A Cost-Benefit Analysis of On-Premise Large Language Model Deployment: Breaking Even with Commercial LLM Services

Guanzhong Pan
Carnegie Mellon University
arthurp@andrew.cmu.edu

Vishal Chodnekar
Unaffiliated
vishal.chodnekar@gmail.com

Abinas Roy
Unaffiliated
Abinasroy@gmail.com

Haibo Wang
Carnegie Mellon University
haibowan@alumni.cmu.edu

Abstract—Large language models (LLMs) are becoming increasingly widespread. Organizations that want to use AI for productivity now face an important decision. They can subscribe to commercial LLM services or deploy models on their own infrastructure. Cloud services from providers such as OpenAI, Anthropic, and Google are attractive because they provide easy access to state-of-the-art models and are easy to scale. However, concerns about data privacy, the difficulty of switching service providers, and long-term operating costs have driven interest in local deployment of open-source models. This paper presents a cost-benefit analysis framework to help organizations determine when on-premise LLM deployment becomes economically viable compared to commercial subscription services. We consider the hardware requirements, operational expenses, and performance benchmarks of the latest open-source models, including Qwen, Llama, Magistral, and etc. Then we compare the total cost of deploying these models locally with the major cloud providers subscription fee. Our findings provide an estimated breakeven point based on usage levels and performance needs. These results give organizations a practical framework for planning their LLM strategies.

Index Terms—Large Language Models, On-Premise Deployment, Cost-Benefit Analysis, Total Cost of Ownership.

I. INTRODUCTION

The rapid development of Large Language Models(LLMs) has driven organizations to apply them in user-facing services for more adaptive user-facing services [1]–[3]. As adoption grows, organizations face a critical strategic choice: whether to rely on commercial cloud-based services subscription or to invest in their own on-premise deployment infrastructure [4], [5]. Providers such as OpenAI, Anthropic, and Google offer easy API access to state-of-the-art and agentic models [6], but costs scale quickly with usage [7]. These solutions also raise concerns over compliance, data protection, and the challenge of transferring service providers [8], [9]. For example, Desai et al. [10] note that privacy issues hinder LLM adoption in finance, where trust and regulation are critical.

At the same time, organizations now have the option to deploy open-source models, including LLaMA [11], Mistral [12], and Qwen [13]. Recent advances in GPU hardware (NVIDIA H100 [14], AMD MI300X [15]) and inference optimization frameworks (vLLM [16], NVIDIA TensorRT-LLM [17], DeepSpeed [18]) also contributed to making local deployment more feasible. As a result, it is worth for enterprises to reconsider whether building their own AI infrastructure may be a more economically viable option.

Despite rising interest in hybrid deployment, few systematic and quantitative comparisons of the two approaches exist. Organizations need a framework to determine when local deployment outweighs commercial services in cost-effectiveness.

This paper provides a comprehensive cost-benefit analysis framework for on-premise open-source LLM deployment. Our main contributions are:

1) A survey of commercial LLM pricing models and open-source alternatives suitable for local deployment.
2) Mathematical models for total cost of ownership (TCO) analysis comparing local open-source LLM deployment and commercial API usage.
3) A playground[1] where enterprise users can apply the cost-benefit framework to explore hardware/API trade-offs.

Our analysis reveals that on-premise deployment are economically viable, with break-even periods typically within a few months for small models, 2 years for medium models and 5 years for larger models, making it viable primarily for organizations with extreme high-volume processing requirements ($\geq$50M tokens/month) or strict data residency mandates.

II. BACKGROUND AND RELATED WORK

A. LLM Deployment and Cost Trade-offs

Modern LLM deployment spans three paradigms: *cloud*, *on-premise*, and *hybrid*. Cloud services provide immediate access to state-of-the-art models but raise issues of recurring cost, data security, and sovereignty [19], [20]. On-premise deployments offer full control and compliance for sensitive domains such as healthcare, finance, and law [21], yet require significant upfront investment. Hybrid solutions balance these trade-offs by running critical workloads locally while offloading scalable or latency-sensitive tasks to the cloud [22].

Prior cost analyses of LLM deployment focus on four directions: (1) reducing API expenditure through pricing strategies and model cascades [7], [23], (2) minimizing inference cost per token via quantization, batching, and speculative decoding [16], [24]–[26], (3) improving system efficiency through serverless or multi-tenant provisioning [27]–[30], and (4)

---

[1]Available at: https://v0-ai-cost-calculator.vercel.app/

modeling TCO considering hardware amortization, precision formats, and energy efficiency [31]–[33].

Despite these advances, few frameworks jointly analyze the economics of cloud and local deployment. Existing studies typically isolate API or infrastructure costs and leave a gap in unified evaluation. This work addresses this gap by integrating pricing, performance, and governance into a single cost-benefit framework for LLM deployment.

