Online Learning with LLM Experts from Limited Feedback

Anonymous ACL submission

Abstract

We study adaptive routing of prompts to large language model (LLM) experts to maximize response quality in an online setting with limited feedback. We formulate it as a bandit problem with K actions that represent experts and d features that encode prompts, over a horizon of T rounds. We propose algorithms that strategically select and observe rewards to minimize regret. In the full-information setting, we achieve a regret of $\tilde{O}(dT/\sqrt{m})$, while in the bandit setting we achieve $\tilde{O}(dT\sqrt{K/m})$, where $m \ll T$ is a budget on feedback. Our experiments show that we efficiently learn high-quality routing strategies across diverse LLMs from limited feedback.

1 Introduction

Large language models (LLMs) are pervasive nowadays. OpenAI's GPT models (OpenAI et al., 2023), LLaMA3 (Dubey et al., 2024), and Mistral AI (Mistral AI, 2025) have been successfully used to solve many tasks, such as document processing and code generation. Different LLMs have different costs and capabilities (Artificial Analysis, 2025). The diversity of costs, even within the same family of models, can be stark (OpenAI, 2025). The diverse capabilities of LLMs are obvious from public datasets. For instance, the Nectar dataset (Zhu et al., 2024) contains responses to 183k prompts of many popular models judged by GPT-4. When GPT-3.5-Turbo, GPT-3.5-Turbo-Instruct, GPT-4, GPT-4-0613, LLaMA-2-7B-Chat, and Mistral-7B-Instruct are judged, the win rates of the models are 0.182, 0.091, 0.203, 0.319, 0.073, and 0.132. Therefore, no model dominates the others more than a third of the time and adaptation is beneficial.

We formulate the problem of online learning with *LLM experts* as follows. We have K different LLMs and interact with them sequentially over T rounds. In each round, a prompt arrives and we route it to one expert, conditioned on the prompt. The expert responds and its response is associated with some reward, which is unobserved. This is because the responses of LLMs are generally not evaluated by their users. Our goal is to learn to route each prompt to the expert with the highest mean reward. This is impossible without feedback. Therefore, we make a realistic assumption that we get access to limited feedback that is as good as that from humans. This could be human feedback or a stronger LLM used as an *LLM judge* (Li et al., 2024b,a; LLM-as-a-Judge, 2025). This feedback is expensive, either because of human labor or computation cost, and thus we can only use it $m \ll T$ times, where m is determined by the available budget. The tradeoff between rewards and feedback is not clear a priori. Therefore, we maximize rewards under the constraint on feedback rather than a linear combination of the two quantities. Finally, this problem is inherently online since user prompts are revealed only upon interaction, making offline solutions infeasible. Routing decisions must be made sequentially in real time, with no prior knowledge of the prompt distribution.

We solve our problem as a *contextual bandit* (Langford and Zhang, 2008; Li et al., 2010; Lattimore and Szepesvari, 2019) with K experts, where each LLM is an expert and the context is an embedding of the prompt. The main difference from all prior work on contextual bandits is that only *m rewards out of T can be observed*. The agent can decide *what to observe* and *when to observe it*, as long as it could have observed it before. The main challenge in the algorithm design is doing it at a near-optimal rate in a regret minimization setting. The control over what to observe and when to observe it differentiates our work from other bandit settings that involve partial observations and we discuss these extensively in Appendix C. The online setting with limited feedback on LLM response quality differentiates our work from prior LLM optimization approaches, which are typically

studied in offline settings or do not directly optimize response quality. We also discuss them in Appendix C. We make the following contributions:

1. We study the full-information setting (Section 3), where the agent can observe the rewards of all experts in any past round. The key idea in our algorithm is to progressively observe rewards of the experts with the highest information gain. The algorithm is computationally efficient and its regret is $\tilde{O}(dT/\sqrt{m})$, which matches our lower bound up to $\sqrt{d}$. Note that the bound is $\tilde{O}(\sqrt{T})$ when $m = \Omega(T)$.

2. We also study the bandit setting (Section 4), where the agent can only observe at a certain round the reward of the expert who has generated the response for that round. The key idea in our algorithm is to progressively observe rewards of the experts with the highest information gain, separately for each LLM expert. The algorithm is computationally efficient and its regret is $\tilde{O}(dT\sqrt{K/m})$. The additional $\sqrt{K}$ factor comparing to the full-information setting is due K times less feedback.

3. We extend the bandit setting to varying expert costs where the price of evaluating the responses of experts is non-uniform. Due to space constraints, this result has been moved to Appendix A.

4. We evaluate all algorithms empirically on the Nectar dataset (Zhu et al., 2024) and show that they can learn a high-quality routing agent for many popular LLMs experts, such as GPT-3.5, GPT-4, LLaMA, and Mistral (Section 5).

Technical Novelty: A key challenge in our analysis is bounding the reward gap between the optimal and selected experts for prompts without feedback. Unlike standard linear bandits, limited feedback prevents immediate updates to the covariance matrix, making it difficult to bound regret using standard techniques. To address this in the full-information setting, we introduce a novel feedback strategy: at regular intervals, we select past prompts that maximize the determinant of the covariance matrix. This approach quickly reduces uncertainty in important directions and allows us to bound regret. We further show that periodic feedback selection performs nearly as well as hindsight-optimal choices. In the bandit setting, we extend this approach by maintaining separate confidence sets per expert to ensure accurate regret guarantees.

Outline: In Section 2, we state our problem and define our model. In Sections 3 and 4, we describe our algorithms for the full-information and bandit settings, respectively, along with providing theoretical guarantees. In Appendix A, we extend our bandit algorithm to the setting with varying costs of evaluating experts. In Section 5, we provide empirical results using our algorithms. Finally, in Appendix B, we provide detailed proofs for all our results.

2 Problem Setting

Notation: We denote by $[K]$ the set $\{1, 2, \ldots, K\}$. We denote scalars and vectors by lowercase letters (say x). We denote matrices and fixed global parameters by capital letters (say X). For a vector x, x_i denotes its ith entry. For an indexed vector x_j, $x_{j,i}$ denotes its ith entry. We let $||x||_A$ be the weighted 2-norm $\sqrt{x^T A x}$ with respect to a positive semi-definite matrix A. We define the corresponding inner product as $\langle x, y\rangle_A = x^T A y$. 0_d and I_d are the zero vector and identity matrix in d dimensions, respectively. $\mathcal{N}(0_d, \Sigma)$ is the Gaussian distribution in d-dimensions with zero mean and covariance matrix Σ. For a matrix A, we write A_i to denote its ith column. Let $\mathcal{B}^d \equiv \{x \in \mathbb{R}^d \mid ||x||_2 \leq 1\}$ be the unit ball in d dimensions.

Problem Formulation: We introduce our setting as a variant of a classic linear bandit (Abbasi-Yadkori et al., 2011). We have T rounds and K LLM *experts*. Each expert is indexed by $a \in [K]$ and associated with an unknown *parameter vector* $\theta_a \in \mathcal{B}^d$. At each round $t \in [T]$, a *prompt* arrives and we denote it by $x_t \in L \cdot \mathcal{B}^d$ for some $L > 0$. The prompt is then treated as *context* at round t.

Given a prompt $x_t \in \mathbb{R}^d$, the algorithm chooses an expert $a_t \in [K]$ and obtains its response. In the linear model, the *expected reward* for the response of expert a in round t is $\langle \theta_a, x_t\rangle$. We denote the vector of all stochastic *rewards* in round t by $r_t = (r_{t,a})_{a=1}^K$ and define each reward as

$$r_{t,a} = \langle \theta_a, x_t\rangle + \eta_{a,t}\,. \tag{1}$$

We assume that the noise $\{\eta_{a,t}\}_{a\in[K],t\in[T]}$ is independent, both across the experts and rounds, and sub-Gaussian with a variance proxy 1. The linear model is realistic since the reward is computed as a linear head on top of a frozen transformer embedding. Specifically, x_t is the embedding of the

prompt produced by a transformer, and θ_a is the linear head for expert a. This is standard in reward modeling, with the only distinction being that we do not fine-tune the embeddings.

Further, at each round t, the context x_t arrives, the algorithm selects an expert a_t, and feedback is optionally requested afterward. This ordering reflects a key constraint: feedback is expensive and the decision of whether to query it is made after the action, as part of the exploration strategy. One may ask whether prompt evaluations could instead be performed before selecting a_t, by looking back at past contexts similar to x_t and leveraging their outcomes to make a more informed decision. However, doing so would require additional feedback queries at every round, violating the budget constraint $m \ll T$. Our formulation instead leverages past feedback frugally: routing decisions are made with whatever has been learned up to round t, without retroactive evaluation of past prompts triggered by the current context. We consider this alternative as the *NoLookBack* baseline in our experiments.

Limited Feedback: In our setting, the feedback is expensive (Li et al., 2024b), due to time or monetary constraints[1], and thus limited. In the classic linear bandit, the noisy reward is typically observed partially or completely at each round t. *The key difference in our setting is that the reward vector generated at any round t is not immediately observed.* However, feedback—provided as observations of rewards—is crucial for improving the selection of experts over successive rounds (Lattimore and Szepesvari, 2019).

The algorithm has a budget of $m < T$ observations, meaning that it can observe at most m rewards, either whole vectors or its entries, across the entire time horizon. Crucially, the algorithm can adaptively decide at which rounds to obtain feedback. If, at round $t \in [T]$, the algorithm decides to collect feedback, it may choose any round $s_t \leq t$ and observe the noisy reward vector r_{s_t} or its entry $r_{s_t,a_{s_t}}$. Many prior works in the bandit literature studied limited feedback (Appendix C). The main difference in our setting is that the algorithm not only selects an expert at each round but also chooses, exploiting the problem structure, when and for which past prompts to collect feedback.

The expert with the highest expected reward in round t given the prompt x_t is

$$a_t^\star = \arg\max_{a \in [K]} \langle \theta_a, x_t \rangle .$$

We define the *cumulative regret* in T rounds as

$$\mathsf{Reg}(T) = \mathbb{E}\Big[\sum_{t=1}^{T} \langle \theta_{a_t^\star}, x_t \rangle - \sum_{t=1}^{T} \langle \theta_{a_t}, x_t \rangle \Big] , \quad (2)$$

where the expectation is with respect to the randomness of the algorithm. In the remainder of the paper, we study different forms of limited feedback that can be obtained in practice and analyze regret in these settings.

3 Full-Information Feedback

We start with the full-information setting, where the agent can observe rewards of all experts at any past round. While this setting is simpler than the bandit setting in Section 4, it already exhibits basic properties of its algorithm design, that the problem can be solved by choosing observations with the highest information gain at regular time intervals. The former guarantees sub-linear regret and the latter allows us to trivially satisfy the observation budget.

