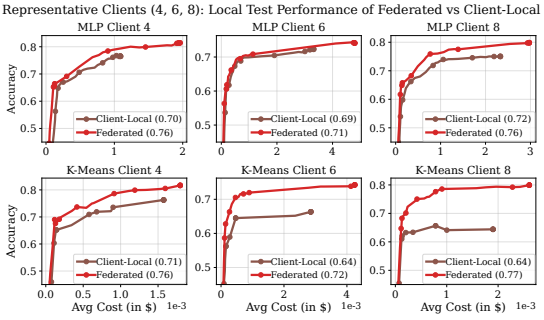


Figure 3: **Federated vs. client-local routers evaluated on local test sets.** Representative clients for `MLP-Router` and `K-Means-Router` are shown. Federated training improves the in-distribution accuracy–cost frontier via better effective model coverage.

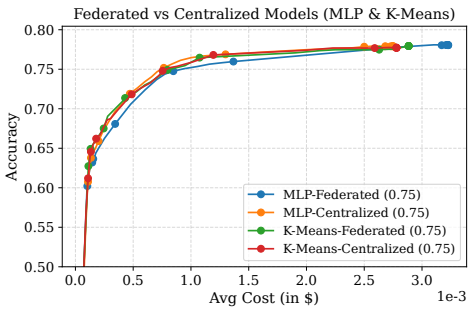


Figure 4: **Federated vs. centralized routers.** Centralized routers use pooled client training data. Global-test frontiers closely overlap; parentheses denote normalized AUC.

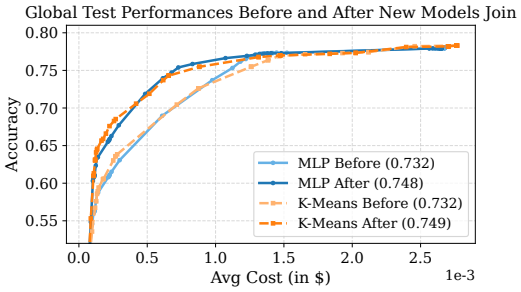


Figure 5: **Adapting to new models.** Global test frontiers for routers trained with three models withheld and then incorporated via a lightweight calibration step.

6.3 When New Models are Added to the Pool

In deployed systems, the set of available LLMs is not static since new models are introduced every day, requiring routers to support *cheap* extension without full retraining [26, 35]. Both `MLP-Router` and `K-Means-Router` naturally support such expansions. For `MLP-Router`, to add a new model, we append a new head and train only that head by keeping the trunk and existing heads frozen. For `K-Means-Router`, since the query embedding space is unchanged, the introduction of a new model reduces to estimating its accuracy & cost statistics over existing regions of the embedding space. We simulate a scenario where three models are withheld during initial router training and introduced later. Upon introduction, each client evaluates the new models on a small calibration subset (10% of its prompts) to estimate the required per-model statistics (for `K-Means-Router`) or to train the newly added heads (for `MLP-Router`). Figure 5 shows the accuracy–cost trade-off before and after expansion on the global test distribution. Both routers benefit from the enlarged model pool after a lightweight adaptation without re-training the full router.

6.4 Adaptive Personalization for High Heterogeneity

Under extreme client data heterogeneity, a federated router can underperform on some clients' local distributions, even on test sets. For instance, if a client mostly has biology prompts while other clients rarely observe it, the federated router may be poorly trained for that client, making a local router preferable despite the reduced model-coverage it suffers from. To address this, we propose a lightweight *adaptive personalization* that interpolates between the federated router and the client's locally trained router based on their empirical calibration errors on the client's existing data points. Assume each client i maintains the global federated ($\mathsf{A}^{(m)}$, $\mathsf{C}^{(m)}$) and locally trained accuracy and cost estimators ($\mathsf{A}^{(i,m)}$, $\mathsf{C}^{(i,m)}$).

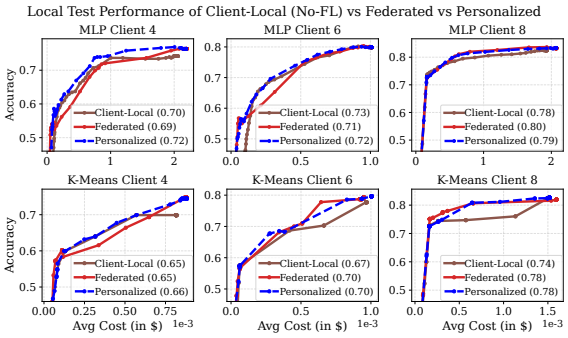


Figure 6: **Adaptive personalization under extreme heterogeneity.** Evaluated on local test sets of clients with federated training, isolated local training, and the proposed adaptive mixture.

After training, client i calculates the mean absolute error (denoted as operator $e(\cdot)$) of its training samples (no additional model calls) for the accuracy and cost of each model for both routers. Then, at inference time, for every model $m \in \mathcal{M}$, the client calculates the weights of the federated and local estimators inversely proportional to their errors separately for accuracy and cost,

$$w_a^{(i,m)} = \frac{e(\mathsf{A}^{(m)})}{e(\mathsf{A}^{(m)}) + e(\mathsf{A}^{(i,m)})}, \quad w_c^{(i,m)} = \frac{e(\mathsf{C}^{(m)})}{e(\mathsf{C}^{(m)}) + e(\mathsf{C}^{(i,m)})}.$$

Then, they weight estimates of local and federated estimators for accuracy by $w_a^{(i,m)}\mathsf{A}^{(i,m)} + (1 - w_a^{(i,m)})\mathsf{A}^{(m)}$, and cost by $w_c^{(i,m)}\mathsf{C}^{(i,m)} + (1 - w_c^{(i,m)})\mathsf{C}^{(m)}$ to plug in the routing rule in Eq. (1) in the inference time. Relative local errors serve as a heuristic aggregation signal for this personalized extension targeting settings with extreme heterogeneity. We do not assume that training error predicts test AUC.

We observe that reusing the same training data points for this calibration does not result in underperformance in our experiments.

We repeat the main experiment under extreme heterogeneity, using Dirichlet parameter $\alpha = 0.03$. Figure 6 reports local-test accuracy–cost frontiers for representative clients, comparing federated routers, local routers, and our adaptive personalization. Under this extreme heterogeneity, federated `MLP-Router` can underperform local routers for some clients. In contrast, adaptive personalization generally matches or improves upon both by combining broader model coverage with client-specific correction. For `K-Means-Router`, federated routing remains competitive with local routers. We attribute this to the robustness of K-means under heterogeneity [21], while local `K-Means-Router` suffers under low coverage since it estimates per-centroid, per-model statistics from limited client data. Additional experimental results are in Appendix D. Appendix D.4 discusses the adaptation of new clients joining the system after initial training. We also compare with a federated adaptation of EmbedLLM [82], a recent centralized LLM routing method, in Appendix H, showing that FL benefits another router architecture beyond our proposed `MLP-Router` and `K-Means-Router` designs. Ablations in Appendices D.1 and D.2 show that federated router's gains persist across settings with different number of clients and query heterogeneity. Additional experiments on SPROUT dataset [55], where the accuracy signal is graded quality of the model response, are presented in Appendix G.

7 Conclusion

To the best of our knowledge, this is the first work to tackle *sparse* and *decentralized* query–model evaluation data for training language query routers. We examine the practical scenario where prompts and evaluations are local to clients, and query–model coverage is incomplete and imbalanced across the model pool. We propose federated training as a solution across both canonical routing paradigms: a parametric `MLP-Router` and a nonparametric `K-Means-Router`, enabling collaborative learning from clients' offline data. Empirically, federated routing improves the accuracy–cost frontier over client-local baselines on both global and client-level test distributions, and adaptive personalization further improves robustness under extreme heterogeneity. We validate our empirical observations through theoretical results characterizing how federation reduces routing suboptimality compared to client-local routers. Overall, federated learning offers a practical foundation for training LLM routers from privacy-sensitive, fragmented data. Online routing, where the router is updated after deployment with new query evaluations, remains a promising direction for future work.

Acknowledgements

This work was partially supported by the US Department of Energy under grant DESC0025652 and the US National Science Foundation under grants CNS-2409138 and CNS-2533813 to CJW. The work was also partially supported by the AI2C Seed grant and the US National Science Foundation under grants CCF 2045694, CNS-2112471, CPS-2111751, and ONR N00014-23-1-2149 to GJ. This work used PSC Bridges-2 GPU at Pittsburgh Supercomputing Center through allocation CIS250087 from the Advanced Cyberinfrastructure Coordination Ecosystem: Services & Support (ACCESS) program, which is supported by US National Science Foundation grants #2138259, #2138286, #2138307, #2137603, and #2138296.

References

[1] Pranjal Aggarwal, Aman Madaan, Ankit Anand, Srividya Pranavi Potharaju, Swaroop Mishra, Pei Zhou, Aditya Gupta, Dheeraj Rajagopal, Karthik Kappaganthu, Yiming Yang, Shyam Upadhyay, Manaal Faruqui, and Mausam. Automix: Automatically mixing language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL `https://openreview.net/forum?id=e6WrwIvgzX`.

[2] Janice Ahn, Rishu Verma, Renze Lou, Di Liu, Rui Zhang, and Wenpeng Yin. Large language models for mathematical reasoning: Progresses and challenges. In Neele Falk, Sara Papi, and Mike Zhang, editors, *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 225–237, St. Julian's, Malta, March 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.eacl-srw.17.

[3] Simran Arora, Patrick Lewis, Angela Fan, Jacob Kahn, and Christopher Ré. Reasoning over public and private data in retrieval-based systems. *Transactions of the Association for Computational Linguistics*, 11:902–921, 2023.

[4] Artificial Analysis. Artificial analysis: Ai model & api providers analysis, 2025. URL https://artificialanalysis.ai/.

[5] Rishi Bommasani et al. On the opportunities and risks of foundation models. *ArXiv*, 2021. URL https://crfm.stanford.edu/assets/report.pdf.

[6] Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. Practical secure aggregation for federated learning on user-held data. *arXiv preprint arXiv:1611.04482*, 2016.

[7] Keith Bonawitz, Hubert Eichner, Wolfgang Grieskamp, Dzmitry Huba, Alex Ingerman, Vladimir Ivanov, Chloe Kiddon, Jakub Konečnỳ, Stefano Mazzocchi, Brendan McMahan, et al. Towards federated learning at scale: System design. *Proceedings of machine learning and systems*, 1: 374–388, 2019.

[8] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

[9] Erik Brynjolfsson, Danielle Li, and Lindsey Raymond. Generative ai at work. *The Quarterly Journal of Economics*, 140(2):889–942, 2025.

[10] Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Brian Strope, and Ray Kurzweil. Universal sentence encoder for English. In Eduardo Blanco and Wei Lu, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 169–174, Brussels, Belgium, November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-2029. URL https://aclanthology.org/D18-2029/.

[11] Chaochao Chen, Xiaohua Feng, Yuyuan Li, Lingjuan Lyu, Jun Zhou, Xiaolin Zheng, and Jianwei Yin. Integration of large language models and federated learning. *Patterns*, 5(12), 2024.

[12] Lingjiao Chen, Matei Zaharia, and James Zou. FrugalGPT: How to use large language models while reducing cost and improving performance. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=cSimKw5p6R. Featured Certification.

[13] Shuhao Chen, Weisen Jiang, Baijiong Lin, James Kwok, and Yu Zhang. Routerdc: Query-based router by dual contrastive learning for assembling large language models. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 66305–66328. Curran Associates, Inc., 2024. doi: 10.52202/079017-2120. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/7a641b8ec86162fc875fb9f6456a542f-Paper-Conference.pdf.

[14] Zhou Chen, Zhiqiang Wei, Yuqi Bai, Xue Xiong, and Jianmin Wu. TagRouter: Learning route to LLMs through tags for open-domain text generation tasks. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Findings of the Association for Computational Linguistics: ACL 2025*, pages 21539–21564, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.1110. URL https://aclanthology.org/2025.findings-acl.1110/.

[15] Yujun Cheng, Weiting Zhang, Zhewei Zhang, Chuan Zhang, Shengjin Wang, and Shiwen Mao. Towards federated large language models: Motivations, methods, and future directions. *IEEE Communications Surveys & Tutorials*, 2024.

[16] Sinho Chewi, Jonathan Niles-Weed, and Philippe Rigollet. Statistical optimal transport. *arXiv preprint arXiv:2407.18163*, 3, 2024.

[17] Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios Nikolas Angelopoulos, Tianle Li, Dacheng Li, Banghua Zhu, Hao Zhang, Michael Jordan, Joseph E. Gonzalez, and Ion Stoica. Chatbot arena: An open platform for evaluating LLMs by human preference. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=3MW8GKNyzI.

[18] Yae Jee Cho, Jianyu Wang, and Gauri Joshi. Client selection in federated learning: Convergence analysis and power-of-choice selection strategies. *CoRR*, abs/2010.01243, 2020. URL https://arxiv.org/abs/2010.01243.

[19] Fatemeh Dehghani, Roya Dehghani, Yazdan Naderzadeh Ardebili, and Shahryar Rahnamayan. Large language models in legal systems: A survey. *Humanities and Social Sciences Communications*, 12(1):1977, 2025.

[20] Jasper Dekoninck, Maximilian Baader, and Martin Vechev. A unified approach to routing and cascading for LLMs, 2025. URL https://openreview.net/forum?id=rgDwRdMwoS.

[21] Don Kurian Dennis, Tian Li, and Virginia Smith. Heterogeneity for the win: One-shot federated clustering. In *International conference on machine learning*, pages 2611–2620. PMLR, 2021.

[22] Dujian Ding, Ankur Mallick, Chi Wang, Robert Sim, Subhabrata Mukherjee, Victor Rühle, Laks V. S. Lakshmanan, and Ahmed Hassan Awadallah. Hybrid LLM: Cost-efficient and quality-aware query routing. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=02f3mUtqnM.

[23] Dujian Ding, Ankur Mallick, Shaokun Zhang, Chi Wang, Daniel Madrigal, Mirian Del Carmen Hipolito Garcia, Menglin Xia, Laks V. S. Lakshmanan, Qingyun Wu, and Victor Rühle. BEST-route: Adaptive LLM routing with test-time optimal compute. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=tFBIbCVXkG.

[24] David Dohan, Winnie Xu, Aitor Lewkowycz, Jacob Austin, David Bieber, Raphael Gontijo Lopes, Yuhuai Wu, Henryk Michalewski, Rif A. Saurous, Jascha Sohl-dickstein, Kevin Murphy, and Charles Sutton. Language model cascades, 2022. URL https://arxiv.org/abs/2207.10342.

[25] Benj Edwards. Exponential growth brews 1 million ai models on hugging face. *Ars Technica*, September 2024. URL https://arstechnica.com/information-technology/2024/09/ai-hosting-platform-surpasses-1-million-models-for-the-first-time/. Hugging Face surpassed 1 million AI model listings for the first time.

[26] Tao Feng, Yanzhen Shen, and Jiaxuan You. Graphrouter: A graph-based router for LLM selections. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=eU39PDsZtT.

[27] Evelyn Fix and J. L. Hodges. Discriminatory analysis. nonparametric discrimination: Consistency properties. *International Statistical Review / Revue Internationale de Statistique*, 57(3): 238–247, 1989. ISSN 03067734, 17515823. URL http://www.jstor.org/stable/1403797.

[28] Dylan J. Foster and Alexander Rakhlin. Foundations of reinforcement learning and interactive decision making, 2023. URL https://arxiv.org/abs/2312.16730.

[29] Evan Frick, Connor Chen, Joseph Tennyson, Tianle Li, Wei-Lin Chiang, Anastasios Nikolas Angelopoulos, and Ion Stoica. Prompt-to-leaderboard: Prompt-adaptive LLM evaluations. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=7VPRrzFEN8.

[30] Neel Guha, Mayee F Chen, Trevor Chow, Ishan S. Khare, and Christopher Re. Smoothie: Label free language model routing. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=pPSWHsgqRp.

[31] J. A. Hartigan and M. A. Wong. Algorithm as 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1):100–108, 1979. ISSN 00359254, 14679876. URL http://www.jstor.org/stable/2346830.

[32] 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. In *Agentic Markets Workshop at ICML 2024*, 2024. URL https://openreview.net/forum?id=IVXmV8Uxwh.

[33] Zhongzhan Huang, Guoming Ling, Yupei Lin, Yandong Chen, Shanshan Zhong, Hefeng Wu, and Liang Lin. RouterEval: A comprehensive benchmark for routing LLMs to explore model-level scaling up in LLMs. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng, editors, *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 3860–3887, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-335-7. doi: 10.18653/v1/2025.findings-emnlp.208. URL https://aclanthology.org/2025.findings-emnlp.208/.

[34] Divyansh Jhunjhunwala, Pranay Sharma, Aushim Nagarkatti, and Gauri Joshi. Fedvarp: Tackling the variance due to partial client participation in federated learning. In *Uncertainty in Artificial Intelligence*, pages 906–916. PMLR, 2022.

[35] 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 LLM routing with correctness-based representation. In *First Workshop on Scalable Optimization for Efficient and Adaptive Foundation Models*, 2025. URL https://openreview.net/forum?id=QpOCijgaBE.

