Dynamic Model Routing and Cascading for Efficient LLM Inference: A Survey

Anonymous authors
Paper under double-blind review

Abstract

The rapid growth of large language models (LLMs) with diverse capabilities, costs, and domains has created a critical need for intelligent model selection at inference time. While smaller models suffice for routine queries, complex tasks demand more capable models. However, static model deployment does not account for the complexity and domain of incoming queries, leading to suboptimal performance and increased costs. Dynamic routing systems that adaptively select models based on query characteristics have emerged as a solution to this challenge.

We provide a systematic analysis of state-of-the-art multi-LLM routing and cascading approaches. In contrast to mixture-of-experts architectures, which route within a single model, we study routing across multiple independently trained LLMs. We cover diverse routing paradigms, including query difficulty, human preferences, clustering, uncertainty quantification, reinforcement learning, multimodality, and cascading. For each paradigm, we analyze representative methods and examine key trade-offs. Beyond taxonomy, we introduce a conceptual framework that characterizes routing systems along three dimensions: when decisions are made, what information is used, and how they are computed. This perspective highlights that practical systems are often compositional, integrating multiple paradigms under operational constraints.

Our analysis demonstrates that effective multi-LLM routing requires balancing competing objectives. Choosing the optimal routing strategy depends on deployment and computational constraints. Well-designed routing systems can outperform even the most powerful individual models by strategically leveraging specialized capabilities across models while maximizing efficiency gains. Meanwhile, open challenges remain in developing routing mechanisms that generalize across diverse architectures, modalities, and applications.

1 Introduction

1.1 Problem and Motivation

Production deployment of large language models faces a fundamental cost-performance dilemma. Queries vary substantially in complexity, from simple factual questions to complex multi-step reasoning problems. When a single model handles all requests, simple queries consume unnecessary resources when routed to powerful models, while complex queries may exceed the capabilities of smaller models.

Recent advances in multi-LLM deployment have introduced dynamic routing systems that address this challenge. These systems analyze each query and select from a pool of models with different capabilities, costs, and specializations. By matching computational resources to query requirements, adaptive routing can reduce costs while maintaining or improving output quality.

1.2 Routing and Cascading

This survey covers two complementary approaches to adaptive model selection at inference time:

Model routing analyzes each input and selects the most appropriate model based on query characteristics. The router makes a single decision, mapping the query to one model from the available pool.

Model cascading operates sequentially, first attempting inference with smaller, faster models and escalating to larger, more capable models only when the initial response is deemed insufficient based on quality estimation.

Both approaches aim to optimize the performance-cost trade-off by matching queries to appropriate models. Production systems often combine routing and cascading strategies to maximize efficiency.

1.3 Scope and Organization

This survey focuses on routing between independently trained LLMs at inference time, excluding mixture-of-experts architectures that route within a single model. We organize methods into six main paradigms based on their routing strategy, while acknowledging that methods may span multiple categories:

- **Difficulty-aware routing** (Section 2): Routes based on estimated query complexity
- **Human preference-aligned routing** (Section 3): Leverages preference data from human feedback
- **Clustering-based routing** (Section 4): Groups similar queries using unsupervised learning
- **Reinforcement learning routing** (Section 5): Learns routing decisions through reinforcement learning, including policy optimization (e.g. PPO, GRPO) and bandit methods
- **Uncertainty-based routing** (Section 6): Routes based on model confidence estimates
- **Cascading** (Section 7): Sequential multi-model approaches

Additionally, we briefly explore research on routing of multimodal models (Section 8), and cover evaluation approaches, including benchmarks and metrics (Section 9). We then synthesize the surveyed methods through a multidimensional lens in Section 10. Finally, we discuss open challenges and future directions (Section 11), providing a foundation for advancing research in efficient multi-LLM deployment.

1.4 Conceptual Design Space for LLM Routing

The paradigms we cover in this survey (cf. Section 1.3) provide a useful foundation for organizing and understanding the literature. In practice, real-world systems draw on more than one of these paradigms simultaneously. To complement the paradigm-based organization, routing approaches can be categorized through a multidimensional conceptual framework.

When the routing decision is made. Routing systems can rely on either pre-generation or post-generation decisions, or they can adopt a multi-stage process. Pre-generation routing selects a model before generating any output, relying entirely on properties of the incoming query, while post-generation routing systems make their decision after an initial response has been produced, using output quality or confidence as the primary signal. Instead of making a one-time decision, some approaches embed routing within a sequential multi-stage process, escalating across models based on response quality. Multi-stage systems are not mutually exclusive with pre- or post-generation routing; they typically combine both within a sequential escalation structure.

What information the routing mechanism uses. Routing systems differ substantially in the richness of their input signals. The simplest approaches operate on the query alone, using lexical or semantic features to characterize the request. More informed systems additionally incorporate metadata about available models to guide selection, such as cost, latency, or domain specialization. Post-generation approaches further extend this by incorporating response-level signals, such as confidence scores, token probabilities, or verifier outputs. Some systems also accumulate external feedback over time, adapting their routing behaviour based on user interactions or downstream task performance.

How the decision is computed. Routing decisions vary considerably in their computational complexity. At one end, simple threshold rules or cost-based heuristics require no training and can be applied directly at inference time. At the other end, supervised classifiers trained on historical performance data learn to predict which model is likely to handle a given query best. More sophisticated approaches either employ bandit algorithms that update routing behaviour through ongoing interaction at deployment time, or reinforcement learning methods such as PPO and GRPO that optimize a routing policy during training. In practice, many systems combine these mechanisms, for example using a classifier to make an initial routing decision and a threshold rule to trigger escalation within a cascade.

These dimensions are not independent of the paradigms. Table 1 illustrates this by mapping representative methods from each of the six paradigms to these dimensions. As Table 1 illustrates, clustering and difficulty-aware methods are generally pre-generation approaches that operate on query-level signals, while uncertainty-based methods and cascades typically involve post-generation decisions on response-level signals. Interestingly, cascading approaches tend to employ layers of these individual techniques for diverse objectives, ranging from balancing performance-cost trade-offs to implementing safety measures on input queries and output generations. This multidimensional view helps reveal where systems genuinely overlap, as real-world deployments rarely conform to a single paradigm. Production systems typically combine multiple routing strategies, adapting their behaviour to the demands of diverse queries, operational constraints, and evolving user needs. This framework therefore offers a unifying lens that complements the paradigm-based organization of the sections that follow.

2 Difficulty-aware Routing

When an LLM is able to answer a query correctly, it is often because the query is relatively simple or closely aligned with the model's training data. On the contrary, more complex queries such as maths or sophisticated coding tasks may require advanced reasoning capabilities. Driven by this observation that some queries are inherently more "difficult" than others, researchers have explored the idea of routing queries to different LLMs based on their difficulty level.

Query difficulty can be assessed using various methods, including heuristic-based approaches (e.g. text length, word rarity, idiomatic language, syntactic complexity), learned classifiers, or even leveraging the LLMs themselves to estimate the complexity of a query, usually referred to as "LLM as a judge" (Proietti et al., 2025). Once the difficulty of a query is estimated, it can be routed to an appropriate LLM. For instance, simpler queries can be directed to smaller, more cost-effective models, while complex queries can be sent to larger, more capable models. This optimizes resource allocation while maintaining performance. In this sense, several routing methods can fall under this dynamic model selection paradigm, tailored to query difficulty. Implementations can prioritize different objectives, such as cost, latency, accuracy, or human preference.

Method	When				What signals				How computed			
	Pre-gen.	Post-gen.	Multi-stage	Online	Query	Model meta	Response	Feedback	Heuristic	Supervised	Bandit	RL policy
Difficulty-aware												
BEST-Route	■				■	■			■	■		
vLLM Router	■				■					■		
RouteLMT	■				■					■		
EmbedLLM	■				■	■				■		
ICL-Router	■				■	■				■		
GraphRouter	■				■	■				■		
IRT-Router	■				■	■				■		
Preference-aligned												
RouteLLM	■				■			■		■		
Hybrid LLM	■				■			■		■		
Arch-Router	■				■					■		
P2L	■				■			■		■		
Eagle	■							■	■			
Zooter	■				■					■		
Clustering-based												
UniRoute	■				■	■			■	■		
Avengers-Pro	■				■	■			■	■		
Reinforcement Learning												
Router-R1	■	■	■		■	■				■		■
R2-Reasoner	■				■	■			■	■		■
SCOPE	■				■	■			■	■		■
MetaLLM	■			■	■			■			■	
MixLLM	■			■	■	■		■			■	
PILOT	■			■	■	■		■			■	
GreenServ	■			■	■			■	■	■	■	
Dueling Feedback	■			■	■	■		■		■	■	
TI-UCB	■			■				■			■	
Uncertainty-based												
CP-Router		■					■		■			
Probe-based UQ		■					■			■		
Cascading												
FrugalGPT	■	■	■		■	■	■		■	■		
Cascade Routing	■	■	■		■	■	■		■	■		
AutoMix		■	■				■		■			
Self-REF		■	■				■		■	■		
LM-Blender		■	■				■			■		

Table 1: Design-space matrix mapping representative routing and cascading methods to the survey's conceptual framework along three axes: *when* routing decisions are made (Pre-gen., Post-gen., Multi-stage, Online/adaptive), *what* signals are used (Query, Model metadata, Response, Feedback), and *how* decisions are computed (Heuristic, Supervised, Bandit, RL policy). Shaded cells indicate the property applies; cell and row colour identify the paradigm group. Multi-stage denotes a sequential escalation structure and is not mutually exclusive with Pre-gen. and Post-gen., which describe the types of decisions made within that structure. Multiple filled cells in the *how* group indicate that a method combines mechanisms, e.g. a learned classifier paired with a threshold rule.

Among the work that can be classified under the difficulty-aware routing paradigm is BEST-Route (Ding et al., 2025) which dynamically allocates queries to different LLMs depending on their assessed difficulty. BEST-Route uses a multi-head router based on DeBERTa-v3-small (He et al., 2021; 2023) to estimate query difficulty and select the optimal model and sampling strategy. At inference time, it employs best-of-n sampling (Stiennon et al., 2020; Nakano et al., 2021) to enhance the performance of small models, selecting the best response out of n generated responses, using a proxy reward model fine-tuned from OpenAssistant RM, which is based on a DeBERTa-v3-large. If necessary, sampling is increased, or harder queries are routed to a larger model. At inference time, users can set thresholds to balance cost and accuracy, where higher thresholds favour the reference model, improving quality at increased costs. Since multiple small-model and best-of-n combinations can meet a given threshold, BEST-Route effectively selects among them using cost estimation to ensure high-quality responses at minimal cost. In the area of machine translation, Wu et al. (2025b) apply a similar intuition, proposing source-side routing policies that decide whether to use an NMT system or an LLM based on the complexity of the source sentence.

