Harnessing Multiple Large Language Models: A Survey on LLM Ensemble

Zhijun Chen[1], **Xiaodong Lu**[1], **Jingzheng Li**[2], **Pengpeng Chen**[3], **Zhuoran Li**[4], **Kai Sun**[5], **Yuankai Luo**[6], **Qianren Mao**[2], **Ming Li**[7], **Likang Xiao**[1], **Dingqi Yang**[8], **Yikun Ban**[1,*], **Hailong Sun**[1,*]
[1]State Key Laboratory of Complex & Critical Software Environment, Beihang University, Beijing, China
[2]Zhongguancun Laboratory, Beijing, China
[3]Aviation System Engineering Institute of China, Beijing, China
[4]Beijing University of Posts and Telecommunications, Beijing, China
[5]Xi'an Jiaotong University, Xi'an, China
[6]Nanjing University, Nanjing, China
[7]Tsinghua University, Beijing, China
[8]University of Macau, Macau SAR, China
{zhijunchen, xiaodonglu, jingzhengli, yikunb, sunhl}@buaa.edu.cn, maoqr@zgclab.edu.cn

Abstract

LLM Ensemble—which involves the comprehensive use of multiple large language models (LLMs), each aimed at handling user queries during downstream inference, to benefit from their individual strengths—has gained substantial attention recently. The widespread availability of LLMs, coupled with their varying strengths and out-of-the-box usability, has profoundly advanced the field of LLM Ensemble. This paper presents the first systematic review of recent developments in LLM Ensemble. First, we introduce our taxonomy of LLM Ensemble and discuss several related research problems. Then, we provide a more in-depth classification of the methods under the broad categories of "ensemble-before-inference, ensemble-during-inference, ensemble-after-inference", and review relevant methods. Finally, we introduce related benchmarks and applications, summarize existing studies, and suggest future research directions. GitHub project link is: https://github.com/junchenzhi/Awesome-LLM-Ensemble.

1 Introduction

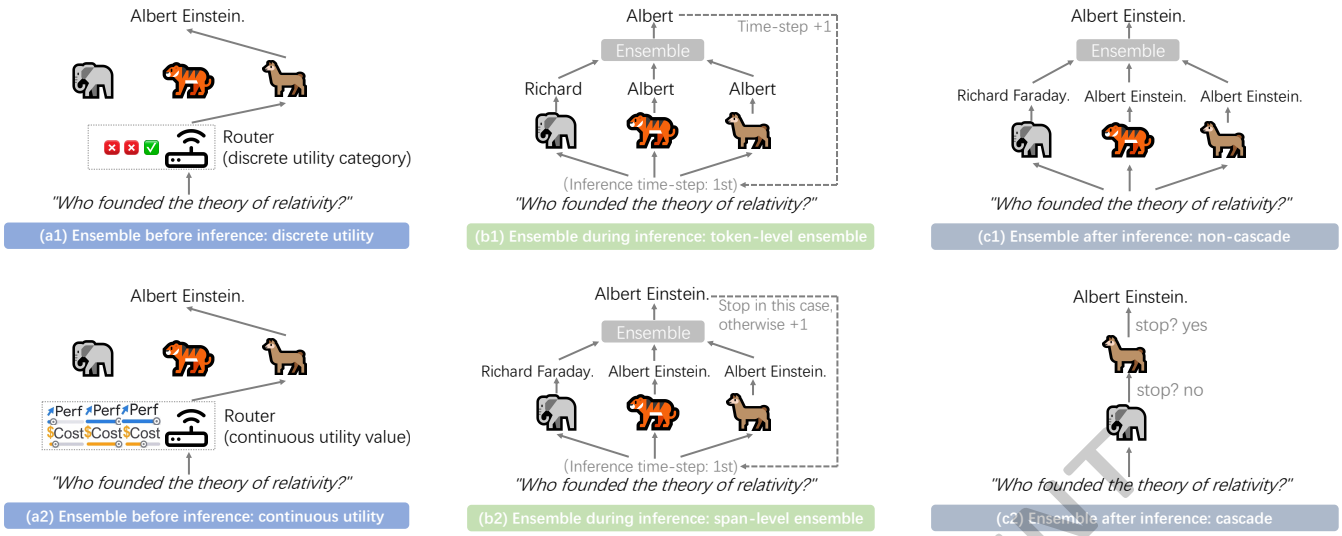
In recent years, the landscape of artificial intelligence has been dramatically reshaped by the development of Large Language Models (LLMs), including Gemini [Team *et al.*, 2023], GPT-4 [Achiam *et al.*, 2023], Llama [Touvron *et al.*, 2023], and the recently introduced DeepSeek [Liu *et al.*, 2024a]. The success of these LLMs continues to fuel widespread research enthusiasm, with a remarkable total of over 182,000 large language models now accessible in the Hugging Face library. [1]

Behind this research enthusiasm, however, we can identify two main aspects: 1) *The performance concerns*: The direct out-of-the-box capability of LLMs (from zero-shot inference) and their indirect out-of-the-box capability (from in-context-learning few-shot inference) still raise performance worries, including accuracy, hallucinations, and misalignment with human intent, among others; 2) *The varying strengths and weaknesses of LLMs, each with different inference costs*: Due to differences in architecture, size, tokenization, dictionary, training data, methodology, these LLMs exhibit substantial variability and their responses can differ significantly. With the above two aspects in mind and drawing on the spirit of Ensemble Learning [Dong *et al.*, 2020; Zhou, 2021; Mohammed and Kora, 2023], it is natural to consider that, for each query, rather than relying on a single LLM based on public rankings or other criteria, it might be more advantageous to simultaneously consider multiple LLM candidates (usable out-of-the-box) and harness their distinct strengths. This is exactly what the emerging field of *LLM Ensemble* explores.

Existing LLM Ensemble methods can be broadly categorized into three types, depending on the sequence of *LLM inference* and *ensemble*: 1) *Ensemble-before-inference* approach, utilizes the given query information while considering the diverse characteristics of all LLM candidates to route an appropriate model for inference (this approach is similar to the *hard voting* strategy in Ensemble Learning); 2) *Ensemble-during-inference* approach, aggregates incomplete responses (e.g., token-level information) from multiple LLMs during the decoding process and feeds the combined result back into all the models; 3) *Ensemble-after-inference* approach, performs the ensemble after full responses (instead of fragments) have been generated by all models or a subset of them. Despite the emergence of numerous methods derived from these paradigms recently, there is still no formal survey that offers a comprehensive review of the core ideas and related research.

We present the first comprehensive survey on LLM Ensemble, introducing recent advances and focusing on taxonomy, related problems, methods, benchmarks, applications, and future directions. We hope that this survey will provide a thorough review for researchers and inspire further exploration.

*Corresponding authors.
[1]https://huggingface.co/models.



(a) Ensemble before inference. (b) Ensemble during inference. (c) Ensemble after inference.

Figure 1: Illustration of the LLM Ensemble taxonomy. (Note that for *(b) ensemble-during-inference*, there is also a *(b3) process-level ensemble* approach not depicted in the figure, due to layout considerations and the status that this approach is rarely instantiated.)

2 LLM Ensemble Taxonomy and Related Problems

2.1 LLM Ensemble Taxonomy

This section formally introduces our LLM Ensemble taxonomy, illustrated by the schematic shown in Figure 1 and the detailed taxonomy in Figure 2. As mentioned in Section 1, the following three broad categories of LLM Ensemble exist.

(a) Ensemble before inference. In essence, this approach employs a routing algorithm prior to LLM inference to allocate a specific query to the most suitable model, allowing the selected model that is specialized for the query and typically more cost-efficient inference to perform the task. As illustrated in Figure 1a and Figure 2, existing methods can be classified into two categories, depending on whether the router utility is discretized: (a1) *discrete utility methods* and (a2) *continuous utility methods*.