[36] Peter Kairouz et al. Advances and open problems in federated learning. *Found. Trends Mach. Learn.*, 14(1–2):1–210, June 2021. ISSN 1935-8237. doi: 10.1561/2200000083. URL https://doi.org/10.1561/2200000083.

[37] Anastasia Koloskova, Nicolas Loizou, Sadra Boreiri, Martin Jaggi, and Sebastian Stich. A unified theory of decentralized sgd with changing topology and local updates. In *International conference on machine learning*, pages 5381–5393. PMLR, 2020.

[38] Yaniv Leviathan, Matan Kalman, and Yossi Matias. Fast inference from transformers via speculative decoding, 2023. URL https://arxiv.org/abs/2211.17192.

[39] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. *Proceedings of Machine learning and systems*, 2:429–450, 2020.

[40] Yang Li. Llm bandit: Cost-efficient llm generation via preference-conditioned dynamic routing, 2025. URL https://arxiv.org/abs/2502.02743.

[41] Yang Li. Rethinking predictive modeling for llm routing: When simple knn beats complex learned routers, 2025. URL https://arxiv.org/abs/2505.12601.

[42] Bill Yuchen Lin, Chaoyang He, Zihang Ze, Hulin Wang, Yufen Hua, Christophe Dupuy, Rahul Gupta, Mahdi Soltanolkotabi, Xiang Ren, and Salman Avestimehr. Fednlp: Benchmarking federated learning methods for natural language processing tasks. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 157–175, 2022.

[43] S. Lloyd. Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28(2): 129–137, 1982. doi: 10.1109/TIT.1982.1056489.

[44] Keming Lu, Hongyi Yuan, Runji Lin, Junyang Lin, Zheng Yuan, Chang Zhou, and Jingren Zhou. Routing to the expert: Efficient reward-guided ensemble of large language models. In Kevin Duh, Helena Gomez, and Steven Bethard, editors, *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 1964–1974, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-long.109. URL https://aclanthology.org/2024.naacl-long.109/.

[45] Akash Maharaj, Kun Qian, Uttaran Bhattacharya, Sally Fang, Horia Galatanu, Manas Garg, Rachel Hanessian, Nishant Kapoor, Ken Russell, Shivakumar Vaithyanathan, et al. Evaluation and continual improvement for an enterprise ai assistant. In *Proceedings of the Fifth Workshop on Data Science with Human-in-the-Loop (DaSH 2024)*, pages 17–24, 2024.

[46] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017.

[47] Justin K Miller and Wenjia Tang. Evaluating llm metrics through real-world capabilities, 2025. URL https://arxiv.org/abs/2505.08253.

[48] 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*, 2025. URL https://openreview.net/forum?id=8sSqNntaMr.

[49] OpenAI. Gpt-4 technical report, 2024. URL https://arxiv.org/abs/2303.08774.

[50] Shivam Patel, Neharika Jali, Ankur Mallick, and Gauri Joshi. Proxrouter: Proximity-weighted llm query routing for improved robustness to outliers, 2025. URL https://arxiv.org/abs/2510.09852.

[51] Manhin Poon, XiangXiang Dai, Xutong Liu, Fang Kong, John C. S. Lui, and Jinhang Zuo. Online multi-llm selection via contextual bandits under unstructured context evolution, 2025. URL https://arxiv.org/abs/2506.17670.

[52] Arian Raje, Baris Askin, Divyansh Jhunjhunwala, and Gauri Joshi. Ravan: Multi-head low-rank adaptation for federated fine-tuning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=gyn4n8oC9B.

[53] Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1410. URL https://aclanthology.org/D19-1410/.

[54] Marija Sakota, Maxime Peyrard, and Robert West. Fly-swat or cannon? cost-effective language model choice via meta-modeling. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, WSDM '24, page 606–615. ACM, March 2024. doi: 10.1145/3616855.3635825. URL http://dx.doi.org/10.1145/3616855.3635825.

[55] 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. In *ICLR 2025 Workshop on Foundation Models in the Wild*, 2025. URL https://openreview.net/forum?id=xEBOy2ze1U.

[56] Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. Mpnet: Masked and permuted pre-training for language understanding, 2020. URL https://arxiv.org/abs/2004.09297.

[57] Wei Song, Zhenya Huang, Cheng Cheng, Weibo Gao, Bihan Xu, GuanHao Zhao, Fei Wang, and Runze Wu. IRT-router: Effective and interpretable multi-LLM routing via item response theory. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15629–15644, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.761. URL https://aclanthology.org/2025.acl-long.761/.

[58] 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, 2024. URL https://arxiv.org/abs/2408.12320.

[59] Youbang Sun, Zitao Li, Yaliang Li, and Bolin Ding. Improving loRA in privacy-preserving federated learning. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=NLPzL6HWNl.

[60] Gemini Team. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

[61] Asterios Tsiourvas, Wei Sun, and Georgia Perakis. Causal LLM routing: End-to-end regret minimization from observational data. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=iZC5xoQQkX.

[62] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008. URL http://jmlr.org/papers/v9/vandermaaten08a.html.

[63] Daniel Vila. How to choose the best open source llm for your project in 2025, September 2025. URL https://huggingface.co/blog/dvilasuero/choosing-best-open-source-ai-models? The Hugging Face Hub hosts over 2 million public models as of late 2025.

[64] Martin J Wainwright. *High-dimensional statistics: A non-asymptotic viewpoint*, volume 48. Cambridge university press, 2019.

[65] Jianyu Wang, Qinghua Liu, Hao Liang, Gauri Joshi, and H Vincent Poor. Tackling the objective inconsistency problem in heterogeneous federated optimization. *Advances in neural information processing systems*, 33:7611–7623, 2020.

[66] Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 5776–5788. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[67] Ziyao Wang, Zheyu Shen, Yexiao He, Guoheng Sun, Hongyi Wang, Lingjuan Lyu, and Ang Li. Flora: Federated fine-tuning large language models with heterogeneous low-rank adaptations. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 22513–22533. Curran Associates, Inc., 2024. doi: 10.52202/079017-0708.

[68] Herbert Woisetschläger, Ryan Zhang, Shiqiang Wang, and Hans Arno Jacobsen. Mess+: Dynamically learned inference-time llm routing in model zoos with service level guarantees. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[69] Cong Xie, Sanmi Koyejo, and Indranil Gupta. Asynchronous federated optimization. *arXiv preprint arXiv:1903.03934*, 2019.

[70] Yuhang Yao, Jianyi Zhang, Junda Wu, Chengkai Huang, Yu Xia, Tong Yu, Ruiyi Zhang, Sungchul Kim, Ryan Rossi, Ang Li, Lina Yao, Julian McAuley, Yiran Chen, and Carlee Joe-Wong. Federated large language models: Current progress and future directions, 2025. URL https://arxiv.org/abs/2409.15723.

[71] Mikhail Yurochkin, Mayank Agarwal, Soumya Ghosh, Kristjan Greenewald, Nghia Hoang, and Yasaman Khazaeni. Bayesian nonparametric federated learning of neural networks. In *International conference on machine learning*, pages 7252–7261. PMLR, 2019.

[72] Wenhao Zhan, Masatoshi Uehara, Nathan Kallus, Jason D Lee, and Wen Sun. Provable offline preference-based reinforcement learning. In *The Twelfth International Conference on Learning Representations*, 2024.

[73] Haozhen Zhang, Tao Feng, and Jiaxuan You. Router-r1: Teaching LLMs multi-round routing and aggregation via reinforcement learning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=DWf4vroKWJ.

[74] Jianyi Zhang, Saeed Vahidian, Martin Kuo, Chunyuan Li, Ruiyi Zhang, Tong Yu, Guoyin Wang, and Yiran Chen. Towards building the federatedgpt: Federated instruction tuning. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6915–6919. IEEE, 2024.

[75] Tong Zhang. *Mathematical analysis of machine learning algorithms*. Cambridge University Press, 2023.

[76] Xuechen Zhang, Zijian Huang, Ege Onur Taga, Carlee Joe-Wong, Samet Oymak, and Jiasi Chen. Treacle: Thrifty reasoning via context-aware llm and prompt selection. *CoRR*, abs/2404.13082, 2024. URL https://doi.org/10.48550/arXiv.2404.13082.

[77] Yiqun Zhang, Hao Li, Jianhao Chen, Hangfan Zhang, Peng Ye, Lei Bai, and Shuyue Hu. Beyond gpt-5: Making llms cheaper and better via performance-efficiency optimized routing, 2025. URL https://arxiv.org/abs/2508.12631.

[78] Yiqun Zhang, Hao Li, Chenxu Wang, Linyao Chen, Qiaosheng Zhang, Peng Ye, Shi Feng, Daling Wang, Zhen Wang, Xinrun Wang, Jia Xu, Lei Bai, Wanli Ouyang, and Shuyue Hu. The avengers: A simple recipe for uniting smaller language models to challenge proprietary giants, 2025. URL https://arxiv.org/abs/2505.19797.

[79] Kai Zhao, Zhaohui Yang, Chongwen Huang, Xiaoming Chen, and Zhaoyang Zhang. Fedsllm: federated split learning for large language models over communication networks. In *2024 International Conference on Ubiquitous Communication (Ucom)*, pages 438–443. IEEE, 2024.

[80] Zesen Zhao, Shuowei Jin, and Z. Morley Mao. Eagle: Efficient training-free router for multi-llm inference, 2024. URL https://arxiv.org/abs/2409.15518.

[81] Yue Zheng, Yuhao Chen, Bin Qian, Xiufang Shi, Yuanchao Shu, and Jiming Chen. A review on edge large language models: Design, execution, and applications. *ACM Comput. Surv.*, 57(8), March 2025. ISSN 0360-0300. doi: 10.1145/3719664. URL https://doi.org/10.1145/3719664.

[82] Richard Zhuang, Tianhao Wu, Zhaojin Wen, Andrew Li, Jiantao Jiao, and Kannan Ramchandran. EmbedLLM: Learning compact representations of large language models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=Fs9EabmQrJ.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. In particular, we study federated training of LLM routers (routing policies) to reduce the cost of LLM use while preserving client privacy by keeping prompts and model-evaluation signals decentralized. There are many potential societal consequences of our work, none of which we feel must be specifically highlighted here.

A Extended Related Work

A.1 Federated Learning and LLMs

Federated learning (FL) studies how to train a shared model from decentralized data while keeping raw client data on-device [46]. In practice, FL must cope with *statistical heterogeneity* (client data are non-iid) and tight communication and compute budgets [65]. These challenges have motivated a rich set of optimization methods, including proximal regularization [39] and variance-reduction techniques [34]. At scale, practical deployments also rely on systems mechanisms such as partial participation, asynchronous aggregation, secure aggregation, and failure tolerance [69, 7, 6]. Overall, most FL work focuses on collaboratively training *a single* global model (or personalized variants) under decentralized data and resource constraints.

LLMs have advanced rapidly in scale and capability, making them a default backbone for many language-centric applications. However, bringing LLMs into FL is challenging because naïvely communicating or fine-tuning billions of parameters is prohibitive for many clients. As a result, a major thread in "federated LLM" research centers on parameter-efficient adaptation (e.g., adapter- or LoRA-style methods) and quantization-aware fine-tuning to reduce memory and communication, alongside benchmarks and frameworks for federated NLP/LLM evaluation [67, 52, 42]. Recent methods study federated instruction tuning under heterogeneous client resources and tasks [74, 15, 11], as well as hybrid paradigms such as federated split learning that offload parts of the model when clients cannot host the full network [79]. Notably, these efforts largely assume a single deployed global model, whereas we target the complementary problem of learning *federated routing policies* that select among a pool of LLMs without centralizing client prompts. This distinction is crucial because high-quality routing typically requires access to private and broad query–model evaluations, which are naturally fragmented across clients and often cannot be uploaded to a central server. By leveraging FL, clients can collaboratively learn a router that generalizes across diverse prompt distributions and improves model-coverage across the pool, while keeping prompt data local. *To our knowledge, this work is the first to study LLM routing in a federated setting*.

A.2 LLM Query Routing

Language query routing selects a suitable model for generating response to a query based on relative importance on performance vs cost. This requires capturing query tasks through a convenient representation, usually obtained from sentence encoders [10, 53]. Parametric routers estimate suitability scores or correctness probabilities of language models for given queries using trainable neural networks or scoring functions [32, 22, 23, 82, 54, 33, 44, 76]. Beyond direct performance and cost prediction, routing is also formulated for estimating relative model quality or human judgments [48, 29, 17], and via auxiliary signals such as task tags or learned difficulty estimators [14, 76].

An alternate direction of router design avoids learning a dedicated routing network and instead makes predictions using the geometry of the query embedding space and empirical performance statistics therein. Common versions of such nonparametric routers include partitioning the query space (e.g., clustering) and assigning models based on cluster-level summaries, or selecting models based on local neighborhoods under similarity search [32, 35, 58, 41, 78, 77, 50]. These designs are often attractive in settings where the model pool evolves, since additional queries or new models can be incorporated by updating clusters or query embedding pool rather than retraining from scratch.

Routing is also closely related to broader multi-model inference strategies that allocate computation adaptively rather than committing to a single model for generating a response. Cascading and escalation frameworks route queries through sequences of models, using intermediate performance signals to decide when to stop or when to route to a stronger model [24, 12, 20]. Other strategies

exploit multiple parallel candidate generations, such as best-of-n sampling [23], or reduce generation cost via speculative decoding, where a small model proposes tokens and a larger model verifies them [38]. These techniques are largely orthogonal to routing policies and can be composed with them.

Finally, several works examine routing as an online decision problem under explicit resource constraints. Bandit and budgeted-selection formulations make model choices based on sequential evaluations of model responses, seeking improved utility while respecting cost constraints [40, 76, 51], and related solutions also include POMDP-based mixtures of models [1]. Separately, interpretability and structure in routing signals has been demonstrated through model-comparison scores such as Elo [80], psychometric inspired formulations such as Item Response Theory [57], and weakly supervised or metric learning based approaches [30, 26, 13].

B Details on Datasets and Federated Setup

Details on the task, prompting techniques and evaluation scheme for RouterBench-Data and ProxRouter-Data are elaborated in Hu et al. [32] and Patel et al. [50], respectively. The quality labels in RouterBench-Data combine exact-match and LLM-based evaluations, whereas ProxRouter-Data evaluates numerical and multiple-choice answers against benchmark targets. More generally, $\mathrm{acc}(\mathbf{x}, m)$ can be considered as a representation of expected scalar feedback such as user approval or task success.

B.1 Query Heterogeneity.

To simulate heterogeneity, we split the samples of RouterBench-Data and ProxRouter-Data across clients following Dirichlet distribution over subtasks [71] with α parameters of 0.6 and 0.4, respectively.

We provide visualization of the induced client heterogeneity used in our experiments. To visualize the query distribution, we compute a single 2D t-SNE embedding [62] on the full dataset (using the same query representations for all plots), and then visualize each client's local subset by highlighting the corresponding points in the shared embedding. Figure 7 reports the global distribution alongside client-level views, illustrating the non-IID partitions induced by the Dirichlet split ($\alpha = 0.6$ in Section 6). Similarly, analogous t-SNE visualization for ProxRouter-Data (Dirichlet with $\alpha = 0.4$) is seen in Figure 8.

B.2 Model Heterogeneity

In our experiments, we assume only a single model evaluation per query is logged in the training sets. To simulate heterogeneous model assignments across clients, for each client i, we first sample a client-specific distribution over M models with the Dirichlet distribution with $\alpha = 0.45$ for RouterBench-Data. Then, for each local sample $j \in [|\mathcal{D}_i|]$, we assign a model index by sampling with probabilities of the client-specific distribution. This procedure induces client-dependent, potentially highly non-uniform model proportions within $\mathcal{D}_i$. For RouterBench-Data, Figure 9 shows the bubble plot of the proportions of models in the client datasets. For ProxRouter-Data, we use uniform model logging for variety in experiments.

C Details on Experimental Setting

We provide additional details for main text experiments here.

Client simulation and train/test split. We simulate $N = 10$ clients. Within each client, we split its allocated data into local train/test with fractions $0.75/0.25$, and the *global* train/test sets are defined as the unions of client train/test splits.

Evaluation protocol. For acc–cost tradeoff curves, we sweep the λ parameter on a log grid $\lambda \in [10^{-2}, 10^{7}]$ with 100 points and report the resulting curves/AUC.

