DECISION-CENTRIC ACTIVE LEARNING FOR LLM ROUTING

Anonymous authors
Paper under double-blind review

ABSTRACT

LLM routing selects, for each query, the model from an available pool that best trades off answer quality and cost. Predictive routers estimate these criteria for every model and pick the best one, which presumes training data in which queries have been evaluated on all models—data that is costly to collect and must be collected again whenever a new model is added. We study this *model-onboarding* setting: a router over a set of incumbent models is already trained, and a new LLM must be integrated using as few labeled queries as possible. Classical active learning reduces labeling cost by querying where a predictor is most uncertain, but it ignores the routing decision and therefore spends much of its budget on queries the new model would never win. We propose Adaptive Decision-Centric Query-by-Committee (ADC-QBC), which mixes, within every batch, standard query-by-committee samples that reduce global uncertainty about the new model with decision-centric samples on which the committee disagrees about whether the new model should replace the best incumbent, shifting the budget from the former to the latter through a decaying schedule. On the SPROUT benchmark, onboarding GPT-4o mini and Claude 3.5 Sonnet, ADC-QBC attains higher routing accuracy than standard query-by-committee in 54 of 72 combinations of label budget and cost sensitivity; when onboarding GPT-4o mini, with 4,000 labels it outperforms standard query-by-committee with 8,000 at intermediate costs.

1 INTRODUCTION

LLM routing assigns each query to the most suitable model from a pool (Shnitzer et al., 2023), trading off quality and cost (Hu et al., 2024; Somerstep et al., 2025). A standard formulation selects

$$t^*(x) = \arg\max_{t \in \mathcal{T}} \big[a_x(t) - \lambda\, c_x(t)\big],$$

where $a_x(t)$ and $c_x(t)$ are the quality score and cost of model t on query x, and $\lambda \geq 0$ is a cost-sensitivity parameter. Routing performance is evaluated along the accuracy–cost curve obtained by sweeping λ (Hu et al., 2024; Somerstep et al., 2025).

A critical bottleneck is *labeling cost*: each (query, model) quality label requires an actual LLM call and evaluation. The full-feedback assumption of most prior work (Hu et al., 2024; Ong et al., 2025; Somerstep et al., 2025)—every query evaluated against every model—is prohibitively expensive, and must be repeated each time a new model is introduced. We focus on the *model-onboarding* scenario: a new LLM t_{new} enters an existing pipeline of well-characterized incumbents, and a limited annotation budget must be allocated to determine where it is competitive.

Simply training on a small random labeled set is insufficient. As the predict-then-optimize literature establishes (Elmachtoub & Grigas, 2022; Donti et al., 2017), minimizing prediction error is not equivalent to minimizing decision regret: the issue is *where* labels are placed, not merely how many. A predictor trained on uniform samples allocates capacity regardless of routing relevance, wasting budget on queries where the routing decision is clear-cut.

We build on Query-by-Committee (QBC) (Seung et al., 1992), which maintains a committee of regressors for the accuracy of t_{new} and uses their disagreement to decide which queries to label. **Standard QBC (S-QBC)** labels the queries on which the committee disagrees most about the predicted accuracy of t_{new}. This criterion depends neither on the incumbents nor on λ, so it spends

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

labels on queries where t_{new} is clearly dominated as readily as on queries where it would be selected. We introduce **Decision-Centric QBC (DC-QBC)**, which instead labels the queries on which the committee disagrees about the routing decision itself: whether t_{new} should be chosen over the best incumbent. DC-QBC alone, however, relies on committee estimates that are unreliable while few labels are available, and risks a self-confirming bias in which regions where the committee underestimates t_{new} are never labeled. Our method, **Adaptive Decision-Centric QBC (ADC-QBC)**, mixes both signals within every batch and shifts the budget from S-QBC to DC-QBC through a decaying schedule.

Our contributions are:

- We introduce **DC-QBC**, an acquisition criterion that measures committee disagreement at the level of routing decisions, weights it by the utility margin to the best incumbent, and aggregates it across cost-sensitivity levels.
- We propose **ADC-QBC**, which mixes S-QBC and DC-QBC within each batch via an inverse-time decay probability, smoothly shifting from global coverage to decision-centric acquisition as the labeled set grows.
- We evaluate ADC-QBC on the **SPROUT** benchmark (Somerstep et al., 2025), onboarding GPT-4o mini and Claude 3.5 Sonnet into a router over a set of incumbents. Across five seeds, ADC-QBC reaches the performance of a router trained with full information using fewer labels than S-QBC, and we report routing accuracy across label budgets and cost-sensitivity levels.

2 Related Work

LLM routing. Routing methods range from cascades and ensembles, which query several models per request (Chen et al., 2023; Wang et al., 2024a; Huang et al., 2025), to *predictive routing*, which commits to a single model per query using a learned predictor of model performance (Shnitzer et al., 2023; Shekhar et al., 2024; Ong et al., 2025; Somerstep et al., 2025). As Tsiourvas et al. (2025) note, most predictive routers first predict accuracy and cost and then select a model, which compounds errors and requires costly full-feedback data; they address both issues with a causal end-to-end framework that minimizes routing regret from observational data. Closest to our setting, Jitkrittum et al. (2026) route among LLMs unseen during training by representing each new model through its predictions on a set of representative prompts. Our work is complementary: rather than evaluating a new model on a given set of prompts, we choose adaptively *which* queries to label for it under a fixed annotation budget.

Query-by-Committee. QBC (Seung et al., 1992) selects queries of maximal committee disagreement, shrinking the version space as rapidly as possible. For noise-free binary classification, Freund et al. (1997) showed that, under suitable conditions, this yields an *exponential* reduction in label complexity over random sampling, which underpins QBC's theoretical appeal. Committees are commonly built by bagging (Abe & Mamitsuka, 1998), which makes QBC model-agnostic; for regression, disagreement is measured by the variance of the committee's predictions (Burbidge et al., 2007), and bagged committees have been applied to NLP tasks such as sequence labeling (Settles & Craven, 2008). Related work reduces the cost of evaluating LLMs by selecting a small set of informative examples (Polo et al., 2024). We use Standard QBC as the exploration component of ADC-QBC; DC-QBC uses the same committee, preserving the model-agnostic character of the full framework.