(b) Ensemble during inference. As the most granular form of ensemble among the three broad categories, this type of approach encompasses: (b1) *token-level ensemble* methods, integrate the token-level outputs of multiple models at the finest granularity of decoding; (b2) *span-level ensemble methods*, perform ensemble at the level of a sequence fragment (e.g., a span of four words); (b3) *process-level ensemble methods*, select the optimal reasoning process step-by-step within the reasoning chain for a given complex reasoning task. *Note that for these ensemble-during-inference methods, the aggregated text segments will be concatenated with the preceding text and fed back into the models.*

(c) Ensemble after inference. These methods can be classified into two categories: (c1) *Non-cascade methods*, perform ensemble by integrating multiple complete responses contributed from all LLM candidates; (c2) *Cascade methods*, consider both performance and inference costs, progressively performing inference through a chain of LLM candidates ranked primarily by model size to identify the most suitable response and terminate the cascade process.

2.2 Related Problems

Here we briefly introduce the closely related problems.

LLM Merging, also known as **LLM Fusion** [Yang *et al.*, 2024; Yang *et al.*, 2025; Ruan *et al.*, 2025; Li *et al.*, 2023], integrates parameters from various LLMs to construct a universal model without requiring original training data or extensive computation. It is highly relevant to LLM Ensemble, as both promote knowledge fusion and transfer.

LLM Collaboration leverages the distinct strengths of each model to approach tasks with increased flexibility [Du *et al.*, 2024; Lu *et al.*, 2024a; Chen *et al.*, 2025b; Tran *et al.*, 2025; Zhang *et al.*, 2025a; Li *et al.*, 2024c; Guo *et al.*, 2024]. Unlike LLM Ensemble where models are employed with equal status to *directly faced to user queries*, the collaboration approach assigns distinct roles to each LLM, exchanging response information to enhance task resolution. Furthermore, a closely related line of research to LLM Collaboration—which mainly focuses on zero-shot inference—is **Multi-LLM Reinforcement Learning** [Sun *et al.*, 2024; Jin *et al.*, 2025; Li *et al.*, 2026b], which investigates how multiple agents learn optimal decision-making strategies in a shared environment.

Weak Supervision [Zhang *et al.*, 2021; Chen *et al.*, 2023b; Zhang *et al.*, 2022; Ratner *et al.*, 2017], sometimes referred to as **Learning from Crowds** [Chen *et al.*, 2020; Chen *et al.*, 2022; Zhang, 2022], uses weak labels contributed from multiple weak annotation sources to perform information aggregation [Chen *et al.*, 2023a; Zheng *et al.*, 2017] (corresponding *non-cascade ensemble after inference* in LLM Ensemble) or directly train a classifier [Zhang *et al.*, 2021], yet most methods focus on classification rather than general open-ended generation tasks.

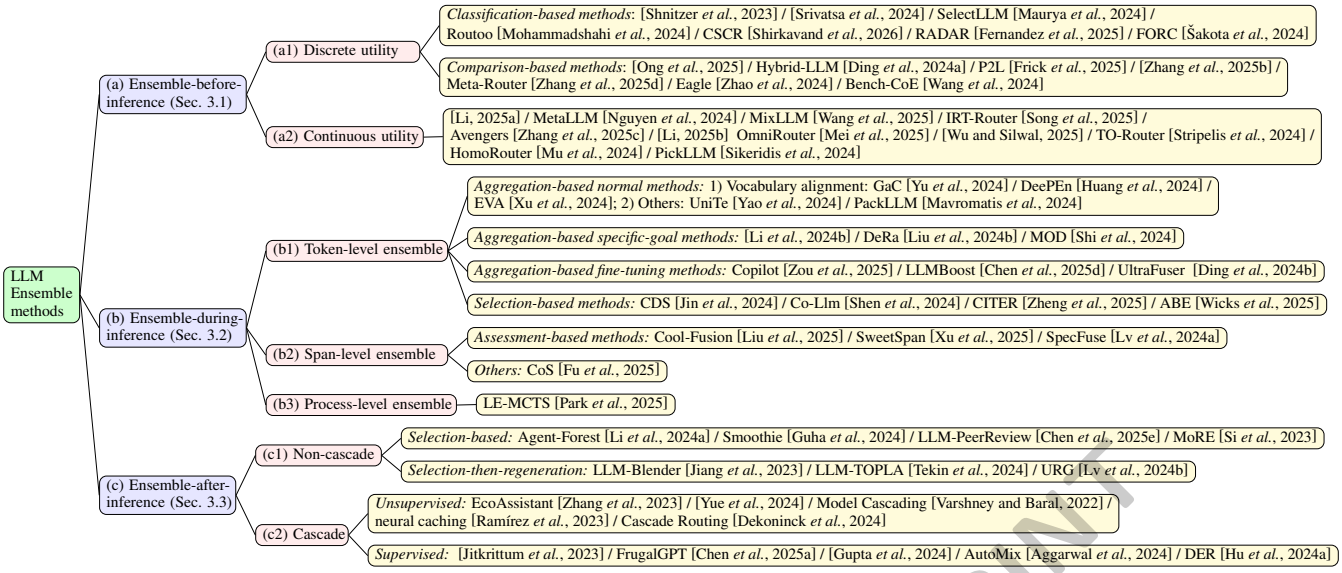


Figure 2: Taxonomy of LLM Ensemble methods.

3 Methodology

In this section, following the taxonomy in Section 2.1, we systematically review the three types of methods—ensemble before inference, ensemble during inference and ensemble after inference—in Sections 3.1, 3.2, and 3.3, respectively.

3.1 Ensemble Before Inference

Since these methods route a query to the most suitable LLM before inference, their core lies in predicting candidate model utility for a given query under specific preferences (e.g., performance or cost). Based on how utility is formulated, we divide existing methods into *discrete utility methods* and *continuous utility methods*, as shown in Table 1.

(a1) Discrete Utility Methods

Discrete utility methods discretize the model utility into *categorical labels*. Depending on whether categories are defined independently or via pairwise comparisons, they are divided into *classification-based* and *comparison-based methods*.

Classification-based methods. For this type of method, the utility of a candidate LLM for a given query is typically treated as a discrete variable reflecting the quality of the model's response. Typically, the utility is binary: a value of 1 indicates that the model produces a satisfactory response, whereas 0 indicates an unsatisfactory one. For example, in a commonsense question-answering (QA) task, a correct answer is labeled as 1 and an incorrect answer as 0 (e.g., [Shnitzer *et al.*, 2023]). Under this formulation, the routing problem naturally becomes a multi-label binary classification task, where the router estimates the probability that each candidate LLM will produce a satisfactory output. Based on different optimization objectives, one can then design corresponding decision strategies—for instance, when model costs are known, the router can compute a combined value via a weighted aggregation of the predicted score and cost [Fernandez *et al.*, 2025], and select the model with the highest combined value to balance performance and cost. Following this line, existing methods extend the classification-based methods along *empirical analysis* [Srivatsa *et al.*, 2024], *decision policy* [Maurya *et al.*, 2024; Shnitzer *et al.*, 2023; Šakota *et al.*, 2024], and *router architecture* [Shirkavand *et al.*, 2026; Mohammadshahi *et al.*, 2024].

Comparison-based methods. Compared to multi-label binary classification routing, which requires estimating an absolute (pointwise) utility score for each candidate LLM's output, comparison-based approaches only need to infer relative preferences among a pair of model outputs, thereby simplifying the routing objective and reducing supervision difficulty [Zhang *et al.*, 2025d]. Specifically, a preference datum can be represented as a tuple (q, M_1, M_2, y), where q denotes a given query, M_1 and M_2 are two candidate LLMs, and y is a binary label such that $y = 1$ indicates that the response of M_1 is better than that of M_2, while $y = 0$ indicates otherwise (e.g., [Ong *et al.*, 2025]). In this setting, the router aims to predict the win rate of M_1, i.e., the probability that M_1's response is better given the query q and the model pair (M_1, M_2). Following this idea, subsequent methods extend the comparison-based routing framework along several directions, including *data construction* [Ong *et al.*, 2025; Zhang *et al.*, 2025b], *learning target* [Frick *et al.*, 2025; Chen *et al.*, 2025c; Wang *et al.*, 2024], *routing strategy* [Zhang *et al.*, 2025d; Zhao *et al.*, 2024]

