

# Beyond GPT-5: Making LLMs Cheaper and Better via Performance-Efficiency Optimized Routing

Yiqun Zhang<sup>\*,†</sup>, Hao Li<sup>\*</sup>, Jianhao Chen<sup>\*</sup>, Hangfan Zhang<sup>\*</sup>, Peng Ye, Lei Bai, Shuyue Hu<sup>‡,†</sup>  
 zhangyiqun344@gmail.com, {lihao4, chenjianhao, zhanghangfan, yepeng, bailei, hushuyue}@pjlab.org.cn  
 Shanghai Artificial Intelligence Laboratory  
 Shanghai, China

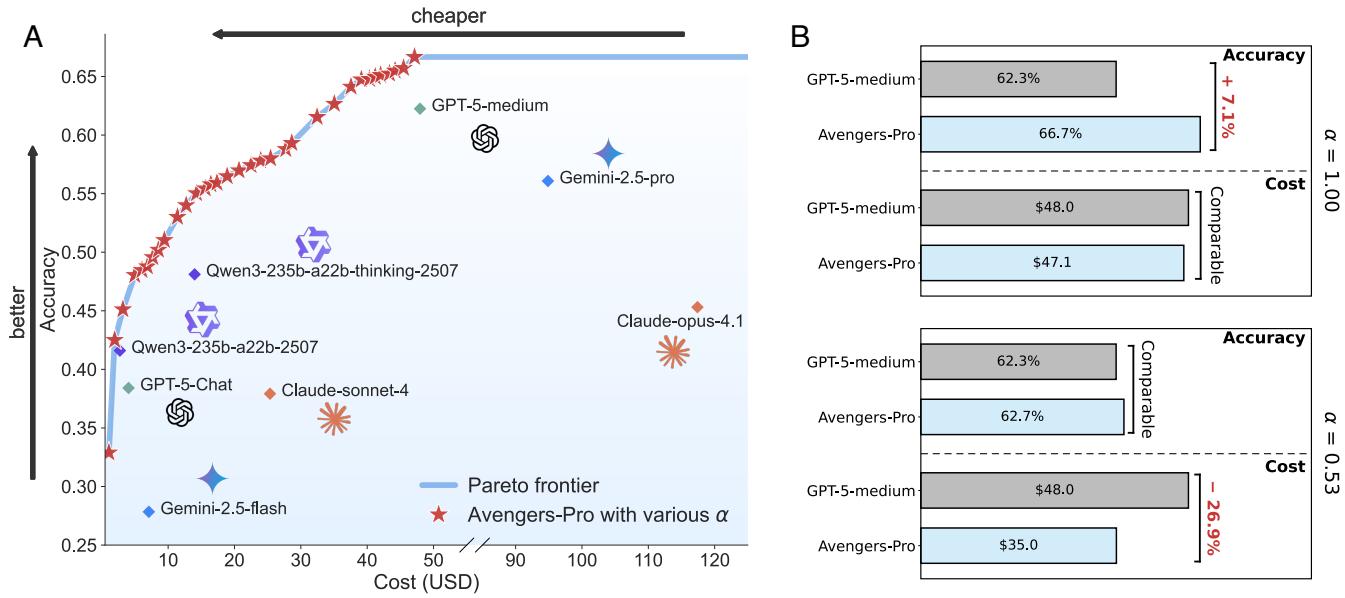


Figure 1: *Avengers-Pro* optimizes the trade-off between performance (accuracy) and efficiency (cost). (A) By varying a trade-off parameter  $\alpha$ , *Avengers-Pro* establishes a Pareto frontier. Compared to all single models, it achieves the highest accuracy for any given cost, and achieves the lowest cost for any given accuracy. (B) With comparable cost, *Avengers-Pro* outperforms the strongest single model GPT-5-medium by 7.1%. With comparable performance, *Avengers-Pro* achieves a 26.9% cost reduction compared to GPT-5-medium.

## Abstract

Balancing performance and efficiency is a central challenge in large language model (LLM) advancement. GPT-5 addresses this with test-time routing, dynamically assigning queries to either an efficient or a high-capacity model during inference. In this work, we present *Avengers-Pro*, a test-time routing framework that ensembles LLMs of varying capacities and efficiencies, providing a unified solution for all performance-efficiency tradeoffs. The *Avengers-Pro* embeds and clusters incoming queries, then routes each to the most suitable

model based on a performance-efficiency score. Across 6 challenging benchmarks and 8 leading models—including GPT-5-medium, Gemini-2.5-pro, and Claude-opus-4.1—*Avengers-Pro* achieves state-of-the-art results: by varying a performance-efficiency trade-off parameter, it can **surpass the strongest single model** (GPT-5-medium) by **+7% in average accuracy**. Moreover, it can **match** the average accuracy of the strongest single model at **27% lower cost**, and reach ~90% of that performance at **63% lower cost**. Last but not least, it achieves a Pareto frontier, consistently yielding the highest accuracy for any given cost, and the lowest cost for any given accuracy, among all single models. Code is available at <https://github.com/ZhangYiqun018/AvengersPro>.

## Keywords

Large Language Models, Model Routing, Cost-effective, Multi-objective Optimization

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

DAI, London

© 2025 Copyright held by the owner/author(s). Publication rights licensed to ACM.  
 ACM ISBN 978-x-xxxx-xxxx-x/YYY/MM  
<https://doi.org/10.1145/3772429.3772445>

<sup>\*</sup>Equal contribution.

<sup>†</sup>Corresponding author.

<sup>‡</sup>Project lead, [hushuyue@pjlab.org.cn](mailto:hushuyue@pjlab.org.cn), [zhangyiqun344@gmail.com](mailto:zhangyiqun344@gmail.com)

## 1 Introduction

A fundamental dilemma in LLM advancement is the trade-off between performance and efficiency. To navigate this, a defining feature of GPT-5 is its *test-time routing* between models. As described in *Introducing GPT-5*<sup>1</sup>:

“GPT-5 is a unified system with a **smart, efficient** model that answers most questions, a **deeper reasoning** model (GPT-5 thinking) for harder problems, and a **real-time router** that quickly decides which to use based on conversation type, complexity ...”