Active learning, predict-then-optimize, and decision-aware acquisition. Standard active learning criteria (Settles, 2009)—uncertainty sampling, with uncertainty estimated for instance by Monte Carlo dropout (Gal & Ghahramani, 2016) or deep ensembles (Lakshminarayanan et al., 2017); coreset selection (Sener & Savarese, 2018); variance reduction and experimental design (Cohn et al., 1996; Yu et al., 2006); and expected model change—aim to reduce predictive error across the input space, without regard to downstream decisions. The predict-then-optimize literature (Elmachtoub & Grigas, 2022; Donti et al., 2017; Wilder et al., 2019) shows that minimizing prediction error is not equivalent to minimizing decision regret. Decision-aware active learning addresses this mismatch by selecting labels according to their effect on downstream decisions: Sundin et al. (2019) query observations that

reduce the probability of inferring the wrong sign of a treatment effect, and Liu et al. (2025) propose a margin-based active learning method for contextual linear optimization whose criterion is informed by the decision error, close in spirit to the utility margin in DC-QBC. We bring this principle to LLM routing, where the decision is a discrete choice among models and DC-QBC directs labels to queries on which the committee disagrees about that choice.

3 METHODOLOGY

3.1 LLM ROUTING FORMULATION

Let $\mathcal{T}_{\text{old}}$ denote the set of *incumbent* LLMs with known utility estimates, and $t_{\text{new}} \notin \mathcal{T}_{\text{old}}$ a newly introduced LLM whose performance is unknown. We write $\mathcal{T} = \mathcal{T}_{\text{old}} \cup \{t_{\text{new}}\}$. Each query is represented by $x \in \mathcal{X} \subset \mathbb{R}^d$, typically an embedding. The *utility* of model t on query x is

$$y_x(t) \;:=\; a_x(t) - \lambda \cdot c_x(t), \tag{1}$$

where $a_x(t) \geq 0$ is the quality score, $c_x(t) \geq 0$ the cost, and $\lambda \geq 0$ a user-specified cost-sensitivity parameter. The router selects

$$t^*(x) \;=\; \arg\max_{t \in \mathcal{T}} \; y_x(t), \tag{2}$$

and routing quality is assessed along the accuracy–cost curve obtained by sweeping λ over a discrete grid $\Lambda \subset \mathbb{R}_{\geq 0}$.

For all $t \in \mathcal{T}_{\text{old}}$, reliable utility estimates $\hat{y}_x(t)$ are assumed available. The utility of t_{new} must instead be learned from a labeled set $\mathcal{L}_r$ that grows across acquisition rounds; each label requires running t_{new} on the query and evaluating its response. The learner starts from an initial batch of n_0 queries drawn uniformly at random without replacement from the pool, since the committee described below needs labels before it can score queries. Then, at each round r, the active learner selects a batch $\mathcal{B}_r \subset \mathcal{U}_r$ of size B from the unlabeled pool $\mathcal{U}_r$, queries the labels $\{a_x(t_{\text{new}})\}_{x \in \mathcal{B}_r}$, and refits its predictor on the updated $\mathcal{L}_r$ before proceeding. For simplicity, we assume the cost $c_x(t_{\text{new}})$ is known and focus on learning the accuracy of the introduced model.

3.2 STANDARD QUERY-BY-COMMITTEE (S-QBC)

We use Standard QBC (Seung et al., 1992) as our base active learning algorithm. We maintain a committee of K regressors $\{\hat{a}^{(k)}(\cdot)\}_{k=1}^{K}$ for the accuracy of t_{new}, each fit on a bootstrap resample of $\mathcal{L}_r$. The committee mean and variance of the predicted accuracy at query x are

$$\bar{a}_x \;=\; \frac{1}{K}\sum_{k=1}^{K} \hat{a}_x^{(k)}(t_{\text{new}}), \qquad \sigma_x^2 \;=\; \frac{1}{K}\sum_{k=1}^{K}\big(\hat{a}_x^{(k)}(t_{\text{new}}) - \bar{a}_x\big)^2, \tag{3}$$

and S-QBC scores each unlabeled query by the committee variance,

$$\text{Score}_{\text{S-QBC}}(x) \;=\; \sigma_x^2. \tag{4}$$

S-QBC does not depend on λ. The score is defined on predicted accuracy alone and is therefore independent of the cost-sensitivity parameter. Defining it on predicted utility instead would give the same score: because $c_x(t_{\text{new}})$ is known, it shifts every committee member's utility estimate by the same amount $\lambda\, c_x(t_{\text{new}})$, so the committee variance of the utility equals σ_x^2 for every λ. S-QBC therefore selects the same queries regardless of the cost level at which the router will operate.

This choice is motivated by two properties: (i) for noise-free binary classification, Freund et al. (1997) show that QBC can achieve an *exponential* reduction in label complexity over passive sampling; the guarantee does not transfer directly to our regression setting, but it motivates committee disagreement as an acquisition signal; and (ii) the committee variance σ_x^2 is a natural, model-agnostic disagreement measure that requires no distributional assumptions, is compatible with any base regressor, and reuses the committee that DC-QBC needs anyway. However, because the score ignores both λ and the incumbents, S-QBC may spend labels on regions dominated by other models, where better accuracy estimates for t_{new} never change a routing decision. Taking the incumbents' performance into account lets us select queries that matter for the router.