Similarly, vLLM Semantic Router (Wang et al., 2025a) addresses the challenge of adaptive reasoning, where costly step-by-step inference is applied only when beneficial, while maintaining low latency and efficiency for straightforward queries. The system uses a ModernBERT-based classifier (Warner et al., 2025) to analyze query intent and complexity, routing queries that require reasoning to models with chain-of-thought capabilities while directing simpler queries to standard inference without reasoning. This reasoning-aware approach demonstrates that selective application of computational resources based on query complexity can simultaneously improve both accuracy and efficiency.

Recently, RouteLMT (Luo et al., 2026) has proposed an in-model router that probes the small translator's prompt-token representations via lightweight LoRA adaptation, without requiring external models or hypothesis decoding. It identifies the marginal gain, i.e., the large model's improvement over the small model, as the optimal signal for routing decisions.

In a relevant line of work, EmbedLLM (Zhuang et al., 2024) employs matrix factorization to learn compact vector embeddings of LLMs, capturing model characteristics such as domain specialization. At routing time, these embeddings are used to predict which model will answer a given query correctly, effectively estimating query-model compatibility without requiring explicit difficulty labels. The approach enables efficient routing and benchmark accuracy prediction, though it requires retraining when incorporating new models.

ICL-Router (Wang et al., 2026) represents model capabilities as compact in-context vectors, derived from a small set of query-performance pairs that reflect each model's capability profile. An LLM-based router is trained in two stages, first to align query embeddings with its semantic space, then to predict whether each model can correctly answer a given query using these vectors as in-context input. A key advantage is that new models can be integrated without retraining the router, by simply evaluating them on a representative query set to generate their capability profile.

GraphRouter (Feng et al., 2024) takes a different approach by modelling the relationships among tasks, queries, and LLMs through a heterogeneous graph structure. The method constructs a graph with task, query, and LLM nodes, where edges represent interactions between these entities. A Graph Neural Network (GNN) trained on historical performance and cost data performs edge prediction to forecast both the effectiveness and expense of routing a query to a specific LLM. This inductive learning framework enables GraphRouter to generalize to newly introduced LLMs without retraining, as it learns patterns from node characteristics rather than memorizing specific models. Experiments demonstrate substantial performance improvements while reducing computational overhead.

IRT-Router (Song et al., 2025) explicitly models two complementary properties for routing: the ability of each LLM, learned from its profile metadata, and the difficulty and discrimination of each query, extracted from its embedding. These are combined in an interactive layer to predict the probability that each candidate LLM will correctly answer the query. The predicted performance is then combined with each LLM's cost to produce a routing score, and the query is directed to the highest-scoring model. This formulation, inspired by Item Response Theory from educational psychometrics, provides interpretable routing decisions and handles unseen queries at deployment time through a semantic similarity warm-up mechanism.

While our survey focuses on multi-LLM routing, it is worth mentioning that difficulty estimation has also been explored in single-LLM contexts. Recent research found that LLMs frequently "overthink" simple queries and "underthink" complex ones (Aggarwal et al., 2026), leading to inefficiencies. Moreover, sophisticated chain-of-thought reasoning primarily improves performance on complex maths and logic tasks, with limited gains for other tasks (Wei et al., 2022; Sprague et al., 2025). On one hand, difficulty estimation can be calculated before sending the query to a reasoning LLM to decide whether to activate the thinking mode[1] (Zhang et al., 2025b;d), or to constrain the thinking token budget (Shen et al., 2025). On the other hand, probing the LLM during the reasoning process can help identify whether to continue with further reasoning steps or to stop early (Wu et al., 2025a; Jiang et al., 2025). This line of work can fall under adaptive computation approaches that aim to optimize resource usage within a single LLM.

3 Human Preference-aligned Routing

In this paradigm, human preference data is used to train routers that generalize across domains and LLMs. For example, RouteLLM (Ong et al., 2025) introduces a learning framework for training router models that dynamically select between strong and weak LLMs at inference time, optimizing the trade-off between response quality and cost. It formulates routing as a binary decision between strong (high-quality, high-cost) and weak (lower-quality, low-cost) LLMs, and employs a win prediction model to estimate the probability that the strong model will outperform the weak model for a given query. RouteLLM optimizes router parameters by maximizing the likelihood of observed preference data. Unlike Hybrid-LLM (Ding et al., 2024) and Arch-Router (Tran et al., 2025) that use solely synthetic preference data generated by an LLM, RouteLLM (Ong et al., 2025) leverages human preference labels from Chatbot Arena (Chiang et al., 2024). In addition, RouteLLM authors obtain pairwise comparison labels using an LLM judge. This data augmentation technique with synthetic preference labels substantially improves router performance. They experiment with four router architectures, namely similarity-weighted SW Ranking, matrix factorization, BERT, and causal LLM (based on Llama 3 8B). The matrix factorization router and causal LLM routers perform very competitively on MT Bench (Zheng et al., 2023) when trained on the augmented dataset of both Chatbot Arena and LLM judge, with matrix factorization being the most efficient and cost-effective router.

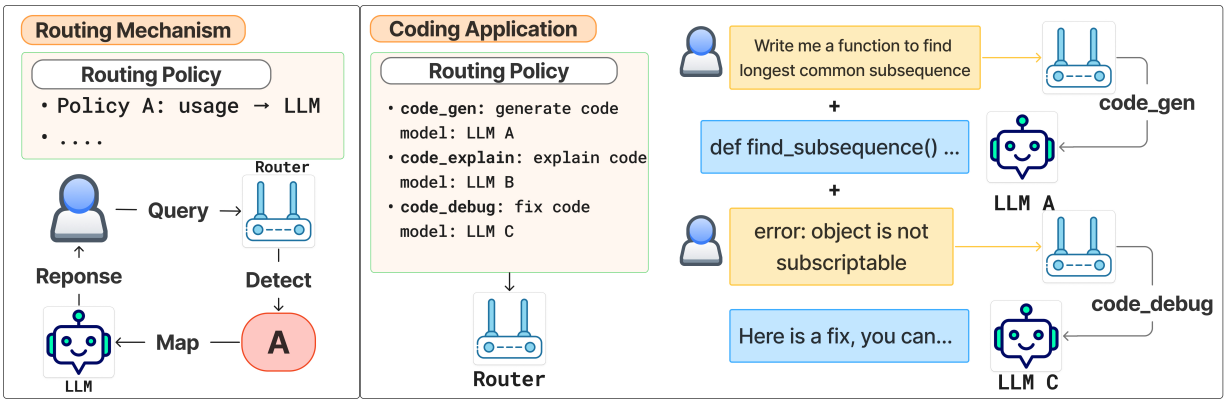

Figure 1: Arch-Router's Preference-Aligned Routing Mechanism. The routing policies and user conversation are provided to the router to select the appropriate policy and corresponding LLM. Example usage in coding is shown on the right.

Similarly, Arch-Router (Tran et al., 2025) aligns routing decisions with explicit user preferences. However, rather than optimizing purely for cost or performance, it allows users to define domain-action pairs. For

[1]Some approaches such as AdaptThink (Zhang et al., 2025b) decide whether to activate the thinking mode from the beginning, while other approaches such as Continue-Thinking (Zhang et al., 2025d) first estimate the quality of the generated response without thinking to decide whether to continue generation with thinking.

instance, legal summarization queries can be routed to one model, while code generation requests will be sent to another model. The key insight is that routing policies are provided as part of the input context to a 1.5B parameter model, which means users can update their preferences without retraining. This is particularly valuable in production environments where routing requirements evolve frequently. Nevertheless, this flexibility comes with trade-offs, as a 1.5B model introduces additional computational overhead compared to lightweight classification-based routers, which may be suboptimal in latency-sensitive applications. Figure 1 illustrates the preference-aligned routing mechanism that Arch-Router has introduced.

Prompt-to-Leaderboard (P2L) (Frick et al., 2025) extends traditional preference-based leaderboard evaluation by training an LLM to generate prompt-specific Bradley-Terry coefficients for model ranking. Unlike standard leaderboards that average performance across all tasks, P2L takes a user's prompt as input and outputs customized strength scores for each model, predicting which LLM will perform best for that particular query. The method enables task-specific evaluation, optimal cost-constrained routing, and personalized model selection based on user history.

Eagle (Zhao et al., 2024) introduces a training-free approach to preference-aligned routing by using ELO ranking systems (Elo, 1967). The method combines two complementary modules: Eagle-Global, which evaluates model overall capability using ELO ratings across the entire historical dataset, and Eagle-Local, which assesses specialized abilities through ELO rankings on similar past queries. By transforming sparse pairwise comparisons into comprehensive rankings, Eagle efficiently integrates user feedback without requiring model training.

While the aforementioned methods directly optimize against human or synthetic preference comparisons, some approaches rely on reward models trained on such data to provide supervisory signals. Zooter (Lu et al., 2024) takes a reward-guided supervised learning approach, training a routing classifier using labels derived from the QwenRM reward model (Bai et al., 2023), which scores candidate LLM responses on training queries.[2] The routing function outputs a categorical distribution over a pool of LLMs, representing the likelihood that each model can best handle the query. This knowledge distillation approach employs a BERT-style router (mDeBERTa-v3-base (He et al., 2023)). However, since training relies on a fixed set of LLMs, Zooter's adaptability to new or unseen models is limited.

4 Clustering-based Routing

Clustering-based routing groups similar queries using unsupervised learning and assigns each cluster to the most suitable LLM. This approach balances performance and cost by routing different query types to their most cost-effective models, without requiring explicit task labels.

As illustrated in Figure 2, UniRoute (Jitkrittum et al., 2026) applies K-means clustering to an unlabelled training dataset to identify K centroids, then partitions a validation set into these representative clusters. Each candidate LLM is evaluated on the validation data within each cluster, with scores adjusted by a predefined cost value for that model. At inference time, incoming queries are compared against the cluster centroids to determine which LLM should handle the request, with the accuracy-cost trade-off controlled through dynamic cost adjustment during routing.

Among the main practical advantages of this clustering approach is the ability to add new LLMs at inference time without retraining, as they are simply evaluated on existing clusters to obtain their performance profile. Moreover, the method can operate entirely without task labels, relying solely on query embeddings and cluster assignments. UniRoute demonstrates effective scaling, handling routing decisions across unseen LLMs.

