# INFERENCEDYNAMICS: Efficient Routing Across LLMs through Structured Capability and Knowledge Profiling

Anonymous ACL submission

## Abstract

Large Language Model (LLM) routing is a pivotal technique for navigating a diverse landscape of LLMs, aiming to select the best-performing LLMs tailored to the domains of user queries, while managing computational resources. However, current routing approaches often face limitations in scalability when dealing with a large pool of specialized LLMs, or in their adaptability to extending model scope and evolving capability domains. To overcome those challenges, we propose **InferenceDynamics**, a flexible and scalable multi-dimensional routing framework by modeling the capability and knowledge of models. We operate it on our comprehensive dataset **RouteMix**, and demonstrate its effectiveness and generalizability in group-level routing using modern benchmarks including MMLU-Pro, GPQA, BigGenBench, and LiveBench, showcasing its ability to identify and leverage top-performing models for given tasks, leading to superior outcomes with efficient resource utilization. The broader adoption of Inference Dynamics can empower users to harness the full specialized potential of the LLM ecosystem, and our code will be made publicly available to encourage further research.

## 1 Introduction

The rapid proliferation of Large Language Models (LLMs) has unveiled a rich landscape of specialized capabilities, with different models demonstrating unique strengths across a multitude of domains and tasks (Matarazzo and Torlone, 2025; Li et al., 2024a). This specialization necessitates a sophisticated approach to model selection, where the primary goal is to identify and utilize the LLM best suited to the specific demands of a user's query. LLM routing (Chen et al., 2025) emerges as a critical paradigm to address this, creating mechanisms to strategically dispatch queries to the most capable model from a diverse pool, thereby maximizing performance, relevance, and the quality of outcomes, while also considering factors like inference cost and latency.

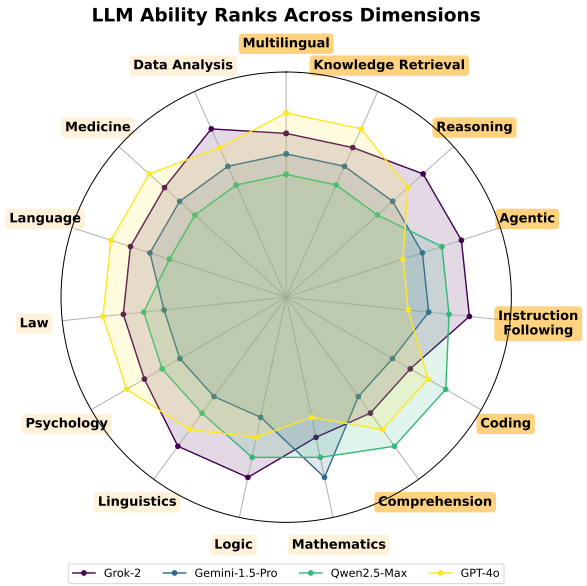


Figure 1: Quantification of Knowledge and Capability of top 4 models among candidate LLMs.

Early explorations in LLM routing often simplified the selection problem, for instance, by framing it as a binary classification task—e.g., choosing between a generalist small model and a powerful large model. Methods like AutoMix (Aggarwal et al., 2024), HybridLLM (Ding et al., 2024), and RouteLLM (Ong et al., 2025) demonstrated the viability of this approach, typically focusing on cost-performance trade-offs. While valuable for two-model scenarios, such binary frameworks face inherent scalability challenges, as selecting the best-performing model from many candidates using only pairwise comparisons becomes computationally costly and inefficient.

More recent works have advanced the field by leveraging richer model representations to better evaluate and route LLMs based on their specific capabilities. While methods including Rou-

terDC (Chen et al., 2024), C2MAB-V (Dai et al., 2024), and P2L (Frick et al., 2025) offer more sophisticated mechanisms for capturing model strengths, their primary limitation lies in the significant retraining or recalibration required to effectively support newly introduced LLMs, hindering their agility in a rapidly evolving model landscape. Model-SAT (Zhang et al., 2025) aimed to resolve this weakness through human-defined, model-independent capability decompositions. However, its reliance on predefined capability sets undermined adaptability to new capability dimensions, thereby limiting nuanced performance capture in specialized domains.

To address this gap, we introduce **InferenceDynamics**, a novel system designed for performant, scalable, and adaptable LLM routing. **InferenceDynamics** operates by extracting capability requirements and domain-specific knowledge from incoming queries, modeling the corresponding capabilities and knowledge profiles of available LLMs, and then intelligently routing queries to the most suitable models. To demonstrate the effectiveness and generalizability of our approach, we constructed a comprehensive dataset aggregated from 24 diverse benchmarks. We then evaluated our routing algorithm on four challenging out-of-distribution (OOD) benchmarks: MMLU-Pro (Wang et al., 2024b), GPQA (Rein et al., 2023), BigGenBench (Kim et al., 2024), and LiveBench (White et al., 2025). Experimental results show that our routing algorithm achieved the highest average score, surpassing the top-performing single LLM by a substantial margin of 1.28 points under optimal routing conditions. Furthermore, when operating under cost constraints, our algorithm delivered competitive performance comparable to the best single LLM, while utilizing nearly half the budget.

The contributions of our work are summarized as follows:

- We introduce **RouteMix**, a comprehensive dataset aggregated from 24 diverse benchmarks, specifically curated for rigorously evaluating the generalization capabilities of LLM routing algorithms.
- We propose **InferenceDynamics**, an efficient routing algorithm demonstrating generalization capabilities on previously unseen queries.
- Experimental results validate that **InferenceDynamics** significantly enhances LLM routing, substantially outperforming the leading single model while concurrently reducing computational overhead.

## 2 Related Works

### 2.1 Multi-LLM System

A Multi-LLM system (Chen et al., 2025) refers to the architecture that combines LLMs to collaboratively solve tasks more effectively than any single model. The rapid proliferation of diverse LLMs has spurred significant interest in such systems, which are realized through several architectural patterns. LLM ensembling (Jiang et al., 2023; Li et al., 2024b) enhances accuracy or robustness by processing the same input through several models and then aggregating their responses. Cascaded systems (Zhang et al., 2024; Kolawole et al., 2024; Chen et al., 2023) strategically employ a sequence of models—often initiating with smaller, faster LLMs for initial processing or simpler queries and escalating to more powerful, resource-intensive ones only when necessary—thereby optimizing resource use. Furthermore, the development of collaborative LLM agents (Wang et al., 2024a; Xu et al., 2024; Liang et al., 2024) involves multiple LLMs, with distinct roles or access to different tools, interacting to address complex, multi-step problems that demand sophisticated coordination. While these multi-LLM approaches demonstrate considerable advancements, they often necessitate querying multiple models, which can increase computational cost and latency. Moreover, as the number and diversity of available LLMs continue to grow, it becomes critical to route queries to the most suitable model, effectively balancing performance with operational costs.

### 2.2 LLM Routing

LLM routing seeks to identify the most suitable language model for a given query, with various strategies proposed. Early methods include LLM-Blender (Jiang et al., 2023), which employs an ensemble framework querying multiple LLMs to select the optimal response, and AutoMix (Aggarwal et al., 2024), which utilizes a smaller model for self-verification before potentially escalating to a larger model. While these can improve performance, their reliance on multiple querying inherently increases latency. Other strategies, such as HybridLLM (Ding et al., 2024) and RouteLLM (Ong

et al., 2025), focus on training a binary classifier to choose between a human-defined strong and weak model. However, these methods' efficacy is highly contingent on the subjective definition of model strength and can be computationally expensive when applied to a large pool of LLMs. More recent research has shifted towards multi-LLM routing. RouterDC (Chen et al., 2024),C2MAB-V (Dai et al., 2024), and Prompt-to-Leaderboard (Frick et al., 2025) trains a parametric router to route queries. Concurrently, ModelSpider (Zhang et al., 2023) and EmbedLLM (Zhuang et al., 2025) encode LLMs into learnable representations to facilitate routing. Despite these advancements, a significant limitation is the need to retrain the entire routing mechanism when new models are introduced. Addressing this, Model-SAT (Zhang et al., 2025) aimed to resolve the retraining weakness through human-defined, model-independent capability decompositions. However, its reliance on predefined capability sets undermined adaptability to new capability dimensions.

