The *Avengers*: A Routing Recipe for Collective Intelligence in Language Models

Yiqun Zhang[1,2*], **Hao Li**[2,3*], **Chenxu Wang**[2,4], **Linyao Chen**[2,5], **Qiaosheng Zhang**[2], **Peng Ye**[2], **Shi Feng**[1†], **Xinrun Wang**[6], **Jia Xu**[2], **Lei Bai**[2], **Shuyue Hu**[2†]

[1]Northeastern University, Shenyang, China
[2]Shanghai Artificial Intelligence Laboratory, Shanghai, China
[3]Northwestern Polytechnical University, Xi'an, China
[4]Beijing Institute of Technology, Beijing, China
[5]The University of Tokyo, Tokyo, Japan
[6]Singapore Management University, Singapore

Abstract

Proprietary models are increasingly dominating the race for ever-larger language models. Can open-source, smaller models remain competitive across a broad range of tasks? In this paper, we present the *Avengers*—a lightweight framework that leverages collective intelligence in these smaller models. The *Avengers* builds upon four lightweight operations: (i) *embedding*: encode queries using a text embedding model; (ii) *clustering*: group queries based on their semantic similarity; (iii) *scoring*: scores each model's performance within each cluster; and (iv) *voting*: improve outputs via repeated sampling and voting. At inference time, each query is embedded and assigned to its nearest cluster. The top-performing model(s) within that cluster are selected to generate the response with repeated sampling. Remarkably, **with 10 open-source models ($\sim$7B parameters each), the *Avengers* surpasses GPT-4o, 4.1, and 4.5 in average performance across 15 diverse datasets spanning mathematics, coding, logical reasoning, general knowledge, and affective tasks.** In particular, it surpasses GPT-4.1 on mathematics tasks by 18.21% and on code tasks by 7.46%. Furthermore, the *Avengers* delivers superior out-of-distribution generalization, and remains robust across various embedding models, clustering algorithms, ensemble strategies, data efficiency, and values of its sole parameter—the number of clusters.

Code — https://github.com/ZhangYiqun018/Avengers

Introduction

The current language models (LMs) landscape is dominated by few proprietary models, such as GPT and Claude, fueled by unprecedented computational resources. In contrast, the open-source community typically works with smaller LMs. Although recent work shows that individual small models can excel in specific domains (Liu, Guo et al. 2025; Hui et al. 2024; Zhang, Zeng et al. 2024; Cui et al. 2025), a more critical question remains: can it remain competitive not just on narrow tasks, but also in achieving comparable or even superior *overall* performance across a *broad* range of tasks?

A positive answer would have important implications. Since open-source, small models are easily accessible and more amenable, a positive answer would represent a meaningful step toward democratizing AI (Vryn and Das 2025)—enabling a broader community of researchers and practitioners to engage in and contribute to cutting-edge LM research. Moreover, the open-source ecosystem has already invested substantial effort and computational resources in model development.[1] Demonstrating strong performance from these models would justify the reuse of existing efforts. This supports more sustainable AI research (Schwartz et al. 2020) and would provide an alternative to the ever increasing of model size.

We argue that the solution lies in *collective intelligence*—while no single small model may rival a large one in generality, a coordinated ensemble might. Yet realizing this vision raises critical design questions. How should we orchestrate these models to maximize their collective performance? For a notable example, for each incoming query, should we rely on a router-based approach (Jiang, Ren, and Lin 2023; Chen et al. 2024b; Zhuang et al. 2024; Zhang, Zhan, and Ye 2025) that selects the most capable model, or should we adopt a mixture-based strategy (e.g., Mixture-of-Agents (Wang et al. 2024; Li et al. 2024, 2025; Chen 2025)) where multiple models respond in parallel and their outputs are aggregated into a final answer? Given the infeasibility of leveraging all available models in the community, which subset should we use? Moreover, with the rapid evolution of LMs and tasks, how can such systems remain adaptive?

To this end, we introduce the *Avengers*—a lightweight framework that effectively leverages the collective intelligence of smaller LMs. Given a set of models and a dataset (potentially spanning multiple tasks and benchmarks), the *Avengers* operates as follows. First, it analyzes the tasks by embedding queries from the validation set and clustering them based on their semantic representations. Next, to identify each model's strengths and weaknesses, it evaluates each model on the validation set and constructs a cluster-wise capability profile—a simple, numeric vector represent-

[*]These authors contributed equally.
[†]Corresponding authors.

[1]As of July 2025, there are over 1,621 models fine-tuned on LLaMA-3.1-8B-Instruct and 2466 on Qwen-2.5-7B-Instruct.

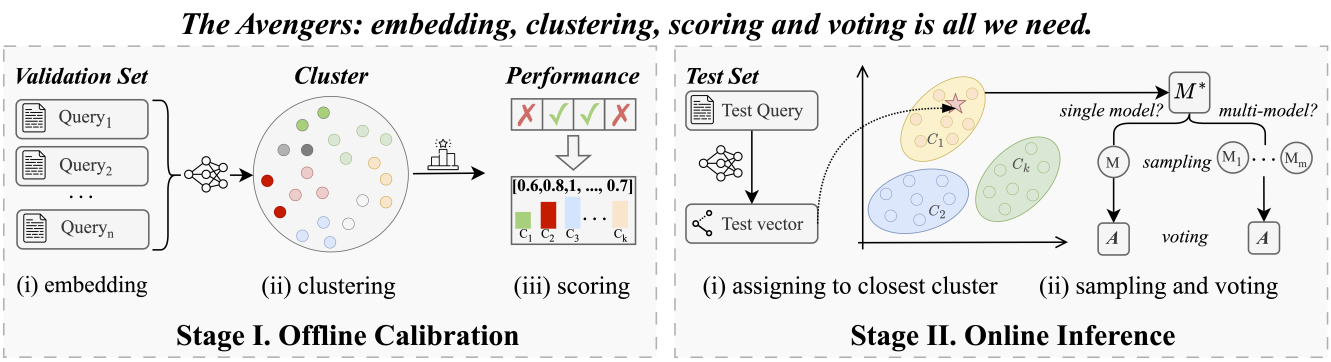


Figure 1: The *Avengers*, a lightweight framework for *collective intelligence* in smaller language models.

ing model performance across clusters. At inference time, for each query in the test set, the *Avengers* computes its embedding and assigns it to the nearest cluster. Based on the capability profiles, the model(s) with the highest performance in that cluster are then selected to handle the query. The final output is generated using a repeated sampling and voting strategy over the selected model(s). An overview of the *Avengers* is presented in Figure 1.

