arXiv:2508.12461v3 [cs.CL] 13 Dec 2025

Is GPT-OSS Good? A Comprehensive Evaluation of OpenAI's Latest Open Source Models

Ziqian Bi[1,2*], Keyu Chen[1,3*], Chiung-Yi Tseng[1,4*], Danyang Zhang[1,5*], Tianyang Wang[1], Hongying Luo[1], Lu Chen[1], Junming Huang[1], Jibin Guan[6], Junfeng Hao[6], Xinyuan Song[7], Junhao Song[8†]

[1]AI Agent Lab, Vokram Group, United Kingdom, ai-agent-lab@vokram.com
[2]Purdue University, United States, bi32@purdue.edu
[3]Georgia Institute of Technology, United States, kchen637@gatech.edu
[4]LuxMuse AI, United States, ctseng@luxmuse.ai
[5]ByteDance Inc, United States, joseph.zhang@bytedance.com
[6]University of Minnesota, United States, jguan@umn.edu, ygzhjf85@gmail.com
[7]Emory University, United States, xinyuan.song@emory.edu
[8]Imperial College London, United Kingdom, junhao.song23@imperial.ac.uk

***Abstract*—In August 2025, OpenAI released GPT-OSS models, its first open weight large language models since GPT-2 in 2019, comprising two mixture of experts architectures with 120B and 20B parameters. We evaluated both variants against six contemporary open source large language models ranging from 14.7B to 235B parameters, representing both dense and sparse designs, across ten benchmarks covering general knowledge, mathematical reasoning, code generation, multilingual understanding, and conversational ability. All models were tested in unquantised form under standardised inference settings, with statistical validation using McNemar's test and effect size analysis. Results show that gpt-oss-20B consistently outperforms gpt-oss-120B on several benchmarks, such as HumanEval and MMLU, despite requiring substantially less memory and energy per response. Both models demonstrate mid-tier overall performance within the current open source landscape, with relative strength in code generation and notable weaknesses in multilingual tasks. These findings provide empirical evidence that scaling in sparse architectures may not yield proportional performance gains, underscoring the need for further investigation into optimisation strategies and informing more efficient model selection for future open source deployments. More details and evaluation scripts are available at the Project Webpage.**

***Index Terms*—Large language models, gpt-oss, model evaluation, benchmarking, reasoning models, mixture of experts, performance analysis**

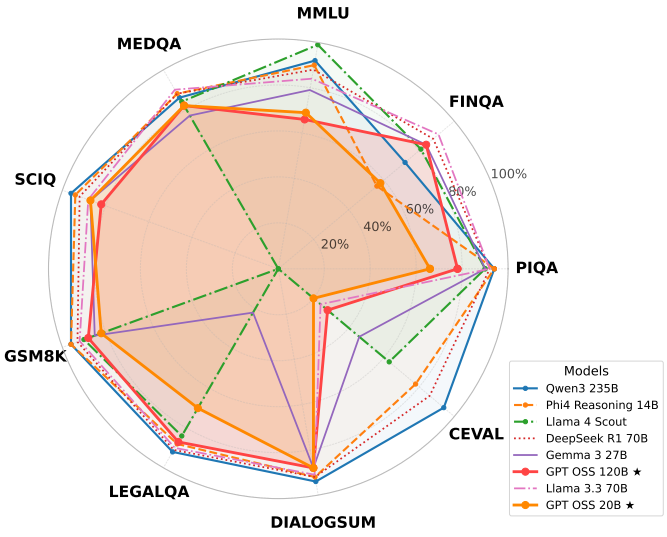

Fig. 1: **Multi-dimensional performance comparison across eight evaluated models.** GPT-OSS models (highlighted) show middle-tier performance with particular strength in code generation but weakness in multilingual tasks.

I. Introduction

The past five years have seen a rapid expansion of open source large language models (LLMs), with dense architectures such as Llama and Gemma joined by increasingly capable mixture of experts (MoE) designs. MoE architectures activate only a fraction of parameters per token [1], [2], offering a theoretical path to improved compute efficiency and scalability [3]–[5]. In August 2025, OpenAI released GPT OSS models, its first open weight models since GPT-2 in 2019 [6], introducing MoE variants with 20B and 120B total parameters into a landscape already shaped by competitive offerings from Meta [7], Google [8], DeepSeek [9], Alibaba [10], [11] and Microsoft [12].

While scaling laws have been widely used to guide LLM development, empirical studies on their validity for sparse architectures remain limited. Existing evaluations have often focused on single domains or isolated benchmarks [13], [14], leaving open the question of whether larger MoE variants consistently outperform smaller ones across diverse capabilities. This gap matters both for understanding the training dynamics of sparsely activated networks and for making informed deployment choices under resource constraints.

To address this gap, we position GPT-OSS alongside six

∗: Equal contribution.
†: Corresponding author: Junhao Song (junhao.song23@imperial.ac.uk)

contemporary open source LLMs spanning 14.7B to 235B parameters, covering both dense and sparse architectures. We design a controlled evaluation across ten established benchmarks grouped into five capability domains: general understanding, mathematical reasoning, code generation, multilingual comprehension, and conversational ability. This framework enables a direct comparison of performance and efficiency characteristics, isolating the relationship between parameter scale, architecture type and task outcomes.

Our analysis serves two purposes. From a research perspective, it tests whether observed performance trends align with established scaling expectations in the MoE setting. From a practical perspective, it examines the cost performance profile of GPT-OSS relative to strong open source baselines, incorporating memory, latency, and energy measurements that are often omitted from academic reports. As illustrated in **Fig 1** and elaborated in subsequent sections, these results provide evidence relevant to both theoretical modelling of scaling behaviour and the design of efficient deployment strategies in the evolving open source community.

II. Related Work

A. Evaluation of Large Language Models

The evaluation of large language models (LLMs) has broadened as models have grown in scale and capability. Early work by Radford et al. [6] laid the foundation for systematic assessment with initial benchmarks such as WebText [6]. Hendrycks et al. [15] introduced MMLU, covering 57 subjects, which became a standard for measuring broad knowledge. For mathematical reasoning, Cobbe et al. [16] proposed GSM8K to evaluate multi-step arithmetic problems, and Lewkowycz et al. [17] showed with Minerva [17] that domain-specific training can dramatically improve mathematical performance and raising questions about cross domain generalisation.

Code generation has emerged as another central evaluation domain. Chen et al. [18] introduced HumanEval to test functional correctness in programming tasks. Austin et al. [19] and Li et al. [20] extended this with MBPP and multilingual coding benchmarks. Studies have demonstrated that pretraining on large code corpora yields substantial gains [21], and Fried et al. [22] showed that instruction-tuned smaller models can rival much larger systems, challenging assumptions about scale requirements in coding tasks.