B. Open-Source Model Ecosystem

The open-source LLM ecosystem has accelerated, with many models achieving performance as well as commercial models. Meta's LLaMA-3 family (8B–405B parameters) matches or surpasses models like Claude and Gemini in reasoning and knowledge tests [34]. Similarly, Alibaba's open-source Qwen-3 sets new benchmarks in mathematics, coding, and multilingual understanding [35]. The rapid spread of such models allows organizations to run advanced LLMs in-house for greater cost control, feature customization, and data security. As a result, many formerly reliant on commercial APIs are now considering open-source alternatives.

III. SURVEY METHODOLOGY AND MODEL SELECTION

A. Performance Evaluation Framework

We developed a performance evaluation framework to ensure our cost-benefit analysis reflects deployment conditions.

Accuracy Metrics. While accuracy is essential, enterprises require evidence of task-specific performance. We evaluate models across diverse benchmarks encompassing reasoning, mathematics, coding, and multi-domain understanding [61]–[63], [65], [66]. This captures analytical, computational, and applied challenges representative of enterprise workloads.

We then compare open-weight models (e.g., LLaMA, Qwen-3) with commercial models (e.g., ChatGPT, Claude) to quantify performance gaps and assess the feasibility of replacing APIs with open-source alternatives. Results are presented in Table I (adapted from [64]) and summarized in Table IV. This evaluation in realistic tasks ensures organization's model selection aligns with their needs and budget.

B. Model Selection Criteria

Integrating accuracy and operational factors, we define four criteria for model inclusion:

1) **Performance Parity:** Benchmark scores within 20% of top commercial models, reflecting enterprise norms where small accuracy gaps are offset by cost, security, and integration benefits.
2) **Deployment Feasibility:** Hardware requirements suitable for typical enterprise infrastructure.
3) **License Compatibility:** Open-weight models under permissive licenses enabling commercial use.
4) **Community Support:** Active development communities ensuring continuous improvement and stability.

Table I presents our selected models and their benchmark performance compared to commercial alternatives.

IV. COMMERCIAL LLM PRICING MODELS

The API subscription model charges per processed token (input and output), making it suitable for integrating LLMs into custom applications. Costs vary by usage, batching, and model choice. Detailed breakdowns appear in Table I.

V. ON-PREMISE DEPLOYMENT: A COST BREAKDOWN

On-premise deployment means running LLMs entirely using an organization's own data centers or specially designed hardware. This approach does not require any external cloud providers, which provides full control of privacy. We decompose the cost structure into these 3 aspects:

- **Capital Expenditures (CapEx)**: Hardware (GPUs, servers, storage), initial setup, networking. [56]
- **Operational Expenditures (OpEx)**: Electricity, cooling, maintenance, personnel, software licensing. [56]
- **Scaling Costs**: Additional hardware and operational costs as user base or workload grows.

A. Capital Expenditures: The Hardware

The largest upfront cost is compute infrastructure. GPU selection is crucial:

- **Data Center Grade (e.g., NVIDIA A100 [58])**: Great performance for large-scale, multi-user deployments. Enables running the largest models and highest throughput.
- **Prosumer/Workstation Grade (e.g., RTX 5090 [57])**: Suitable for small teams or research, lower cost but limited scalability.

Table II lists the memory, power, and approximate price for NVIDIA 5090-32GB and A100-80GB GPUs. Table III lists the hardware resources needed to set up the service.

VI. COST MODEL AND BREAK-EVEN ANALYSIS

We developed a quantitative model to assess the cost of matching the throughput of leading commercial LLMs. Our model uses available benchmark, hardware, and pricing data.

A. Cost-Performance Tradeoffs of Open-Source LLM Deployment

Tables I and IV show that open-weight models can deliver competitive performance. Although large open models such as Kimi-K2, GLM-4.5, and Qwen3-235B require GPU clusters costing over $200k, their accuracy on enterprise benchmarks places them close to leading closed models. This suggests that, technically, open alternatives can approach commercial state-of-the-art, albeit with higher operational complexity.

Notably, the gap between "large" and "medium" deployments is smaller than expected. Medium-scale models like *gpt-oss-120B*, *GLM-4.5-Air*, and *Llama-3.3-70B* run efficiently on two A100-80GB GPUs ($30k), with less than 10% accuracy loss. They thus offer substantially lower ownership costs while maintaining strong performance across reasoning, coding, and domain-specific tasks.