Avengers-Pro (Zhang et al., 2025e) employs a similar clustering strategy while emphasizing cost-performance optimization. The method demonstrates that strategic routing can establish a Pareto frontier that surpasses strong individual models such as GPT-5-medium, achieving better accuracy at a lower cost through effective cluster-based model allocation.

[2] QwenRM is typically used for RL training (e.g., GRPO) or SFT data construction through rejection sampling. Zooter uses it solely to generate routing labels.

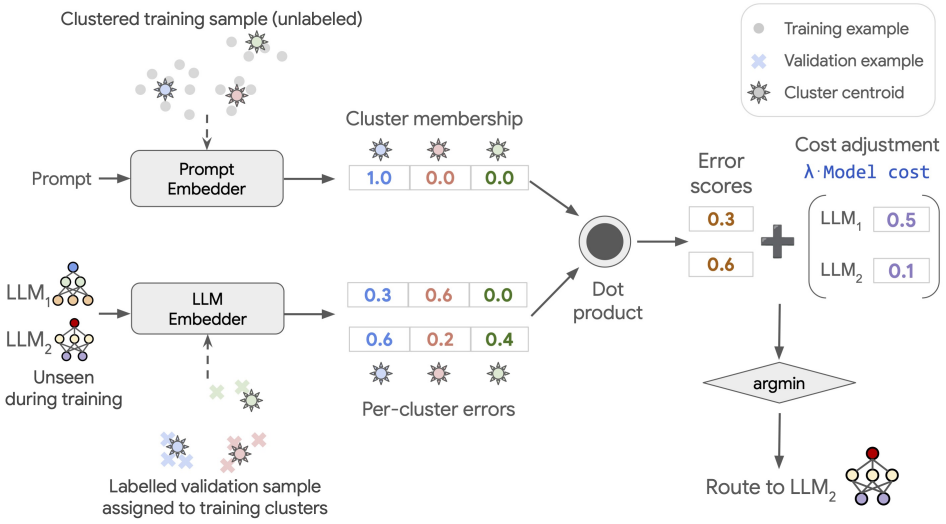

Figure 2: UniRoute cluster-based router. First, perform K-means on a training set to find K centroids, and then partition the validation set into K representative clusters. Each test-time LLM can then be represented as a K-dimensional feature vector of per-cluster errors. This yields an intuitive routing rule: for each test prompt, route to the LLM with the smallest cost-adjusted average error on the cluster the prompt belongs to. The prompt embedder may either be completely unsupervised (as shown in the figure), or fitted via supervised learning using labels from a set of training LLMs different from those seen during test time.

5 Reinforcement Learning Routing

Reinforcement learning (RL) approaches formulate routing as decision-making under uncertainty. Methods in this section include policy optimization algorithms that iteratively refine routing decisions through multi-step interactions and online bandit algorithms that balance exploration and exploitation with real-time feedback.

5.1 Policy Optimization Methods

Policy optimization approaches such as Router-R1 (Zhang et al., 2025a) and Route-and-Reason (R2-Reasoner) (Shao et al., 2025) optimize routing decisions through multi-step interactions. These RL approaches leverage the strengths of multiple LLMs to solve complex reasoning tasks while optimizing performance-cost trade-offs. While Router-R1 addresses multi-round routing and aggregation, R2-Reasoner focuses on subtask-level decomposition.

Router-R1 formulates routing as a sequential decision-making process. It alternates between performing internal reasoning ("think" action) and assigning a specific model from a pool of available LLMs ("route" action), iteratively refining answers. The router itself is an LLM that can leverage its reasoning ability to dynamically select and aggregate responses from multiple models. Router-R1 initiates a maximum of 4 routing steps per input query. The authors use Qwen2.5-3B-Instruct (Yang et al., 2024) and LLaMA-3.2-3B-Instruct (Dubey et al., 2024) as base models. They are trained with the Proximal Policy Optimization (PPO) (Schulman et al., 2017) reinforcement learning algorithm, on a training dataset of question–score pairs representing the quality of responses of each LLM. The authors argue that Router-R1 can generalize to new LLMs without retraining by conditioning on simple descriptors such as pricing and latency.

Similarly, R2-Reasoner decomposes complex reasoning tasks into simpler subtasks using a dedicated Task Decomposer. These subtasks are then allocated to the most suitable model by a Subtask Allocator. R2-Reasoner employs a staged training pipeline, combining supervised fine-tuning and reinforcement learning with Group Relative Policy Optimization (GRPO) (Shao et al., 2024). This iterative approach refines

the decision-making capabilities of the decomposer and allocator without requiring end-to-end gradient propagation. The authors argue that the framework achieves significant cost reductions (84.46% API cost savings) while maintaining competitive reasoning accuracy. It also generalizes well to unseen models, making it adaptable to dynamic real-world scenarios.

Although both Router-R1 and R2-Reasoner aim to optimize cost, reasoning steps can incur multiple model calls, introducing inference latency. In this sense, these approaches are best suited for scenarios where cost savings outweigh latency concerns, such as routing to commercial LLM APIs with high per-call costs.

SCOPE (Scalable and Controllable Outcome Performance Estimator) (Cao et al., 2026) takes a distinct approach by using GRPO to train a performance estimator rather than a routing policy. Given a query and a candidate model, SCOPE retrieves behavioural examples from a pre-computed model fingerprint and predicts both the model's correctness and token cost via chain-of-thought reasoning. The final routing decision is then made by a simple rule-based selection using a user-defined accuracy–cost trade-off, requiring no further learning. Unlike Router-R1 and R2-Reasoner, which use RL to directly optimize routing decisions, SCOPE's RL component is closer to reward-shaped prediction training than policy optimization. Furthermore, by characterizing models through behavioural fingerprints rather than fixed identifiers, SCOPE generalizes to unseen models without retraining.

5.2 Bandit-Based Methods

Bandit-based methods frame routing as an online learning problem, balancing exploration of model capabilities with exploitation of known strengths while adapting to feedback over time.

MetaLLM (Nguyen et al., 2025) formulates routing as a multi-armed bandit problem, dynamically selecting the least expensive LLM likely to provide a correct answer for each query. MetaLLM does not rely on reward models for training. Instead, it optimizes routing decisions based on an accuracy-cost trade-off, adapting to query difficulty and model performance over time.

MixLLM (Wang et al., 2025b) employs a contextual bandit framework with policy gradient methods to optimize query-LLM assignments. The system enhances query embeddings with domain-aware tags through unsupervised fine-tuning, enabling more accurate predictions of LLM-specific response quality and cost. A meta decision-maker selects assignments to balance quality, cost, and latency constraints. During online deployment, MixLLM updates its routing policy using binary user feedback through a dynamic feedback score mechanism, applying policy gradient optimization to maximize expected reward. This continual learning approach allows the system to adapt to evolving queries and changing LLM candidate sets. The framework demonstrates strong practical performance, achieving 97.25% of GPT-4's quality at only 24.18% of the cost under time constraints.

PILOT (Preference-prior Informed LinUCB for Adaptive Routing) (Panda et al., 2025) extends contextual bandits by integrating offline human preference data with online evaluative feedback. The method reformulates routing as a contextual bandit problem where the system selects LLMs based on binary success/failure signals rather than full supervision. Building on the LinUCB algorithm (Li et al., 2010) widely used in recommender systems, PILOT incorporates preference priors from sources like Chatbot Arena to accelerate learning. The authors additionally model cost constraints as an online multi-choice knapsack problem (Chakrabarty et al., 2008) to ensure resource-efficient routing under budget limits. This combination enables dynamic, personalized LLM selection that balances performance and cost.

GreenServ (Ziller et al., 2026) extends the contextual bandit framework with an explicit focus on energy efficiency. Rather than optimizing for cost proxies such as API prices or token budgets, GreenServ measures GPU energy consumption directly during inference. It extracts a lightweight context vector per query, encoding task type, semantic cluster, and textual complexity, and then employs LinUCB (Li et al., 2010) to learn routing policies online across a pool of 16 open-access LLMs. A tunable trade-off parameter λ allows operators to interpolate between accuracy-first and energy-first policies. Compared to random routing, GreenServ achieves a 22% accuracy gain while reducing cumulative energy consumption by 31%, with a routing overhead of under 8 ms per query.

LLM Routing with Duelling Feedback (Chiang et al., 2025) addresses routing through a contextual duelling bandit formulation, learning from pairwise preference comparisons rather than absolute performance scores (cf. Section 3). The method introduces Category-Calibrated Fine-Tuning (CCFT), which constructs model embeddings by first fine-tuning text encoders with contrastive learning to cluster queries by category, then computing weighted combinations of category embeddings based on each LLM's domain expertise and cost profile. These learned embeddings enable the deployment of Feel-Good Thompson Sampling for Contextual Duelling Bandits (FGTS.CDB), a theoretically grounded Bayesian algorithm for online model selection. Evaluations on RouterBench and MixInstruct demonstrate that the approach achieves lower cumulative regret and faster convergence than baselines using general-purpose embeddings, while maintaining robust performance-cost balance.

Similarly, Time-Increasing UCB (TI-UCB) (Xia et al., 2024) addresses online model selection through a time-increasing bandit algorithm that accounts for the increasing-then-converging performance trend observed during iterative model fine-tuning, a common pattern as LLMs are progressively improved. Unlike traditional bandit methods that assume stationary reward distributions, TI-UCB incorporates a change detection mechanism to identify convergence points, enabling more efficient exploration-exploitation balance. The approach achieves logarithmic regret bounds and demonstrates practical effectiveness in both classification model selection and online LLM selection scenarios.

6 Uncertainty-based Routing

Researchers have been investigating whether there is a reliable way to estimate if a model is confident about its own response, where such confidence must be aligned with actual correctness. This notion has several applications in routing, as effective uncertainty quantification enables systems to identify when to escalate queries to more capable models.

6.1 Probing and Probability-based Uncertainty Estimation

In their work, Chuang et al. (2025b) benchmark eight uncertainty quantification methods for routing from SLMs to LLMs on edge devices. They observe that probe-based methods (trained classifiers) and perplexity-based methods significantly outperform verbalization approaches (self-reported confidence). They demonstrate that SLMs match LLM performance on high-confidence (top 20%) queries. These results are in line with the experiments by Mahaut et al. (2024) who demonstrate that trained hidden-state probes provide the most reliable confidence estimates, even though at the expense of requiring access to weights and supervision data.