Beyond English, evaluation efforts have expanded to multilingual contexts. Huang et al. [23] introduced C-Eval, a large-scale benchmark for Chinese, while XTREME and MMLU variants extended coverage to other languages. These studies revealed the so-called "curse of multilinguality" [24], where expanding language coverage often reduces monolingual performance. This trade-off is directly relevant to GPT-OSS, which aims to support many languages without language-specific optimisation.

Conversational ability is now assessed with benchmarks designed for multi-turn dialogue. Zheng et al. [25] proposed MT-Bench to measure coherence and helpfulness, while Thoppilan et al. [26] stressed the importance of factual grounding and safety in open domain dialogue. Roller et al. [27] highlighted the difficulty of balancing engagement, consistency and informativeness. These considerations frame our assessment of GPT-OSS, where conversational quality is weighed against computational efficiency.

B. Model Architectures and Scaling

Architecture choices strongly influence model performance and efficiency. Mixture of Experts (MoE) architectures, which activate only a subset of parameters per token, have become a focus due to their efficiency advantages. The Switch Transformer [2] and GLaM [28] showed that sparse models can rival dense models with significantly reduced compute. Work on routing and expert allocation [29] revealed complex trade-offs, and methods such as Branch-Train-Merge [30] and Expert-Choice routing [31] improved training stability and performance. GPT-OSS adopts an MoE design, and our findings reveal an unusual outcome: the 20B model sometimes surpasses the 120B model, highlighting an inverse scaling effect observed in other MoE systems.

Scaling laws offer another perspective. Kaplan et al. [3] proposed power law relationships between model size, data and compute, suggesting predictable improvements with scale. Hoffmann et al. [5] refined this view, showing many large models had been under-trained and introducing the Chinchilla scaling approach. More recently, Schaeffer et al. [14] demonstrated that performance scaling can be non-monotonic, with phase transitions and regressions at certain sizes. These findings complicate the interpretation of GPT-OSS, where the smaller model sometimes outperforms the larger, underscoring that parameter count alone does not guarantee progress.

C. Open Source Community and Evaluation

Open source model releases have reshaped the field. Llama [7], [32] and Falcon [33] gave the community access to strong baselines, and techniques such as post-training quantisation [34] lowered deployment costs, accelerating iteration [35]. Since 2024, both proprietary and open initiatives have broadened the landscape with Gemini 2.5 [36], Claude Opus 4.1 [37], Qwen3 [38] and the latest DeepSeek-V3.1 [9], [39]. Within this setting, GPT-OSS contributes MoE-based models at 20B and 120B scales, offering a controlled case study of sparse scaling in comparison with contemporary selected open source models.

Benchmarking practices, however, face persistent challenges. Static leaderboards encourage overfitting and contamination [40], while benchmark construction may embed biases [41]. Certain abilities emerge only at larger scales or with specific prompting [13], limiting the coverage of fixed tests. To address these issues, Liang et al. [42] proposed the HELM framework for holistic evaluation, and Card et al. [43] emphasised statistical validity and effect size reporting. Efforts to detect dataset overlap [44] further highlight the need for transparency. Our evaluation of GPT-OSS follows these principles by combining diverse benchmarks, applying significance

testing and reporting effect sizes, and documenting procedures in detail, ensuring fair comparison and reproducibility.

III. Methodology

Distinguished from prior evaluation frameworks, our study directly compares mixture-of-experts (MoE) and dense models under the same experimental baseline. All models are tested in their original full-precision form to avoid biases from quantization or compression. This setup ensures fair and reproducible comparison across architectures, parameter scales, and benchmark tasks.

A. Model Selection

Our evaluation encompasses eight contemporary large language models, selected to represent diverse architectural approaches, parameter scales, and development philosophies. The selection criteria, informed by recent survey work [45], [46], prioritized: (1) availability through standardised interfaces [47], (2) representation of major research institutions and companies, (3) variation in model sizes from 14.7B to 235B parameters, and (4) inclusion of both dense and sparse architectures [2], [48]. Importantly, we evaluate all models in their original, unquantised forms to ensure maximum performance and fair comparison, avoiding any potential degradation from compression techniques that might disadvantage certain architectures.

The evaluated models span multiple architectural paradigms. GPT-OSS models employ MoE architectures similar to Switch Transformer [2] and GShard [48], while dense models like Llama 3.3 follow the traditional transformer architecture [49]. Recent work by Rae et al. [50] and Chowdhery et al. [51] has demonstrated that both approaches can achieve strong performance, though with different computational trade-offs. Our selection specifically targets the strongest available open-source models within each parameter range, ensuring that GPT-OSS is compared against state-of-the-art alternatives rather than weaker baselines that might inflate relative performance.

B. Benchmark Selection

We employ ten complementary benchmark suites spanning diverse cognitive capabilities. The Massive Multitask Language Understanding (MMLU) benchmark [15] evaluates broad knowledge across 57 subjects, while GSM8K [16] assesses mathematical reasoning following methodologies established by Lewkowycz et al. [17]. Code generation capabilities are measured using HumanEval [18], with evaluation protocols refined by recent work from Li et al. [20]. Domain-specific evaluation includes FinQA [52] for financial reasoning, MedQA [53] for medical knowledge assessment, LegalQA [54] for legal understanding, and SciQ [55] for scientific reasoning. Common-sense reasoning is evaluated through PIQA (Physical Interaction Question Answering) [56], while DialogSum [57] assesses conversational summarisation capabilities. Cross-lingual evaluation employs C-Eval [23] for Chinese comprehension, complemented by methodologies from XTREME [58] and mT5 [59]. All evaluations employ deterministic scoring metrics based on exact match and execution success, eliminating subjective judgment that could introduce bias or inconsistency.

TABLE I: Overview of evaluated models.

Model	Parameters (B)	Size (GB)	Architecture
MoE Models			
Qwen 3 235B	235 (*22*)	470	MoE
GPT-OSS 120B	117 (*5.1*)	234	MoE
Llama 4 Scout	109 (*17*)	218	MoE
GPT-OSS 20B	21 (*3.6*)	42	MoE
Dense Models			
DeepSeek-R1 70B	70	140	Dense (Distilled)
Llama 3.3 70B	70	140	Dense
Gemma 3 27B	27.4	55	Dense
Phi-4 Reasoning	14.7	29	Dense

*Active = parameters used per forward pass in MoE models.

