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

Yiqun Zhang*†, Hao Li*, Jianhao Chen*, Hangfan Zhang*, Peng Ye, Lei Bai, Shuyue Hu†

Shanghai Artificial Intelligence Laboratory

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 $\sim$**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`.

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*[2]:

> "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. We build upon our earlier work *Avengers* [15]—which showed that a simple routing recipe using ten models ($\sim$7B parameters each) surpass GPT-4.1 and 4.5 across 15 datasets—and introduce the

*Equal contributions. †Project lead, hushuyue@pjlab.org.cn, zhangyiqun344@gmail.com.

[2]https://openai.com/index/introducing-gpt-5/

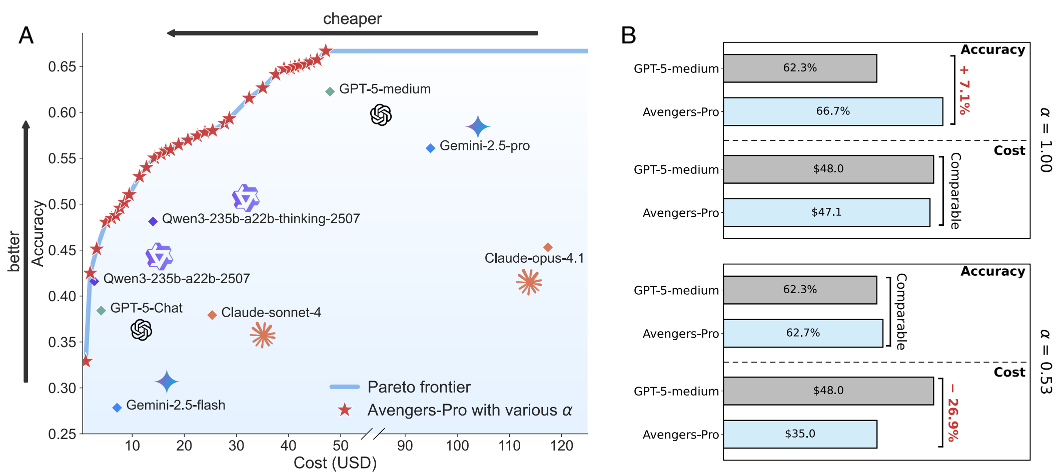

Figure 1: *Avengers-Pro* optimizes the trade-off between performance (accuracy) and efficiency (cost). **(A)** By varying a trade-off parameter α, *Avengers-Pro* establishes a Pareto frontier. Compared to all single models, it achieves the highest accuracy for any given cost, and achieves the lowest 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.

Avengers-Pro. With a focus on performance-efficiency trade-off, the *Avengers-Pro* operates through three lightweight operations: (i) embedding: encode queries using a text embedding model, (ii) clustering: group queries by semantic similarity, and (iii) scoring: evaluate models within each cluster based on a performance-efficiency score weighted by a trade-off parameter α. 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 evaluate the *Avengers-Pro* on 6 challenging benchmarks: GPQA-Diamond [11], Human's Last Exam [10], HealthBench [1], ARC-AGI [4], SimpleQA [12], LiveCodeBench [8], and τ2-bench [2]. 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 α, 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.

2 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 $\mathcal{C} = \{c_1, \ldots, c_k\}$, 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 $\mathbf{p}^i = [p^i_1, \ldots, p^j_k]^\top$ be a cluster-wise **per-**

formance profile for model i, where p_j^i denotes model i's accuracy on queries within cluster c_j. Similarly, let $\mathbf{q}^i = [q_1^i, \ldots, q_k^j]^\top$ 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.

3 Experiments

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

3.1 Experimental Settings

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

- **Google**: Gemini-2.5-flash [7], Gemini-2.5-Pro [7].
- **Anthropic**: Claude-4.1-opus [5], Claude-4-sonnet [6].
- **OpenAI**: GPT-5-chat [9], GPT-5-medium [9].
- **Qwen**: Qwen3-235B-A22B-2507 (or Qwen3) [13], Qwen3-235B-A22B-thinking-2507 (or Qwen3-thinking) [13].

We access these models through the OpenRouter API[3], 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.

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

- **GPQA-Diamond** [11]: A graduate-level google-proof Q&A benchmark.
- **Human's Last Exam (HLE)** [10]: 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** [4]: 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** [12]: 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.

[3] https://openrouter.ai/

- **LiveCodeBench** [8]: 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.
- τ^2**-bench** [2]: A controlled testbed for agents that must reason effectively and guide user actions.

For all benchmarks, we use the official repositories/implementations with their recommended parameter settings.

Implementation Details We use k-means clustering with $k = 60$ clusters. Each query is encoded by the Qwen3-embedding-8B [14] into a 4,096-dimensional semantic vector. Following common practice in routing [3, 16, 15], 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.

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-25074	≈0.13	≈0.6
Qwen3-235B-A22B-thinking-2507	≈0.13	≈0.6

Table 2: Benchmark information.