3.3 DECISION-CENTRIC QUERY-BY-COMMITTEE (DC-QBC)

DC-QBC defines acquisition scores at the level of *routing decisions* rather than accuracy predictions: it queries where the committee is most uncertain about whether t_{new} should be selected, and where that uncertainty has the greatest consequence for routing quality. Unlike S-QBC, DC-QBC depends on λ, because the routing decision does. For a fixed λ, each committee member induces a utility estimate

$$\hat{y}_x^{(k)}(t_{\text{new}}) \;=\; \hat{a}_x^{(k)}(t_{\text{new}}) - \lambda \cdot c_x(t_{\text{new}}), \qquad \bar{y}_x \;=\; \frac{1}{K}\sum_{k=1}^{K} \hat{y}_x^{(k)}(t_{\text{new}}). \tag{5}$$

To lighten notation, we suppress the dependence on λ of these and the following quantities until Equation 11.

Best incumbent. For a given query x, define the best incumbent as

$$t^-(x) \;:=\; \arg\max_{t \in \mathcal{T}_{\text{old}}} \hat{y}_x(t), \qquad \hat{y}_x(t^-) \;=\; \max_{t \in \mathcal{T}_{\text{old}}} \hat{y}_x(t). \tag{6}$$

The routing decision for t_{new} reduces to whether $y_x(t_{\text{new}}) \geq \hat{y}_x(t^-)$.

Vote entropy. For each committee member k, define the binary routing vote

$$v_k(x) \;=\; \mathbf{1}\Big[\hat{y}_x^{(k)}(t_{\text{new}}) \;\geq\; \hat{y}_x(t^-)\Big], \tag{7}$$

and let $p(x) = \frac{1}{K}\sum_{k=1}^{K} v_k(x)$ be the fraction of members voting to route to t_{new}. Decision-level uncertainty is quantified by the *vote entropy*

$$H(x) \;=\; -p(x)\log p(x) \;-\; \big(1-p(x)\big)\log\big(1-p(x)\big), \tag{8}$$

which is maximized when $p(x) = \frac{1}{2}$, i.e., the committee is evenly split on whether to route to t_{new}.

Utility-margin weighting. Vote entropy alone does not distinguish between queries where the routing decision is tight and consequential and queries where both models have nearly identical utility. To focus labeling effort on decisions that matter, we weight $H(x)$ by the *utility margin*

$$\Delta(x) \;=\; \bar{y}_x - \hat{y}_x(t^-), \tag{9}$$

the gap between the committee mean and the best incumbent, and define the per-λ score

$$\text{Score}^{\lambda}(x) \;=\; H(x) \cdot g\big(|\Delta(x)|\big), \tag{10}$$

where $g : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is a non-decreasing weighting function. We use $g(u) = \min\{u, \tau\}$ with threshold $\tau > 0$, capping the influence of very large margins; in this work, $\tau = 1$.

Aggregation over cost sensitivities. A query is informative if its routing decision is uncertain at some cost level of interest. We therefore compute the per-λ score of Equation 10 for every λ in a grid Λ and average:

$$\text{Score}_{\text{DC-QBC}}(x) \;=\; \frac{1}{|\Lambda|}\sum_{\lambda \in \Lambda} \text{Score}^{\lambda}(x). \tag{11}$$

Pure decision-centric acquisition has a known failure mode: in early rounds, $\bar{y}_x$ may be unreliable, causing DC-QBC to concentrate labels in a narrow region and never discover where else t_{new} may be competitive. This *self-confirming bias* motivates the balanced strategy that follows.

3.4 ADAPTIVE DECISION-CENTRIC QUERY-BY-COMMITTEE (ADC-QBC)

Our proposed method, **Adaptive Decision-Centric Query-by-Committee (ADC-QBC)**, combines two acquisition signals: (i) S-QBC, which targets general uncertainty about the accuracy of t_{new}, and (ii) DC-QBC, which emphasizes uncertainty that is likely to change routing decisions. Both are mixed within each batch via a decaying schedule. At each round r, a fraction P_r of the batch is

Algorithm 1 ADC-QBC for LLM Onboarding

Require: Unlabeled pool $\mathcal{U}$; incumbent estimates $\{\hat{y}_x(t)\}_{t\in\mathcal{T}_{\text{old}}}$; initial batch size n_0; batch size B; committee size K; schedule $(P_0, P_{\min}, \gamma, u)$; margin cap τ; cost grid Λ; number of rounds R
Ensure: Labeled set $\mathcal{L}$; fitted committee $\{\hat{a}^{(k)}(\cdot)\}_{k=1}^{K}$
$\mathcal{L} \leftarrow n_0$ queries drawn uniformly at random from $\mathcal{U}$; query their labels; $\mathcal{U} \leftarrow \mathcal{U} \setminus \mathcal{L}$
for $r = 0, 1, \ldots, R-1$ **do**
  Fit committee $\{\hat{a}^{(k)}(\cdot)\}_{k=1}^{K}$ on $\mathcal{L}$ via bootstrap resampling
  Compute P_r via Eq. 12; $B_{\text{S-QBC}} \leftarrow \lceil P_r B \rceil$; $B_{\text{DC-QBC}} \leftarrow B - B_{\text{S-QBC}}$
  $\mathcal{B}_{\text{S-QBC}} \leftarrow$ top-$B_{\text{S-QBC}}$ points of $\mathcal{U}$ by $\text{Score}_{\text{S-QBC}}$ (Eq. 4)
  $\mathcal{B}_{\text{DC-QBC}} \leftarrow$ top-$B_{\text{DC-QBC}}$ points of $\mathcal{U} \setminus \mathcal{B}_{\text{S-QBC}}$ by $\text{Score}_{\text{DC-QBC}}$ (Eq. 11)
  $\mathcal{B}_r \leftarrow \mathcal{B}_{\text{S-QBC}} \cup \mathcal{B}_{\text{DC-QBC}}$
  Query $\{a_x(t_{\text{new}})\}_{x\in\mathcal{B}_r}$; $\mathcal{L} \leftarrow \mathcal{L} \cup \mathcal{B}_r$; $\mathcal{U} \leftarrow \mathcal{U} \setminus \mathcal{B}_r$
end for

sampled using S-QBC and the remainder using DC-QBC, where P_r follows an inverse-time decay in the number of labels collected:

$$P_r \;=\; P_{\min} + \big(P_0 - P_{\min}\big)\,(1+\rho_r)^{-\gamma}, \qquad \rho_r \;=\; \frac{|\mathcal{L}_r| - n_0}{u}. \tag{12}$$

Here ρ_r measures progress since the initial batch in units of u labels, $P_0 \in (0, 1]$ is the initial exploration rate, $P_{\min} > 0$ is a minimum exploration floor maintained throughout, and $\gamma \geq 0$ controls the decay speed. Expressing the schedule in labels rather than rounds makes it independent of the batch size B. We set $P_0 = 1$ so that the first round after the initial batch is fully exploratory, and $P_{\min} = 0.05$ to ensure a persistent exploration budget; with $P_0 = 1$, $\gamma = 0$ recovers S-QBC.