Small models such as *EXAONE 4.0 32B*, *Qwen3-30B*, and *Magistral Small* show that sub-30B deployments are feasible

Model	Total Params	GPQA	MATH-500	LiveCodeBench	MMLU-Pro	API Cost (In/Out, USD / 1M tokens)
Large Open Models						
Kimi-K2 [36], [37]	1T	76.6%	97.1%	55.6%	82.4%	–
GLM-4.5 [38], [39]	355B	78.2%	97.9%	73.8%	83.5%	–
Qwen3-235B [40], [41]	235B	79.0%	98.4%	78.8%	84.3%	–
Medium Open Models						
gpt-oss-120B [42], [43]	120B	78.2%	–	63.9%	80.8%	–
GLM-4.5-Air [38], [44]	106B	73.3%	96.5%	68.4%	81.5%	–
Llama-3.3-70B [45], [46]	70B	49.8%	77.3%	28.8%	71.3%	–
Small Open Models						
EXAONE 4.0 32B [47], [48]	32B	73.9%	97.7%	74.7%	81.8%	–
Qwen3-30B [40], [49]	30B	70.7%	97.6%	70.7%	80.5%	–
Magistral Small [50], [51]	24B	64.1%	96.3%	51.4%	74.6%	–
Commercial Reference						
GPT-5 (by OpenAI [52])	–	85.4%	**99.4%**	66.8%	**87.1%**	$1.25 / $10.00
Claude-4 Opus (by Anthropic [53])	–	70.1%	94.1%	54.2%	86.0%	$15.00 / $75.00
Claude-4 Sonnet (by Anthropic [53])	–	68.3%	93.4%	44.9%	83.7%	$3.00 / $15.00
Grok-4 (by xAI [54])	–	**87.7%**	99.0%	**81.9%**	86.6%	$3.00 / $15.00
Gemini 2.5 Pro (by Google [55])	–	84.4%	96.7%	80.1%	86.2%	$1.25 / $10.00

TABLE II
COMPARISON OF A100-80GB AND RTX 5090-32GB GPUS

GPU	Memory	Power	Price (USD)
NVIDIA RTX 5090-32GB [57]	32 GB	575 W	$2,000
NVIDIA A100-80GB [58]	80 GB	400 W	$15,000

on a single consumer-grade RTX 5090 ($2k). With performance comparable to medium models, they suit small and mid-sized enterprises prioritizing cost efficiency and local control. Although they trail the highest benchmarks, the practical performance gap between 30B- and 70B-class models remains modest, indicating that smaller deployments can meet a wide range of enterprise needs.

While commercial APIs still hold a slight edge in peak accuracy and efficiency, open-weight models now provide viable, cost-effective alternatives. The performance gaps across large, medium, and small open deployments are far narrower than their order-of-magnitude hardware cost differences. For many organizations, deploying a medium or even small open model locally offers a sustainable break-even option balancing capability, cost, and autonomy from external providers.

B. Cost Model

The one-time infrastructure cost is defined as:

$$C_{\text{hardware}} = N_{\text{GPU}} \cdot C_{\text{GPU}} \tag{1}$$

Assuming business operation of 8 hours/day and 20 days/month, the monthly electricity cost is:

$$C_{\text{electricity}} = N_{\text{GPU}} \cdot P_{\text{GPU}} \cdot H_{\text{operation}} \cdot R_{\text{electricity}} \tag{2}$$

The total local deployment cost becomes:

$$C_{\text{local}}(t) = C_{\text{hardware}} + C_{\text{electricity}} \cdot t \tag{3}$$

where t is the number of months after deployment, C_{GPU} and N_{GPU} denote per-unit GPU cost and quantity, P_{GPU} power consumption, $R_{\text{electricity}}$ the electricity rate, and $H_{\text{operation}}$ the monthly operating hours.

C. Commercial API Throughput Analysis

For a fair comparison, API costs are normalized by the token generation capacity Q_{capacity} achievable on local hardware:

$$Q_{\text{capacity}} = T_{\text{throughput}} \cdot H_{\text{operation}} \cdot 3600 \tag{4}$$

The equivalent API cost for producing the same token volume per month is:

$$C_{\text{API}}(Q_{\text{capacity}}) = \frac{Q_{\text{capacity}}}{3} \cdot \frac{C_{\text{input}}}{1M} + \frac{2Q_{\text{capacity}}}{3} \cdot \frac{C_{\text{output}}}{1M} \tag{5}$$

and scales linearly with time:

$$C_{\text{API}}(t) = C_{\text{API}}(Q_{\text{capacity}}) \cdot t \tag{6}$$

Here C_{input} and C_{output} are API prices per million input/output tokens, using a 2:1 ratio typical of real workloads.

D. Break-Even Analysis

Local and API costs are modeled as time-dependent functions:

$$C_{\text{local}}(t) = C_{\text{API}}(t) \tag{7}$$

Solving yields the break-even time t^*, where cumulative local and API costs are equal. For $t > t^*$, local deployment becomes more economical, particularly under sustained, high-throughput usage. The next section reports t^* values across open-source and commercial models.