## 3 Methodology

In this section, we introduce **InferenceDynamics**, which involves: (i) identifying the knowledge and capability required for a given query, (ii) quantifying the knowledge and capability of LLMs, and (iii) routing queries to LLMs based on their scores.

### 3.1 Problem Setup

Let $\mathcal{M}_T = \{M_1, M_2, \ldots, M_t\}$ denote a set of LLMs, and let $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_n$ be a dataset where $\boldsymbol{x}_i$ represents a query and $y_i$ its corresponding ground truth. For an unseen query $\boldsymbol{x} \in \mathcal{Q}$, where $x \notin \mathcal{D}$, LLM routing is formalized as a function $\mathcal{R} : \mathcal{Q} \to \mathcal{M}_T$. This function maps the query $x$ to the model $M_{\text{best}} \in \mathcal{M}_T$ that is considered most suitable, based on a joint assessment of both cost and performance. Our objective is to develop a routing algorithm with the dataset $\mathcal{D}$, that effectively generalizes to OOD queries.

### 3.2 Knowledge and Capability Generation

It is widely acknowledged that no single LLM demonstrates universal proficiency across the full spectrum of query types. Previous research (Wang et al., 2024c; Li et al., 2024c) substantiates that distinct queries necessitate specific underlying capabilities and domain-specific knowledge. Accordingly, assessing an LLM's aptitude for a given query necessitates identifying the requisite capabilities and knowledge pertinent to that query. Let $\mathcal{C}$ denote the set of defined LLM capabilities and $\mathcal{K}$ represent the world knowledge space. For a given query $x$, we utilize an auxiliary LLM $\mathcal{M} \notin \mathcal{M}_T$ to predict two sets: $\mathcal{C}_x = \{c_1, c_2, \cdots \mid c_i \in \mathcal{C}\}$: This set comprises the capabilities deemed necessary to address query $x$, ranked in descending order of importance. $\mathcal{K}_x = \{k_1, k_2, \cdots \mid k_i \in \mathcal{K}\}$: This set encompasses the knowledge areas considered essential for resolving query $x$, also ranked in descending order of importance.

Following Minaee et al. (2024), we categorize capabilities into reasoning, comprehension, instruction following, agentic, knowledge retrieval, coding, and multilingual. With regard to the knowledge dimension ($\mathcal{K}_x$), we impose no predefined constraints to fully accommodate its inherent diversity.

### 3.3 Scoring

To quantify the proficiency of a model $M_t$ with respect to specific capabilities and knowledge, we utilize the accessible set $\mathcal{D}$. The performance score $s_i^t$ of model $M_t$ for a given query-response pair $(\boldsymbol{x_i}, y_i) \in \mathcal{D}_{index}$ is determined by averaging over $K$ independent trials:

$$s_i^t = \frac{1}{K} \sum_{k=1}^{K} eval(M_t(\boldsymbol{x_i})_k, y_i)$$

where $M_t(\boldsymbol{x_i})_k$ is the model's $k$-th generated response to the input query $\boldsymbol{x_i}$, and $eval(\cdot, \cdot)$ represents the query-specific evaluation metric employed to compare the model's response against the ground truth $y_i$. To incorporate the trade-off between performance and computational expenditure, we record the average computational cost $c_i^t$ incurred by model $M_t$ when processing query $\boldsymbol{x_i}$.

Subsequent to the identification of the knowledge and capability sets and computing the scores for all queries in the set $\mathcal{D}$, we define a refined score for model $M_t$. This score, $S_\beta^\alpha(M_t, \boldsymbol{x_i}, e)$, quantifies the model's effectiveness for a specific element $e$ (which can be a knowledge item $k \in \mathcal{K}_{\boldsymbol{x_i}}$ or a capability $c \in \mathcal{C}_{\boldsymbol{x_i}}$) associated with query $\boldsymbol{x_i}$. Illustrating with a knowledge element $k$, this score is formulated as:

$$S_\beta^\alpha(M_t, \boldsymbol{x_i}, k) = \sum_{j=1}^{|\mathcal{K}_{\boldsymbol{x_i}}|} (s_i^t - \beta c_i^t) \mathbb{1}(k = k_j) \frac{\alpha^{j-1}}{\sum_{m=1}^{|\mathcal{K}_{\boldsymbol{x_i}}|} \alpha^{m-1}}$$

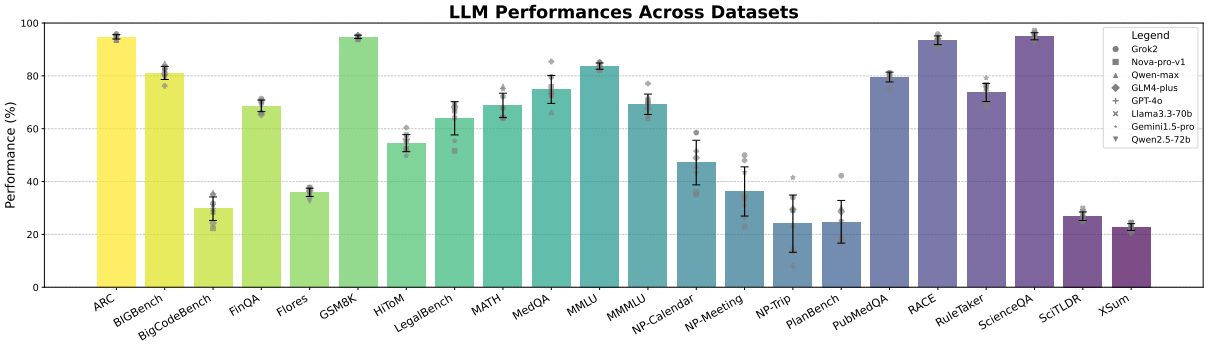


Figure 2: LLM performances across 20 datasets in **RouteMix**. Dataset labels including "*PlanBench*" indicate subsets of the PlanBench benchmark. For detailed metric information, refer to Appx. §A.

In this formulation, the hyperparameter $\alpha$ serves to attenuate the influence of less critical knowledge elements, based on their rank $j$. The hyperparameter $\beta$ acts as a coefficient penalizing higher computational costs. The denominator, $\sum_{k=1}^{|\mathcal{K}_{\boldsymbol{x}_i}|} \alpha^{k-1}$, functions as a normalization factor, ensuring that each query contributes equitably to the knowledge score, regardless of the number of knowledge elements it encompasses.

Building upon these per-query, per-element scores, the aggregate score of model $M_t$ for a specific knowledge element $k$ across the entire indexing dataset $\mathcal{D}$ is computed as:

$$S_\beta^\alpha(\mathcal{M}_t, \mathcal{D}, k) = \frac{1}{|\mathcal{D}^k|} \sum_{i=1}^{N} S_\beta^\alpha(M_t, \boldsymbol{x_i}, k)$$

where $\mathcal{D}^k = \{(\boldsymbol{x}_i, y_i) \mid k \in \mathcal{K}_i\}$ denotes the subset of query-response pairs in which knowledge $k$ is present in the knowledge set. A similar methodology is employed for the computation of aggregate capability scores.

### 3.4 Routing when inference

For an unseen query $\boldsymbol{x}$ with its knowledge and capability sets, we compute the knowledge score $KS$ and capability score $CS$ for each candidate model $M_t$ to guide routing. The knowledge score is given by:

$$KS^\alpha(M_t, \boldsymbol{x}) = \sum_{i=1}^{|\mathcal{K}_{\boldsymbol{x}}|} S_\beta^\alpha(M_t, \mathcal{D}, k_i) \frac{\alpha^{i-1}}{\sum_{m=1}^{|\mathcal{K}_{\boldsymbol{x}}|} \alpha^{m-1}}, \quad (1)$$

The capability score, $CS^\alpha(M_t, \boldsymbol{x})$, is computed analogously. Normalization across both knowledge and capability score calculations ensures that these two distinct types of scores are on a comparable scale, facilitating a balanced routing decision.

