Online Multi-LLM Selection via Contextual Bandits Under Unstructured Context Evolution

Manhin Poon[1], **Xiangxiang Dai**[2], **Xutong Liu**[3], **Fang Kong**[4], **John C.S. Lui**[2], **Jinhang Zuo**[1*]

[1]City University of Hong Kong
[2]The Chinese University of Hong Kong
[3]University of Washington
[4]Southern University of Science and Technology
manhpoon4-c@my.cityu.edu.hk, xxdai23@cse.cuhk.edu.hk, xutongl@uw.edu, kongf@sustech.edu.cn
cslui@cse.cuhk.edu.hk, jinhang.zuo@cityu.edu.hk

Abstract

Large language models (LLMs) exhibit diverse response behaviors, costs, and strengths, making it challenging to select the most suitable LLM for a given user query. We study the problem of adaptive multi-LLM selection in an online setting, where the learner interacts with users through multi-step query refinement and must choose LLMs sequentially without access to offline datasets or model internals. A key challenge arises from unstructured context evolution: the prompt dynamically changes in response to previous model outputs via a black-box process, which cannot be simulated, modeled, or learned. To address this, we propose the first contextual bandit framework for sequential LLM selection under unstructured prompt dynamics. We formalize a notion of myopic regret and develop a LinUCB-based algorithm that provably achieves sublinear regret without relying on future context prediction. We further introduce budget-aware and positionally-aware (favoring early-stage satisfaction) extensions to accommodate variable query costs and user preferences for early high-quality responses. Our algorithms are theoretically grounded and require no offline fine-tuning or dataset-specific training. Experiments on diverse benchmarks demonstrate that our methods outperform existing LLM routing strategies in both accuracy and cost-efficiency, validating the power of contextual bandits for real-time, adaptive LLM selection.

Extended version — https://arxiv.org/abs/2506.17670

1 Introduction

Large Language Models (LLMs) have demonstrated remarkable capabilities across a wide range of tasks, from open-ended dialogue to complex reasoning and mathematical problem solving (Bubeck et al. 2023; Han et al. 2025; Zhu et al. 2025). However, these capabilities come with significant trade-offs: more powerful models tend to be slower and more expensive, while faster, cheaper models often lack robustness or reasoning ability (Yuan et al. 2025; Wang et al. 2025a). In practice, no single model is universally optimal across all inputs or user preferences (Anil et al. 2023; Dai et al. 2025).

This growing heterogeneity has motivated the deployment of *multi-LLM systems*, where multiple LLMs with different strengths (e.g., accuracy, latency, cost) are available, and a routing or selection mechanism determines which one to use for a given user query. Such multi-model setups are increasingly adopted in both academic research and industrial applications, where balancing performance and efficiency is critical (Tang et al. 2025; Murali et al. 2025). A lightweight model like Mistral may suffice for factual lookups, while more complex tasks might require high-end models like GPT-4 (Achiam et al. 2023). Selecting the right LLM at the right time can yield substantial improvements in cost-effectiveness and user satisfaction (Chen et al. 2025; Dai et al. 2024).

Existing approaches to multi-LLM routing (Nguyen et al. 2024; Dai et al. 2024; Aggarwal et al. 2024; Wang et al. 2025b) often rely on static policies, supervised learning on offline datasets, or single-step decision-making. However, real-world user interactions are often *sequential and adaptive*. Users refine their queries based on prior answers, leading to multi-step interactions where the system can revise its LLM choices dynamically. For example, a user might ask a factual question, follow up with a clarification request, and then pose a deeper reasoning task. Each step introduces new contextual information, typically derived from the model's earlier responses, which influences subsequent model selection.

To support this type of dynamic, multi-step interaction, we propose a new framework for *online multi-LLM selection under unstructured context evolution*. In our setting, the system interacts with a user over a sequence of steps: at each step, it receives a prompt context, potentially incorporating previous model responses, and must select an LLM to generate the next reply. Crucially, the prompt evolves over time according to a *black-box transformation* that may include concatenation, rewriting, user edits, or proprietary formatting. Because this evolution process is unknown, non-differentiable, and potentially stochastic, conventional planning-based approaches such as reinforcement learning (RL) are ill-suited (Zhang et al. 2025). In effect, modeling how a prompt evolves itself would require reasoning at the level of an LLM, rendering the transition dynamics unlearnable in practice.

To address the challenges of multi-LLM decision-making, online adaptation, and unstructured prompt dynamics, we

*Corresponding author.

adopt a contextual bandit framework. Rather than modeling full interaction trajectories, we decompose the process into a series of myopic decisions, where the system selects the most suitable LLM based solely on the current prompt context. This sidesteps the need to simulate or learn the black-box context evolution function and enables continual improvement from feedback. As illustrated in Figure 1, at each step the system observes a prompt, selects an LLM, receives feedback, and updates its strategy accordingly—without predicting future prompts. We formalize this per-step optimization objective using myopic regret, which compares each LLM choice against the best possible selection for the current context. Building on this formulation, we develop a LinUCB-based algorithm that maintains confidence-aware estimates of each model's performance across varying prompt contexts. To handle practical deployment constraints, we further extend the algorithm to account for stochastic per-query costs and budget constraints, and introduce a heuristic that prioritizes high-quality responses early in the interaction to reflect user positional bias. Together, our framework offers a principled online solution for adaptive, cost-aware multi-LLM selection.

The key contributions of this work are:

- We introduce the first contextual bandit framework for multi-step LLM selection under *unstructured context evolution*—a realistic and underexplored setting that captures the adaptive nature of interactive user–LLM interactions.
- We propose a Greedy LinUCB algorithm for per-step LLM selection and establish a sublinear regret bound with respect to *myopic regret*. Our results show that efficient online learning is achievable even without modeling how prompts evolve over time.
- We extend our method to handle *stochastic cost constraints* and *position-sensitive utility*, designing budget-aware and position-aware variants that improve decision quality in cost-limited settings and prioritize early user satisfaction in multi-turn interactions.
- We conduct extensive experiments across diverse benchmarks, demonstrating that our algorithms consistently outperform existing routing methods in both accuracy and cost-efficiency. The results further confirm the value of context evolution and the effectiveness of lightweight online adaptation.

Overall, our work provides a principled and lightweight foundation for orchestrating multiple LLMs in open-ended, multi-step environments. By using contextual bandits under unstructured prompt evolution, we demonstrate that practical and scalable multi-LLM systems can be built without relying on offline training or complex environment simulation. Due to space constraints, proofs and additional experimental details are deferred to the full version.