The batch of size B is partitioned as

$$B_{\text{S-QBC}} = \lceil P_r B \rceil, \qquad B_{\text{DC-QBC}} = B - B_{\text{S-QBC}}. \tag{13}$$

The $B_{\text{S-QBC}}$ queries in $\mathcal{B}_{\text{S-QBC}}$ are the top-ranked points by $\text{Score}_{\text{S-QBC}}$; the $B_{\text{DC-QBC}}$ queries in $\mathcal{B}_{\text{DC-QBC}}$ are the top-ranked remaining points by $\text{Score}_{\text{DC-QBC}}$. As r grows, P_r decays towards $P_{\min}$, progressively shifting the batch towards DC-QBC while the floor $P_{\min}$ prevents the training distribution from collapsing onto a narrow region. Algorithm 1 summarizes ADC-QBC; S-QBC and DC-QBC are recovered as the special cases $P_r = 1$ and $P_r = 0$ for all r, respectively.

Computational complexity. Each round requires predictions from K committee members on $|\mathcal{U}_r|$ unlabeled queries, evaluation of the DC-QBC score at $|\Lambda|$ cost levels, and refitting K regressors on $|\mathcal{L}_r|$ examples. All three scale linearly in $|\mathcal{U}_r|$ and $|\mathcal{L}_r|$ for tree-based regressors, making ADC-QBC lightweight relative to the cost of LLM calls.

4 Experiments

We evaluate whether ADC-QBC onboards a new LLM with fewer labels than S-QBC, the standard active learning baseline. We report two sets of results: accuracy–cost curves at several label budgets, which show how closely each method approaches a router trained with full information (Section 4.2), and routing accuracy at fixed label budgets across cost-sensitivity levels (Section 4.3).

4.1 Experimental Setup

Dataset. We use SPROUT (Somerstep et al., 2025), a cost-aware routing benchmark of 44,241 prompts answered by 13 LLMs, with a quality score and a cost recorded for every prompt–model pair. The prompts are drawn from six tasks: GPQA (Rein et al., 2024), MuSR (Sprague et al., 2024), MMLU-Pro (Wang et al., 2024b), MATH (Hendrycks et al., 2021), OpenHermes (Teknium, 2023), and RAGBench (Friel et al., 2024). SPROUT is pre-divided into training, validation, and test splits of roughly 70%, 15%, and 15%. Following CARROT's preprocessing, we merge the training and validation splits (37,604 prompts); the unlabeled pool $\mathcal{U}$ is a random 70% subsample of this set (26,322 queries), drawn once and fixed across seeds. All results are reported on SPROUT's test split (6,637 queries), which is never used for acquisition or training. Queries are represented by

384-dimensional `all-MiniLM-L12-v2` sentence embeddings (Reimers & Gurevych, 2019; Wang et al., 2020), computed with CARROT's preprocessing code.

Onboarding protocol. Rather than routing over all 13 models, we use five: GPT-4o, GPT-4o mini, Claude 3.5 Sonnet, Amazon Titan Text Premier, and Mixtral 8x7B Instruct. We run two onboarding experiments, each designating one of **GPT-4o mini** and **Claude 3.5 Sonnet** as t_{new} and treating the other four models as the incumbents $\mathcal{T}_{\text{old}}$; the two experiments thus use the same five models with the roles of GPT-4o mini and Claude 3.5 Sonnet swapped. These are the cheapest and the most expensive of the five models ($0.15/$0.60 and $3/$15 per million input/output tokens, respectively). Incumbent accuracy estimates, both during acquisition and at evaluation, come from the full-information regressor described below. All labels for t_{new} are withheld at the start; the active learner draws an initial batch of $n_0 = 250$ queries and then acquires labels in batches of $B = 250$.

Committee and hyperparameters. We maintain a committee of $K = 10$ XGBoost regressors (Chen & Guestrin, 2016) with 400 trees and learning rate 0.05, each fit on a bootstrap resample of the current labeled set $\mathcal{L}_\tau$. The router evaluated after acquisition predicts the accuracy of t_{new} with an XGBoost regressor of the same configuration fit on $\mathcal{L}$. For DC-QBC we use $\tau = 1$ and average the per-λ score over the grid $\Lambda = \{0, 0.01, \ldots, 1\}$ (101 values). For the ADC-QBC schedule we use $P_0 = 1$, $P_{\min} = 0.05$, $\gamma = 0.7$, and $u = 50$ labels.

Baselines and evaluation. We compare ADC-QBC with S-QBC, which uses the same committee, base learner, and batch size, so the two methods differ only in how queries are selected. As a reference, we report the **full-information** router, which uses a multi-output XGBoost regressor trained on every label in the pool for all 13 SPROUT models (600 trees, learning rate 0.05, maximum depth 6, minimum child weight 2, row and column subsampling of 0.8, L1 and L2 penalties of 10^{-3} and 1, and histogram-based splitting with 256 bins). Restricted to our five models, it yields the same router in both experiments; it is the target an active learner aims to reach with as few labels as possible. We measure the answer accuracy of the routed responses, i.e., the average quality score of the model the router selects for each evaluation query, at $\lambda \in \{0, 0.2, 0.4, 0.6, 0.8, 1\}$. Every configuration is run with five random seeds, and we report the mean and standard deviation across seeds.