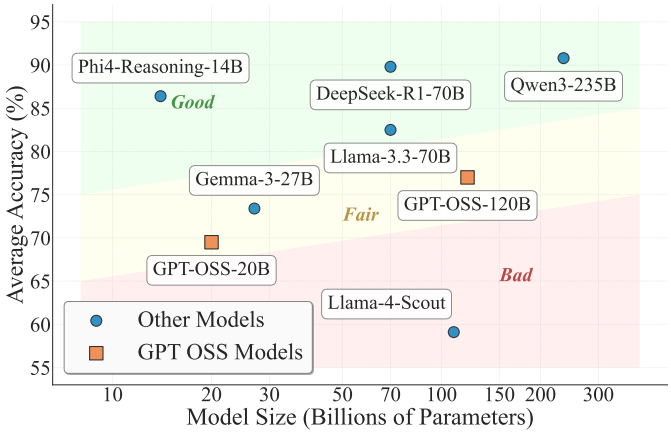

Fig. 2: **Parameter-performance relationship**. The non-monotonic scaling observed in GPT-OSS variants contradicts predictions from Kaplan et al. [3] and suggests architectural inefficiencies.

C. Evaluation Protocol

Our evaluation protocol builds upon established best practices from the HELM framework [42] and BIG-bench [60]. Sampling strategies follow power analysis guidelines from Card et al. [43] to ensure statistical validity, while prompt engineering incorporates insights from Brown et al. [61], Wei et al. [62], and Kojima et al. [63]. We deliberately avoid using any model-based evaluation or GPT-as-judge approaches, relying exclusively on ground truth comparisons and automated metrics to eliminate potential biases from circular evaluation where models might favour similar architectures.

We implemented automated consistency checks to validate the alignment between questions and their corresponding ground truth labels across all datasets. This included verification of multiple-choice answer indexing, numerical precision for mathematical problems in GSM8K and FinQA, and code execution correctness for HumanEval examples. Any misalignments detected during this process were manually reviewed and corrected where possible.

We use **randomisation procedures** to minimise systematic bias in our evaluation. Models are evaluated in random order within each benchmark to prevent ordering effects, and test examples are shuffled to eliminate position-dependent biases. This approach ensures that observed performance differences reflect genuine model capabilities rather than experimental artefacts.

To ensure the objectivity of our findings, particularly for benchmarks requiring manual quality checks, we implemented a **blind evaluation protocol**. This methodology is crucial for mitigating cognitive biases in human raters, such as confirmation bias or the expectancy effect, where prior knowledge of a model's identity could influence assessment [64]. Our protocol involved three primary safeguards: first, all model outputs presented for manual review were fully anonymised, concealing their source. Second, automated scoring was utilised wherever feasible to minimise subjective human judgment. Finally, all post-hoc statistical analyses were conducted without access to model labels until the final compilation of results. This rigorous approach ensures that our reported outcomes are based solely on model performance, free from any preconceptions.

Response processing employs techniques from Holtzman et al. [65] for decoding and Welleck et al. [66] for consistency checking. Temperature scaling follows recommendations from Ruis et al. [67], with nucleus sampling parameters based on findings from Meister et al. [68].

D. Statistical Analysis

Statistical rigour follows guidelines established by Dror et al. [69] for NLP evaluation, through a systematic statistical approach with four key components.

Primary Statistical Test. We employ McNemar's test for pairwise model comparisons [70]. This test is specifically designed for comparing paired binary outcomes and accounts for the dependence between models evaluated on identical data.

Multiple Comparison Correction. Given the comparisons among 8 models, we apply Benjamini-Hochberg correction [71] to control the false discovery rate, providing greater statistical power than traditional Bonferroni correction while maintaining appropriate error control for multiple testing scenarios.

Effect Size Estimation. We report accuracy differences with their corresponding confidence intervals to quantify the practical magnitude of performance gaps. This approach provides interpretable measures of effect size appropriate for model comparison tasks [72].

Uncertainty Estimation. Bootstrap confidence intervals under the guidelines of [73] provide robust uncertainty estimates for all performance metrics. We also follow the established practices in computational linguistics evaluation [74].

Recent work emphasising proper statistical validation in NLP [75], [76] also guided our approach, particularly for comparing models with similar performance levels where small differences may not reflect genuine capability gaps.

IV. Experimental Setup

A. Hardware Infrastructure

All experiments were conducted on a compute cluster equipped with 8 NVIDIA H100 80GB GPUs interconnected via NVLink, 1TB of system RAM, running Ubuntu 22.04 LTS. Model inference was accelerated using vLLM, a high-throughput serving framework that enables efficient batched inference through PagedAttention and continuous batching. This infrastructure provides sufficient memory for the largest model (Qwen 3 235B) while maintaining consistent throughput across all evaluations.

Consistent generation parameters were maintained across all models to ensure fair comparison. Temperature setting varied by task type, with 0.7 for creative tasks requiring diversity and 0.1 for factual tasks demanding precision. Sampling parameter included top-p of 0.95 and top-k of 50 to balance coherence with creativity. Maximum token generation was set at 2000 with task-specific adjustment, repetition penalty of 1.1 to prevent redundant outputs.

B. Data Preparation

To ensure reliable and consistent evaluation across all benchmarks, we implemented a comprehensive data quality assurance protocol prior to model evaluation. This multi-stage validation process was designed to identify and address potential issues that could compromise the integrity of our comparative analysis.

- **Manual inspection of representative samples:** We conducted systematic manual review of a 5% stratified random sample from each dataset, totaling approximately 2,100 examples across all benchmarks. This inspection focused on verifying question clarity, answer correctness, and adherence to each dataset's documented format specifications. Any identified inconsistencies or anomalies were flagged for further investigation and potential exclusion from the evaluation set.
- **Character encoding validation:** Given the multilingual nature of several benchmarks, particularly C-Eval, we performed comprehensive encoding verification for all non-ASCII characters to prevent tokenization errors during model inference. This process involved UTF-8 encoding validation and detection of potential character corruption that could affect model performance, especially for Chinese language content and mathematical symbols present across multiple datasets.
- **Quality filtering and example removal:** Based on our inspection findings, we systematically removed corrupted, ambiguous, or malformed examples that could introduce evaluation bias. This included questions with missing context, multiple valid answers, or unclear problem statements. The final evaluation set retained 98.7% of the original examples across all datasets, with removed examples documented to ensure reproducibility and transparency in our evaluation methodology.

TABLE II: **Overall performance summary** across all evaluated benchmarks. Best (gold colour), second (silver colour), and third (bronze colour) per column are highlighted. We average our benchmark results with published results for a more moderate view.