The final routing decision is determined by the following algorithm:

$$\mathcal{R}_{\mathcal{M}_T}(\boldsymbol{x}) = \arg\max_{M_t \in \mathcal{M}_T} (\gamma KS^\alpha(M_t, \boldsymbol{x}) + \delta CS^\alpha(M_t, \boldsymbol{x})) \quad (2)$$

which aims to identify the model with the highest weighted average of the knowledge and capability scores. A key advantage of this framework is its adaptability. New LLMs are efficiently integrated by evaluating them on $\mathcal{D}$ to quantify their knowledge and capability scores, which are then used in routing. Similarly, when queries introduce novel knowledge, the LLMs' scores for this new knowledge can be computed and integrated, refining subsequent routing decisions.

## 4 Experiment

### 4.1 Dataset

In this section, we introduce our comprehensive dataset: **RouteMix**, which consist of the Index Set and Evaluation Set.

#### 4.1.1 Index Set

The term 'Index Set' designates the dataset utilized during the development of our routing algorithm. Given that our methodology is parameter-free, this nomenclature serves to differentiate it from datasets conventionally used in training-dependent methods. The 'Index Set' is thus employed primarily for characterizing and indexing the capabilities and knowledge of LLMs. To construct a sufficiently diverse 'Index Set' for robust LLM profiling, we have curated 20 distinct datasets. These datasets span a wide array of domains and are instrumental in quantifying the specific knowledge and capabilities of each model. Comprehensive details regarding the statistics, data processing methodologies, and evaluation metrics for each dataset are presented in Appx. §A.

| Method | MMLU-Pro | GPQA | BigGenBench | LiveBench | Avg. |
|---|---|---|---|---|---|
| **Single Large Language Model** | | | | | |
| Gemini-1.5-Pro | **82.83** | 75.76 | 80.92 | 53.79 | 73.33 |
| GPT-4o | 79.71 | 74.24 | **85.36** | 49.62 | 72.23 |
| Grok-2 | 80.14 | 76.26 | 83.66 | 53.26 | 73.33 |
| Qwen2.5-Max | 75.86 | 71.21 | 82.48 | 52.77 | 70.58 |
| GLM-4-Plus | 79.06 | 75.76 | 83.27 | 47.32 | 71.35 |
| Nova-Pro | 77.49 | 70.20 | 83.01 | 44.38 | 68.77 |
| Llama-3.3-70B-Instruct | 76.27 | 69.70 | 78.17 | 50.67 | 68.70 |
| Qwen-2.5-72B-Instruct | 75.41 | 73.23 | 82.61 | 49.83 | 70.27 |
| Random | 78.26 | 72.22 | 82.61 | 48.83 | 70.48 |
| **Routing Algorithm (Ours)** | | | | | |
| Routing by *Knowledge* | 80.99 | **78.28** | 82.61 | 53.17 | 73.76 |
| Routing by *Capability* | 80.09 | 76.26 | 84.18 | 53.65 | 73.55 |
| *Inference Dynamics* | 80.85 | 77.78 | 84.31 | **55.57** | **74.55** |

Table 1: LLM routing results across four benchmarks are presented. The metrics we used are introduced in §4.2. The best performances are **bold-faced**, while the second-best performances are underlined. "Routing by Knowledge" denotes routing decisions made solely based on the knowledge score, whereas "Routing by Capability" refers to routing based only on the capability score. "Mixed Routing" indicates a simultaneous consideration of both scores during the routing process.

#### 4.1.2 Evaluation Set

We incorporate four benchmarks that comprehensively evaluate the LLM as the evaluation set of **RouteMix**: (i) MMLU-Pro (Wang et al., 2024b) spans 14 diverse domains and includes approximately 12,000 instances. (ii) GPQA (Rein et al., 2023) consists of multiple choice questions at the graduate level in subdomains of physics, chemistry, and biology. For our evaluation, we utilize the Diamond subset. (iii) BigGenBench (Kim et al., 2024) comprises 77 distinct tasks evaluating core abilities of LLM, with a total of 765 human-written instances. (iv) LiveBench (White et al., 2025) is a real-time updated benchmark with 18 tasks across 6 categories, including math, reasoning, coding, data analysis, language and instruction following. In the evaluation, we utilize the snapshot released on 2024-11-25.

### 4.2 Experiment Setup

For the candidate models, we select eight high-performing LLMs: Gemini-1.5-Pro(Reid et al., 2024), GPT-4o (Hurst et al., 2024), Grok-2, Qwen2.5-Max (Yang et al., 2024), GLM-4-Plus (Zeng et al., 2024), Nova-Pro (Intelligence, 2024), Llama-3.3-70B-Instruct (AI@Meta, 2024), and Qwen-2.5-72B-Instruct (Yang et al., 2024). To ensure a fair comparison when testing these models, all parameters and the input prompt are kept consistent across evaluations. To derive the Knowledge and Capability attributes, we employ GPT-4o-mini to generate these characteristics. Since knowledge may include semantically similar phrases, we utilize MiniLM-L6 (Wang et al., 2020) to consolidate Knowledge entries with a cosine similarity score greater than 0.6. Additionally, Knowledge entries with a frequency lower than 10 are filtered out and designated as 'Other' knowledge. When the system encounters a query containing previously unseen knowledge elements, these are also classified as 'Other' knowledge. By default, for unconstrained routing, the parameters $\alpha$ and $\beta$ are set to 0.5 and 0, respectively. The weights for the Knowledge and Capability scores are both set to 1.0 by default. In terms of evaluation, the exact match score is employed for both the MMLU-Pro and GPQA datasets. For BigGenBench, we follow the methodology proposed by Sprague et al. (2025), using GPT-4o-mini as a language model-based judge. Instances receiving a score greater than 4 are classified as correct. For LiveBench, we adhere to the original evaluation script, and the metric is average score across six categories.

### 4.3 Capability and Knowledge Quantification

The performance of the candidate models on the Index Set is presented in Fig. 2. Generally, these models do not exhibit substantial performance distinctions when evaluated across the entire Index Set. However, their relative strengths become apparent on specific subsets, where different models tend to outperform one another. This observation suggests that the model pool consists of LLMs with broadly comparable overall abilities, yet with varying specializations.

Subsequent to the computation of average performance scores, the top four models are selected for more detailed analysis. Their respective capability and knowledge scores are visualized in Fig. 1. For clarity and simplification in this visualization, we focus on the eight most frequently occurring knowledge elements and capabilities within the Index Set. The fact that the highest-scoring model changes with the specific knowledge or capability further substantiates the premise: LLMs, even those exhibiting similar aggregate performance levels, possess distinct areas of specialized expertise.

### 4.4 Optimal Routing

The optimal routing results, presented in Tab. 1, highlight the clear superiority of our proposed routing strategies. Among these, our *Mixed Routing* strategy, which combines both *Knowledge* and *Capability* scores, achieves the highest average performance, outperforming the best single model, Gemini-1.5-Pro, by a margin of 1.28. This strategy secures top results on LiveBench and ranks second on GPQA and BigGenBench, demonstrating the effectiveness and versatility of our comprehensive routing algorithm. Additionally, the Routing by *Knowledge* and Routing by *Capability* approaches also deliver strong results, consistently surpassing the best single model and significantly outperforming random routing on average. Notably, Routing by *Knowledge* excels in knowledge-intensive tasks, achieving the best score on GPQA and the second-best on MMLU-Pro. This underscores its ability to effectively direct queries requiring accurate factual recall and nuanced domain understanding. Similarly, Routing by *Capability* performs exceptionally well on capability-driven benchmarks, particularly on BigGenBench, highlighting the importance of leveraging a model's inherent strengths in complex reasoning and generation tasks. Both approaches play an integral role in the success of the *Mixed Routing* system.

These findings also emphasize that no single LLM universally dominates across all tasks. Models like Gemini-1.5-Pro and GPT-4o exhibit varying strengths, further validating the necessity and advantages of intelligent LLM routing systems.