The efficient model offers lower computational cost and latency at the expense of capability, while the deeper reasoning model incurs higher cost and latency but delivers greater capability. During inference, GPT-5’s router dynamically assigns each query to exactly one model, striking a balance between performance and efficiency.

In this work, we advance test-time routing to optimize the performance–efficiency trade-off, and introduce the *Avengers-Pro*. Given a set of models and a set of labeled query-answer pairs, the *Avengers-Pro* operates through three lightweight operations: embedding, clustering and scoring. Specifically, first, it encodes the queries from the dataset using a text embedding model, and then clusters them based on their semantic representations. Next, to assess each model’s capabilities and efficiency, it evaluates each model on the dataset and computes a performance-efficiency score for each cluster. Weighted by a trade-off parameter  $\alpha$ , this score reflects both a model’s performance (measured by its accuracy on the queries within a cluster) and its efficiency (quantified by the cost incurred when answering those queries). During inference, each query is embedded and mapped to its top- $p$  nearest clusters. The model with the highest performance-efficiency score aggregated over those clusters is selected to generate the response.

In our experiments, the *Avengers-Pro* consists of 8 models from 4 families: GPT-5-chat, GPT-5-medium, Claude-4.1-opus, Claude-4-sonnet, Gemini-2.5-pro, Gemini-2.5-flash, Qwen3-235B-A22B-thinking-2507, and Qwen3-235B-A22B-2507. We evaluated *Avengers-Pro* on 6 challenging benchmarks: GPQA-Diamond [26], Human’s Last Exam [25], HealthBench [2], ARC-AGI [7], SimpleQA [31], LiveCodeBench [16], and  $\tau_2$ -bench [3]. We find that compared to the strongest single model GPT-5-medium (average accuracy: 62.25%, cost: \$47.96), the *Avengers-Pro* can attain 7% performance gain with a comparable cost (average accuracy: 66.66, cost: \$47.13), and cut 27% cost with a comparable performance (average accuracy: 62.66, cost: \$35.05). By varying the trade-off parameter  $\alpha$ , the *Avengers-Pro* achieves an even more favorable balance between performance and efficiency. For example, to reach 90% of GPT-5-medium’s performance—a level comparable to Gemini-2.5-pro—the *Avengers-Pro* reduces cost by 63% relative to GPT-5-medium and by 81% relative to Gemini-2.5-pro. Furthermore, we observe that the *Avengers-Pro* achieves a Pareto frontier: for any fixed cost, it consistently delivers the highest performance among all models at that expenditure. Conversely, for any fixed performance target, it provides the lowest cost compared to other models attaining the same accuracy.

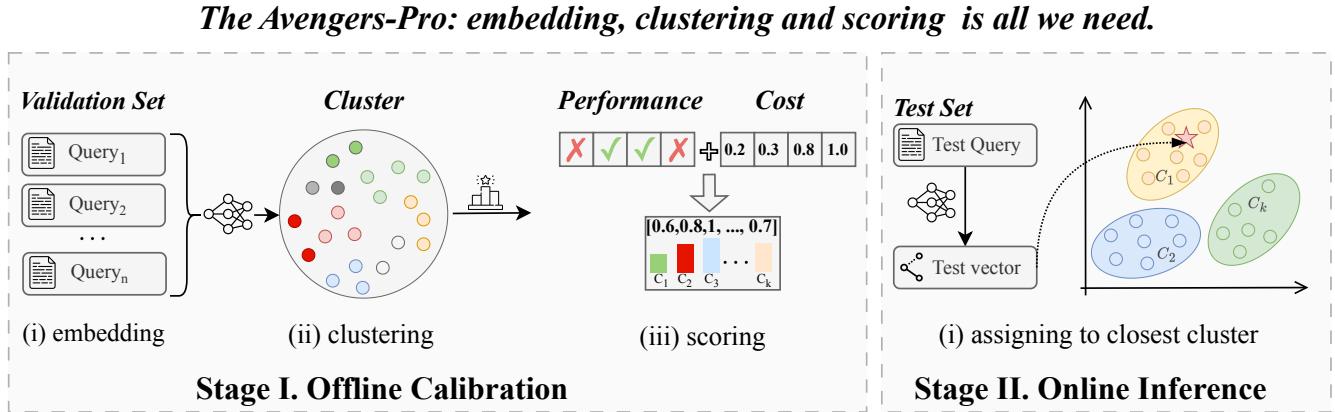
Note that *Avengers-Pro* is not the first study to explore test-time routing. As we will discuss in detail later, there is a growing line of research that leverages router-based methods to harness collective intelligence from multiple models [5, 34, 38]. While most works in this area focus primarily on improving overall performance [5, 38], a few recent studies have started to investigate the trade-off between performance and efficiency by routing among smaller and larger models [14, 19]. Compared to these prior studies, the *Avengers-Pro* stands out for its simplicity and effectiveness. Most previous approaches rely on training additional neural networks and require retraining when incorporating new models. In contrast, *Avengers-Pro* involves only three lightweight operations, requires no neural network training, and is straightforward to implement. Moreover, it is highly reproducible and does not depend on hand-crafted prompts. To incorporate newly available models, it only requires an incremental evaluation of the new models on the dataset. Despite its simplicity, the *Avengers-Pro* is remarkably effective. To our knowledge, this is the first demonstration that test-time routing can be used to surpass state-of-the-art proprietary single models in terms of both efficiency and performance.

## 2 Related Work

The harnessing of collective intelligence from multiple models constitutes one of the frontiers of AI and ML research, and has recently attracted much interest [12, 21, 29, 30, 33, 37]. Existing approaches in this area generally fall into three paradigms: router-based, mixture-based, and merging-based methods. This work is most closely aligned with the router-based paradigm.