3.1 Algorithm

Our algorithm `LimFullFeed` is presented in Algorithm 1 and we describe it next. The inputs are K LLM experts, the number of rounds T, the feedback budget m, and a hyperparameter λ. We also initialize all statistics for tracking reward models of all experts online (Section 2), such as the common covariance matrix V_0 and *ordinary least squares (OLS)* estimates $\widehat{\theta}_{0,a}$ for all experts.

In any round $t \in [T]$, `LimFullFeed` observes a prompt x_t and chooses the best expert a_t based on its estimated mean reward in Equation (3). `LimFullFeed` may decide to obtain feedback. The feedback is obtained when $\lfloor b(t-1) \rfloor < \lfloor bt \rfloor$ holds. Roughly speaking, this happens every T/m rounds since $b = m/T$. By following this strategy, we trivially guarantee that the observation budget constraint is satisfied. When the algorithm decides to obtain feedback, it can select any past unobserved round. Any round can be observed at most once because two repeated observations would be identical and hence not independent. More formally, we denote the set of past rounds where the feedback was previously obtained by $S_t \subseteq [t]$ and let the algorithm choose any round in $[t] \setminus S_t$. We denote

[1]The cost and latency of using LLMs as judges depends on the token counts and pricing of the service provider, such as Azure Databricks (Databricks, 2025a,b), or on the computational resources available to host such LLMs on premise.

Algorithm 1 `LimFullFeed`: Limited full-information feedback for expert selection.

1: Initialize $b \leftarrow m/T$, $V_0 \leftarrow \lambda I_d$, $\mathcal{S}_0 \leftarrow \emptyset$; and $y_{0,a} \leftarrow 0_d$, $\widehat{\theta}_{0,a} \leftarrow 0_d$ for all $a \in [K]$
2: **for** round $t = 1, \ldots, T$ **do**
3: Obtain prompt x_t and choose expert

$$a_t = \arg\max_{a \in [K]} \langle \widehat{\theta}_{t-1,a}, x_t \rangle \tag{3}$$

4: **if** $\lfloor b(t-1) \rfloor < \lfloor bt \rfloor$ **then**
5: Find most informative past observation

$$s_t = \arg\max_{\ell \in [t] \setminus \mathcal{S}_{t-1}} \mathsf{det}(V_{t-1} + x_\ell x_\ell^T) \tag{4}$$

6: Observe reward vector r_{s_t} and update all statistics

$$V_t \leftarrow V_{t-1} + x_{s_t} x_{s_t}^T, \quad \mathcal{S}_t \leftarrow \mathcal{S}_{t-1} + \{s_t\}$$
$$\forall a \in [K] : y_{t,a} \leftarrow y_{t-1,a} + r_{s_t,a} x_{s_t}, \quad \widehat{\theta}_{t,a} \leftarrow V_t^{-1} y_{t,a} \tag{5}$$

7: **else**
8: Update $V_t \leftarrow V_{t-1}$, $S_t \leftarrow S_{t-1}$; and $y_{t,a} \leftarrow y_{t-1,a}$, $\widehat{\theta}_{t,a} \leftarrow \widehat{\theta}_{t-1,a}$ for all $a \in [K]$

the chosen round by $s_t \in [t] \setminus S_t$ and the observed rewards by r_{s_t}. Since we are in the full-information setting, r_{s_t} is a vector of the rewards of all experts. After the feedback is obtained, all statistics are updated, such as the OLS estimates for all experts, $\widehat{\theta}_{t,a}$ in Equation (8).

The technical novelty in our algorithm design is in how s_t is chosen. We consider all past prompts $\{x_s\}_{s \leq t}$ and choose the one that maximally increases the determinant of the covariance matrix in Equation (4). The intuitive idea behind this choice is that this increases all eigenvalues of the covariance matrix uniformly and thus leads to uniformly decreasing confidence intervals in all previously observed directions, encoded by the embeddings of the prompts. In turn, this yields sub-linear regret.

We would like to comment on two more aspects of `LimFullFeed`. First, the best expert in Equation (3) is chosen using the mean reward estimate. This is because in the full-information setting, all experts are trained on the same past prompts and hence have the same covariance matrices. In the bandit setting (Section 4), we account for non-uniform data collection across the experts. Second, a naive implementation of `LimFullFeed` has a $O(d^3)$ per-round time complexity, due to inverting $d \times d$ matrices and computing their determinants. This can be reduced to $O(d^2)$ by using the Sherman-Morrison formula for the former and the matrix determinant lemma for the latter.

3.2 Main Results

Our goal is to minimize regret under the constraint of obtaining feedback at most $m \leq T$ times. The constraint is satisfied trivially (Section 3.1). Therefore, we only need to prove a regret bound. We start by borrowing standard assumptions from linear bandit analyses (Lattimore and Szepesvári, 2020, Chapter 19).

Assumption 3.1. All expert parameters $\{\theta_a\}_{a \in [K]}$ satisfy $||\theta_a||_2 \leq 1$. All prompts $\{x_t\}_{t=1}^T$ satisfy $||x_t||_2 \leq L$. We also assume that $\max_{a,b \in [k], t \in [T]} \langle \theta_a - \theta_b, x_t \rangle \leq 1$.

Theorem 3.2. *Choose any K experts, d features, horizon T, budget $m < T$, and $\lambda > 0$. Suppose that Assumption 3.1 holds. Then the regret of* `LimFullFeed` *is* $\mathsf{Reg}(T) = O(dTm^{-1/2} \log(TL))$.

Due to space constraints, we only sketch the proof. The detailed proof is in Appendix B.1

Proof Sketch. Consider a fixed expert $a \in [K]$ with unknown θ_a and a prompt x_t at round t. Using the OLS estimate $\widehat{\theta}_{t,a}$ for θ_a, the error in estimated mean reward given by $\langle \theta_a - \widehat{\theta}_{t,a}, x_t \rangle$ scales as $||x_t||_{V_t^{-1}}$. This error can be rewritten as $\log \mathsf{det}(V_t + x_t x_t^T) - \log \mathsf{det} V_t$. In standard linear bandits, we update the covariance matrix $V_{t+1} = V_t + x_t x_t^T$. Hence, we can add up the error over all rounds to obtain a telescoping sum. However, in our setting with limited feedback, V_t is not updated at every round and therefore the above does not hold. The key idea in our proof stems from Line 7 in `LimFullFeed` where we look back to update V_t with the prompt that increases its determinant most. Hence we can still bound $\log \mathsf{det}(V_t + x_t x_t^T) - \log \mathsf{det} V_t$ from above

by $\log \det V_{t+1} - \log \det V_t$. This upper bound is identical for all the rounds between two consecutive feedback and associated updates to the covariance matrix. Finally, we solicit feedback at regular intervals with interval length of T/m rounds. Therefore we still get a telescoping sum after adding the errors but the limited feedback leads to an extra multiplicative factor of T/m in the telescoping sum. This leads to an additional multiplicative factor of $\sqrt{T/m}$ in the regret bound. □

Note that regret does not depend on the number of experts K since in the full-information setting, we obtain responses for all experts jointly. Therefore, when $m = T$, we can obtain feedback for all rounds and the regret guarantee that is achieved is $O(d\sqrt{T}\log T)$. This is reminiscent of the standard regret guarantee achieved for linear bandits (Lattimore and Szepesvári, 2020). The additional cost of limited feedback arises in the form of a multiplicative factor of $\sqrt{T/m}$ leading to higher cost with lesser feedback. We also prove the following lower bound (detailed proof is in Appendix B.4).

Theorem 3.3. *Consider the online expert selection problem with limited full information feedback, d features, K experts, horizon T, feedback budget of $m < T$. Suppose all observations are Gaussian random variables with noise variance 1. Let prompts $\{x_t\}_{t=1}^T \in \{-1,+1\}^d$ and $\{\theta_a\}_{a\in[K]} \in \{-\delta,+\delta\}^d$ for $\delta = (md)^{-1/2}$. Then there exists an instance such that the regret incurred must satisfy*

$$\mathsf{Reg}(\mathsf{T}) \geq \frac{T\sqrt{d}\exp(-4)}{8\sqrt{m}}.$$

Discussion: Note that there is a gap of $\sqrt{d}$ between the upper and lower bounds. This gap stems from the fact that we do not know the prompts in advance and therefore, at all rounds t we need to bound $\langle \theta_a - \widehat{\theta}_{t,a}, x\rangle$ for all possible prompts $x \in \mathbb{R}^d$ and all experts $a \in [K]$. Such bounds are obtained in online settings with correlated observed random variables via a tail inequality on self-normalized martingales (Abbasi-Yadkori et al., 2012). Intuitively, ensuring the error bound is small for all vectors in the d-dimensional space leads to a union bound over d dimensions that in turn leads to the additional $\sqrt{d}$ factor in the upper bound. If on the other hand, the prompts $\{x_t\}_{t=1}^T$ was known, then we would only need a union bound over T vectors instead of all vectors in $\mathbb{R}^d$. This removes the additional $\sqrt{d}$ factor from the regret upper bound to make it tight up to logarithmic factors.

Next we argue that obtaining feedback at regular intervals leads to an optimal regret bound in m. Suppose that the algorithm knew the prompts $\{x_t\}_{t=1}^T$ in advance and could also decide the order in which to obtain feedback. Then the algorithm would choose a subset S of m most-informative prompts, obtain feedback, and learn the expert parameters from them. The regret of this approach would be $O(T \max_{t\in[T]} ||x_t||_{V^{-1}})$, where $\max_{t\in[T]} ||x_t||_{V^{-1}}$ is the maximum confidence interval width and $V = \sum_{i\in S} x_i x_i^T$. An optimal solution to this problem is known as the G-optimal optimal design (Lattimore and Szepesvári, 2020, Chapter 21) and its maximum confidence interval width is $O(\sqrt{d/m})$. This leads to a regret of $O(T\sqrt{d/m})$ over T rounds and completes our argument.

We can consider a simpler algorithm which does not look back and use past prompts. The algorithm requests feedback for the input prompts at the rounds it decided to obtain feedback. Notice that such a baseline algorithm (we will call `NoLookBack`) might suffer linear regret if an adversary provides prompts at the feedback rounds that are in an orthogonal subspace to the prompts in remaining rounds.

4 Bandit Setting

In this setting, we consider bandit feedback. This means that unlike the full information setting, here an agent can observe the reward of only one expert, that which was chosen to generate the response of a prompt at any past round. Each expert might receive feedback a different number of times (unlike in the full-information setting). We are able to guarantee sub-linear cumulative regret in this setting as well, while satisfying the budget on feedback trivially, by algorithm design.