2 Related Work

Model Routers and Orchestration Systems. The proliferation of LLMs has spurred the development of model routing systems. Many works formulate this as a single-turn problem, employing static policies or offline-trained classifiers (Chen, Zaharia, and Zou 2023; Jiang, Ren, and Lin 2023; Ong et al. 2024; Zhao, Jin, and Mao 2024; Liu et al. 2024b; Lu et al. 2023; Jitkrittum et al. 2025; Parkar et al. 2024; Ding et al. 2024a). Even frameworks designed for multi-turn dialogues, such as AutoMix (Aggarwal et al. 2024), Graphrouter (Feng, Shen, and You 2024), and MixLLM (Wang et al. 2025b), typically treat each turn as an independent, stateless subproblem. Our work departs from this paradigm by framing routing as a stateful, online sequential decision-making task. We introduce a contextual bandit framework that continuously learns throughout interaction, making it the first approach specifically designed to handle unstructured context evolution.

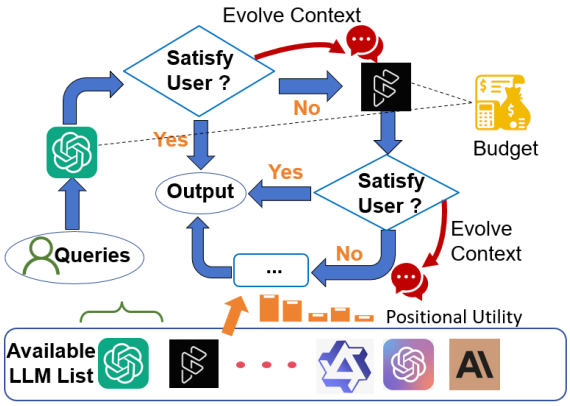


Figure 1: Online multi-LLM selection with contextual bandits under unstructured context evolution.

Sequential Decision-Making with LLMs. Sequential decision-making with LLMs is widely explored in agentic frameworks for multi-step reasoning (Wu et al. 2024; Zhao et al. 2025; Qiu et al. 2025; Feng et al. 2025). Paradigms like ReAct (Yao et al. 2023b) and Tree of Thoughts (Yao et al. 2023a) use hand-crafted heuristics in structured pipelines, while other research integrates LLMs into reinforcement learning (RL) loops (Wang et al. 2024a; Ding et al. 2024b; Li et al. 2023). These RL methods, however, require a learnable or simulatable environment model to function. Our work addresses a fundamentally different setting where the context transition, i.e., how a prompt evolves after a model's response, is an unlearnable black-box process. This intractability of the environment dynamics makes traditional RL and planning methods unsuitable, necessitating our myopic bandit approach that avoids forecasting future states.

Contextual Bandits for LLM Selection. Our work is built upon the theory of contextual bandits (Agrawal and Devanur 2016; Li et al. 2010), a classic approach for online learning in dynamic environments like recommendation (Luo et al. 2018; Zeng et al. 2025; Wang et al. 2025c) or pricing (Xiong et al. 2019). While prior works have applied bandits to LLM routing (Nguyen et al. 2024; Xia et al. 2024; Wang et al. 2025b; Dai et al. 2024; Shi et al. 2024), we are the first to formulate and solve the problem of sequential selection under unstructured context evolution. We introduce the concept of myopic regret to make the problem theoretically tractable without modeling the full trajectory. This allows our LinUCB-based algorithm to achieve provably sublinear regret, with novel extensions that handle costs and positional utility.

3 Problem Formulation

We consider an online sequential decision-making problem for adaptive multi-LLM selection, motivated by real-time user interactions with LLMs. Let $[K] = \{1, \ldots, K\}$ denote the set of available LLMs. Each LLM $k \in [K]$ is associated with an unknown feature vector $\theta_k^* \in \mathbb{R}^d$, capturing its alignment with user preferences across different prompt contexts. A learning agent interacts with users over T rounds. In each round $t \in [T]$, the agent receives a user query Q_t and engages in a multi-step adaptive interaction with the user, lasting at most H steps. Let $x_{t,1} \in \mathbb{R}^d$ denote the initial context vector derived from Q_t (e.g., an embedding or prompt representation). At each step $h \in [H]$, the agent observes context $x_{t,h}$ and selects an LLM $a_{t,h} \in [K]$ to generate a response. Let $R_{t,h}$ denote the output of LLM $a_{t,h}$ when invoked on context $x_{t,h}$. The user provides binary feedback $r_{t,h} \in \{0, 1\}$, indicating satisfaction with the result. For analytical simplicity, we assume a linear model for the conditional expectation of the feedback: with the selected LLM $a_{t,h}$,

$$\mathbb{E}[r_{t,h} \mid x_{t,h}, a_{t,h}] = \langle x_{t,h}, \theta^*_{a_{t,h}} \rangle.$$

This surrogate model enables the application of linear contextual bandit algorithms. If $r_{t,h} = 1$, the round ends. Otherwise, the agent proceeds to the next step, where the context is updated by incorporating the previous LLM response:

$$x_{t,h+1} = g(x_{t,h}, a_{t,h}, R_{t,h}, r_{t,h}),$$

where g is a black-box function representing how the context evolves in response to the LLM's output (e.g., through concatenation, rewriting, or augmentation). Importantly, as discussed in Section 1, g is assumed to be unstructured and unlearnable: it may be stochastic, nonstationary, or governed by proprietary model behavior, and cannot be explicitly modeled or simulated. This interaction continues for up to H steps or until the user is satisfied. The goal is to design an online decision policy that maximizes the expected cumulative user satisfaction over T rounds.

Myopic Regret. Due to the unstructured and unobservable nature of the context evolution function g, multi-step planning or reinforcement learning methods that rely on modeling or simulating future states are inapplicable. Instead, we adopt a myopic regret minimization framework, where the agent selects each LLM based solely on the current context, without modeling future transitions. At step h of round t, define the instantaneous regret as:

$$\text{Reg}_{t,h} = \max_{k \in [K]} \langle x_{t,h}, \theta_k^* \rangle - \langle x_{t,h}, \theta^*_{a_{t,h}} \rangle,$$

which quantifies the expected shortfall in user satisfaction due to a suboptimal LLM choice in the given context. The total regret over T rounds is:

$$\mathcal{R}(T) = \textstyle\sum_{t=1}^{T} \sum_{h=1}^{H_t} \text{Reg}_{t,h},$$

where $H_t \leq H$ denotes the number of steps taken in round t.