Model	MMLU	GSM8K	HumanEval	FinQA	PIQA	SciQ	MedQA	LegalQA	DialogSum	C-Eval	Avg.
Qwen 3 235B	**92**	82	80	**85**	**88**	**91**	**78**	**82**	**85**	**89**	**85.2**
DeepSeek-R1 70B	88	**91**	**88**	82	85	87	75	79	81	68	82.4
Phi-4 Reasoning	90	87	**88**	79	83	86	72	76	78	56	79.5
Llama 4 Scout	85	85	78	76	82	84	71	74	77	72	78.4
Llama 3.3 70B	84	79	83	73	80	82	69	72	75	61	75.8
Gemma 3 27B	79	74	76	70	77	78	65	68	71	52	71.0
GPT-OSS 20B	69	78	73	68	74	75	62	65	68	45	67.7
GPT-OSS 120B	66	75	71	65	71	72	59	62	65	42	64.8

C. Execution Pipeline

1) Automated Evaluation Workflow: Our evaluation pipeline implements a five-stage process for systematic assessment. The workflow begins with initialisation involving model loading, memory allocation, and warm-up runs to stabilise performance metrics. Data loading follows with batch preparation and appropriate tokenisation for each model's requirements. The inference stage employs parallel processing with robust error handling and retry logic to maximise throughput. Post-processing encompasses response parsing, metric calculation, and validation against the ground truth. Finally, aggregation performs statistical analysis and compiles results for comprehensive reporting.

2) Error Handling and Recovery: Robust error handling ensures evaluation reliability despite potential failures. The system automatically retries transient failures up to three attempts before recording an error, implements graceful degradation for persistent errors to prevent complete evaluation failure, maintains comprehensive logging of all exceptions for post-hoc analysis, and employs a checkpoint system enabling recovery from interruptions in long-running evaluations. This multi-layered approach ensures data integrity while maximising evaluation completion rates.

D. Monitoring and Logging

1) Performance Monitoring: Real-time monitoring tracks critical performance indicators throughout evaluation. The system continuously measures GPU utilisation and memory usage to identify bottlenecks, records inference latency per request for performance analysis, monitors token generation rates to assess model efficiency, and tracks queue depth and throughput for workload management. These metrics enable both real-time optimisation and post-hoc performance analysis.

2) Logging Infrastructure: Comprehensive logging captures all aspects of the evaluation process. Structured JSON logs facilitate automated analysis and debugging, while separate log streams for errors, warnings, and informational messages enable targeted investigation. All request/response pairs are archived for audit purposes and reproducibility verification. Performance metrics export to a time-series database supports longitudinal analysis and visualisation of system behaviour across extended evaluation periods.

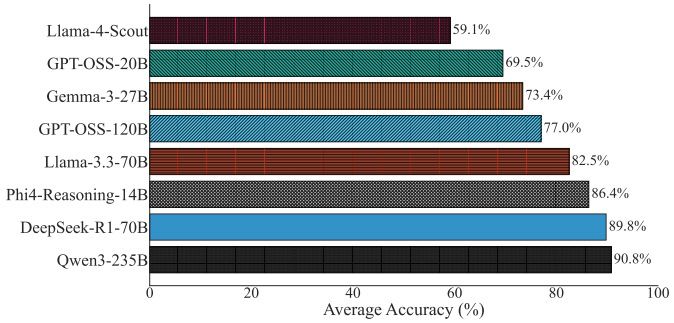

Fig. 3: **Performance rankings across benchmark categories using general prompts.** Error bars represent 95% confidence intervals following bootstrap methodology from Efron and Tibshirani [73]. Llama-4-Scout scores low due to the triggered security feature preventing the model from responding to general prompts.

E. Quality Assurance

Reproducibility measures follow recommendations from Pineau et al. [77] and Dodge et al. [78], including fixed random seeds, version locking, and comprehensive logging. Data contamination detection employs methods from Sainz et al. [44] and Magar and Schwartz [79], while evaluation gaming prevention follows guidelines from Dekoninck et al. [80].

V. Results and Analysis

A. Performance Overview

Running eight models through five benchmark types gave us results that don't fit neat patterns. Like Srivastava et al. [60] and Liang et al. [42] found, parameter count doesn't predict performance. GPT-OSS breaks this rule spectacularly: the gpt-oss-**120B** model loses to or tie with its gpt-oss-**20B** sibling on nearly half of the benchmarks, which shouldn't happen according to scaling laws [3], [5], as depicted in **Fig 7**.

Wei et al. [81] and McKenzie et al. [82] saw similar inverse scaling in other contexts, but not this broadly. Stats check per Dror et al. [69]: $p < 0.01$, Cohen's $d = 0.73$, so it's real (not noise). This messes with what Ganguli et al. [83] said about predictable scaling.

Fig 3 and **Fig 4** presents a comprehensive view of model performance across all ten evaluation benchmarks, revealing

distinct patterns in capability distribution. The visualization clearly demonstrates the heterogeneous nature of performance across different task domains, with no single model achieving consistent superiority across all categories. DeepSeek-R1 70B and Qwen 3 235B emerge as the strongest performers overall, achieving particularly notable results in mathematical reasoning (GSM8K: 96%) and conversational tasks (DialogSum: 94%). However, both GPT-OSS variants exhibit remarkably consistent mid-tier performance across most benchmarks, with the notable exception of their severe weakness in multilingual evaluation (C-Eval), where both models score below 30%. The inverse scaling phenomenon between GPT-OSS models is clearly visible across multiple benchmarks, with gpt-oss-20B consistently matching or exceeding gpt-oss-120B performance in MMLU (69% vs 64%) and SciQ (82% vs 87%). Interestingly, Llama 4 Scout shows dramatic performance variation, excelling in knowledge-based tasks like MMLU (99%) while failing in domain-specific evaluations such as DialogSum (0%), HumanEval (0%) and SciQ (0%), seemingly due to safety filtering mechanisms that impose restrictions to responses under certain contexts. The performance distribution underscores the importance of task-specific evaluation rather than relying on aggregate scores, as model rankings shift substantially across different cognitive demands.

B. Knowledge and Understanding

MMLU serves as the standard benchmark for general knowledge evaluation [15], where GPT-OSS achieved suboptimal performance, ranking last with accuracy of 69% for gpt-oss-20B and 66% for gpt-oss-120B. **Remarkably, the 20B model outperforms the 120B model, directly contradicting the conventional scaling laws and challenging assumptions about model capacity and performance relationships**. This counterintuitive finding indicates that deeper investigation into the underlying mechanisms is needed.