(a2) Continuous Utility Methods

Unlike prior discrete-category approaches, continuous-utility methods model LLM utility as real-valued variables, such as response length or performance scores. This formulation enables a fine-grained characterization of model behavior, capturing subtle performance differences obscured by categorical definitions. Moreover, continuous utilities naturally aggregate multiple objectives (e.g., latency and cost) into a unified scalar, allowing the router to jointly optimize

	Methods	Param./Non-param.§	Goals	Loss functions	Tasks*	Generalization†	Code
(a1)	[Shnitzer *et al.*, 2023]	Parametric	Performance	Binary CE loss	OE-G/EM-G	✗	-
	[Srivatsa *et al.*, 2024]	Parametric	Performance	Class-balanced CE loss	EM-G	✓	[Link]
	SelectLLM [Maurya *et al.*, 2024]	Parametric	Performance and cost	Class-balanced CE loss	EM-G	✗	-
	Routoo [Mohammadshahi *et al.*, 2024]	Parametric	Performance and cost	CE loss	EM-G	✗	-
	CSCR [Shirkavand *et al.*, 2026]	Parametric	Performance and cost	InfoNCE	OE-G/EM-G	✗	-
	RADAR [Fernandez *et al.*, 2025]	Parametric	Perf., cost and query time	Binary CE loss	EM-G	✗	-
	FORC [Šakota *et al.*, 2024]	Parametric	Performance and cost	Avg. of multi-metric integration	OE-G/EM-G	✓	[Link]
	[Ong *et al.*, 2025]	Parametric	Performance and cost	Binary CE loss	OE-G/EM-G	✓	[Link]
	Hybrid-LLM [Ding *et al.*, 2024a]	Parametric	Performance and cost	Binary CE loss	OE-G/EM-G	✗	[Link]
	P2L [Frick *et al.*, 2025]	Parametric	Performance and cost	Binary CE loss	OE-G/EM-G	✗	[Link]
	[Zhang *et al.*, 2025b]	Parametric	Performance and cost	Binary CE loss	OE-G/EM-G	✗	-
	Meta-Router [Zhang *et al.*, 2025d]	Non-parametric	Performance and cost	-	OE-G/EM-G	✗	
	Eagle [Zhao *et al.*, 2024]	Non-parametric	Performance and cost	-	OE-G/EM-G	✗	-
	Bench-CoE [Wang *et al.*, 2024]	Parametric	Performance	CE loss	EM-G	✓	[Link]
(a2)	[Li, 2025a]	Parametric	Performance and cost	MSE loss	OE-G/EM-G	✓	-
	MetaLLM [Nguyen *et al.*, 2024]	Parametric	Performance	MSE loss	EM-G	✗	[Link]
	MixLLM [Wang *et al.*, 2025]	Parametric	Perf., cost and query time	Negative Expected Reward	OE-G/EM-G	✗	-
	IRT-Router [Song *et al.*, 2025]	Parametric	Perf., cost and query time	Binary CE loss	OE-G/EM-G	✗	[Link]
	Avengers [Zhang *et al.*, 2025c]	Non-parametric	Performance	Clustering	OE-G/EM-G	✗	[Link]
	OmniRouter [Mei *et al.*, 2025]	Parametric	Performance and cost	MSE loss, Binary CE loss	EM-G	✗	[Link]
	[Li, 2025b]	Non-parametric	Performance and cost	-	OE-G/EM-G	✗	-
	[Wu and Silwal, 2025]	Non-parametric	Performance and cost	-	OE-G/EM-G	✗	[Link]
	TO-Router [Stripelis *et al.*, 2024]	Parametric	Perf., cost and query time	KL divergence	OE-G/EM-G	✓	-
	HomoRouter [Mu *et al.*, 2024]	Parametric	Performance and cost	MSE loss	EM-G	✓	-
	PickLLM [Sikeridis *et al.*, 2024]	Parametric	Perf., cost and query time	Negative Expected Reward	OE-G/EM-G	✓	-

[1] §: "Param./Non-param.§" indicates whether the router have learnable parameters.
[2] *: OE-G denotes Open-Ended Generation; EM-G denotes Exact-Match Generation, characterized by objectively verifiable answers (e.g., mathematical solutions).
[3] †: It denotes whether the router can generalize to new domains.

Table 1: Summary of ensemble-before-inference methods.

these objectives and learn more flexible policies. The most straightforward continuous-utility approach formulates routing as a utility prediction problem, where the router is often trained to predict the utility of each candidate model using regression-based losses [Song *et al.*, 2025; Zhang *et al.*, 2025c; Mei *et al.*, 2025; Li, 2025b; Wu and Silwal, 2025; Mu *et al.*, 2024]. For example, OmniRouter [Mei *et al.*, 2025] trains two MLP predictors to estimate performance and cost, respectively, and then considers batch query routing as a cost-constrained performance optimization problem, thereby achieving a trade-off between performance and cost. Beyond such explicit utility modeling, other methods instead directly learn a routing policy that outputs selection probabilities over candidate LLMs, treating utility as a feedback signal to optimize the policy [Nguyen *et al.*, 2024; Li, 2025a; Lu *et al.*, 2024b; Stripelis *et al.*, 2024; Sikeridis *et al.*, 2024]. For example, Li et al. [2025a] define the reward as a weighted sum of cost and performance, and adopt a multi-objective policy gradient algorithm to learn the selection policy.

3.2 Ensemble During Inference

We present a comprehensive review of *token-level*, *span-level*, and *process-level* ensemble methods in the following subsections and offer a summary analysis in Table 2.

(b1) Token-Level Ensemble

For these methods, at each decoding time-step, 1) *aggregation* approach produces the final token distribution for generation by (weighted) averaging probability distributions from multiple LLMs, whereas 2) *selection* approach opts to directly adopt the output token from a selected single model.

Aggregation-based normal methods. When attempting token-level aggregation, *vocabulary discrepancies* across different LLMs, which produce varying embedding lengths, impede the fusion and averaging of multiple probability distributions, creating a substantial obstacle. Thus, several methods focus on addressing the *vocabulary alignment* issue. ***(i)*** GaC [Yu *et al.*, 2024] is the most straightforward method, which constructs a "union dictionary" by combining the vocabularies of multiple models to include all tokens, and subsequently projects the distribution information onto this new merged dictionary for averaging aggregation; ***(ii)*** Further, DeePEn [Huang *et al.*, 2024] and EVA [Xu *et al.*, 2024] project the output distributions of multiple models into a shared *relative/pivot space*, followed by either an averaging aggregation [Huang *et al.*, 2024] or a weighted aggregation [Huang *et al.*, 2024; Xu *et al.*, 2024] to derive a final distribution. [2] Furthermore, there are additional advanced aggregation-based token-level ensemble methods. Specifically, ***(i)*** Yao et al. [2024] propose UniTe, which focuses solely on the TOP-K portion of each model's output distribution for performing averaging aggregation, reducing computational overhead; ***(ii)*** Mavromatis et al. [2024] present a *weighted* averaging approach, where the weights were determined by the *perplexity* of each model at each time step, indicating whether the current input aligns with its expertise.

Aggregation-based specific-goal methods. Unlike the aggregation-based normal methods introduced above, which are primarily intended to improve task performance in general scenarios, there are other methods that have their own specific goals. These methods aim to mitigate the prominent weaknesses of a single large LLM [Li *et al.*, 2024b] or achieve a balanced effects (including decoding-time realignment [Liu *et al.*, 2024b] and multi-objective decoding [Shi *et al.*, 2024]) by *weighted averaging* the token-level outputs from multiple LLMs with different characteristics. Among them, Li et al. [2024b] reduce negative issues during the deployment of LLMs, such as copyright infringement and data poisoning, by integrating the token-level outputs of a benign small model with a large model using weighted averaging aggregation.

[2] Beyond the aforementioned research focusing on vocabulary alignment, all other token-level aggregation-based ensemble methods either employ existing alignment techniques or assume that the LLMs under consideration share a common vocabulary.