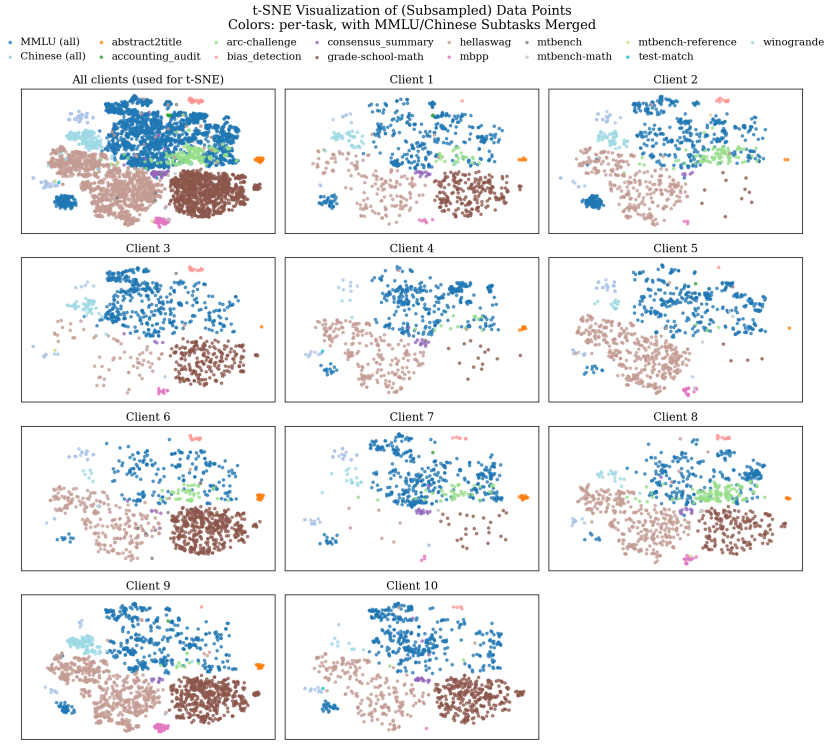


Figure 7: **t-SNE visualization of query distributions.** We run t-SNE once on the full dataset and reuse the same 2D embedding for all panels. The grid highlights each client's local subset, illustrating (subsampled) the resulting heterogeneity across clients.

C.1 Federated `MLP-Router`

Model. The router is an MLP with a shared trunk of two hidden layers of widths $(512, 512)$, each followed by LayerNorm, GELU, and dropout ($p = 0.1$), and with per-model heads that predict (i) an accuracy logit (sigmoid at inference) and (ii) a normalized cost scalar.

Optimization and FL hyperparameters. We train with FedAvg. The local optimizer is AdamW with learning rate $\eta = 10^{-3}$ and weight decay $3 \cdot 10^{-4}$. Each communication round performs 1 local epoch (over full local train data) per participating client (mini-batch size 128; gradient clipping with max-norm 1.0). The router contains approximately 670K parameters, corresponding to a one-way model communication of 2.68 MB per active client per round.

C.2 Federated K-Means Router

Router and clustering hyperparameters. This router is training-free (no learning rate). Each client runs Lloyd's K-means [43] in embedding space using Euclidean distance. We use $K_{\text{local}} = 15$ clusters per client and $K_{\text{global}} = 20$ clusters at the server. Both local and global K-means use $n_{\text{init}} = 3$ random restarts and at most 30 iterations. We treat the overall procedure as a single federated clustering stage ($T = 1$), consisting of (i) clients $\rightarrow$ server upload of local centroids (with sizes), (ii) server-side clustering into K_{global} centers, and (iii) clients $\rightarrow$ server aggregation of per-cluster accuracy/cost statistics for routing. The communication cost in our experiment is approximately 0.5 MB.

D RouterBench-Data Experiments not Presented in the Main Text due to the Space Limit

In this section, we present the remaining main experimental results using RouterBench-Data.

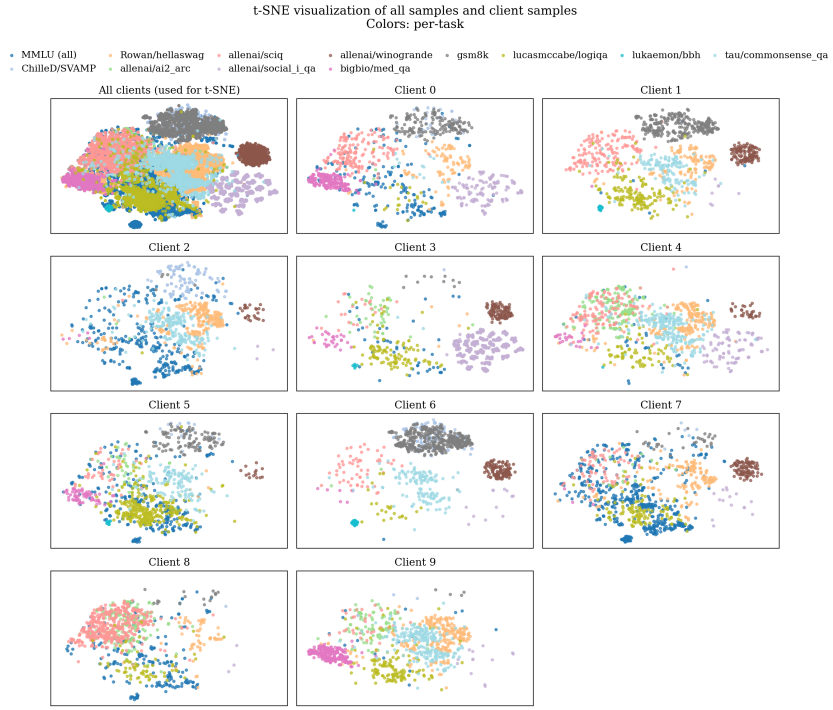


Figure 8: **t-SNE visualization of query distributions for ProxRouter-Data.**



Figure 9: **Client-specific model heterogeneity for RouterBench-Data.** Bubble plot of the per-model proportions in each client local dataset $\mathcal{D}_i$ under the Dirichlet assignment scheme ($\alpha = 0.45$).

D.1 Sensitivity to the Number of Clients

We repeat the main RouterBench-Data experiment with $N \in \{5, 10, 20, 40\}$ clients, keeping the total dataset size and participation fraction fixed. Table 1 reports global-test normalized AUCs across random seeds. Federated performance remains stable as N increases, whereas client-local performance generally decreases as the available training data are divided among more clients. FL thus preserves collective query and model coverage despite reduced local sample sizes.

D.2 Sensitivity to Query Heterogeneity

We vary the Dirichlet concentration parameter $\alpha \in \{0.03, 0.10, 0.30, 0.60, 5.00\}$ in the main RouterBench-Data setting, where smaller α induces greater heterogeneity in client query distributions. As shown in Table 2, both federated routers remain robust across the sweep and consistently outperform client-local training. The gains are particularly significant at $\alpha = 0.03$, where individual clients have highly restricted query coverage.

Table 1: Sensitivity to the number of clients on RouterBench-Data. Global-test normalized AUCs are reported as mean ± standard error across seeds; client-local results are averaged over clients.

	`MLP-Router`		`K-Means-Router`	
N	Federated	Client-local	Federated	Client-local
5	0.747 ± 0.004	0.679 ± 0.010	0.733 ± 0.007	0.656 ± 0.023
10	0.750 ± 0.005	0.679 ± 0.008	0.742 ± 0.007	0.646 ± 0.018
20	0.750 ± 0.003	0.661 ± 0.007	0.745 ± 0.003	0.601 ± 0.011
40	0.749 ± 0.009	0.652 ± 0.003	0.749 ± 0.005	0.584 ± 0.008

Table 2: Sensitivity to query heterogeneity on RouterBench-Data. Global-test normalized AUCs are reported as mean ± standard error across seeds; client-local results are averaged over clients.

	`MLP-Router`		`K-Means-Router`	
α	Federated	Client-local	Federated	Client-local
0.03	0.742 ± 0.007	0.640 ± 0.015	0.731 ± 0.007	0.601 ± 0.033
0.10	0.743 ± 0.012	0.666 ± 0.010	0.742 ± 0.005	0.614 ± 0.014
0.30	0.744 ± 0.012	0.669 ± 0.010	0.734 ± 0.015	0.619 ± 0.014
0.60	0.746 ± 0.015	0.672 ± 0.011	0.741 ± 0.008	0.650 ± 0.018
5.00	0.748 ± 0.012	0.669 ± 0.013	0.749 ± 0.004	0.625 ± 0.015

D.3 The Local Test Results for Federated Models vs. Locally Trained Models on All Clients

We report the complete set of local test results across all clients. For each client, we compare the federated router trained collaboratively across clients against a client-local (no-FL) router trained using that client's data, where evaluation is performed on the same client's local test distribution. Figures 10 and 11 show the full grids for `MLP-Router` and `K-Means-Router`, respectively, confirming the consistent improvements discussed in the main text.

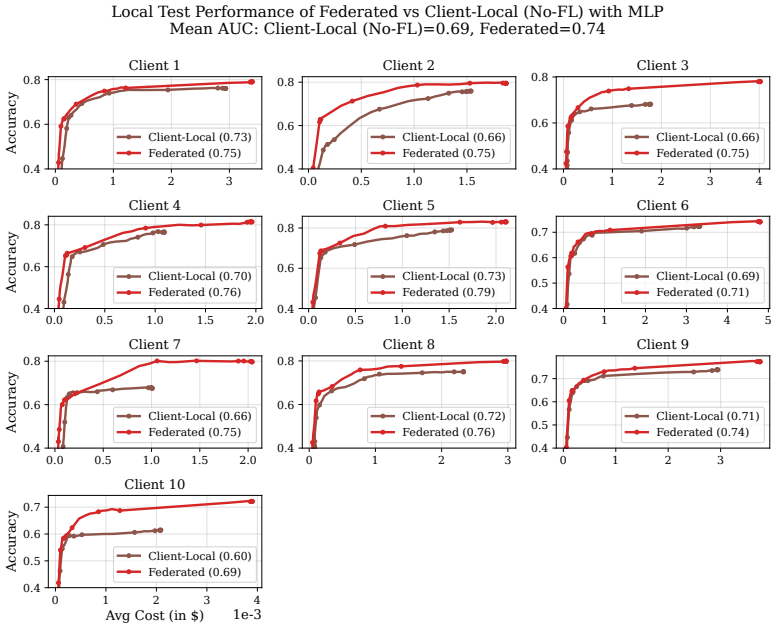


Figure 10: **All-clients local test results (`MLP-Router`).** Each panel corresponds to one client and reports accuracy–cost trade-offs for the federated router versus the client-local (no-FL) router, evaluated on that client's local test set.

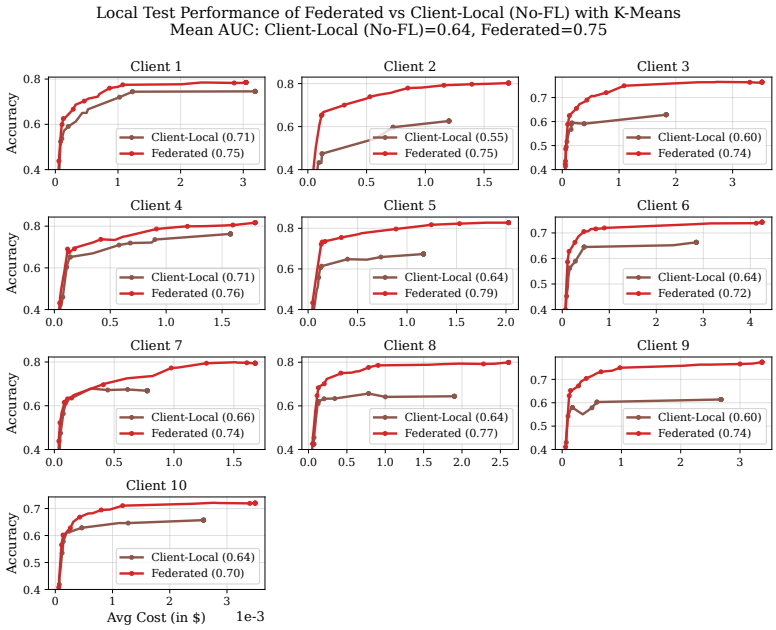


Figure 11: **All-clients local test results (`K-Means-Router`).** Each panel corresponds to one client and reports accuracy–cost trade-offs for the federated router versus the client-local (no-FL) router, evaluated on that client's local test set.

D.4 When New Clients Join the System

In practice, new clients may join after the router has already been trained. An effective system should (i) improve using the new clients' data, (ii) preserve performance on previously-seen clients (avoid *catastrophic forgetting*), and (iii) require no additional training/communication from the existing clients. We study this setting by initially training the router with 7 clients and then introducing 3 new clients whose data account for 30% of the unique task labels in the system. For `MLP-Router`, we continue training of existing router using only newly joined clients, while adding a distillation-style regularizer that penalizes deviation from the original router's outputs to preserve the initial routing policy. Let $\theta^{(0)}$ denote the router parameters before the new clients join (kept frozen during adaptation). We regularize by distilling the base router's model-wise predictions on the new clients' prompts with regularization loss, $\mathbb{E}_{\mathbf{x}\sim\mathcal{D}_{\text{new}}}\left[\frac{1}{|\mathcal{M}|}\sum_{m\in\mathcal{M}}\left(\mathsf{A}_\theta(\mathbf{x},m)-\mathsf{A}_{\theta^{(0)}}(\mathbf{x},m)\right)^2+\left(\mathsf{C}_\theta(\mathbf{x},m)-\mathsf{C}_{\theta^{(0)}}(\mathbf{x},m)\right)^2\right]$. `K-Means-Router` adaptation is training-free by weighted updating the accuracy and cost statistics of new clients in the server. Figure 12 reports the global accuracy–cost trade-off before and after new clients join. Both methods improve after incorporating the new clients, and `K-Means-Router` achieves this with minimal additional machinery.

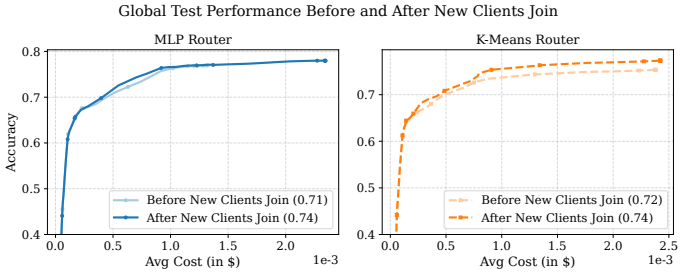


Figure 12: **Adapting to new clients.** Routers are trained initially by 7 clients, then adapted by 3 new clients, without additional participation from the original clients. `MLP-Router` uses continued training with a distillation regularizer to retain the base policy; `K-Means-Router` updates cluster-level statistics via weighted averaging. We report accuracy–cost frontiers on the global test set.

D.5 Adaptive Personalization Under High Heterogeneity

Figures 13 and 14 provide the same accuracy–cost frontier comparisons as in Figure 6, but for all clients. Overall, the per-client plots confirm the main-text trend: under extreme heterogeneity, federated `MLP-Router` can be suboptimal for a subset of clients, while adaptive personalization reliably recovers and sometimes improves upon the best achievable tradeoff between federated and local routing. For `K-Means-Router`, federated routing remains competitive across clients, whereas isolated local routing can degrade sharply due to insufficient per-centroid model-coverage.

Global-test performance. In the same high-heterogeneity setting, evaluating the personalized routers on the global test set yields normalized AUCs of 0.735 for `MLP-Router` and 0.691 for `K-Means-Router`, compared with average client-local AUCs of 0.673 and 0.636, respectively. Personalization reduces global performance relative to the unpersonalized federated router as the estimates adapt toward individual clients' distributions. Nevertheless, it retains better global generalization than client-local training while improving robustness on local test distributions.