Remark. While it may seem natural to compare against the best full sequence of LLMs per query (i.e., sequence-wise regret), such analysis requires modeling or learning the transition function g, which is infeasible in practice. Moreover, the space of possible LLM trajectories is combinatorially large and highly nonstationary. Thus, sequence-wise regret is neither practically meaningful nor theoretically tractable in our setting. By contrast, myopic regret minimization ensures robust performance at each decision point, without relying on assumptions about future context evolution, making it more suitable for real-time multi-LLM decision-making under unstructured prompt dynamics

Algorithm 1: Greedy LinUCB for Multi-LLM Selection

1: **Input:** Regularization and confidence parameters λ, α
2: Initialize $A_k \leftarrow \lambda I_d$, $b_k \leftarrow \mathbf{0} \in \mathbb{R}^d$ for all $k \in [K]$
3: **for** round $t = 1, 2, \cdots$ **do**
4: Receive initial query Q_t and build context $x_{t,1} \in \mathbb{R}^d$
5: **for** step $h = 1$ to H **do**
6: **for** each $k \in [K]$ **do**
7: $\hat{\theta}_k \leftarrow A_k^{-1} b_k$;
8: $\text{UCB}_k \leftarrow \langle x_{t,h}, \hat{\theta}_k \rangle + \alpha \cdot \sqrt{x_{t,h}^\top A_k^{-1} x_{t,h}}$
9: **end for**
10: Query LLM $a_{t,h} \leftarrow \arg\max_k \text{UCB}_k$ with context $(Q_t, R_{t,1}, \cdots, R_{t,h-1})$
11: Receive output $R_{t,h}$ and binary feedback $r_{t,h}$
12: $A_{a_{t,h}} \leftarrow A_{a_{t,h}} + x_{t,h} x_{t,h}^\top$;
13: $b_{a_{t,h}} \leftarrow b_{a_{t,h}} + r_{t,h} x_{t,h}$
14: **if** $r_{t,h} = 1$ **then**
15: **break**
16: **else**
17: $x_{t,h+1} \leftarrow g(x_{t,h}, a_{t,h}, R_{t,h}, r_{t,h})$
18: **end if**
19: **end for**
20: **end for**

4 LinUCB-based Greedy Algorithm

In this section, we propose a LinUCB-based greedy algorithm for sequential LLM selection. As discussed in Section 3, since future contexts evolve via an unknown and unstructured function g, multi-step planning is intractable. We therefore adopt a myopic strategy that focuses on optimizing per-step feedback using contextual bandits. Specifically, we adapt the LinUCB algorithm to this setting by maintaining a separate linear model for each LLM and selecting the arm with the highest upper confidence bound (UCB) at each step.

As described in Algorithm 1, the greedy LinUCB algorithm maintains a regularized least-squares model for each LLM $k \in [K]$, represented by a matrix A_k and a response vector b_k. At each decision step (t, h), the algorithm computes the ridge regression estimate $\hat{\theta}_k = A_k^{-1} b_k$ and evaluates the LinUCB index for each LLM (line 7). This index combines the predicted reward $\hat{\theta} k^\top x t, h$ with an exploration bonus proportional to the estimate's uncertainty, promoting the selection of under-explored LLMs. The LLM with the highest LinUCB score is chosen for querying. This approach is crucial, as selecting solely based on the highest predicted reward (pure exploitation) risks suboptimal decisions by neglecting parameter uncertainty. After receiving the model response $R_{t,h}$ and observing the binary user feed-

back $r_{t,h} \in \{0, 1\}$, the algorithm updates the corresponding model parameters $A_{a_{t,h}}$ and $b_{a_{t,h}}$. If the user is satisfied (i.e., $r_{t,h} = 1$), the round terminates. Otherwise, the context evolves to $x_{t,h+1} = g(x_{t,h}, a_{t,h}, R_{t,h}, r_{t,h})$, and the algorithm proceeds to the next step. This greedy procedure is repeated at each decision point without modeling the future evolution of context. Under the following assumptions, we provide a high-probability upper bound on the cumulative myopic regret of Algorithm 1.

Assumption 1 (Bounded Parameters and Context Norms)**.** *For all $k \in [K]$, $\|\theta_k^*\|_2 \leq S$. For all $x_{t,h}$, $\|x_{t,h}\|_2 \leq L$.*

Theorem 1. *With probability at least $1 - \delta$, the cumulative myopic regret of the Greedy LinUCB algorithm satisfies:*

$$\mathcal{R}(T) = O\left(d\sqrt{KTH} \cdot \left(SL + \sqrt{\lambda}S\right) \cdot \log\left(\frac{KTL^2}{\lambda\delta}\right)\right).$$

The regret bound in Theorem 1 matches the structure of standard LinUCB bounds in contextual bandits, up to an additional factor of $\sqrt{H}$ reflecting the multi-step interaction in each round. The regret scales as $\widetilde{O}(dSL\sqrt{KTH})$, where the $\widetilde{O}$ notation hides logarithmic terms. Compared to classical contextual bandits (Abbasi-Yadkori, Pál, and Szepesvári 2011), the challenge in our setting stems from the evolving context dynamics. Nevertheless, by minimizing per-step regret and avoiding assumptions on the transition function g, the greedy LinUCB algorithm remains provably efficient and robust in this unstructured sequential environment.

5 Budget-Aware Multi-LLM Selection

In real-time LLM selection settings, different models incur different costs due to latency, compute, or token-based pricing. Moreover, these costs can vary stochastically with user queries. Motivated by this, we extend our algorithm to handle per-query *random costs*, where each user specifies a budget and the learner must select LLMs adaptively within that budget while maximizing user satisfaction.

Problem Setup. In each round $t \in [T]$, the user provides a total budget $B_t > 0$. At each step $h \in [H]$, when the learner selects arm $a_{t,h}$, it observes a reward $r_{t,h} \in \{0, 1\}$ and incurs a cost $c_{t,h,a_{t,h}} \in [0, C_{\max}]$ drawn i.i.d. from a fixed but unknown distribution with mean $\mu_{a_{t,h}}$. The learner observes the cost only after querying the LLM. The cumulative cost must satisfy $\sum_{h=1}^{H_t} c_{t,h,a_{t,h}} \leq B_t$, where H_t is the number of steps taken in round t. The goal is to maximize cumulative user satisfaction under this per-round budget constraint.

5.1 Budget-Aware Greedy LinUCB Algorithm.

At each step, the algorithm computes:

- *Reward estimate:* $\text{UCB}_k(x_{t,h}) = \langle x_{t,h}, \hat{\theta}_k \rangle + \alpha_k$, where $\hat{\theta}_k$ is the estimator and α_k is the confidence width.
- *Cost estimate:* $\hat{c}_k$ is the empirical mean of observed costs for arm k, with confidence interval width $\beta_k = \sqrt{\log(2TK/\delta)/(2N_k)}$, where N_k is the number of times arm k has been selected.
- *Score:* The algorithm computes

$$\text{Score}_k(x_{t,h}) = \frac{\text{UCB}_k(x_{t,h})}{\max\{\hat{c}_k - \beta_k, \varepsilon\}},$$

where ε is a small positive constant to avoid a denominator of 0, and selects the arm with the highest score whose upper-bound cost estimate $\hat{c}_k + \beta_k$ fits within the remaining budget.