4.1 Algorithm

Here we describe our proposed algorithm `LimBanFeed` (Algorithm 2) that chooses experts to generate response in the bandit feedback setting. Since different experts can potentially get feedback different number of times, `LimBanFeed` adaptively chooses the experts and the corresponding past prompts (routed to them) to collect the feedback. The key idea is to observe rewards for those experts with highest information gain, by

Algorithm 2 `LimBanFeed`: Limited bandit feedback for expert selection.

1: Initialize $z \leftarrow T/m$, $V_{0,a} \leftarrow \lambda I_d$, $\mathcal{S}_{0,a} \leftarrow \emptyset$; and $y_{0,a} \leftarrow 0_d$, $\widehat{\theta}_{0,a} \leftarrow 0_d$ for all $a \in [K]$.
2: **for** rounds $t = 1, 2, \ldots, T$ **do**
3: Obtain prompt x_t, choose expert a_t and increase its counter by 1.

$$a_t = \arg\max_{a \in [K]} \langle \widehat{\theta}_{t-1,a}, x_t \rangle + \beta \, ||x_t||_{V_{t-1,a}^{-1}} \tag{6}$$
$$n_{a_t} \leftarrow n_{a_t} + 1$$

4: **if** $n_{a_t} \geq z$ **then**
5: Find the most informative past observation for the expert a_t. Reset its counter.

$$s_t = \arg\max_{s \in \{s \in [t] : a_s = a\} \setminus S_{t-1,a}} \mathsf{det}(V_{t-1,a_t} + x_\ell x_\ell^T) \tag{7}$$
$$n_{a_t} \leftarrow 0$$

6: Observe reward r_{s_t,a_t} for the expert a_t and update the corresponding statistics for a_t.

$$V_{t,a_t} \leftarrow V_{t-1,a_t} + x_{s_t} x_{s_t}^T, \quad \mathcal{S}_{t,a_t} \leftarrow \mathcal{S}_{t-1,a_t} + \{s_t\}$$
$$\forall a \in [K] : y_{t,a_t} \leftarrow y_{t-1,a_t} + r_{s_t,a_t} x_{s_t}, \quad \widehat{\theta}_{t,a_t} \leftarrow V_{t,a_t}^{-1} y_{t,a_t} \tag{8}$$

7: **else**
8: Update $V_{t,a} \leftarrow V_{t-1,a}$, $y_{t,a} \leftarrow y_{t-1,a}$, $\widehat{\theta}_{t,a} = \widehat{\theta}_{t-1,a}$ and $S_{t,a} \leftarrow S_{t-1,a}$ for all experts $a \in [K]$.

appropriately considering confidence bounds in reward estimates.

`LimBanFeed` takes as input the same parameters as `LimFullFeed` along with the additional hyperparameter β corresponding to the confidence width. In the bandit setting, we initialize the covariance matrix $V_{0,a}$ along with other hyperparameters separately for each expert $a \in [K]$. In Line 2, we also initialize a variable z which is set to be the average number of rounds between feedback, namely T/m.

At each round t, the prompt x_t arrives as context. Given the input prompt x_t, `LimBanFeed` chooses the expert a_t in (6) to generate the desired response by computing the upper confidence bound on the reward for each expert, given the OLS parameter estimate, and picking the one with the highest upper confidence reward. We maintain a counter n_a for every expert $a \in [K]$ that keeps track of the number of times an expert is chosen for generating response. Once an expert has been used z times, feedback is solicited for that expert by choosing a past prompt (it had served) appropriately and the counter is reset.

Clearly, experts that are chosen more frequently will get more feedback. We show here that the choice of z helps respect the overall feedback budget of m rounds. The number of rounds where an expert $a \in [K]$ has been used to generate a response in the time horizon T is $|S_{T,a}|$, hence the number of times the ath expert has been evaluated is $\lfloor |S_{T,a}| / z \rfloor$. Setting $z \geq \frac{T}{m}$ helps satisfy the feedback budget m:

$$\sum_{a \in [K]} \lfloor \frac{|S_{T,a}|}{z} \rfloor \leq \sum_{a \in [K]} \frac{|S_{T,a}|}{z} = \frac{T}{z} \leq m \implies z \geq \frac{T}{m}.$$

For getting feedback for an expert a, `LimBanFeed` chooses a prompt index among the past prompts that maximizes the increase in the determinant of the covariance matrix for the expert a (see (7)). We only choose prompts that have not yet been evaluated.

We update the OLS estimate of the parameter vector $\widehat{\theta}_{t,a}$ for expert a_t (see (8)) by using the observed scalar reward $r_{s_t,a}$ for the chosen prompt.

4.2 Main Results

We show our main result below:

Theorem 4.1. *Consider the online expert selection problem with limited bandit feedback, K experts, d features, horizon T and feedback budget $m < T$. Suppose Assumption 3.1 is true. Then for any $\lambda > 0$ and $\beta = \sqrt{\lambda} + \sqrt{6 \log T + d \log(1 + TL^2/d)}$, the regret of* `LimBanFeed` *is*

$$\mathsf{Reg}(T) = O(dTK^{1/2} m^{-1/2} \log(TL))$$

The detailed proof is provided in Appendix B.2, but we discuss our key ideas here.

Proof Sketch. In the bandit setting, the covariance matrix for each expert is updated separately. However, the key idea for strategically choosing the

prompt in history, given the expert, remains the same as in the full-information setting. To obtain feedback at round t for expert a, we look back and choose the prompt for which the determinant of the covariance matrix increases the most, allowing us to suitably bound the error $||x_t||_{V_{t-1,a}^{-1}}$ from above. Additionally, in our analysis, we use the fact that each expert is updated after being used z = $\lceil T/m \rceil$ times. This ensures that the error from an expert decreases over time, with the rate depending on how frequently the expert is invoked. □

Note that the number of feedback observations in the full-information setting is K times that of the bandit setting for the same number of feedback rounds. Therefore, roughly speaking, in contrast to the full-information setting, the regret guarantee with the bandit feedback has an additional multiplicative factor of $\sqrt{K}$. Further, when $m = T$ that is, we can obtain feedback for all rounds, Algorithm 2 achieves a regret guarantee of $O(d\sqrt{TK}\log(TL))$. As in the full-information setting, the cost of limited feedback is a multiplicative factor of $\sqrt{T/m}$. Next, we prove a lower bound on the regret in the bandit feedback setting (detailed proof is in Appendix B.4):

Theorem 4.2. *Consider the online expert selection problem with limited bandit feedback, d features, K experts, horizon T, feedback budget of $m < T$. Suppose all observations are Gaussian random variables with noise variance 1. Let prompts $\{x_t\}_{t=1}^T \in \{-1,+1\}^d$ and $\{\theta_a\}_{a\in[K]} \in \{-\delta,+\delta\}^d$ for $\delta = (md)^{-1/2}$. Then there exists an instance such that the regret incurred must satisfy*

$$\mathsf{Reg(T)} \geq \frac{TK\exp(-4)}{8\sqrt{m}}$$

There is a gap of $d/\sqrt{K}$ between the upper and lower bounds in the bandit setting. This arises because the lower-bound analysis reduces to hypothesis testing in a K-dimensional subspace. If we were to ignore the feature structure and treat the experts as arms in a standard multi-armed bandit setting, we could establish a lower bound of $\Omega(T\sqrt{K/m})$, which is looser by a factor of $\sqrt{K}$. Similar to the full-information setting, the upper bound includes an additional factor of $\sqrt{d}$ due to the algorithm's lack of knowledge about the prompt vectors, requiring a union bound over all vectors in the d-dimensional space. This extra factor can be eliminated if the algorithm has prior knowledge of the prompts. The remaining gap of $\sqrt{d/K}$ persists because the lower bound analysis is based on hypothesis testing with K vectors and may be further improved. It will be an interesting direction of future work to improve the lower bound in the bandit setting.

5 Experiments

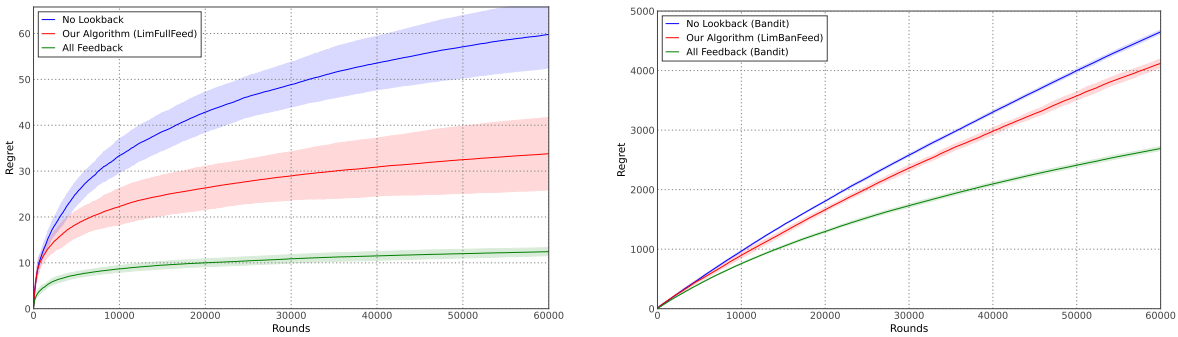
We evaluate our approach on large-scale LLM routing benchmarks.

Datasets and Baselines. We consider two widely used routing benchmarks: RouterBench(Hu et al., 2024) and Nectar(Zhu et al., 2024). Both datasets provide prompt-level evaluations across multiple LLM experts, enabling controlled simulation of online routing with ground-truth rewards. We follow a standard protocol and sample a subset of prompts to simulate an online interaction stream.

We compare against two representative baselines. **NoLookBack** requests feedback only for the current round when querying, without leveraging past data. This corresponds to a naive strategy that ignores the structure of the problem. **AllFeedback** assumes access to feedback at every round and serves as a performance upper bound.

We simulate an online routing process over a fixed horizon T, with a feedback budget $m \ll T$. At each round, the algorithm selects an expert based on the observed context, and feedback is collected according to the algorithm's strategy. We report cumulative regret with respect to the best expert in hindsight. All results are averaged over 5 runs. Details are provided in Appendix D.

Main Results. Figure 1 and 2 show the regret as a function of the number of rounds in both the full-information and bandit settings on RouterBench and Nectar. We observe that `LimFullFeed` consistently outperforms `NoLookBack`. Interpreting regret as the cumulative number of suboptimal decisions, the gap between the two methods becomes substantial over time. In particular, by the end of the horizon ($T = 60000$), `LimFullFeed` achieves noticeably lower regret than `NoLookBack`, indicating a significantly smaller number of mistakes. This improvement highlights the importance of selecting informative prompts when querying feedback. By looking back and choosing past observations that maximize information gain, `LimFullFeed` reduces uncertainty more efficiently and accelerates learning. In contrast, `NoLookBack`



(a) Full Information Setting

(b) Bandit Setting

Figure 1: Results comparing our approaches to the other methods (RouterBench): (a) `NoLookBack` (evaluates prompt at round when feedback is requested) and (b) `AllFeedback` (observes feedback at all rounds). Clearly, `LimFullFeed` has better regret guarantees than `NoLookBack` by careful choice of feedback. `AllFeedback` suffers the smallest regret due to more data.