CP-Router (Su et al., 2025) applies Conformal Prediction to route between standard LLMs and Large Reasoning Models (LRMs) like DeepSeek-R1, which often generate verbose reasoning for simple queries. For multi-choice question answering, the method extracts logits for answer options, applies softmax to obtain a probability distribution, and computes uncertainty scores as 1 minus the probability for each option. It then builds a prediction set based on these scores against a calibration-derived threshold. Queries with a single plausible option (high confidence) are handled by the standard LLM, while those with multiple plausible options (high uncertainty) are routed to the more capable but costly LRM.

6.2 LLM-as-a-Judge Uncertainty Estimation

Instead of confidence calculations, some researchers explored using LLMs themselves to estimate the quality of either their own responses or those generated by other models.

Zhang et al. (2025c) propose a confidence-driven LLM router that leverages uncertainty estimation derived from response quality evaluation. Rather than estimating confidence from model-internal signals, the method employs an external LLM-as-a-Judge to simulate human rating preferences, jointly optimizing response quality and system cost for edge–cloud routing.

In contrast, other researchers employ self-verification or self-reporting, where an LLM is asked to report its own confidence. Chuang et al. (2025b) refer to these approaches as "verbalization" and classify them as

one-step verbalization, where an LLM is prompted to output both the answer and numeric confidence in one step, and two-step verbalization, where the confidence score is obtained in a separate, follow-up query after the model has provided an answer (Tian et al., 2023). However, their experiments indicate that both verbalization techniques consistently exhibit low alignment between reported uncertainty and prediction correctness.

Despite these limitations, recent work demonstrates that self-verification can be effective in cascading systems when combined with other mechanisms. AutoMix (Aggarwal et al., 2024) uses few-shot in-context learning for self-verification, while Self-REF (Chuang et al., 2025a) employs lightweight fine-tuning to improve confidence calibration. These methods are discussed in detail in Section 7 on cascading approaches, as they rely on self-verification as part of multi-stage routing pipelines rather than as standalone uncertainty quantification.

7 Cascades

While routing preselects an appropriate LLM for a query, cascading sequentially queries a pool of LLMs until a reliable response is obtained. Cascading strengthens the routing process by allowing multiple LLMs to be involved in answering a single query. In a typical cascading setup, a smaller and cheaper LLM is first queried to generate an initial response. Based on the quality of this response, the system decides whether to accept it or escalate the query to a larger and more capable LLM for further refinement or regeneration. This cascading system design can be particularly useful when dealing with queries of varying difficulty levels, as it allows the system to adaptively allocate resources based on the complexity of the task at hand.

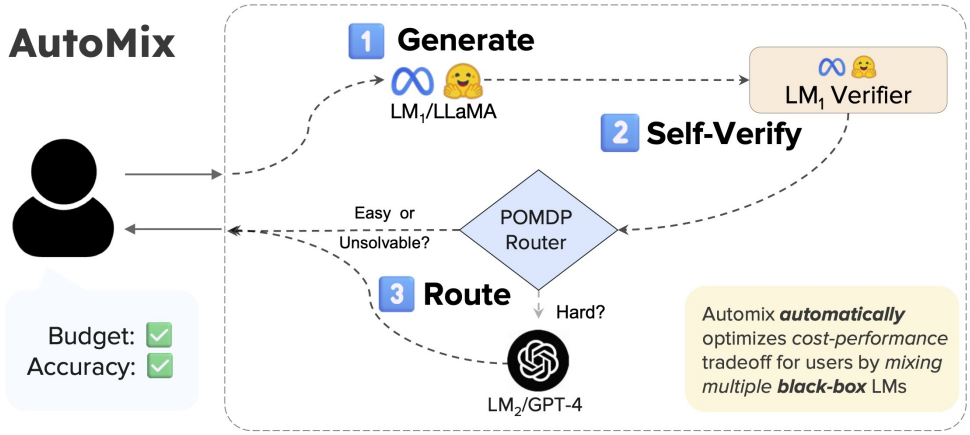

Figure 3: AutoMix example for two-model setup. Instead of relying only on small model (SLM) with low performance or a large model (LLM) with high cost, AutoMix automatically mixes multiple black-box language models, based on user desired cost-quality tradeoff. AutoMix works in a 3-step process: (i) generation by a small model (LM1), (ii) self-verification of the generated answer, (iii) using confidence assessments from self-verification to do appropriate routing to a larger model (LM2). For N-model setup, the process is repeated till the final answer is reported.

FrugalGPT (Chen et al., 2024) mixes routing and cascading through employing three main components: an LLM router, a threshold-based quality estimator, and a stop judge. Given a user query q, the router first selects an LLM to obtain its response to the query. However, in their setup, they adopted query-agnostic permutation for simplicity. Next, the quality estimator based on DistilBERT (Sanh et al., 2019) takes the query, the response, and the selected LLM as input and generates a quality score as output. Based on the quality estimation and the invoked LLM service, a cost-aware stop judge determines whether (i) to stop and return the answer, or (ii) to repeat the process of invoking the LLM router and generation scorer. Similarly, Dekoninck et al. (2025) propose cascade routing, a unified framework that integrates routing and cascading into a single system. Unlike pure routing, which selects a single model per query, or cascading,

which runs models sequentially from smallest to largest, cascade routing iteratively selects the best model at each step, allowing it to skip or reorder models dynamically. The authors identify quality estimation as the critical factor for model selection success and show that cascade routing consistently outperforms individual approaches.

Instead of relying on an external model for quality estimation, some researchers have proposed "self-verification" as an efficient approach for model cascading (Aggarwal et al., 2024; Chuang et al., 2025a). In this paradigm, an efficient LLM learns to estimate the quality of its own response. The self-verification score is then used to decide how to proceed. For instance, AutoMix (Aggarwal et al., 2024) uses a smaller model to generate an initial answer and self-verify its response before potentially routing the query to a larger model. The approach relies on few-shot prompting without fine-tuning, making it suitable for black-box models. While previous work finds self-verification unreliable to repair the model's output (Huang et al., 2024) or estimate its confidence (Mahaut et al., 2024; Chuang et al., 2025b), AutoMix demonstrates that few-shot self-verification provides a valuable signal that can be used for routing to an appropriate model. The final step involves using a router to decide whether to escalate the query to a larger model if it is a hard query, or output the current answer to the user if it is easy or unsolvable.[3] The router is modelled as a Partially Observable Markov Decision Process (POMDP) (Åström, 1965; Wu et al., 2021), which relies only on self-verification confidence as input. Figure 3 illustrates the AutoMix approach.

While AutoMix relies solely on few-shot in-context learning without any fine-tuning, Self-REF (Chuang et al., 2025a) introduces a lightweight fine-tuning strategy that enables LLMs to express confidence in their predictions using special confidence tokens. This confidence signal is then used for downstream routing, where queries with low confidence are routed to stronger (but more costly) LLMs, or rejection where the system should fall back on a safe default behaviour. Self-REF augments training data with confidence tokens based on correctness, fine-tunes the LLM to generate these tokens, and extracts a confidence score for each prediction.[4] The approach outperforms baselines in both routing and rejection learning tasks, providing a practical mechanism for confidence-aware LLM cascades.

Beyond single-model confidence estimation, ensemble approaches take a complementary direction by querying multiple models simultaneously. LM-Blender (Jiang et al., 2023) uses an ensemble framework that queries multiple LLMs during inference, selects the best responses, and then merges them into one response. The framework consists of two main modules: Pair Ranker, which uses pairwise comparisons to rank candidate outputs from multiple LLMs, and Gen Fuser, which fuses the top-ranked outputs to generate an improved response. First, the Pair Ranker employs a cross-attention encoder to jointly encode the input and candidate pairs, achieving high correlation with human and ChatGPT-based rankings. Then, the Gen Fuser further enhances output quality by synthesizing the best aspects of top candidates. The approach is evaluated on the MixInstruct benchmark, demonstrating significant improvements over individual LLMs and baseline ensembling methods.

While several cascading approaches restart generation from scratch when a query is escalated to a more capable model, others simply refine the response already generated by the smaller model. In the area of machine translation (MT), cascaded system designs have been extensively studied and deployed in production. Work on automatic post-editing (APE) formulates translation as a two-stage process, in which a base MT system produces an initial hypothesis that is selectively refined by a more capable model when necessary. Usually, quality estimation (QE) models are used to score initial translations, and the routing system decides whether to accept a translation or to escalate to an APE model depending on predefined criteria or thresholds (Specia et al., 2018; Hendy et al., 2023; Moslem et al., 2023a; Zerva et al., 2024). This cascading approach has also been extended to terminology-constrained APE, instructing an LLM to integrate the required terms into the translations (Moslem et al., 2023b). Skipping post-editing for translations that already include pre-approved terms can reduce the overall post-editing load. Such an efficient workflow can save resources, and minimise latency at inference time. Similarly, there are potential advantages of employing an LLM for post-editing rather than for direct translation. Instead of solely relying on the translation quality of the LLM, quality

[3] The idea of blocking "unsolvable" queries from being routed to the larger model as an efficiency booster has been investigated in detail in recent work such as Firewall Routing (Peng et al., 2025).

[4] To obtain a continuous confidence score after training, the authors compute the probability of the confident token *<CN>*, normalized over the sum of the probabilities of the unconfident *<UN>* and confident *<CN>* tokens.

estimation can be performed to select the best MT model in general or for the current source text segment. Ultimately, only segments that do not meet quality criteria are then passed to the LLM for post-editing. Furthermore, QE-based deferral has also been applied directly to routing between small and large translation models, bypassing post-editing entirely. Farinhas et al. (2025) use QE metrics as deferral rules in a cascaded system, invoking the larger model only for a fraction of segments. These cascading approaches demonstrate that quality estimation can effectively guide escalation decisions in hybrid model deployment, reducing large model calls while maintaining quality.

8 Multimodal Model Routing

Research on model routing has been focused on text-based LLMs. Many of these approaches can be extended to other modalities such as speech, image, and video. However, representations of these modalities are different from text, which poses new challenges.

Model-Spider (Zhang et al., 2023) is a model selection and ranking method designed to efficiently choose suitable pre-trained models from a large model pool, including both visual models and LLMs, for a downstream task. Unlike the multi-LLM routing mechanisms we covered so far, Model-Spider does not dynamically dispatch queries to different models at inference time. It mainly evaluates candidate models from the available pool and ranks them according to predicted task fitness.