The *Avengers* is straightforward to implement, requiring no neural network training. It is highly reproducible, avoiding reliance on well-designed prompts and certain model choices. It is plug-and-play, allowing seamless integration of newly available models by incrementally evaluating them on the validation set and computing their capability profiles. Moreover, it is well-suited for cold-start, assuming no human expertise for model set composition; given a large candidate model pool, the construction of capability profiles enables automatic selection of high-performing models while preserving diversity. Conceptually, the *Avengers* can be viewed as a routing-based method. However, as detailed in the Related Work section and summarized in the qualitative comparison in Table 1, it differs significantly from most prior work in this category—both methodologically, through its clustering-based routing strategy, and empirically, in the notably strong performance it achieves.

Our extensive experiments show that, the *Avengers*, using 10 models (∼7B each), achieves an average score of 70.54 across 15 diverse datasets, slightly ahead of GPT-4.1 (69.20) [2] and GPT-4.5 (68.18) [3] , and substantially surpassing GPT-4o (60.22) by 17.14%. Specifically, it outperforms GPT-4.1 and GPT-4.5 on 9 datasets, and GPT-4o on 10 datasets. Notable performance gains against GPT-4.1 include +37.04% on AIME, +31.86% on MBPP, and +15.41% on KORBench.

When isolating the effect of the routing mechanism, the *Avengers*' clustering-based router still outperforms state-of-the-art (SOTA) performance-oriented routers (RouterDC (Chen et al. 2024b), embedLLM (Zhuang et al. 2024), and MODEL-SAT (Zhang, Zhan, and Ye 2025)). . Unlike these baselines, which require neural network (e.g., LM) training and retraining for new models or tasks, the *Avengers* offers strong generalization without additional training. On 5 out-of-distribution tasks, its advantage is pronounced: outperforming RouterDC by 10.31% embedLLM by 8.14%, and MODEL-SAT by 8.56%.

Additional experiments demonstrate that the *Avengers* is compatible with a variety of embedding models, clustering algorithms, and ensemble strategies for aggregating outputs from selected models. Furthermore, given a pool of 22 models, the *Avengers* can match GPT-4.1/4.5's performance using only three models selected via its automatic strategy, while performance continues to improve as more models are added. Regarding data efficiency, we show that the *Avengers* continues to outperform GPT-4.1/4.5 even when only 30% of the data is used for validation, leaving the remaining 70% for testing (Figure 2c). Finally, the *Avengers* is robust to variations in its sole hyperparameter—the number of clusters k.

Related Work

Harnessing collective intelligence from multiple models constitutes one of the frontiers of AI and ML research, and has recently attracted much interest (Lu et al. 2024; Guo et al. 2024; Zhang et al. 2025; Subramaniam et al. 2025; Wan et al. 2025; Zheng et al. 2025). Existing approaches for LMs generally fall into three paradigms: router-based, mixture-based, and merging-based methods. This work is most closely aligned with the router-based paradigm. Due to space constraints, we focus our discussion on this line of research and defer coverage of the other two paradigms to the *Supplementary Material*.

Router-based approaches aim to boost the performance of smaller models by *training a neural router* that sends each query to the most capable model (Chen, Zaharia, and Zou 2023; Shnitzer et al. 2023; Ong et al. 2024; Feng, Shen, and You 2025; Shen et al. 2024; Huang et al. 2025). LLM-Blender selects the top-k models per query through pairwise comparisons and fuses their outputs (Jiang, Ren, and Lin 2023), while ZOOTER performs reward-guided routing with tag-based label enhancement (Lu and Yuan 2024). More recently, RouterDC applies dual contrastive learning to improve routing accu-

[2]https://openai.com/index/gpt-4-1: OpenAI's flagship model released on April 14, 2025.

[3]https://openai.com/index/introducing-gpt-4-5: OpenAI's best chat model released on Feb. 27, 2025.

Method	Trainable NN†	Arch.	Prompt	New Task ‡	New Model‡	Model Scale§	Auto Init. Sel.
RouterDC	Emb+MLP	Free	Free	LS+Retrain	NN	✓	✗
EmbedLLM	MLP	Free	Free	LS+Retrain	NN	✓	✗
MODEL-SAT	Emb+MLP+SLM	Free	Hand-Craft	LS+Retrain	NN	✓	✗
MoA	Free	Hand-Craft	Hand-Craft	Free		✗	✗
Symb.-MoE	Free	Free	Hand-Craft	LS+Regen Profile		✓	✗
Avenger (ours)	Free	Free	Free	LS+Recluster	LS	✓	✓

Table 1: Methodological comparison between baseline methods and the *Avengers*. **Trainable NN**: additional neural components requiring training. **Arch.**: effort in architecture design. **Prompt Eng.**: effort in prompt design. **New Task**: how to adapt to new tasks. **New Model**: how to adapt to new models. **Model Scale**: can scale with the number of SLMs. **Auto Init. Sel.**: can automatically select the initial SLM subset. † Emb = trainable embedding model; MLP = small feed-forward head. ‡ LS = labeled samples. § Mixture of Agents exceeds the maximum context window for SLMs (e.g., 8192 tokens for Gemma-2-9B) as the number of SLMs increases.

racy (Chen et al. 2024b); EmbedLLM leverages compact model and query embeddings for routing prediction (Zhuang et al. 2024); and Model-SAT generates capability instructions with text-aligned embeddings to steer a lightweight LLM in choosing optimal candidate models (Zhang, Zhan, and Ye 2025).

In parallel with the above performance-oriented research, several recent works trade performance for computational cost. Routellm learns a binary preference classifier that routes each query to either a strong or weak LLM (Ong et al. 2024). GraphRouter encodes tasks, queries, and LLMs in a heterogeneous graph and predicts a performance-cost score on every edge (Feng, Shen, and You 2025). RouterBench contributes benchmarks plus strategies such as sending a query to the LLM with the best cost-penalized average score among its k nearest neighbours (Hu et al. 2024). Likewise, (Jitkrittum et al. 2025) clusters queries with K-means on an unlabeled corpus and selects an LLM using cost-adjusted per-cluster error computed on a labeled set.

The *Avengers* falls within the router-based paradigm but departs significantly from prior work in two important ways. First, unlike most methods that rely on neural routers—requiring retraining to handle new tasks or models (Chen et al. 2024b; Zhuang et al. 2024; Zhang, Zhan, and Ye 2025)—the *Avengers* is entirely free of neural network training, yet remains highly adaptable. It offers a lightweight, reproducible alternative without sacrificing performance. Second, while K-NN (Hu et al. 2024) or K-means routing (Jitkrittum et al. 2025), which do not require neural network, have been explored in cost-performance tradeoff settings, they typically operate on model pools with large size disparities (including models far larger than those used in our work), yet *fail* to demonstrate competitive performance. To our knowledge, this is the first work to show that a router-based framework, even without neural network training, can elevate small open-source models to match the *overall* performance of proprietary, flagship LLMs.