### 4.5 Routing with Constraints

To investigate the system's performance under varying cost constraints, we systematically adjusted the $\beta$ parameter, maintaining all other experimental configurations as previously defined. The evaluation employed two distinct metrics. The first metric, termed **Performance Ratio**, quantifies the efficacy of the *Mixed Routing* strategy. This is calculated as the ratio of the performance achieved by *Mixed Routing* to that of the best-performing single candidate LLM on the respective benchmark. The second metric, **Cost Ratio**, assesses the economic efficiency of the routing algorithm. It is defined as the total cost incurred by the routing process (encompassing both knowledge generation and capability assessment costs) relative to the operational cost of the best-performing single LLM.

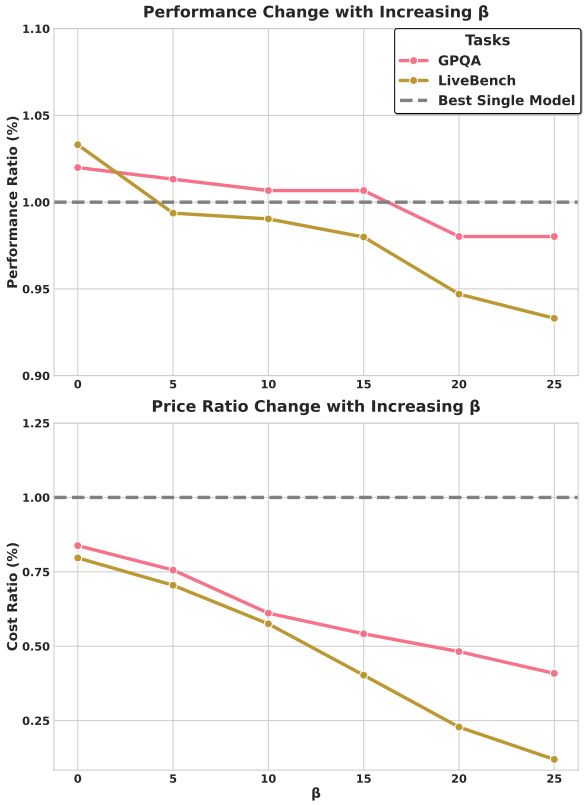


Figure 3: Performance Ratio (%) and Cost Ratio (%) variation on GPQA and LiveBench. The "Best Single Model" refers to the most performant LLM for each task.

The empirical results of this sensitivity analysis are depicted in Fig. 3. In scenarios without stringent price constraints (i.e., $\beta = 0$ ), our routing system demonstrates superior performance compared to the best single model, while operating at approximately $80\%$ of the latter's budget. As the $\beta$ parameter is incrementally increased, thereby prioritizing cost reduction, the operational cost of the routing algorithm decreases significantly. Concurrently, the system maintains a competitive performance level relative to the best single model. Notably, at a $\beta$ value of 15, our routing algorithm

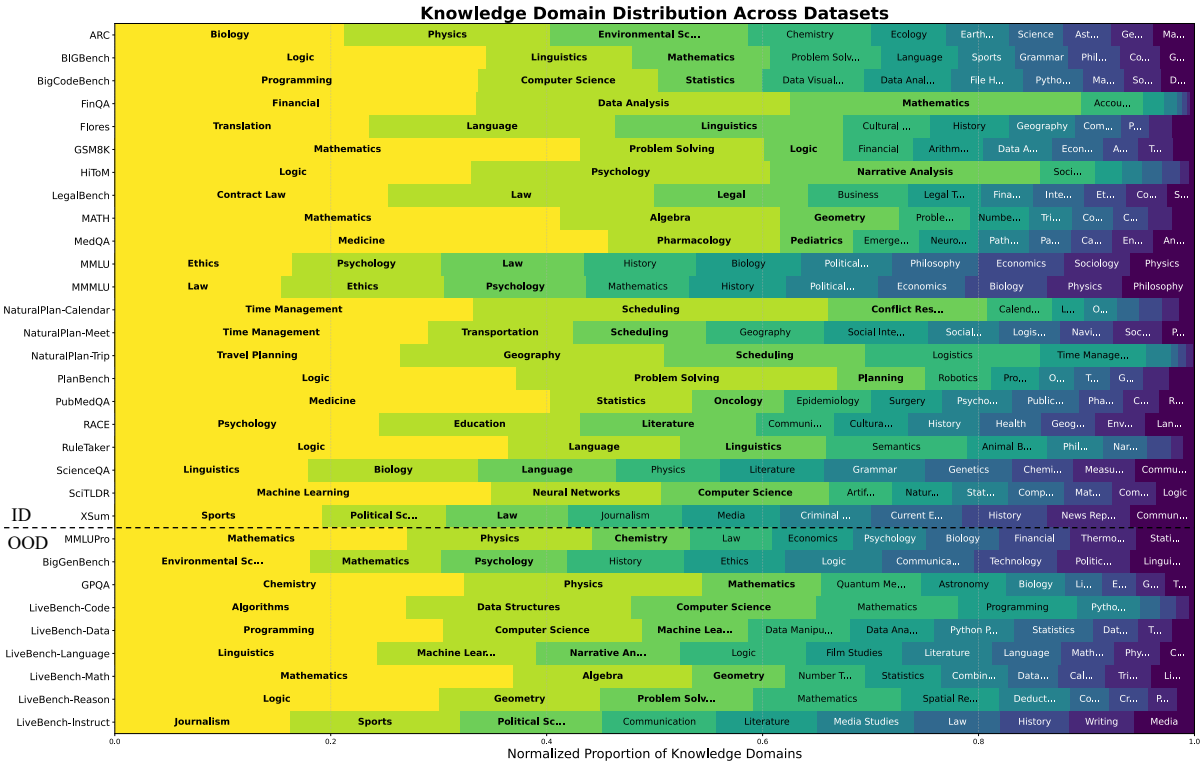


Figure 4: Distribution of knowledge domains across 24 datasets in **RouteMix**. The In-Domain (ID) subset is utilized for quantifying *Knowledge* and *Capability*, while the Out-of-Domain (OOD) subset is employed for evaluating the routing algorithm. Dataset labels including "*LiveBench*" indicate subsets of the LiveBench benchmark, and labels including "*NaturalPlan*" similarly denote subsets of the NaturalPlan benchmark. The algorithm to compute the normalized proportion is included in Appx. §B.

achieves performance nearly equivalent to the best single model but utilizes only approximately half the associated cost.

An interesting observation is the differential sensitivity of benchmarks to changes in $\beta$. Specifically, the performance and cost metrics for LiveBench, a text generation benchmark, exhibit more pronounced variations in response to adjustments in $\beta$ compared to those observed for GPQA, a question-answering benchmark. This suggests that text generation tasks are more sensitive to the price penalty imposed by $\beta$ than QA tasks.

## 5 Analysis

### 5.1 Model Selection

The distribution of model selections under various conditions is illustrated in Fig. 5. Consistent with findings in previous works (Chen et al., 2024; Frick et al., 2025), cost-efficient models are infrequently selected in optimal routing scenarios; instead, the strategy predominantly converges towards higher-performing models. For comprehensive benchmarks such as BigGenBench, our approach primarily routes queries to expensive yet high-performing models like GPT-4o and Grok-2, reflecting a tendency to leverage top-tier capabilities for broad-ranging tasks. Conversely, for task sets demanding highly specialized capabilities, the routing algorithm typically assigns queries directly to the most proficient model. For instance, within the coding subset of LiveBench, 91% of queries are routed to Qwen-Max, which demonstrates the strongest coding capabilities. This model's leading performance in coding is further corroborated by its results on BigCodeBench and its specific Coding capability score, as detailed in Fig. 1 and Fig. 2, respectively. These observations collectively indicate that our routing algorithm effectively directs queries to the most suitable models based on specific task demands.

In the context of cost-constrained routing, an increasing cost penalty prompts the router to progressively shift its selections from expensive, top-performing models towards more affordable, albeit less powerful, alternatives.