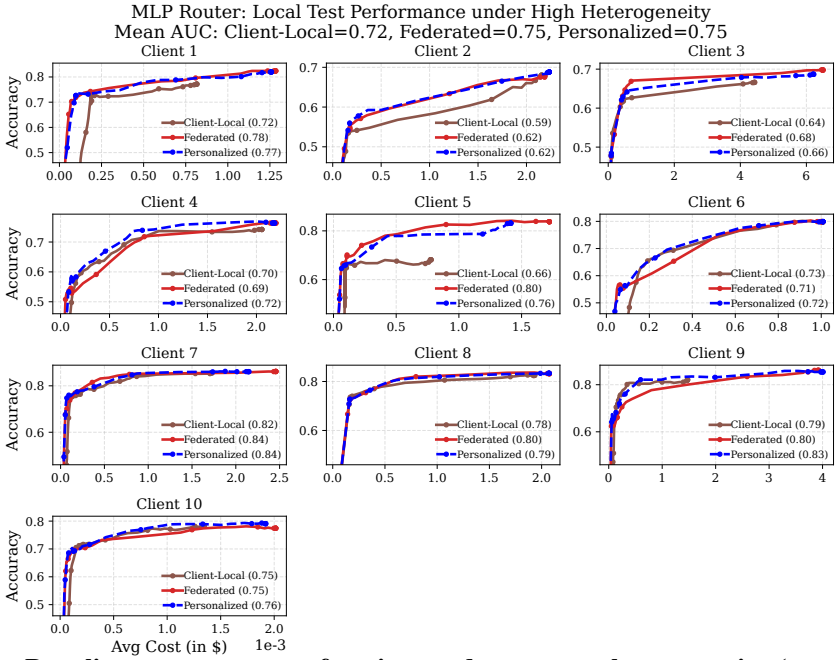


Figure 13: **Per-client accuracy–cost frontiers under extreme heterogeneity ($\alpha = 0.03$) for `MLP-Router`.**

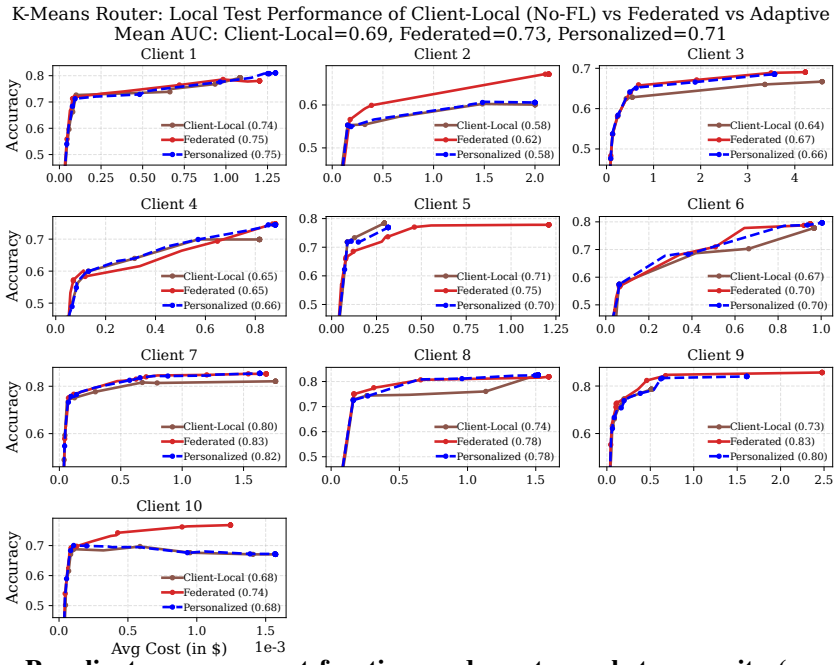


Figure 14: **Per-client accuracy–cost frontiers under extreme heterogeneity ($\alpha = 0.03$) for `K-Means-Router`.**

E Experimental Results with Different Sentence Encoder Models

We present ablations on centralized router performance on both RouterBench-Data and ProxRouter-Data by varying the choice of sentence encoders used for representing queries. In the main text, we utilize all_mpnet_base_v2 [56] sentence encoder (768 dimensional query representation). We additionally esperiment with all_minilm_l6_v2 [66] (384 dimensional query representation) and paraphrase_albert_small_v2 [53] (768 dimensional query representation). We observe that routing performance is relatively constant across sentence encoders, highlighting that presented approach is generalizable to a variety of sentence encoders.

Table 3: Performance (through normalized AUC) of `K-Means-Router` and `MLP-Router` in centralized training regime across different choices of sentence encoders used for representing queries, for both RouterBench-Data and ProxRouter-Data.

Sentence Encoder (Emb. Dim.)	**RouterBench-Data**		**ProxRouter-Data**	
	`K-Means-Router`	`MLP-Router`	`K-Means-Router`	`MLP-Router`
all_mpnet_base_v2 (768)	0.747	0.759	0.759	0.760
all_minilm_l6_v2 (384)	0.737	0.761	0.728	0.751
paraphrase_albert_ small_v2 (768)	0.746	0.758	0.751	0.755

F Experimental Results with ProxRouter-Data

We repeat the global- and local-test comparisons, model and client expansion experiments, and high-heterogeneity personalization analysis using ProxRouter-Data. In this section, we present the results and conclusions for ProxRouter-Data experiments.

Federated vs. client-local routers on the global test distribution. In Figure 15, we repeat the comparison in Figure 2 using ProxRouter-Data. As in RouterBench-Data, federated training

improves global test distribution generalization for both `MLP-Router` and `K-Means-Router` in ProxRouter-Data.

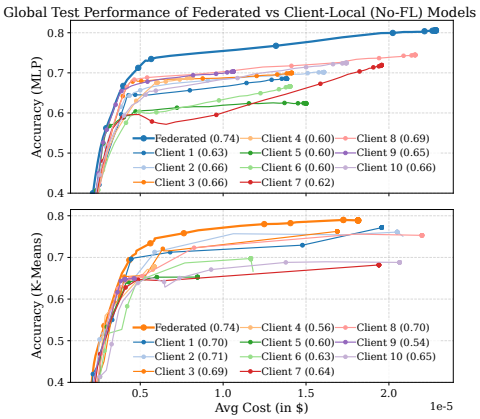


Figure 15: **(ProxRouter-Data) Federated vs. client-local (no-FL) routers on the global test distribution.** On ProxRouter-Data, we evaluate federated model and locally trained models on the global test set as in Figure 2.

Federated learning improves in-distribution local performance via better model coverage. We show the local-test set evaluation results of the federated model and locally trained models in Figure 16 on ProxRouter-Data, and again observe consistent improvements from federated collaboration. Also, in Figure 17 and Figure 18, we provide the results for all clients.

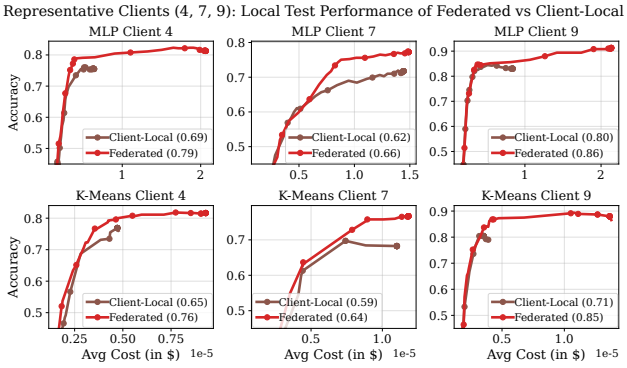


Figure 16: **(ProxRouter-Data - Subsampled Clients) Local test set results of subsampled clients.** We evaluate the federated and local models on ProxRouter-Data, as in Figure 3.

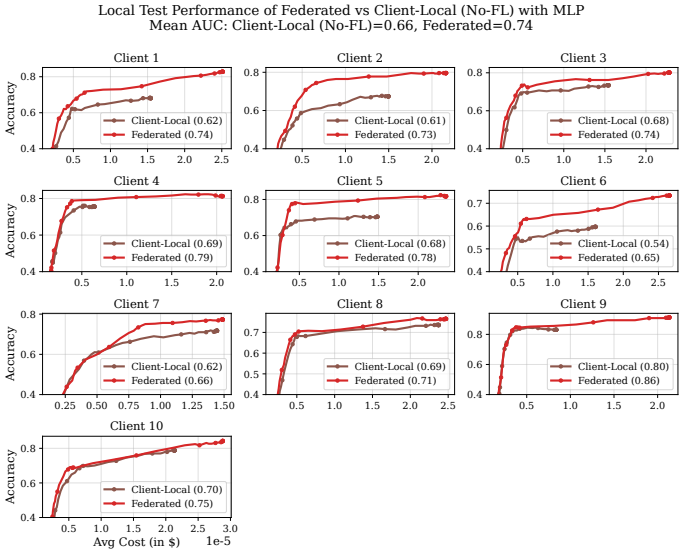


Figure 17: **(ProxRouter-Data) Local test set results of all clients with `MLP-Router`.** We evaluate the federated and local models on ProxRouter-Data, as Figure 10.

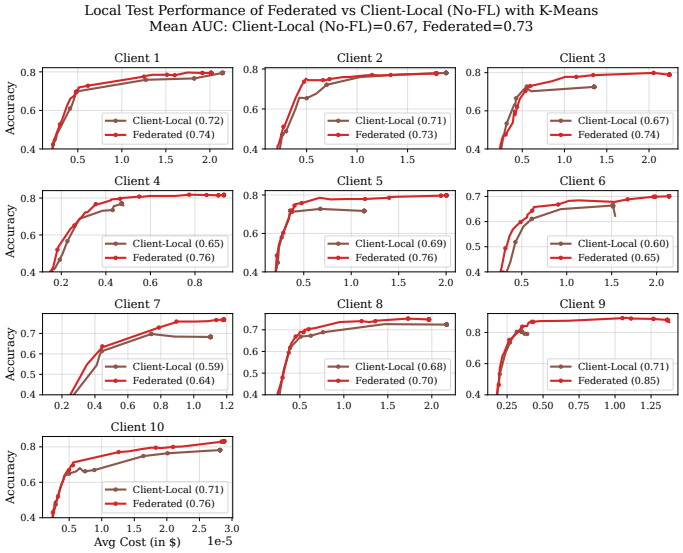


Figure 18: **(ProxRouter-Data) Local test set results of all clients with `K-Means-Router`.** We evaluate the federated and local models on ProxRouter-Data, as Figure 11.

Federated routers vs. centralized training. On ProxRouter-Data, we repeat the comparison of federated routers and centralized baseline and show the results in Figure 19.

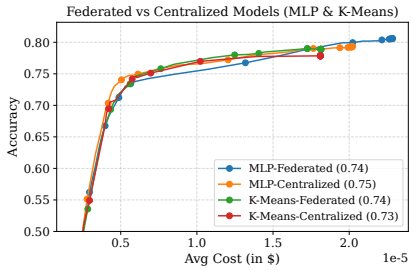


Figure 19: **(ProxRouter-Data) Federated vs. centralized training.**

Model expansion (onboarding new models). We repeat the model-expansion experiment from Section 6.3 with ProxRouter-Data with two models are withheld during initial training and introduced later. We plot the results in Figure 20.

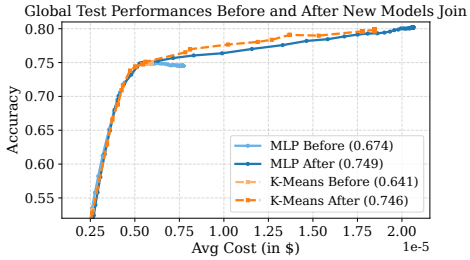


Figure 20: **(ProxRouter-Data) Adapting to newly joined models.** We observe that our framework allows successful integration of the new models introduced in the system.

Adapting to newly joined clients. We repeat the client-expansion experiment from Appendix D.4 with ProxRouter-Data. We study this setting by initially training the router with 7 clients and then introducing 3 new clients whose data accounts for 65% of the unique task labels in the system. We plot the results in Figure 21.

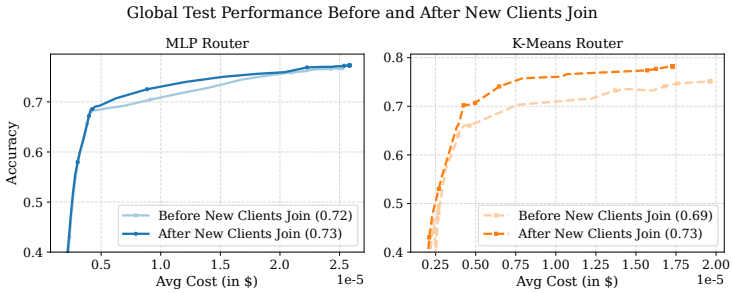


Figure 21: **(ProxRouter-Data) Adapting to newly joined clients.** We observe that our framework allows successful integration of the new clients joining in the system.

Adaptive personalization under high heterogeneity. Using the ProxRouter-Data, we repeat the high-heterogeneity experiment from the main text (Section 6.4), where client query distributions are highly non-iid (Dirichlet distribution with $\alpha = 0.04$). In Figure 22, we report local-test performance for a set of representative clients, comparing (i) client-local routers, (ii) federated routers, and (iii) our adaptive personalization. We further provide the full per-client grids in Figures 23 and 24 for `MLP-Router` and `K-Means-Router`, respectively. On ProxRouter-Data, the federated router's local-test performance varies more sharply across clients than the variation in RouterBench-Data. We observe that adaptive personalization generally matches or improves upon the better of the federated and client-local routers on each client.

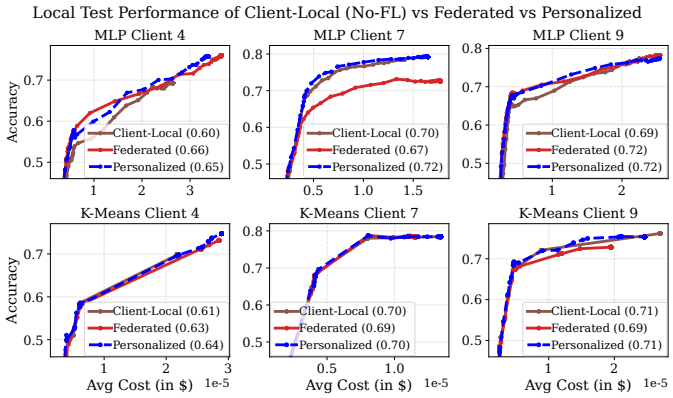


Figure 22: **(ProxRouter-Data) Adaptive personalization under high heterogeneity for representative clients with both `MLP-Router` and `K-Means-Router`.** Local test set performance of client-local, federated, and adaptive routers.

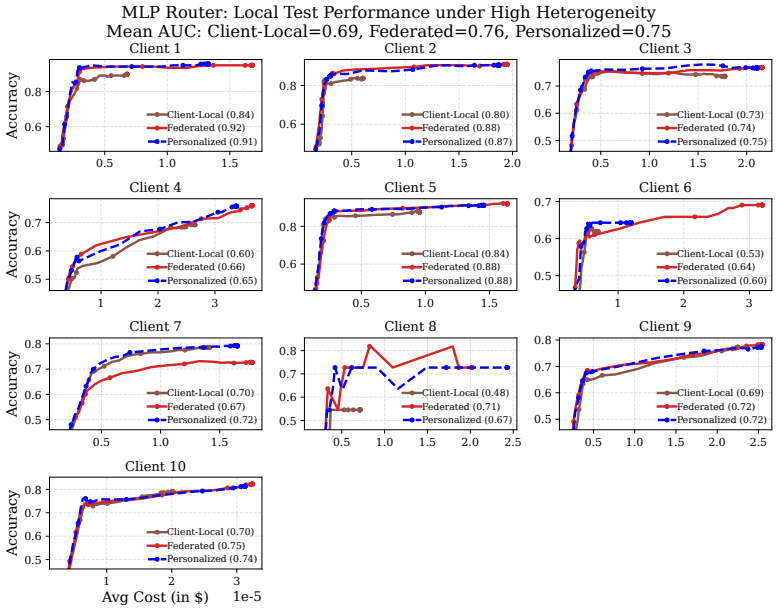


Figure 23: **(ProxRouter-Data) Adaptive personalization under high heterogeneity for all clients with `MLP-Router`.** Local test set performance of client-local (no-FL), federated, and personalized routers.

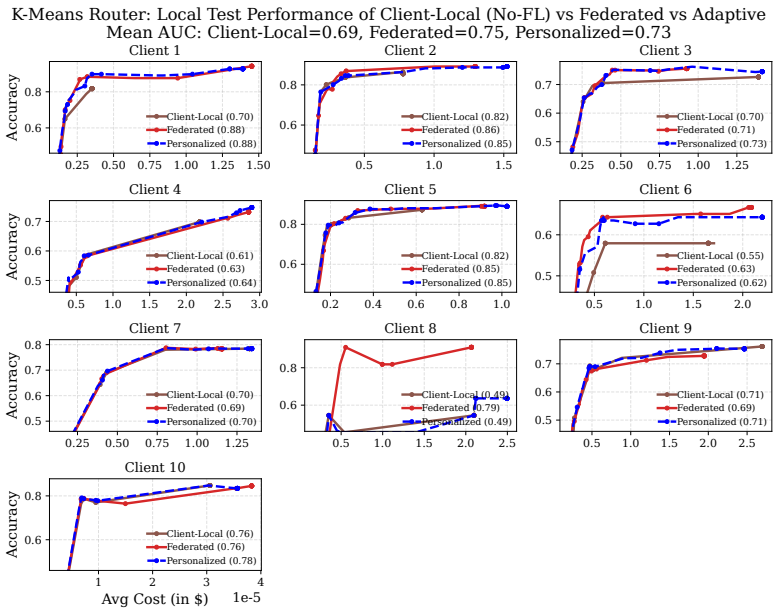


Figure 24: **(ProxRouter-Data) Adaptive personalization under high heterogeneity for all clients with `K-Means-Router`.** Local test set performance of client-local (no-FL), federated, and personalized routers.

G Experimental Results with SPROUT Dataset

We further evaluate our framework on SPROUT dataset, introduced by Somerstep et al. [55], which includes explanatory and open-ended queries with graded response-quality scores obtained through LLM evaluation against reference answers. We compare federated and client-local routers with $N = 10$ clients under sparse query–model evaluations, using both `MLP-Router` and `K-Means-Router`.

Figure 25 reports performance on the global test distribution, while Figure 26 reports local-test performance for representative clients. Federated training improves both global generalization and local routing performance, supporting its effectiveness with graded quality supervision by LLM feedback.

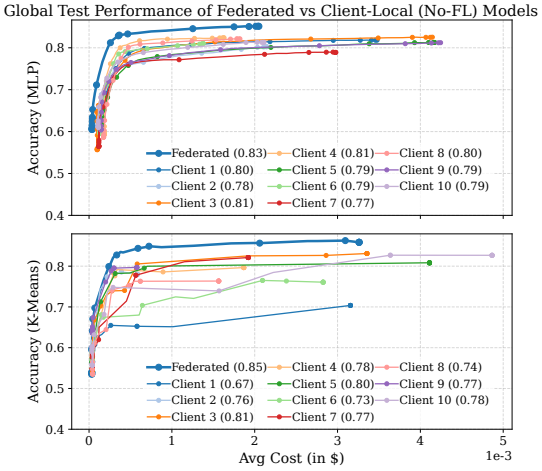


Figure 25: **SPROUT global-test performance.** Federated and client-local routers for `MLP-Router` and `K-Means-Router`. The accuracy axis denotes mean benchmark quality score, and parentheses denote normalized AUC.