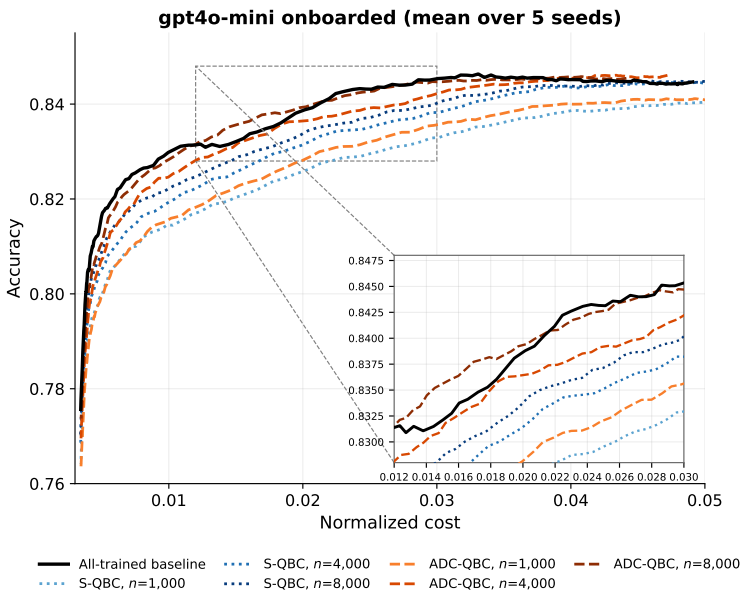
4.2 Accuracy–Cost Curves

Figure 1 shows the accuracy–cost curves traced by sweeping λ over a logarithmic grid, for the routers obtained with ADC-QBC and S-QBC after 1,000, 4,000, and 8,000 labels, together with the full-information router; the insets zoom into normalized costs between 0.012 and 0.030.

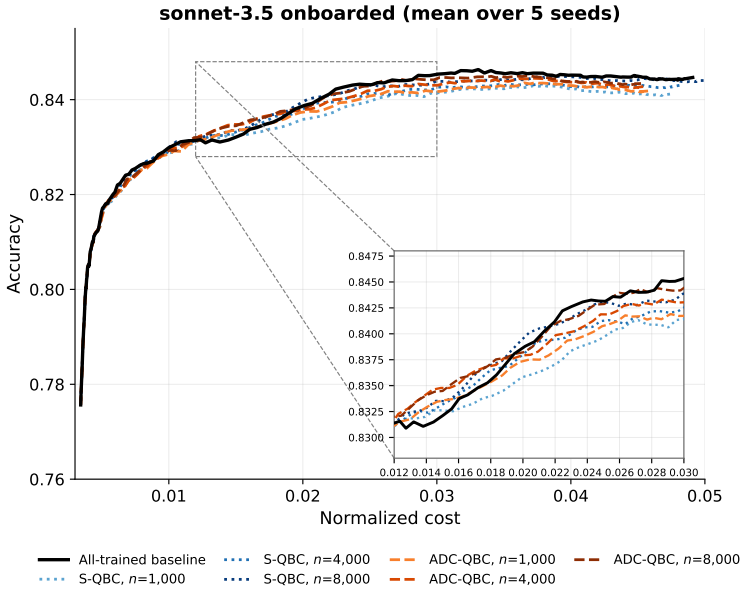
When onboarding GPT-4o mini (Figure 1a), the curve of ADC-QBC lies above that of S-QBC at every label budget over most of the cost range. The gap is widest at intermediate costs: between normalized costs of 0.012 and 0.030, ADC-QBC with 4,000 labels lies above S-QBC with 8,000 labels, achieving higher accuracy with half as many labels. In the same range, ADC-QBC with 8,000 labels lies above the full-information curve up to a normalized cost of about 0.021 and close to it beyond, whereas S-QBC with 8,000 labels stays below it. At the highest costs, the curves with 4,000 and 8,000 labels converge to the full-information curve, while both methods remain below it with 1,000 labels.

When onboarding Claude 3.5 Sonnet (Figure 1b), all curves coincide at normalized costs below about 0.012, where the expensive model is rarely worth selecting, and the differences at higher costs are smaller than for GPT-4o mini. ADC-QBC lies clearly above S-QBC with 1,000 labels; with 4,000 and 8,000 labels the curves are close, with ADC-QBC ahead at the lower end of the inset and S-QBC with 8,000 labels briefly ahead between normalized costs of about 0.018 and 0.023. Beyond about 0.024, ADC-QBC with 8,000 labels tracks the full-information curve most closely.

The gains when onboarding GPT-4o mini also extend beyond the cost levels that the acquisition targets. The full-information router attains 84.2% accuracy at $\lambda = 1$, the largest value in the grid Λ (Table 1), and reaches that accuracy only at normalized costs above about 0.022 (inset of Figure 1a). Much of the range in which ADC-QBC leads therefore corresponds to $\lambda > 1$, a regime that DC-QBC never scores. The labels it selects thus improve routing at cost levels outside Λ, which suggests that they sharpen the accuracy estimates of GPT-4o mini on the queries where it competes with the incumbents, rather than only near the decision boundaries at the sampled values of λ.



(a) Onboarding GPT-4o mini



(b) Onboarding Claude 3.5 Sonnet

Figure 1: Accuracy–cost curves on the SPROUT test split, traced by sweeping λ over a logarithmic grid, when onboarding (a) GPT-4o mini and (b) Claude 3.5 Sonnet with $n \in \{1{,}000, 4{,}000, 8{,}000\}$ labels. Dashed lines show ADC-QBC and dotted lines S-QBC, with darker shades for larger n; ADC-QBC and S-QBC curves are means over five seeds. The solid black line is the full-information router ("All-trained baseline" in the legend). Insets zoom into normalized costs between 0.012 and 0.030.

This is consistent with the design of ADC-QBC. After the initial exploratory rounds, most of each batch is spent on queries where the committee disagrees about whether t_{new} should replace the best incumbent. S-QBC, by contrast, keeps allocating labels to queries with high variance in predicted accuracy, including many where t_{new} is dominated at every cost level and the additional labels leave the router's decisions unchanged.