Table 1: Cumulative regret with respect to feedback budget m on Nectar (K=6, d=40, T=60000).

	Full-information		Bandit	
m	LimFullFeed	NoLookBack	LimBanFeed	NoLookBack
1000	672.9 ± 37.4	$\mathbf{631.6 \pm 22.1}$	$\mathbf{3583.4 \pm 68.0}$	3789.6 ± 30.4
2000	441.4 ± 42.6	$\mathbf{408.0 \pm 26.2}$	$\mathbf{3084.1 \pm 25.6}$	3580.0 ± 73.3
5000	$\mathbf{214.1 \pm 7.7}$	218.7 ± 10.6	$\mathbf{2295.7 \pm 42.3}$	3020.7 ± 17.2
10000	$\mathbf{128.0 \pm 8.7}$	129.0 ± 11.0	$\mathbf{1839.1 \pm 29.1}$	2538.4 ± 22.2
20000	$\mathbf{73.6 \pm 6.0}$	75.8 ± 6.1	$\mathbf{1567.9 \pm 13.4}$	2090.7 ± 21.2

queries feedback only at the current round, which leads to less informative data collection and slower error reduction. We also observe that the variance of `AllFeedback` is consistently smaller across rounds. This is expected, as `AllFeedback` has access to substantially more feedback, resulting in more stable estimates and reduced variability across runs.

The same trend holds in the bandit setting, where `LimBanFeed` consistently achieves lower regret than `NoLookBack`, with the gap increasing over time. As in the full-information setting, `AllFeedback` exhibits lower variance due to the larger amount of available data.

Finally, regret in the full-information setting is uniformly lower than in the bandit setting. This aligns with our theoretical analysis, as the full-information setting provides richer feedback per observation, enabling faster reduction of uncertainty.

Effect of the Feedback Budget. Table 1 shows that regret decreases as m increases, consistent with the $\sqrt{T/m}$ dependence in our theory. In the bandit setting, `LimBanFeed` consistently outperforms `NoLookBack`, with the gap widening for larger m. In the full-information setting, the gap is smaller for small m, since each feedback reveals all experts, but `LimFullFeed` becomes competitive as m increases.

Runtime Analysis. Table 2 shows that both methods scale approximately linearly with T. `LimBanFeed` scales linearly with K, while `LimFullFeed` is largely independent of K due to its shared covariance structure.

6 Conclusions

In this paper, we tackled adaptive prompt routing to LLM experts in an online learning setting with limited feedback. We framed this as a contextual bandit problem and introduced new algorithms for both full-information and bandit feedback settings. Our key innovation is showing when and where to observe feedback, which is expert-dependent in the bandit setting. Theoretical guarantees on regret show near-optimal learning rates, and empirical evaluations confirm our methods' effectiveness. Future work could explore dynamic expert availability, varying prompt distributions, and lower bounds in the bandit setting to refine regret minimization strategies.

Limitations

We assume that feedback for past prompts can be obtained retrospectively, which may require storing model outputs and relying on noisy human or LLM-based judgments. Our experiments use static benchmarks to simulate online routing, which may not fully capture non-stationary user behavior or deployment constraints.

References

Yasin Abbasi-Yadkori, David Pal, and Csaba Szepesvari. 2011. Improved algorithms for linear stochastic bandits. In *Advances in Neural Information Processing Systems 24*, pages 2312–2320.

Yasin Abbasi-Yadkori, David Pal, and Csaba Szepesvari. 2012. Online-to-confidence-set conversions and application to sparse stochastic bandits. In *Artificial Intelligence and Statistics*, pages 1–9. PMLR.

Pranjal Aggarwal, Aman Madaan, Ankit Anand, Srividya Pranavi Potharaju, Swaroop Mishra, Pei Zhou, Aditya Gupta, Dheeraj Rajagopal, Karthik Kappaganthu, Yiming Yang, and 1 others. 2023. Automix: Automatically mixing language models. *arXiv preprint arXiv:2310.12963*.

Rajeev Agrawal, Demosthenis Teneketzis, and Venkatachalam Anantharam. 1989. Asymptotically efficient adaptive allocation schemes for controlled i.i.d. processes: Finite parameter space. *IEEE Transactions on Automatic Control*, 34(3):258–267.

Shipra Agrawal and Nikhil Devanur. 2016. Linear contextual bandits with knapsacks. In *Advances in Neural Information Processing Systems 29*.

Artificial Analysis. 2025. LLM leaderboard.

Jean-Yves Audibert and Sebastien Bubeck. 2010. Regret bounds and minimax policies under partial monitoring. *Journal of Machine Learning Research*, 11(94):27852836.

Gabor Bartok, Dean Foster, David Pal, Alexander Rakhlin, and Csaba Szepesvari. 2014. Partial monitoring - classification, regret bounds, and algorithms. *Mathematics of Operations Research*, 39(4):967–997.

Gabor Bartok and Csaba Szepesvari. 2012. Partial monitoring with side information. In *Proceedings of the 23rd International Conference on Algorithmic Learning Theory*, pages 305–319.

Leonardo Cella, Massimiliano Pontil, and Claudio Gentile. 2021. Best model identification: A rested bandit formulation. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 1362–1372. PMLR.

Nicolo Cesa-Bianchi, Gabor Lugosi, and Gilles Stoltz. 2005. Minimizing regret with label efficient prediction. *IEEE Transactions on Information Theory*, 51(6):2152–2162.

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

Corinna Cortes, Giulia DeSalvo, Claudio Gentile, Mehryar Mohri, and Scott Yang. 2018. Online learning with abstention. In *Proceedings of the 35th International Conference on Machine Learning*.

Xiangxiang Dai, Jin Li, Xutong Liu, Anqi Yu, and John Lui. 2024. Cost-effective online multi-llm selection with versatile reward models. *arXiv preprint arXiv:2405.16587*.

Databricks. 2025a. How quality, cost, and latency are assessed by agent evaluation.

Databricks. 2025b. Mosaic ai agent evaluation.

Dujian Ding, Ankur Mallick, Chi Wang, Robert Sim, Subhabrata Mukherjee, Victor Rühle, Laks VS Lakshmanan, and Ahmed Hassan Awadallah. Hybrid llm: Cost-efficient and quality-aware query routing. In *The Twelfth International Conference on Learning Representations ICLR 2024, Vienna, Austria, May 7-11, 2024*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Dylan J Foster, Akshay Krishnamurthy, and Haipeng Luo. 2019. Model selection for contextual bandits. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Surya Narayanan Hari and Matt Thomson. 2023. Tryage: Real-time, intelligent routing of user prompts to large language model. *arXiv preprint arXiv:2308.11601*.

David Helmbold and Sandra Panizza. 1997. Some label efficient learning results. In *Proceedings of the 10th Annual Conference on Computational Learning Theory*, pages 218–230.

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

Keke Huang, Yimin Shi, Dujian Ding, Yifei Li, Yang Fei, Laks Lakshmanan, and Xiaokui Xiao. 2025. Thriftllm: On cost-effective selection of large language models for classification queries. *arXiv preprint arXiv:2501.04901*.

Mohammad Reza Karimi, Nezihe Merve Gürel, Bojan Karlaš, Johannes Rausch, Ce Zhang, and Andreas Krause. 2021. Online active model selection for pre-trained classifiers. In *International Conference on Artificial Intelligence and Statistics*, pages 307–315. PMLR.

Abbas Kazerouni, Mohammad Ghavamzadeh, Yasin Abbasi-Yadkori, and Benjamin Van Roy. 2017. Conservative contextual linear bandits. In *Advances in Neural Information Processing Systems 30*.

Branislav Kveton, Csaba Szepesvari, Zheng Wen, and Azin Ashkan. 2015a. Cascading bandits: Learning to rank in the cascade model. In *Proceedings of the 32nd International Conference on Machine Learning*.

Branislav Kveton, Zheng Wen, Azin Ashkan, and Csaba Szepesvari. 2015b. Combinatorial cascading bandits. In *Advances in Neural Information Processing Systems 28*, pages 1450–1458.

John Langford and Tong Zhang. 2008. The epoch-greedy algorithm for contextual multi-armed bandits. In *Advances in Neural Information Processing Systems 20*, pages 817–824.

Tor Lattimore and Csaba Szepesvari. 2019. *Bandit Algorithms*. Cambridge University Press.

Tor Lattimore and Csaba Szepesvári. 2020. *Bandit algorithms*. Cambridge University Press.

Dawei Li, Bohan Jiang, Liangjie Huang, Alimohammad Beigi, Chengshuai Zhao, Zhen Tan, Amrita Bhattacharjee, Yuxuan Jiang, Canyu Chen, Tianhao Wu, Kai Shu, Lu Cheng, and Huan Liu. 2024a. From generation to judgment: Opportunities and challenges of llm-as-a-judge. *arXiv preprint arXiv:2411.16594*.

Haitao Li, Qian Dong, Junjie Chen, Huixue Su, Yujia Zhou, Qingyao Ai, Ziyi Ye, and Yiqun Liu. 2024b. Llms-as-judges: A comprehensive survey on llm-based evaluation methods. *arXiv preprint arXiv:2412.05579*.

Lihong Li, Wei Chu, John Langford, and Robert Schapire. 2010. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th International Conference on World Wide Web*.

Shuai Li, Baoxiang Wang, Shengyu Zhang, and Wei Chen. 2016. Contextual combinatorial cascading bandits. In *Proceedings of the 33rd International Conference on Machine Learning*, pages 1245–1253.

LLM-as-a-Judge. 2025. LLM as a judge.

Keming Lu, Hongyi Yuan, Runji Lin, Junyang Lin, Zheng Yuan, Chang Zhou, and Jingren Zhou. 2024. Routing to the expert: Efficient reward-guided ensemble of large language models. In *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.

Mistral AI. 2025. Mistral AI.

Quang H Nguyen, Duy C Hoang, Juliette Decugis, Saurav Manchanda, Nitesh V Chawla, and Khoa D Doan. 2024. Metallm: A high-performant and cost-efficient dynamic framework for wrapping llms. *arXiv preprint arXiv:2407.10834*.

Isaac Ong, Amjad Almahairi, Vincent Wu, Wei-Lin Chiang, Tianhao Wu, Joseph E Gonzalez, M Waleed Kadous, and Ion Stoica. 2024. Routellm: Learning to route llms with preference data. *arXiv preprint arXiv:2406.18665*.

OpenAI. 2025. OpenAI API pricing.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Abraham Toluwase Owodunni and Chris Chinenye Emezue. 2023. Koya: A recommender system for large language model selection. In *4th Workshop on African Natural Language Processing*.