	Methods	Granularities	Main Strategies	Unsup.	# Models†	Other Traits§	Efficiency-Aware	Tasks*	Code
(b1)	GaC [Yu *et al.*, 2024]	Token	Averaging agg.,Vocabulary align.	✓	≥ 2	Union dictionary	✗	OE-G/EM-G	[Link]
	DeePEn [Huang *et al.*, 2024]	Token	Aggregation,Vocabulary align.	✓	≥ 2	Relative representation	✗	OE-G/EM-G	[Link]
	EVA [Xu *et al.*, 2024]	Token	Averaging agg.,Vocabulary align.	✓	≥ 2	Relative representation	✗	OE-G/EM-G	[Link]
	UniTe [Yao *et al.*, 2024]	Token	Averaging agg.,TOP-K union	✓	≥ 2	Union dictionary	✓	OE-G/EM-G	-
	PackLLM [Mavromatis *et al.*, 2024]	Token	Weighted avg. agg.,Perplexity	✓	≥ 2	Greedy optimization	✗	OE-G/EM-G	[Link]
	[Li *et al.*, 2024b]	Token	Weighted-averaging agg.	✓	$= 2$	Reduce negative issues	✗	EM-G	-
	DeRa [Liu *et al.*, 2024b]	Token	Weighted-averaging agg.	✓	$= 2$	Decoding-time realignment	✗	OE-G/EM-G	[Link]
	MOD [Shi *et al.*, 2024]	Token	Weighted-averaging agg.	✓	≥ 2	Multi-objective decoding	✗	OE-G/EM-G	[Link]
	Copilot [Zou *et al.*, 2025]	Token	Weighted avg. agg.,Boosting	✗	$= 2$	Token-level boosting,SFT	✗	OE-G/EM-G	[Link]
	LLMBoost [Chen *et al.*, 2025d]	Token	Weighted avg. agg.,Boosting	✗	≥ 2	Token-level boosting,SFT	✗	OE-G/EM-G	-
	UltraFuser [Ding *et al.*, 2024b]	Token	Weighted avg. agg.,Gating	✗	≥ 2	Gating mechanism,SFT	✗	OE-G/EM-G	-
	CDS [Jin *et al.*, 2024]	Token	Token-Level routing (selection)	✗	$= 2$	Critical token classifier	✗	EM-G	-
	Co-Llm [Shen *et al.*, 2024]	Token	Token-Level routing (selection)	✗	$= 2$	Classifer,PGM	✗	OE-G/EM-G	[Link]
	CITER [Zheng *et al.*, 2025]	Token	Token-Level routing (selection)	✗	$= 2$	Reinforcement learning	✓	EM-G	[Link]
	ABE [Wicks *et al.*, 2025]	Token	Token-Level routing (selection)	✓	$= 2$	Agreement-based ensembling	✗	OE-G/EM-G	[Link]
(b2)	Cool-Fusion [Liu *et al.*, 2025]	Span	Generation-assessment-selection	✓	≥ 2	Perplexity, common words	✗	OE-G/EM-G	-
	SweetSpan [Xu *et al.*, 2025]	Span	Generation-assessment-selection	✓	≥ 2	Perplexity, fixed-length	✗	OE-G/EM-G	-
	SpecFuse [Lv *et al.*, 2024a]	Span	Generation-assessment-selection	✓	≥ 2	Perplexity, fixed-length	✓	OE-G/EM-G	-
	CoS [Fu *et al.*, 2025]	Span	Collaborative decoding	✓	≥ 2	Speculation decoding	✓	OE-G/EM-G	[Link]
(b3)	LE-MCTS [Park *et al.*, 2025]	Process	Reasoning process selection	✗	≥ 2	Monte Carlo Tree Search	✗	OE-G/EM-G	-

[1] †: It denotes to the number of models that can be used in the ensemble.
[2] §: "Other Traits§" denotes unique and relatively important capabilities, modules or characteristics that are noteworthy.
[3] *: OE-G denotes Open-Ended Generation; EM-G denotes Exact-Match Generation, characterized by objectively verifiable answers (e.g., mathematical solutions).

Table 2: Summary of ensemble-during-inference methods.

Aggregation-based fine-tuning methods. Additionally, there are methods such as Copilot [Zou *et al.*, 2025], LLM-Boost [Chen *et al.*, 2025d] and UltraFuser [Ding *et al.*, 2024b], which consider task-downstream supervised fine tuning. These methods use labeled data to fine-tune multiple pre-trained LLMs based on the Ensemble Learning paradigm [Zhou, 2021], i.e., boosting [Zou *et al.*, 2025; Chen *et al.*, 2025d] and MOE (Mixture-of-Experts) [Ding *et al.*, 2024b]. Finally, they perform token-level weighted averaging ensemble during the inference phase.

Selection-based methods. These methods directly choose a token from a specific model's output. ***(i)*** Jin et al. [Jin *et al.*, 2024] propose CDS, which trains a binary classifier based on the preceding context. This classifier determines whether to invoke an unaligned model (to leverage its factual knowledge) or an aligned model (to utilize its instruction-following and safety capabilities) at each decoding step; ***(ii)*** Similar to CDS, Co-Llm [Shen *et al.*, 2024] employs a routing classifier to switch between a general-purpose model and a specialized model for optimal output; ***(iii)*** Similarly, Zheng et al. [2025] introduce CITER, which uses reinforcement learning to train a policy function; ***(iv)*** Wicks et al. [Wicks *et al.*, 2025] use an approach they termed "agreement-based ensembling" to select the intersection of various token information.

(b2) Span-Level Ensemble

(i) These methods mostly adhere to a "generation-assessment-selection" pipeline. These methods first employ multiple LLMs to generate word-containing fragments, then utilize the concept of *perplexity*, to make each model score all generated model responses, ultimately selecting the fragment with the highest cumulative score. The key distinction lies in that Cool-Fusion [Liu *et al.*, 2025] have each source LLM individually generate text segments until the word boundary of each segment is common to all LLMs, whereas SweetSpan [Xu *et al.*, 2025] and SpecFuse [Lv *et al.*, 2024a] adopt a straightforward approach by setting the span as a fragment with a fixed word count; ***(ii)*** Further, Fu et al. [2025] incorporate *speculative decoding*, a well-known acceleration technique, into the ensemble process to achieve faster inference.

(b3) Process-Level Ensemble

Confronted with complex step-by-step reasoning tasks, Park et al. [2025] use a trained Monte Carlo Tree Search strategy at each reasoning step to select the output with the highest reward value from multiple model reasoning outputs.

3.3 Ensemble After Inference

We review the *non-cascade* and *cascade* methods in the following, and provide a summary analysis in Table 3.

(c1) Non-Cascade

As shown in Figure 2 and Table 3, these methods include: 1) *selection-based*, which is dedicated to selecting a single response from multiple candidates; 2) *selection-then-regeneration*, which involves initially selecting a subset of candidates and subsequently feeding this refined subset into a generative model for regeneration to obtain the final output.

Selection-based methods. First, as *unsupervised* methods, **1)** Agent-Forest [Li *et al.*, 2024a] and Smoothie [Guha *et al.*, 2024] leverage the similarity between model responses for selection, adhering to a majority voting (MV) principle: *selecting the response most similar to others*. Specifically, ***(i)*** Li et al. [2024a] obtain multiple responses by repeatedly querying a single model (equivalent to using multiple homogeneous models) and employ the MV ensemble, discovering a scaling property when ensembling more responses; they use BLEU scores and occurrence frequencies to quantify similarities for open-ended and exact-match generative tasks, respectively; ***(ii)*** Guha et al. [2024] input multiple <query, response> pairs into SentenceBERT, leveraging the Euclidean distance between the resulting low-dimensional encoded vectors as a measure of similarity; **2)** Unlike these similarity-based MV ensemble methods, Chen et al. [2025e] propose LLM-PeerReview, an academic peer-review-inspired framework that leverages LLM-as-a-Judge to score response quality, selecting the one with the highest aggregated score.

Further, as a *supervised* method, MoRE [Si *et al.*, 2023] utilizes some training data to train a random forest classifier for selecting among multiple responses, using the response similarity among LLMs as one of the constructed features.