This two-level confidence mechanism ensures both optimism in reward and conservatism in cost, balancing exploration and budget feasibility. We now provide a regret bound comparing against a per-step myopic oracle that knows true reward and cost expectations.

Myopic Regret Definition. In the presence of budget constraints and stochastic costs, it is infeasible to compare against a full-horizon optimal policy or to use standard cumulative reward as the regret benchmark. Instead, we adopt a *per-step myopic regret* formulation. At each step (t, h), define the best feasible arm (oracle) as:

$$k_{t,h}^* = \arg \max_{k \in [K],\ \mu_k \leq b_{t,h}} \frac{\langle x_{t,h}, \theta_k^* \rangle}{\mu_k},$$

where $b_{t,h}$ is the remaining budget before step h in round t. This oracle greedily selects the arm with the highest expected reward per unit cost that fits within the remaining budget. Let the algorithm's selected arm be $a_{t,h}$. Then the total regret is:

$$\mathcal{R}_{\text{budget}}(T) = \sum_{t=1}^{T} \sum_{h=1}^{H_t} \left(\langle x_{t,h}, \theta_{k_{t,h}^*}^* \rangle - \langle x_{t,h}, \theta_{a_{t,h}}^* \rangle\right).$$

Assumption 2 (Stochastic Cost)**.** *For each arm $k \in [K]$, the cost $c_{t,h,k}$ is sub-Gaussian with mean $\mu_k \in (0, C_{\max}]$.*

$$\mathcal{R}_{\text{budget}}(T) = \widetilde{O}\left(dSL \cdot \sqrt{KTH} + \textstyle\sum_{k=1}^{K} \frac{C_{\max}}{\mu_k^2} \cdot \sqrt{T \log\left(\frac{TK}{\delta}\right)}\right). \tag{1}$$

This result extends classical LinUCB analysis to the setting with budget constraints and stochastic, unknown costs. The reward regret scales with $d\sqrt{KTH}$, while the additional cost regret grows with $\sqrt{T}$ and is amplified by small mean costs μ_k. Despite this challenge, the algorithm remains budget-feasible and achieves sublinear cumulative regret.

5.2 Positionally-Aware Knapsack Heuristic

The greedy budget-aware algorithm introduced earlier selects LLMs sequentially by maximizing a cost-normalized reward score. While effective in balancing reward and cost, this approach may overly prioritize LLMs and delay the use of more capable ones until later in the interaction. In many real-world scenarios, however, users exhibit positional bias: they prefer to receive high-quality responses early and may disengage if early attempts fail. This motivates the need to consider *positional utility*, where earlier successful responses are more valuable in practice than later ones.

To better balance early accuracy and overall cost, we propose a PAKH in Algorithm 2. The key idea is to incorporate cost estimates and UCB reward scores into a dynamic

Algorithm 2: Positionally-Aware Knapsack Heuristic

Require: Budget B_t, UCB scores $\widehat{r}_k$, cost estimates $\hat{c}_k$ for all $k \in [K]$

Initialize remaining budget $b \leftarrow B_t$ and candidate list $\mathcal{C} \leftarrow \emptyset$
while $b > 0$ **do**
  Solve a 0-1 knapsack problem on arms $[K] \setminus \mathcal{C}$ with budget b and objective $(\widehat{r}_k, \hat{c}_k)$
  Let $k^{\text{next}} \leftarrow$ arm with highest $\widehat{r}_k$ among selected items in knapsack solution
  if k^{next} does not exist or $\hat{c}_{k^{\text{next}}} > b$ **then**
    break
  end if
  Add k^{next} to candidate list $\mathcal{C}$
  $b \leftarrow b - \hat{c}_{k^{\text{next}}}$
end while
return candidate list $\mathcal{C}$

selection procedure that prioritizes early deployment of high-confidence LLMs while respecting the per-query budget. At each step, the algorithm solves a 0-1 knapsack problem over the remaining LLM candidates, using UCB scores as reward estimates and empirical costs as weights. From the knapsack solution, it selects the arm with the highest UCB score, adds it to the candidate list, and updates the remaining budget. This process is repeated until no feasible arms remain or the budget is exhausted. This budget-aware strategy prioritizes strong LLMs early without exceeding cost limits.

Remark. This knapsack-based strategy explicitly targets positional utility by favoring the early use of high-performing models, rather than deferring them due to their cost. Unlike purely greedy cost-normalized approaches, which may over-prioritize cheap but weak models, the knapsack heuristic balances exploration and resource allocation more effectively. While the method is heuristic and lacks a formal regret bound, it aligns closely with practical deployment goals in interactive systems. As demonstrated in our experiments, the algorithm delivers higher effective utility—especially when stronger models are costly and positional preferences matter.

6 Experiments

6.1 Experimental Setup

We evaluate our proposed methods against several baselines, including two single-step contextual bandit routers, MetaLLM (Nguyen et al. 2024) and MixLLM (Wang et al. 2025b), and Majority Voting (Li et al. 2024). Our candidate LLM pool comprises six models representing a wide spectrum of capabilities and costs, including *Llama-4-Maverick*, *Gemini-2.0-Flash*, and *Mistral-Small-3.1*. Experiments are conducted on four benchmarks covering distinct domains: MMLU-Pro (Wang et al. 2024b) and GPQA (Rein et al. 2024) for general knowledge, and AIME and Math500 (Lightman et al. 2023) for mathematical reasoning.

For evaluation, we use accuracy as the primary metric. On tasks without discrete labels (AIME, GPQA, Math500), correctness is determined by a *Deepseek-R1* (Liu et al. 2024a) grader model providing binary feedback. Each dataset is randomly split into 20% for initialization and 80% for online evaluation.

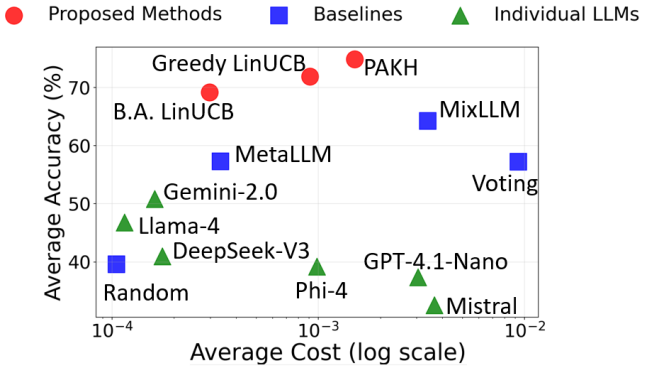


Figure 2: Avg. performance of various methods.