The main goal of most router-based methods is to enhance the overall performance of a set of smaller models through query routing; a neural network-based router is often trained to select, for each incoming query, the model most capable of handling it [4, 11, 15, 23, 27, 28]. LLM-Blender [17] utilizes pairwise comparisons to select the top- $k$  models for each query and then fuses their outputs to enhance performance. ZOOTER [22] proposes reward-guided query routing, employing tag-based label enhancement to improve training stability. RouterDC [6], which employs dual contrastive learning to enhance routing accuracy, and EmbedLLM [38] leverages learned compact model embeddings along with query embeddings to predict routing correctness. Additionally, ModelSAT [36] generates capability instructions from model aptitude outcomes and employs text-aligned embeddings to guide a lightweight LLM in selecting optimal candidate models. More recently, few recent studies have also explored routing that account for the trade-off between performance and computational cost. Routellm [23] trains a binary classifier using preference data to dynamically route queries to either a stronger or a weaker LLM during inference. GraphRouter [11] constructs a heterogeneous graph comprising task, query, and LLM nodes, and predicts the performance-cost score via an edge prediction mechanism. RouterBench [14] introduces a suite of benchmarks. Our work is most closely related to [34] and [19], which both employ clustering in their routers. While the former focuses only on performance, the latter considers model efficiency to be a universal cost across different tasks. The *Avengers-Pro* is the first study to show that test-time routing can be used to

<sup>1</sup><https://openai.com/index/introducing-gpt-5/>



**Figure 2: The Avengers Pro, a unified framework for dynamic routing, optimizes performance-efficiency trade-offs by intelligently ensembling language models.**

surpass state-of-the-art proprietary single models in terms of both efficiency and performance.

Driven primarily by efficiency, another related line of research explore single-model *hybrid reasoning* methods that trim latency and token cost by adjusting the amount or stages of “thinking” within one model (e.g., Qwen3 “thinking” mode<sup>2</sup> and DeepSeek-V3.1<sup>3</sup>), or learn to allocate a reasoning budget via policies[13] or test-time guidance[18]. Representative approaches include adaptive “think/non-think” modes, hybrid-reasoning training pipelines, and token-budget or inference-aware policies. While such methods can cut redundant CoT tokens or speed up simple cases, they *cannot* realize cross-model complementarity by design (all computation remains within a single backbone), and their effectiveness is often modest [1, 13, 20].

### 3 Routing for Performance-Efficiency Trade-off

The *Avengers-Pro* ensembles a set of heterogeneous LLMs of varying capabilities and efficiencies with a router. Appropriate routing depends on an accurate understanding of each model’s capability and efficiency across different types of tasks or queries. To build this understanding, the router requires a set  $\mathcal{D}$  of labeled query-answer pairs. Each query  $d \in \mathcal{D}$  is first encoded into a semantic vector using a text **embedding** model. These embeddings are then grouped into  $k$  clusters using a **clustering** algorithm, producing a set  $C = \{c_1, \dots, c_k\}$ , where each cluster represents a semantically coherent query type.

Let  $\mathcal{M}$  denote the set of models in our system. We evaluate each model  $i \in \mathcal{M}$  on  $\mathcal{D}$ , measures its performance and efficiency within each cluster. Let  $\mathbf{p}^i = [p_1^i, \dots, p_k^i]^\top$  be a cluster-wise **performance profile** for model  $i$ , where  $p_j^i$  denotes model  $i$ ’s accuracy on queries within cluster  $c_j$ . Similarly, let  $\mathbf{q}^i = [q_1^i, \dots, q_k^i]^\top$  be a cluster-wise **efficiency profile** for model  $i$ , where  $q_j^i$  denotes model  $i$ ’s efficiency on queries within cluster  $c_j$ . We measure the efficiency in terms

of cost such that  $q_j^i$  denotes the total cost incurred by model  $i$  to answer all queries within cluster  $c_j$ .

We calculate the **cluster-wise performance-efficiency score**  $x_j^i$  for model  $i$  on  $c_j$  by

$$x_j^i = \alpha \tilde{p}_j^i + (1 - \alpha) (1 - \tilde{q}_j^i),$$

where  $\alpha \in [0, 1]$  controls the trade-off between performance and efficiency, and  $\tilde{p}_j^i$  and  $\tilde{q}_j^i$  are the normalized values of  $p_j^i$  and  $q_j^i$ . The normalization is given by

$$\tilde{p}_j^i = \frac{p_j^i - p_j^{\min}}{p_j^{\max} - p_j^{\min}}, \quad \tilde{q}_j^i = \frac{q_j^i - q_j^{\min}}{q_j^{\max} - q_j^{\min}},$$

where  $p_j^{\min}$  and  $p_j^{\max}$  (or  $q_j^{\min}$  and  $q_j^{\max}$ ) denote the minimum and maximum performance (or cost) among all models for cluster  $j$ .

During inference, an incoming query is encoded with the text embedding model, and is assigned to the top- $p$  nearest cluster(s) in the embedding space. For each model  $i \in \mathcal{M}$ , we sum up its cluster-wise performance-efficiency scores over those top- $p$  clusters. The model with the highest sum of those scores is selected to generate the response.

## 4 Experiments

Our experiments compare the performance and efficiency of *Avengers-Pro* against leading single models.

### 4.1 Experimental Settings

**4.1.1 Models.** We consider 8 leading models, which vary in capability and efficiency, as follows:

- **Google:** Gemini-2.5-flash [10], Gemini-2.5-Pro [10].
- **Anthropic:** Claude-4.1-opus [8], Claude-4-sonnet [9].
- **OpenAI:** GPT-5-chat [24]<sup>4</sup>, GPT-5-medium [24]<sup>5</sup>.