	Methods	Un-/Supervised	Main Strategies	# Models	Other Traits[§]	Efficiency-aware	Tasks[*]	Code
(c1)	Agent-Forest [Li *et al.*, 2024a]	Unsupervised	Similarity-based selection	> 2	Scaling property	✗	OE-G/EM-G	[Link]
	Smoothie [Guha *et al.*, 2024]	Unsupervised	Similarity-based selection	> 2	Graphical model	✗	OE-G/EM-G	[Link]
	LLM-PeerReview [Chen *et al.*, 2025e]	Unsupervised	PeerReview framework	> 2	Probabilistic Graphical Model	✗	OE-G/EM-G	[Link]
	MoRE [Si *et al.*, 2023]	Supervised	Supervised Sim.-based sel.	> 2	Feature engineering	✗	EM-G	[Link]
	LLM-Blender [Jiang *et al.*, 2023]	Supervised	Selection-then-regeneration	> 2	Pairwise ranking	✗	OE-G/EM-G	[Link]
	LLM-TOPLA [Tekin *et al.*, 2024]	Supervised	Selection-then-regeneration	> 2	Maximising diversity	✗	OE-G/EM-G	[Link]
	URG [Lv *et al.*, 2024b]	Supervised	Selection-then-regeneration	> 2	Unified processes	✓	OE-G/EM-G	-
(c2)	EcoAssistant [Zhang *et al.*, 2023]	Unsupervised	User judgment	≥ 2	Prompt engineering	✓	OE-G (code generation)	[Link]
	[Yue *et al.*, 2024]	Unsupervised	Answer consistency	$\geq 2^{\dagger}$	Sampling and checking	✓	EM-G	[Link]
	Model Cascading [Varshney and Baral, 2022]	Unsupervised	Class uncertainty	≥ 2	Cascade-pioneer	✓	EM-G	-
	neural caching [Ramírez *et al.*, 2023]	Unsupervised	Class unc./Ans. cons.	$= 2$	Distillation	✓	EM-G	[Link]
	Cascade Routing [Dekoninck *et al.*, 2024]	Unsupervised	Class uncertainty	> 2	Routing-equipped	✓	EM-G	[Link]
	[Jitkrittum *et al.*, 2023]	Supervised	(Upgraded) Class unc.	$\geq 2^{\dagger}$	Post-hoc deferral	✓	EM-G	-
	FrugalGPT [Chen *et al.*, 2025a]	Supervised	Score function	≥ 2	Prompt engineering, etc.	✓	EM-G	-
	[Gupta *et al.*, 2024]	Supervised	Score function	$= 2$	Quantile features	✓	OE-G/EM-G	-
	AutoMix [Aggarwal *et al.*, 2024]	Supervised	MDP	≥ 2	Routing-equipped, etc.	✓	EM-G	-
	DER [Hu *et al.*, 2024a]	Supervised	MDP,Ans. Cons.	≥ 2	Routing-equipped	✓	EM-G	-

[1] §: "Other Traits[§]" denotes unique and relatively important capabilities, modules or characteristics that are noteworthy.
[2] *: OE-G denotes Open-Ended Generation; EM-G denotes Exact-Match Generation, characterized by objectively verifiable answers (e.g., mathematical solutions).
[3] †: "$\geq 2^{\dagger}$" denotes that the study considers a 2-models cascade scenario in its methodology introduction/derivation, but it can be easily adapted to K-models cascade scenarios.

Table 3: Summary of ensemble-after-inference methods.

Selection-then-regeneration methods. These methods stem from the pioneering work LLM-Blender, introduced by Jiang et al. [2023]. ***(i)*** LLM-Blender uses some training data to train the "PairRanker" selection module in the first phase, serving to select a subset from multiple responses, and to train the "GenFuser" generator in the subsequent phase for synthesizing the final response; ***(ii)*** Further, the improvement made by LLM-TOPLA [Tekin *et al.*, 2024] mainly lies in optimizing the selection process in the first step by "maximizing the diversity among multiple response candidates"; ***(iii)*** URG [Lv *et al.*, 2024b] proposes an end-to-end framework that integrates selection and regeneration.

(c2) Cascade

Methodologically, all cascade methods revolve around the *deferral rule/decision maker*: determining whether to adopt the output of the current model at hand or to invoke a subsequent, more powerful model.

Unsupervised methods. As shown in Table 3, these methods can be further categorized into three types according to the pivotal "main strategy": *user judgment* [Zhang *et al.*, 2023], *answer consistency* [Yue *et al.*, 2024; Ramírez *et al.*, 2023], and *class uncertainty* [Varshney and Baral, 2022; Ramírez *et al.*, 2023; Dekoninck *et al.*, 2024].

As one most straightforward method, Zhang et al. [2023] propose a cascade method, EcoAssistant, that utilizes *user judgment* rather than *machine algorithms* to determine whether the cascading inference should be terminated. Further, Yue et al. [2024] introduce a method based on *answer consistency*—more precisely, on the observation of whether the inference results generated from multiple identical/diverse prompts exhibit sufficient uniformity—to solve the cascade problem for in-context-learning reasoning tasks.

Last but certainly not least, a critically important category of approaches is grounded in *class uncertainty*——an approach that also manifests in the following "supervised methods". Class uncertainty, also known as *confidence*, can be implemented through various variants, including Maximum Softmax Probability [Varshney and Baral, 2022], Distance To Uniform Distribution [Varshney and Baral, 2022], Margin Sampling [Ramírez *et al.*, 2023], and Prediction Entropy [Ramírez *et al.*, 2023]. However, at its core, it evaluates whether the probability of the dominant class inferred by the current model exceeds a sufficient threshold. When this probability is sufficiently high—signifying adequate confidence—the cascading process is concluded; otherwise, it proceeds.

Supervised methods. Unlike the aforementioned unsupervised cascade methods, supervised cascade methods use some training data to train cascade-related modules, including *post-hoc deferral based on upgraded class-uncertainty strategy* [Jitkrittum *et al.*, 2023], *scoring functions* [Chen *et al.*, 2025a; Gupta *et al.*, 2024], and *Markov Decision Processes (MDPs)* [Aggarwal *et al.*, 2024; Hu *et al.*, 2024a]. Specifically, ***(i)*** Jitkrittum et al. [2023] use the training data to learn a cascade method called *post-hoc deferral*; and the core idea is that, in contrast to aforementioned *class-uncertainty* cascade methods which solely focus on the class uncertainty of the current model's output, it also takes into account the estimated uncertainty of the next, stronger model's output; ***(ii)*** Further, Chen et al. [2025a] and Gupta et al. [2024] use the training data to learn a *scoring function* that produces a confidence score regarding the current model's output, enabling cascading judgments; ***(iii)*** Both Aggarwal et al. [2024] and Hu et al. [2024a] adopt theidea of "introducing routing in the cascading process" and train an MDP for decision-making.

4 Benchmarks and Applications

Here we briefly discuss popular benchmarks and applications in LLM Ensemble. First, there are two types of benchmarks specifically tailored for LLM Ensemble evaluation. 1) Benchmark MIXINSTRUCT proposed by Jiang et al. [2023], which serves to assess the *performance* of ensemble-after-inference methods; 2) In contrast, most later benchmarks target ensemble-before-inference, i.e., LLM routing: RouterEval [Huang *et al.*, 2025] is mainly performance-oriented, RouterBench [Hu *et al.*, 2024b], FusionFactory [Feng *et al.*, 2025], RouterArena [Lu *et al.*, 2025] and LLMRouterBench [Li *et al.*, 2026a] further evaluate performance–cost. In terms of applications, Lee et al. [2023] leverage the ROUGE-L metric and employed a similarity-based selection strategy for ensemble to produce Instruction-Tuning data; additionally, several studies focus on tabular data imputation [He *et al.*, 2025], tool routing for RAG systems [Mu *et al.*, 2024], and so on.

	Approaches	Ensemble Strategies	Ensemble Granularities	Ensemble Goals
(a) Ensemble before inference	(a1) Discrete utility methods	Selection	Response-level ♣	Performance (and cost)
	(a2) Continuous utility methods	Selection	Response-level ♣	Performance and cost
(b) Ensemble during inference	(b1) Token-level ensemble	Aggregation, Selection	Token-level ♣ ♣ ♣	Performance[§]
	(b2) Span-level ensemble	Selection	Span-level ♣ ♣	Performance
	(b3) Process-level ensemble	Selection	Process-level ♣ ♣	Performance
(c) Ensemble after inference	(c1) Non cascade	Selection, Regeneration	Response-level ♣	Performance
	(c2) Cascade	Selection	Response-level ♣	Performance and cost

[1] ♣: It represents the level of granularity, and a higher quantity indicates finer ensemble granularity.
[2] [§]: As a specific example, Li et al. [2024b] primarily aim to minimize the negative issues of large models during deployment.