The *Avengers*

We introduce the Avengers, a simple yet effective framework for leveraging collective intelligence in multiple small language models. Open-source repositories already supply a rich variety of such models, whose diversity may stem from two sources: (i) domain-specialised experts fine-tuned for tasks like mathematical reasoning, and (ii) general-purpose models that still differ because of disparate training data, architectures, and design choices across stakeholders.

This diversity provides a strong foundation for collective intelligence. However, operationalizing it requires careful consideration of several key design decisions. A core principle underlying the *Avengers* is the idea of "**horses for courses**"—selecting the most suitable model(s) for each incoming query and routing the query to that model. We prioritize a routing strategy over a mixture strategy, because unlike larger models, smaller models tend to be more sensitive to prompt variations and often struggle with instruction following. Prior work has shown that mixture-based methods, while effective with larger models, can lead to degraded performance when applied to smaller ones (Li et al. 2025).

Let M denote the model pool and $\mathcal{D}$ a set of query–answer pairs, split into a validation set $\mathcal{D}_{\text{val}}$ and a test set $\mathcal{D}_{\text{test}}$. The *Avengers* tackles the problem in two phases: (i) offline calibration, which profiles query categories and model abilities on $\mathcal{D}_{\text{val}}$; and (ii) online inference, which routes every $\mathcal{D}_{\text{test}}$ query to the most capable model and refines the final answer. *Note that the use of a validation set is standard in all router-based methods (Chen et al. 2024b; Zhuang et al. 2024; Hu et al. 2024; Zhang, Zhan, and Ye 2025), as it helps characterize model strengths and query types*. However, unlike prior approaches that require neural network training on this set (sometimes referred to as a training set), our method is free of neural network training.

Offline Calibration During this stage, the *Avengers* constructs a structured understanding of the query type via embedding and clustering, and characterizes model capabilities through cluster-wise performance scores. First, to characterize query types, each query $d \in \mathcal{D}_{\text{val}}$ is encoded using a text **embedding** model, producing a semantic vector representation. Then, these vectors are clustered into k groups with the use of a **clustering** method (e.g., K-means), which results in a set $\mathcal{C}$ of k distinct clusters. Each cluster $c \in \mathcal{C}$ represents a semantically coherent group of queries, i.e., a query

type. Next, to characterize model capabilities, each model $m \in \mathcal{M}$ is evaluated on $\mathcal{D}_{\text{val}}$, and its performance score is recorded within each cluster. This yields a cluster-wise **capability profile** for each model, which can be represented as a vector $p = [p_1, \cdots, p_k]$, where p_i denotes the model's performance score on cluster $c_i \in \mathcal{C}$. These profiles inform routing decisions at the inference time.

Online Inference At the inference time, the *Avengers* no longer performs clustering or profiling. For each query in the test set $\mathcal{D}_{\text{test}}$, the *Avengers* first computes its **embedding** using the same text embedding model used during offline calibration. The query is then **routed** to the nearest cluster $c_* \in \mathcal{C}$, based on distance in the embedding space. Given the nearest cluster c_*, the *Avengers* consult the capability profiles and select the top-n model(s) with the highest performance score p_* within that cluster. Notably, the selected model(s) is not necessarily the one with the best overall performance on $\mathcal{D}_{\text{val}}$ or $\mathcal{D}_{\text{test}}$. Rather, it is selected for its strength in the specific query type (or cluster). Once the top-n model(s) is selected, the *Avengers* generates responses using **repeated sampling**, followed by majority **voting** to determine the final output. That is, when $n = 1$, it adopts the Self-Consistency (SC) (Wang et al. 2022). When $n > 1$, it adopts the Model-Switch (Chen et al. 2025), a multi-model, sample-efficient extension of SC.

Automatic Model Set Construction It is worth mentioning that the composition of the model set $\mathcal{M}$ can significantly influence collective performance. Intuitively, strong overall performance is unlikely to emerge from a group of models that underperform across all tasks. While prior approaches often rely on human expertise or domain knowledge to curate such model sets, the *Avengers* automates the selection of models with complementary strengths. Given a predefined deployment budget of $|\mathcal{M}|$ models and a larger candidate pool $\mathcal{M}'$, the *Avengers* enables automatic model selection from $\mathcal{M}'$. During the offline calibration stage, for each candidate model $m' \in \mathcal{M}'$, the *Avengers* computes the cluster-wise capability profile and calculates an overall score across clusters: $s(m') = \sum_{c \in \mathcal{C}} 1/r^c_{m'}$, where $r^c_{m'}$ denotes the rank of model m' within cluster c, determined by its cluster-wise performance. A higher score $s(m')$ indicates that the model either excels in specific clusters (yielding high ranks) or maintains strong performance across multiple query types. By selecting the top $|\mathcal{M}|$ models with the highest overall scores, the *Avengers* naturally constructs a complementary model set $\mathcal{M}$ that balances specialization and diversity across different query types.

New Datasets and Newly Available Models The fast-paced evolution of LM research regularly introduces new datasets, tasks, and models. While prior routing-based methods often require retraining neural networks to accommodate such changes, the *Avengers* is designed to be highly adaptable. To incorporate a new, out-of-distribution dataset $\mathcal{D}'$, the *Avengers* re-executes the offline calibration stage. This requires only *incremental* re-evaluation of existing models $\mathcal{M}$ on the validation set $\mathcal{D}'_{\text{val}}$, followed by re-clustering the queries from $\mathcal{D}'_{\text{val}} \cup \mathcal{D}_{\text{val}}$ based on updated query embeddings. Incorporating a newly available model $m^\dagger$ is even more lightweight. The *Avengers* simply *incrementally* evaluates the new model $m^\dagger$ on the existing validation set $\mathcal{D}_{\text{val}}$ and then computes its cluster-level performance profile, even without re-clustering.