ReLope (Zeng et al., 2026) extends probe-based routing to multimodal LLMs (MLLMs), where visual inputs introduce new challenges for efficient deployment. The authors highlight the limitation that hidden-state probes effective in text-only LLMs degrade substantially when visual inputs are introduced, as visual tokens weaken the separability of correctness signals in hidden states. To address this, they propose two complementary techniques. The Attention Probe aggregates token-level hidden states from the preceding layer using attention scores to recover distributed correctness signals. ReLope (KL-Regularized LoRA Probe) inserts a lightweight LoRA adapter and applies a KL regularizer to learn routing-aware representations that suppress task-irrelevant visual noise. Experiments across five multimodal benchmarks show that both methods consistently outperform existing probe-based baselines in hybrid MLLM systems.

MMR-Bench (Haoxuan et al., 2026) is a modality-aware evaluation framework. It covers a wide range of vision-language tasks, including OCR, general VQA, and multimodal math reasoning. The benchmark includes strong single-model baselines, oracle upper bounds, and representative routing policies, enabling systematic comparison of routing strategies under fixed candidate sets and cost models.

Despite MMR-Bench's contributions, multimodal routing remains underexplored compared to text-only settings. Key challenges include developing unified representations across modalities, handling queries that require multiple modalities simultaneously, and adapting routing strategies to modality-specific characteristics. Future work should explore how difficulty estimation, uncertainty quantification, and preference alignment can be extended from text-only to multimodal settings.

9 Evaluation

In order to optimize routing efficiency, it is imperative to establish the criteria by which routing systems are evaluated. This section reviews commonly used benchmarks and metrics for assessing routing quality, efficiency, and cost.

9.1 Routing Benchmarks

Several specialized benchmark datasets have been developed to evaluate LLM routing systems systematically.

RouterBench (Hu et al., 2024) has been widely adopted for evaluating routing methods. It provides over 405k precomputed inference outputs from eleven diverse LLMs across seven tasks (MMLU, MT-Bench, MBPP, HellaSwag, WinoGrande, GSM8K, ARC). The dataset includes detailed performance and cost metadata, enabling systematic analysis of routing strategies under realistic constraints.

RouterEval (Huang et al., 2025) is a large-scale open-source benchmark for systematic evaluation of LLM routing methods. The benchmark comprises over 200 million performance records from more than 8,500 LLMs across 12 widely used benchmarks, including ARC, HellaSwag, MMLU, TruthfulQA, GSM8K, and MMLU-PRO. RouterEval frames routing as an m-way classification problem with varying difficulty levels (m = 3, 5, 10, 100, 1000) and supports evaluation across all-strong, all-weak, and mixed model groups. Key findings demonstrate that capable routers can achieve performance surpassing the best single model through model complementarity.

MixInstruct (Jiang et al., 2023) is an instruction-following benchmark designed to evaluate routing and ensemble methods using preference-based supervision. It aggregates 110k examples from multiple instruction datasets and provides oracle pairwise preferences obtained via LLM-based comparisons.

LLMRouterBench (Li et al., 2026) comprises over 400K instances from 21 datasets and 33 models. Moreover, it provides comprehensive metrics for both performance-oriented routing and performance-cost trade-off routing, and integrates 10 routing baselines. The LLMRouterBench framework integrates collector, evaluator, and adaptor components for standardized evaluation of LLM routing methods.

Beyond routing-specific benchmarks, many routing methods are evaluated on standard LLM benchmarks such as MT-Bench (Zheng et al., 2023) for multi-turn conversations, MMLU (Hendrycks et al., 2021) for knowledge-based reasoning, and MATH-500 (Lightman et al., 2024) and GSM8K (Cobbe et al., 2021) for mathematical problem-solving.

9.2 Evaluation Metrics

Evaluating routing systems requires metrics that capture both the quality of routing decisions and their computational efficiency. We organize these metrics into performance, efficiency, and cost metrics.

9.2.1 Performance and Quality Metrics

The fundamental goal of routing is to select models that produce high-quality responses. **Routing accuracy** measures the percentage of queries routed to the optimal model. In addition, a confusion matrix identifies false positives and false negatives, which helps calculate the overall performance of the routing system on the current task.

While routing accuracy measures whether the router has selected the "optimal" model for each query, task performance determines whether the selected model has produced a correct or high-quality response. **Task performance** evaluates response quality using domain-specific metrics such as accuracy on multiple-choice questions, exact match for factual queries, pass@k (Kulal et al., 2019; Chen et al., 2021) for code generation, and chrF (Popović, 2017) or COMET (Rei et al., 2020) for translation. Moreover, LLM-as-a-Judge evaluations are increasingly adopted, where a strong LLM ranks the quality of responses from different models. The overall performance of a routing system is the aggregate task performance across all routed queries, effectively treating the router as a meta-system that combines multiple models.

For preference-based routing methods (cf. Section 3), **win rate** measures how often the routed model's response is preferred over a baseline in pairwise comparisons, whether through human judgements or LLM-as-a-Judge evaluations. **Area Under Curve (AUC)** is used to summarise routing performance across different operating points, such as performance across varying cost budgets or deferral thresholds, providing a single metric for comparing routers independent of specific cost constraints.

9.2.2 Efficiency and Cost Metrics

While routing aims to reduce computational costs, it introduces additional system overhead that must be carefully controlled. Obviously, adding larger models to the LLM routing pool contributes to this overhead. Additionally, the overhead resulting from the routing process itself must be negligible.

Latency refers to the time between sending a request and receiving a response, and it is commonly measured by the time to first token metric (TTFT) during prefilling and the time per output token metric (TPOT) during decoding (Huyen, 2025).

Throughput captures system capacity, and it is measured in tokens per second (TPS) or queries per second (QPS). There is an inherent trade-off between latency and throughput, where serving requests quickly (low latency) often conflicts with serving many requests efficiently (high throughput). **Goodput** measures throughput that meets predefined constraints.[5]

Cost represents API charges, compute resources, and token consumption. Typically, cost can be reported relative to strong baselines (e.g., "97% of GPT-4's quality at 24% of the cost") or as a simple monetary value. To jointly evaluate quality and efficiency, quality-cost trade-offs are visualized through **Pareto frontiers** that plot achievable combinations of performance and cost, where effective routers should dominate any single model.

Beyond direct costs, routing systems can reduce environmental impact by favouring smaller models. **Energy consumption** can be reported per token, per query, or as total inference energy, with lower consumption indicating improved efficiency. **Carbon footprint** measures CO2 emissions, estimated from energy consumption and grid carbon intensity. These environmental metrics are increasingly adopted as environmental sustainability becomes central to efficient large-scale deployment.

10 Towards Multidimensional Routing Systems

As we outlined in Section 1.4, the methods reviewed in this survey indicate that practical routing systems are increasingly multidimensional. Rather than selecting a single paradigm, production deployments tend to compose mechanisms across paradigms to satisfy heterogeneous constraints on quality, latency, cost, and safety. They combine pre-generation and post-generation decisions, integrate query-level and response-level signals, and couple offline learning with online adaptation.

One way to design such systems in production is to treat routing as a control pipeline with three coordinated stages: (i) an initial low-cost pre-router based on query and model metadata subject to cost constraints, (ii) a post-generation verifier that estimates response quality or uncertainty of efficient/weaker models, and (iii) an escalation policy that decides whether to accept, refine, reject, or defer to stronger models. In this sense, cascades are a natural extension in routing systems.

To demonstrate how this compositional logic plays out in practice, we can consider representative approaches we have discussed in this survey. For example, both FrugalGPT and Cascade Routing instantiate all three pipeline stages, where a pre-router selects an initial model based on query and cost signals, a quality estimator evaluates the response, and an escalation policy decides whether to accept or defer to a stronger model. Different compositional patterns emerge in bandit-based systems such as MixLLM and PILOT, which couple a pre-generation routing decision with online feedback to continuously refine the policy, effectively collapsing the verifier and escalation stages into an implicit reward signal. Uncertainty-based methods such as CP-Router and Self-REF occupy a middle ground, using post-generation confidence estimates as the escalation trigger without requiring explicit quality estimation models. As illustrated by our design-space matrix (cf. Table 1), these examples suggest that the three-stage pipeline is not a rigid architecture but a flexible template whose stages can be combined, collapsed, or reordered depending on deployment constraints.

This multidimensional perspective also motivates clearer method characterization and comparison. In particular, evaluating routing methods should explicitly report: when decisions are made (pre-generation, post-generation, or multi-stage routing), what signals are used (query, model metadata, response, feedback), and how decisions are computed (heuristic, supervised, bandit, RL policy, or combinations thereof). Methods may activate multiple categories simultaneously, and this compositionality should be reflected in both taxonomy and empirical reporting.

In addition to architectural mapping, our design-space matrix (cf. Table 1) also reveals structural gaps in the current literature that point to concrete directions for future work. First, no current method simultaneously pairs response-level signals with online adaptation, as uncertainty-based approaches and cascades exploit response signals but remain static once deployed, while bandit-based methods adapt online but operate

[5]Efficient inference frameworks such as vLLM (Kwon et al., 2023) offer server versions that can report several standard metrics such as TTFT, TPOT, throughput, and goodput under constraints such as concurrency and maximum token budget.

on query-level signals alone. This suggests an opportunity for bandit-style escalation policies that refine thresholds from deployment feedback. Second, reinforcement learning is limited in the cascading paradigm, and combining learned escalation policies with multi-stage architectures remains largely unexplored. Third, while some systems balance quality, cost, and latency via fixed tradeoff parameters or constraints, few formulate and solve a unified multi-objective optimization problem that treats all metrics as first-class, tunable objectives. Addressing these gaps will be critical for building routing systems that fully exploit the rich design space revealed by the multidimensional framework.

11 Conclusion and Future Directions

This survey has covered state-of-the-art approaches to dynamic routing methods for multi-LLM deployment at inference time. We organized approaches into six main paradigms: difficulty-aware routing, human preference alignment, clustering-based methods, reinforcement learning approaches, uncertainty quantification, and cascading systems. Table 2 provides a consolidated summary of all reviewed methods, organised by paradigm. Additionally, we examined emerging work on multimodal routing and discussed evaluation frameworks, including benchmarks and metrics. The key insight is that effective routing systems can outperform even the strongest individual models by strategically leveraging model complementarity and specialization while achieving considerable efficiency gains.

Despite rapid progress, various open challenges remain, with the following potential research directions:

- **Generalization:** Several routing methods are evaluated on a fixed set of LLMs and struggle to generalize to new models, domains, or data distributions. Developing retraining-free approaches that transfer across diverse architectures, tasks, and deployment scenarios remains an open challenge.
- **Multi-stage cascades:** Most work explores one-stage routing rather than cascading systems (cf. Section 7). As we outlined in Section 1.4 and Section 10, real-world deployments rarely conform to a single paradigm. Future research should reflect real-world applications, where queries and outputs are processed at multiple levels, aiming to strike a balance between quality and efficiency while also ensuring safety.
- **Multimodality:** As models increasingly span vision, audio, and language, routing research should deeply address modality-specific challenges, including fusing multimodal inputs for routing decisions, handling queries requiring multiple modalities, and accounting for varying computational costs of cross-modal architectures.