Dataset	**Metrics**	**Size**
ARC-AGI-v1 [4]	pass@1	200
GPQA-Diamond [11]	pass@1	198
HLE [10]	pass@1	500
LiveCodeBench-v6 [8]	pass@1	1,055
τ^2-bench [2]	pass@1	150
SimpleQA [12]	pass@1	500
Total		2,603

3.2 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 α affects the performance and efficiency in Figure 2. We show the proportion of model usage by *Avengers-Pro* in Figure 3.

***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 α, 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.

***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 Cluade-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.

Effects of the trade-off parameter As shown in Figure 2, we gradually increase the trade-off parameter α, placing more weight on performance over efficiency. As α increases, the average accuracy increases rapidly for small α and then plateaus near $\alpha \approx 0.6$. On the other hand, as α increases, cost remains low until about $\alpha \approx 0.4$ before rising sharply. These trends reveal two elbows (around 0.4 and 0.6) that offer favorable trade-offs.

Table 3: The performance and efficiency of *Avengers-Pro* vs. single models. Note that GPT-5-chat has no score on the τ^2-bench benchmark because this model does not support tool calling. **Bold** indicates the best performance of a given benchmark. 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	τ^2-bench	Avg. A	Cost
Gemini-2.5-flash	9.62	21.72	7.20	62.84	28.99	36.67	27.84	$7.10
Gemini-2.5-pro	33.08	84.85	23.09	78.67	54.80	62.00	56.08	$94.87
Claude-4.1-opus	22.12	74.24	6.41	64.07	31.00	74.00	45.31	$117.40
Claude-4-sonnet	16.15	68.69	4.60	59.05	15.00	64.00	37.92	$25.35
Qwen3	9.22	58.59	9.22	66.26	53.00	53.33	41.60	$2.73
Qwen3-thinking	19.23	80.81	12.68	77.99	44.60	53.33	48.11	$13.99
GPT-5-chat	6.73	73.73	7.80	63.60	40.20	-	38.41	$4.04
GPT-5-medium	44.42	84.85	26.20	88.44	47.60	**82.00**	62.25	$47.96
Avengers Pro ($\alpha = 0$)	15.33	58.67	10.13	66.94	46.27	0.00	32.89	$1.08
Avengers Pro ($\alpha = 0.25$)[1]	29.33	67.00	10.00	76.53	53.60	72.89	51.56	$9.69
Avengers Pro ($\alpha = 0.39$)[2]	29.33	78.67	12.67	84.79	55.07	76.89	56.24	$17.81
Avengers Pro ($\alpha = 0.53$)[3]	51.67	80.00	25.46	87.45	54.93	76.44	62.66	$35.05
Avengers Pro ($\alpha = 0.8$)	**59.67**	81.00	27.60	89.34	**56.93**	78.22	65.46	$44.65
Avengers Pro ($\alpha = 1$)	**59.67**	**85.67**	**28.67**	**89.59**	56.40	80.00	**66.66**	$47.13

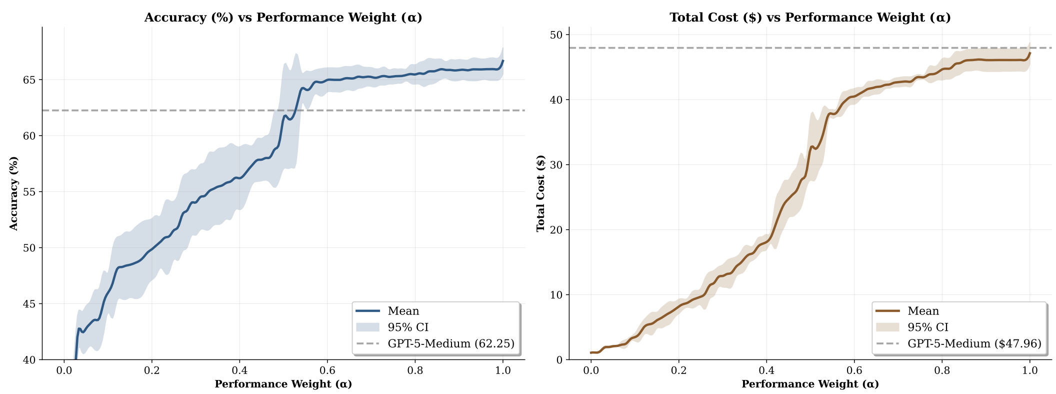

Figure 2: Effects of the trade-off parameter α on the performance and efficiency. A greater value of α prioritizes performance over efficiency. The increase in performance is usually accompanied the increase in cost.

Proportion of model usage As shown in Figure 3, when α is low, *Avengers-Pro* tends to favor the Qwen3 and Qwen3-thinking model, routing a great proportion of queries to these two models with a low unit price. As α 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.

4 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.