Filip Radlinski, Robert Kleinberg, and Thorsten Joachims. 2008. Learning diverse rankings with multi-armed bandits. In *Proceedings of the 25th International Conference on Machine Learning*, pages 784–791.

Guillem Ramirez, Alexandra Birch, and Ivan Titov. 2024. Optimising calls to large language models with uncertainty-based two-tier selection. *arXiv preprint arXiv:2405.02134*.

Marija Šakota, Maxime Peyrard, and Robert West. 2024. 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*, pages 606–615.

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

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

Long Tran-Thanh, Archie Chapman, Alex Rogers, and Nicholas Jennings. 2012. Knapsack based optimal policies for budget–limited multi–armed bandits. In *Proceedings of the 26th AAAI Conference on Artificial Intelligence*, pages 1134–1140.

Aaron Tucker, Caleb Biddulph, Claire Wang, and Thorsten Joachims. 2023. Bandits with costly reward observations. In *Proceedings of the 39th Conference on Uncertainty in Artificial Intelligence*.

Claire Vernade, Alexandra Carpentier, Tor Lattimore, Giovanni Zappella, Beyza Ermis, and Michael Bruckner. 2020. Linear bandits with stochastic delayed feedback. In *Proceedings of the 37th International Conference on Machine Learning*.

Xingyu Wu, Yan Zhong, Jibin Wu, Bingbing Jiang, Kay Chen Tan, and 1 others. 2024. Large language model-enhanced algorithm selection: towards comprehensive algorithm representation. International Joint Conference on Artificial Intelligence.

Yifan Wu, Roshan Shariff, Tor Lattimore, and Csaba Szepesvari. 2016. Conservative bandits. In *Proceedings of the 33rd International Conference on Machine Learning*, pages 1254–1262.

Yu Xia, Fang Kong, Tong Yu, Liya Guo, Ryan A Rossi, Sungchul Kim, and Shuai Li. 2024. Which llm to play? convergence-aware online model selection with time-increasing bandits. In *Proceedings of the ACM on Web Conference 2024*, pages 4059–4070.

Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. 2023. C-pack: Packaged resources to advance general chinese embedding. *Preprint*, arXiv:2309.07597.

Zhengyuan Zhou, Renyuan Xu, and Jose Blanchet. 2019. Learning in generalized linear contextual bandits with stochastic delays. In *Advances in Neural Information Processing Systems 32*.

Banghua Zhu, Evan Frick, Tianhao Wu, Hanlin Zhu, Karthik Ganesan, Wei-Lin Chiang, Jian Zhang, and Jiantao Jiao. 2024. Starling-7B: Improving helpfulness and harmlessness with RLAIF. In *Proceedings of the 1st Conference on Language Modeling*.

Appendix

A Bandit Feedback with Variable Costs

We can also extend Algorithm 2 to the setting where the K experts have different costs of being evaluated and there is a budget on the overall cost. More precisely, suppose the total cost budget is m and the cost of evaluation of the jth expert is z_j. Without loss of generality, we can assume that $z_1 \geq z_2 \geq \cdots \geq z_K$. For the expert j with cost z_j, we evaluate it after it has been used $z_j \cdot \widetilde{z}$ times for some $\widetilde{z} > 0$. In other words, the experts is evaluated with a frequency that is proportional to their cost. In Algorithm 2, each expert had the same cost and therefore, the strategy of evaluation is identical for every expert. Now, given the budget on the feedback rounds m, $\widetilde{z}$ can be computed as follows: Again, recall that the number of rounds where the expert a has been used to generate a response in the entire time horizon is $|S_{T,a}|$. In that case, the number of times the ath expert has been evaluated is $\lfloor |S_{T,a}| / (z_j \cdot \widetilde{z}) \rfloor$. We need the total cost to be smaller than m, hence,

$$\sum_j \lfloor \frac{|S_{T,a}|}{z_j \cdot \widetilde{z}} \rfloor \cdot z_j \leq \sum_j \frac{|S_{T,a}|}{z_j \cdot \widetilde{z}} \cdot z_j$$

$$= \sum_j \frac{|S_{T,a}|}{\widetilde{z}} = \frac{T}{\widetilde{z}} \leq m \implies \widetilde{z} \geq \frac{T}{m}.$$

Now the entire analysis proceeds in the same way as the proof for Theorem 4.1. The only difference lies in the non-uniform rate at which each expert is updated. More precisely, instead of a common interval length of z for every expert $a \in [K]$ as in Algorithm 2, now we have an interval length of $z_a \cdot \widetilde{z}$ for the ath expert. Going through the same set of calculations, we can show that the regret guarantee in this case is going to be

$$\mathsf{Reg}(T) = O\Big(Tm^{-1/2}\sqrt{d\beta\Big(\sum_j z_j\Big)\log\Big(\frac{d\lambda + m}{d}\Big)}\Big)$$

Substituting the value of $\beta_{m/K}$ as in Theorem 4.1, the final regret guarantee can be written as

$$\mathsf{Reg}(T) = O(dTm^{-1/2}(\sum_j z_j)^{1/2}\log T).$$

Notice that when all the costs are 1, we get the same regret guarantee as in Theorem 4.1. Further, as is common in this framework, the additional cost of limited feedback comes in the form of a multiplicative factor of $\sqrt{T/m}$.

B Detailed Proofs

B.1 Proof of Theorem 3.2

Proof of Theorem 3.2. Notations: We start by setting up some notations. Let $O_t \subseteq [t]$ denote the set of rounds until (including) round t in which the algorithm chooses to get feedback. Note that $O_t \subseteq O_{t'}$ for all $t' \geq t$. For the set of prompts evaluated until round t, recall S_t to be the set of past rounds when the prompts were provided as input to the algorithm. Let us denote $\tau_t \triangleq \max\{\ell \in [t-1] : \ell \in O_t \wedge (\ell - 1) \notin O_t\}$ denote the last round before round t when the experts were updated. We denote $a \wedge b = \min(a, b)$.

We can decompose the RHS in (2) as follows:

$$\sum_{t=1}^{T}\langle \theta_{a_t^\star}, x_t\rangle - \sum_{t=1}^{T}\langle \theta_{a_t}, x_t\rangle = \sum_{t=1}^{T}\langle \theta_{a_t^\star} - \widehat{\theta}_{t,a_t^\star}, x_t\rangle + \sum_{t=1}^{T}\langle \widehat{\theta}_{t,a_t^\star} - \widehat{\theta}_{t,a_t}, x_t\rangle + \sum_{t=1}^{T}\langle \widehat{\theta}_{t,a_t} - \theta_{a_t}, x_t\rangle.$$

Note that the second term is negative. This is because, by choice of our algorithm, we have picked the expert index a_t at each round t such that $\langle \widehat{\theta}_{t,a_t^\star} - \widehat{\theta}_{t,a_t}, x_t\rangle \leq 0$. Therefore, we can ignore the second

term when bounding the LHS from above and then apply the Cauchy Schwarz inequality Note that the covariance matrices for all experts remain identical throughout the time horizon. We denote the covariance matrix of all experts at the beginning of round t by V_{t-1}. Recall that feedback is solicited at every interval of T/m rounds. Hence, the covariance matrix for all experts remains unchanged between two distinct feedback collections. From standard guarantees on confidence sets regarding the OLS estimator in the linear model (see Chapter 19 in (Lattimore and Szepesvári, 2020)), we have the following at every round $t \in [\mathsf{T}]$ and every expert $a \in [K]$ with probability at least $1 - o(T^{-2})$:

$$\left|\left|\theta_t - \widehat{\theta}_{t,a}\right|\right|_{V_{t-1}} \leq \beta \text{ where } \beta = \sqrt{\lambda} + \sqrt{6\log T + d\log\left(\frac{d\lambda + mL^2}{d\lambda}\right)}. \tag{9}$$

Denote the event $\mathcal{E}$ where (9) is true for all rounds $t \in [T]$ and all experts $a \in [K]$. Moving forward, we can now condition on the above high probability event $\mathcal{E}$. We now show the following for any round t:

$$\langle \theta_{a_t^\star}, x_t\rangle - \langle \theta_{a_t}, x_t\rangle \leq \left|\left|\theta_{a_t^\star} - \widehat{\theta}_{t,a_t^\star}\right|\right|_{V_{t-1}} ||x_t||_{V_{t-1}^{-1}} + \left|\left|\widehat{\theta}_{t,a_t} - \theta_{a_t}\right|\right|_{V_{t-1}} ||x_t||_{V_{t-1}^{-1}} \leq 2\beta\, ||x_t||_{V_{t-1}^{-1}} = 2\beta\, ||x_t||_{V_{\tau_t}^{-1}} \tag{10}$$

The last step follows from the fact that by definition τ_{t,a_t} is the last round (before round t) when the expert a_t had been updated. Note that for each round t, we also have $\langle \theta_{a_t^\star}, x_t\rangle - \langle \theta_{a_t}, x_t\rangle \leq 1$ from the assumption in theorem statement. Therefore, we can combine to show

$$\sum_{t=1}^{T}\langle \theta_{a_t^\star}, x_t\rangle - \sum_{t=1}^{T}\langle \theta_{a_t}, x_t\rangle \leq 2\beta \sum_{t=1}^{T}\left(1 \wedge ||x_t||_{V_{t-1}^{-1}}\right) \leq 2\sqrt{T\beta \sum_{t=1}^{T}\left(1 \wedge ||x_t||^2_{V_{t-1}^{-1}}\right)}$$

where the final step follows from application of Cauchy-Schwarz inequality. Now, continuing from (10), for any round t, we use the fact that for any $u \geq 0$, we have $u \wedge 1 \leq 2\log(1+u)$:

$$\begin{aligned} 1 \wedge ||x_t||^2_{V_{\tau_t}^{-1}} &\leq \log\left(1 + ||x_t||^2_{V_{\tau_t}^{-1}}\right) = \log\det\left(I + V_{\tau_t}^{-1/2} x_t x_t^T V_{\tau_t}^{-1/2}\right) \\ &= \log\det\left(V_{\tau_t} + x_t x_t^T\right) - \log\det V_{\tau_t} \end{aligned}$$