To ensure a fair comparison, all methods were evaluated under the same multi-step protocol, allowing up to $H = 4$ attempts per query. While our method adaptively selects a new LLM at each step, the baseline routers retry their initial selection upon failure, per their single-step design. We use 384-dimensional BGE embeddings (Xiao et al. 2023) to represent queries. Key hyperparameters for the bandit algorithm are set to $\alpha = 0.675$ and $\lambda = 0.45$ based on preliminary validation. For our budget-aware (B.A.) methods, the per-query budget is defined relative to the average cost incurred by the unconstrained Greedy LinUCB algorithm, ensuring a realistic and challenging constraint. The detailed experimental setup is deferred to the extended version.

6.2 Main Results

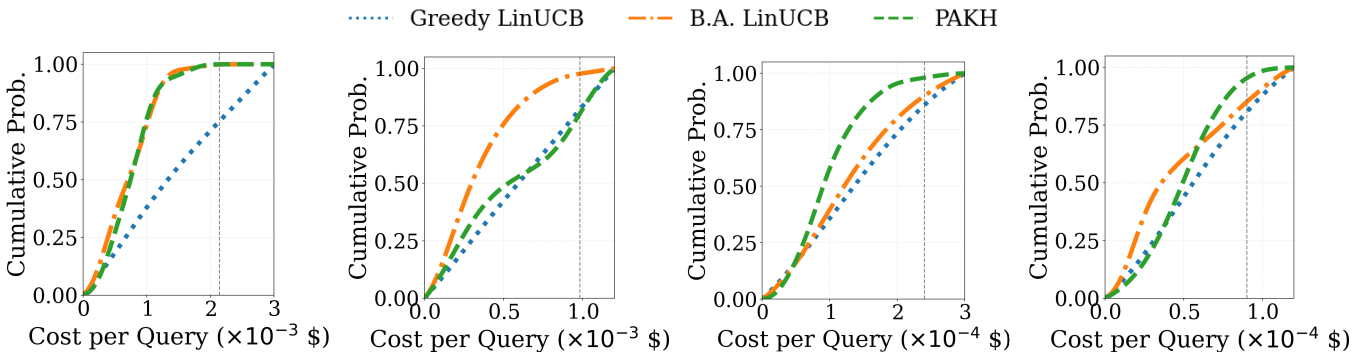
As shown in Fig. 2, our proposed online selection methods consistently demonstrate a more effective trade-off between accuracy and cost compared to all baseline approaches. As seen in Section 6.1 our methods occupy the desirable top-left region of the plot, indicating a capacity to achieve higher accuracy for a given operational cost. The detailed results in Section 6.1 corroborate this trend across the four diverse datasets. The performance of individual LLMs varies significantly by task: for instance, *Gemini-2.0-Flash* is most accurate on MMLU-Pro, while *Llama-4-Maverick* excels on AIME. This underscores the necessity for an adaptive routing mechanism rather than relying on a single model.

The PAKH, in particular, emerges as the most effective method overall. Analyzing its performance in Section 6.1, PAKH achieves the highest accuracy on the challenging reasoning datasets, AIME (62.50%) and GPQA (66.67%). When averaged across all tasks, it outperforms the strongest routing baseline, MixLLM, by a significant margin in accuracy while simultaneously being more cost-effective. This demonstrates its ability to strategically allocate resources to more powerful models when necessary to maximize performance.

Our other proposed methods also show distinct advantages. The Greedy LinUCB algorithm, which optimizes for accuracy without budget constraints, achieves the highest performance on MMLU-Pro (86.25%) and Math500 (96.25%), confirming

	Method	MMLU-Pro Acc.	MMLU-Pro Cost	AIME Acc.	AIME Cost	GPQA Acc.	GPQA Cost	Math500 Acc.	Math500 Cost	Avg. Acc.	Avg. Cost
Candidate LLMs	Mistral-Small-3.1	48.80	**2.0E-05**	1.67	3.72E-03	22.22	5.05E-03	57.60	1.54E-04	32.57	2.24E-03
	Phi-4	51.50	**2.0E-05**	8.33	3.82E-03	29.80	5.05E-03	67.20	1.54E-04	39.21	2.26E-03
	Llama-4-Maverick	41.77	8.3E-05	**20.00**	**1.4E-04**	**39.90**	**1.0E-04**	85.40	**1.1E-04**	46.77	**1.08E-04**
	Gemini-2.0-Flash	**62.10**	2.8E-05	**20.00**	3.01E-04	35.30	4.10E-04	**86.00**	1.61E-04	**50.85**	2.25E-04
	GPT-4.1-Nano	41.33	2.7E-05	6.67	1.19E-02	29.80	2.07E-01	71.60	1.95E-04	37.35	5.48E-02
	DeepSeek-V3	58.80	1.16E-04	3.33	2.37E-04	31.31	2.85E-04	70.40	1.85E-04	40.96	2.06E-04
	Voting	63.30	3.38E-04	67.39	2.31E-02	47.48	8.10E-02	50.85	9.34E-03	57.26	2.84E-02
Routing	MixLLM	76.32	1.27E-04	41.67	3.61E-03	53.79	4.90E-03	85.20	4.37E-04	64.25	2.27E-03
	MetaLLM	72.11	6.73E-05	20.83	**4.26E-04**	52.40	7.81E-04	83.80	5.05E-04	57.29	4.45E-04
	Random	49.75	4.73E-05	13.64	1.56E-04	27.85	9.81E-05	67.00	1.07E-04	39.56	1.02E-04
	Greedy LinUCB	**86.25**	9.28E-05	54.16	2.14E-03	51.28	3.40E-03	**96.25**	3.98E-04	71.99	1.51E-03
	B.A. LinUCB	77.29	**4.89E-05**	56.25	7.00E-04	59.12	**9.72E-05**	84.00	**2.05E-04**	69.17	**2.63E-04**
z	PAKH	82.20	5.12E-05	**62.50**	7.52E-04	**66.67**	2.10E-04	88.00	2.97E-04	**74.84**	3.28E-04

Table 1: Performance comparison. We compare accuracy (Acc. %) and cost (in US Dollars). The best result is in **bold** and the worst is underlined. The *Random* baseline is excluded from highlighting.