Experiments

Experimental Setup

Datasets We primarily consider 15 datasets covering five categories: **Mathematics** (AIME, Math500 (Lightman et al. 2023), and LiveMathBench (Liu et al. 2024)), **Code** (MBPP (Austin et al. 2021) and HumanEval (Chen and Tworek 2021)), **Logic** (KORBench (Ma, Du et al. 2024), Knights and Knaves (Xie et al. 2024), and BBH (Suzgun, Scales et al. 2022)), **Knowledge** (ARC Challenge (Clark et al. 2018), MMLUPro (Wang, Ma et al. 2024), GPQA (Rein et al. 2024), FinQA (Chen, Chen et al. 2021), and MedQA (Jin et al. 2021)), and **Affective** (EmoryNLP (Byrkjeland, de Lichtenberg, and Gambäck 2018) and MELD (Poria et al. 2019)). To assess **out-of-distribution (OOD) generalization**—defined as in earlier routing studies—we add one dataset per category: MathBench (Liu, Zheng et al. 2024), StudentEval (Babe et al. 2023), Winogrande (Sakaguchi et al. 2021), BRAINTEASER (Rahimi et al. 2024), and DailyDialog (Li et al. 2017). These five sets are used solely for OOD evaluation and are **never included in clustering and validation**; full dataset details appear in the *Supplementary Material*.

Implementation of the *Avengers* We uniformly employ *gte-qwen2-7B-instruct* (Li et al. 2023) as the embedding model and utilize the classic K-Means with the number of clusters set to $K = 64$. Following common practice in the literature (Chen et al. 2024b; Zhuang et al. 2024), each dataset is randomly split into 70% for fitting cluster centroids and the remaining 30% for performance evaluation. Initially, we consider a pool consisting of 22 open-source candidate LLMs (∼7B), from which $m = 10$ models are automatically selected. During inference, we adopt the Self-Consistency (SC) strategy (Wang et al. 2022) by default, setting the number of sampling rounds to 10 to balance efficiency and accuracy. All experiments are repeated five times using five random seeds (42, 999, 2024, 2025, and 3407), and average results are reported.

Baselines We compare the *Avengers* against baselines from two lines of work—mixture-based and router-based—and also report three proprietary references: GPT-4o-2024-08-06, GPT-4.5, and GPT-4.1-2025-04-14. The baselines considered are including **RouterDC** (Chen et al. 2024b), **EmbedLLM** (Zhuang et al. 2024), **MODEL-SAT** (Zhang, Zhan, and Ye 2025), **Mixture of Agents (MoA)** (Wang et al. 2024) and **Symbolic-MoE (SMoE)** (Chen 2025). The methodological differences between these baseline methods and the *Avengers* are summarized in Table 1. Additionally, we introduce two simpler baselines for reference: a **Random Router**, which randomly assigns models to queries, and an **LLM Router**, which uses an LLM to assign models based on query content and model

Setting	Mathematics			Code		Logical			Knowledge					Affective		Avg.
	AIME	M500.	LMB.	MBPP	HE.	KOR.	K&K.	BBH	ARCC	MP.	GPQA	FinQA	MedQA	Emory.	MELD	
Small Language Model (Enhanced)																
DS-Qwen	61.67	93.00	65.71	56.67	43.90	52.00	76.10	80.00	88.82	59.14	59.15	68.53	40.85	29.84	38.64	60.93
Fin-R1	13.33	80.20	35.00	69.30	77.44	37.52	18.14	68.24	91.89	51.55	20.76	70.71	67.09	41.18	57.55	53.33
Qwen-it	15.00	78.60	37.14	70.73	81.10	49.60	35.92	63.24	88.91	58.34	30.13	68.00	65.28	39.89	57.87	55.98
Qwen-Coder	11.67	75.60	35.00	76.39	78.66	34.64	27.57	61.48	86.52	49.25	32.14	65.13	55.38	40.46	59.58	52.63
gemma-2-it	1.67	54.00	22.14	62.32	64.02	34.32	15.71	63.52	89.42	53.55	34.38	66.26	66.46	39.60	54.38	48.12
glm-4-chat	5.00	58.40	22.14	62.01	65.85	37.60	21.57	47.59	92.15	51.75	31.25	58.59	64.96	41.32	57.46	47.84
Llama-3.1-it	1.67	49.80	20.71	61.81	71.95	27.60	11.71	65.74	88.48	47.35	25.67	53.97	69.76	35.15	51.62	45.53
Granite-3.1-it	1.67	61.40	20.00	37.17	39.63	31.44	19.29	36.39	85.24	44.06	34.82	65.74	59.70	39.60	48.30	41.63
UltraMedical	1.67	76.20	15.71	52.77	58.54	18.72	20.14	40.46	85.67	43.76	19.87	60.68	72.90	31.28	43.34	42.78
cogito-v1	3.33	59.60	22.86	51.33	70.12	44.48	28.00	75.83	90.01	61.94	31.92	64.69	69.36	39.02	55.68	51.21
Average	11.67	68.68	29.64	60.05	65.12	36.79	27.42	60.25	88.71	52.07	32.01	64.23	63.17	37.73	52.44	50.00
Max Expert	61.67	93.00	65.71	76.39	81.10	52.00	76.10	80.00	92.15	61.94	59.15	70.88	72.90	41.32	59.58	69.59
Oracle*	*63.33*	*96.60*	*69.29*	*89.43*	*95.73*	*65.76*	*82.43*	*96.85*	*96.83*	*86.41*	*85.49*	*85.44*	*91.99*	*66.86*	*82.71*	*83.68*
Proprietary Model Baseline																
GPT-4o	10.00	76.00	35.71	82.64	85.36	57.68	32.57	79.53	93.86	59.84	44.42	72.28	82.17	38.31	52.92	60.22
GPT-4.5	31.67	88.60	50.00	86.69	69.51	57.44	89.29	91.36	94.60	59.94	51.78	71.75	92.85	39.45	47.84	68.18
GPT-4.1	45.00	89.20	52.14	57.70	92.07	48.48	84.43	90.74	95.39	65.13	62.05	71.05	88.77	39.45	56.33	69.20
Mixture-based Baseline																
MoA	23.33	82.67	33.33	54.61	84.51	49.87	34.28	47.53	91.81	64.54	50.37	71.04	71.20	39.17	52.76	56.70
MoA (Oracle)	27.77	84.67	54.76	69.40	83.54	51.28	31.14	57.50	92.75	61.74	50.44	70.71	76.05	39.45	51.95	60.21
SMoE	43.00	88.20	42.86	65.81	70.12	48.88	43.71	67.69	87.63	58.94	43.30	65.21	64.96	37.30	51.38	58.60
Router-based Baseline (Enhanced)																
Random Router	10.00	66.20	32.86	61.29	62.80	35.60	27.57	61.20	88.65	51.25	33.04	64.60	62.61	38.16	50.89	49.78
LLM Router	61.67	91.80	64.29	69.92	78.05	37.12	74.71	80.28	88.99	60.14	45.31	66.78	73.84	32.71	45.78	64.76
RouterDC†	58.89	89.87	64.76	68.53	73.20	44.64	73.33	73.33	90.00	54.68	43.41	68.12	66.81	42.00	58.87	64.70
MODEL-SAT†	61.11	91.73	65.24	69.42	80.00	54.77	72.76	82.96	90.34	65.12	58.67	68.70	66.70	40.19	58.00	68.38
EmbedLLM†	61.11	90.80	64.76	74.27	78.80	55.73	73.81	85.06	91.14	65.32	57.33	67.12	71.83	41.33	60.22	69.24
***Avengers* (ours)**	61.67	92.89	65.71	76.08	84.86	55.95	76.14	84.07	92.39	66.21	55.76	71.53	73.89	41.38	59.61	70.54
- vs GPT-4.1 (%)	↑37.04	↑4.14	↑26.03	↑31.86	↓7.83	↑15.41	↓9.82	↓7.35	↓3.15	↑1.66	↓10.14	↑0.67	↓16.76	↑4.90	↑5.83	↑1.95
- vs MoA (Oracle) (%)	↑122.06	↑9.71	↑20.00	↑9.63	↑1.58	↑9.11	↑144.52	↑46.21	↓0.39	↑7.24	↑10.54	↑1.16	↓2.84	↑4.90	↑14.75	↑17.16
- vs EmbedLLM (%)	↑0.91	↑2.31	↑1.47	↑2.44	↑7.69	↑0.39	↑3.16	↓1.16	↑1.37	↑1.37	↓2.75	↑6.57	↑2.87	↑0.12	↓1.00	↑1.88