Table 4: Summary analysis of the key attributes of LLM Ensemble approaches.

5 Discussion

5.1 Summarization

Here we provide a summary analysis of all LLM Ensemble approaches, as illustrated in Table 4, considering the most critical methodological attributes: *ensemble strategy*, *granularity* and *ensemble goal*. From the perspective of ensemble strategy, *aggregation* methods (involving the average or weighted ensemble of all model outputs) are more sophisticated compared to *selection*-based methods, which select a single output (equivalent to a *hard voting* process). However, *regeneration*-based methods are burdened by the additional need for large model-specific training data preparation and model training. From the perspective of ensemble granularity, it is apparent that *response*-level ensemble methods are relatively coarse-grained; while methods with finer granularity, particularly token-level ensemble methods, can more effectively harness the distribution information from each model during the decoding phase. Further, from the perspective of ensemble goals, (b) ensemble-during-inference methods and (c1) non-cascaded-based methods, unconstrained by cost considerations, can employ more flexible ensemble strategies beyond *selection* and utilize finer-grained ensemble approaches, yielding greater performance potential.

5.2 Existing Limitations and Future Directions

Summary on Future Directions

- 1) Label-efficient un-/weakly-supervised model profiling for ensemble-before-inference methods.
- 2) Efficient adaptation under evolving model distributions for ensemble-before-inference.
- 3) Principled span-level ensemble-during-inference approach.
- 4) Sophisticated unsupervised cascade ensemble-after-inference approach for open-ended generation.
- 5) Enable active exploration by multiple LLM agents during reasoning, beyond fixed ensemble methods.

Direction 1. Existing ensemble-before-inference methods typically require costly supervised data to estimate domain competence. However, a model's internal states often inherently encode signals about its reliability. A promising direction is developing un-/weakly-supervised profiling strategies that infer model strengths directly from intrinsic signals (e.g., prediction uncertainty, self-consistency, or internal representations), enabling low-cost profiling without extensive domain-specific annotations.

Direction 2. Existing ensemble-before-inference methods are typically evaluated in closed and static settings with a fixed model pool. However, real-world deployments are dynamic: the set of available models and their capabilities continuously evolve, leading to shifts in the candidate model distribution. This calls for routing strategies that can *rapidly and data-efficiently* adapt to evolving model distributions, enabling low-cost few-shot recalibration when models are added, removed, or updated.

Direction 3. Span-level ensemble-during-inference methods already offer sufficient granularity and show higher performance potential compared to response-level methods. However, current segment segmentation techniques are still overly simplistic (e.g., rigidly defining segment lengths as a 4-word sequence). Shifting to a more principled segment segmentation approach could provide richer, more valuable information for the subsequent autoregressive ensemble.

Direction 4. Compared to ensemble-before-inference, cascade methods can address cost constraints while utilizing intermediate responses. However, they often lack generalizability, struggling with open-ended tasks or relying on supervised data. Thus, developing general unsupervised cascades is highly significant.

Direction 5. Current ensemble approaches lack sufficient intrinsic *agent exploration* [Ban *et al.*, 2026] and cross-paradigm integration; future work should allow multi-agent iterative reasoning to dynamically select sub-models and ensemble strategies per task.

6 Conclusion

LLM Ensemble stands as a direct manifestation of ensemble learning in the era of large language models. The accessibility, diversity, and out-of-the-box usability of large language models have made the spirit of ensemble learning shine even brighter, fueled by the rapidly developing LLM Ensemble studies. This paper provides a comprehensive taxonomy and a review of the methods in the field of LLM Ensemble, introduces relevant applications and benchmarks, conducts a summative analysis of these methods, and proposes several potential research directions. We hope that this review will offer valuable insights to researchers and inspire further advancements in LLM Ensemble and related research areas.

Acknowledgments

This work was supported by the National Key Research and Development Program of China (Grant No. 2024YFB3309602) and the National Natural Science Foundation of China (Grant Nos. 62472017 and 62506024).

References

[Achiam *et al.*, 2023] Josh Achiam, Steven Adler, et al. Gpt-4 technical report. *ArXiv preprint*, abs/2303.08774, 2023.

[Aggarwal *et al.*, 2024] Pranjal Aggarwal, Aman Madaan, et al. Automix: Automatically mixing language models. In *NeurIPS*, 2024.

[Ban *et al.*, 2026] Yikun Ban, Fengkai Yang, et al. Agent exploration toward artificial general intelligence. *SSRN preprint SSRN:6748619*, 2026.

[Chen *et al.*, 2020] Zhijun Chen, Huimin Wang, et al. Structured probabilistic end-to-end learning from crowds. In *IJCAI*, pages 1512–1518. ijcai.org, 2020.

[Chen *et al.*, 2022] Pengpeng Chen, Hailong Sun, et al. Adversarial learning from crowds. In *AAAI*, 2022.

[Chen *et al.*, 2023a] Pengpeng Chen, Yongqiang Yang, et al. Black-box data poisoning attacks on crowdsourcing. In *IJCAI*, pages 2975–2983. ijcai.org, 2023.

[Chen *et al.*, 2023b] Zhijun Chen, Hailong Sun, et al. Neural-hidden-crf: A robust weakly-supervised sequence labeler. In *SIGKDD*, pages 274–285. ACM, 2023.

[Chen *et al.*, 2025a] L. Chen, M. Zaharia, et al. Frugalgpt: How to use large language models while reducing cost and improving performance. *TMLR*, 2025.

[Chen *et al.*, 2025b] Yi Chen, JiaHao Zhao, et al. A survey on collaborative mechanisms between large and small language models. *ArXiv preprint*, abs/2505.07460, 2025.

[Chen *et al.*, 2025c] Z. Chen, Z. Wei, et al. Tagrouter: Learning route to llms through tags for open-domain text generation tasks. *ArXiv preprint*, abs/2506.12473, 2025.

[Chen *et al.*, 2025d] Zehao Chen, Tianxiang Ai, et al. Llm-boost: Make large language models stronger with boosting. *ArXiv preprint*, abs/2512.22309, 2025.

[Chen *et al.*, 2025e] Zhijun Chen, Zeyu Ji, et al. Scoring, reasoning, and selecting the best! ensembling large language models via a peer-review process. *ArXiv preprint*, abs/2512.23213, 2025.

[Dekoninck *et al.*, 2024] Jasper Dekoninck, Maximilian Baader, et al. A unified approach to routing and cascading for llms. *ArXiv preprint*, abs/2410.10347, 2024.

[Ding *et al.*, 2024a] D. Ding, A. Mallick, et al. Hybrid LLM: cost-efficient and quality-aware query routing. In *ICLR*, 2024.

[Ding *et al.*, 2024b] N. Ding, Y. Chen, et al. Mastering text, code and math simultaneously via fusing highly specialized language models. *ArXiv*, abs/2403.08281, 2024.

[Dong *et al.*, 2020] Xibin Dong, Zhiwen Yu, et al. A survey on ensemble learning. *Frontiers of Computer Science*, 14(2):241–258, 2020.

[Du *et al.*, 2024] Yilun Du, Shuang Li, et al. Improving factuality and reasoning in language models through multiagent debate. In *ICML*. OpenReview.net, 2024.

[Feng *et al.*, 2025] Tao Feng, Haozhen Zhang, et al. Fusionfactory: Fusing llm capabilities with multi-llm log data. *arXiv preprint arXiv:2507.10540*, 2025.

[Fernandez *et al.*, 2025] Nigel Fernandez, Branislav Kveton, et al. Radar: Reasoning-ability and difficulty-aware routing for reasoning llms. *ArXiv*, abs/2509.25426, 2025.

[Frick *et al.*, 2025] Evan Frick, Connor Chen, et al. Prompt-to-leaderboard. *ArXiv preprint*, abs/2502.14855, 2025.