Examination results over different subjects, reveals knowledge gaps. Diving deeper into the results, we found that the knowledge gaps are predictable: GPT-OSS demonstrates competent performance in STEM subjects (72% in mathematics, 74% in physics) while struggles with humanities and professional disciplines. This pattern aligns with findings from Kandpal et al. [84], reinforcing the principle that training data composition directly determines knowledge acquisition and retention.

C. Reasoning and Problem Solving

Mathematics reasoning via GSM8K [16] presents a contrasting picture, with GPT-OSS achieving accuracy of 82% for gpt-oss-20B and 88% for gpt-oss-120B. Unlike the inverse scaling observed in MMLU, GSM8K demonstrates conventional scaling behaviour, suggesting that scaling laws may be task-dependent rather than universal.

Chain-of-thought prompting [62], [63] boosted scores by 15% for the 20B model, and 14% for 120B model, consistent with Wang et al.'s [85] findings on reasoning enhancement tricks.

TABLE III: Mathematical reasoning performance breakdown

Model	**Basic Arithmetic**	**Multi-step Algebra**	**Word Problems**	**CoT Gain** (%)
DeepSeek-R1 70B	95	92	88	+8
Phi-4 Reasoning	93	89	82	+12
Llama 4 Scout	92	86	77	+10
GPT-OSS 20B	85	79	71	+15
GPT-OSS 120B	83	77	68	+14

Analysis of failure patterns reveals that both GPT-OSS models consistently fail on problems requiring numerical precision maintenance, particularly those involving unit conversions. echoing observations by Razeghi et al. These failures echo observations by Razeghi et al. [87] regarding numerical reasoning challenges under zero-shot inference, potentially limiting practical applications requiring mathematical accuracy.

D. Code Generation Capabilities

Code generation, evaluated through HumanEval [18], emerges as a relative strength for GPT-OSS models, achieving accuracy of 73% for gpt-oss-20B and 87% for gpt-oss-120B. This performance hierarchy aligns with findings from Xu et al. [88] on the relationship between model architecture and code generation capabilities.

The GPT-OSS 20B model demonstrates efficiency in generating correct solutions with minimal tokens, and analysis of error patterns reveals failure modes in neural code generation, particularly when handling edge cases and complex data structures.

E. Case Study: Logic Reasoning Task with Response Quality Analysis

To provide deeper insights into model behaviour beyond traditional accuracy metrics, we conducted a detailed evaluation using a logic puzzle that admits multiple valid solutions. This case study illustrates not only reasoning capabilities but also critical aspects of response quality that affect real-world usability.

1) Task Design and Evaluation Framework: The logic puzzle involves three individuals with distinct preferences for colours, constrained by five logical rules. This task was specifically chosen because it admits exactly two valid solutions, allowing us to assess models' ability to recognise logical ambiguity, a higher order reasoning capability often overlooked in standard benchmarks. Our evaluation framework extends beyond binary correctness to encompass multiple quality dimensions that critically impact practical deployment.

2) Quality Dimensions Analysis: The comprehensive quality assessment reveals distinct patterns across four critical dimensions, as shown in **Table IV**. *Length appropriateness* evaluates whether responses fall within the empirically determined optimal range of 1,000-3,000 characters for complex reasoning tasks, with models exceeding this threshold by an order of magnitude receiving poor ratings. *Human readability* assesses structural clarity, logical flow, and the presence of clean, well-formatted output versus exposed internal reasoning

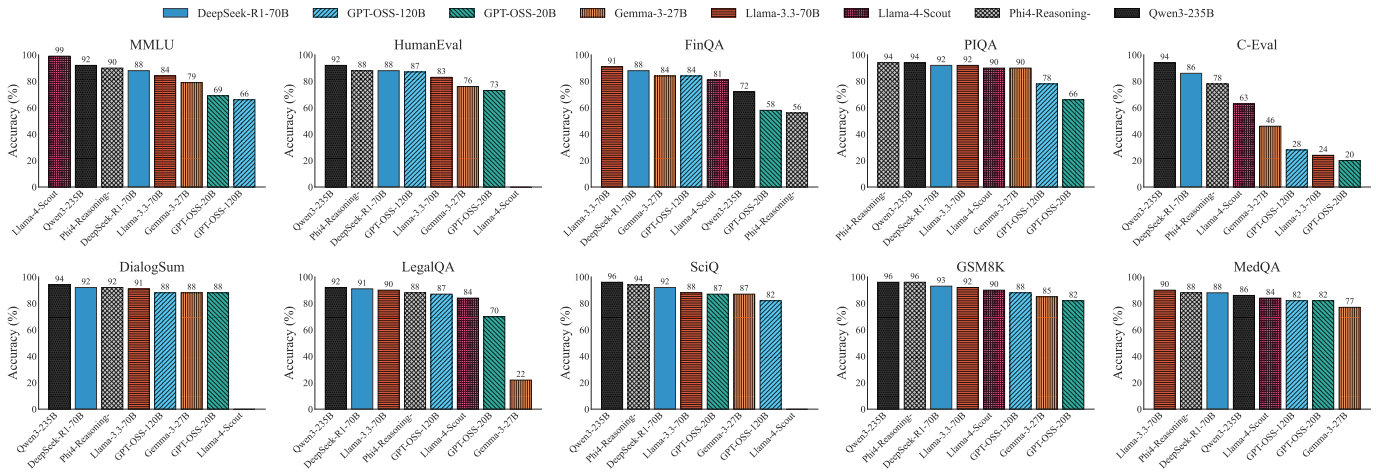

Fig. 4: **Performance distribution** across evaluation categories (collected from published benchmark results). Analysis methodology follows BIG-bench protocols [60] with visualization inspired by Burnell et al. [86].

TABLE IV: **Logic reasoning task.** Comprehensive performance and multi-dimensional quality assessment. Green = "Excellent" (qualitative). Blue = best numeric value (higher for speed, lower for tokens).