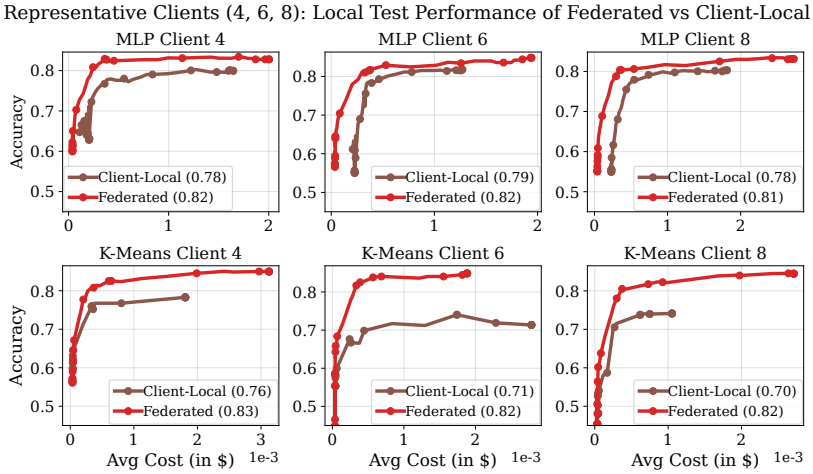


Figure 26: **SPROUT local-test performance.** Federated and client-local routers evaluated on the local test distributions of representative clients. The accuracy axis denotes mean benchmark quality score, and parentheses denote normalized AUC.

H Comparison with Federated EmbedLLM

Existing router baselines are typically designed for centralized training, while our setting assumes decentralized and sparse query–model evaluations. To the best of our knowledge, there is no prior

federated LLM router baseline directly applicable to this setup. For a broader comparison, we adapt EmbedLLM [82], a recent router that learns compact representations of language models for routing, to the federated setting. Specifically, we train EmbedLLM using the same client partitions, sparse one-model-per-query logging protocol, and federated simulation setup used for our proposed `MLP-Router` and `K-Means-Router` routers. This comparison serves two purposes: (i) to compare our proposed federated routers against a strong recent routing architecture adapted to FL, and (ii) to test whether the benefits of federation extend beyond the specific router designs proposed in this paper.

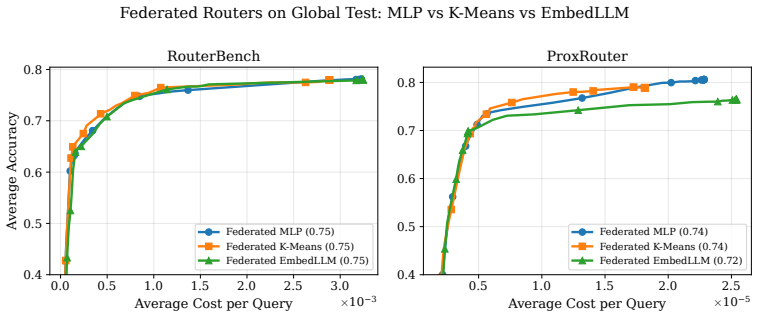


Figure 27: **Federated routers on the global test distribution with EmbedLLM.** We compare the proposed federated `MLP-Router` and federated `K-Means-Router` routers against a federated adaptation of EmbedLLM [82]. Results are shown on RouterBench-Data and ProxRouter-Data. Numbers in parentheses denote normalized AUC. The proposed methods outperform or are competitive with the federated EmbedLLM baseline across datasets.

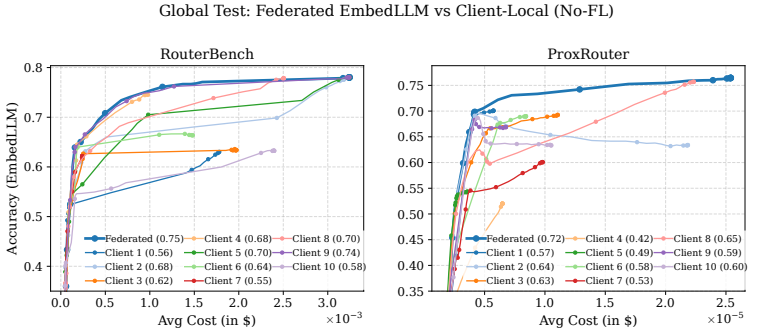


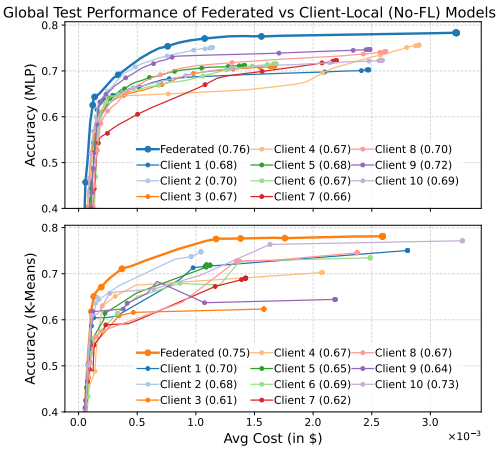
Figure 28: **Federated EmbedLLM vs. client-local EmbedLLM on the global test distribution.** We compare federated EmbedLLM against client-local, no-FL EmbedLLM routers trained independently on each client's local data. Results are shown on RouterBench-Data and ProxRouter-Data, with normalized AUC shown in parentheses. Federated training consistently improves global-distribution generalization over client-local training, supporting the value of federated collaboration beyond the proposed `MLP-Router` and `K-Means-Router` router architectures.

As shown in Figure 27, the proposed federated `MLP-Router` and `K-Means-Router` routers are competitive with or improve upon the federated EmbedLLM adaptation on the global test distribution. In addition, Figure 28 shows that applying federated training to EmbedLLM itself improves over client-local EmbedLLM training. These results reinforce the central conclusion of our work: federated collaboration improves router learning under fragmented client data and sparse model coverage, and this benefit is not limited to a single router architecture.

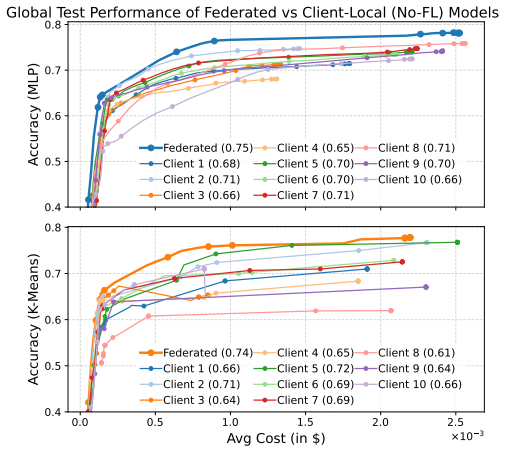
I Additional Seed Results

We repeat the main global-generalization experiment with different random seeds. The results are shown in Figure 29 for RouterBench-Data and Figure 30 for ProxRouter-Data. Across both datasets,

the results are consistent across seeds, confirming that federated learning improves over client-local routers under different random simulations.

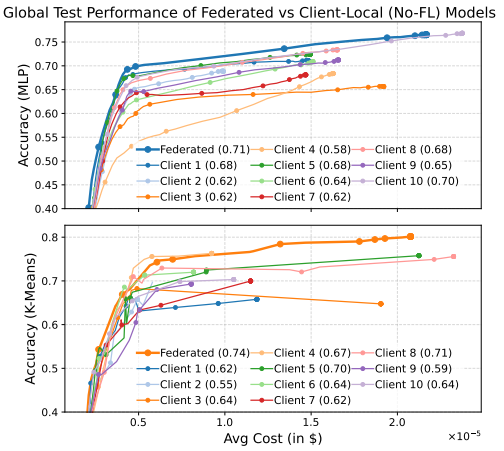


(a) Seed 1.

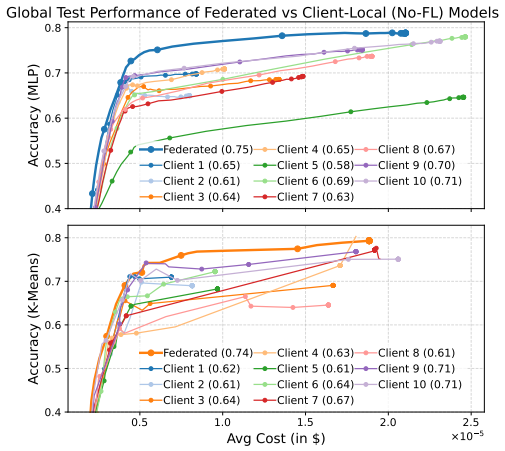


(b) Seed 2.

Figure 29: **RouterBench-Data global-generalization results under different seeds.** Federated routers consistently outperform client-local routers across both random seeds.



(a) Seed 1.



(b) Seed 2.

Figure 30: **ProxRouter-Data global-generalization results under different seeds.** Federated routers consistently outperform client-local routers across both random seeds.

I.1 Compute Resources

All experiments reported in this paper were run on a single NVIDIA V100 GPU. The full set of experiments for each benchmark and router configuration can be reproduced within a few hours on this hardware. Our experiments use lightweight router models and fixed pretrained query embeddings, so the computational cost is dominated by router training, clustering, and evaluation rather than by training or fine-tuning.

J Theory

In this section, we present the proofs and explanations for the theoretical claims in Section 5. In Appendix J.1, we present the theoretical results for `MLP-Router`, and in Appendix J.2, we present the results for `K-Means-Router`.

J.1 Theoretical Results for Federated `MLP-Router`

Setup and data model. Recall that we consider N clients indexed by $i \in \{1, \dots, N\}$ and a fixed set of M candidate models indexed by $m \in \mathcal{M} := \{1, \dots, M\}$. Client i holds a local dataset of size $D_i := |\mathcal{D}_i|$, and $D = \sum_i^N D_i$ denotes the number of total samples across clients in the federated setting.

$$\mathcal{D}_i \;=\; \left\{ \left(\mathbf{x}_i^{(j)},\, m_i^{(j)},\, \widehat{\text{acc}}_i^{(j)},\, \widehat{\text{cost}}_i^{(j)} \right) \right\}_{j=1}^{D_i},$$

where $\mathbf{x}_i^{(j)} \in \mathcal{X}$ is the query embedding,[5] and $m_i^{(j)} \in \mathcal{M}$ is selected by the logging (data-collection) policy. Given $(m_i^{(j)}, \mathbf{x}_i^{(j)})$, the system samples a model response, from which we compute an accuracy signal $\widehat{\text{acc}}_i^{(j)} \in [0,1]$ (binary in our setting, $\widehat{\text{acc}}_i^{(j)} \in \{0,1\}$, although the analysis extends to any bounded reward) and $\widehat{\text{cost}}_i^{(j)} \in [0, c_{\max}]$. Equivalently, this can be modeled using an unknown joint conditional distribution $\rho(\cdot, \cdot \mid m, \mathbf{x})$ such that whenever $\mathbf{x}$ is routed to model $m \in \mathcal{M}$ we jointly sample $(\text{acc}, \text{cost}) \sim \rho(\cdot, \cdot | m, \mathbf{x})$. Let $\text{acc}(\mathbf{x}, m)$ and $\text{cost}(\mathbf{x}, m)$ denote the associated expected accuracy and cost, thus we have

$$(\widehat{\text{acc}}_i^{(j)}, \widehat{\text{cost}}_i^{(j)}) \sim \rho(\cdot, \cdot \mid \mathbf{x}_i^{(j)}, m_i^{(j)}),$$

and

$$\mathbb{E}\left[\widehat{\text{acc}}_i^{(j)} \mid \mathbf{x}_i^{(j)}, m_i^{(j)}\right] = \text{acc}\left(\mathbf{x}_i^{(j)}, m_i^{(j)}\right), \qquad \mathbb{E}\left[\widehat{\text{cost}}_i^{(j)} \mid \mathbf{x}_i^{(j)}, m_i^{(j)}\right] = \text{cost}\left(\mathbf{x}_i^{(j)}, m_i^{(j)}\right).$$

Since $\text{acc}_i^{(j)} \in [0,1]$ and $\text{cost}_i^{(j)} \in [0, c_{\max}]$, both random variables are sub-Gaussian (not independent) because they are bounded.

`MLP-Router` **estimators and objectives.** To facilitate routing, we learn a neural network that estimates $\text{acc}(m, \mathbf{x})$ and $\text{cost}(m, \mathbf{x})$ for all models $m \in \mathcal{M}$ given an input $\mathbf{x}$. We parameterize the neural network as follows: let $\mathbf{\Theta} \subseteq \mathbb{R}^{d_{\text{MLP}}}$ denote the parameter space of the `MLP-Router` (shared trunk + M heads as in Algorithm 1). The parameters $\theta \in \mathbf{\Theta}$ induce, for each $m \in \mathcal{M}$, an accuracy-estimator function $\mathsf{A}_\theta(\cdot, \cdot) : \mathcal{X} \times \mathcal{M} \to [0,1]$ and a cost-estimator function $\mathsf{C}_\theta(\cdot, \cdot) : \mathcal{X} \times \mathcal{M} \to [0, c_{\max}]$. Accordingly, we define the estimator function class,

$$\mathcal{F} = \left\{ \left[\mathsf{A}_\theta(\cdot, \cdot),\, \mathsf{C}_\theta(\cdot, \cdot) \right] :\; \theta \in \mathbf{\Theta} \right\}. \tag{6}$$

The loss for the $(i,j)^{\text{th}}$ sample is the L_2 norm of the prediction error that renders the loss function (Algorithm 1):

$$\mathcal{L}_i(\theta) := \frac{1}{D_i} \sum_{j=1}^{D_i} \left(\left(\mathsf{A}_\theta\left(\mathbf{x}_i^{(j)}, m_i^{(j)}\right) - \widehat{\text{acc}}_i^{(j)} \right)^2 + \left(\mathsf{C}_\theta\left(\mathbf{x}_i^{(j)}, m_i^{(j)}\right) - \widehat{\text{cost}}_i^{(j)} \right)^2 \right).$$

The global objective function is defined as the average of local objectives weighted with the number of local samples:

$$\mathcal{L}(\theta) := \sum_{i=1}^{N} \frac{D_i}{D} \mathcal{L}_i(\theta).$$

Assumption J.1 (Realizability)**.** The neural network function class, $\mathbf{\Theta}$, is expressive enough such that there exist $\theta^\star \in \mathbf{\Theta}$ such that $[\mathsf{A}_{\theta^\star}(\mathbf{x}, m), \mathsf{C}_{\theta^\star}(\mathbf{x}, m)] = [\text{acc}(\mathbf{x}, m), \text{cost}(\mathbf{x}, m)]$ for all $\mathbf{x} \in \mathcal{X}$ and $m \in \mathcal{M}$.

Assumption J.2 (Smoothness)**.** Each local objective $\mathcal{L}_i$ is L-smooth: for all $i \in [N]$ and for every $\theta_1, \theta_2 \in \mathbf{\Theta}$,

$$\|\nabla \mathcal{L}_i(\theta_1) - \nabla \mathcal{L}_i(\theta_2)\| \le L \|\theta_1 - \theta_2\|.$$

Assumption J.3 (Unbiased gradients and bounded heterogeneity)**.** For each client i, stochastic gradient over a mini batch ξ denoted by $\nabla \mathcal{L}_i(\theta, \xi)$ satisfies

$$\mathbb{E}_\xi[\nabla \mathcal{L}_i(\theta, \xi)] = \nabla \mathcal{L}_i(\theta), \qquad \mathbb{E}_\xi\left[\|\nabla \mathcal{L}_i(\theta, \xi) - \nabla \mathcal{L}_i(\theta)\|^2 \right] \le \sigma^2.$$

[5] We use *query* and *query embedding* interchangeably throughout this section.

for every $\theta \in \mathbf{\Theta}$ where ξ denote the stochasticity in stochastic gradients and is independent to everything else. Moreover, there exist $\beta^2 \geq 1$ and $\kappa^2 \geq 0$ such that

$$\sum_{i=1}^{N} w_i \|\nabla \mathcal{L}_i(\theta)\|^2 \leq \beta^2 \left\| \sum_{i=1}^{N} w_i \nabla \mathcal{L}_i(\theta) \right\|^2 + \kappa^2.$$

The assumption J.1 is common in learning theory literature and is mild given a sufficiently large neural network [75, 28]. Assumptions J.2 and J.3 are standard in the federated learning literature [65, 37].

J.1.1 Convergence analysis

We first provide the analysis for the convergence of Algorithm 1.