[Fu *et al.*, 2025] Jiale Fu, Yuchu Jiang, et al. Fast large language model collaborative decoding via speculation. *ArXiv preprint*, abs/2502.01662, 2025.

[Guha *et al.*, 2024] N. Guha, M. F. Chen, et al. Smoothie: Label free language model routing. In *NeurIPS*, 2024.

[Guo *et al.*, 2024] Taicheng Guo, Xiuying Chen, et al. Large language model based multi-agents: a survey of progress and challenges. In *IJCAI*, 2024.

[Gupta *et al.*, 2024] Neha Gupta, Harikrishna Narasimhan, et al. Language model cascades: Token-level uncertainty and beyond. In *ICLR*, 2024.

[He *et al.*, 2025] Xinrui He, Yikun Ban, et al. Llm-forest: Ensemble learning of llms with graph-augmented prompts for data imputation. In *Findings of ACL*, 2025.

[Hu *et al.*, 2024a] J. Hu, Y. Wang, et al. Dynamic ensemble reasoning for llm experts. *ArXiv*, abs/2412.07448, 2024.

[Hu *et al.*, 2024b] Qitian Jason Hu, Jacob Bieker, et al. Routerbench: A benchmark for multi-llm routing system. *ArXiv preprint*, abs/2403.12031, 2024.

[Huang *et al.*, 2024] Yichong Huang, Xiaocheng Feng, et al. Ensemble learning for heterogeneous large language models with deep parallel collaboration. In *NeurIPS*, 2024.

[Huang *et al.*, 2025] Z. Huang, G. Ling, et al. Routereval: A comprehensive benchmark for routing llms to explore model-level scaling up in llms. *ArXiv preprint*, abs/2503.10657, 2025.

[Jiang *et al.*, 2023] Dongfu Jiang, Xiang Ren, et al. LLM-blender: Ensembling large language models with pairwise ranking and generative fusion. In *ACL*, 2023.

[Jin *et al.*, 2024] Lifeng Jin, Baolin Peng, et al. Collaborative decoding of critical tokens for boosting factuality of large language models. *ArXiv preprint*, abs/2402.17982, 2024.

[Jin *et al.*, 2025] Weiqiang Jin, Hongyang Du, et al. A comprehensive survey on multi-agent cooperative decision-making: Scenarios, approaches, challenges and perspectives. *arXiv preprint arXiv:2503.13415*, 2025.

[Jitkrittum *et al.*, 2023] Wittawat Jitkrittum, Neha Gupta, et al. When does confidence-based cascade deferral suffice? In *NeurIPS*, 2023.

[Lee *et al.*, 2023] Young-Suk Lee, Md Arafat Sultan, et al. Ensemble-instruct: Generating instruction-tuning data with a heterogeneous mixture of lms. *ArXiv preprint*, abs/2310.13961, 2023.

[Li *et al.*, 2023] Weishi Li, Yong Peng, et al. Deep model fusion: A survey. *ArXiv preprint*, abs/2309.15698, 2023.

[Li *et al.*, 2024a] Junyou Li, Qin Zhang, et al. More agents is all you need. *ArXiv preprint*, abs/2402.05120, 2024.

[Li *et al.*, 2024b] Tianlin Li, Qian Liu, et al. Purifying large language models by ensembling a small language model. *ArXiv preprint*, abs/2402.14845, 2024.

[Li *et al.*, 2024c] Xinyi Li, Sai Wang, et al. A survey on llm-based multi-agent systems: workflow, infrastructure, and challenges. *Vicinagearth*, 2024.

[Li *et al.*, 2026a] Hao Li, Yiqun Zhang, et al. Llmrouterbench: A massive benchmark and unified framework for llm routing. *arXiv preprint arXiv:2601.07206*, 2026.

[Li *et al.*, 2026b] Zhongyi Li, Wan Tian, et al. Adaptive robust estimator for multi-agent reinforcement learning. *arXiv preprint arXiv:2603.21574*, 2026.

[Li, 2025a] Yang Li. Llm bandit: Cost-efficient llm generation via preference-conditioned dynamic routing. *ArXiv preprint*, abs/2502.02743, 2025.

[Li, 2025b] Yang Li. Rethinking predictive modeling for llm routing: When simple knn beats complex learned routers. *ArXiv preprint*, abs/2505.12601, 2025.

[Liu *et al.*, 2024a] Aixin Liu, Bei Feng, et al. Deepseek-v2: A strong, economical, and efficient mixture-of-experts language model. *ArXiv preprint*, abs/2405.04434, 2024.

[Liu *et al.*, 2024b] T. Liu, S. Guo, et al. Decoding-time realignment of language models. In *ICML*, 2024.

[Liu *et al.*, 2025] C. Liu, X. Quan, et al. Cool-fusion: Fuse large language models without training. In *ACL*, 2025.

[Lu *et al.*, 2024a] Jinliang Lu, Ziliang Pang, et al. Merge, ensemble, and cooperate! a survey on collaborative strategies in the era of large language models. *ArXiv preprint*, abs/2407.06089, 2024.

[Lu *et al.*, 2024b] Keming Lu, Hongyi Yuan, et al. Routing to the expert: Efficient reward-guided ensemble of large language models. In *NAACL*, 2024.

[Lu *et al.*, 2025] Yifan Lu, Rixin Liu, et al. Routerarena: An open platform for comprehensive comparison of llm routers. *arXiv preprint arXiv:2510.00202*, 2025.

[Lv *et al.*, 2024a] Bo Lv, Chen Tang, et al. Specfuse: Ensembling large language models via next-segment prediction. *ArXiv preprint*, abs/2412.07380, 2024.

[Lv *et al.*, 2024b] Bo Lv, Chen Tang, et al. Urg: A unified ranking and generation method for ensembling language models. In *Findings of the ACL*, 2024.

[Maurya *et al.*, 2024] Kaushal Kumar Maurya, KV Srivatsa, et al. Selectllm: Query-aware efficient selection algorithm for large language models. *ArXiv preprint*, abs/2408.08545, 2024.

[Mavromatis *et al.*, 2024] Costas Mavromatis, Petros Karypis, et al. Pack of llms: Model fusion at test-time via perplexity optimization. *ArXiv*, abs/2404.11531, 2024.

[Mei *et al.*, 2025] Kai Mei, Wujiang Xu, et al. Omnirouter: Budget and performance controllable multi-llm routing. *ACM SIGKDD Explorations Newsletter*, 2025.

[Mohammadshahi *et al.*, 2024] Alireza Mohammadshahi, Arshad Rafiq Shaikh, et al. Routoo: Learning to route to large language models effectively. *ArXiv preprint*, abs/2401.13979, 2024.

[Mohammed and Kora, 2023] A. Mohammed and R. Kora. A comprehensive review on ensemble deep learning: Opportunities and challenges. *Journal of King Saud University-Computer and Information Sciences*, 2023.

[Mu *et al.*, 2024] Feiteng Mu, Yong Jiang, et al. Adaptive selection for homogeneous tools: An instantiation in the rag scenario. *ArXiv preprint*, abs/2406.12429, 2024.

[Nguyen *et al.*, 2024] Quang H Nguyen, Duy C Hoang, et al. Metallm: A high-performant and cost-efficient dynamic framework for wrapping llms. *ArXiv preprint*, abs/2407.10834, 2024.

[Ong *et al.*, 2025] Isaac Ong, Amjad Almahairi, et al. Routellm: Learning to route llms from preference data. In *ICLR*, 2025.

[Park *et al.*, 2025] Sungjin Park, Xiao Liu, et al. Ensembling large language models with process reward-guided tree search for better complex reasoning. In *NAACL*, 2025.

[Ramírez *et al.*, 2023] Guillem Ramírez, Matthias Lindemann, et al. Cache & distil: Optimising api calls to large language models. *ArXiv preprint*, abs/2310.13561, 2023.

[Ratner *et al.*, 2017] Alexander Ratner, Stephen H Bach, et al. Snorkel: Rapid training data creation with weak supervision. In *VLDB*, 2017.

[Ruan *et al.*, 2025] Wei Ruan, Tianze Yang, et al. From task-specific models to unified systems: A review of model merging approaches. *ArXiv*, abs/2503.08998, 2025.