Table 1: Answer accuracy (%) of the routed responses when onboarding GPT-4o mini, after n labeled queries and at six cost-sensitivity levels λ. Entries are mean $\pm$ standard deviation over five seeds; the better mean for each budget and λ is in bold (both if tied). The full-information router trains the regressor of every model on all available labels.

		Cost sensitivity λ					
Budget n	Method	0.0	0.2	0.4	0.6	0.8	1.0
1,000	S-QBC	84.05 $\pm$ 0.10	83.92 $\pm$ 0.17	83.76 $\pm$ 0.12	83.51 $\pm$ 0.08	83.28 $\pm$ 0.09	83.09 $\pm$ 0.08
	ADC-QBC	**84.14** $\pm$ 0.14	**84.03** $\pm$ 0.14	**83.96** $\pm$ 0.10	**83.67** $\pm$ 0.15	**83.53** $\pm$ 0.17	**83.34** $\pm$ 0.18
2,000	S-QBC	84.28 $\pm$ 0.13	**84.21** $\pm$ 0.13	**84.14** $\pm$ 0.09	**83.89** $\pm$ 0.12	**83.68** $\pm$ 0.17	83.48 $\pm$ 0.16
	ADC-QBC	**84.32** $\pm$ 0.21	84.19 $\pm$ 0.23	84.09 $\pm$ 0.16	**83.89** $\pm$ 0.14	**83.68** $\pm$ 0.09	**83.59** $\pm$ 0.09
4,000	S-QBC	84.47 $\pm$ 0.14	84.39 $\pm$ 0.15	84.32 $\pm$ 0.15	84.04 $\pm$ 0.14	83.83 $\pm$ 0.12	83.64 $\pm$ 0.13
	ADC-QBC	**84.61** $\pm$ 0.14	**84.50** $\pm$ 0.13	**84.35** $\pm$ 0.17	**84.08** $\pm$ 0.10	**83.92** $\pm$ 0.17	**83.81** $\pm$ 0.15
8,000	S-QBC	84.49 $\pm$ 0.15	84.41 $\pm$ 0.12	84.34 $\pm$ 0.21	84.16 $\pm$ 0.20	83.95 $\pm$ 0.19	83.80 $\pm$ 0.21
	ADC-QBC	**84.54** $\pm$ 0.08	**84.52** $\pm$ 0.07	**84.47** $\pm$ 0.09	**84.32** $\pm$ 0.15	**84.18** $\pm$ 0.14	**84.02** $\pm$ 0.10
10,000	S-QBC	**84.56** $\pm$ 0.09	**84.51** $\pm$ 0.07	**84.46** $\pm$ 0.07	**84.31** $\pm$ 0.03	84.09 $\pm$ 0.06	83.89 $\pm$ 0.04
	ADC-QBC	84.50 $\pm$ 0.18	84.47 $\pm$ 0.19	84.42 $\pm$ 0.17	84.22 $\pm$ 0.16	**84.10** $\pm$ 0.17	**83.95** $\pm$ 0.15
12,000	S-QBC	**84.65** $\pm$ 0.12	84.54 $\pm$ 0.10	84.47 $\pm$ 0.10	84.32 $\pm$ 0.07	**84.22** $\pm$ 0.06	**84.10** $\pm$ 0.08
	ADC-QBC	84.64 $\pm$ 0.08	**84.58** $\pm$ 0.05	**84.54** $\pm$ 0.07	**84.37** $\pm$ 0.09	84.19 $\pm$ 0.09	84.06 $\pm$ 0.07
Full information		84.47	84.51	84.61	84.45	84.32	84.23

4.3 Accuracy Across Label Budgets and Cost Sensitivities

Tables 1 and 2 report the answer accuracy of the routed responses after n labeled queries for t_{new}, at six values of λ.

When onboarding Claude 3.5 Sonnet, ADC-QBC attains the higher mean in 29 of the 36 budget–λ settings, and its advantage grows with the budget. From 10,000 labels onward it is ahead at every λ, by 0.15 to 0.34 percentage points, a margin larger than the standard deviation across seeds in 11 of these 12 settings. At the same budgets it matches or exceeds the full-information router at every λ, whereas S-QBC does so in only one of the 36 settings ($\lambda = 0.8$ with 12,000 labels). That an actively labeled regressor can match the full-information one is not a contradiction: the latter is fit to minimize prediction error over all queries rather than routing regret, so it need not be an upper bound on routing accuracy.

When onboarding GPT-4o mini, the differences are smaller. ADC-QBC is ahead at every λ with 1,000, 4,000, and 8,000 labels, by at most 0.25 points, while with 2,000, 10,000, and 12,000 labels the two methods trade places and differ by less than one standard deviation. Overall, ADC-QBC attains the higher mean in 25 of the 36 settings. For $\lambda \geq 0.4$, neither method reaches the full-information router within 12,000 labels. The cost levels in the tables ($\lambda \leq 1$) cover the high-accuracy end of the curves in Figure 1, where full-information accuracy is between 84.2% and 84.6%; for GPT-4o mini, the largest gaps between the methods lie at lower costs (Figure 1a).

4.4 Discussion

Where the gains arise. The two experiments differ in where the new model competes with the incumbents. GPT-4o mini is the cheapest model in the pool, so it is a routing candidate across a wide range of cost levels, and the gap between the methods is widest at intermediate costs (Figure 1a), presumably because there the choice between GPT-4o mini and a more expensive incumbent is close for many queries. Claude 3.5 Sonnet is the most expensive model, so it is worth selecting only when cost matters little: the curves coincide at low cost (Figure 1b), and the advantage of ADC-QBC in Table 2 grows with the budget. Plausibly, the queries on which Sonnet improves on the incumbents enough to justify its cost form a small set that the committee can locate only once enough labels are available.