TABLE III
HARDWARE AND CAPACITY SUMMARY FOR DEPLOYABLE OPEN-SOURCE MODELS (FP8/W8A16 ON A100 [58]/RTX 5090 [57], ELECTRICITY AT $0.15/KWH)

Model	MoE	VRAM (FP8)	Hardware Deployment	Throughput (tok/sec)	Hardware Cost	Token Capacity/Month
Large Open Models						
Kimi-K2 [37]	Yes	1000 GB	16× A100-80GB	800	$240k	506.9M
GLM-4.5 [39]	Yes	355 GB	6× A100-80GB	400	$90k	253.4M
Qwen3-235B [41]	Yes	235 GB	4× A100-80GB	400	$60k	253.4M
Medium Open Models						
gpt-oss-120B [43]	Yes	120 GB	2× A100-80GB	220	$30k	139.4M
GLM-4.5-Air [44]	Yes	106 GB	2× A100-80GB	200	$30k	126.7M
Llama-3.3-70B [46]	No	70 GB	1× A100-80GB	190	$15k	120.4M
Small Open Models						
EXAONE 4.0 32B [48]	No	32 GB	1× RTX 5090	200	$2k	126.7M
Qwen3-30B [49]	No	30 GB	1× RTX 5090	180	$2k	114.0M
Magistral Small [51]	No	24 GB	1× RTX 5090	150	$2k	95.0M

TABLE IV
BREAK-EVEN ANALYSIS SUMMARY FOR ALL MODEL-API COMBINATIONS (MONTHS) WITH AMORTIZED PERFORMANCE DIFFERENCES

Open Model	GPT-5	Claude-4 Opus	Claude-4 Sonnet	Grok-4	Gemini 2.5 Pro	Range
Large Open Models						
Kimi-K2	69.3 (-6.75%)	8.7 (**+1.83%**)	44.0 (+5.35%)	44.0 (-10.88%)	63.1 (-8.93%)	8.7-69.3
GLM-4.5	51.5 (-1.32%)	6.5 (+7.25%)	32.8 (+10.78%)	32.8 (-5.45%)	47.0 (-3.50%)	6.5-51.5
Qwen3-235B	34.0 (**+0.45%**)	4.3 (+9.03%)	21.8 (+12.55%)	21.8 (**-3.68%**)	31.1 (**-1.73%**)	4.3-34.0
Medium Open Models						
gpt-oss-120B	30.9 (-5.47%)	3.9 (+4.20%)	19.8 (+8.67%)	19.8 (-11.10%)	28.2 (-9.27%)	3.9-30.9
GLM-4.5-Air	34.0 (-4.75%)	4.3 (+3.83%)	21.8 (+7.35%)	21.8 (-8.88%)	31.1 (-6.93%)	4.3-34.0
Llama-3.3-70B	17.8 (-27.88%)	2.3 (-19.30%)	11.4 (-15.78%)	11.4 (-32.00%)	16.2 (-30.05%)	2.3-17.8
Small Open Models						
EXAONE 4.0 32B	2.26 (-2.65%)	0.3 (+5.93%)	1.4 (+9.45%)	1.4 (-6.43%)	2.06 (-4.48%)	0.3-2.26
Qwen3-30B	2.5 (-5.25%)	0.3 (+3.38%)	1.6 (+6.90%)	1.6 (-9.00%)	2.3 (-7.05%)	0.3-2.5
Magistral Small	3.0 (-12.25%)	0.4 (-3.23%)	1.9 (**+0.28%**)	1.9 (-15.73%)	2.76 (-13.78%)	0.4-3.0

E. Comprehensive Break-Even Analysis Results

By substituting the values from Table I and Table II into Equation (7), we obtain the results summarized in Table IV, with cost details in Table III. Our analysis across 54 deployment scenarios—covering nine open-source models and six commercial APIs—shows substantial variation in economic viability across model sizes.

Small Models. Sub-30B models prove highly cost-effective, often reaching break-even within three months. Their low hardware cost enables smaller organizations to sustain competitive AI capabilities without ongoing subscription fees.

Medium Models. Medium-scale deployments require moderate investment but achieve favorable returns over time. They offer a practical balance between performance and cost and are suitable for enterprises with steady, high-volume workloads.

Large Models. Large-scale setups face the steepest economic barriers. Despite competitive accuracy, their high hardware and power demands extend break-even horizons. This challenge makes them viable only for organizations with sustained, large-scale inference needs.

F. Deployment Decision Framework

We analyzed break-even points by both model size and enterprise type. Small, medium, and large organizations have different needs for computing power, regulatory compliance, and financial resources. These factors shape whether on-premise deployment is practical. Table IV reports the quantitative results. The following sections explain what these results mean for organizations of different sizes.