Now, consider a round $t \in O_T$ where feedback has been observed for some prompt in the history. As we defined before, τ_t corresponds to the previous round in history where feedback has been observed. From Algorithm 1, recall that s_t was the prompt chosen at round $t \in O_T$ for feedback following which we had updated $V_{t-1} = V_{\tau_t} + x_{s_t} x_{s_t}^T$. Further, recall that the prompt index s_t at round t was chosen in the following way:

$$s_t = \mathsf{argmax}_{\ell \in [t] \setminus \mathcal{S}_{t-1}} \det(V_{t-1} + x_\ell x_\ell^T) = \mathsf{argmax}_{\ell \in [t] \setminus \mathcal{S}_{t-1}} \det(V_{\tau_t} + x_\ell x_\ell^T)$$

implying that the prompt s_t increases the determinant of the covariance matrix V_{τ_t} the most. Thus, for every round $r \in [\tau_t + 1, t]$, we must have $\det(V_r + x_r x_r^T) = \det(V_{\tau_t} + x_r x_r^T) \leq \det(V_{\tau_t} + x_{s_t} x_{s_t}^T)$. Hence, we have

$$\sum_{r \in [\tau_t + 1, t]} \log\det(V_r + x_r x_r^T) \leq \lceil T/m \rceil \log\det(V_{\tau_t} + x_{s_t} x_{s_t}^T) = \lceil T/m \rceil \log\det V_{t-1}.$$

Therefore, using the fact that the covariance matrix remains unchanged for T/m rounds, we can write

$$\begin{aligned} \sum_{t=1}^{T}\left(1 \wedge ||x_t||^2_{V_{\tau_t}^{-1}}\right) &\leq \lceil T/m \rceil \sum_{t \in O_T}\left(\log\det V_t - \log\det V_{\tau_t}\right) + Tm^{-1} \\ &= \lceil T/m \rceil \left(\log\det V_T - \log\det V_0\right) + Tm^{-1}. \end{aligned}$$

Notice that $V_{t-1} = \sum_{t \in O_T} x_{s_t} x_{s_t}^T$, $|O_T| = m$ (since we are getting feedback for m rounds). Moreover, for all prompts $\{x_t\}_{t \in [T]}$, we have $\mathsf{Tr}(x_t x_t^T) = ||x_t||_2^2 \leq 1$ since all prompts are within the unit ball. Now, we use the AM-GM inequality and linearity of trace operation to show

$$\mathsf{det} V_T \leq \Big(d^{-1}\mathsf{Tr}(V_T)\Big)^d \leq \Big(d^{-1}\Big(\mathsf{Tr}(V_0) + m\Big)\Big)^d$$

We have $\mathsf{Tr}(V_0) \leq d\lambda$ and therefore, we bound the regret conditioned on the event $\mathcal{E}$ (denote by $\mathsf{Reg}(T) \mid \mathcal{E}$) as

$$\mathsf{Reg}(T) \mid \mathcal{E} = O\Big(Tm^{-1/2}\sqrt{d\beta \log\Big(\frac{d\lambda + m}{d}\Big)}\Big)$$

When the event $\mathcal{E}$ is false, then the regret can be bounded by the $2T$ (worst-case). Therefore the final regret can be written as

$$\mathsf{Reg}(T) = \mathsf{Reg}(T) \mid \mathcal{E} + 2T\,\mathrm{Pr}(\mathcal{E}^c) = \mathsf{Reg}(T) \mid \mathcal{E} + o(T^{-1}).$$

Now, we can substitute the value for β to obtain the final theorem statement. □

B.2 Proof of Theorem 4.1

Proof of Theorem 4.1. Notations: We start by introducing new notations. For any expert $a \in [K]$, let $O_{t,a} \subseteq [t]$ denote the set of rounds until round t in which the algorithm has opted to obtain a feedback for expert a. As before, for any $a \in [K]$, we have $O_{t,a} \subseteq O_{t',a}$ for all $t' \geq t$. Similar to full-information, for the set of prompts evaluated until round t, define S_t to be the set of past rounds when the prompts were provided as input to the algorithm. We denote $\tau_{t,a} \triangleq \max\{\ell \in [t-1] : \ell \in O_{t,a} \wedge (\ell - 1) \notin O_{t,a}\}$ denote the last round before round t when the expert a was updated.

As in the proof of Theorem 3.2, we will exploit standard guarantees on confidence sets regarding the OLS estimator in the linear model. We have the following at every round $t \in [\mathsf{T}]$ and every expert $a \in [K]$ with probability at least $1 - o(T^{-2})$:

$$\Big|\Big|\theta_t - \widehat{\theta}_{t,a}\Big|\Big|_{V_{t-1,a}} \leq \beta \text{ where } \beta = \sqrt{\lambda} + \sqrt{6\log T + d\log\Big(\frac{d\lambda + mL^2}{d\lambda}\Big)}. \tag{11}$$

Denote the event $\mathcal{E}$ where (11) is true for all rounds $t \in [T]$ and all experts $a \in [K]$. For any prompt x_t at round t, the parameter vector in the confidence region that leads to the highest reward is given by

$$\widetilde{\theta}_{t,a} = \sup_{\phi \in \left|\left|\phi - \widehat{\theta}_{t,a}\right|\right|_{V_{t-1,a}} \leq \beta} \langle \phi, x_t \rangle.$$

and therefore, the expert a_t to generate the response is chosen as $\max_{a \in [K]} \langle \widetilde{\theta}_{t,a}, x_t \rangle$. It is interesting to note that the bilinear optimization problem has a nice closed form expression given by (see Sec 19.3.1 in (Lattimore and Szepesvári, 2020))

$$a_t = \arg\max_{a \in [K]} \langle \widehat{\theta}_{t,a}, x_t \rangle + \beta \cdot ||x_t||_{V_{t-1,a}^{-1}}$$

Now, we can decompose the RHS in (2) for any round $t \in [T]$ as follows:

$$\langle \theta_{a_t^\star}, x_t \rangle - \langle \theta_{a_t}, x_t \rangle \leq \langle \widehat{\theta}_{a_t^\star}, x_t \rangle + \beta\,||x_t||_{V_{t,a_t^\star}^{-1}} - \langle \theta_{a_t}, x_t \rangle \leq \langle \widehat{\theta}_{a_t}, x_t \rangle + \beta\,||x_t||_{V_{t,a_t}^{-1}} - \langle \theta_{a_t}, x_t \rangle$$

For the first inequality we bound the reward for the expert $a_t^\star$ from above by the upper confidence bound on the reward. The second inequality follows from $\langle \widehat{\theta}_{t,a_t^\star} - \widehat{\theta}_{t,a_t}, x_t \rangle \leq 0$. This is because, by choice of our algorithm, we have picked the expert index a_t at each round t having the largest upper confidence reward on prompt x_t.

The covariance matrices for each of the experts are updated separately in the bandit feedback setting. Recall that the noisy reward for only a single expert is observed whenever the algorithm opts for a feedback. We denote the covariance matrix of expert $a \in [K]$ at round t by $V_{t-1,a}$. Again, recall that feedback is solicited for expert a after the expert a has been used to generate a response z times. Now, we can show the following for any round t

$$\begin{aligned}\langle \theta_{a_t^\star}, x_t\rangle - \langle \theta_{a_t}, x_t\rangle &\leq \beta\, ||x_t||_{V_{t,a_t}^{-1}} + \left|\left|\widehat{\theta}_{t,a_t} - \theta_{a_t}\right|\right|_{V_{t,a_t}} ||x_t||_{V_{t,a_t}^{-1}} \\ &\leq 2\beta\, ||x_t||_{V_{t,a_t}^{-1}}\end{aligned}$$

Now, we need to bound from above the final term on the RHS in terms of the individual experts as follows:

$$\begin{aligned}\sum_{t=1}^{T} ||x_t||_{V_{t,a_t}^{-1}} &= \sum_{a=1}^{K}\sum_{t=1}^{T} ||x_t||_{V_{t,a_t}^{-1}} \mathbf{1}[a_t = a] \\ &\leq \sum_{a=1}^{K}\sum_{t=1}^{T} \Big(1 \wedge ||x_t||_{V_{t,a_t}^{-1}} \mathbf{1}[a_t = a]\Big) \\ &\leq \sqrt{T\sum_{a=1}^{K}\sum_{t=1}^{T} \Big(1 \wedge ||x_t||^2_{V_{t,a_t}^{-1}}\Big)\mathbf{1}[a_t = a]\Big)}\end{aligned}$$

In the pre-final step, we used that in each round t, we also have $\langle \theta_{a_t^\star}, x_t\rangle - \langle \theta_{a_t}, x_t\rangle \leq 1$ from the assumption in theorem statement. In the final step, we used the Cauchy-Schwarz inequality. Note that there are only T terms in the RHS since at each round, only a single expert is chosen. Consider a particular expert $a \in [K]$ - it is chosen for obtaining feedback after being used for generating responses z times since the last time feedback was obtained for expert a.

At any round t, recall $\tau_{t,a} = \max\{\ell \in [t-1] : \ell \in O_{t,a} \wedge (\ell - 1) \notin O_{t,a}\}$ denote the last round from t when the expert a was chosen for obtaining feedback (after the counter for expert a reached z). For any round $t \in O_{t,a}$, as per our notation, recall that $V_{t-1,a}$ is the covariance matrix for the expert a at round t. Then $\tau_{t-1,a}$ is the previous round when the expert a was updated. For any round t, we can use the fact that for any $u \geq 0$, we have $u \wedge 1 \leq 2\log(1+u)$,

$$\begin{aligned}1 \wedge ||x_t||^2_{V_{t,a_t}^{-1}} &\leq \log\left(1 + ||x_t||^2_{V_{t,a_t}^{-1}}\right) = \log\mathsf{det}\Big(I + V_{t,a_t}^{-1/2} x_t x_t^T V_{t,a_t}^{-1/2}\Big) \\ &= \log\mathsf{det}\Big(V_{t,a_t} + x_t x_t^T\Big) - \log\mathsf{det} V_{t,a_t} = \log\mathsf{det}\Big(V_{\tau_{t,a_t},a_t} + x_t x_t^T\Big) - \log\mathsf{det} V_{\tau_{t,a_t},a_t}\end{aligned}$$

Now for a fixed expert $a \in [K]$, consider a round $t \in O_{T,a}$ where feedback has been observed for some prompt in the history of expert a. As we defined before, $\tau_{t-1,a}$ corresponds to the previous round in history (relative to round t) where feedback has been observed for expert a. From Algorithm 2, recall that s_t was the prompt chosen at round $t \in O_{t,a}$ for feedback following which we had updated $V_{t-1,a} = V_{\tau_{t-1},a} + x_{s_t} x_{s_t}^T$. Further, recall that the prompt index s_t at round t was chosen in the following way:

$$s_t = \underset{s \in \{s \in [t] : a_s = a\} \setminus S_{t-1,a}}{\arg\max} \mathsf{det}(V_{t-1,a} + x_s x_s^T) = \underset{s \in \{s \in [t] : a_s = a\} \setminus \mathcal{S}_{t-1,a}}{\arg\max} \mathsf{det}(V_{\tau_{t-1,a},a} + x_x x_s^T)$$

implying that the prompt s_t was responded to by the expert a_t and further, the prompt s_t increases the determinant of the covariance matrix $V_{\tau_{t-1,a},a}$ the most. Thus, for every round $r \in [\tau_{t-1,a}+1, t]$ satisfying $\mathbf{1}[a_r = a]$, we must have $\mathsf{det}(V_{r,a} + x_r x_r^T) = \mathsf{det}(V_{\tau_{t-1,a},a} + x_r x_r^T) \leq \mathsf{det}(V_{\tau_{t-1,a},a} + x_{s_t} x_{s_t}^T)$. Hence, we have

$$\sum_{r \in [\tau_{t-1}+1, t]} \log\mathsf{det}(V_{r,a} + x_r x_r^T) \leq c \cdot \log\mathsf{det}(V_{\tau_{t-1,a},a} + x_{s_t} x_{s_t}^T) = \lceil T/m \rceil \log\mathsf{det} V_{t-1,a}.$$