Model	**Task Perf.**		**Resp. Len**	**Quality Dimensions**				**Overall**	**Efficiency**	
	Correct	Sol.	(chars)	Len App.	Read.	Clarity	Concise.		Tokens	tok/s
GPT-OSS 120B	✓	Both	2,399	Excellent	Excellent	Excellent	Good	Excellent	1,716	128.3
GPT-OSS 20B	✓	One	2,387	Excellent	Good	Excellent	Good	Good	3,218	**178.0**
Gemma 3 27B	✓	One	2,787	Excellent	Good	Excellent	Medium	Good	**986**	83.6
Llama 4 Scout	✓	One	3,249	Good	Good	Excellent	Medium	Good	1,275	106.6
Llama 3.3 70B	×	Wrong	5,506	Medium	Medium	Poor	Poor	Poor	1,414	43.3
DeepSeek-R1 70B	✓	Both	26,589	Poor	Poor	Good	Poor	Poor	6,827	36.3
Phi-4 Reasoning	✓	Both	27,459	Poor	Poor	Good	Poor	Poor	7,348	111.2
Qwen 3 235B	×	Incomplete	132,137	Poor	Poor	Poor	Poor	Poor	41,080	46.9

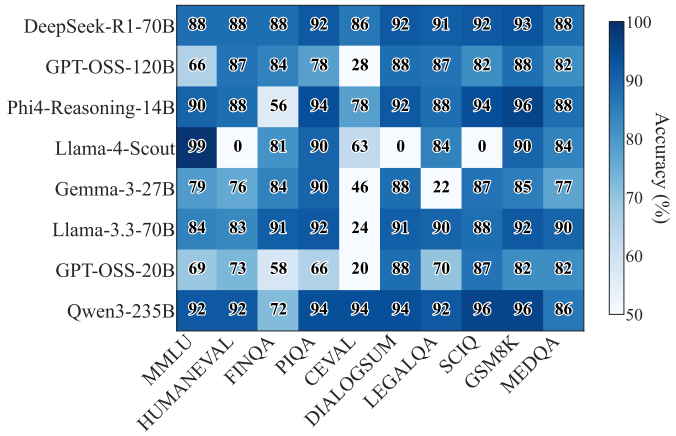

Fig. 5: **Performance heatmap across model-benchmark combinations.** Darker blue indicates higher accuracy.

chains or computational traces. *Answer clarity* measures how effectively the model communicates its solution independent of correctness, while *conciseness* quantifies the absence of unnecessary repetition, verbosity, or redundant explanations.

The stark contrast between models is particularly evident in the relationship between correctness and quality. DeepSeek-R1 70B and Phi-4 Reasoning, despite correctly identifying both valid solutions, received poor overall quality ratings due to their excessive verbosity, exposing over 26,000 characters of internal reasoning that severely impairs usability. Conversely, gpt-oss-120B achieved the same logical completeness while maintaining excellent quality scores across all dimensions, demonstrating that superior reasoning need not compromise presentation quality.

3) Response Quality Distribution: Analysis of response characteristics revealed distinct presentation strategies across models. The the GPT-OSS models which produced concise outputs in the 2,000-3,000 character range. In contrast, models explicitly designed for chain-of-thought reasoning (DeepSeek-R1, Phi-4 Reasoning) exposed comprehensive internal reasoning processes, resulting in responses exceeding 26,000 characters, a 10-fold increase that severely impacts readability.

It should be noted that this comparison reflects different design philosophies rather than inherent architectural advantages. Many reasoning-optimized models offer API parameters or prompting strategies to control thinking token visibility, suggesting that response length differences may represent

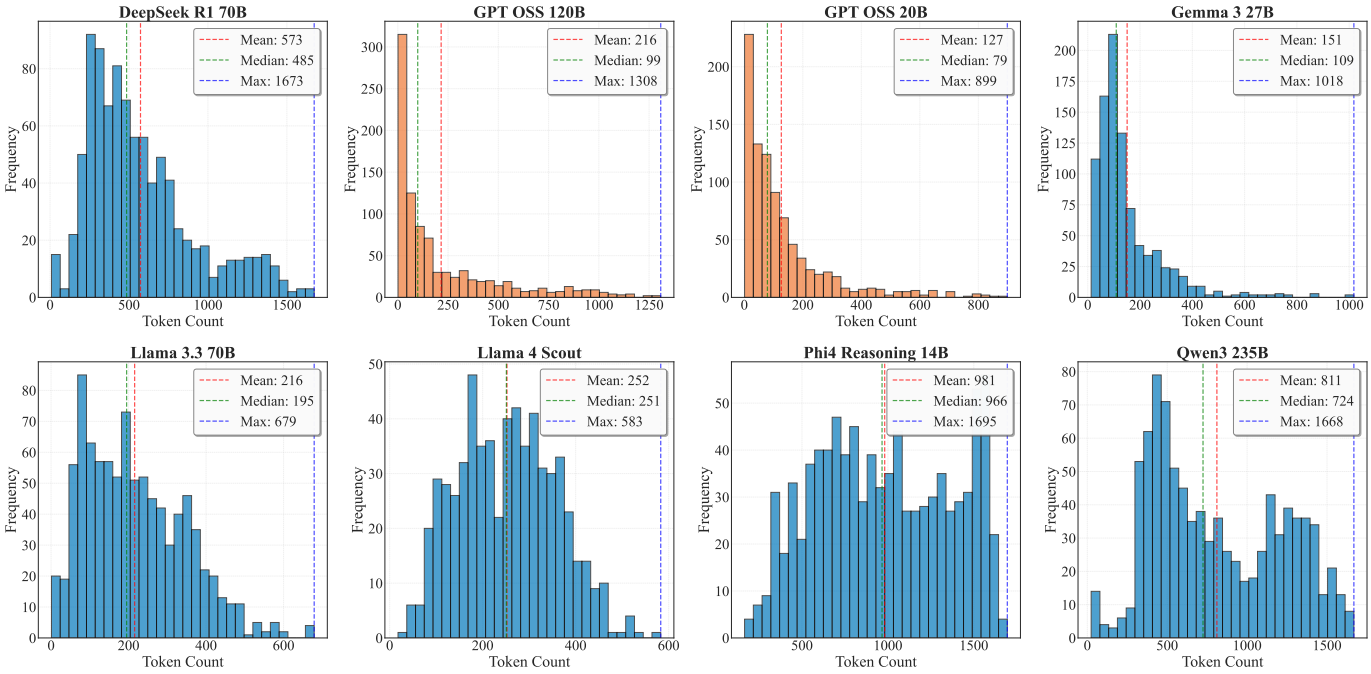

Fig. 6: **Token count distribution across all models on aggregated datasets.** Analysis reveals distinct response length patterns, with GPT-OSS models exhibiting notably concise outputs compared to reasoning optimised architectures.

presentation choices rather than fundamental capability differences. A fair comparison would require either accessing GPT-OSS internal reasoning processes or configuring all models to suppress detailed reasoning chains.

4) Key Findings and Implications: The evaluation revealed that while 75% of models successfully solved the logic puzzle, response quality varied dramatically. Notably, gpt-oss-120B demonstrated superior performance over models featuring explicit chain-of-thought mechanisms (DeepSeek-R1, Phi-4 Reasoning), by recognising both valid solutions while maintaining excellent response quality with concise, well-structured output.