Table 2: Comparison of the *Avengers* with baselines. *Oracle** represents the best achievable score by selecting the optimal model per query. †Peak score on the test set. Max Expert represents the best performance of the ten models on the dataset.

descriptions. More detailed implementation specifics can be found in *Supplementary Material*.

Enhancing Baselines for Fair Comparison Note that baseline performance often depends on *manually* chosen models. To ensure fairness, all baseline methods in our study use the same set of 10 models, automatically selected by our method. For MoA, we additionally evaluate an oracle variant—MoA (Oracle)—in which we manually **select the top-3 performing models** per task instead of using all 10 models. This approach addresses the performance degradation arising from excessively long context windows in the original MoA setup. For RouterDC, EmbedLLM, and MODEL-SAT, we report each method's **peak test performance** rather than using a fixed training step. All SLMs and router-based results in Tables 2 and 3 also use the same Self-Consistency (SC) strategy as *Avengers*, with 10 samples per query, to ensure fair and robust comparison. These enhancements are not part of the original baseline designs but empirically improve their performance.

The *Avengers* Achieves SOTA Performance

Table 2 presents a detailed performance comparison between the *Avengers* and three groups of baselines across 15 datasets in five task categories. First, the *Avengers* achieves an average score of 70.54, slightly ahead of GPT-4.1's 69.20 and GPT-4.5's 68.18, and significantly surpassing GPT-4o by 17.14%, demonstrating that dynamically selecting among smaller models yields performance comparable to large-scale proprietary models. Specifically, the *Avengers* outperforms GPT-4.1 and GPT-4.5 on 9 of the 15 datasets, and GPT-4o on 10 datasets as well. It achieves an average improvement of 18.21% over GPT-4.1 on mathematical tasks, with notable margins of 31.86% on MBPP and 15.41% on KORBench. However, GPT-4.1 maintains its advantage on specialized knowledge-intensive tasks (e.g., GPQA, MedQA), likely due to its larger parameter size and broader knowledge base. Nevertheless, the *Avengers* exceeds the max expert by 6.89% on MMLUPro (61.94 to 66.21), nearly matching GPT-4.1, underscoring the comple-

Setting	MathBench	StudentEval	Winogrande	BrainTeaser	DailyDialog	Avg.
DS-Qwen	97.33	53.83	65.98	68.46	38.30	64.78
Fin-R1	65.33	65.69	77.82	70.13	43.30	64.45
Qwen-it	83.33	65.48	69.85	67.78	53.30	67.95
Qwen-Coder	75.33	63.80	67.56	59.84	57.70	64.85
gemma-2-it	53.30	64.22	73.16	76.00	50.90	63.52
glm-4-chat	56.67	60.65	77.74	62.68	59.10	63.37
Llama-3.1-it	49.33	68.84	64.09	70.03	37.70	58.00
Granite-3.1-it	57.33	40.50	72.77	61.80	35.20	53.52
UltraMedical	38.67	57.61	62.43	62.88	32.60	50.84
cogito-v1	66.00	56.45	64.09	73.16	37.70	59.48
Average	64.26	59.71	69.55	67.28	44.58	61.07
Max Expert	97.33	68.84	77.82	76.00	59.10	67.95
Oracle	*100.00*	*84.89*	*95.26*	*94.32*	*75.30*	*89.95*
Random Router	64.67	55.82	69.53	66.70	43.60	60.06
LLM Router	94.67	67.79	65.27	67.48	37.10	66.46
RouterDC†	77.27	62.50	69.77	70.91	59.20	66.89
MODEL-SAT†	97.20	58.22	68.37	70.58	48.40	68.55
EmbedLLM†	97.20	63.06	71.11	69.34	47.98	69.77
***Avengers* (ours)**	96.67	68.52	77.43	71.79	57.70	74.42
- vs EmbedLLM (%)	↓0.55	↑4.98	↑19.17	↑2.20	↑24.30	↑8.14

Table 3: Out-of-Distribution performance. †Peak score on the test set.

mentary potential of smaller models.

Second, mixture-based methods (MoA, SMoE, and MoA (Oracle)) perform poorly under our setting, primarily because they rely on aggregating outputs from multiple models—a process requiring strong instruction-following skills and longer context windows (Li et al. 2025). Small models ($\sim$7B) typically lack these abilities, thus limiting the effectiveness of mixture-based approaches (score < 61) compared to the *Avengers*. Previous mixture-based studies usually employed larger models (e.g., $\sim$32B) (Wang et al. 2024), where these capabilities are more readily available.

Third, even when we enhance router-based methods by selecting their peak on test sets, the *Avengers* still **achieves the best average performance** without involving additional training parameters. Moreover, traditional router-based methods often struggle in Out-of-Distribution (OOD) settings. To compare their generalization ability, we introduce a new dataset for each of the five task categories to evaluate performance. As shown in Table 3, the *Avengers* **exhibits the most robust generalization** among all compared methods, achieving an average score of 74.42. This performance surpasses the best router-based baseline by EmbedLLM (+8.14%).