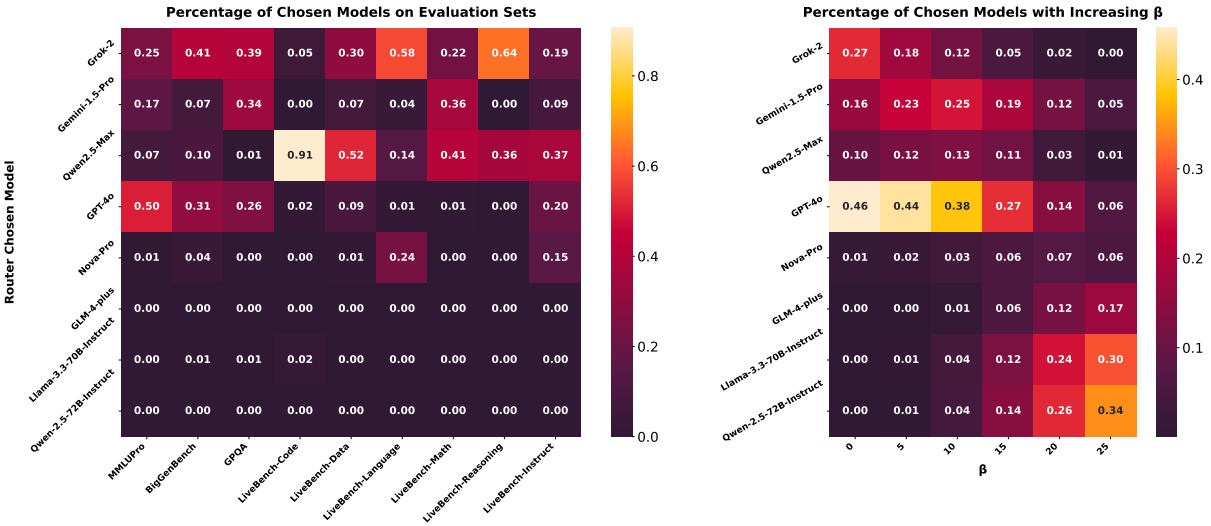


Figure 5: Comparative distribution of router-selected models. Lighter colors signify a higher selection ratio for a given model. The left panel details model selection across evaluation benchmarks using the Optimal *Mixed Routing* strategy. The right panel illustrates the impact of an increasing cost penalty coefficient ($\beta$) on the model selection distribution.

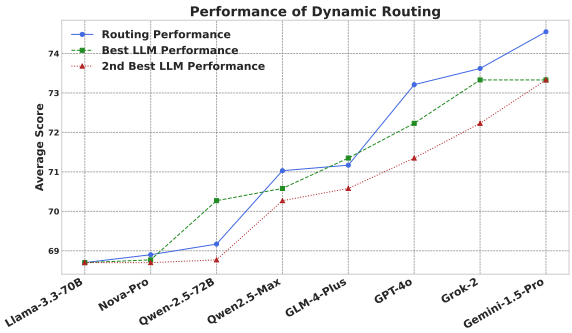


Figure 6: Routing Performance (%) in Dynamic LLM Pools.

### 5.2 Knowledge Distribution

As shown in Fig. 4, the distribution of generated knowledge highlights the *RouteMix* benchmark's comprehensive span of knowledge domains, ranging from highly specific academic areas to practical applications. On datasets with broad knowledge requirements, such as MMLU-Pro, the generated knowledge exhibits a relatively balanced distribution. For benchmarks targeting one or two specific domains, like MATH-500, the model typically generates more fine-grained knowledge components related to the core domain. This facilitates a more nuanced quantification of the model's domain-specific knowledge.

### 5.3 Dynamics Routing

In this section, we investigate the scalability of our framework with respect to dynamic LLM pools. The corresponding results are presented in Fig. 6. The x-axis in this figure represents the progressive addition of specific new models to the LLM candidate pool. Initially, the pool consists solely of Llama-3.3-70B; subsequently, one new model is added to the candidate pool at each increment along the x-axis. Notably, our routing algorithm consistently maintains a top-2 performance ranking and surpasses the best single model across the five evaluated candidate pool configurations. This outcome demonstrates the robust scalability of our framework when new models are introduced, crucially without the need for any additional training.

## 6 Conclusions

This paper introduces **InferenceDynamics**, a scalable and adaptable LLM routing framework that quantifies model capabilities and domain-specific knowledge to match queries with the most suitable LLMs. Evaluated on the new comprehensive RouteMix benchmark, InferenceDynamics demonstrated superior performance, outperforming the best single LLM by 1.28 on average and achieving comparable results at approximately half the cost under budget constraints. Key contributions include the **RouteMix** dataset for evaluating generalization and the **InferenceDynamics** algorithm, which generalizes to unseen queries and effectively routes them within dynamic model pools without retraining. Our work enables more efficient and tailored utilization of the diverse LLM ecosystem.

## Limitations

Despite the promising results and the robust design of InferenceDynamics, several limitations warrant discussion and offer avenues for future research:

**Niche Suitability for Highly Constrained Environments** InferenceDynamics is engineered for scalability and adaptability, demonstrating its strengths when dealing with a large, diverse, and evolving pool of LLMs, or when new capability and knowledge domains are frequently encountered. However, in scenarios characterized by a very limited and static set of LLMs and a narrowly defined, unchanging task scope, a dedicated learning-based routing approach (e.g., a fine-tuned classifier) might be more appropriate or yield marginally superior, hyper-specialized performance. Our framework prioritizes generalizability and efficient adaptation to dynamic conditions, which is a different niche than hyper-optimization for small, fixed-scope problems.

**Benchmark-Driven Evaluation vs. Real-World Application Complexity** The current evaluation of InferenceDynamics relies on the comprehensive RouteMix dataset, which is composed of various established benchmarks. While these benchmarks cover a wide array of tasks and domains, they may not fully capture the intricacies and dynamic nature of real-world application systems. For instance, the utility and performance of InferenceDynamics in more complex, interactive systems like multi-agent environments, where task allocation might depend on evolving collaborative states, have not been explicitly tested. Exploring the deployment and effectiveness of InferenceDynamics in such real-application scenarios remains an important direction for future work.

Addressing these limitations will be crucial for broadening the applicability and enhancing the robustness of InferenceDynamics and similar LLM routing frameworks.

## Ethics Statement

Our study utilizes publicly available datasets and accesses Large Language Models (LLMs) through their respective APIs. The ethical considerations pertaining to this research are as follows:

**Datasets:** This research exclusively employs publicly available datasets, strictly for academic research purposes. We affirm that no personally identifiable information or private data was involved in our study.

**LLM APIs:** Our application of LLMs via APIs rigorously conforms to the policies set forth by the API providers. This includes adherence to fair use guidelines and respect for intellectual property rights.

**Transparency:** In line with standard academic research practices, we provide detailed descriptions of our methodology and the prompts utilized in our experiments. Furthermore, the source code for this research will be made publicly available upon the acceptance of this paper.

## References


Pranjal Aggarwal, Aman Madaan, Ankit Anand, Srividya Pranavi Potharaju, Swaroop Mishra, Pei Zhou, Aditya Gupta, Dheeraj Rajagopal, Karthik Kappaganthu, Yiming Yang, Shyam Upadhyay, Manaal Faruqui, and Mausam. 2024. Automix: Automatically mixing language models. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

AI@Meta. 2024. Llama 3 model card.

Isabel Cachola, Kyle Lo, Arman Cohan, and Daniel S. Weld. 2020. TLDR: extreme summarization of scientific documents. In *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020*, volume EMNLP 2020 of *Findings of ACL*, pages 4766–4777. Association for Computational Linguistics.

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

Shuhao Chen, Weisen Jiang, Baijiong Lin, James T. Kwok, and Yu Zhang. 2024. Routerdc: Query-based router by dual contrastive learning for assembling large language models. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

Zhijun Chen, Jingzheng Li, Pengpeng Chen, Zhuoran Li, Kai Sun, Yuankai Luo, Qianren Mao, Dingqi Yang, Hailong Sun, and Philip S. Yu. 2025. Harnessing multiple large language models: A survey on LLM ensemble. *CoRR*, abs/2502.18036.