<sup>4</sup>GPT-5-chat: points to the GPT-5 snapshot currently used in ChatGPT; OpenAI recommends gpt-5 for most API usage, while gpt-5-chat exposes the latest improvements for chat use cases. Do not support function/tool call now.

<sup>5</sup>GPT-5-medium: GPT-5 is OpenAI’s flagship model for coding, reasoning, and agentic tasks across domains. GPT-5-medium denotes GPT-5 with reasoning\_effort=medium.

<sup>2</sup><https://huggingface.co/Qwen/Qwen3-235B-A22B>

<sup>3</sup><https://huggingface.co/deepseek-ai/DeepSeek-V3.1>

- **Qwen**: Qwen3-235B-A22B-2507 (or Qwen3) [32], Qwen3-235B-A22B-thinking-2507 (or Qwen3-thinking) [32].

We access these models through the OpenRouter API<sup>6</sup>, as its standardized interface simplifies the process of running identical experiments across multiple models. The pricing for these models is detailed in Table 1. Prices for the Qwen3 family may vary across providers; throughout this paper we report the prices listed by OpenRouter.

**Table 1: Model cost information (OpenRouter).**

Model	Input Price (\$/1M tokens)	Output Price (\$/1M tokens)
Gemini-2.5-flash	0.30	2.50
Gemini-2.5-Pro	1.25	10
Claude-4.1-opus	15	75
Claude-4-sonnet	3	15
GPT-5-chat	1.25	10
GPT-5-medium	1.25	10
Qwen3-235B-A22B-2507 <sup>4</sup>	≈0.13	≈0.6
Qwen3-235B-A22B-thinking-2507	≈0.13	≈0.6

**4.1.2 Benchmarks.** We consider 6 challenging benchmarks, as summarized in Table 2, covering advanced reasoning and general knowledge:

**Table 2: Benchmark information.**

Dataset	Metrics	Size
ARC-AGI-v1 [7]	pass@1	200
GPQA-Diamond [26]	pass@1	198
HLE [25]	pass@1	500
LiveCodeBench-v6 [16]	pass@1	1,055
$\tau^2$ -bench [3]	pass@1	150
SimpleQA [31]	pass@1	500
Total		2,603

- **GPQA-Diamond** [26]: A graduate-level google-proof Q&A benchmark.
- **Human’s Last Exam (HLE)** [25]: A frontier multi-modal benchmark of closed-ended academic questions. In this study, we use the *text-only* setting without custom patches, tool use, or retrieval during evaluation. For efficiency and reproducibility, we use the first **500** questions from the released pool and report accuracy under the official evaluation protocol.
- **ARC-AGI** [7]: A benchmark focused on fluid intelligence, testing the ability to reason and solve novel problems. We use the first **200** questions from the released pool and report accuracy under the official evaluation protocol.
- **SimpleQA** [31]: A factuality benchmark for short, fact-seeking questions. We use the *official* implementation with the default configuration and report accuracy under the official scoring. We evaluate on a subset of **500** examples uniformly sampled from the released dataset.

<sup>6</sup><https://openrouter.ai/>

- **LiveCodeBench** [16]: A dynamic, contamination-controlled coding benchmark that continuously ingests newly released problems. We evaluate on the latest public release (**v6**) using the *official* implementation and evaluation harness with the default configuration, without custom patches or post-processing.
- **$\tau^2$ -bench** [3]: A controlled testbed for agents that must reason effectively and guide user actions, we randomly sampled 50 examples from each of the three categories.

For all benchmarks, we use the official implementations with their recommended prompts and hyperparameters; for context length, we set each model to its documented maximum context window and apply no additional truncation, chunking, or sliding-window processing.

**4.1.3 Implementation Details.** We use k-means clustering with  $k = 60$  clusters. Each query is encoded by the *Qwen3-embedding-8B* [35] into a 4,096-dimensional semantic vector. Following common practice in routing [5, 34, 39], we randomly split the data: 70% is used to fit the clustering model and estimate per-cluster statistics, and the remaining 30% is reserved for routing and evaluation. At inference time, we compute the embedding of the incoming query and retrieve the top- $p$  nearest clusters ( $p = 4$ ) in the embedding space. For each model  $i$ , we then sum its cluster-wise cost–capability scores  $q_j^i$  over these three clusters and select the model with the highest total to generate the response.

## 4.2 Results and Analysis

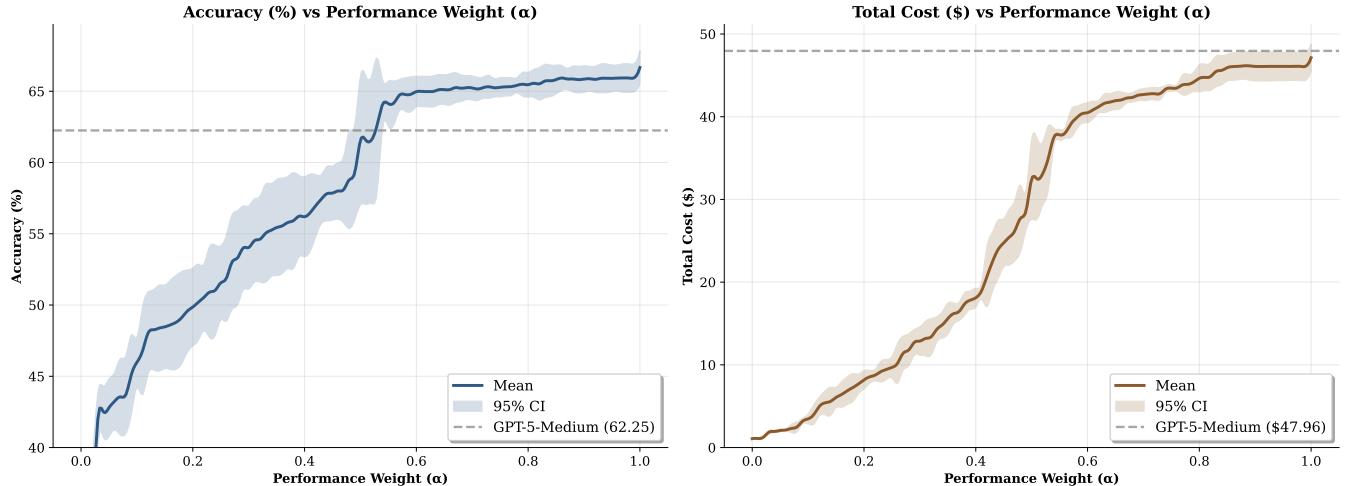
We present the comparisons of *Avengers-Pro* and single models in terms of performance and efficiency in Table 3. We show how the trade-off parameter  $\alpha$  affects the performance and efficiency in Figure 3. We show the proportion of model usage by *Avengers-Pro* in Figure 4.