As the landscape of large foundation models continues to diversify, intelligent routing is becoming increasingly critical for efficient deployment. This survey aims at providing researchers and practitioners with a foundation for understanding current approaches and highlighting opportunities for advancing the state of the art in dynamic multi-LLM systems.

Method	Paradigm	Routing Model / Mechanism	Training Data
BEST-Route [§2] (Ding et al., 2025)	Difficulty	DeBERTa-v3-small (multi-head)	Question–score pairs
Semantic Router [§2] (Wang et al., 2025a)	Difficulty	ModernBERT classifier	Query intent labels
RouteLMT [§2] (Luo et al., 2026)	Difficulty	LoRA probe on small translator representations	Parallel translation pairs (ComMT)
EmbedLLM [§2] (Zhuang et al., 2024)	Difficulty	Matrix factorization (model embeddings)	Question-answer correctness data
ICL-Router [§2] (Wang et al., 2026)	Difficulty	LLM router + projector + in-context capability vectors	Query-LLM performance pairs
GraphRouter [§2] (Feng et al., 2024)	Difficulty	Graph Neural Network (GNN)	Historical performance + cost data
IRT-Router [§2] (Song et al., 2025)	Difficulty	IRT-based performance predictor (MIRT/NIRT)	Query-LLM performance pairs
RouteLLM [§3] (Ong et al., 2025)	Preference	Matrix factorization / BERT / Causal LLM	Chatbot Arena + LLM judge
Arch-Router [§3] (Tran et al., 2025)	Preference	1.5B parameter LLM	Synthetic preference data
Hybrid LLM [§3] (Ding et al., 2024)	Preference	LLM-based classifier	Synthetic preference data
P2L [§3] (Frick et al., 2025)	Preference	LLM (Bradley-Terry coefficients)	Chatbot Arena preferences
Eagle [§3] (Zhao et al., 2024)	Preference	ELO ranking (training-free)	Historical pairwise comparisons
Zooter [§3] (Lu et al., 2024)	Preference	mDeBERTa-v3-base classifier	Reward model preference labels
UniRoute [§4] (Jitkrittum et al., 2026)	Clustering	K-means + unsupervised text embedder	Unlabelled queries + validation set
Avengers-Pro [§4] (Zhang et al., 2025e)	Clustering	K-means + text embedder + scoring	Labelled query-answer pairs
Router-R1 [§5] (Zhang et al., 2025a)	RL (Policy Opt.)	Qwen2.5-3B / LLaMA-3.2-3B (PPO)	Question–score pairs
R2-Reasoner [§5] (Shao et al., 2025)	RL (Policy Opt.)	Task Decomposer + Subtask Allocator (GRPO)	SFT + RL training data
MetaLLM [§5] (Nguyen et al., 2025)	RL (Bandit)	Multi-armed bandit	Online accuracy-cost feedback
MixLLM [§5] (Wang et al., 2025b)	RL (Bandit)	Contextual bandit + policy gradient	Binary user feedback (online)
PILOT [§5] (Panda et al., 2025)	RL (Bandit)	LinUCB contextual bandit	Chatbot Arena priors + online feedback
GreenServ [§5] (Ziller et al., 2026)	RL (Bandit)	LinUCB contextual bandit	Online energy consumption + accuracy feedback
Dueling Feedback [§5] (Chiang et al., 2025)	RL (Bandit)	CCFT text encoder + FGTS.CDB	Pairwise preference comparisons
TI-UCB [§5] (Xia et al., 2024)	RL (Bandit)	Time-increasing UCB bandit	Online model performance
CP-Router [§6] (Su et al., 2025)	Uncertainty	Conformal prediction on logits	Calibration set
LLM-as-a-Judge [§6] (Zhang et al., 2025c)	Uncertainty	External LLM-as-a-Judge	Human rating preferences
AutoMix [§7] (Aggarwal et al., 2024)	Uncertainty (Cascade)	POMDP router + few-shot self-verification	Few-shot examples (no fine-tuning)
Self-REF [§7] (Chuang et al., 2025a)	Uncertainty (Cascade)	Fine-tuned LLM w/ confidence tokens	Correctness-labelled training data
FrugalGPT [§7] (Chen et al., 2024)	QE (Cascade)	LLM router + DistilBERT quality estimator	Query-response pairs
Cascade Routing [§7] (Dekoninck et al., 2025)	QE (Cascade)	Iterative model selection + quality estimator	Query-response pairs (log probabilities)
LM-Blender [§7] (Jiang et al., 2023)	Ensemble (Cascade)	Cross-attention Pair Ranker + Gen Fuser	Pairwise preference labels

Table 2: Summary of LLM routing and cascading methods, reflecting the main paradigms of this survey.

References

Pranjal Aggarwal, Aman Madaan, Ankit Anand, Srividya Pranavi Potharaju, Swaroop Mishra, Pei Zhou, Aditya Gupta, Dheeraj Rajagopal, Karthik Kappaganthu, Yiming Yang, Shyam Upadhyay, Mausam, and Manaal Faruqui. AutoMix: Automatically Mixing Language Models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems (NeurIPS 2024)*, 2024. URL https://openreview.net/forum?id=e6WrwIvgzX.

Pranjal Aggarwal, Seungone Kim, Jack Lanchantin, Sean Welleck, Jason Weston, Ilia Kulikov, and Swarnadeep Saha. OptimalThinkingBench: Evaluating Over and Underthinking in LLMs. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=N5kWa3sRJt.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. Qwen Technical Report. *arXiv [cs.CL]*, September 2023. URL http://dx.doi.org/10.48550/arXiv.2309.16609.

Qi Cao, Shuhao Zhang, Ruizhe Zhou, Ruiyi Zhang, Peijia Qin, and Pengtao Xie. Models under SCOPE: Scalable and controllable routing via pre-hoc reasoning. *arXiv [cs.LG]*, 13 February 2026. URL http://dx.doi.org/10.48550/arXiv.2601.22323.

Deeparnab Chakrabarty, Yunhong Zhou, and Rajan Lukose. Online Knapsack Problems. In *Workshop on internet and network economics (WINE)*, pp. 1–9, 2008.

Lingjiao Chen, Matei Zaharia, and James Zou. FrugalGPT: How to Use Large Language Models While Reducing Cost and Improving Performance. *Transactions on Machine Learning Research*, 2024. URL https://openreview.net/forum?id=cSimKw5p6R.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating large language models trained on code. *arXiv [cs.LG]*, 7 July 2021. URL http://dx.doi.org/10.48550/arXiv.2107.03374.

Chao-Kai Chiang, Takashi Ishida, and Masashi Sugiyama. LLM Routing with Dueling Feedback. *arXiv [cs.LG]*, October 2025. URL http://dx.doi.org/10.48550/arXiv.2510.00841.

Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios N Angelopoulos, Tianle Li, Dacheng Li, Banghua Zhu, Hao Zhang, Michael I Jordan, Joseph E Gonzalez, and Ion Stoica. Chatbot Arena: an open platform for evaluating LLMs by human preference. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24, Vienna, Austria, 2024. JMLR.org. URL https://arxiv.org/abs/2403.04132.

Yu-Neng Chuang, Prathusha Kameswara Sarma, Parikshit Gopalan, John Boccio, Sara Bolouki, Xia Hu, and Helen Zhou. Learning to Route LLMs with Confidence Tokens. In *Forty-second International Conference on Machine Learning*, June 2025a. URL https://openreview.net/forum?id=U08mUogGDM.

Yu-Neng Chuang, Leisheng Yu, Guanchu Wang, Lizhe Zhang, Zirui Liu, Xuanting Cai, Yang Sui, Vladimir Braverman, and Xia Hu. Confident or seek stronger: Exploring uncertainty-based on-device LLM routing from benchmarking to generalization. *arXiv [cs.CL]*, February 2025b. URL http://dx.doi.org/10.48550/arXiv.2502.04428.

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. Training verifiers to solve math word problems. *arXiv [cs.LG]*, 27 October 2021. URL http://arxiv.org/abs/2110.14168.

Jasper Dekoninck, Maximilian Baader, and Martin Vechev. A Unified Approach to Routing and Cascading for LLMs. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=AAl89VNNy1.

Dujian Ding, Ankur Mallick, Chi Wang, Robert Sim, Subhabrata Mukherjee, Victor Rühle, Laks V S Lakshmanan, and Ahmed Hassan Awadallah. Hybrid LLM: Cost-Efficient and Quality-Aware Query Routing. In *The Twelfth International Conference on Learning Representations (ICLR)*, 2024. URL https://openreview.net/forum?id=02f3mUtqnM.

Dujian Ding, Ankur Mallick, Shaokun Zhang, Chi Wang, Daniel Madrigal, Mirian Del Carmen Hipolito Garcia, Menglin Xia, Laks V S Lakshmanan, Qingyun Wu, and Victor Rühle. BEST-Route: Adaptive LLM Routing with Test-Time Optimal Compute. In *Forty-second International Conference on Machine Learning*, June 2025. URL https://openreview.net/forum?id=tFBIbCVXkG.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vladan Petrovic, Weiwei Chu,

Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaoqing Ellen Tan, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aaron Grattafiori, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alex Vaughan, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Franco, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, Danny Wyatt, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Firat Ozgenel, Francesco Caggioni, Francisco Guzmán, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Govind Thattai, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, U, Kam Hou, Karan Saxena, Karthik Prasad, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kun Huang, Kunal Chawla, Kushal Lakhotia, Kyle Huang, Lailin Chen, Lakshya Garg, A, Lavender, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Maria Tsimpoukelli, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Michael L Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikolay Pavlovich Laptev, Ning Dong, Ning Zhang, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Rohan Maheswari, Russ Howes, Ruty Rinott, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Kohler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vítor Albiero, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaofang Wang, Xiaojian Wu, Xiaolan Wang, Xide Xia, Xilun Wu, Xinbo Gao, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yuchen Hao, Yundi Qian, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, and Zhiwei Zhao. The Llama 3 herd of models. *arXiv [cs.AI]*, July 2024. URL `http://arxiv.org/abs/2407.21783`.

Arpad E Elo. The Proposed USCF Rating System, Its Development, Theory, and Applications. *Chess Life*, 22(8):242–247, August 1967.