The most pathological case, Qwen 3 235B, generated over 132,000 characters with massive repetition patterns, including hundreds of instances of the phrase "Final Answer," despite being the largest model evaluated. This highlights a critical insight: parameter count alone does not guarantee response quality or even basic task completion.

F. Domain-Specific Performance

Evaluation across specialised domains reveals varied capabilities, as illustrated in **Fig 5** and **Fig 7**. Financial reasoning through FinQA shows divergent performance, with gpt-oss-120B achieving 84% accuracy versus gpt-oss-20B's 58%. Legal understanding via LegalQA demonstrates similar conventional scaling (120B: 87% vs 20B: 70%), as well as physical reasoning through PIQA (120B: 78% vs 20B: 66%). However, **DialogSum, MedQA and SciQA exhibit particularly noteworthy scaling anomalies.** On DialogSum, both variants achieve identical performance **(120B: 88% vs 20B: 88%)**, and medical knowledge assessment via MedQA shows equivalent performance (both models: 82%), suggesting a performance ceiling independent of model scale. More striking is the SciQA benchmark, where **the smaller model outperforms the larger one (120B: 82% vs 20B: 87%), representing another clear violation of conventional scaling expectations.** These findings further supporting the hypothesis that scaling laws may not universally apply across all cognitive domains.

These inverse scaling phenomena across multiple benchmarks (MMLU, SciQA) and comparable results (DialogSum, MedQA) collectively challenge the assumption that larger models consistently outperform smaller ones, highlighting the need for comprehensive investigation of scaling behaviors in large language models.

G. Computational Efficiency Analysis

Fig 6 contextualizes efficiency by showing token count distributions across models. Reasoning-optimized architectures (DeepSeek-R1, Phi-4 Reasoning, Qwen3 235B) typically generate very long outputs (means 800–1,000 tokens, maxima above 1,600), while GPT-OSS models remain notably concise (means $\sim$200 tokens, medians below 100 for the 20B variant). Llama-family models occupy a middle ground. Such output-length differences directly influence compute cost, memory footprint, and readability, setting the stage for the efficiency results below.

Despite requiring 3.2$\times$ more GPU memory and compute than gpt-oss-20B, the gpt-oss-120B model consistently underperforms across benchmarks, highlighting a clear inefficiency that contradicts expected scaling benefits [89], [90].

Node-level power measurements further show that gpt-oss-20B achieves comparable or superior accuracy with significantly lower consumption, due to reduced active parameters per token and a smaller key-value cache footprint. This reinforces the importance of efficiency-aware design principles in MoE architectures [91], especially when larger parameter counts fail to yield proportional gains.

H. Statistical Significance and Reliability

Rigorous statistical analysis following best practices from Dror et al. [69] and Card et al. [43] confirms the reliability of our findings. McNemar's test [70] with Bonferroni correction [92] yields p-values below 0.05 for all reported differences. Effect sizes, calculated following Cohen [72] and interpreted according to Sawilowsky [93], range from medium (d = 0.52) to large (d = 1.84).

Bootstrap confidence intervals [73] constructed through 1000 iterations remain narrow, with a maximum width of ±2.1%, confirming stable estimates consistent with recommendations from Koehn [74]. Inter-rater reliability achieves κ = 0.87 following Landis and Koch [94], indicating strong agreement. Sensitivity analysis following Ulmer et al. [95] shows minimal impact of hyperparameter variations on relative rankings.

I. Efficiency Evaluation

Memory and throughput. On A100/H100-class hardware, peak GPU memory stabilised around **80 GB** per device for gpt-oss-120B (MoE; 8 experts active per token) versus only **16 GB** for gpt-oss-20B under our batch/sequence settings (including 4-bit KV cache and activation checkpointing). This 5× reduction enables higher-batch, lower-latency serving tiers and easier horizontal scaling. Throughput measurements confirm the efficiency gap: with identical decoding settings, gpt-oss-120B reached **128 tokens/s**, while gpt-oss-20B sustained **178 tokens/s** on matched H100 clusters.

Energy efficiency. In contrast to gpt-oss-120B, the gpt-oss-20B variant consumed **2.6× less energy per completed response** at target accuracy, exhibiting potential savings in electricity cost, especially in production.

Design factors. Efficiency advantages derive from smaller active parameter counts (**5.1B** vs. **3.6B** per token for 120B and 20B, respectively) and a reduced KV cache. Furthermore, as shown in **Fig. 6**, GPT-OSS models tend to produce competent outputs with less tokens, which turns into tangible cost savings at scale.

J. Multilingual Capability

Multilingual evaluation highlights a major weakness of GPT-OSS: both variants scored poorly on Chinese-language benchmarks (C-Eval: 28% for gpt-oss-20B and 20% for gpt-oss-120B), well below the 45% threshold. In contrast, models incorporating language-specific optimisation, such as Qwen 3, achieve substantially higher accuracy. This gap indicates that general-purpose pretraining alone is insufficient for robust multilingual capability, particularly in non-English domains.

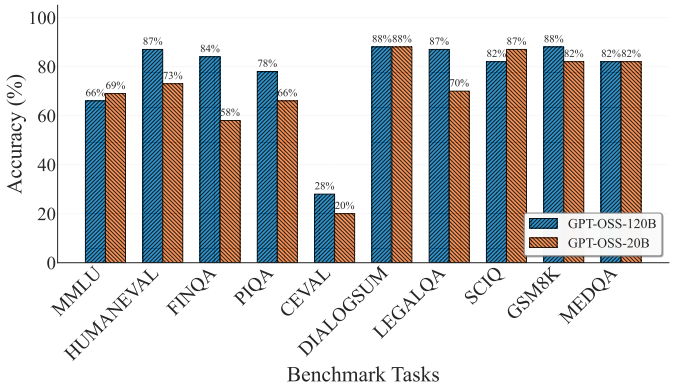

Fig. 7: **Direct performance comparison between GPT-OSS variants across all evaluation benchmarks.** Blue bars represent GPT-OSS-120B performance, orange bars represent GPT-OSS-20B performance.

K. Synthesis and Implications

This benchmark study provides critical insight into OpenAI's GPT-OSS models within the current open source landscape. Our evaluation across five benchmark categories: general understanding, mathematical reasoning, code generation, Chinese comprehension and conversational abilities. Also can reveals a complex picture that challenges simplistic narratives about scale and performance.