1) Small Enterprises (SMEs): For SMEs with limited budgets and moderate workloads (<10M tokens/month), small open-source models such as EXAONE 4.0 32B and Qwen3-30B offer the most viable entry point. Our results show that break-even can occur in as little as 0.3–3 months depending on the commercial baseline (e.g., Claude-4 Opus vs. GPT-5). This aligns with recent findings that SMEs prioritize cost savings and data control over absolute performance [21], [23]. The feasibility of deploying on consumer-grade GPUs (e.g., RTX 5090 at ∼$2,000) further reduces capital barriers. Typical use cases include customer support automation, internal knowledge search, and lightweight document analysis—tasks where 30B-class models provide sufficient accuracy [35], [47].

2) Medium Enterprises: Medium-scale enterprises (processing 10–50M tokens/month) represent the sweet spot for on-premise adoption. Medium models such as GLM-4.5-Air and Llama-3.3-70B demonstrate balanced economics, with break-even periods ranging from 3.8 to 34 months depending on provider comparison. Hardware requirements remain

manageable ($15k–$30k for dual A100 setups), and throughput levels (120–220 tokens/sec) are sufficient for concurrent business workloads such as code assistance, analytics, and customer-facing applications. This tier benefits most from hybrid strategies, where sensitive workloads run locally and burst traffic leverages cloud APIs [8], [22]. Regulatory-driven industries such as healthcare and finance particularly favor medium-scale deployments for balancing compliance and cost [10].

3) Large Enterprises: For large enterprises with extreme-scale workloads (>50M tokens/month), large open-source models (e.g., Qwen3-235B, Kimi-K2) become economically attractive, albeit with longer break-even horizons (3.5–69.3 months). While upfront investments exceed $40k–$190k, these organizations often already operate GPU clusters for other workloads, reducing incremental CapEx [31], [56]. Use cases include enterprise-wide generative applications, advanced research, and domain-specific copilots requiring reasoning depth comparable to commercial APIs. However, when compared against aggressively priced providers such as Gemini 2.5 Pro, break-even can extend to 5–9 years, challenging the economic case unless privacy, sovereignty, or vendor lock-in are overriding concerns [19], [33]. Thus, for this tier, non-financial factors (e.g., strategic autonomy, compliance) often weigh more heavily than pure cost.

VII. CONCLUSION

This study evaluates 54 deployment scenarios to clarify the economic trade-offs of on-premise versus API-based LLM use. Results show that deployment economics are highly context-dependent and challenge common assumptions about local feasibility.

- **Small-scale:** Break even within 3 months, making local deployment accessible for smaller organizations.
- **Medium-scale:** Offer balanced performance and cost, with recovery typically within 6–24 months.
- **Large-scale:** Require longer horizons (often beyond 2 years), feasible only for sustained, high-volume workloads.
- **Pricing variability:** Differences among providers create large swings in cost-effectiveness, with premium tiers enabling faster local payback.

Strategic Implications. Deployment choices can be grouped into short-term (0–6 months), mid-term (6–24 months), and long-term (≥24 months) investments, which offers organizations a practical lens to align cost recovery with strategic goals. The rapid evolution of models, hardware, and pricing means deployment is a continuous optimization problem rather than a one-time decision.

Future Work. Further research should empirically validate these break-even estimates and extend TCO models to include staffing and maintenance, and explore hybrid paradigms that combine economic efficiency with reliability. Continued benchmarking will be essential to track performance and cost convergence between open and commercial ecosystems.

Overall, this study situates break-even analysis within the broader landscape of evolving LLM technology and highlights both the growing accessibility of local deployment and the ongoing economic tension with cloud-based services.

REFERENCES

[1] F. Shareef, "Enhancing Conversational AI with LLMs for Customer Support Automation," in *Proc. 2024 2nd Int. Conf. Self Sustainable Artif. Intell. Syst. (ICSSAS)*, Oct. 2024, pp. 239–244, doi: 10.1109/icssas64001.2024.10760403.

[2] S. Minaee, T. Mikolov, N. Nikzad, M. Chenaghlu, R. Socher, X. Amatriain, and J. Gao, "Large Language Models: A Survey," *arXiv*, Feb. 2024, doi: 10.48550/arxiv.2402.06196.

[3] W. X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong, Y. Du, C. Yang, Y. Chen, Z. Chen, J. Jiang, R. Ren, Y. Li, X. Tang, Z. Liu, P. Liu, J.-Y. Nie, and J.-R. Wen, "A Survey of Large Language Models," *arXiv:2303.18223 [cs]*, Mar. 2023. [Online]. Available: https://arxiv.org/abs/2303.18223

[4] Z. Zhang, J. Shi, and S. Tang, "Cloud or On-Premise? A Strategic View of Large Language Model Deployment," SSRN, Jun. 16, 2025. [Online]. Available: https://ssrn.com/abstract=5296479. doi: 10.2139/ssrn.5296479

[5] H. Huang *et al.*, "Position: On-Premises LLM Deployment Demands a Middle Path: Preserving Privacy Without Sacrificing Model Confidentiality," *arXiv*, 2024. [Online]. Available: https://arxiv.org/abs/2410.11182. [Accessed: Aug. 27, 2025].