How much of the budget is decision-centric. With our schedule (Equation 12), the exploration probability drops from $P_r = 1$ right after the initial batch to 0.32 after 500 labels, 0.19 after 1,000,

Table 2: Answer accuracy (%) of the routed responses when onboarding Claude 3.5 Sonnet, in the same format as Table 1. The full-information router is the same as in Table 1, since it trains every model's regressor on all available labels.

		Cost sensitivity λ					
Budget n	Method	0.0	0.2	0.4	0.6	0.8	1.0
1,000	S-QBC	84.12 ± 0.19	84.23 ± 0.14	84.23 ± 0.15	84.18 ± 0.18	84.08 ± 0.19	84.04 ± 0.15
	ADC-QBC	**84.18** ± 0.05	**84.32** ± 0.05	**84.26** ± 0.10	**84.23** ± 0.14	**84.15** ± 0.15	**84.10** ± 0.14
2,000	S-QBC	**84.19** ± 0.19	**84.25** ± 0.20	**84.26** ± 0.15	**84.23** ± 0.15	84.15 ± 0.14	84.07 ± 0.16
	ADC-QBC	84.14 ± 0.17	**84.25** ± 0.16	**84.26** ± 0.16	**84.23** ± 0.09	**84.21** ± 0.13	**84.14** ± 0.14
4,000	S-QBC	**84.30** ± 0.16	84.28 ± 0.10	84.29 ± 0.04	84.25 ± 0.06	84.12 ± 0.10	84.05 ± 0.11
	ADC-QBC	84.27 ± 0.11	**84.39** ± 0.10	**84.39** ± 0.13	**84.32** ± 0.18	**84.29** ± 0.23	**84.18** ± 0.19
8,000	S-QBC	**84.40** ± 0.13	**84.45** ± 0.07	84.41 ± 0.09	84.37 ± 0.13	84.30 ± 0.15	84.20 ± 0.10
	ADC-QBC	84.34 ± 0.14	84.44 ± 0.07	**84.48** ± 0.06	**84.47** ± 0.12	**84.43** ± 0.11	**84.27** ± 0.10
10,000	S-QBC	84.39 ± 0.17	84.36 ± 0.15	84.37 ± 0.13	84.27 ± 0.08	84.21 ± 0.09	84.12 ± 0.08
	ADC-QBC	**84.54** ± 0.14	**84.59** ± 0.14	**84.62** ± 0.19	**84.56** ± 0.19	**84.53** ± 0.18	**84.46** ± 0.10
12,000	S-QBC	84.37 ± 0.12	84.41 ± 0.11	84.42 ± 0.12	84.43 ± 0.10	84.33 ± 0.08	84.18 ± 0.11
	ADC-QBC	**84.56** ± 0.12	**84.66** ± 0.14	**84.67** ± 0.12	**84.63** ± 0.14	**84.53** ± 0.16	**84.44** ± 0.15
Full information		84.47	84.51	84.61	84.45	84.32	84.23

and 0.08 after 8,000, approaching the floor $P_{\min} = 0.05$. Beyond the first few rounds, most labels are therefore chosen by DC-QBC, while the floor keeps a small share of every batch on S-QBC. At 1,000 labels the labeled set is still largely exploratory, yet ADC-QBC already leads for both models (Tables 1 and 2); for GPT-4o mini, the gap at intermediate costs then widens with the budget (Figure 1a).

Stability. The standard deviations of ADC-QBC across seeds are comparable to those of S-QBC (Tables 1 and 2), so concentrating labels near routing decisions does not make the procedure less stable.

5 Conclusion

We studied efficient LLM onboarding: how to allocate a fixed annotation budget to characterize a newly introduced model t_{new} and integrate it into an existing router. ADC-QBC mixes, within every batch, S-QBC, which reduces uncertainty about the accuracy of t_{new} independently of λ and of the incumbents, with DC-QBC, which targets queries on which the committee disagrees about whether t_{new} should replace the best incumbent, and shifts the budget toward DC-QBC as the committee becomes reliable. On SPROUT, ADC-QBC reaches the full-information router with fewer labels than S-QBC: when onboarding GPT-4o mini, 4,000 labels selected by ADC-QBC outperform 8,000 selected by S-QBC at intermediate costs, and when onboarding Claude 3.5 Sonnet, ADC-QBC matches the full-information router at every tabulated cost level from 10,000 labels on, which S-QBC achieves in only one of 36 settings. Since each label requires a costly LLM call, these savings matter in practice.

More broadly, this work advances the idea that the predict-then-optimize insight—aligning the learning objective with the downstream decision—applies not only to the training criterion (Elmachtoub & Grigas, 2022; Donti et al., 2017) but equally to the *data collection* stage. Directing annotation effort toward the routing boundary rather than toward regions of high predictive uncertainty improves label efficiency without changing the router architecture or training procedure.

Limitations and future work. We onboard one model at a time; extending ADC-QBC to the simultaneous onboarding of multiple new models is a natural next step. We also assume the cost of the new model is known, whereas in practice output length, and hence cost, must be predicted as well. Finally, the ADC-QBC framework is not specific to LLM routing and may apply to any active learning problem where labels are costly and decisions are the ultimate downstream objective.

AI USE STATEMENT

In this work, we used generative AI tools to help draft and polish the text of the paper, format the LaTeX source and results tables, and assist with writing code. We did not use generative AI tools for research ideation, conceptualization, or the development of the method and its mathematical formulation. All authors reviewed the AI-assisted content, and we checked every citation and every reported number against the original sources and our experimental outputs. We take responsibility for the final content of this work, including text, claims, and code produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

Section 3.4 and Algorithm 1 fully specify ADC-QBC, and Section 4.1 lists the dataset, onboarded and incumbent models, committee, hyperparameters, and number of seeds used in our experiments. SPROUT is publicly available.

REFERENCES

Naoki Abe and Hiroshi Mamitsuka. Query learning strategies using boosting and bagging. In *Proceedings of the Fifteenth International Conference on Machine Learning*, pp. 1–9. Morgan Kaufmann, 1998.

Robert Burbidge, Jem J. Rowland, and Ross D. King. Active learning for regression based on query by committee. In *Intelligent Data Engineering and Automated Learning – IDEAL 2007*, volume 4881 of *Lecture Notes in Computer Science*, pp. 209–218. Springer, 2007.