**4.2.1 Avengers-Pro outperforms top single models.** Of the eight single models evaluated, GPT-5-medium demonstrates the highest average accuracy (62.25%) across the six benchmarks. This is followed by Gemini-2.5-pro (56.08%) and Qwen3-thinking (48.11%), respectively. The *Avengers-Pro* surpasses the performance of *all* individual models with a sufficiently large value of  $\alpha$ , prioritizing performance over efficiency. Specifically, its average accuracy is up to 66.66% with  $\alpha = 1.0$ , which is 7% higher compared to GPT-5-medium and 19% higher compared to Gemini-2.5-pro.

**4.2.2 Avengers-Pro achieves a superior performance-efficiency trade-off.** At a performance level comparable to the strongest single model GPT-5-medium, *Avengers-Pro* ( $\alpha = 0.53$ ) incurs significantly lower costs, resulting in a cost reduction of 27%. Similarly, at a 90% performance level of GPT-5-medium, the *Avengers-Pro* ( $\alpha = 0.39$ ) cuts cost by 63%. At a performance level comparable to the second-strongest single model Gemini-2.5-pro, it ( $\alpha = 0.39$ ) reduces cost by 81%. At a performance level comparable to Claude-4.1-opus, it ( $\alpha = 0.25$ ) achieves a cost reduction of 92%. Moreover, as shown in Figure 1A, the *Avengers-Pro* achieves a Pareto frontier—no single model can simultaneously deliver higher performance and greater efficiency than *Avengers-Pro*. In other words, *Avengers-Pro* offers the highest performance for any given cost and the lowest cost for any given level of performance.

**Table 3: The performance and efficiency of *Avengers-Pro* vs. single models.** Note that GPT-5-chat has no score on the  $\tau^2$ -bench benchmark because this model does not support tool calling. **Bold** indicates the best performance of a given benchmark, and **underline** indicates the second-best. With  $\alpha = 0.1$ , *Avengers-Pro*, surpasses GPT-5-medium in average accuracy with a 7% performance gain. With  $\alpha = 0.53$ , it matches GPT-5-medium’s average accuracy, while cutting the cost by 27%. With  $\alpha = 0.39$ , it reaches 90% of GPT-5-medium’s performance at a 63% lower cost.

Setting	ARC-AGI	GPQA-Diamond	HLE	LiveCodeBench	SimpleQA	$\tau^2$ -bench	Avg. A	Cost
<b>Gemini-2.5-flash</b>	9.62	21.72	7.20	62.84	28.99	36.67	27.84	\$7.10
<b>Gemini-2.5-pro</b>	33.08	84.85	23.09	78.67	54.80	62.00	56.08	\$94.87
<b>Claude-4.1-opus</b>	22.12	74.24	6.41	64.07	31.00	74.00	45.31	\$117.40
<b>Claude-4-sonnet</b>	16.15	68.69	4.60	59.05	15.00	64.00	37.92	\$25.35
<b>Qwen3</b>	9.22	58.59	9.22	66.26	53.00	53.33	41.60	\$2.73
<b>Qwen3-thinking</b>	19.23	80.81	12.68	77.99	44.60	53.33	48.11	\$13.99
<b>GPT-5-chat</b>	6.73	73.73	7.80	63.60	40.20	-	38.41	\$4.04
<b>GPT-5-medium</b>	44.42	<u>84.85</u>	26.20	88.44	47.60	<b>82.00</b>	62.25	\$47.96
<b>Avengers Pro (<math>\alpha = 0</math>)</b>	15.33	58.67	10.13	66.94	46.27	0.00	32.89	\$1.08
<b>Avengers Pro (<math>\alpha = 0.25</math>)<sup>1</sup></b>	29.33	67.00	10.00	76.53	53.60	72.89	51.56	\$9.69
<b>Avengers Pro (<math>\alpha = 0.39</math>)<sup>2</sup></b>	29.33	78.67	12.67	84.79	55.07	76.89	56.24	\$17.81
<b>Avengers Pro (<math>\alpha = 0.53</math>)<sup>3</sup></b>	<u>51.67</u>	80.00	25.46	87.45	54.93	76.44	62.66	\$35.05
<b>Avengers Pro (<math>\alpha = 0.8</math>)</b>	<b>59.67</b>	81.00	<u>27.60</u>	<u>89.34</u>	<b>56.93</b>	78.22	<u>65.46</u>	\$44.65
<b>Avengers Pro (<math>\alpha = 1</math>)</b>	<b>59.67</b>	<b>85.67</b>	<b>28.67</b>	<b>89.59</b>	56.40	<u>80.00</u>	<b>66.66</b>	\$47.13