[Šakota *et al.*, 2024] Marija Šakota, Maxime Peyrard, et al. Fly-swat or cannon? cost-effective language model choice via meta-modeling. In *WSDM*, pages 606–615, 2024.

[Shen *et al.*, 2024] Shannon Zejiang Shen, Hunter Lang, et al. Learning to decode collaboratively with multiple language models. *ArXiv preprint*, abs/2403.03870, 2024.

[Shi *et al.*, 2024] Ruizhe Shi, Yifang Chen, et al. Decoding-time language model alignment with multiple objectives. In *NeurIPS*, 2024.

[Shirkavand *et al.*, 2026] Reza Shirkavand, Shangqian Gao, Peiran Yu, and Heng Huang. Cost-aware contrastive routing for llms. *NeurIPS*, 2026.

[Shnitzer *et al.*, 2023] Tal Shnitzer, Anthony Ou, et al. Large language model routing with benchmark datasets. In *NeurIPS*, 2023.

[Si *et al.*, 2023] Chenglei Si, Weijia Shi, et al. Getting MoRE out of mixture of language model reasoning experts. In *Findings of EMNLP*, 2023.

[Sikeridis *et al.*, 2024] Dimitrios Sikeridis, Dennis Ramdass, et al. Pickllm: Context-aware rl-assisted large language model routing. *ArXiv preprint*, abs/2412.12170, 2024.

[Song *et al.*, 2025] Wei Song, Zhenya Huang, et al. Irt-router: Effective and interpretable multi-llm routing via item response theory. *ArXiv*, abs/2506.01048, 2025.

[Srivatsa *et al.*, 2024] Kv Aditya Srivatsa, Kaushal Maurya, et al. Harnessing the power of multiple minds: Lessons learned from LLM routing. In *Workshop on Insights from Negative Results in NLP*, 2024.

[Stripelis *et al.*, 2024] Dimitris Stripelis, Zijian Hu, et al. Tensoropera router: A multi-model router for efficient llm inference. *ArXiv preprint*, abs/2408.12320, 2024.

[Sun *et al.*, 2024] Chuanneng Sun, Songjun Huang, et al. Llm-based multi-agent reinforcement learning: Current and future directions. *arXiv arXiv:2405.11106*, 2024.

[Team *et al.*, 2023] Gemini Team, Rohan Anil, et al. Gemini: a family of highly capable multimodal models. *ArXiv preprint*, abs/2312.11805, 2023.

[Tekin *et al.*, 2024] Selim Tekin, Fatih Ilhan, et al. Llm-topla: Efficient llm ensemble by maximising diversity. In *Findings of EMNLP*, 2024.

[Touvron *et al.*, 2023] Hugo Touvron, Thibaut Lavril, et al. Llama: Open and efficient foundation language models. *ArXiv preprint*, abs/2302.13971, 2023.

[Tran *et al.*, 2025] Khanh-Tung Tran, Dung Dao, et al. Multi-agent collaboration mechanisms: A survey of llms. *ArXiv preprint*, abs/2501.06322, 2025.

[Varshney and Baral, 2022] Neeraj Varshney and Chitta Baral. Model cascading: Towards jointly improving efficiency and accuracy of NLP systems. In *EMNLP*, 2022.

[Wang *et al.*, 2024] Yuanshuai Wang, Xingjian Zhang, et al. Bench-coe: a framework for collaboration of experts from benchmark. *ArXiv preprint*, abs/2412.04167, 2024.

[Wang *et al.*, 2025] Xinyuan Wang, Yanchi Liu, et al. Mixllm: Dynamic routing in mixed large language models. *ArXiv preprint*, abs/2502.18482, 2025.

[Wicks *et al.*, 2025] Rachel Wicks, Kartik Ravisankar, et al. Token-level ensembling of models with different vocabularies. *ArXiv preprint*, abs/2502.21265, 2025.

[Wu and Silwal, 2025] Fangzhou Wu and Sandeep Silwal. Efficient training-free online routing for high-volume multi-llm serving. *ArXiv preprint*, abs/2509.02718, 2025.

[Xu *et al.*, 2024] Yangyifan Xu, Jinliang Lu, et al. Bridging the gap between different vocabularies for LLM ensemble. In *NAACL*, 2024.

[Xu *et al.*, 2025] Yangyifan Xu, Jianghao Chen, et al. Hit the sweet spot! span-level ensemble for large language models. In *COLING*, pages 8314–8325, 2025.

[Yang *et al.*, 2024] Enneng Yang, Li S., et al. Model merging in llms, mllms, and beyond: Methods, theories, applications and opportunities. *ArXiv*, abs/2408.07666, 2024.

[Yang *et al.*, 2025] Zongzhen Yang, Binhang Qi, et al. Cabs: Conflict-aware and balanced sparsification for enhancing model merging. *ArXiv preprint*, abs/2503.01874, 2025.

[Yao *et al.*, 2024] Yuxuan Yao, Han Wu, et al. Determine-then-ensemble: Necessity of top-k union for large language model ensembling. *ArXiv*, abs/2410.03777, 2024.

[Yu *et al.*, 2024] Yao-Ching Yu, Chun-Chih Kuo, et al. Breaking the ceiling of the llm community by treating token generation as a classification for ensembling. *ArXiv preprint*, abs/2406.12585, 2024.

[Yue *et al.*, 2024] Murong Yue, Jie Zhao, et al. Large language model cascades with mixture of thought representations for cost-efficient reasoning. In *ICLR*, 2024.

[Zhang *et al.*, 2021] Jieyu Zhang, Yue Yu, et al. Wrench: A comprehensive benchmark for weak supervision. *ArXiv preprint*, abs/2109.11377, 2021.

[Zhang *et al.*, 2022] Jieyu Zhang, Cheng-Yu Hsieh, et al. A survey on programmatic weak supervision. *ArXiv preprint*, abs/2202.05433, 2022.

[Zhang *et al.*, 2023] Jieyu Zhang, Ranjay Krishna, et al. Ecoassistant: Using llm assistant more affordably and accurately. *ArXiv preprint*, abs/2310.03046, 2023.

[Zhang *et al.*, 2025a] Hangfan Zhang, Zhiyao Cui, et al. If multi-agent debate is the answer, what is the question. *ArXiv preprint*, abs/2502.08788, 2025.

[Zhang *et al.*, 2025b] Tuo Zhang, Asal Mehradfar, et al. Leveraging uncertainty estimation for efficient llm routing. *ArXiv preprint*, abs/2502.11021, 2025.

[Zhang *et al.*, 2025c] Y. Zhang, H Li, et al. The avengers: A simple recipe for uniting smaller language models to challenge proprietary giants. *ArXiv*, abs/2505.19797, 2025.

[Zhang *et al.*, 2025d] Y. Zhang, F. Xie, et al. Meta-router: Bridging gold-standard and preference-based evaluations in large language model routing. *ArXiv preprint*, abs/2509.25535, 2025.

[Zhang, 2022] Jing Zhang. Knowledge learning with crowdsourcing: A brief review and systematic perspective. *IEEE/CAA Journal of Automatica Sinica*, 2022.

[Zhao *et al.*, 2024] Zesen Zhao, Shuowei Jin, et al. Eagle: Efficient training-free router for multi-llm inference. *ArXiv preprint*, abs/2409.15518, 2024.

[Zheng *et al.*, 2017] Yudian Zheng, Guoliang Li, et al. Truth inference in crowdsourcing: Is the problem solved? *Proceedings of the VLDB Endowment*, 10(5):541–552, 2017.

[Zheng *et al.*, 2025] Wenhao Zheng, Yixiao Chen, et al. Citer: Collaborative inference for efficient large language model decoding with token-level routing. *ArXiv preprint*, abs/2502.01976, 2025.

[Zhou, 2021] Zhi-Hua Zhou. Ensemble learning. In *Machine learning*, pages 181–210. Springer, 2021.

[Zou *et al.*, 2025] Jiaru Zou, Yikun Ban, et al. Transformer copilot: Learning from the mistake log in llm fine-tuning. *ArXiv preprint*, abs/2505.16270, 2025.