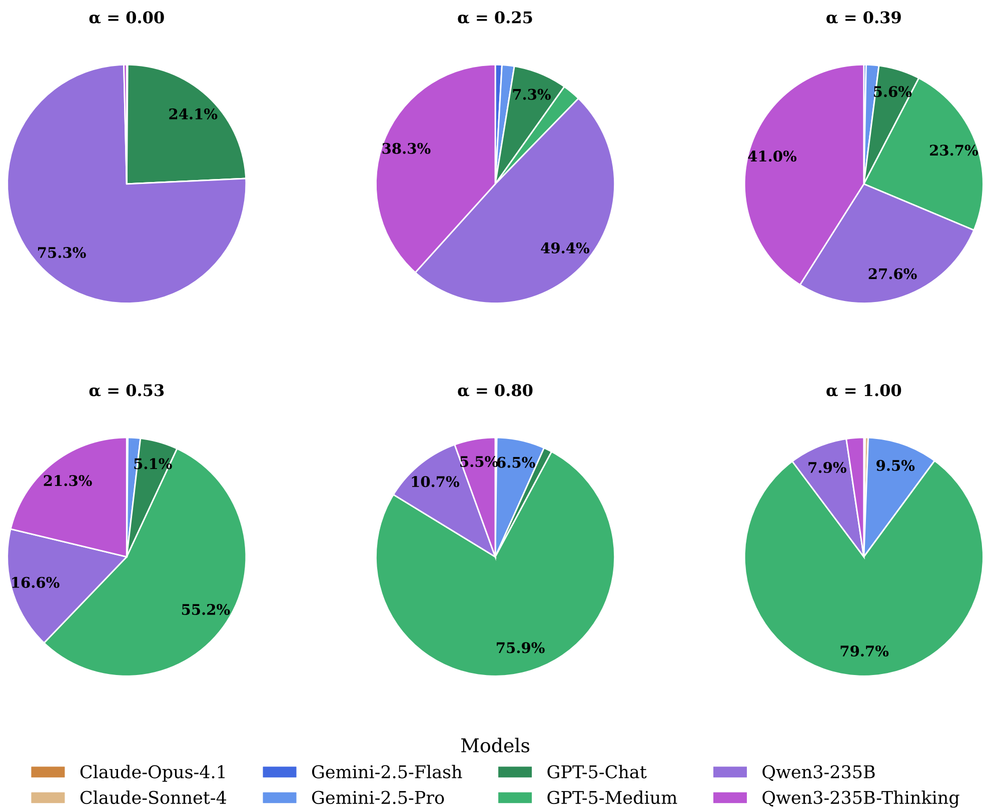

Figure 3: Proportion of model usage, given various trade-off parameters α. When α is low, *Avengers-Pro* tend to route queries to Qwen3 and Qwen3-thinking. With a greater value of α, *Avengers-Pro* favors GPT5-medium and Qwen3-thinking.

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 *Avengers* [15].

References

[1] 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. Healthbench: Evaluating large language models towards improved human health. *arXiv preprint arXiv:2505.08775*, 2025.

[2] Victor Barres, Honghua Dong, Soham Ray, Xujie Si, and Karthik Narasimhan. τ^2-bench: Evaluating conversational agents in a dual-control environment. *arXiv preprint arXiv:2506.07982*, 2025.

[3] Shuhao Chen, Weisen Jiang, Baijiong Lin, James Kwok, and Yu Zhang. Routerdc: Query-based router by dual contrastive learning for assembling large language models. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 66305–66328. Curran Associates, Inc., 2024.

[4] Francois Chollet, Mike Knoop, Gregory Kamradt, and Bryan Landers. Arc prize 2024: Technical report. *arXiv preprint arXiv:2412.04604*, 2024.

[5] Claude. System card addendum: Claude opus 4.1. *www.anthropic.com/news/claude-opus-4-1*, 2025.

[6] Claude. System card: Claude opus 4 & claude sonnet 4. *www.anthropic.com/claude/sonnet*, 2025.

[7] Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.

[8] Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. Livecodebench: Holistic and contamination free evaluation of large language models for code. *arXiv preprint arXiv:2403.07974*, 2024.

[9] OpenAI. Gpt-5 system card. *openai.com/index/gpt-5-system-card*, 2025.

[10] Long Phan, Alice Gatti, Ziwen Han, Nathaniel Li, Josephina Hu, Hugh Zhang, Chen Bo Calvin Zhang, Mohamed Shaaban, John Ling, Sean Shi, et al. Humanity's last exam. *arXiv preprint arXiv:2501.14249*, 2025.

[11] David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. Gpqa: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*, 2024.

[12] Jason Wei, Nguyen Karina, Hyung Won Chung, Yunxin Joy Jiao, Spencer Papay, Amelia Glaese, John Schulman, and William Fedus. Measuring short-form factuality in large language models. *arXiv preprint arXiv:2411.04368*, 2024.

[13] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

[14] 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. Qwen3 embedding: Advancing text embedding and reranking through foundation models, 2025.

[15] Yiqun Zhang, Hao Li, Chenxu Wang, Linyao Chen, Qiaosheng Zhang, Peng Ye, Shi Feng, Daling Wang, Zhen Wang, Xinrun Wang, et al. The avengers: A simple recipe for uniting smaller language models to challenge proprietary giants. *arXiv preprint arXiv:2505.19797*, 2025.

[16] Richard Zhuang, Tianhao Wu, Zhaojin Wen, Andrew Li, Jiantao Jiao, and Kannan Ramchandran. Embedllm: Learning compact representations of large language models, 2024.