Proposition J.4 (Convergence of federated optimization error in Algorithm 1)**.** *Under Assumptions J.2 and J.3, and choosing step size $\eta = \Theta\big(\sqrt{N/(\tau T)}\big)$ (up to constants including L), Algorithm 1 produces iterates $\{\theta^{(t)}\}_{t=1}^{T}$ such that*

$$\min_{t \in \{1,\ldots,T\}} \mathbb{E}\Big[\big\|\nabla \mathcal{L}(\theta^{(t)})\big\|^2\Big] \leq \mathcal{O}\left(\frac{1}{\sqrt{N\tau T}}\right) + \mathcal{O}\left(\frac{A\sigma^2}{\sqrt{N\tau T}}\right) + \mathcal{O}\left(\frac{N\sigma^2(\tau-1)}{\tau T}\right) + \mathcal{O}\left(\frac{N\kappa^2(\tau-1)}{T}\right),$$

where $A = N\sum_{i=1}^{N}(D_i/D)^2$.

Proof. The proof follows Theorem 1 of [65] with the same number of local steps τ. □

The proof extends to partial client participation using standard arguments from literature [18, 34].

Remark J.5 (Implication). When all the clients have the same amount of data, i.e., $D_i/D = 1/N \; \forall i \in [N]$, we obtain a linear speedup of N (number of clients) in the slowest decaying term $1/\sqrt{T}$ in the convergence to a stationary-point.

J.1.2 Sample complexity analysis

We now assume ideal optimization and analyze statistical error to show that more data can reduce the routing suboptimality. We compare local empirical risk minimization (ERM) in client's local-only training vs federated ERM. Let $\widehat{\theta}_i \in \arg\min_\theta \mathcal{L}_i(\theta)$ denote the local ERM of client i, and

$$\widehat{\theta}_{\text{fed}} \in \arg\min_{\theta} \sum_{i=1}^{N} w_i \mathcal{L}_i(\theta)$$

denote the federated ERM (equivalently, ERM in $\mathcal{D} := \bigcup_{i=1}^{N} \mathcal{D}_i$).

Each client i faces its own test-time query distribution $\mathbb{D}_i^{\text{test}}$ (possibly different from the distribution of its dataset $\mathcal{D}_i$) and has a cost tradeoff $\lambda_i \geq 0$. Define the *true* expected utility of query $\mathbf{x}$ evaluated with model m

$$U_i(\mathbf{x}, m) := \text{acc}(\mathbf{x}, m) - \lambda_i \text{cost}(\mathbf{x}, m),$$

and the estimated utility under parameters θ,

$$\widehat{U}_i(\mathbf{x}, m; \theta) := \mathsf{A}_\theta(\mathbf{x}, m) - \lambda_i \mathsf{C}_\theta(\mathbf{x}, m).$$

we also have $\widehat{U}_i(\mathbf{x}, m; \theta^\star) = U_i(\mathbf{x}, m)$ by Assumption J.1. Let the optimal router and a learned router policies be

$$\pi_i^\star(\mathbf{x}) \in \arg\max_{m \in \mathcal{M}} U_i(\mathbf{x}, m), \qquad \widehat{\pi}_{i,\theta}(\mathbf{x}) \in \arg\max_{m \in \mathcal{M}} \widehat{U}_i(\mathbf{x}, m; \theta).$$

Let $\Pi_i := \{\widehat{\pi}_{i,\theta} : \theta \in \mathbf{\Theta}\}$ be the induced routing-policy class at client i using model θ.

Definition J.6 (Router suboptimality)**.** We define the suboptimality of the router as the utility gap between the optimal router and the learned router under the test distribution. For any policy $\widehat{\pi}$, we define

$$\text{Subopt}_i(\widehat{\pi}) := \mathbb{E}_{\mathbf{x} \sim \mathbb{D}_i^{\text{test}}}\big[U_i\big(\mathbf{x}, \pi_i^\star(\mathbf{x})\big)\big] - \mathbb{E}_{\mathbf{x} \sim \mathbb{D}_i^{\text{test}}}\big[U_i\big(\mathbf{x}, \widehat{\pi}(\mathbf{x})\big)\big].$$

A data-dependent coverage coefficient. To separate *estimation error on the training support* from *test-time distribution shift*, we define a (client-specific) coefficient that measures how well the training sample controls test expectations. For any dataset $S = \{(\mathbf{x}^{(j)}, m^{(j)})\}_{j=1}^{|S|}$, define

$$\Gamma_{\mathrm{acc}}^{(i)}(S) := \sup_{\substack{\pi\in\Pi_i \\ \theta\in\mathbf{\Theta}}} \frac{\left|\mathbb{E}_{\mathbf{x}\sim\mathbb{D}_i^{\text{test}}}[\mathsf{A}_{\theta^\star}(\mathbf{x},\pi(\mathbf{x})) - \mathsf{A}_\theta(\mathbf{x},\pi(\mathbf{x}))]\right|}{\sqrt{\sum_{j=1}^{|S|}\left(\mathsf{A}_{\theta^\star}\left(\mathbf{x}^{(j)},m^{(j)}\right) - \mathsf{A}_\theta\left(\mathbf{x}^{(j)},m^{(j)}\right)\right)^2}}, \tag{7}$$

$$\Gamma_{\mathrm{cost}}^{(i)}(S) := \sup_{\substack{\pi\in\Pi_i \\ \theta\in\mathbf{\Theta}}} \frac{\left|\mathbb{E}_{\mathbf{x}\sim\mathbb{D}_i^{\text{test}}}[\mathsf{C}_{\theta^\star}(\mathbf{x},\pi(\mathbf{x})) - \mathsf{C}_\theta(\mathbf{x},\pi(\mathbf{x}))]\right|}{\sqrt{\sum_{j=1}^{|S|}\left(\mathsf{C}_{\theta^\star}\left(\mathbf{x}^{(j)},m^{(j)}\right) - \mathsf{C}_\theta\left(\mathbf{x}^{(j)},m^{(j)}\right)\right)^2}}, \tag{8}$$

and define $\Gamma^{(i)}(S) := \max\{\Gamma_{\mathrm{acc}}^{(i)}(S), \Gamma_{\mathrm{cost}}^{(i)}(S)\}$. Intuitively, smaller $\Gamma^{(i)}(S)$ indicates better coverage of client i's test distribution by dataset S. This definition is an adapted version of the concentrability coefficient in [72] for our setting. It quantifies the uncertainty inherently present in the dataset. Variations of this are used to quantify the difficulty of offline (via coverage) and online learning (via Eluder dimension) problems in learning theory literature [75].

Remark J.7. Note that the coverage coefficients used in the weaker results presented in the maintext are those of the global test distribution, which are defined as follows

$$\Gamma_{\mathrm{acc}}(S) := \sup_{\substack{\pi\in\Pi_\lambda \\ \theta\in\mathbf{\Theta}}} \frac{|\mathbb{E}_{\mathbf{x}\sim\mathbb{D}^{\text{test}}}[\mathsf{A}_{\theta^\star}(\mathbf{x},\pi(\mathbf{x})) - \mathsf{A}_\theta(\mathbf{x},\pi(\mathbf{x}))]|}{\sqrt{\sum_{j=1}^{|S|}\left(\mathsf{A}_{\theta^\star}\left(\mathbf{x}^{(j)},m^{(j)}\right) - \mathsf{A}_\theta\left(\mathbf{x}^{(j)},m^{(j)}\right)\right)^2}},$$

$$\Gamma_{\mathrm{cost}}(S) := \sup_{\substack{\pi\in\Pi_\lambda \\ \theta\in\mathbf{\Theta}}} \frac{|\mathbb{E}_{\mathbf{x}\sim\mathbb{D}^{\text{test}}}[\mathsf{C}_{\theta^\star}(\mathbf{x},\pi(\mathbf{x})) - \mathsf{C}_\theta(\mathbf{x},\pi(\mathbf{x}))]|}{\sqrt{\sum_{j=1}^{|S|}\left(\mathsf{C}_{\theta^\star}\left(\mathbf{x}^{(j)},m^{(j)}\right) - \mathsf{C}_\theta\left(\mathbf{x}^{(j)},m^{(j)}\right)\right)^2}},$$

and $\Gamma(S) := \max \Gamma_{\mathrm{acc}}(S), \Gamma_{\mathrm{cost}}(S)$, here Π_λ is the set of routing policies induced by the tradeoff parameter λ for $\theta \in \mathbf{\Theta}$. The proof for the maintext version follows the same arguments as its stronger counterpart Theorem J.12.

Lemma J.8 (Lemma 34 in [28])**.** *Let $\{X_t\}_{t\le T}$ be any sequence of real-valued random variables adapted to a filtration $(\mathcal{F}_t)_{t\le T}$. Then for any $\delta \in (0,1)$, with probability at least $1-\delta$, the following holds for any fixed $T' \le T$:*

$$\sum_{t=1}^{T'} X_t \;\le\; \sum_{t=1}^{T'} \log\big(\mathbb{E}_{t-1}[e^{X_t}]\big) \;+\; \log(\delta^{-1}). \tag{9}$$

In the following, we will state and prove a vector adapted version of Theorem 13.15 from [75] under zero misspecification error for completeness.

Lemma J.9 (Adapted version of Theorem 2.29 + Ville's inequality in [75])**.** *Let $X_1, \dots, X_n$ be conditionally independent (adapted to filtration $\mathcal{F}_t$) zero-mean σ^2-sub-Gaussian random variables. Then for any $\delta \in (0,1)$, with probability at least $1-\delta$,*

$$\sum_{i=1}^{n} X_i^2 \;\le\; n\sigma^2 + 2\sigma^2\sqrt{n\log(\delta^{-1})} + 2\sigma^2\log(\delta^{-1}) \le 2n\sigma^2 + 3\sigma^2\log(\delta^{-1}). \tag{10}$$

Lemma J.10. *Let $u = (u_1, u_2) \in \mathbb{R}^2$ and let $\varepsilon = (\varepsilon^{(1)}, \varepsilon^{(2)})$ be a random vector. Suppose that, conditioned on an event Y, the random variables $\varepsilon^{(1)}$ and $\varepsilon^{(2)}$ are subgaussian with zero-mean with parameters σ_1^2 and σ_2^2, respectively (not necessarily independent). Then we have*

$$\ln \mathbb{E}[\exp(\langle u, \varepsilon\rangle) \mid Y] \le u_1^2\sigma_1^2 + u_2^2\sigma_2^2 \le \sigma_{\max}^2\|u\|_2^2.$$

Proof. By Cauchy–Schwarz $\mathbb{E}\Big[e^{u_1\varepsilon^{(1)}+u_2\varepsilon^{(2)}} \mid \cdot\Big] \le \Big(\mathbb{E}[e^{2u_1\varepsilon^{(1)}} \mid \cdot]\Big)^{1/2}\Big(\mathbb{E}[e^{2u_2\varepsilon^{(2)}} \mid \cdot]\Big)^{1/2}$ - no independence required. Taking logs and applying the conditional sub-Gaussian bounds yields $\ln(\cdot) \le \frac{1}{2}\cdot\frac{(2u_1)^2\sigma_1^2}{2} + \frac{1}{2}\cdot\frac{(2u_2)^2\sigma_2^2}{2} = u_1^2\sigma_1^2 + u_2^2\sigma_2^2 \le \sigma_{\max}^2\|u\|_2^2$. □

Lemma J.11 (In sample error - vector adapted version Theorem 13.15 [75]). *Let $\{(X_t, \varepsilon_t)\}_{t\geq 1}$ be a filtered sequence with history $\mathcal{S}_{t-1}$, where $\varepsilon_t = (\varepsilon_t^{(1)}, \varepsilon_t^{(2)}) \in \mathbb{R}^2$, $Y_t = f^\star(X_t) + \varepsilon_t \in \mathbb{R}^2$ and $f^\star \in \mathcal{F} \subseteq (\mathcal{X} \to \mathbb{R}^2)$. Assume that for each coordinate $j \in \{1, 2\}$ the noise is conditionally zero mean subgaussian and all $\lambda \in \mathbb{R}$,*

$$\mathbb{E}[\varepsilon_t^{(j)} \mid X_t, \mathcal{S}_{t-1}] = 0, \qquad \ln \mathbb{E}\Big[\exp(\lambda \varepsilon_t^{(j)}) \mid X_t, \mathcal{S}_{t-1}\Big] \leq \frac{\lambda^2 \sigma_j^2}{2},$$

and no independence *is assumed between $\varepsilon_t^{(1)}$ and $\varepsilon_t^{(2)}$. Let $N(\epsilon) := N(\epsilon, \mathcal{F}, \|\cdot\|_\infty)$ be the corresponding covering number with $\|f\|_\infty = \sup_x \|f(x)\|_2$ For each $t \geq 1$, let $\hat{f}_t$ be an empirical risk minimizer:*

$$\hat{f}_t = \arg\min_{f \in \mathcal{F}} \sum_{s=1}^{t} \|f(X_s) - Y_s\|_2^2.$$

Define $\sigma_{\max} := \max\{\sigma_1, \sigma_2\}$ and $\bar{\sigma} := \sqrt{\sigma_1^2 + \sigma_2^2}$. Then for any $\delta \in (0, 1)$, with probability at least $1 - \delta$, for any fixed $t \geq 1$,

$$\sum_{s=1}^{t} \|\hat{f}_t(X_s) - f^\star(X_s)\|_2^2 \;\leq\; \inf_{\epsilon > 0} \left\{ 10t\epsilon^2 + 12\bar{\sigma}\, t\epsilon + 32\sigma_{\max}^2 \ln\Big(\frac{2N(\epsilon)}{\delta}\Big) + 12\bar{\sigma}^2 \ln\Big(\frac{4}{\delta}\Big) \right\}.$$

Proof. Fix $\delta \in (0, 1)$ and $\epsilon > 0$. For $f \in \mathcal{F}$, define

$$\phi(f, Z_t) := \|f(X_t) - Y_t\|_2^2 - \|f^\star(X_t) - Y_t\|_2^2, \qquad Z_t := (X_t, Y_t).$$

Let $\Delta_t := f(X_t) - f^\star(X_t) \in \mathbb{R}^2$. Since $Y_t = f^\star(X_t) + \varepsilon_t$,

$$\phi(f, Z_t) = \|\Delta_t\|_2^2 - 2\langle \Delta_t, \varepsilon_t \rangle. \tag{11}$$

Now, let $\mathcal{F}_\epsilon$ be an ϵ-cover of $(\mathcal{F}, \|\cdot\|_\infty)$ with $|\mathcal{F}_\epsilon| = N(\epsilon)$. Choose $\eta := \frac{1}{8\sigma_{\max}^2}$, using (11) and Lemma J.10 with $u = 2\eta\Delta_t$,

$$\begin{aligned} \ln \mathbb{E}\big[\exp\big(-\eta\phi(f, Z_t)\big) \mid X_t, \mathcal{S}_{t-1}\big] &= \ln \mathbb{E}\Big[\exp\Big(-\eta\|\Delta_t\|_2^2 + 2\eta\langle \Delta_t, \varepsilon_t\rangle\Big) \mid \cdot\Big] \\ &\leq -\eta\|\Delta_t\|_2^2 + 4\eta^2\sigma_{\max}^2\|\Delta_t\|_2^2 = -\frac{\|\Delta_t\|_2^2}{16\sigma_{\max}^2}. \end{aligned} \tag{12}$$

Now using equation (12), the lemma J.8 and a union bound over $f \in \mathcal{F}_\epsilon$ implies that with probability at least $1 - \delta/2$, for the fixed $t \geq 1$ and all $f \in \mathcal{F}_\epsilon$,

$$-\eta \sum_{i=1}^{t} \phi(f, Z_i) + \frac{1}{16\sigma_{\max}^2} \sum_{i=1}^{t} \|f(X_i) - f^\star(X_i)\|_2^2 \leq \ln\Big(\frac{2N(\epsilon)}{\delta}\Big).$$

Rearranging gives

$$\sum_{i=1}^{t} \|f(X_i) - f^\star(X_i)\|_2^2 \leq 2\sum_{i=1}^{t} \phi(f, Z_i) + 16\sigma_{\max}^2 \ln\Big(\frac{2N(\epsilon)}{\delta}\Big). \tag{13}$$

Now for each coordinate j, using the lemma J.9

$$\sum_{i=1}^{t} (\varepsilon_i^{(j)})^2 \leq 2t\sigma_j^2 + 3\sigma_j^2 \ln\Big(\frac{4}{\delta}\Big).$$

Taking a union bound over $j \in \{1, 2\}$, with probability at least $1 - \delta/2$, for the fixed $t \geq 1$,

$$A_t := \sum_{i=1}^{t} \|\varepsilon_i\|_2^2 = \sum_{i=1}^{t} \big((\varepsilon_i^{(1)})^2 + (\varepsilon_i^{(2)})^2\big) \leq 2t\bar{\sigma}^2 + 3\bar{\sigma}^2 \ln\Big(\frac{4}{\delta}\Big). \tag{14}$$

Fix $t \geq 1$. Let $f \in \mathcal{F}_\epsilon$ satisfy $\|f - \hat{f}_t\|_\infty \leq \epsilon$. Then for each $i \leq t$, $\|f(X_i) - Y_i\|_2 \leq \|\hat{f}_t(X_i) - Y_i\|_2 + \epsilon$. Therefore,

$$\Big(\sum_{i=1}^{t} \|f(X_i) - Y_i\|_2^2\Big)^{1/2} \leq \Big(\sum_{i=1}^{t} \|\hat{f}_t(X_i) - Y_i\|_2^2\Big)^{1/2} + \sqrt{t}\,\epsilon.$$