Note that the baseline results include enhancements described in the previous section. We report these improved versions rather than the original implementations for two reasons: (i) to isolate the effect of the routing mechanism for fairer comparison with our clustering-based approach; and (ii) to show that some of *Avengers*' gains can be replicated by other routing methods, with additional, lightweight operations introduced by the *Avengers*.

The Key Elements that Make the *Avengers* Work

To understand why the *Avengers* is both lightweight and effective, we conduct targeted ablations on five design choices—embedding model, clustering method, model selection, ensemble strategy and data efficiency—while keeping all other components fixed.

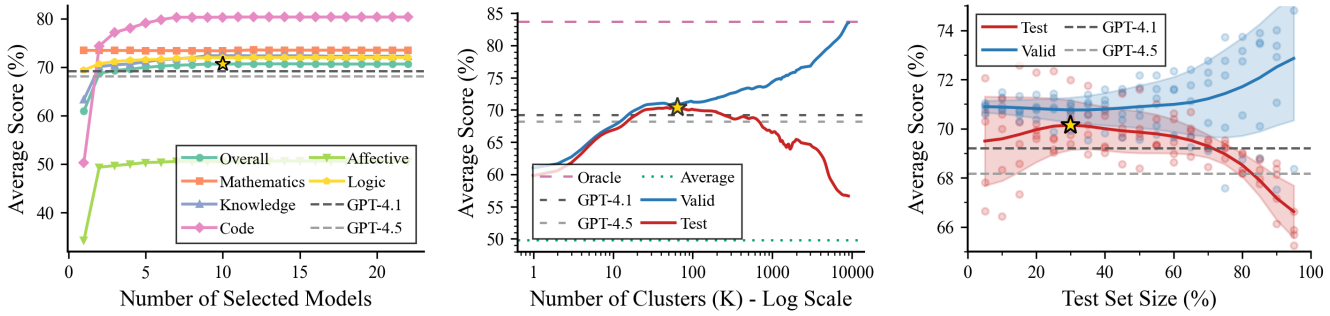
Embedding Model Since clustering quality depends on the embedding model's semantic representations, we examine how different embedding models affect performance. We compare five models with varying parameter sizes and embedding dimensions (Table 4a): bge-m3(Chen et al. 2024a), text-embedding-3-small (text-3-small)(OpenAI 2024), gte-qwen2-1.5B-instruct (gte-q2-1.5B) (Li et al. 2023), text-embedding-3-large (text-3-large), and gte-qwen2-7B-instruct (gte-q2-7B). The results show **minimal performance variation** (70.28 $\rightarrow$ 70.70), indicating that our method is robust to embedding model choice.

Clustering Method Since the *Avengers* relies on query clustering, the choice of clustering method could potentially impact overall performance. We conduct experiments using five classic clustering algorithms: K-Means (Lloyd 1982; MacQueen 1967), Hierarchical Clustering (Ward Jr 1963), Gaussian Mixture Models (GMM) (Dempster, Laird, and Rubin 1977), Spectral Clustering (Von Luxburg 2007) and BIRCH (Zhang, Ramakrishnan, and Livny 1996). We fix the number of clusters at $K = 64$. As shown in Table 4b, there is **minimal performance variation** (70.04–70.71), indicating that the *Avengers* framework is robust and stable across different clustering algorithms.

Model Selection Figure 2a shows performance as a function of the number of models automatically selected by *Avengers*. Notably, **with only three selected models,**

(a) Embedding Models			(b) Clustering		(c) Ensemble	
Embedding	**Para. (B)/Dim.**	**Avg.**	**Method**	**Avg.**	**Strategy**	**Avg.**
bge-m3	0.56/1 024	70.28	K-Means	70.54	Direct (CoT)	59.19
text-3-small†	—/1 536	70.42	Hierarchical	**70.71**	Aggregation	56.73
gte-q2-1.5B	1.5/1 536	70.48	GMM	70.58	Model-Switch	66.42
text-3-large†	—/3 072	**70.70**	Spectral	70.61	Self-Consistency	**66.98**
gte-q2-7B	7/3 584	70.54	BIRCH	70.04		

Table 4: (a) Average score of the *Avengers* with different embedding models; (b) Average scores for various clustering methods; (c) Ablation on ensemble strategies (code tasks excluded).



(a) Selected models (b) Cluster count K (c) Test-set proportion

Figure 2: Ablation studies for *Avengers*: (a) impact of the number of selected models; (b) impact of cluster count K; (c) impact of test-set proportion.

***Avengers* already matches the performance of GPT-4.1 and GPT-4.5**, underscoring the strength of its automatic model selection strategy. Gains are most pronounced in knowledge, code, and affective tasks, while math and logic see limited improvement due to single-model dominance (DS-Qwen). These results confirm that *Avengers* leverages model complementarity to achieve strong general-purpose performance without using all available models.

The Impact of Cluster Count K Figure 2b demonstrates the performance dynamics of the *Avengers* on both validation and test sets as the cluster count K varies from 1 to 10,000 (logarithmic scale on the horizontal axis). Notably, **our method surpasses GPT-4.1/4.5 across a wide range of K values** (approximately 14 to 140), indicating robustness and minimal sensitivity to the choice of K.

Ensemble Strategy Table 4c compares Self-Consistency, Model-Switch, Direct (CoT), and Aggregation. Self-Consistency performs best overall due to simplicity, effectiveness, and robustness, followed closely by Model-Switch. Consistent with MoA's results, Aggregation ranks last, as it requires processing longer context windows from multiple models. Further analysis (see Supplementary Material) reveals Model-Switch excels on knowledge-intensive tasks, while SC outperforms on mathematical and reasoning tasks.

Data Efficiency We vary the test-set share from 0.05 to 0.95 and find, in Figure 2c, that *Avengers* peaks at a 0.30–0.35 split. **Even with only 30% of the data used for clustering, it already surpasses GPT-4.1/4.5**, confirming that our approach works without large labeled sets.