Zhiyu Chen, Wenhu Chen, Charese Smiley, Sameena Shah, Iana Borova, Dylan Langdon, Reema Moussa, Matt Beane, Ting-Hao Kenneth Huang, Bryan R. Routledge, and William Yang Wang. 2021. Finqa: A dataset of numerical reasoning over financial data.

In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 3697–3711. Association for Computational Linguistics.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv:1803.05457v1*.

Peter Clark, Oyvind Tafjord, and Kyle Richardson. 2020. Transformers as soft reasoners over language. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, pages 3882–3890. ijcai.org.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *CoRR*, abs/2110.14168.

Xiangxiang Dai, Jin Li, Xutong Liu, Anqi Yu, and John C. S. Lui. 2024. Cost-effective online multi-llm selection with versatile reward models. *CoRR*, abs/2405.16587.

Dujian Ding, Ankur Mallick, Chi Wang, Robert Sim, Subhabrata Mukherjee, Victor Rühle, Laks V. S. Lakshmanan, and Ahmed Hassan Awadallah. 2024. 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*. OpenReview.net.

Evan Frick, Connor Chen, Joseph Tennyson, Tianle Li, Wei-Lin Chiang, Anastasios N. Angelopoulos, and Ion Stoica. 2025. Prompt-to-leaderboard. *CoRR*, abs/2502.14855.

Naman Goyal, Cynthia Gao, Vishrav Chaudhary, Peng-Jen Chen, Guillaume Wenzek, Da Ju, Sanjana Krishnan, Marc'Aurelio Ranzato, Francisco Guzmán, and Angela Fan. 2022. The flores-101 evaluation benchmark for low-resource and multilingual machine translation. *Trans. Assoc. Comput. Linguistics*, 10:522–538.

Neel Guha, Julian Nyarko, Daniel E. Ho, Christopher Ré, Adam Chilton, K. Aditya, Alex Chohlas-Wood, Austin Peters, Brandon Waldon, Daniel N. Rockmore, Diego Zambrano, Dmitry Talisman, Enam Hoque, Faiz Surani, Frank Fagan, Galit Sarfaty, Gregory M. Dickinson, Haggai Porat, Jason Hegland, Jessica Wu, Joe Nudell, Joel Niklaus, John J. Nay, Jonathan H. Choi, Kevin Tobia, Margaret Hagan, Megan Ma, Michael A. Livermore, Nikon Rasumov-Rahe, Nils Holzenberger, Noam Kolt, Peter Henderson, Sean Rehaag, Sharad Goel, Shang Gao, Spencer Williams, Sunny Gandhi, Tom Zur, Varun Iyer, and Zehua Li. 2023. Legalbench: A collaboratively built benchmark for measuring legal reasoning in large language models. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *CoRR*, abs/2009.03300.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the MATH dataset. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*.

Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, Aleksander Madry, Alex Baker-Whitcomb, Alex Beutel, Alex Borzunov, Alex Carney, Alex Chow, Alex Kirillov, Alex Nichol, Alex Paino, Alex Renzin, Alex Tachard Passos, Alexander Kirillov, Alexi Christakis, Alexis Conneau, Ali Kamali, Allan Jabri, Allison Moyer, Allison Tam, Amadou Crookes, Amin Tootoonchian, Ananya Kumar, Andrea Vallone, Andrej Karpathy, Andrew Braunstein, Andrew Cann, Andrew Codispoti, Andrew Galu, Andrew Kondrich, Andrew Tulloch, Andrey Mishchenko, Angela Baek, Angela Jiang, Antoine Pelisse, Antonia Woodford, Anuj Gosalia, Arka Dhar, Ashley Pantuliano, Avi Nayak, Avital Oliver, Barret Zoph, Behrooz Ghorbani, Ben Leimberger, Ben Rossen, Ben Sokolowsky, Ben Wang, Benjamin Zweig, Beth Hoover, Blake Samic, Bob McGrew, Bobby Spero, Bogo Giertler, Bowen Cheng, Brad Lightcap, Brandon Walkin, Brendan Quinn, Brian Guarraci, Brian Hsu, Bright Kellogg, Brydon Eastman, Camillo Lugaresi, Carroll L. Wainwright, Cary Bassin, Cary Hudson, Casey Chu, Chad Nelson, Chak Li, Chan Jun Shern, Channing Conger, Charlotte Barette, Chelsea Voss, Chen Ding, Cheng Lu, Chong Zhang, Chris Beaumont, Chris Hallacy, Chris Koch, Christian Gibson, Christina Kim, Christine Choi, Christine McLeavey, Christopher Hesse, Claudia Fischer, Clemens Winter, Coley Czarnecki, Colin Jarvis, Colin Wei, Constantin Koumouzelis, and Dane Sherburn. 2024. Gpt-4o system card. *CoRR*, abs/2410.21276.

Amazon Artificial General Intelligence. 2024. The amazon nova family of models: Technical report and model card. *Amazon Technical Reports*.

Dongfu Jiang, Xiang Ren, and Bill Yuchen Lin. 2023. Llm-blender: Ensembling large language models with pairwise ranking and generative fusion. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 14165–14178. Association for Computational Linguistics.

Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. 2020. What disease does this patient have? A large-scale open domain question answering dataset from medical exams. *CoRR*, abs/2009.13081.

Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William W. Cohen, and Xinghua Lu. 2019. Pubmedqa: A dataset for biomedical research question answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 2567–2577. Association for Computational Linguistics.

Seungone Kim, Juyoung Suk, Ji Yong Cho, Shayne Longpre, Chaeeun Kim, Dongkeun Yoon, Guijin Son, Yejin Choi, Sheikh Shafayat, Jinheon Baek, Sue Hyun Park, Hyeonbin Hwang, Jinkyung Jo, Hyowon Cho, Haebin Shin, Seongyun Lee, Hanseok Oh, Noah Lee, Namgyu Ho, Se June Joo, Miyoung Ko, Yoonjoo Lee, Hyungjoo Chae, Jamin Shin, Joel Jang, Seonghyeon Ye, Bill Yuchen Lin, Sean Welleck, Graham Neubig, Moontae Lee, Kyungjae Lee, and Minjoon Seo. 2024. The biggen bench: A principled benchmark for fine-grained evaluation of language models with language models. *CoRR*, abs/2406.05761.

Steven Kolawole, Don Kurian Dennis, Ameet Talwalkar, and Virginia Smith. 2024. Revisiting cascaded ensembles for efficient inference. *CoRR*, abs/2407.02348.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard H. Hovy. 2017. RACE: large-scale reading comprehension dataset from examinations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 785–794. Association for Computational Linguistics.

Jiawei Li, Yizhe Yang, Yu Bai, Xiaofeng Zhou, Yinghao Li, Huashan Sun, Yuhang Liu, Xingpeng Si, Yuhao Ye, Yixiao Wu, Yiguan Lin, Bin Xu, Ren Bowen, Chong Feng, Yang Gao, and Heyan Huang. 2024a. Fundamental capabilities of large language models and their applications in domain scenarios: A survey. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 11116–11141. Association for Computational Linguistics.

Junyou Li, Qin Zhang, Yangbin Yu, Qiang Fu, and Deheng Ye. 2024b. More agents is all you need. *CoRR*, abs/2402.05120.

Moxin Li, Yong Zhao, Yang Deng, Wenxuan Zhang, Shuaiyi Li, Wenya Xie, See-Kiong Ng, and Tat-Seng Chua. 2024c. Knowledge boundary of large language models: A survey. *CoRR*, abs/2412.12472.

Tian Liang, Zhiwei He, Wenxiang Jiao, Xing Wang, Yan Wang, Rui Wang, Yujiu Yang, Shuming Shi, and Zhaopeng Tu. 2024. Encouraging divergent thinking in large language models through multi-agent debate. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pages 17889–17904. Association for Computational Linguistics.