[6] Mohammad Luqman, Himaanshu Gauba, Prabhat Kumar, Akshar Prabhu Desai, Ajay Yadav, Ritu Prajapati, and Pranjul Yadav. Agentic AI in Finance: A Comprehensive Overview. 2025.

[7] L. Chen, M. Zaharia, and J. Zou, "FrugalGPT: How to Use Large Language Models While Reducing Cost and Improving Performance," *arXiv*, 2023. [Online]. Available: https://arxiv.org/abs/2305.05176. [Accessed: Aug. 27, 2025].

[8] K. Chen, X. Zhou, Y. Lin, S. Feng, L. Shen, and P. Wu, "A Survey on Privacy Risks and Protection in Large Language Models," *arXiv*, 2025. [Online]. Available: https://arxiv.org/abs/2505.01976. [Accessed: Aug. 27, 2025].

[9] F. Dennstädt, J. Hastings, P. M. Putora, M. Schmerder, and N. Cihoric, "Implementing large language models in healthcare while balancing control, collaboration, costs and security," *npj Digital Medicine*, vol. 8, no. 1, Mar. 2025, doi: 10.1038/s41746-025-01476-7.

[10] A. P. Desai, T. Ravi, M. Luqman, G. Mallya, N. Kota, and P. Yadav, "Opportunities and challenges of generative-AI in finance," in *Proc. 2024 IEEE Int. Conf. Big Data (BigData)*, 2024, pp. 4913–4920.

[11] H. Touvron et al., "Llama 2: Open Foundation and Fine-Tuned Chat Models," *arXiv preprint arXiv:2307.09288*, 2023.

[12] A. Q. Jiang et al., "Mistral 7B," *arXiv preprint arXiv:2310.06825*, 2023.

[13] J. Bai et al., "Qwen Technical Report," *arXiv preprint arXiv:2309.16609*, 2023.

[14] NVIDIA, "NVIDIA H100 Tensor Core GPU Architecture Overview," NVIDIA, [Online]. Available: https://resources.nvidia.com/en-us-data-center-overview-mc/en-us-data-center-overview/gtc22-whitepaper-hopper. [Accessed: Aug. 27, 2025].

[15] AMD, "AMD Instinct MI300X Accelerators," AMD, [Online]. Available: https://www.amd.com/en/products/accelerators/instinct/mi300. [Accessed: Aug. 27, 2025].

[16] W. Kwon *et al.*, "Efficient Memory Management for Large Language Model Serving with PagedAttention," *arXiv*, 2023. [Online]. Available: https://arxiv.org/abs/2309.06180. [Accessed: Nov. 24, 2024].

[17] NVIDIA, "NVIDIA TensorRT-LLM," *NVIDIA Documentation*, 2024. [Online]. Available: https://docs.nvidia.com/tensorrt-llm/index.html. [Accessed: Aug. 27, 2025].

[18] R. Y. Aminabadi *et al.*, "DeepSpeed Inference: Enabling Efficient Inference of Transformer Models at Unprecedented Scale," *arXiv*, 2022. [Online]. Available: https://arxiv.org/abs/2207.00032. [Accessed: Aug. 27, 2025].

[19] S. Park, S. Jeon, C. Lee, S. Jeon, B.-S. Kim, and J. Lee, "A Survey on Inference Engines for Large Language Models: Perspectives on Optimization and Efficiency," *arXiv.org*, 2025. [Online]. Available: https://arxiv.org/abs/2505.01658. [Accessed: Aug. 08, 2025].

[20] W. X. Zhao *et al.*, "A Survey of Large Language Models," *arXiv (Cornell University)*, Mar. 2023. doi: 10.48550/arxiv.2303.18223.

[21] H. Wu and Y. Cao, "Membership Inference Attacks on Large-Scale Models: A Survey," *arXiv.org*, 2025. [Online]. Available: https://arxiv.org/abs/2503.19338. [Accessed: Mar. 26, 2025].

[22] Z. Hao, H. Jiang, S. Jiang, J. Ren, and T. Cao, "Hybrid SLM and LLM for Edge-Cloud Collaborative Inference," *EdgeFM 24*, Jun. 2024. doi: 10.1145/3662006.3662067.

[23] M. Yan, S. Agarwal, and S. Venkataraman, "Decoding Speculative Decoding," *arXiv.org*, 2024. [Online]. Available: https://arxiv.org/abs/2402.01528. [Accessed: Aug. 28, 2025].

[24] R. Aminabadi *et al.*, "DeepSpeed Inference: Enabling Efficient Inference of Transformer Models at Unprecedented Scale," *arXiv.org*, 2022. [Online]. Available: https://arxiv.org/pdf/2207.00032. [Accessed: May 03, 2024].