Conclusion

We introduce the *Avengers*, a lightweight yet powerful framework that orchestrates multiple small language models (SLMs) to rival much larger, proprietary LLMs. Our framework relies solely on embedding, clustering, scoring, and voting—requiring no neural network training, prompt design, or human expertise in model selection—while remaining highly adaptable to new domains and evolving model pools. Using 10 models (∼7B parameters each), it achieves strong results on 15 tasks: it surpasses GPT-4.1 and GPT-4.5 each on 9 datasets, and GPT-4o on 10 datasets, averaging a 17.14% improvement over GPT-4o across all tasks. It also beats state-of-the-art router baselines on out-of-distribution tasks by 8%. Ablations confirm robustness to: embedding models, clustering algorithms, the number of clusters, and ensemble strategies. As more models are added, complementary strengths emerge, especially in knowledge, code, and affective tasks.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (Grant Nos. 62272092, 62172086, 6250076060), the Fundamental Research Funds for the Central Universities (N25XQD004, N25ZLL045), and the Shanghai Municipal Science and Technology Major Project. This work was done during Yiqun Zhang, Hao Li, Chenxu Wang, and Linyao Chen's internships at Shanghai Artificial

Intelligence Laboratory.

References

Austin, J.; Odena, A.; Nye, M.; Bosma, M.; Michalewski, H.; Dohan, D.; Jiang, E.; Cai, C.; Terry, M.; Le, Q.; et al. 2021. Program Synthesis with Large Language Models. *arXiv preprint arXiv:2108.07732*.

Babe, H. M.; Nguyen, S.; Zi, Y.; Guha, A.; Feldman, M. Q.; and Anderson, C. J. 2023. StudentEval: A benchmark of student-written prompts for large language models of code. *arXiv preprint arXiv:2306.04556*.

Byrkjeland, M.; de Lichtenberg, F. G.; and Gambäck, B. 2018. Ternary Twitter sentiment classification with distant supervision and sentiment-specific word embeddings. In *Proceedings of the 9th workshop on computational approaches to subjectivity, sentiment and social media analysis*, 97–106.

Chen, J.; Xiao, S.; Zhang, P.; Luo, K.; Lian, D.; and Liu, Z. 2024a. BGE M3-Embedding: Multi-Lingual, Multi-Functionality, Multi-Granularity Text Embeddings Through Self-Knowledge Distillation. arXiv:2402.03216.

Chen, J.; Xun, Z.; Zhou, B.; Qi, H.; Zhang, Q.; Chen, Y.; Hu, W.; Qu, Y.; Ouyang, W.; and Hu, S. 2025. Do We Truly Need So Many Samples? Multi-LLM Repeated Sampling Efficiently Scales Test-Time Compute. arXiv:2504.00762.

Chen, J. C.-Y. 2025. Symbolic Mixture-of-Experts: Adaptive Skill-based Routing for Heterogeneous Reasoning. arXiv:2503.05641.

Chen, L.; Zaharia, M.; and Zou, J. 2023. FrugalGPT: How to Use Large Language Models While Reducing Cost and Improving Performance. *Transactions on Machine Learning Research*.

Chen, M.; and Tworek, J. 2021. Evaluating Large Language Models Trained on Code. arXiv:2107.03374.

Chen, S.; Jiang, W.; Lin, B.; Kwok, J.; and Zhang, Y. 2024b. Routerdc: Query-based router by dual contrastive learning for assembling large language models. *Advances in Neural Information Processing Systems*, 37: 66305–66328.

Chen, Z.; Chen, W.; et al. 2021. FinQA: A Dataset of Numerical Reasoning over Financial Data. *Proceedings of EMNLP 2021*.

Clark, P.; Cowhey, I.; Etzioni, O.; Khot, T.; Sabharwal, A.; Schoenick, C.; and Tafjord, O. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.

Cui, G.; Yuan, L.; Wang, Z.; Wang, H.; Li, W.; He, B.; Fan, Y.; Yu, T.; Xu, Q.; Chen, W.; et al. 2025. Process reinforcement through implicit rewards. *arXiv preprint arXiv:2502.01456*.

Dempster, A. P.; Laird, N. M.; and Rubin, D. B. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the royal statistical society: series B (methodological)*, 39(1): 1–22.

Feng, T.; Shen, Y.; and You, J. 2025. GraphRouter: A Graph-based Router for LLM Selections. In *The Thirteenth International Conference on Learning Representations*.

Guo, T.; Chen, X.; Wang, Y.; Chang, R.; Pei, S.; Chawla, N. V.; Wiest, O.; and Zhang, X. 2024. Large language model based multi-agents: A survey of progress and challenges. *arXiv preprint arXiv:2402.01680*.

Hu, Q. J.; Bieker, J.; Li, X.; Jiang, N.; Keigwin, B.; Ranganath, G.; Keutzer, K.; and Upadhyay, S. K. 2024. RouterBench: A Benchmark for Multi-LLM Routing System. In *Agentic Markets Workshop at ICML 2024*.

Huang, Z.; Ling, G.; Liang, V. S.; Lin, Y.; Chen, Y.; Zhong, S.; Wu, H.; and Lin, L. 2025. RouterEval: A Comprehensive Benchmark for Routing LLMs to Explore Model-level Scaling Up in LLMs. *arXiv preprint arXiv:2503.10657*.

Hui, B.; Yang, J.; Cui, Z.; Yang, J.; Liu, D.; Zhang, L.; Liu, T.; Zhang, J.; Yu, B.; Lu, K.; et al. 2024. Qwen2.5-coder technical report. *arXiv preprint arXiv:2409.12186*.

Jiang, D.; Ren, X.; and Lin, B. Y. 2023. LLM-Blender: Ensembling Large Language Models with Pairwise Ranking and Generative Fusion. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 14165–14178.

Jin, D.; Pan, E.; Oufattole, N.; Weng, W.-H.; Fang, H.; and Szolovits, P. 2021. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *Applied Sciences*, 11(14): 6421.

Jitkrittum, W.; Narasimhan, H.; Rawat, A. S.; Juneja, J.; Wang, Z.; Lee, C.-Y.; Shenoy, P.; Panigrahy, R.; Menon, A. K.; and Kumar, S. 2025. Universal Model Routing for Efficient LLM Inference. *arXiv preprint arXiv:2502.08773*.

Li, D.; Tan, Z.; Qian, P.; Li, Y.; Chaudhary, K. S.; Hu, L.; and Shen, J. 2024. Smoa: Improving multi-agent large language models with sparse mixture-of-agents. *arXiv preprint arXiv:2411.03284*.

Li, W.; Lin, Y.; Xia, M.; and Jin, C. 2025. Rethinking Mixture-of-Agents: Is Mixing Different Large Language Models Beneficial? *arXiv preprint arXiv:2502.00674*.

Li, Y.; Su, H.; Shen, X.; Li, W.; Cao, Z.; and Niu, S. 2017. Dailydialog: A manually labelled multi-turn dialogue dataset. *arXiv preprint arXiv:1710.03957*.