Pan Lu, Swaroop Mishra, Tanglin Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. 2022. Learn to explain: Multimodal reasoning via thought chains for science question answering. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Andrea Matarazzo and Riccardo Torlone. 2025. A survey on large language models with some insights on their capabilities and limitations. *CoRR*, abs/2501.04040.

Shervin Minaee, Tomás Mikolov, Narjes Nikzad, Meysam Chenaghlu, Richard Socher, Xavier Amatriain, and Jianfeng Gao. 2024. Large language models: A survey. *CoRR*, abs/2402.06196.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 1797–1807. Association for Computational Linguistics.

Isaac Ong, Amjad Almahairi, Vincent Wu, Wei-Lin Chiang, Tianhao Wu, Joseph E. Gonzalez, M. Waleed Kadous, and Ion Stoica. 2025. Routellm: Learning to route llms from preference data. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.

Machel Reid, Nikolay Savinov, Denis Teplyashin, Dmitry Lepikhin, Timothy P. Lillicrap, Jean-Baptiste Alayrac, Radu Soricut, Angeliki Lazaridou, Orhan Firat, Julian Schrittwieser, Ioannis Antonoglou, Rohan Anil, Sebastian Borgeaud, Andrew M. Dai, Katie Millican, Ethan Dyer, Mia Glaese, Thibault Sottiaux, Benjamin Lee, Fabio Viola, Malcolm Reynolds, Yuanzhong Xu, James Molloy, Jilin Chen, Michael Isard, Paul Barham, Tom Hennigan, Ross McIlroy, Melvin Johnson, Johan Schalkwyk, Eli Collins, Eliza Rutherford, Erica Moreira, Kareem Ayoub, Megha Goel, Clemens Meyer, Gregory Thornton, Zhen Yang, Henryk Michalewski, Zaheer Abbas, Nathan Schucher, Ankesh Anand, Richard Ives, James Keeling, Karel Lenc, Salem Haykal, Siamak Shakeri, Pranav Shyam, Aakanksha Chowdhery, Roman Ring, Stephen Spencer, Eren Sezener, and et al.

2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *CoRR*, abs/2403.05530.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. 2023. GPQA: A graduate-level google-proof q&a benchmark. *CoRR*, abs/2311.12022.

Zayne Rea Sprague, Fangcong Yin, Juan Diego Rodriguez, Dongwei Jiang, Manya Wadhwa, Prasann Singhal, Xinyu Zhao, Xi Ye, Kyle Mahowald, and Greg Durrett. 2025. To cot or not to cot? chain-of-thought helps mainly on math and symbolic reasoning. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V. Le, Ed H. Chi, Denny Zhou, and Jason Wei. 2023. Challenging big-bench tasks and whether chain-of-thought can solve them. In *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 13003–13051. Association for Computational Linguistics.

Karthik Valmeekam, Matthew Marquez, Alberto Olmo Hernandez, Sarath Sreedharan, and Subbarao Kambhampati. 2023. Planbench: An extensible benchmark for evaluating large language models on planning and reasoning about change. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Qineng Wang, Zihao Wang, Ying Su, Hanghang Tong, and Yangqiu Song. 2024a. Rethinking the bounds of LLM reasoning: Are multi-agent discussions the key? In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 6106–6131. Association for Computational Linguistics.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. 2024b. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

Zhihu Wang, Shiwan Zhao, Yu Wang, Heyuan Huang, Jiaxin Shi, Sitao Xie, Zhixing Wang, Yubo Zhang, Hongyan Li, and Junchi Yan. 2024c. Re-task: Revisiting LLM tasks from capability, skill, and knowledge perspectives. *CoRR*, abs/2408.06904.

Colin White, Samuel Dooley, Manley Roberts, Arka Pal, Benjamin Feuer, Siddhartha Jain, Ravid Shwartz-Ziv, Neel Jain, Khalid Saifullah, Sreemanti Dey, Shubh-Agrawal, Sandeep Singh Sandha, Siddartha V. Naidu, Chinmay Hegde, Yann LeCun, Tom Goldstein, Willie Neiswanger, and Micah Goldblum. 2025. Livebench: A challenging, contamination-limited LLM benchmark. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.

Yufan Wu, Yinghui He, Yilin Jia, Rada Mihalcea, Yulong Chen, and Naihao Deng. 2023. Hi-tom: A benchmark for evaluating higher-order theory of mind reasoning in large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pages 10691–10706. Association for Computational Linguistics.

Frank F. Xu, Yufan Song, Boxuan Li, Yuxuan Tang, Kritanjali Jain, Mengxue Bao, Zora Zhiruo Wang, Xuhui Zhou, Zhitong Guo, Murong Cao, Mingyang Yang, Hao Yang Lu, Amaad Martin, Zhe Su, Leander Maben, Raj Mehta, Wayne Chi, Lawrence Keunho Jang, Yiqing Xie, Shuyan Zhou, and Graham Neubig. 2024. Theagentcompany: Benchmarking LLM agents on consequential real world tasks. *CoRR*, abs/2412.14161.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. 2024. Qwen2.5 technical report. *CoRR*, abs/2412.15115.

Aohan Zeng, Bin Xu, Bowen Wang, Chenhui Zhang, Da Yin, Diego Rojas, Guanyu Feng, Hanlin Zhao, Hanyu Lai, Hao Yu, Hongning Wang, Jiadai Sun, Jiajie Zhang, Jiale Cheng, Jiayi Gui, Jie Tang, Jing Zhang, Juanzi Li, Lei Zhao, Lindong Wu, Lucen Zhong, Mingdao Liu, Minlie Huang, Peng Zhang, Qinkai Zheng, Rui Lu, Shuaiqi Duan, Shudan Zhang, Shulin Cao, Shuxun Yang, Weng Lam Tam, Wenyi Zhao, Xiao Liu, Xiao Xia, Xiaohan Zhang, Xiaotao Gu, Xin Lv, Xinghan Liu, Xinyi Liu, Xinyue Yang, Xixuan Song, Xunkai Zhang, Yifan An, Yifan Xu, Yilin Niu, Yuantao Yang, Yueyan Li, Yushi Bai, Yuxiao Dong, Zehan Qi, Zhaoyu Wang, Zhen Yang, Zhengxiao Du, Zhenyu Hou, and Zihan Wang. 2024. Chatglm: A family of large language mod-

els from GLM-130B to GLM-4 all tools. *CoRR*, abs/2406.12793.

Kai Zhang, Liqian Peng, Congchao Wang, Alec Go, and Xiaozhong Liu. 2024. LLM cascade with multi-objective optimal consideration. *CoRR*, abs/2410.08014.

Yi-Kai Zhang, Ting-Ji Huang, Yao-Xiang Ding, De-Chuan Zhan, and Han-Jia Ye. 2023. Model spider: Learning to rank pre-trained models efficiently. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Yi-Kai Zhang, De-Chuan Zhan, and Han-Jia Ye. 2025. Capability instruction tuning: A new paradigm for dynamic llm routing. *Preprint*, arXiv:2502.17282.

Huaixiu Steven Zheng, Swaroop Mishra, Hugh Zhang, Xinyun Chen, Minmin Chen, Azade Nova, Le Hou, Heng-Tze Cheng, Quoc V. Le, Ed H. Chi, and Denny Zhou. 2024. NATURAL PLAN: benchmarking llms on natural language planning. *CoRR*, abs/2406.04520.

Richard Zhuang, Tianhao Wu, Zhaojin Wen, Andrew Li, Jiantao Jiao, and Kannan Ramchandran. 2025. Embedllm: Learning compact representations of large language models. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.

Terry Yue Zhuo, Minh Chien Vu, Jenny Chim, Han Hu, Wenhao Yu, Ratnadira Widyasari, Imam Nur Bani Yusuf, Haolan Zhan, Junda He, Indraneil Paul, et al. 2024. Bigcodebench: Benchmarking code generation with diverse function calls and complex instructions. *arXiv preprint arXiv:2406.15877*.