By the property of ERM and $f^\star \in \mathcal{F}$,

$$\sum_{i=1}^{t} \|\hat{f}_t(X_i) - Y_i\|_2^2 \le \sum_{i=1}^{t} \|f^\star(X_i) - Y_i\|_2^2 = \sum_{i=1}^{t} \|\varepsilon_i\|_2^2 = A_t.$$

Hence, using (14),

$$\begin{aligned}
&\sum_{i=1}^{t} \phi(f, Z_i) = \sum_{i=1}^{t} \|f(X_i) - Y_i\|_2^2 - A_t \le (\sqrt{A_t} + \sqrt{t}\epsilon)^2 - A_t = t\epsilon^2 + 2\epsilon\sqrt{tA_t} \\
&\le t\epsilon^2 + 2\sqrt{2}\,\bar{\sigma}\,t\epsilon + 2\sqrt{3}\,\bar{\sigma}\,\epsilon\sqrt{t\ln(4/\delta)} \\
&\le 2t\epsilon^2 + 2\sqrt{2}\,\bar{\sigma}\,t\epsilon + 3\bar{\sigma}^2 \ln\Big(\frac{4}{\delta}\Big) \qquad (15)
\end{aligned}$$

On the intersection of the events (13) and (14) (probability at least $1-\delta$), plug (15) into (13) to obtain, for the fixed $t \ge 1$ and the selected $f \in \mathcal{F}_\epsilon$,

$$\sum_{i=1}^{t} \|f(X_i) - f^\star(X_i)\|_2^2 \le 4t\epsilon^2 + 4\sqrt{2}\,\bar{\sigma}\,t\epsilon + 6\bar{\sigma}^2 \ln\Big(\frac{4}{\delta}\Big) + 16\sigma_{\max}^2 \ln\Big(\frac{2N(\epsilon)}{\delta}\Big).$$

Also, since $\|f - \hat{f}_t\|_\infty \le \epsilon$, $\Big(\sum_{i=1}^t \|\hat{f}_t(X_i) - f^\star(X_i)\|_2^2\Big)^{1/2} \le \Big(\sum_{i=1}^t \|\hat{f}_t(X_i) - f(X_i)\|_2^2\Big)^{1/2} + \Big(\sum_{i=1}^t \|f(X_i) - f^\star(X_i)\|_2^2\Big)^{1/2} \le \sqrt{t}\,\epsilon + \sqrt{B_t}$, where B_t denotes the right-hand side of the previous display. Thus

$$\sum_{i=1}^{t} \|\hat{f}_t(X_i) - f^\star(X_i)\|_2^2 \le 2t\epsilon^2 + 2B_t \le 10t\epsilon^2 + 12\bar{\sigma}\,t\epsilon + 32\sigma_{\max}^2 \ln\Big(\frac{2N(\epsilon)}{\delta}\Big) + 12\bar{\sigma}^2 \ln\Big(\frac{4}{\delta}\Big),$$

□

Main bound: federated ERM vs local ERM. We apply Lemma J.11 jointly to the vector-valued estimator $f_\theta = (\mathsf{A}_\theta, \mathsf{C}_\theta)$, whose empirical squared loss coincides with our joint ERM objective. The lemma directly controls the combined estimation error of both outputs, so no separate application or union bound over the two heads is required.

Theorem J.12 (Federated ERM can yield lower routing suboptimality)**.** *Assume realizability (Assumption J.1). Fix a client i and let $\widehat{\theta}_i$ be the local ERM on $\mathcal{D}_i$. Then, for a universal constant $C > 0$ and our estimator function class $\mathcal{F}$ (Eq.* (6)*), with probability at least $1-\delta$,*

$$\mathrm{Subopt}_i(\pi_{\widehat{i,\theta}_i}) \le C\,\max\{1, \lambda_i\}\,\Gamma^{(i)}(\mathcal{D}_i)\,c_{\max}\sqrt{\log\Big(\frac{2\mathcal{N}(1/D_i, \mathcal{F}, \|\cdot\|_\infty)}{\delta}\Big)}. \qquad (16)$$

Also, let $\widehat{\theta}_{\mathrm{fed}}$ be the federated ERM on $\mathcal{D} = \bigcup_{j=1}^N \mathcal{D}_j$. Then, with probability at least $1-\delta$,

$$\mathrm{Subopt}_i(\pi_{i,\widehat{\theta}_{\mathrm{fed}}}) \le C\,\max\{1, \lambda_i\}\,\Gamma^{(i)}(\mathcal{D})\,c_{\max}\sqrt{\log\Big(\frac{2\mathcal{N}(1/D, \mathcal{F}, \|\cdot\|_\infty)}{\delta}\Big)}. \qquad (17)$$

where $N(\epsilon, \mathcal{F}, \|\cdot\|_\infty)$ is the covering number of the class of estimators we use in the $\|\cdot\|_\infty$ norm, i.e, $\|f(x) - f'(x)\|_\infty = \sup_x \|f(x) - f'(x)\|_2$.

Proof. Let $\theta \in \mathbf{\Theta}$ be arbitrary model weights, abbreviate $\widehat{\pi}(\mathbf{x}) := \widehat{\pi}_{i,\theta}(\mathbf{x})$ and $\pi^\star(\mathbf{x}) := \pi_i^\star(\mathbf{x})$, and drop $\mathbb{D}_i^{\text{test}}$ from notation of sampling $\mathbf{x}$ for conciseness. By definition,

$$\begin{aligned}
\mathrm{Subopt}_i(\theta) &= \mathbb{E}_{\mathbf{x}\sim\mathbb{D}_i^{\text{test}}}[U_i(\mathbf{x}, \pi^\star(\mathbf{x})) - U_i(\mathbf{x}, \widehat{\pi}(\mathbf{x}))] \\
&= \mathbb{E}_{\mathbf{x}}\Big[\widehat{U}_i(\mathbf{x}, \pi^\star(\mathbf{x}); \theta) - \widehat{U}_i(\mathbf{x}, \widehat{\pi}(\mathbf{x}); \theta)\Big] + \mathbb{E}_{\mathbf{x}}\Big[U_i(\mathbf{x}, \pi^\star(\mathbf{x})) - \widehat{U}_i(\mathbf{x}, \pi^\star(\mathbf{x}); \theta)\Big] \\
&\quad + \mathbb{E}_{\mathbf{x}}\Big[\widehat{U}_i(\mathbf{x}, \widehat{\pi}(\mathbf{x}); \theta) - U_i(\mathbf{x}, \widehat{\pi}(\mathbf{x}))\Big]. \qquad (18)
\end{aligned}$$

The first term in equation (18) is ≤ 0 because $\widehat{\pi}(\mathbf{x})$ maximizes $\widehat{U}_i(\mathbf{x}, \cdot)\ \forall \mathbf{x} \in \mathcal{X}$. Thus,

$$\begin{aligned}\mathrm{Subopt}_i(\theta) &\leq |\mathbb{E}_{\mathbf{x}}[\mathsf{A}_{\theta^\star}(\mathbf{x}, \pi^\star(\mathbf{x})) - \mathsf{A}_\theta(\mathbf{x}, \pi^\star(\mathbf{x}))]| + \lambda_i\ |\mathbb{E}_{\mathbf{x}}[\mathsf{C}_{\theta^\star}(\mathbf{x}, \pi^\star(\mathbf{x})) - \mathsf{C}_\theta(\mathbf{x}, \pi^\star(\mathbf{x}))]| \\ &\quad + |\mathbb{E}_{\mathbf{x}}[\mathsf{A}_{\theta^\star}(\mathbf{x}, \widehat{\pi}(\mathbf{x})) - \mathsf{A}_\theta(\mathbf{x}, \widehat{\pi}(\mathbf{x})]| + \lambda_i\ |\mathbb{E}_{\mathbf{x}}[\mathsf{C}_{\theta^\star}(\mathbf{x}, \widehat{m}(\mathbf{x})) - \mathsf{C}_\theta(\mathbf{x}, \widehat{m}(\mathbf{x})]| \\ &\leq 2\max\{1, \lambda_i\} \left(\sup_{\pi \in \Pi_i} |\mathbb{E}_{\mathbf{x}}[\mathsf{A}_{\theta^\star}(\mathbf{x}, \pi(\mathbf{x})) - \mathsf{A}_\theta(\mathbf{x}, \pi(\mathbf{x})]| + \sup_{\pi \in \Pi_i} |\mathbb{E}_{\mathbf{x}}[\mathsf{C}_{\theta^\star}(\mathbf{x}, \pi(\mathbf{x})) - \mathsf{C}_\theta(\mathbf{x}, \pi(\mathbf{x})]| \right).\end{aligned}$$

Now apply the definitions (7)–(8) with a dataset S, yielding

$$\mathrm{Subopt}_i(\theta) \leq 2\max\{1, \lambda_i\}\, \Gamma^{(i)}(S) \left(\sqrt{\sum_{j=1}^{|S|} \Delta a_j^2} + \sqrt{\sum_{j=1}^{|S|} \Delta c_j^2} \right),$$

where Δa_j and Δc_j denote the pointwise estimation errors on $(\mathbf{x}^{(j)}, m^{(j)})$. Using $\sqrt{u} + \sqrt{v} \leq \sqrt{2(u+v)}$ gives

$$\mathrm{Subopt}_i(\theta) \leq 2\sqrt{2}\max\{1, \lambda_i\}\, \Gamma^{(i)}(S) \sqrt{\sum_{j=1}^{|S|} \left(\Delta a_j^2 + \Delta c_j^2\right)}.$$

Finally, applying Lemma J.11 to the joint estimator $f_\theta = (\mathsf{A}_\theta, \mathsf{C}_\theta)$ with $\epsilon = 1/|S|$ bounds the combined squared estimation error, yielding (16) for $S = \mathcal{D}_i$ (local-only training) and (17) for $S = \mathcal{D}$ (federated training). □

Remark J.13 (Federated learning helps with suboptimality.). To compare the suboptimality bounds of local-only `MLP-Router` (Eq. (16)) on client i and federated `MLP-Router` (Eq. (17)), we need to compare $\Gamma^{(i)}(\mathcal{D}_i)$ vs $\Gamma^{(i)}(\mathcal{D})$, and $\sqrt{\log\left(\frac{2\mathcal{N}(1/D_i, \mathcal{F}, \|\cdot\|_\infty)}{\delta}\right)}$ vs $\sqrt{\log\left(\frac{2\mathcal{N}(1/D, \mathcal{F}, \|\cdot\|_\infty)}{\delta}\right)}$. For the former comparison, one can see that since $\mathcal{D}_i \subset \mathcal{D}$, $\Gamma^{(i)}(\mathcal{D}_i) \geq \Gamma^{(i)}(\mathcal{D})$ always holds and decays roughly as $\frac{1}{\sqrt{D_i}}$ and $\frac{1}{\sqrt{D}}$, respectively, [64]. For the latter comparison, terms grow with $\log(D_i)$ and $\log(D)$, respectively [64]. Since $\sqrt{\cdot}$ grows faster, federated learning provides an advantage in data coverage in terms of smaller suboptimality.

J.2 Theoretical Results for Federated `K-Means-Router`

Setup and notation. We reuse the federated data model from Appendix J.1. Fix a test-time tradeoff parameter $\lambda \geq 0$, and let $\mathbb{D}^{\text{test}}$ denote the *global test distribution* on embeddings $\mathbf{x} \in \mathcal{X}$ (e.g., a mixture of client test distributions). For each model $m \in \mathcal{M}$, recall the conditional expected accuracy and cost functions $\mathrm{acc}(\mathbf{x}, m)$ and $\mathrm{cost}(\mathbf{x}, m)$, and define the *true expected utility*

$$U_\lambda(\mathbf{x}, m) \triangleq \mathrm{acc}(\mathbf{x}, m) - \lambda\, \mathrm{cost}(\mathbf{x}, m). \tag{19}$$

The optimal router under this setting is given by

$$\pi^\star(\mathbf{x}) \in \arg\max_{m \in \mathcal{M}} U_\lambda(\mathbf{x}, m), \tag{20}$$

where all ties are broken uniformly at random and independently of all other randomness. For any routing policy $\pi : \mathcal{X} \to \mathcal{M}$, define the *routing suboptimality*

$$\mathrm{Subopt}(\pi) \triangleq \mathbb{E}_{\mathbf{x} \sim \mathbb{D}^{\text{test}}}\Big[U_\lambda\big(\mathbf{x}, \pi^\star(\mathbf{x})\big) - U_\lambda\big(\mathbf{x}, \pi(\mathbf{x})\big)\Big]. \tag{21}$$

`K-Means-Router` **quantization map.** Given any collection of centers $\mu := \mu_{1:K_{\text{global}}} \subseteq \mathbb{R}^{d_{\text{emb}}}$ and define the nearest-center assignment and quantized embedding

$$k_\mu(\mathbf{x}) = \arg\min_{k \in \{1, \ldots, K_{\text{global}}\}} \|\mathbf{x} - \mu_k\|_2, \qquad \mu(\mathbf{x}) \triangleq \mu_{k_\mu(\mathbf{x})}, \tag{22}$$

with uniform random tie-breaking in $\arg\min$. Intuitively, $\mu(\mathbf{x})$ is the quantized embedding used by the router.

Population piecewise-constant oracle (given centers). Given the partition induced by $k_\mu(\cdot)$, define the *cluster-mean utilities*

$$\bar{U}_k(m) \triangleq \mathbb{E}_{\mathbf{x}\sim\mathbb{D}^{\text{test}}}\Big[U_\lambda(\mathbf{x},m)\,\Big|\,k_\mu(\mathbf{x})=k\Big], \qquad k\in[K_{\text{global}}],\ m\in\mathcal{M}. \tag{23}$$

The best router that is allowed to use *only* the cluster index (equivalently, only the representative $\mu(\mathbf{x})$) is the *piecewise constant oracle*

$$\pi_\mu(\mathbf{x}) \in \arg\max_{m\in\mathcal{M}} \bar{U}_{k_\mu(\mathbf{x})}(m). \tag{24}$$

Implemented `K-Means-Router` router (cluster-level estimation). Our implementation estimates, for each cluster k and model m, the cluster means of accuracy/cost using the logged data, yielding estimators $\mathsf{A}_k^{(m)}$ and $\mathsf{C}_k^{(m)}$ and the estimated cluster utility

$$\widehat{U}_k(m) \triangleq \mathsf{A}_k^{(m)} - \lambda\,\mathsf{C}_k^{(m)}. \tag{25}$$

Given a query embedding $\mathbf{x}$, the resulting `K-Means-Router` routing rule is

$$\widehat{\pi}(\mathbf{x}) = \arg\max_{m\in\mathcal{M}} \widehat{U}_{k_\mu(\mathbf{x})}(m), \tag{26}$$

again with uniform random tie-breaking.

Assumption J.14. (Sample splitting during dataset construction and uniform logging (for analysis).) Let $\mathbb{D}_i^{\mathbf{x}}$ denote the query distribution at client i. The dataset $\mathcal{D}_i := \mathcal{D}_i^{\text{cen}} \bigcup \mathcal{D}_i^{\text{util}}$ at client i is constructed as follows

1. $\mathcal{D}_i^{\text{cen}}$ construction : Sample $D_i/2$ queries IID from $\mathbb{D}_i^{\mathbf{x}}$ and for each query *arbitrarily* select a model $m\in\mathcal{M}$, observes $(\widehat{\text{acc}},\widehat{\text{cost}})$ with conditional means $\mathbb{E}[\widehat{\text{acc}}\mid\mathbf{x},M]=\text{acc}(\mathbf{x},m)$ and $\mathbb{E}[\widehat{\text{cost}}\mid\mathbf{x},M]=\text{cost}(\mathbf{x},m)$ and appends $\{\mathbf{x},m,\widehat{\text{acc}},\widehat{\text{cost}}\}$ to the dataset $\mathcal{D}_i^{\text{cen}}$.
2. $\mathcal{D}_i^{\text{util}}$ construction : Sample $D_i/2$ queries IID from $\mathbb{D}_i^{\mathbf{x}}$ and then for each query $\mathbf{x}$ draws a logged model $M\sim\text{Unif}(\mathcal{M})$ independently of $\mathbf{x}$, observes $(\widehat{\text{acc}},\widehat{\text{cost}})$ with conditional means $\mathbb{E}[\widehat{\text{acc}}\mid\mathbf{x},M]=\text{acc}(\mathbf{x},m)$ and $\mathbb{E}[\widehat{\text{cost}}\mid\mathbf{x},M]=\text{cost}(\mathbf{x},m)$ and appends $\{\mathbf{x},m,\widehat{\text{acc}},\widehat{\text{cost}}\}$ to the dataset $\mathcal{D}_i^{\text{util}}$.