Li, Z.; Zhang, X.; Zhang, Y.; Long, D.; Xie, P.; and Zhang, M. 2023. Towards general text embeddings with multi-stage contrastive learning. *arXiv preprint arXiv:2308.03281*.

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

Liu, H.; Zheng, Z.; et al. 2024. Mathbench: Evaluating the theory and application proficiency of llms with a hierarchical mathematics benchmark. *arXiv preprint arXiv:2405.12209*.

Liu, J.; Liu, H.; Xiao, L.; Wang, Z.; Liu, K.; Gao, S.; Zhang, W.; Zhang, S.; and Chen, K. 2024. Are Your LLMs Capable of Stable Reasoning? *arXiv preprint arXiv:2412.13147*.

Liu, Z.; Guo, X.; et al. 2025. Fin-r1: A large language model for financial reasoning through reinforcement learning. *arXiv preprint arXiv:2503.16252*.

Lloyd, S. 1982. Least squares quantization in PCM. *IEEE transactions on information theory*, 28(2): 129–137.

Lu, J.; Pang, Z.; Xiao, M.; Zhu, Y.; Xia, R.; and Zhang, J. 2024. Merge, ensemble, and cooperate! a survey on collaborative strategies in the era of large language models. *arXiv preprint arXiv:2407.06089*.

Lu, K.; and Yuan, H. 2024. Routing to the Expert: Efficient Reward-guided Ensemble of Large Language Models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, 1964–1974.

Ma, K.; Du, X.; et al. 2024. KOR-Bench: Benchmarking Language Models on Knowledge-Orthogonal Reasoning Tasks. arXiv:2410.06526.

MacQueen, J. 1967. Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*, volume 5, 281–298. University of California press.

Ong, I.; Almahairi, A.; Wu, V.; Chiang, W.-L.; Wu, T.; Gonzalez, J. E.; Kadous, M. W.; and Stoica, I. 2024. RouteLLM: Learning to Route LLMs from Preference Data. In *The Thirteenth International Conference on Learning Representations*.

OpenAI. 2024. New embedding models and API updates. https://openai.com/index/new-embedding-models-and-api-updates/. Accessed: 2025-05-10.

Poria, S.; Hazarika, D.; Majumder, N.; Naik, G.; Cambria, E.; and Mihalcea, R. 2019. MELD: A Multimodal Multi-Party Dataset for Emotion Recognition in Conversations. arXiv:1810.02508.

Rahimi, Z.; Shirzady, M. M.; Taghavi, Z.; and Sameti, H. 2024. NIMZ at SemEval-2024 task 9: Evaluating methods in solving brainteasers defying commonsense. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, 148–154.

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

Sakaguchi, K.; Bras, R. L.; Bhagavatula, C.; and Choi, Y. 2021. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9): 99–106.

Schwartz, R.; Dodge, J.; Smith, N. A.; and Etzioni, O. 2020. Green ai. *Communications of the ACM*, 63(12): 54–63.

Shen, Z.; Lang, H.; Wang, B.; Kim, Y.; and Sontag, D. 2024. Learning to Decode Collaboratively with Multiple Language Models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 12974–12990.

Shnitzer, T.; Ou, A.; Silva, M.; Soule, K.; Sun, Y.; Solomon, J.; Thompson, N.; and Yurochkin, M. 2023. Large Language Model Routing with Benchmark Datasets. In *First Conference on Language Modeling*.

Subramaniam, V.; Du, Y.; Tenenbaum, J. B.; Torralba, A.; Li, S.; and Mordatch, I. 2025. Multiagent finetuning: Self improvement with diverse reasoning chains. *arXiv preprint arXiv:2501.05707*.

Suzgun, M.; Scales, N.; et al. 2022. Challenging BIG-Bench Tasks and Whether Chain-of-Thought Can Solve Them. *arXiv preprint arXiv:2210.09261*.

Von Luxburg, U. 2007. A tutorial on spectral clustering. *Statistics and computing*, 17: 395–416.

Vryn, M. F.; and Das, M. 2025. Building Community–Centered AI Collaborations. *Stanford Social Innovation Review*.

Wan, Z.; Li, Y.; Song, Y.; Wang, H.; Yang, L.; Schmidt, M.; Wang, J.; Zhang, W.; Hu, S.; and Wen, Y. 2025. Rema: Learning to meta-think for llms with multi-agent reinforcement learning. *arXiv preprint arXiv:2503.09501*.

Wang, J.; Wang, J.; Athiwaratkun, B.; Zhang, C.; and Zou, J. 2024. Mixture-of-agents enhances large language model capabilities. *arXiv preprint arXiv:2406.04692*.

Wang, X.; Wei, J.; Schuurmans, D.; Le, Q.; Chi, E.; Narang, S.; Chowdhery, A.; and Zhou, D. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

Wang, Y.; Ma, X.; et al. 2024. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *arXiv preprint arXiv:2406.01574*.

Ward Jr, J. H. 1963. Hierarchical grouping to optimize an objective function. *Journal of the American statistical association*, 58(301): 236–244.

Xie, C.; Huang, Y.; Zhang, C.; Yu, D.; Chen, X.; Lin, B. Y.; Li, B.; Ghazi, B.; and Kumar, R. 2024. On memorization of large language models in logical reasoning. *arXiv preprint arXiv:2410.23123*.

Zhang, H.; Cui, Z.; Wang, X.; Zhang, Q.; Wang, Z.; Wu, D.; and Hu, S. 2025. If Multi-Agent Debate is the Answer, What is the Question? *arXiv preprint arXiv:2502.08788*.

Zhang, K.; Zeng, S.; et al. 2024. Ultramedical: Building specialized generalists in biomedicine. *Advances in Neural Information Processing Systems*, 37: 26045–26081.

Zhang, T.; Ramakrishnan, R.; and Livny, M. 1996. BIRCH: an efficient data clustering method for very large databases. *ACM sigmod record*, 25(2): 103–114.

Zhang, Y.-K.; Zhan, D.-C.; and Ye, H.-J. 2025. Capability Instruction Tuning: A New Paradigm for Dynamic LLM Routing. *arXiv preprint arXiv:2502.17282*.

Zheng, S.; Wang, H.; Huang, C.; Wang, X.; Chen, T.; Fan, J.; Hu, S.; and Ye, P. 2025. Decouple and Orthogonalize: A Data-Free Framework for LoRA Merging. *arXiv preprint arXiv:2505.15875*.

Zhuang, R.; Wu, T.; Wen, Z.; Li, A.; Jiao, J.; and Ramchandran, K. 2024. EmbedLLM: Learning Compact Representations of Large Language Models. *arXiv preprint arXiv:2410.02223*.