## A Benchmark Overview Table

Table 2: Overview of Benchmarks, Data Processing, Prompts, and Metrics

| Benchmark Name | Data Processing Manner | Prompt Type | Metric Used |
|---|---|---|---|
| ARC (Clark et al., 2018) | Sample 500 instances according to the portion of ARC-Easy and ARC-Challenge. | Zero-shot DA | Accuracy |
| BigBench-Hard (Suzgun et al., 2023) | Sample 40 instances from each category except $web_of_lies$, to avoid collision with LiveBench. Formulate into MCQA for Yes/No and QA question. Remain the free-response question unchanged. | Zero-shot CoT | Exact Match (EM) |
| BigCodeBench (Zhuo et al., 2024) | We directly use the BigCodeBench-Hard subset, with 148 instances. | DA for code completion | Pass@1 |
| FinQA (Chen et al., 2021) | Sample 500 instances from the dataset. | CoT from Sprague et al. (2025) | Exact Match(EM) |
| Flores200 (Goyal et al., 2022) | We incorporate the top10 commonly used language except for English. And sample 100 instances for each language. | Translation Prompt | Chrf++ (Goyal et al., 2022) |
| GSM8K (Cobbe et al., 2021) | Sample 500 instances from the dataset. | CoT from Sprague et al. (2025) | Exact Match(EM). |
| HiToM (Wu et al., 2023) | Sample 500 instances under CoT settings. | CoT from Official Repo | Accuracy |
| LegalBench (Guha et al., 2023) | Sample 4 instances from each category except for short answering task, resulting in 616 instances. | Few-shot DA | Accuracy |
| MATH (Hendrycks et al., 2021) | We use the subset MATH-500. | CoT from Sprague et al. (2025) | Exact Match(EM) |
| MedQA (Jin et al., 2020) | Sample 500 instances from the dataset | DA | Accuracy |
| MMLU (Hendrycks et al., 2020) | We sample instances according to the portion of different categories, and make sure each category has at least 10 instances. Resulting in 1262 instances. | DA | Accuracy |
| MMMLU (Hendrycks et al., 2020) | We sample 100 instances for all languages except for English. Result in 1400 instances. | DA | Accuracy |
| NaturalPlan (Zheng et al., 2024) | Sample 200 instances from each subset, including scheduling, calendar meeting, and trip planning. | DA | Accuracy |
| PlanBench (Valmeekam et al., 2023) | Use the subset of PlanGeneration in BlocksWorld. | DA | Accuracy |

Continued on next page...

Table 2 – continued from previous page

| Benchmark Name | Data Processing Manner | Prompt Type | Metric Used |
|---|---|---|---|
| PubMedQA (Jin et al., 2019) | Sample 500 instances from original dataset | DA | Accuracy |
| RACE (Lai et al., 2017) | Sample 500 instances from original dataset | DA | Accuracy |
| RuleTaker (Clark et al., 2020) | Sample 500 instances from original dataset | DA | Accuracy |
| ScienceQA (Lu et al., 2022) | Sample 500 instances which don't have corresponding picture. | DA | Accuracy |
| SciTLDR (Cachola et al., 2020) | Directly use the test set | Summarization Prompt | RogueL. |
| XSum (Narayan et al., 2018) | Sample 500 instances for the dataset | Summarization Prompt | RogueL |

Specifically, when quantifying the capability and knowledge of LLMs for translation and summarization tasks, we establish a performance threshold. An output is considered correct if its evaluation score or relevant metric exceeds this threshold.

## B Knowledge Domain Distribution

The dataset's knowledge domain distribution is determined by a weighted rank approach. For each domain $D \in \mathcal{D}$ (where $\mathcal{D}$ is the set of all unique domains), its frequency at each rank $r$ (denoted $F_{D,r}$, for $r = 1, \ldots, N$) is multiplied by a corresponding rank weight $W_r$ (typically $W_r = 1/r$). These products are summed to yield a weighted score $S_D$:

$$S_D = \sum_{r=1}^{N} (F_{D,r} \times W_r)$$

The final distribution percentage $P_D$ for each domain is then its $S_D$ normalized by the sum of all domain weighted scores ($S_{\text{total}} = \sum_{D' \in \mathcal{D}} S_{D'}$), expressed as a percentage:

$$P_D = \left( \frac{S_D}{\sum_{D' \in \mathcal{D}} S_{D'}} \right) \times 100\%$$

This method ensures higher-ranked domain occurrences contribute more significantly, with all $P_D$ summing to 100%.

## C BigGenBench Evaluation

Following Sprague et al. (2025), we employ GPT-4o-mini as LLM-as-a-Judge to evaluate the BigGen-Bench, and instances with a score larger than 4 is considered correct. The specific prompt is shown below:

**Prompt for evaluation BigGenBench**

Task Description:
An instruction (might include an Input inside it), a response to evaluate, a reference answer that gets a score of 5, and a score rubric representing a evaluation criteria are given.
1. Write a detailed feedback that assess the quality of the response strictly based on the given score rubric, not evaluating in general.
2. After writing a feedback, write a score that is an integer between 1 and 5. You should refer to the score rubric.
3. The output format should look as follows: "Feedback: (write a feedback for criteria) [RESULT] (an integer number between 1 and 5)"
4. Please do not generate any other opening, closing, and explanations.

The instruction to evaluation:
example question

Response to evaluate:
example solution

Reference Answer (Score 5):
reference score

Score Rubrics:
Criteria:
criteria

Description of a Score 1 response:
score1 description

Description of a Score 2 response:
score2 description

Description of a Score 3 response:
score3 description

Description of a Score 4 response:
score4 description

Description of a Score 5 response:
score5 description

Feedback:
Remember, you must strictly evaluate the response based on the given score rubric, and finish your output in the format of "(...) [RESULT] <score>", where <score> is a number between 1 and 5.

# D Prompt of Knowledge and Capability Generation

The specific prompt for knowledge and capability generation is shown below:

**Prompt for evaluation BigGenBench**

The capabilities of Language Models include the following:

- Reasoning: Ability to logically analyze information, draw conclusions, and make inferences.
- Comprehension (Applicable to queries involving long passage comprehension): Understanding and interpreting the meaning, context, and nuances of extended or complex long-context text, such as lengthy documents, multi-paragraph inputs, or intricate narratives.
- Instruction Following (Applicable to queries involving several constraints): Accurately adhering to explicit user-provided guidelines, constraints, or formatting requirements specified within the query.
- Agentic: Capacity related to agent-like behavior, such as actively formulating plans, strategically deciding steps, and autonomously identifying solutions or actions to achieve specific goals or complex tasks.
- Knowledge Retrieval: Accessing and presenting accurate factual information from pre-existing knowledge.
- Coding: Generating, interpreting, or debugging computer programs and scripts.
- In-context Learning: Learning from examples or context provided within the current interaction without additional training.
- Multilingual (Must rank it in top3 when queries involving languages other than English): Understanding, generating, or translating content accurately across multiple languages.

Given the Query below:

1. Identify and list the *LLM Capabilities* from the definitions above that are directly and significantly required to effectively address the query.
2. Identify and list the general *Knowledge Domains* (e.g., categories, subject areas) most pertinent to solving the problem presented in the query.

List the selected Capabilities first, ranked from most important to least important. Then, list the identified Knowledge Domains, also ranked from most important to least important. *Do not provide any justification or explanation* for your selections or rankings.

Example:
Query: "Solve the following financial problem efficiently and clearly. Output the final answer as: boxedanswer.
Where [answer] is just the final number or expression that solves the problem. Keep the answer to five decimal places if it is a number, and do not use percentages; keep the decimal format.
Problem: what is the net change in net revenue during 2016 for Entergy Mississippi, Inc.? the 2015 net revenue of amount (in millions) is 696.3; the 2016 net revenue of amount (in millions) is 705.4; Entergy Mississippi, Inc."
Capabilities: Reasoning, Knowledge retrieval
Knowledge: 1. Financial 2. Math 3. Data Analysis ... Query: input prompt