(a) AIME (b) GPQA (c) Math500 (d) MMLU-Pro

Figure 3: Cumulative Distribution Function (CDF) of per-query costs across different benchmarks.The gray vertical line represents the fixed per-query budget. Our methods consistently achieve higher satisfaction at lower operational costs.

the effectiveness of the underlying contextual bandit formulation. When cost is a primary concern, the B.A. LinUCB variant offers a compelling profile. It obtains a higher average accuracy than both MetaLLM and MixLLM, yet operates at a substantially lower average cost than both methods. This result highlights that even without the more complex positional heuristic, our sequential bandit framework provides significant value over single-step routing baselines.

In contrast, the performance of the baseline routers, MetaLLM and MixLLM, is comparatively limited. We attribute this to their underlying single-step decision-making paradigm. Both MetaLLM and MixLLM employ contextual bandit frameworks to make an optimal routing choice based on the initial query's features. While our experimental setup grants these baselines multiple attempts per query for a fair comparison of computational budget, their architecture is not designed to re-evaluate this initial choice. Consequently, a failed attempt results in retrying with the same model on the same unmodified prompt. Our framework, however, is fundamentally designed for sequential interactions. Upon failure at a given step, it not only incorporates the prior response into the context but can also dynamically select a *different* and potentially more suitable LLM for the updated prompt. This ability to adapt its strategy within a single multi-turn interaction allows it to recover from initial suboptimal choices and ultimately achieve higher success rates, representing a key advantage of our proposed approach.

Effectiveness of Positional Awareness To align with the user preference for early, correct answers, our PAKH framework prioritizes high-confidence models. The results in Table 2 confirm the effectiveness of this design. It shows on average, the PAKH method achieves a first-step accuracy of 71.10%. This single-step performance not only surpasses the total average accuracy of the B.A. LinUCB method (69.17%) but also accounts for over 95% of PAKH's own final average accuracy (74.84%). This indicates that for the vast majority of queries, the knapsack heuristic successfully identifies a satisfactory LLM on its first attempt. In stark contrast, both the Greedy and B.A. LinUCB methods rely more heavily on subsequent steps to accumulate their accuracy. For example, Greedy LinUCB achieves only 47.89% accuracy at the first

	Overall		Step Breakdown (%)			
Dataset	Acc. %	Avg. Steps	1st	2nd	3rd	4th
Greedy LinUCB						
MMLU-Pro	86.25	1.87	58.58	15.04	7.12	5.05
AIME	54.16	3.38	18.75	20.83	10.42	4.17
GPQA	51.28	3.31	28.21	15.38	2.56	5.13
Math500	96.25	1.31	86.00	6.25	3.00	1.00
Avg.	71.99	2.52	47.89	15.38	6.03	4.09
Budget-Aware LinUCB						
MMLU-Pro	77.29	2.12	48.21	14.62	9.12	5.33
AIME	56.25	2.04	47.92	4.17	2.08	2.08
GPQA	59.12	2.90	32.70	11.95	8.18	6.29
Math500	84.00	1.08	80.75	3.00	0.25	0.00
Avg.	69.17	2.04	52.39	8.44	4.90	3.49
Positionally-Aware Knapsack						
MMLU-Pro	82.20	2.00	78.11	4.05	0.04	0.00
AIME	62.50	1.54	60.42	2.08	0.00	0.00
GPQA	66.67	1.78	57.86	7.55	1.26	0.00
Math500	88.00	1.01	88.00	0.00	0.00	0.00
Avg.	74.84	1.58	71.10	3.42	0.33	0.00

Table 2: Accuracy and avg. step at each sequential attempt.

step, requiring multiple further attempts to reach its final performance. This front-loading of accuracy also results in superior efficiency, with PAKH requiring the fewest average steps to satisfy a query (1.58). This behavior is attributed to the knapsack formulation, which strategically plans the use of the entire query budget to deploy high-confidence models early, rather than using a myopic, purely cost-normalized selection criteria at each step.

Effectiveness of Budget Awareness Comparing the Greedy LinUCB and the B.A. LinUCB methods isolate the impact of adding an explicit budget constraint. While Greedy LinUCB achieves a slightly higher average accuracy (71.99% vs. 69.17%), it does so at the cost of significantly lower efficiency, requiring an average of 2.52 steps per query. By incorporating a cost-aware selection mechanism, B.A. LinUCB reduces the average number of steps by nearly 20% to 2.04. This shows that enforcing budget constraints encourages selecting a more cost-effective model, which helps avoid long and expensive cascades even if it means forgoing a small amount of potential accuracy on the most difficult queries. As shown in Fig. 3 . The cost distribution for Greedy LinUCB exhibits a long tail, indicating unpredictable and occasionally high-cost queries. Conversely, the B.A. LinUCB algorithm's cost curve saturates near its budget threshold, confirming that it reliably operates within its financial constraints. This shows that even the simpler budget-aware mechanism provides critical control over operational costs, a crucial feature for practical deployment.

Sensitivity Analysis on Budget Constraint We conducted a sensitivity analysis by evaluating their performance across a range of fixed per-query budgets. As seen in Fig. 4, the performance of both the B.A. LinUCB and the PAKH methods

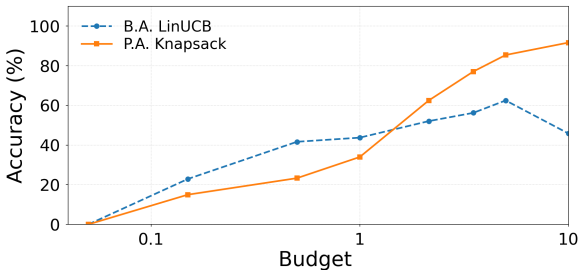


Figure 4: Accuracy under varying budget constraints.

are highly dependent on the available budget. At extremely tight budget constraints (e.g., below $0.001), B.A. LinUCB initially outperforms the PAKH heuristic. This is because its greedy, cost-normalized selection strategy is effective at identifying the most efficient low-cost models when the budget allows for little else. However, as the budget increases, the PAKH method demonstrates superior scalability. Its performance consistently rises, surpassing B.A. LinUCB continues to improve as it effectively leverages the larger budget to deploy more powerful models. Interestingly, the performance of B.A. LinUCB declines after reaching a peak around a budget of $0.0035. This suggests that its myopic selection strategy, which normalizes reward by cost, becomes less effective when budget constraints are relaxed, as it may favor models with a good cost-to-reward ratio over models that are genuinely best for the task. In contrast, the knapsack formulation's more holistic planning allows it to robustly allocate resources to achieve higher accuracy, establishing it as the more scalable and reliable approach across a wider spectrum of operational budgets.