In the last step, we used the threshold on z that would meet the feedback budget m. As we had shown, it is sufficient to have $z \geq T/m$ so that the total amount of feedback rounds is at most m. Therefore, by combining all of these, we get

$$\sqrt{T\sum_{a=1}^{K}\sum_{t=1}^{T}\Big(1 \wedge ||x_t||^2_{V^{-1}_{t,a_t}}\Big)\mathbf{1}[a_t = a]\Big)} = \sqrt{T\sum_{a=1}^{K}\Big(z\sum_{t\in O_{T,a}}\Big(\log\mathsf{det}V_{t-1,a} - \log\mathsf{det}V_{\tau_{t-1,a},a}\Big) + c\Big)} \tag{12}$$

$$\leq Tm^{-1/2}\sqrt{\sum_{j=1}^{K}\Big(\log\mathsf{det}(V_{t-1,a} - \log\mathsf{det}V_{0,a}\Big)} + \sqrt{cKT}. \tag{13}$$

For a fixed expert $a \in [K]$, notice that $V_{t-1,a} = \sum_{t\in O_{T,a}} x_{s_t}x_{s_t}^T$, $|O_T| \leq m$ (since we are getting feedback for at most m rounds for any expert). Moreover, for all prompts $\{x_t\}_{t\in[T]}$, we have $\mathsf{Tr}(x_t x_t^T) = ||x_t||_2^2 \leq 1$ since all prompts are within the unit ball. Now, we use the AM-GM inequality and linearity of trace operation to show

$$\mathsf{det}V_{t-1,a} \leq \Big(d^{-1}\mathsf{Tr}(V_{t-1,a})\Big)^d \leq \Big(d^{-1}\Big(\mathsf{Tr}(V_{0,a}) + m\Big)\Big)^d$$

We have $\mathsf{Tr}(V_{0,a}) \leq d\lambda$ and therefore, we bound the regret conditioned on the event $\mathcal{E}$ (denote by $\mathsf{Reg}(T) \mid \mathcal{E}$) as

$$\mathsf{Reg}(T) \mid \mathcal{E} = O\Big(TK^{1/2}m^{-1/2}\sqrt{d\beta\log\Big(\frac{d\lambda + m}{d}\Big)}\Big)$$

$$\mathsf{Reg}(T) = \mathsf{Reg}(T) \mid \mathcal{E} + 2T\Pr(\mathcal{E}^c) = \mathsf{Reg}(T) \mid \mathcal{E} + o(T^{-1}).$$

Now, we can substitute the value for β to obtain the final theorem statement.

□

B.3 Proof of Theorem 3.3

Proof of Theorem 3.3. Consider number of rounds T, a feedback budget of m (full-information feedback) and all observations to be Gaussian random variables with unit variance. While constructing our instances, let us denote the set of prompts we choose from to be $\mathcal{X} \subset \{-1, +1\}^d$ and the set of model features to be $\theta \subset \{-\delta, +\delta\}^d$ for some $\delta > 0$. For any two vectors (x, y), we can define the Hamming distance $d_h(x, y) = \sum_{i=1}^{d}\mathbf{1}[\text{sign}(x_i) \neq \text{sign}(y_i)]$.

Fix a vector $x \in \{-1, +1\}^d$. Now, we define an instance in the full-information expert selection setting as follows: for each of the T rounds, a single prompt $x \in \mathcal{X}$ is going to be demonstrated and the set of expert parameters in this instance is given by $\theta \equiv \{\theta \in \{-\delta, +\delta\}^d \mid d_h(\theta, x) \leq 1\}$. In other words, the set of allowed experts have feature embedding which are within a Hamming distance of 1 from the prompt embedding vector x. Given this environment, . Recall that the expert chosen at round t is denoted by a_t. Now, we define d alternate learning instances as follows: in each alternate instance, a vector x' satisfying $d_h(x, x') = 1$ is demonstrated at all rounds while the set of model features $\mathcal{T}(x)$ remain the same. Therefore, the set of prompts across the $d + 1$ instances is $\mathcal{X} \equiv \{y \in \{-1, +1\}^d \mid d_h(y, x) \leq 1\}$. Since the distinction between the instances is only in the prompt (the expert features θ remain same in all instances), we denote by $\mathbb{P}_y$ and $\mathbb{E}_y$ the probability and expectation of events under a fixed policy for instance defined by prompt $y \in \mathcal{X}$.

Consider the environment defined by the prompt x. Let us define the following event which is true when the number of times the sign of the ith entry of the prompt and algorithm output (observation of

history's actions) differs is at least $T/2$

$$\mathcal{E}_{x,i} = \mathbf{1}[\sum_{i=1}^{T} \text{sign}(x_i) \neq \text{sign}(a_{ti}) \geq T/2]$$

and the corresponding probability to be $p_{x,i} = \mathbb{P}_x(\mathcal{E}_{x,i})$. Now consider the alternate instance with prompt $x' \in \mathcal{X}$ where $x'_j = x_j$ for all $j \neq i$ and $x'_i = -x_i$. Therefore, x and x' have a Hamming distance of 1 and differ in sign at the index i. Note that the event $\mathcal{E}_{x',i}$ is the complement of the event $\mathcal{E}_{x,i}$. We have

$$p_{x,i} + p_{x',i} = \mathbb{P}_x(\mathcal{E}_{x,i}) + \mathbb{P}_{x'}(\mathcal{E}^c_{x,i}) \geq \frac{1}{2}\exp\Big(-\mathsf{D}_{\mathsf{KL}}(\mathbb{P}_x||\mathbb{P}_{x'})\Big)$$

Above, we used the Bretagnolle Huber inequality (see (Lattimore and Szepesvári, 2020)) and D_{KL} is the Kullback-Leibler Divergence. It is given by

$$\mathsf{D}_{\mathsf{KL}}(\mathbb{P}_x||\mathbb{P}_{x'}) = m\mathbb{E}_x\Big[\mathsf{D}_{\mathsf{KL}}\Big(\mathcal{N}(\theta\cdot x, I_{d+1})||\mathcal{N}(\theta\cdot x', I_{d+1})\Big]\Big) = m\,\big|\big|\theta(x-x')\big|\big|_2^2 = m\delta^2(d+1).$$

In that case, for the value of $\delta = 1/\sqrt{m(d+1)}$, we get $p_{x,i} + p_{x',i} \geq \exp(-1)/2$. Now, we want to extend the analysis to environments defined by prompts $y \in \mathcal{X} \setminus \{x\}$. Fix such a prompt y and define the event which is true when the number of times the sign of the ith entry of the prompt and the expert differs is at least $T/2$

$$\mathcal{E}_{y,i} = \mathbf{1}[\sum_{i=1}^{T} \text{sign}(y_i) \neq \text{sign}(a_{ti}) \geq T/2]$$

Without loss of generality, assume that the index where y and x differs in sign is $j \neq i$ (otherwise the alternate instance trivially becomes the environment defined by x). In that case, consider the alternate instance defined by prompt y' such that $\mathbf{y}'the = x_s$ for all $s \neq i$ and $\mathbf{y}'_i = -x_i$ (y' and x differs in sign at the index i). Therefore, y and y' have a Hamming distance of 2 and differ in sign at the indices i, j. However, note that the event $\mathcal{E}_{y',i}$ is the complement of the event $\mathcal{E}_{y,i}$. By a similar analysis as above, we have

$$p_{y,i} + p_{y',i} \geq \frac{1}{2}\exp\Big(-\mathsf{D}_{\mathsf{KL}}(\mathbb{P}_y||\mathbb{P}_{y'})\Big) = \frac{1}{2}\exp\Big(-m\,\big|\big|\theta(y-y')\big|\big|_2^2\Big) \geq \frac{1}{2}\exp\Big(-4m\delta^2(d+1)\Big).$$

In that case, for the value of $\delta = 1/\sqrt{m(d+1)}$, we have

$$p_{y,i} + p_{y',i} \geq \exp(-4)/2.$$

Since there are $d+1$ possibilities of the prompt x, we can have

$$\frac{1}{|\mathcal{X}|}\sum_{x\in\mathcal{X}}\sum_{i=1}^{d} p_{x,i} = \frac{1}{|\mathcal{X}|}\sum_{i=1}^{d}\sum_{x\in\mathcal{X}} p_{x,i} \geq \frac{d\exp(-4)}{4}.$$

Hence, there must exist a prompt $y \in \mathcal{X}$ for which $\sum_{i=1}^{d} p_{y,i} \geq d\exp(-4)/4$. For this instance defined by the prompt y and expert features θ, we have

$$\begin{aligned}
\mathsf{Reg}(T) &= \mathbb{E}_y\Big[\sum_{t=1}^{T}\langle y, \delta\cdot y - \mathbf{a}_t\rangle\Big] \\
&= \mathbb{E}_y\Big[\sum_{t=1}^{T}\langle y, \delta y - A_t\rangle\Big] = \mathbb{E}_y\Big[\sum_{t=1}^{T}\sum_{i=1}^{d} y_i\Big(\delta y_i - a_{ti}\Big)\Big] = 2\delta\mathbb{E}_y\Big[\sum_{t=1}^{T}\sum_{i=1}^{d}\mathbf{1}[\text{sign}(y_i)\neq\text{sign}(a_{ti})]\Big] \\
&= 2\delta\sum_{i=1}^{d}\mathbb{E}_y\Big[\sum_{t=1}^{T}\mathbf{1}[\text{sign}(y_i)\neq\text{sign}(a_{ti})]\Big] = \delta T\sum_{i=1}^{d}\mathbb{P}_{yi} \geq \frac{\delta dT\exp(-4)}{4} \geq \frac{T\sqrt{d}\exp(-4)}{8\sqrt{m}}.
\end{aligned}$$

This completes the proof of the theorem. □

B.4 Proof of Theorem 4.2

Proof of Theorem 4.2. The proof of the lower bound for the bandit setting follows on similar lines as the proof in Theorem 3.3. Consider number of rounds T, a feedback budget of m and number of experts to be K and all observations to be Gaussian random variables with unit variance. Note that in this setting, for a particular prompt, feedback is only provided to the expert that was used to generate a response for that prompt. As before, we denote the set of prompts to be $\mathcal{X} \subset \{-1, +1\}^d$ and the set of experts features to be $\theta \subset \{-\delta, +\delta\}^d$ for some $\delta > 0$. For any two vectors x, y, we can define the Hamming distance $d_h(x, y) = \sum_{i=1}^{d} \mathbf{1}[\text{sign}(x_i) \neq \text{sign}(y_i)]$.