António Farinhas, Nuno M Guerreiro, Sweta Agrawal, Ricardo Rei, and Andre Martins. Translate smart, not hard: Cascaded translation systems with quality-aware deferral. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 26730–26744, Stroudsburg, PA, USA, November 2025. Association for Computational Linguistics. URL https://aclanthology.org/2025.emnlp-main.1358/.

Tao Feng, Yanzhen Shen, and Jiaxuan You. GraphRouter: A Graph-based Router for LLM Selections. In *The Thirteenth International Conference on Learning Representations*, October 2024. URL https://openreview.net/forum?id=eU39PDsZtT.

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

Ma Haoxuan, Lai Guannan, and Ye Han-Jia. MMR-Bench: A Comprehensive Benchmark for Multimodal LLM Routing. *arXiv [cs.AI]*, 25 January 2026. URL http://dx.doi.org/10.48550/arXiv.2601.17814.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. DeBERTa: Decoding-enhanced BERT with Disentangled Attention. In *Proceedings of the 9th International Conference on Learning Representations (ICLR)*, Virtual, May 2021. URL http://arxiv.org/abs/2006.03654.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. DeBERTaV3: Improving DeBERTa using ELECTRA-Style Pre-Training with Gradient-Disentangled Embedding Sharing. In *The Eleventh International Conference on Learning Representations (ICLR)*, 2023. URL https://openreview.net/forum?id=sE7-XhLxHA.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring Massive Multitask Language Understanding. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=d7KBjmI3GmQ.

Amr Hendy, Mohamed Abdelrehim, Amr Sharaf, Vikas Raunak, Mohamed Gabr, Hitokazu Matsushita, Young Jin Kim, Mohamed Afify, and Hany Hassan Awadalla. How Good Are GPT Models at Machine Translation? A Comprehensive Evaluation. *arXiv [cs.CL]*, 18 February 2023. URL http://arxiv.org/abs/2302.09210.

Qitian Jason Hu, Jacob Bieker, Xiuyu Li, Nan Jiang, Benjamin Keigwin, Gaurav Ranganath, Kurt Keutzer, and Shriyash Kaustubh Upadhyay. RouterBench: A Benchmark for Multi-LLM Routing System. In *Agentic Markets Workshop at ICML 2024*, 2024. URL https://openreview.net/forum?id=IVXmV8Uxwh.

Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. Large Language Models Cannot Self-Correct Reasoning Yet. In *The Twelfth International Conference on Learning Representations (ICLR)*, 2024. URL https://openreview.net/forum?id=IkmD3fKBPQ.

Zhongzhan Huang, Guoming Ling, Yupei Lin, Yandong Chen, Shanshan Zhong, Hefeng Wu, and Liang Lin. RouterEval: A comprehensive benchmark for routing LLMs to explore model-level scaling up in LLMs. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 3860–3887, Stroudsburg, PA, USA, November 2025. Association for Computational Linguistics. URL https://aclanthology.org/2025.findings-emnlp.208/.

Chip Huyen. *AI Engineering*. O'Reilly Media, USA, 2025.

Dongfu Jiang, Xiang Ren, and Bill Yuchen Lin. 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)*, pp. 14165–14178, Stroudsburg, PA, USA, 2023. Association for Computational Linguistics. URL https://aclanthology.org/2023.acl-long.792/.

Shuhao Jiang, Songbo Wang, Yang Qiao, Chun Xu, Chaoyang Zheng, Shengyi Zhou, Huanjun Wang, Fangming Li, Cong Zhang, and Jiyu Wang. Adaptive Overclocking: Dynamic control of thinking path length via real-time reasoning signals. *arXiv [cs.LG]*, 21 September 2025. URL http://dx.doi.org/10.48550/arXiv.2509.17000.

Wittawat Jitkrittum, Harikrishna Narasimhan, Ankit Singh Rawat, Jeevesh Juneja, Congchao Wang, Zifeng Wang, Alec Go, Chen-Yu Lee, Pradeep Shenoy, Rina Panigrahy, Aditya Krishna Menon, and Sanjiv Kumar. Universal model routing for efficient LLM inference. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=ka82fvJ5f1.

Sumith Kulal, Panupong Pasupat, Kartik Chandra, Mina Lee, Oded Padon, Alex Aiken, and Percy Liang. SPoC: Search-based Pseudocode to Code. In Hanna M Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché Buc, Emily Fox, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32, pp. 11883–11894. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/7298332f04ac004a0ca44cc69ecf6f6b-Paper.pdf.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with PagedAttention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, pp. 611–626, New York, NY, USA, 23 October 2023. ACM. URL https://dl.acm.org/doi/10.1145/3600006.3613165.

Hao Li, Yiqun Zhang, Zhaoyan Guo, Chenxu Wang, Shengji Tang, Qiaosheng Zhang, Yang Chen, Biqing Qi, Peng Ye, Lei Bai, Zhen Wang, and Shuyue Hu. LLMRouterBench: A massive benchmark and unified framework for LLM routing. *arXiv [cs.AI]*, 12 January 2026. URL http://dx.doi.org/10.48550/arXiv.2601.07206.

Lihong Li, Wei Chu, John Langford, and Robert E Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, pp. 661–670, New York, NY, USA, April 2010. ACM. URL https://dl.acm.org/doi/10.1145/1772690.1772758.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's Verify Step by Step. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=v8L0pN6EOi.

Keming Lu, Hongyi Yuan, Runji Lin, Junyang Lin, Zheng Yuan, Chang Zhou, and Jingren Zhou. Routing to the expert: Efficient reward-guided ensemble of large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 1964–1974, Stroudsburg, PA, USA, 2024. Association for Computational Linguistics. URL https://aclanthology.org/2024.naacl-long.109.

Yingfeng Luo, Hongyu Liu, Dingyang Lin, Kaiyan Chang, Chenglong Wang, Bei Li, Quan Du, Tong Xiao, and Jingbo Zhu. RouteLMT: Learned Sample Routing for Hybrid LLM Translation Deployment. In *The 64th Annual Meeting of the Association for Computational Linguistics – Industry Track*, 2026. URL https://openreview.net/forum?id=0SMriQrEjL.

Matéo Mahaut, Laura Aina, Paula Czarnowska, Momchil Hardalov, Thomas Müller, and Lluis Marquez. Factual confidence of LLMs: On reliability and robustness of current estimators. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 4554–4570, Stroudsburg, PA, USA, 2024. Association for Computational Linguistics. URL https://aclanthology.org/2024.acl-long.250/.

Yasmin Moslem, Rejwanul Haque, John D Kelleher, and Andy Way. Adaptive Machine Translation with Large Language Models. In *Proceedings of the 24th Annual Conference of the European Association for Machine Translation*, pp. 227–237, Tampere, Finland, June 2023a. European Association for Machine Translation. URL https://aclanthology.org/2023.eamt-1.22.

Yasmin Moslem, Gianfranco Romani, Mahdi Molaei, John D Kelleher, Rejwanul Haque, and Andy Way. Domain Terminology Integration into Machine Translation: Leveraging Large Language Models. In *Proceedings of the Eighth Conference on Machine Translation*, pp. 902–911, Singapore, October 2023b. Association for Computational Linguistics. URL https://aclanthology.org/2023.wmt-1.82.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, Xu Jiang, Karl Cobbe, Tyna Eloundou, Gretchen Krueger, Kevin Button, Matthew Knight, Benjamin Chess, and John Schulman. WebGPT: Browser-assisted question-answering with human feedback. *arXiv [cs.CL]*, December 2021. URL http://dx.doi.org/10.48550/arXiv.2112.09332.

Quang H Nguyen, Thinh Dao, Duy C Hoang, Juliette Decugis, Saurav Manchanda, Nitesh V Chawla, and Khoa D Doan. MetaLLM: A high-performant and cost-efficient dynamic framework for wrapping LLMs. *arXiv [cs.LG]*, April 2025. URL http://dx.doi.org/10.48550/arXiv.2407.10834.

Isaac Ong, Amjad Almahairi, Vincent Wu, Wei-Lin Chiang, Tianhao Wu, Joseph E Gonzalez, M Waleed Kadous, and Ion Stoica. RouteLLM: Learning to Route LLMs with Preference Data. In *The Thirteenth International Conference on Learning Representations (ICLR)*, 2025. URL https://openreview.net/forum?id=8sSqNntaMr.

Pranoy Panda, Raghav Magazine, Chaitanya Devaguptapu, Sho Takemori, and Vishal Sharma. Adaptive LLM Routing under Budget Constraints. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 23934–23949, Stroudsburg, PA, USA, November 2025. Association for Computational Linguistics. URL https://aclanthology.org/2025.findings-emnlp.1301/.

Runyu Peng, Yunhua Zhou, Kai Lv, Yang Gao, Qipeng Guo, and Xipeng Qiu. Firewall routing: Blocking leads to better hybrid inference for LLMs. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 6540–6565, Stroudsburg, PA, USA, November 2025. Association for Computational Linguistics. URL https://aclanthology.org/2025.emnlp-main.331/.

Maja Popović. chrF++: words helping character n-grams. In *Proceedings of the Second Conference on Machine Translation*, pp. 612–618, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. URL https://aclanthology.org/W17-4770.

Lorenzo Proietti, Stefano Perrella, Vilém Zouhar, Roberto Navigli, and Tom Kocmi. Estimating Machine Translation Difficulty. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 24261–24285, Stroudsburg, PA, USA, November 2025. Association for Computational Linguistics. URL https://aclanthology.org/2025.findings-emnlp.1317/.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. COMET: A Neural Framework for MT Evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 2685–2702, Online, November 2020. Association for Computational Linguistics. URL https://aclanthology.org/2020.emnlp-main.213.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. In *5th Workshop on Energy Efficient Machine Learning and Cognitive Computing @ NeurIPS 2019*, 2019. URL https://www.emc2-ai.org/assets/docs/neurips-19/emc2-neurips19-paper-33.pdf.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal Policy Optimization Algorithms. *arXiv [cs.LG]*, 20 July 2017. URL http://arxiv.org/abs/1707.06347.

Chenyang Shao, Xinyang Liu, Yutang Lin, Fengli Xu, and Yong Li. Route-and-reason: Scaling large language model reasoning with Reinforced Model Router. *arXiv [cs.CL]*, December 2025. URL http://dx.doi.org/10.48550/arXiv.2506.05901.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Mingchuan Zhang, Y K Li, Y Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv [cs.CL]*, February 2024. URL http://arxiv.org/abs/2402.03300.