7 Concluding Remarks

In this work, we proposed a novel contextual bandit framework for online multi-LLM selection that effectively handles unstructured context evolution. Our approach, which formulates the problem with a tractable myopic regret, enables provably efficient learning without needing to model complex user interaction dynamics. Experiments show our methods outperform existing routers in accuracy and cost-efficiency by learning to adaptively select LLMs. This research provides a lightweight and theoretically-grounded solution for building more effective and adaptive multi-LLM systems.

Our work also opens several promising directions for future research. While myopic decision-making is well-suited for unstructured and non-predictable environments, it would be interesting to explore hybrid approaches that incorporate limited non-myopic planning when partial structure or reliable abstractions of context evolution are available. In addition, our current framework assumes immediate and reliable user feedback; extending the model to handle delayed, sparse, or noisy feedback remains an important challenge. Finally, adapting the proposed methods to dynamically evolving LLM pools where models may be added, removed, or updated over time represents a critical step toward long-term robustness and practical deployment in multi-LLM systems.

Acknowledgements

The work of Jinhang Zuo was supported by CityUHK 9610706. The work of Xiangxiang Dai was supported by the National Natural Science Foundation of China (625B2163). The work of Fang Kong was supported by the Guangdong Basic and Applied Basic Research Foundation 2025A1515011412. The work of John C.S. Lui was supported in part by the RGC GRF-14215722.

References

Abbasi-Yadkori, Y.; Pál, D.; and Szepesvári, C. 2011. Improved algorithms for linear stochastic bandits. *Advances in neural information processing systems*, 24.

Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F. L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Aggarwal, P.; Madaan, A.; Anand, A.; Potharaju, S. P.; Mishra, S.; Zhou, P.; Gupta, A.; Rajagopal, D.; Kappaganthu, K.; Yang, Y.; et al. 2024. Automix: Automatically mixing language models. *Advances in Neural Information Processing Systems*, 37: 131000–131034.

Agrawal, S.; and Devanur, N. 2016. Linear contextual bandits with knapsacks. *Advances in neural information processing systems*, 29.

Anil, R.; Dai, A. M.; Firat, O.; et al. 2023. PaLM 2 Technical Report. arXiv:2305.10403.

Bubeck, S.; Chadrasekaran, V.; Eldan, R.; Gehrke, J.; Horvitz, E.; Kamar, E.; Lee, P.; Lee, Y. T.; Li, Y.; Lundberg, S.; et al. 2023. Sparks of artificial general intelligence: Early experiments with GPT-4. *arXiv preprint arXiv:2303.12712*.

Chen, L.; Davis, J. Q.; Hanin, B.; Bailis, P.; Zaharia, M.; Zou, J.; and Stoica, I. 2025. Optimizing Model Selection for Compound AI Systems. *arXiv preprint arXiv:2502.14815*.

Chen, L.; Zaharia, M.; and Zou, J. 2023. Frugalgpt: How to use large language models while reducing cost and improving performance. *arXiv preprint arXiv:2305.05176*.

Dai, X.; Li, J.; Liu, X.; Yu, A.; and Lui, J. C. 2024. Cost-Effective Online Multi-LLM Selection with Versatile Reward Models. *arXiv preprint arXiv:2405.16587*.

Dai, X.; Xie, Y.; Liu, M.; Wang, X.; Li, Z.; Wang, H.; and Lui, J. 2025. Multi-Agent Conversational Online Learning for Adaptive LLM Response Identification. *arXiv preprint arXiv:2501.01849*.

Ding, D.; Mallick, A.; Wang, C.; Sim, R.; Mukherjee, S.; Ruhle, V.; Lakshmanan, L. V.; and Awadallah, A. H. 2024a. Hybrid llm: Cost-efficient and quality-aware query routing. *arXiv preprint arXiv:2404.14618*.

Ding, Z.; Ke, R.; Huang, W.; Jiang, G.; Li, Y.; Yang, D.; and Liang, J. 2024b. Adaptive reinforcement learning planning: Harnessing large language models for complex information extraction. *arXiv preprint arXiv:2406.11455*.

Feng, L.; Xue, Z.; Liu, T.; and An, B. 2025. Group-in-group policy optimization for llm agent training. *arXiv preprint arXiv:2505.10978*.

Feng, T.; Shen, Y.; and You, J. 2024. GraphRouter: A Graph-based Router for LLM Selections. In *The Thirteenth International Conference on Learning Representations*.

Han, Z.; Liu, X.; Zhou, R.; Dai, X.; and Lui, J. 2025. Faster, Smaller, and Smarter: Task-Aware Expert Merging for Online MoE Inference. *arXiv preprint arXiv:2509.19781*.

Jiang, D.; Ren, X.; and Lin, B. Y. 2023. LLM-Blender: Ensembling Large Language Models with Pairwise Ranking and Generative Fusion. In *The 61st Annual Meeting Of The Association For Computational Linguistics*.

Jitkrittum, W.; Narasimhan, H.; Rawat, A. S.; Juneja, J.; Wang, Z.; Lee, C.-Y.; Shenoy, P.; Panigrahy, R.; Menon, A. K.; and Kumar, S. 2025. Universal model routing for efficient llm inference. *arXiv preprint arXiv:2502.08773*.

Li, H.; Chong, Y. Q.; Stepputtis, S.; Campbell, J.; Hughes, D.; Lewis, M.; and Sycara, K. 2023. Theory of mind for multi-agent collaboration via large language models. *arXiv preprint arXiv:2310.10701*.

Li, J.; Zhang, Q.; Yu, Y.; Fu, Q.; and Ye, D. 2024. More agents is all you need. *arXiv preprint arXiv:2402.05120*.

Li, L.; Chu, W.; Langford, J.; and Schapire, R. E. 2010. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, 661–670.

Lightman, H.; Kosaraju, V.; Burda, Y.; Edwards, H.; Baker, B.; Lee, T.; Leike, J.; Schulman, J.; Sutskever, I.; and Cobbe, K. 2023. Let's Verify Step by Step. *arXiv preprint arXiv:2305.20050*.

Liu, A.; Feng, B.; Xue, B.; Wang, B.; Wu, B.; Lu, C.; Zhao, C.; Deng, C.; Zhang, C.; Ruan, C.; et al. 2024a. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.

Liu, Y.; Zhang, H.; Miao, Y.; Le, V.-H.; and Li, Z. 2024b. Optllm: Optimal assignment of queries to large language models. In *2024 IEEE International Conference on Web Services (ICWS)*, 788–798. IEEE.

Lu, K.; Yuan, H.; Lin, R.; Lin, J.; Yuan, Z.; Zhou, C.; and Zhou, J. 2023. Routing to the expert: Efficient reward-guided ensemble of large language models. *arXiv preprint arXiv:2311.08692*.