Fix a prompt vector x to be the d-dimensional all ones vector. For simplicity assume that d is divisible by K. Now, we define an instance in the bandit feedback setting as follows: for each of the T rounds, the single prompt $x \in \mathcal{X}$ is going to be demonstrated at all rounds and the set of expert parameters is given by a fixed set of K vectors from the set $\theta \equiv \{\theta \in \{-\delta, +\delta\}^d \mid d_h(\theta, x) \leq 1\}$. Without loss of generality, let $\{\theta_a\}_{a \in [K]}$ be the set of parameter vectors which has its sign flipped from x in one of the first K entries.

Now, we can define K alternate learning instances as follows: in each alternate instance, a vector x' satisfying $d_h(x, x') = 1$ is demonstrated at all rounds while the set of expert features $\mathcal{T}(x)$ remain the same. Further x' has its sign changed only among the first K entries of x. Therefore the last $d - K$ indices are inconsequential and we basically have a K-dimensional problem. Now, we go through the same steps as in the proof of Theorem 3.3. However, since we are in the bandit feedback setting, we will get after using the Bretagnolle Huber inequality that the KL divergence between data distributions of any two instances is $O(\exp(-m\delta^2))$. Hence, substituting $\delta = 1/\sqrt{m}$ and then resuming the same steps gives us the following lower bound:

$$\mathsf{Reg(T)} \geq \frac{TK \exp(-4)}{8\sqrt{m}}.$$

□

C Related Work

There are many notions of limited feedback and resource constraints that have been studied in the past. Partial monitoring (Agrawal et al., 1989; Bartok and Szepesvari, 2012; Bartok et al., 2014) and learning to rank (Radlinski et al., 2008; Kveton et al., 2015a,b; Li et al., 2016) are bandit problems where the agent observes rewards of taken actions only partially. In our setting, the agent observes the reward fully but decides what to observe. In bandits with delayed feedback (Zhou et al., 2019; Vernade et al., 2020), the agent observes rewards of taken action with delays. The delay is not controlled by the agent. Our agent observes rewards with delays but decides what to observe, and thus controls the delay. Further, it is well known that the linear model in bandit algorithms can be updated lazily, whenever the determinant of the covariance matrix increases significantly (Section 5.1 in Abbasi-Yadkori et al. (2011)). When the model is updated, all past observations are used. In our setting, we update the model periodically but only use a subset of past observations selected by the agent. The knapsack is a popular way of modeling resource constraints in bandits (Tran-Thanh et al., 2012; Agrawal and Devanur, 2016). We have a resource constraint but it is significantly simpler. This is why we can minimize the regret greedily at a near-optimal rate by periodically taking the most uncertain action in the past. In conservative bandits (Wu et al., 2016; Kazerouni et al., 2017), the agent takes a greedy exploratory action after accumulating a sufficient exploratory budget. The regret bound involves an extra term due to taking the safe action. In our setting, the agent has more control because it can observe the reward of any past action. Therefore, the extra term does not appear. In online learning with abstention (Cortes et al., 2018), the agent decides whether to predict or abstain. When the agent abstains, it pays a fixed a cost. We incur regret in each round and decide which past rewards to observe.

Recently, model selection under resource constraints has been studied in the context of LLMs. Most techniques aim to reduce the cost of inference by choosing models (LLMs) appropriately. The cost of inference depends on input and output token length and can be either API level cost or the compute cost of hosting and serving LLM. Note that our notion of cost is different - we consider the cost of labeling the LLM responses through human or other forms of gold feedback.

A popular technique towards reducing costs is LLM cascading that invokes LLMs sequentially, progressing to higher cost LLMs, till the response is deemed satisfactory, often by another model (Chen et al., 2023; Aggarwal et al., 2023; Ramirez et al., 2024). More closely related to our bandit setting is LLM routing that aims to route queries directly to the appropriate model, requiring only a single inference (Shekhar et al., 2024; Šakota et al., 2024). One major difference between these works and ours is that while we aim to learn the appropriate models in an online manner (either full information setting or bandit setting) by selectively obtaining labels for certain queries, existing works mostly aim to predict the output quality or performance (either absolute or relative to each other) of the LLMs for given queries for routing them (Ding et al.; Shekhar et al., 2024; Shnitzer et al., 2023; Lu et al., 2024; Hari and Thomson, 2023; Ong et al., 2024; Šakota et al., 2024). The routing is based on the prediction and other considerations such as cost and latency.

Some recent works have modeled model selection with cost considerations as a multi armed bandit problem. MetaLLM (Nguyen et al., 2024) frames the reward as a linear combination of accuracy and cost of inference and proposes a bandit algorithm for learning the best arm. Again, the notion of cost and the modeling of the problem are both different in our setting. Recently, TI-UCB (Xia et al., 2024) predicts the increase of model performances due to training or finetuning and efficiently balances exploration and exploitation in model selection. (Dai et al., 2024) proposed CSMAB-V based on combinatorial multi-armed bandits to select a good LLM combination, which aims to balance cost and rewards. This is also a different setting compared to ours.

Other works such as (Huang et al., 2025) focused on selecting a set of LLMs under a given cost budget to maximize performance. (Owodunni and Emezue, 2023) proposed a recommender system approach called Koya for selecting the best LLM for a given task and language. There has also been work on online model selection with partial information. In particular, (Foster et al., 2019) investigated model selection under contextual bandit feedback whereas (Cella et al., 2021) framed online model selection as a rested bandit problem. Further, Karimi et al. (2021) leverages active learning to identify the best model from a pool of pre-

trained classifiers. Other recent work has focused on the orthogonal problem of leveraging LLMs for model selection (Wu et al., 2024). These works focus mostly on leveraging LLMs to capture the structural and semantic properties of the model for better selection.

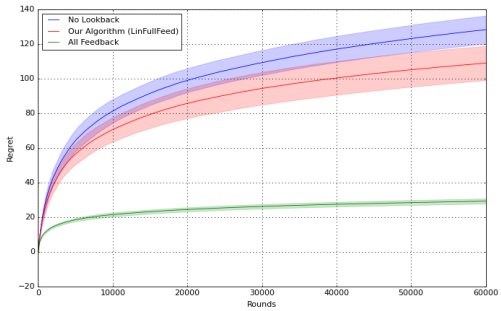
Online learning and bandits with observation budget have been studied extensively in the adversarial setting. This line of work was started by Helmbold and Panizza (1997). Cesa-Bianchi et al. (2005) proved a $T\sqrt{(\log K)/m}$ regret bound for the exponentially-weighted forecaster, where K is the number of experts. Audibert and Bubeck (2010) derived a $T\sqrt{K(\log K)/m}$ regret bound for the bandit setting. None of these works study the contextual setting. However, their scaling with the number of rounds T and observation budget m is similar to our work. The closest related work in the stochastic bandit setting is Tucker et al. (2023). This work proposes a contextual bandit algorithm but it is not analyzed. The algorithm maximizes the difference of rewards and costs, and is not applicable to our problem because the observation cost and reward are not comparable, and thus cannot be simply subtracted. We both propose a practical algorithm and analyze it.

D Implementation Details

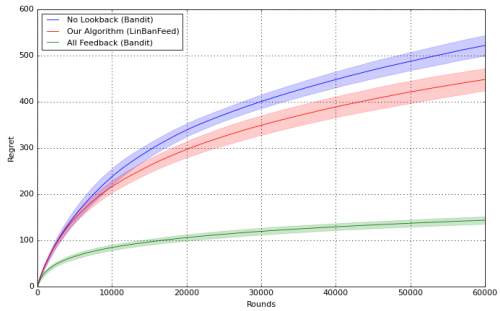
For our experiments, we use the RouterBench (Hu et al., 2024) which is a high-quality routing benchmark consisting a diverse set of 11 Large Language Models. There are approximately $405k$ prompts and for each prompt, responses from the 11 models are evaluated and normalized to 0 and 1. We randomly chose a subset of $10k$ prompts out of the total pool of $405k$ prompts. We found no model dominates the others more than a third of the time (Section 1). Thus, we have $K = 11$ experts in this setting. For each of the $10k$ prompts, we create its 384 dimensional embedding using the bge-small-en-v1.5 model (Xiao et al., 2023). Next we do PCA to reduce the dimension to 40, capturing 50% of the feature variance, resulting in a 10000×40 sized data matrix. In order to generate the expert parameters $\{\theta_a\}_{a\in[K]}$, we take the evaluated ranking from GPT-4 for each of $10k$ prompts and create a one-hot vector of dimension 11. In this vector, only the model whose response has been selected as best is assigned 1 and we have 0 everywhere else. Therefore we have a response vector of dimension 10000×11. Now we obtain the parameter $\{\theta_a\}_{a\in[K]}$ for each of the experts by computing the OLS estimate on this dataset.

Table 2: Runtime in seconds with respect K, d, and T on Nectar.

Axis	Value	LimBanFeed	LimFullFeed
K	2	2.67 ± 0.02	1.73 ± 0.34
	6	4.30 ± 0.03	1.49 ± 0.01
	10	5.93 ± 0.04	1.47 ± 0.01
	16	8.36 ± 0.02	1.49 ± 0.04
	20	9.97 ± 0.00	1.47 ± 0.00
d	8	4.25 ± 0.00	1.49 ± 0.04
	24	4.24 ± 0.00	1.46 ± 0.00
	40	4.27 ± 0.04	1.47 ± 0.00
	96	4.34 ± 0.00	1.51 ± 0.01
	128	4.34 ± 0.00	1.54 ± 0.04
T	500	0.41 ± 0.00	0.15 ± 0.00
	1,000	0.82 ± 0.00	0.29 ± 0.00
	5,000	4.27 ± 0.04	1.47 ± 0.00
	10,000	8.89 ± 0.02	2.98 ± 0.05
	15,000	14.35 ± 0.03	4.47 ± 0.03



(a) Full Information Setting



(b) Bandit Setting

Figure 2: Results comparing our approaches to the other methods (Nectar): (a) `NoLookBack` (evaluates prompt at round when feedback is requested) and (b) `AllFeedback` (observes feedback at all rounds). Clearly, `LimFullFeed` has better regret guarantees than `NoLookBack` by careful choice of feedback. `AllFeedback` suffers the smallest regret due to more data.