Yi Shen, Jian Zhang, Jieyun Huang, Shuming Shi, Wenjing Zhang, Jiangze Yan, Ning Wang, Kai Wang, Zhaoxiang Liu, and Shiguo Lian. DAST: Difficulty-adaptive slow-thinking for large reasoning models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: Industry*

Track, pp. 2322–2331, Stroudsburg, PA, USA, November 2025. Association for Computational Linguistics. URL https://aclanthology.org/2025.emnlp-industry.160/.

Wei Song, Zhenya Huang, Cheng Cheng, Weibo Gao, Bihan Xu, Guanhao Zhao, Fei Wang, and Runze Wu. IRT-router: Effective and interpretable multi-LLM routing via item response theory. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15629–15644, Stroudsburg, PA, USA, 2025. Association for Computational Linguistics. URL https://aclanthology.org/2025.acl-long.761/.

Lucia Specia, Carolina Scarton, and Gustavo Henrique Paetzold. *Quality Estimation for Machine Translation*. Synthesis Lectures on Human Language Technologies. Springer International Publishing, Cham, 2018. URL https://link.springer.com/book/10.1007/978-3-031-02168-8.

Zayne Rea Sprague, Fangcong Yin, Juan Diego Rodriguez, Dongwei Jiang, Manya Wadhwa, Prasann Singhal, Xinyu Zhao, Xi Ye, Kyle Mahowald, and Greg Durrett. To CoT or not to CoT? Chain-of-thought helps mainly on math and symbolic reasoning. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=w6nlcS8Kkn.

Nisan Stiennon, Long Ouyang, Jeff Wu, Daniel M Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul Christiano. Learning to summarize from human feedback. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NeurIPS'20, Red Hook, NY, USA, 2020. Curran Associates Inc. URL https://proceedings.neurips.cc/paper/2020/file/1f89885d556929e98d3ef9b86448f951-Paper.pdf.

Jiayuan Su, Fulin Lin, Zhaopeng Feng, Han Zheng, Teng Wang, Zhenyu Xiao, Xinlong Zhao, Zuozhu Liu, Lu Cheng, and Hongwei Wang. CP-Router: An uncertainty-aware Router between LLM and LRM. *arXiv [cs.CL]*, May 2025. URL http://dx.doi.org/10.48550/arXiv.2505.19970.

Katherine Tian, Eric Mitchell, Allan Zhou, Archit Sharma, Rafael Rafailov, Huaxiu Yao, Chelsea Finn, and Christopher Manning. Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 5433–5442, Stroudsburg, PA, USA, December 2023. Association for Computational Linguistics. URL https://aclanthology.org/2023.emnlp-main.330/.

Co Tran, Salman Paracha, Adil Hafeez, and Shuguang Chen. Arch-Router: Aligning LLM routing with human preferences. *arXiv [cs.CL]*, June 2025. URL http://dx.doi.org/10.48550/arXiv.2506.16655.

Chen Wang, Xunzhuo Liu, Yuhan Liu, Yue Zhu, Xiangxi Mo, Junchen Jiang, and Huamin Chen. When to Reason: Semantic Router for vLLM. In *Proceedings of the Workshop on Machine Learning for Systems (ML for Systems) at NeurIPS 2025*, December 2025a. URL https://openreview.net/forum?id=MGeOrfovNn.

Chenxu Wang, Hao Li, Yiqun Zhang, Linyao Chen, Jianhao Chen, Ping Jian, Peng Ye, Qiaosheng Zhang, and Shuyue Hu. ICL-Router: In-Context Learned Model Representations for LLM Routing. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI-26)*. AAAI Press, 2026.

Xinyuan Wang, Yanchi Liu, Wei Cheng, Xujiang Zhao, Zhengzhang Chen, Wenchao Yu, Yanjie Fu, and Haifeng Chen. MixLLM: Dynamic Routing in Mixed Large Language Models. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 10912–10922. Association for Computational Linguistics, 2025b. URL https://aclanthology.org/2025.naacl-long.545/.

Benjamin Warner, Antoine Chaffin, Benjamin Clavié, Orion Weller, Oskar Hallström, Said Taghadouini, Alexis Gallagher, Raja Biswas, Faisal Ladhak, Tom Aarsen, Griffin Thomas Adams, Jeremy Howard, and Iacopo Poli. Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2526–2547, Stroudsburg, PA, USA, 2025. Association for Computational Linguistics. URL https://aclanthology.org/2025.acl-long.127/.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. In S Koyejo, S Mohamed, A Agarwal, D Belgrave, K Cho, and A Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 24824–24837. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/9d5609613524ecf4f15af0f7b31abca4-Paper-Conference.pdf.

Chenyang Wu, Guoyu Yang, Zongzhang Zhang, Yang Yu, Dong Li, Wulong Liu, and Jianye Hao. Adaptive online packing-guided search for POMDPs. *Advances in Neural Information Processing Systems*, 34: 28419–28430, 2021. URL https://openreview.net/forum?id=0zvTBoQb5PA.

Menghua Wu, Cai Zhou, Stephen Bates, and Tommi Jaakkola. Thought calibration: Efficient and confident test-time scaling. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 14302–14316, Stroudsburg, PA, USA, November 2025a. Association for Computational Linguistics. URL https://aclanthology.org/2025.emnlp-main.722/.

Zhanglin Wu, Daimeng Wei, Xiaoyu Chen, Hengchao Shang, Jiaxin Guo, Zongyao Li, Yuanchang Luo, Jinlong Yang, Zhiqiang Rao, and Hao Yang. Combining the best of both worlds: A method for hybrid NMT and LLM translation. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 5140–5148, Stroudsburg, PA, USA, 2025b. Association for Computational Linguistics. URL https://aclanthology.org/2025.findings-acl.266/.

Yu Xia, Fang Kong, Tong Yu, Liya Guo, Ryan A Rossi, Sungchul Kim, and Shuai Li. Which LLM to play? Convergence-aware online model selection with time-increasing bandits. In *Proceedings of the ACM Web Conference 2024*, volume 1372, pp. 4059–4070, New York, NY, USA, May 2024. ACM. URL https://dl.acm.org/doi/10.1145/3589334.3645420.

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, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 Technical Report. *arXiv [cs.CL]*, December 2024. URL http://arxiv.org/abs/2412.15115.

Yaopei Zeng, Congchao Wang, Blake Jianhang Chen, and Lu Lin. ReLope: KL-regularized LoRA probes for multimodal LLM routing. *arXiv [cs.AI]*, 25 March 2026. URL http://dx.doi.org/10.48550/arXiv.2603.24787.

Chrysoula Zerva, Frederic Blain, José G C. De Souza, Diptesh Kanojia, Sourabh Deoghare, Nuno M Guerreiro, Giuseppe Attanasio, Ricardo Rei, Constantin Orasan, Matteo Negri, Marco Turchi, Rajen Chatterjee, Pushpak Bhattacharyya, Markus Freitag, and André Martins. Findings of the quality estimation shared task at WMT 2024: Are LLMs closing the gap in QE? In *Proceedings of the Ninth Conference on Machine Translation*, pp. 82–109, Stroudsburg, PA, USA, November 2024. Association for Computational Linguistics. URL https://aclanthology.org/2024.wmt-1.3/.

Haozhen Zhang, Tao Feng, and Jiaxuan You. Router-R1: Teaching LLMs Multi-Round Routing and Aggregation via Reinforcement Learning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems (NeurIPS 2025)*, 2025a. URL https://openreview.net/forum?id=DWf4vroKWJ.

Jiajie Zhang, Nianyi Lin, Lei Hou, Ling Feng, and Juanzi Li. AdaptThink: Reasoning models can learn when to think. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 3716–3730, Stroudsburg, PA, USA, November 2025b. Association for Computational Linguistics. URL https://aclanthology.org/2025.emnlp-main.184/.

Tuo Zhang, Asal Mehradfar, Dimitrios Dimitriadis, and Salman Avestimehr. Leveraging Uncertainty Estimation for Efficient LLM Routing. In *ICML 2025 Workshop on Collaborative and Federated Agentic Workflows (CFAgentic@ICML'25)*, Vancouver, Canada, 19 July 2025c. URL https://openreview.net/forum?id=DJpEIwKJt7.

Xiaoyun Zhang, Jingqing Ruan, Xing Ma, Yawen Zhu, Haodong Zhao, Hao Li, Jiansong Chen, Ke Zeng, and Xunliang Cai. When to continue thinking: Adaptive thinking mode switching for efficient reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 5808–5828, Stroudsburg, PA, USA, November 2025d. Association for Computational Linguistics. URL https://aclanthology.org/2025.findings-emnlp.310/.

Yi-Kai Zhang, Ting-Ji Huang, Yao-Xiang Ding, De-Chuan Zhan, and Han-Jia Ye. Model Spider: learning to rank pre-trained models efficiently. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc.

Yiqun Zhang, Hao Li, Jianhao Chen, Hangfan Zhang, Peng Ye, Lei Bai, and Shuyue Hu. Beyond GPT-5: Making LLMs cheaper and better via performance-efficiency optimized routing. In *Proceedings of the 2025 The Seventh International Conference on Distributed Artificial Intelligence*, pp. 122–129, New York, NY, USA, November 2025e. ACM. URL https://dl.acm.org/doi/10.1145/3772429.3772445.

Zesen Zhao, Shuowei Jin, and Z Morley Mao. Eagle: Efficient Training-Free Router for Multi-LLM Inference. In *Machine Learning for Systems Workshop at NeurIPS*, 2024. URL https://mlforsystems.org/assets/papers/neurips2024/paper38.pdf.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph E Gonzalez, and Ion Stoica. Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, November 2023. URL https://openreview.net/forum?id=uccHPGDlao.

Richard Zhuang, Tianhao Wu, Zhaojin Wen, Andrew Li, Jiantao Jiao, and Kannan Ramchandran. EmbedLLM: Learning Compact Representations of Large Language Models. In *The Thirteenth International Conference on Learning Representations*, October 2024. URL https://openreview.net/forum?id=Fs9EabmQrJ.

Thomas Ziller, Shashikant Ilager, Alessandro Tundo, Ezio Bartocci, Leonardo Mariani, and Ivona Brandic. GreenServ: Energy-efficient context-aware dynamic routing for multi-model LLM inference. *arXiv [cs.PF]*, February 2026. URL http://dx.doi.org/10.48550/arXiv.2601.17551.

Karl Johan Åström. Optimal Control of Markov Processes with Incomplete State Information. *Journal of mathematical analysis and applications*, 10:174–205, 1965.