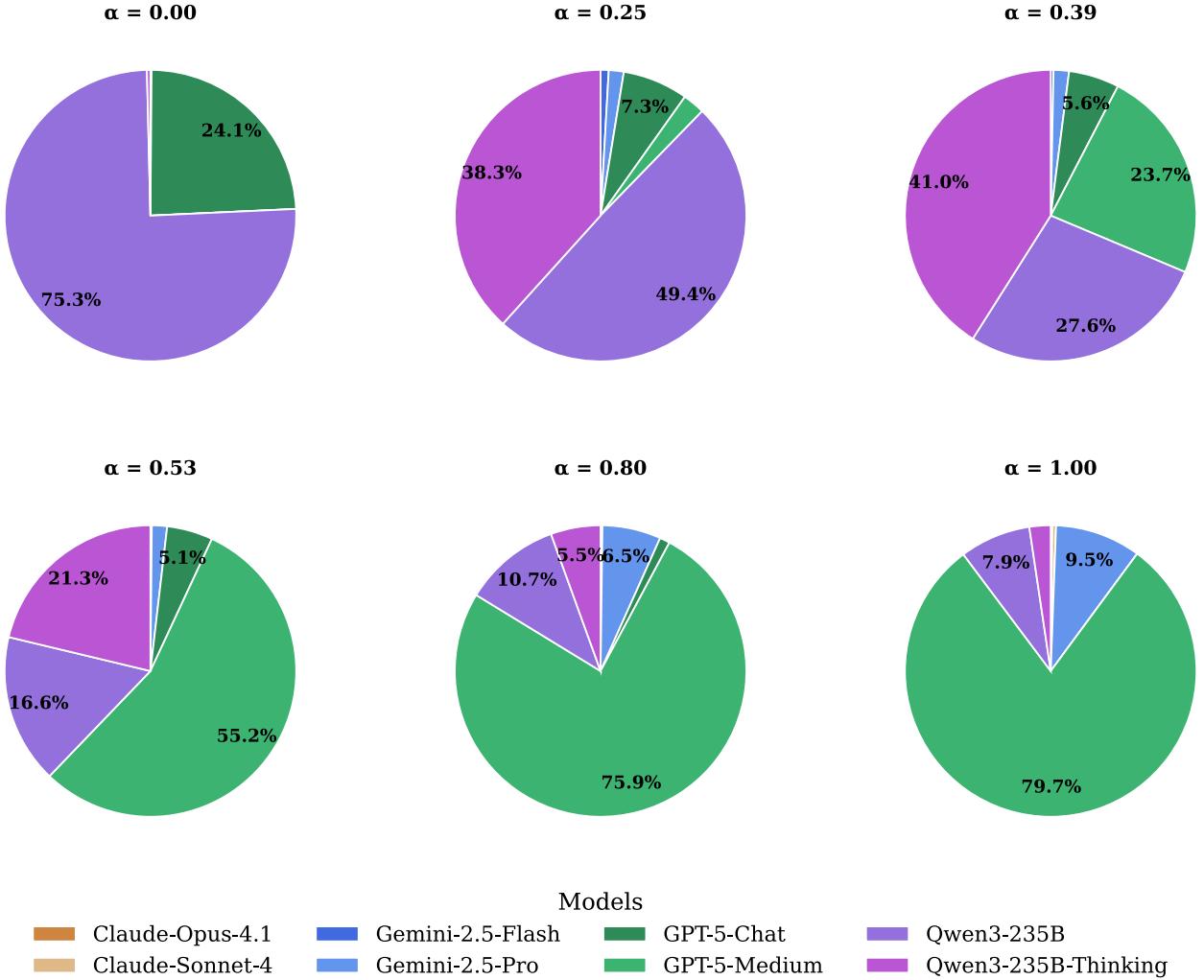


**Figure 3: Effects of the trade-off parameter  $\alpha$  on the performance and efficiency. A greater value of  $\alpha$  prioritizes performance over efficiency. The increase in performance is usually accompanied by the increase in cost.**

**4.2.3 Effects of the trade-off parameter.** As shown in Figure 3, we gradually increase the trade-off parameter  $\alpha$ , placing more weight on performance. Accuracy (left) climbs quickly for small  $\alpha$  and then saturates around  $\alpha \approx 0.6$ ; the shaded 95% CI also narrows as  $\alpha$  grows, indicating more stable outcomes once the router consistently calls stronger models. Cost (right) stays near its minimum up to  $\alpha \approx 0.4$ , then rises steeply before tapering off. The gray dashed baselines mark GPT-5-medium (62.25% accuracy; \$47.96 cost): our curve exceeds the accuracy baseline already at moderate  $\alpha$  while remaining below the cost baseline across a broad range before very high  $\alpha$ . Taken together, the curves reveal two elbows—around

$\alpha \approx 0.4$  and  $\alpha \approx 0.6$ —that correspond to regime changes in routing: (i) *efficiency-first* ( $\alpha \lesssim 0.4$ ), where the router mostly selects cheaper models and yields low cost with moderate accuracy; (ii) a *transition band* ( $0.4 < \alpha < 0.6$ ), where accuracy improves rapidly per marginal dollar as the router begins invoking stronger models on harder clusters; and (iii) *performance-first* ( $\alpha \gtrsim 0.6$ ), where accuracy is near its ceiling and additional  $\alpha$  mainly buys cost increases.

**4.2.4 Proportion of model usage.** As shown in Figure 4, when  $\alpha$  is low, *Avengers-Pro* tends to favor the Qwen3 and Qwen3-thinking model, routing a great proportion of queries to these two models



**Figure 4: Proportion of model usage, given various trade-off parameters  $\alpha$ . When  $\alpha$  is low, *Avengers-Pro* tend to route queries to Qwen3 and Qwen3-thinking. With a greater value of  $\alpha$ , *Avengers-Pro* favors GPT5-medium and Qwen3-thinking.**

with a low unit price. As  $\alpha$  increases, the usage of GPT-5-medium rises rapidly; concurrently, the usage of Gemini-2.5-pro and Claude-opus-4.1, which excel at complex reasoning but have a higher unit price, also increases. Consistent with Figure 1, model usage correlates with proximity to *Avengers-Pro*'s Pareto frontier: models closer to the frontier (GPT-series, Qwen3-series) see higher utilization, whereas those farther away (claude-series, gemini-series) are selected less frequently for a given  $\alpha$ .