[25] Y. Sheng *et al.*, "FlexGen: High-Throughput Generative Inference of Large Language Models with a Single GPU," *arXiv.org*, 2023. [Online]. Available: https://arxiv.org/abs/2303.06865. [Accessed: Aug. 28, 2025].

[26] Y. Leviathan, M. Kalman, and Y. Matias, "Fast Inference from Transformers via Speculative Decoding," *arXiv.org*, 2022. [Online]. Available: https://arxiv.org/abs/2211.17192. [Accessed: Aug. 28, 2025].

[27] Y. Fu *et al.*, "ServerlessLLM: Low-Latency Serverless Inference for Large Language Models," *arXiv.org*, 2024. [Online]. Available: https://arxiv.org/abs/2401.14351.

[28] Microsoft, "Provisioned Throughput Units — Azure OpenAI," 2025. Available: https://learn.microsoft.com/azure/ai-foundry/openai/how-to/provisioned-throughput-onboarding.

[29] Y. Sheng *et al.*, "S-LoRA: Serving Thousands of Concurrent LoRA Adapters," *MLSys*, 2024. Available: https://proceedings.mlsys.org/paper_files/paper/2024/file/906419cd502575b617cc489a1a696a67-Paper-Conference.pdf.

[30] J. Zhao *et al.*, "Accelerating Serverless LLM Inference with Materialization (Medusa)," *SoCC*, 2025. doi: 10.1145/3669940.3707285.

[31] E. Erdil, "Inference Economics of Language Models," *arXiv.org*, 2025. [Online]. Available: https://arxiv.org/abs/2506.04645. [Accessed: Aug. 28, 2025].

[32] J. Kim *et al.*, "An Inquiry into Datacenter TCO for LLM Inference with FP8," *arXiv.org*, 2025. [Online]. Available: https://arxiv.org/abs/2502.01070. [Accessed: Aug. 28, 2025].

[33] J. Fernandez, C. Na, V. Tiwari, Y. Bisk, S. Luccioni, and E. Strubell, "Energy Considerations of Large Language Model Inference and Efficiency Optimizations," *arXiv.org*, 2025. [Online]. Available: https://arxiv.org/abs/2504.17674. [Accessed: May 06, 2025].

[34] H. Touvron *et al.*, "LLaMA: Open and Efficient Foundation Language Models," *arXiv*, Feb. 2023. doi: 10.48550/arxiv.2302.13971.

[35] J. Bai *et al.*, "Qwen Technical Report," *arXiv.org*, Sep. 28, 2023. [Online]. Available: https://arxiv.org/abs/2309.16609.

[36] K. Team *et al.*, "Kimi K2: Open Agentic Intelligence," *arXiv.org*, 2025. [Online]. Available: https://arxiv.org/abs/2507.20534. [Accessed: Aug. 08, 2025].

[37] MoonShot.AI, "moonshotai/Kimi-K2-Instruct · Hugging Face," *Huggingface.co*, Jul. 12, 2025. [Online]. Available: https://huggingface.co/moonshotai/Kimi-K2-Instruct. [Accessed: Aug. 28, 2025].

[38] 5 Team *et al.*, "GLM-4.5: Agentic, Reasoning, and Coding (ARC) Foundation Models," *arXiv.org*, 2025. [Online]. Available: https://arxiv.org/abs/2508.06471. [Accessed: Aug. 28, 2025].

[39] ZAI, "zai-org/GLM-4.5 · Hugging Face," *Huggingface.co*, Aug. 11, 2025. [Online]. Available: https://huggingface.co/zai-org/GLM-4.5. [Accessed: Aug. 28, 2025].

[40] A. Yang *et al.*, "Qwen3 Technical Report," *arXiv.org*, 2025. [Online]. Available: https://arxiv.org/abs/2505.09388.

[41] QWen, "Qwen/Qwen3-235B-A22B · Hugging Face," *Huggingface.co*, Jun. 15, 2025. [Online]. Available: https://huggingface.co/Qwen/Qwen3-235B-A22B. [Accessed: Aug. 28, 2025].

[42] OpenAI *et al.*, "gpt-oss-120b & gpt-oss-20b Model Card," *arXiv.org*, 2025. [Online]. Available: https://arxiv.org/abs/2508.10925. [Accessed: Aug. 28, 2025].

[43] OpenAI, "openai/gpt-oss-120b · Hugging Face," *Huggingface.co*, Aug. 07, 2025. [Online]. Available: https://huggingface.co/openai/gpt-oss-120b. [Accessed: Aug. 28, 2025].

[44] ZAI, "Blocked," *Huggingface.co*, 2025. [Online]. Available: https://huggingface.co/zai-org/GLM-4.5-Air. [Accessed: Aug. 28, 2025].

[45] A. Dubey *et al.*, "The Llama 3 Herd of Models," *arXiv.org*, 2024. [Online]. Available: https://arxiv.org/abs/2407.21783.