Using a disjoint sample split for estimations, we enforce statistical independence between the centroid estimates $\mu_{1:K_{\text{global}}}$ (using points in dataset $\mathcal{D}_i^{\text{cen}}$) and the entities $\{\mathsf{A}_k^{(m)},\mathsf{C}_k^{(m)}\}_{k,m}$ (using the points in the dataset $\mathcal{D}_i^{\text{util}}$). Consequently, conditioning on the learned centers (and therefore on $k_\mu(\cdot)$) does not affect the distributional behavior of the utility estimators.

This split assumption dictates that we have uniform coverage over models $m\in\mathcal{M}$ for every prompt in the utility logging dataset. We need this assumption because K-means relies solely on euclidean proximity in embedding space to form clusters, it cannot capture structured correlations in model behaviors. In particular, neural models can potentially exploit these correlated performance patterns that are not explained by embedding similarity alone to generalize better. Particularly, K-means cannot capture the notion that "similar" models will have "similar" performance and only use the fact that "similar" prompts will get "similar" performance.

J.2.1 A regret decomposition: quantization + estimation

Assumption J.15 (Lipschitz utilities in embedding space)**.** There exist constants $L_{\text{acc}}, L_{\text{cost}}\geq 0$ such that for all $m\in\mathcal{M}$ and all $\mathbf{x},\mathbf{x}'\in\mathbb{R}^{d_{\text{emb}}}$,

$$\big|\text{acc}(\mathbf{x},m)-\text{acc}(\mathbf{x}',m)\big|\leq L_{\text{acc}}\,\|\mathbf{x}-\mathbf{x}'\|_2, \qquad \big|\text{cost}(\mathbf{x},m)-\text{cost}(\mathbf{x}',m)\big|\leq L_{\text{cost}}\,\|\mathbf{x}-\mathbf{x}'\|_2.$$

Consequently, $U_\lambda(\cdot,m)$ is L_λ-Lipschitz with $L_\lambda\triangleq L_{\text{acc}}+\lambda L_{\text{cost}}$.

Proposition J.16 (Decomposition into quantization and estimation terms)**.** *For any fixed centers $\mu_{1:K_{global}}$ (hence fixed $k_\mu(\cdot)$),*

$$\text{Subopt}(\widehat{\pi}) = \mathbb{E}_{\mathbf{x}\sim\mathbb{D}^{\text{test}}}\Big[U_\lambda\big(\mathbf{x},\pi^\star(\mathbf{x})\big)-U_\lambda\big(\mathbf{x},\pi_\mu(\mathbf{x})\big)\Big]+\mathbb{E}_{\mathbf{x}\sim\mathbb{D}^{\text{test}}}\Big[U_\lambda\big(\mathbf{x},\pi_\mu(\mathbf{x})\big)-U_\lambda\big(\mathbf{x},\widehat{\pi}(\mathbf{x})\big)\Big]. \tag{27}$$

J.2.2 Quantization term

Proposition J.17 (Quantization error controls the population piecewise-constant regret)**.** *Under Assumption J.15,*

$$\mathbb{E}_{\mathbf{x}\sim\mathbb{D}^{\text{test}}}\Big[U_\lambda\big(\mathbf{x},\pi^\star(\mathbf{x})\big)-U_\lambda\big(\mathbf{x},\pi_\mu(\mathbf{x})\big)\Big]\;\le\;2L_\lambda\,\mathbb{E}_{\mathbf{x}\sim\mathbb{D}^{\text{test}}}\big[\|\mathbf{x}-\mu(\mathbf{x})\|_2\big]. \tag{28}$$

Proof. Define the centroid-oracle policy $\pi_{\text{cen}}(\mathbf{x})=\arg\max_{m\in\mathcal{M}}U_\lambda(\mu(\mathbf{x}),m)$. A standard add-and-subtract argument plus Lipschitzness gives $\mathbb{E}[U_\lambda(\mathbf{x},\pi^\star(\mathbf{x}))-U_\lambda(\mathbf{x},\pi_{\text{cen}}(\mathbf{x}))]\le 2L_\lambda\,\mathbb{E}\|\mathbf{x}-\mu(\mathbf{x})\|_2$. Since π_μ maximizes the cluster-mean utility in (24) over all policies that depend only on $k_\mu(\mathbf{x})$, it is at least as good as π_{cen} within each cluster, hence has no greater regret. □

J.2.3 Estimation term (uniform logging + shift + concentration)

Logged vs. test distributions. Let $\mathbb{D}^{\mathbf{x}}$ denote the global distribution of all utility split embeddings across clients used to estimate $\{\mathsf{A}_k^{(m)},\mathsf{C}_k^{(m)}\}$. Define the corresponding logged cluster means

$$\bar U_k^{\log}(m)\;\triangleq\;\mathbb{E}_{\mathbf{x}\sim\mathbb{D}^{\mathbf{x}}}\Big[U_\lambda(\mathbf{x},m)\,\big|\,k_\mu(\mathbf{x})=k\Big]. \tag{29}$$

Cell counts and empirical utility. For each (k,m), let $n_{k,m}$ denote the *global* number of logged evaluations in the utility split whose embedding falls in cluster k and whose logged model is m (after federated aggregation across clients). Equivalently, if the client i contributes $n_{i,k,m}$ such evaluations, then $n_{k,m}=\sum_i n_{i,k,m}$. Define the empirical cluster utility (which is equal to (25) in our notation)

$$\widehat U_k(m)\;=\;\frac{1}{n_{k,m}}\sum_{t:\,k_\mu(\mathbf{x}_t)=k,\,M_t=m}\big(\tilde a_t-\lambda\tilde c_t\big),\qquad\text{when } n_{k,m}\ge 1. \tag{30}$$

Since $\tilde a_t\in[0,1]$ and $\tilde c_t\in[0,c_{\max}]$, the utility per-sample $\tilde a_t-\lambda\tilde c_t$ lies in $[-\lambda c_{\max},1]$.

Proposition J.18 (Train–test shift within clusters)**.** *For each cluster k, define the conditional distributions $\mathbb{D}^{\mathbf{x}}(\cdot\mid k)$ and $\mathbb{D}^{test}(\cdot\mid k)$ induced by conditioning on $k_\mu(\mathbf{x})=k$, and define the per-cluster Wasserstein-1 shift*

$$\Delta_k\;\triangleq\;W_1\Big(\mathbb{D}^{\mathbf{x}}(\cdot\mid k),\,\mathbb{D}^{test}(\cdot\mid k)\Big). \tag{31}$$

Then for any L-Lipschitz function f, $\big|\mathbb{E}_{\mathbb{D}^{\mathbf{x}}(\cdot|k)}[f]-\mathbb{E}_{\mathbb{D}^{test}(\cdot|k)}[f]\big|\le L\Delta_k$

(Any equivalent definition of W_1 via optimal couplings is also acceptable; we use it only through the standard Lipschitz transfer bound, cf. Kantorovich–Rubinstein duality [16].)

Lemma J.19 (Uniform deviation to the *test* cluster means)**.** *Under Assumptions J.14 and J.15. Fix the learned centers (equivalently, condition on the centroid split). Then for any $\delta\in(0,1)$, with probability at least $1-\delta$, simultaneously for all (k,m) with $n_{k,m}\ge 1$,*

$$\Big|\widehat U_k(m)-\bar U_k(m)\Big|\;\le\;L_\lambda\,\Delta_k\;+\;B_\lambda\sqrt{\frac{\log\big(\frac{2K_{global}|\mathcal{M}|}{\delta}\big)}{2\,n_{k,m}}},\qquad B_\lambda\triangleq 1+\lambda c_{\max}. \tag{32}$$

Proof. Fix (k,m) and condition on the event selecting the indices with $k_\mu(\mathbf{x})=k$ and $M=m$. By Assumption J.14, M is independent of $\mathbf{x}$, so the selected samples are unbiased for the cluster-conditional law under $\mathbb{D}^{\mathbf{x}}(\cdot\mid k)$, and thus $\mathbb{E}[\widehat U_k(m)\mid\mu_{1:K_{\text{global}}}]=\bar U_k^{\log}(m)$. Hoeffding's inequality for bounded variables in $[-\lambda c_{\max},1]$ gives

$$\Big|\widehat U_k(m)-\bar U_k^{\log}(m)\Big|\le B_\lambda\sqrt{\frac{\log\big(\frac{2K_{\text{global}}|\mathcal{M}|}{\delta}\big)}{2\,n_{k,m}}}$$

uniformly over (k,m) by a union bound. Finally, since $U_\lambda(\cdot,m)$ is L_λ-Lipschitz (Assumption J.15), the standard Lipschitz transfer bound yields $|\bar U_k^{\log}(m)-\bar U_k(m)|\le L_\lambda\Delta_k$, and the claim follows by the triangle inequality. □

Proposition J.20 (Estimation error term)**.** *Conditioned on the event of Lemma J.19, the estimation term in* (27) *satisfies*

$$\mathbb{E}_{\mathbf{x}\sim\mathbb{D}^{\text{test}}}\Big[U_\lambda\big(\mathbf{x},\pi_\mu(\mathbf{x})\big)-U_\lambda\big(\mathbf{x},\widehat{\pi}(\mathbf{x})\big)\Big] \;\le\; 2\,\mathbb{E}_{\mathbf{x}\sim\mathbb{D}^{\text{test}}}\Big[\max_{m\in\mathcal{M}}\Big|\widehat{U}_{k_\mu(\mathbf{x})}(m)-\bar{U}_{k_\mu(\mathbf{x})}(m)\Big|\Big]. \quad (33)$$

In particular, letting $n_{\min}\triangleq\min_{k,m:\,n_{k,m}\ge 1} n_{k,m}$ *and* $\Delta_{\max}\triangleq\max_k\Delta_k$,

$$\mathbb{E}_{\mathbf{x}\sim\mathbb{D}^{\text{test}}}\Big[U_\lambda\big(\mathbf{x},\pi_\mu(\mathbf{x})\big)-U_\lambda\big(\mathbf{x},\widehat{\pi}(\mathbf{x})\big)\Big] \;\le\; 2L_\lambda\,\Delta_{\max} \;+\; 2B_\lambda\sqrt{\frac{\log\!\big(\frac{2K_{global}|\mathcal{M}|}{\delta}\big)}{2\,n_{\min}}}. \quad (34)$$

Proof. Fix $\mathbf{x}$ and write $k=k_\mu(\mathbf{x})$. By construction,

$$\mathbb{E}\Big[U_\lambda(\mathbf{x},\pi_\mu(\mathbf{x}))-U_\lambda(\mathbf{x},\widehat{\pi}(\mathbf{x}))\;\Big|\;k_\mu(\mathbf{x})=k\Big]=\bar{U}_k\big(\pi_\mu(\mu_k)\big)-\bar{U}_k\big(\widehat{\pi}(\mu_k)\big).$$

Since $\widehat{\pi}(\mu_k)=\arg\max_{m\in\mathcal{M}}\widehat{U}_k(m)$, we have

$$\begin{aligned}\bar{U}_k\big(\pi_\mu(\mu_k)\big)-\bar{U}_k\big(\widehat{\pi}(\mu_k)\big) &\le \bar{U}_k\big(\pi_\mu(\mu_k)\big)-\widehat{U}_k\big(\pi_\mu(\mu_k)\big)+\widehat{U}_k\big(\widehat{\pi}(\mu_k)\big)-\bar{U}_k\big(\widehat{\pi}(\mu_k)\big)\\ &\le 2\max_{m\in\mathcal{M}}\big|\widehat{U}_k(m)-\bar{U}_k(m)\big|.\end{aligned}$$

Taking the expectation over k yields $\mathbf{x}\sim\mathbb{D}^{\text{test}}$ giving the equation (33), and equation (34) follows by applying the Lemma J.19 and then upper bounding $n_{k,m}\ge n_{\min}$ and $\Delta_k\le\Delta_{\max}$. □

Theorem J.21 (Regret bound for piecewise-constant `K-Means-Router` routing)**.** *Assume sample splitting, Assumptions J.15 and J.14. Then for any* $\delta\in(0,1)$*, with probability at least* $1-\delta$*,*

$$\text{Subopt}(\widehat{\pi}) \;\le\; 2L_\lambda\,\mathbb{E}_{\mathbf{x}\sim\mathbb{D}^{\text{test}}}\big[\|\mathbf{x}-\mu(\mathbf{x})\|_2\big] \;+\; 2L_\lambda\,\Delta_{\max} \;+\; 2B_\lambda\sqrt{\frac{\log\!\big(\frac{2K_{global}|\mathcal{M}|}{\delta}\big)}{2\,n_{\min}}}. \quad (35)$$

Proof. Combine the decomposition in Proposition J.16 with Proposition J.17 (first term) and Proposition J.20 (second term), using (34). □

Federated vs. local-only (interpretation of the estimation term). In local-only training, the client i would estimate the cell means using only its own counts $n_{i,k,m}$, so the concentration term scales as $\tilde{O}(1/\sqrt{\min_{i,k,m} n_{i,k,m}})$. In federated aggregation, cell counts add: $n_{k,m}=\sum_i n_{i,k,m}$, so typically $n_{\min}\gg\min_{i,k,m} n_{i,k,m}$ whenever clients provide complementary coverage of (k,m) cells. This is precisely the regime where the federated `K-Means-Router` improves the most over the local-only routing.

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 introduction state that the paper formulates LLM routing in a federated setting, proposes federated `MLP-Router` and `K-Means-Router` routers, provides theoretical support, and evaluates the methods empirically. These claims are supported by the problem formulation, methods, theory, and experiments sections.

   Guidelines:

   - The answer [N/A] 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 [N/A] 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: The paper discusses practical limitations and design tradeoffs, including sparse model coverage, heterogeneous client distributions, the potential need for personalization under high heterogeneity, and differences between parametric and nonparametric routers. Additional discussion and experiments are provided in the appendix.

   Guidelines:

   - The answer [N/A] 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: [Yes]

   Justification: The main text states informal convergence and suboptimality results and points to formal theorem statements and proofs in the appendix, including assumptions such as bounded heterogeneity, smoothness, realizability, and model-logging conditions.

   Guidelines:

   - The answer [N/A] 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 paper describes the benchmarks, client partitioning protocol, model-evaluation sparsity setting, router families, evaluation metric, and hyperparameter-selection procedure, with further details in the appendix. The code is included in the supplementary material and is stated to be released publicly upon publication.

   Guidelines:

   - The answer [N/A] 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: [Yes]

   Justification: The experiments use existing public router-evaluation benchmarks, and the paper states that code is included in the supplementary material for reproducibility and will be made public upon publication. The appendix provides additional details needed to reproduce the experimental setup.

   Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.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://neurips.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) necessary to understand the results?

   Answer: [Yes]

   Justification: The main text specifies the datasets, number of clients, partial participation rate, Dirichlet heterogeneity parameter, single-model-per-query evaluation setting, encoder, and evaluation procedure. Additional experimental settings, hyperparameters, and validation details are provided in the appendix.

   Guidelines:

   - The answer [N/A] 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: [Yes]

   Justification: The main figures report accuracy–cost curves and normalized AUC summaries for many clients, router families, and evaluation settings. We omit error bars from the plotted curves because each figure already contains many overlapping curves, and adding uncertainty bands would make the plots difficult to read. Instead, we provide additional experiments with multiple random seeds in Appendix I, showing that the main trends are consistent across different random simulations. We do not include seed sweeps for every experiment because doing so would substantially increase the manuscript length and reduce readability, but the additional seed results support that the main improvements are not specific to a single random split.

   Guidelines:

   - The answer [N/A] 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: Disclosed in appendix.

   Guidelines:

   - The answer [N/A] 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 work uses existing benchmark datasets and proposes routing algorithms without collecting new human-subject data or releasing high-risk generative models. To the best of the authors' knowledge, the research conforms to the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] 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: The paper discusses privacy-preserving collaboration as a motivation and potential benefit of federated routing. It also recognizes deployment-relevant risks and limitations, including sparse coverage, heterogeneous client distributions, and the need for personalization when global collaboration is misaligned with local client distributions.

    Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] 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., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper does not release a new pretrained language model, image generator, or scraped dataset with high misuse risk. The released artifact is code for reproducing router-training and evaluation experiments on existing benchmarks.

    Guidelines:

    - The answer [N/A] 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: The paper cites the datasets, model-evaluation benchmarks, sentence encoders, and prior methods used in the experiments. The licenses and terms of use were followed and can be found online at the respective websites.

    Guidelines:

    - The answer [N/A] 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: [Yes]

    Justification: The paper introduces code implementing the proposed federated routing methods and states that it is provided in the supplementary material for reproducibility. The algorithms and experimental protocol are documented in the main text and appendix.

    Guidelines:

    - The answer [N/A] 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: [N/A]

    Justification: The paper does not involve new crowdsourcing experiments or new research with human subjects. It uses existing benchmark datasets and offline query–model evaluations.

    Guidelines:

    - The answer [N/A] 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: [N/A]

Justification: The paper does not involve new human-subject studies or crowdsourcing. Therefore, IRB approval or equivalent review is not applicable.

Guidelines:

- The answer [N/A] 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 rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: The core method learns routers from existing offline query–model evaluation data and does not use LLMs as an original or non-standard component of the research methodology. LLMs appear only as evaluated models within existing routing benchmarks. LLMs are used for grammar check and basic code assistance.

Guidelines:

- The answer [N/A] 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 in the NeurIPS handbook for what should or should not be described.