## 5 Conclusions

In this work, we introduce *Avengers-Pro*, a test-time routing framework integrating different LLMs to optimize the trade-off between performance and efficiency. By dynamically selecting exactly one model for each incoming query, *Avengers-Pro* optimizes both cost and accuracy. Our experiments involving 8 leading LLMs and 6

challenging benchmarks demonstrate that *Avengers-Pro* can surpass the strongest single model, GPT-5-medium, by up to 7% in accuracy and match its performance at a 27% lower expense. Moreover, *Avengers-Pro* achieves a Pareto frontier, consistently delivering the best performance on any given budget and the lowest cost given any performance target. Our results highlight the significant potential of an intelligent test-time routing framework in creating more powerful, efficient, and scalable LLM systems.

## Acknowledgments

This work was supported by a locally commissioned task from the Shanghai Municipal Government. The authors thank Dr. Qiaosheng Zhang for helpful discussions, and collaborators who contributed meaningfully to our earlier project the *Avengers* [34]. This work was supported by the National Natural Science Foundation of China (Grant No. 6250076060).

## References

- [1] Mohammad Ali Alomrani, Yingxue Zhang, Derek Li, Qianyi Sun, Soumyasundar Pal, Zhanguang Zhang, Yaochen Hu, Rohan Deepak Ajwani, Antonios Valkanas, Raika Karimi, et al. 2025. Reasoning on a Budget: A Survey of Adaptive and Controllable Test-Time Compute in LLMs. *arXiv preprint arXiv:2507.02076* (2025).
- [2] Rahul K Arora, Jason Wei, Rebecca Soskin Hicks, Preston Bowman, Joaquin Quiñonero-Candela, Foivos Tsimpourlas, Michael Sharman, Meghan Shah, Andrea Vallone, Alex Beutel, et al. 2025. Healthbench: Evaluating large language models towards improved human health. *arXiv preprint arXiv:2505.08775* (2025).
- [3] Victor Barres, Honghua Dong, Sohan Ray, Xujie Si, and Karthik Narasimhan. 2025.  $\tau^2$ -Bench: Evaluating Conversational Agents in a Dual-Control Environment. *arXiv preprint arXiv:2506.07982* (2025).
- [4] Lingjiao Chen, Matei Zaharia, and James Zou. 2023. FrugalGPT: How to Use Large Language Models While Reducing Cost and Improving Performance. *Transactions on Machine Learning Research* (2023).
- [5] Shuhao Chen, Weisen Jiang, Baijiong Lin, James Kwok, and Yu Zhang. 2024. RouterDC: Query-Based Router by Dual Contrastive Learning for Assembling Large Language Models. In *Advances in Neural Information Processing Systems*, A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (Eds.), Vol. 37. Curran Associates, Inc., 66305–66328.
- [6] Shuhao Chen, Weisen Jiang, Baijiong Lin, James Kwok, and Yu Zhang. 2024. Routerdc: Query-based router by dual contrastive learning for assembling large language models. *Advances in Neural Information Processing Systems* 37 (2024), 66305–66328.
- [7] Francois Chollet, Mike Knoop, Gregory Kamradt, and Bryan Landers. 2024. Arc prize 2024: Technical report. *arXiv preprint arXiv:2412.04604* (2024).
- [8] Claude. 2025. System Card Addendum: Claude Opus 4. [www.anthropic.com/news/clause-opus-4-1](http://www.anthropic.com/news/clause-opus-4-1) (2025).
- [9] Claude. 2025. System Card: Claude Opus 4 & Claude Sonnet 4. [www.anthropic.com/clause/sonnet](http://www.anthropic.com/clause/sonnet) (2025).
- [10] Gheorghe Comanici, Eric Bieber, Mike Schaeckermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blstein, Ori Ram, Dan Zhang, Evan Rosen, et al. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261* (2025).
- [11] Tao Feng, Yanzhen Shen, and Jiaxuan You. 2025. GraphRouter: A Graph-based Router for LLM Selections. In *The Thirteenth International Conference on Learning Representations*. <https://openreview.net/forum?id=eU39PDsZT>
- [12] Taicheng Guo, Xiuying Chen, Yaqi Wang, Ruidi Chang, Shichao Pei, Nitesh V Chawla, Olaf Wiest, and Xiangliang Zhang. 2024. Large language model based multi-agents: A survey of progress and challenges. *arXiv preprint arXiv:2402.01680* (2024).
- [13] Tingxu Han, Zhenting Wang, Chunrong Fang, Shiyu Zhao, Shiqing Ma, and Zhenyu Chen. 2025. Token-Budget-Aware LLM Reasoning. *arXiv:2412.18547 [cs.CL]* <https://arxiv.org/abs/2412.18547>
- [14] Qitian Jason Hu, Jacob Bieker, Xiuyu Li, Nan Jiang, Benjamin Keigwin, Gaurav Ranganath, Kurt Keutzer, and Shriyash Kaustubh Upadhyay. 2024. RouterBench: A Benchmark for Multi-LLM Routing System. In *Agentic Markets Workshop at ICML 2024*.
- [15] Zhongzhan Huang, Guoming Ling, Vincent S Liang, Yupei Lin, Yandong Chen, Shanshan Zhong, Hefeng Wu, and Liang Lin. 2025. RouterEval: A Comprehensive Benchmark for Routing LLMs to Explore Model-level Scaling Up in LLMs. *arXiv preprint arXiv:2503.10657* (2025).
- [16] Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. 2024. Livecodebench: Holistic and contamination free evaluation of large language models for code. *arXiv preprint arXiv:2403.07974* (2024).