Luo, H.; Wei, C.-Y.; Agarwal, A.; and Langford, J. 2018. Efficient contextual bandits in non-stationary worlds. In *Conference On Learning Theory*, 1739–1776. PMLR.

Murali, K. C.; Harada, N.; Sai; Soundararaj, P.; and Kwatra, S. 2025. OpenAI Cookbook: Practical Guide for Model Selection for Real-World Use Cases. Accessed January 15, 2026. https://cookbook.openai.com/examples/partners/model_selection_guide/model_selection_guide.

Nguyen, Q. H.; Hoang, D. C.; Decugis, J.; Manchanda, S.; Chawla, N. V.; and Doan, K. D. 2024. MetaLLM: A High-performant and Cost-efficient Dynamic Framework for Wrapping LLMs. *arXiv preprint arXiv:2407.10834*.

Ong, I.; Almahairi, A.; Wu, V.; Chiang, W.-L.; Wu, T.; Gonzalez, J. E.; Kadous, M. W.; and Stoica, I. 2024. Routellm: Learning to route llms with preference data. *arXiv preprint arXiv:2406.18665*.

Parkar, R. S.; Kim, J.; Park, J. I.; and Kang, D. 2024. SelectLLM: Can LLMs Select Important Instructions to Annotate? *arXiv preprint arXiv:2401.16553*.

Qiu, J.; Qi, X.; Zhang, T.; Juan, X.; Guo, J.; Lu, Y.; Wang, Y.; Yao, Z.; Ren, Q.; Jiang, X.; et al. 2025. Alita: Generalist agent enabling scalable agentic reasoning with minimal predefinition and maximal self-evolution. *arXiv preprint arXiv:2505.20286*.

Rein, D.; Hou, B. L.; Stickland, A. C.; Petty, J.; Pang, R. Y.; Dirani, J.; Michael, J.; and Bowman, S. R. 2024. GPQA: A Graduate-Level Google-Proof Q&A Benchmark. In *First Conference on Language Modeling*.

Shi, C.; Yang, K.; Chen, Z.; Li, J.; Yang, J.; and Shen, C. 2024. Efficient prompt optimization through the lens of best arm identification. *Advances in Neural Information Processing Systems*, 37: 99646–99685.

Tang, Z.; Ma, Z.; Wang, S.; Li, Z.; Zhang, L.; Zhao, H.; Li, Y.; and Wang, Q. 2025. CoViPAL: Layer-wise Contextualized Visual Token Pruning for Large Vision-Language Models. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, 20701–20714.

Wang, H.; Hao, S.; Dong, H.; Zhang, S.; Bao, Y.; Yang, Z.; and Wu, Y. 2024a. Offline reinforcement learning for llm multi-step reasoning. *arXiv preprint arXiv:2412.16145*.

Wang, S.; Li, Z.; Luohe, S.; Du, B.; Zhao, H.; Li, Y.; and Wang, Q. 2025a. From Parameters to Performance: A Data-Driven Study on LLM Structure and Development. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, 26095–26112.

Wang, X.; Liu, Y.; Cheng, W.; Zhao, X.; Chen, Z.; Yu, W.; Fu, Y.; and Chen, H. 2025b. Mixllm: Dynamic routing in mixed large language models. *arXiv preprint arXiv:2502.18482*.

Wang, X.; Zeng, Q.; Zuo, J.; Liu, X.; Hajiesmaili, M.; Lui, J.; and Wierman, A. 2025c. Fusing Reward and Dueling Feedback in Stochastic Bandits. *arXiv preprint arXiv:2504.15812*.

Wang, Y.; Ma, X.; Zhang, G.; Ni, Y.; Chandra, A.; Guo, S.; Ren, W.; Arulraj, A.; He, X.; Jiang, Z.; et al. 2024b. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Wu, Q.; Bansal, G.; Zhang, J.; Wu, Y.; Li, B.; Zhu, E.; Jiang, L.; Zhang, X.; Zhang, S.; Liu, J.; et al. 2024. Autogen: Enabling next-gen LLM applications via multi-agent conversations. In *First Conference on Language Modeling*.

Xia, Y.; Kong, F.; Yu, T.; Guo, L.; Rossi, R. A.; Kim, S.; and Li, S. 2024. Which llm to play? convergence-aware online model selection with time-increasing bandits. In *Proceedings of the ACM Web Conference 2024*, 4059–4070.

Xiao, S.; Liu, Z.; Zhang, P.; and Muennighoff, N. 2023. C-Pack: Packaged Resources To Advance General Chinese Embedding. arXiv:2309.07597.

Xiong, Z.; Niyato, D.; Wang, P.; Han, Z.; and Zhang, Y. 2019. Dynamic pricing for revenue maximization in mobile social data market with network effects. *IEEE Transactions on Wireless Communications*, 19(3): 1722–1737.

Yao, S.; Yu, D.; Zhao, J.; Shafran, I.; Griffiths, T. L.; Cao, Y.; and Narasimhan, K. 2023a. Tree of Thoughts: Deliberate Problem Solving with Large Language Models. arXiv:2305.10601.

Yao, S.; Zhao, J.; Yu, D.; Du, N.; Shafran, I.; Narasimhan, K.; and Cao, Y. 2023b. ReAct: Synergizing Reasoning and Acting in Language Models. arXiv:2210.03629.

Yuan, W.; Pang, R. Y.; Cho, K.; Li, X.; Sukhbaatar, S.; Xu, J.; and Weston, J. 2025. Self-Rewarding Language Models. arXiv:2401.10020.

Zeng, Q.; He, E.; Hoffmann, R.; Wang, X.; and Zuo, J. 2025. Practical Adversarial Attacks on Stochastic Bandits via Fake Data Injection. *arXiv preprint arXiv:2505.21938*.

Zhang, X.; Cai, X.; Liu, B.; Huang, W.; Zhu, S.-C.; Qi, S.; and Yang, Y. 2025. Differentiable Information Enhanced Model-Based Reinforcement Learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 22605–22613.

Zhao, X.; Sedghi, H.; Bohnet, B.; Schuurmans, D.; and Nova, A. 2025. Improving Large Language Model Planning with Action Sequence Similarity. *arXiv preprint arXiv:2505.01009*.

Zhao, Z.; Jin, S.; and Mao, Z. M. 2024. Eagle: Efficient training-free router for multi-llm inference. *arXiv preprint arXiv:2409.15518*.

Zhu, Q.; Zhao, R.; Yan, H.; He, Y.; Chen, Y.; and Gui, L. 2025. Soft Reasoning: Navigating Solution Spaces in Large Language Models through Controlled Embedding Exploration. In *Forty-second International Conference on Machine Learning*.