Lingjiao Chen, Matei Zaharia, and James Zou. FrugalGPT: How to use large language models while reducing cost and improving performance. *arXiv preprint arXiv:2305.05176*, 2023.

Tianqi Chen and Carlos Guestrin. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785–794, 2016. doi: 10.1145/2939672.2939785.

David A Cohn, Zoubin Ghahramani, and Michael I Jordan. Active learning with statistical models. *Journal of Artificial Intelligence Research*, 4:129–145, 1996.

Priya L. Donti, Brandon Amos, and J. Zico Kolter. Task-based end-to-end model learning in stochastic optimization. In *Advances in Neural Information Processing Systems*, volume 30, pp. 5484–5494, 2017.

Adam N. Elmachtoub and Paul Grigas. Smart "Predict, then Optimize". *Management Science*, 68(1): 9–26, 2022. doi: 10.1287/mnsc.2020.3922.

Yoav Freund, H. Sebastian Seung, Eli Shamir, and Naftali Tishby. Selective sampling using the query by committee algorithm. *Machine Learning*, 28(2–3):133–168, 1997. doi: 10.1023/A:1007330508534.

Robert Friel, Masha Belyi, and Atindriyo Sanyal. RAGBench: Explainable benchmark for retrieval-augmented generation systems. *arXiv preprint arXiv:2407.11005*, 2024.

Yarin Gal and Zoubin Ghahramani. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of the 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pp. 1050–1059, 2016.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. *arXiv preprint arXiv:2103.03874*, 2021.

Qitian Jason Hu, Jacob Bieker, Xiuyu Li, Nan Jiang, Benjamin Keigwin, Gaurav Ranganath, Kurt Keutzer, and Shriyash Kaustubh Upadhyay. RouterBench: A benchmark for multi-LLM routing system. *arXiv preprint arXiv:2403.12031*, 2024.

Keke Huang, Yimin Shi, Dujian Ding, Yifei Li, Yang Fei, Laks Lakshmanan, and Xiaokui Xiao. ThriftLLM: On cost-effective selection of large language models for classification queries. *Proceedings of the VLDB Endowment*, 18(11), 2025.

Wittawat Jitkrittum, Harikrishna Narasimhan, Ankit Singh Rawat, Jeevesh Juneja, Congchao Wang, Zifeng Wang, Alec Go, Chen-Yu Lee, Pradeep Shenoy, Rina Panigrahy, Aditya Krishna Menon, and Sanjiv Kumar. Universal model routing for efficient LLM inference. In *The Fourteenth International Conference on Learning Representations*, 2026.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Mo Liu, Paul Grigas, Heyuan Liu, and Zuo-Jun Max Shen. Active learning for contextual linear optimization: A margin-based approach. *arXiv preprint arXiv:2305.06584*, 2025.

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.

Felipe Maia Polo, Lucas Weber, Leshem Choshen, Yuekai Sun, Gongjun Xu, and Mikhail Yurochkin. tinyBenchmarks: Evaluating LLMs with fewer examples. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 34303–34326, 2024.

Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 3982–3992, 2019.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. GPQA: A graduate-level Google-proof Q&A benchmark. In *First Conference on Language Modeling*, 2024.

Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. In *International Conference on Learning Representations*, 2018.

Burr Settles. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009. URL http://burrsettles.com/pub/settles.activelearning.pdf.

Burr Settles and Mark Craven. An analysis of active learning strategies for sequence labeling tasks. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pp. 1070–1079, 2008.

H Sebastian Seung, Manfred Opper, and Haim Sompolinsky. Query by committee. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, pp. 287–294. ACM, 1992.

Shivanshu Shekhar, Tanishq Dubey, Koyel Mukherjee, Apoorv Saxena, Atharv Tyagi, and Nishanth Kotla. Towards optimizing the costs of LLM usage. *arXiv preprint arXiv:2402.01742*, 2024.

Tal Shnitzer, Anthony Ou, Mírian Silva, Kate Soule, Yuekai Sun, Justin Solomon, Neil Thompson, and Mikhail Yurochkin. Large language model routing with benchmark datasets. *arXiv preprint arXiv:2309.15789*, 2023.

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

Zayne Sprague, Xi Ye, Kaj Bostrom, Swarat Chaudhuri, and Greg Durrett. MuSR: Testing the limits of chain-of-thought with multistep soft reasoning. In *The Twelfth International Conference on Learning Representations*, 2024.

Iiris Sundin, Peter Schulam, Eero Siivola, Aki Vehtari, Suchi Saria, and Samuel Kaski. Active learning for decision-making from imbalanced observational data. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 6046–6055, 2019.

Teknium. OpenHermes 2.5. https://huggingface.co/datasets/teknium/OpenHermes-2.5, 2023.

Asterios Tsiourvas, Wei Sun, and Georgia Perakis. Causal LLM routing: End-to-end regret minimization from observational data. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://arxiv.org/abs/2505.16037.

Hongyi Wang, Felipe Maia Polo, Yuekai Sun, Souvik Kundu, Eric Xing, and Mikhail Yurochkin. Fusing models with complementary expertise. In *The Twelfth International Conference on Learning Representations*, 2024a.

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 *Advances in Neural Information Processing Systems*, volume 33, pp. 5776–5788, 2020.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, et al. MMLU-Pro: A more robust and challenging multi-task language understanding benchmark. In *The Thirty-eighth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024b.

Bryan Wilder, Bistra Dilkina, and Milind Tambe. Melding the data-decisions pipeline: Decision-focused learning for combinatorial optimization. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence*, volume 33, pp. 1658–1665, 2019. doi: 10.1609/aaai.v33i01.33011658.

Kai Yu, Jinbo Bi, and Volker Tresp. Active learning via transductive experimental design. In *Proceedings of the 23rd International Conference on Machine Learning*, pp. 1081–1088. ACM, 2006. doi: 10.1145/1143844.1143980.