- [17] Dongfu Jiang, Xiang Ren, and Bill Yuchen Lin. 2023. LLM-Blender: Ensembling Large Language Models with Pairwise Ranking and Generative Fusion. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 14165–14178.
- [18] Lingjie Jiang, Xun Wu, Shaohan Huang, Qingxiu Dong, Zewen Chi, Li Dong, Xinxing Zhang, Tengchao Lv, Lei Cui, and Furu Wei. 2025. Think only when you need with large hybrid-reasoning models. *arXiv preprint arXiv:2505.14631* (2025).
- [19] Wittawat Jitkrittum, Harikrishna Narasimhan, Ankit Singh Rawat, Jeevesh Juneja, Zifeng Wang, Chen-Yu Lee, Pradeep Shenoy, Rina Panigrahy, Aditya Krishna Menon, and Sanjiv Kumar. 2025. Universal model routing for efficient llm inference. *arXiv preprint arXiv:2502.08773* (2025).
- [20] Yu Kang, Xianghui Sun, Liangyu Chen, and Wei Zou. 2025. C3ot: Generating shorter chain-of-thought without compromising effectiveness. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 39. 24312–24320.
- [21] Jinliang Lu, Ziliang Pang, Min Xiao, Yaochen Zhu, Rui Xia, and Jiajun Zhang. 2024. Merge, ensemble, and cooperate! a survey on collaborative strategies in the era of large language models. *arXiv preprint arXiv:2407.06089* (2024).
- [22] Keming Lu, Hongyi Yuan, Runji Lin, Junyang Lin, Zheng Yuan, Chang Zhou, and Jingren Zhou. 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.
- [23] Isaac Ong, Amjad Almahairi, Vincent Wu, Wei-Lin Chiang, Tianhao Wu, Joseph E Gonzalez, M Waleed Kadous, and Ion Stoica. 2024. RoutELLM: Learning to Route LLMs from Preference Data. In *The Thirteenth International Conference on Learning Representations*.
- [24] OpenAI. 2025. GPT-5 System Card. [openai.com/index/gpt-5-system-card](http://openai.com/index/gpt-5-system-card) (2025).
- [25] Long Phan, Alice Gatti, Ziwen Han, Nathaniel Li, Josephina Hu, Hugh Zhang, Chen Bo Calvin Zhang, Mohamed Shaaban, John Ling, Sean Shi, et al. 2025. Humanity's last exam. *arXiv preprint arXiv:2501.14249* (2025).
- [26] David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. 2024. Gpq: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*.
- [27] Zejiang Shen, Hunter Lang, Bailin Wang, Yoon Kim, and David Sontag. 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.
- [28] Tal Shnitzer, Anthony Ou, Mirian Silva, Kate Soule, Yuekai Sun, Justin Solomon, Neil Thompson, and Mikhail Yurochkin. 2023. Large Language Model Routing with Benchmark Datasets. In *First Conference on Language Modeling*.
- [29] Vighnesh Subramaniam, Yilun Du, Joshua B Tenenbaum, Antonio Torralba, Shuang Li, and Igor Mordatch. 2025. Multiagent finetuning: Self improvement with diverse reasoning chains. *arXiv preprint arXiv:2501.05707* (2025).
- [30] Ziyu Wan, Yunxiang Li, Yan Song, Hanjing Wang, Linyi Yang, Mark Schmidt, Jun Wang, Weinan Zhang, Shuyue Hu, and Ying Wen. 2025. Rema: Learning to meta-think for llms with multi-agent reinforcement learning. *arXiv preprint arXiv:2503.09501* (2025).
- [31] Jason Wei, Nguyen Karina, Hyung Won Chung, Yunxin Joy Jiao, Spencer Papay, Amelia Glasee, John Schulman, and William Fedus. 2024. Measuring short-form factuality in large language models. *arXiv preprint arXiv:2411.04368* (2024).
- [32] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengan Huang, Chenxu Lv, et al. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388* (2025).
- [33] Hangfan Zhang, Zhiyao Cui, Xinrun Wang, Qiaosheng Zhang, Zhen Wang, Dinghao Wu, and Shuyue Hu. 2025. If Multi-Agent Debate is the Answer, What is the Question? *arXiv preprint arXiv:2502.08788* (2025).
- [34] Yiqun Zhang, Hao Li, Chenxu Wang, Linyao Chen, Qiaosheng Zhang, Peng Ye, Shi Feng, Daling Wang, Zhen Wang, Xinrun Wang, et al. 2025. The Avengers: A Simple Recipe for Uniting Smaller Language Models to Challenge Proprietary Giants. *arXiv preprint arXiv:2505.19797* (2025).
- [35] Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. 2025. Qwen3 Embedding: Advancing Text Embedding and Reranking Through Foundation Models. *arXiv:2506.05176 [cs.CL]* <https://arxiv.org/abs/2506.05176>
- [36] Yi-Kai Zhang, De-Chuan Zhan, and Han-Jia Ye. 2025. Capability Instruction Tuning: A New Paradigm for Dynamic LLM Routing. *arXiv preprint arXiv:2502.17282* (2025).
- [37] Shenghe Zheng, Hongzhi Wang, Chenyu Huang, Xiaohui Wang, Tao Chen, Jiayuan Fan, Shuyue Hu, and Peng Ye. 2025. Decouple and Orthogonalize: A Data-Free Framework for LoRA Merging. *arXiv preprint arXiv:2505.15875* (2025).
- [38] Richard Zhuang, Tianhao Wu, Zhaojin Wen, Andrew Li, Jiantao Jiao, and Kannan Ramchandran. 2024. EmbedLLM: Learning Compact Representations of Large Language Models. *arXiv preprint arXiv:2410.02223* (2024).
- [39] Richard Zhuang, Tianhao Wu, Zhaojin Wen, Andrew Li, Jiantao Jiao, and Kannan Ramchandran. 2024. EmbedLLM: Learning Compact Representations of Large Language Models. *arXiv:2410.02223 [cs.CL]* <https://arxiv.org/abs/2410.02223>

Received 22 August 2025; accepted 18 September 2025