[46] Meta, "meta-llama/Llama-3.3-70B-Instruct · Hugging Face," *Huggingface.co*, Dec. 06, 2024. [Online]. Available: https://huggingface.co/meta-llama/Llama-3.3-70B-Instruct.

[47] R. L. AI *et al.*, "EXAONE 4.0: Unified Large Language Models Integrating Non-reasoning and Reasoning Modes," *arXiv.org*, 2025. [Online]. Available: https://arxiv.org/abs/2507.11407. [Accessed: Aug. 28, 2025].

[48] LGAI, "LGAI-EXAONE/EXAONE-4.0-32B · Hugging Face," *Huggingface.co*, Jul. 29, 2025. [Online]. Available: https://huggingface.co/LGAI-EXAONE/EXAONE-4.0-32B. [Accessed: Aug. 28, 2025].

[49] QWen, "Qwen/Qwen3-30B-A3B · Hugging Face," *Huggingface.co*, Aug. 06, 2025. [Online]. Available: https://huggingface.co/Qwen/Qwen3-30B-A3B. [Accessed: Aug. 28, 2025].

[50] Mistral-AI *et al.*, "Magistral," *arXiv.org*, 2025. [Online]. Available: https://arxiv.org/abs/2506.10910. [Accessed: Aug. 28, 2025].

[51] MistralAI, "mistralai/Magistral-Small-2506 · Hugging Face," *Huggingface.co*, 2025. [Online]. Available: https://huggingface.co/mistralai/Magistral-Small-2506. [Accessed: Aug. 28, 2025].

[52] OpenAI, "API Pricing," https://openai.com/api/pricing/, 2024.

[53] Anthropic, "Claude Pricing," https://docs.anthropic.com/en/docs/about-claude/pricing, 2024.

[54] xAI, "Models and Pricing — xAI Docs," Docs.x.ai, 2024. [Online]. Available: https://docs.x.ai/docs/models

[55] Google, "Gemini Developer API Pricing," Google AI for Developers, 2025. [Online]. Available: https://ai.google.dev/gemini-api/docs/pricing

[56] J. Kim *et al.*, "An Inquiry into Datacenter TCO for LLM Inference with FP8," *arXiv.org*, 2025. [Online]. Available: https://arxiv.org/abs/2502.01070.

[57] NVIDIA, "NVIDIA GeForce RTX 5090 Graphics Cards," *NVIDIA*, 2025. [Online]. Available: https://www.nvidia.com/en-us/geforce/graphics-cards/50-series/rtx-5090/.

[58] NVIDIA, "NVIDIA A100 Tensor Core GPU Architecture: Unprecedented Acceleration at Every Scale," *NVIDIA White Paper*, 2020. [Online]. Available: https://images.nvidia.com/aem-dam/en-zz/Solutions/data-center/nvidia-ampere-architecture-whitepaper.pdf.

[59] LiveBench Team, "LiveBench: A Challenging, Contamination-Free LLM Benchmark," https://livebench.ai/, 2024.

[60] A. P. Desai, R. Prajapati, T. Ravi, M. Luqman, and P. Yadav, "Emerging Trends in LLM Benchmarking," in *2024 IEEE International Conference on Big Data (BigData)*, pp. 8805–8807, IEEE, 2024.

[61] D. Rein *et al.*, "GPQA: A Graduate-Level Google-Proof Q&A Benchmark," *arXiv*, 2023. [Online]. Available: https://arxiv.org/abs/2311.12022.

[62] D. Hendrycks *et al.*, "Measuring Mathematical Problem Solving With the MATH Dataset," *arXiv:2103.03874 [cs]*, Nov. 2021. [Online]. Available: https://arxiv.org/abs/2103.03874.

[63] N. Jain *et al.*, "LiveCodeBench: Holistic and Contamination Free Evaluation of Large Language Models for Code," *arXiv*, 2024. [Online]. Available: https://arxiv.org/abs/2403.07974.

[64] A. Analysis, "Model & API Providers Analysis — Artificial Analysis," *artificialanalysis.ai*, 2025. [Online]. Available: https://artificialanalysis.ai/.

[65] Y. Wang *et al.*, "MMLU-Pro: A More Robust and Challenging Multi-Task Language Understanding Benchmark," *arXiv*, 2024. [Online]. Available: https://arxiv.org/abs/2406.01574.

[66] Akshar Prabhu Desai, Ritu Prajapati, Tejasvi Ravi, Mohammad Luqman, and Pranjul Yadav. Emerging Trends in LLM Benchmarking. In *2024 IEEE International Conference on Big Data (BigData)*, pages 8805–8807. IEEE, 2024.

[67] Artificial Analysis, "AI Model & API Providers Analysis," https://artificialanalysis.ai/, accessed on Sept. 27, 2025.