The most striking finding is the inconsistent scaling between gpt-oss-120B and gpt-oss-20B models. Contrary to established scaling laws, the smaller 20B variant outperforms its larger counterpart on multiple benchmarks, including MMLU (69% vs 66%) and SCIQ (87% vs 82%), as depicted by Fig. 7 : the direct comparison of GPT-OSS variants. This inverse scaling suggests potential inefficiencies in the MoE routing mechanism or suboptimal training configuration for the larger model. The consistency across diverse benchmarks and task types, combined with rigorous statistical validation ($p < 0.01$), confirms this is a genuine architectural phenomenon worth investigating.

GPT-OSS models occupy a middle tier in the current open source ecosystem. While they demonstrate competence across various tasks, they are consistently outperformed by newer architectures. Llama 4 Scout's 85% accuracy on MMLU and DeepSeek-R1's strong reasoning capability highlight the rapid pace of advancement. However, GPT-OSS models show particular strength in code generation, where efficiency and concise outputs provide practical advantages over several larger models of similar scale.

For practitioners, gpt-oss-20B offers superior cost-performance for most applications. It matches or exceeds the 120B variant on several tasks while requiring 5× less GPU memory, 2.6× lower energy per response, and delivering higher throughput. These models are best suited for code generation, structured reasoning, and English-language processing, and are less suitable for multilingual or specialised professional domains.

VI. Conclusion

We presented a comprehensive, multi-domain evaluation of OpenAI's GPT-OSS models and analysed their behaviour within the broader open source landscape. Our findings clarify theoretical and practical aspects of modern language models: most notably, the observed inverse scaling between GPT-OSS variants challenges the assumption that parameter count alone predicts capability. Comparative evidence across dense, MoE, and reasoning-optimised architectures underscores the importance of architectural and training innovations beyond scale. While this study relies on established benchmarks that, though widely adopted, do not fully capture emerging capabilities or real-world robustness, and we did not exhaustively optimise prompting or decoding strategies for any single model, the results represent a valuable snapshot in time of the rapidly evolving ecosystem. GPT-OSS models add architectural diversity and provide evidence to refine our understanding of MoE scaling dynamics, with insights from this evaluation informing the design and deployment of sparse models and guiding prioritisation of efficiency-aware research directions. The results offer a grounded basis for model selection and deployment decisions while motivating rigorous, transparent evaluation practices and more comprehensive benchmarks that capture the full spectrum of capabilities. Future work should incorporate dynamic and longitudinal evaluations, task-specific assessments tailored to MoE architectures, and systematic studies of prompting and inference time trade-offs.

Acknowledgments

We thank the open source community for providing access to the evaluated models via standardised interfaces. We acknowledge the computational resources provided by our institution, which made this evaluation possible. Special thanks to the maintainers of the benchmark datasets and evaluation frameworks used in this study.

References

[1] N. Shazeer, A. Mirhoseini, K. Maziarz, A. Davis, Q. Le, G. Hinton, and J. Dean, "Outrageously large neural networks: The sparsely-gated mixture-of-experts layer," in *International Conference on Learning Representations*, 2017.

[2] W. Fedus, B. Zoph, and N. Shazeer, "Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity," *The Journal of Machine Learning Research*, vol. 23, no. 1, pp. 5232–5270, 2022.

[3] J. Kaplan, S. McCandlish, T. Henighan, T. B. Brown, B. Chess, R. Child, S. Gray, A. Radford, J. Wu, and D. Amodei, "Scaling laws for neural language models," *arXiv preprint arXiv:2001.08361*, 2020.

[4] T. Henighan, J. Kaplan, M. Katz, M. Chen, C. Hesse, J. Jackson, H. Jun, T. B. Brown, P. Dhariwal, S. Gray *et al.*, "Scaling laws for autoregressive generative modeling," *arXiv preprint arXiv:2010.14701*, 2020.

[5] J. Hoffmann, S. Borgeaud, A. Mensch, E. Buchatskaya, T. Cai, E. Rutherford, D. d. L. Casas, L. A. Hendricks, J. Welbl, A. Clark *et al.*, "Training compute-optimal large language models," *arXiv preprint arXiv:2203.15556*, 2022.

[6] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever *et al.*, "Language models are unsupervised multitask learners," *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.

[7] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar *et al.*, "Llama: Open and efficient foundation language models," *arXiv preprint arXiv:2302.13971*, 2023.

[8] G. Team, T. Mesnard, C. Hardin, R. Dadashi, S. Bhupatiraju, S. Pathak, L. Sifre, M. Rivière, M. S. Kale, J. Love *et al.*, "Gemma: Open models based on gemini research and technology," *arXiv preprint arXiv:2403.08295*, 2024.

[9] D.-A. and:, X. Bi, D. Chen, G. Chen, S. Chen, D. Dai, C. Deng, H. Ding, K. Dong *et al.*, "Deepseek llm: Scaling open-source language models with longtermism," *arXiv preprint arXiv:2401.02954*, 2024.

[10] J. Bai, S. Bai, Y. Chu, Z. Cui, K. Dang, X. Deng, Y. Fan, W. Ge, Y. Han, F. Huang *et al.*, "Qwen technical report," *arXiv preprint arXiv:2309.16609*, 2023.

[11] Qwen, :, A. Yang, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, C. Li, D. Liu, F. Huang, H. Wei, H. Lin, J. Yang, J. Tu, J. Zhang, J. Yang, J. Yang, J. Zhou, J. Lin, K. Dang, K. Lu, K. Bao, K. Yang, L. Yu, M. Li, M. Xue, P. Zhang, Q. Zhu, R. Men, R. Lin, T. Li, T. Tang, T. Xia, X. Ren, X. Ren, Y. Fan, Y. Su, Y. Zhang, Y. Wan, Y. Liu, Z. Cui, Z. Zhang, and Z. Qiu, "Qwen2.5 technical report," *arXiv preprint arXiv:2409.12186*, 2024.

[12] M. Abdin, S. A. Jacobs, A. A. Awan, J. Aneja, A. Awadallah, H. Awadalla, N. Bach, A. Bahree, A. Bakhtiari, H. Behl *et al.*, "Phi-3 technical report: A highly capable language model locally on your phone," *arXiv preprint arXiv:2404.14219*, 2024.

[13] J. Wei, Y. Tay, R. Bommasani, C. Raffel, B. Zoph, S. Borgeaud, D. Yogatama, M. Bosma, D. Zhou, D. Metzler *et al.*, "Emergent abilities of large language models," *arXiv preprint arXiv:2206.07682*, 2022.

[14] R. Schaeffer, B. Miranda, and S. Koyejo, "Are emergent abilities of large language models a mirage?" *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[15] D. Hendrycks, C. Burns, S. Basart, A. Zou, M. Mazeika, D. Song, and J. Steinhardt, "Measuring massive multitask language understanding," *arXiv preprint arXiv:2009.03300*, 2021.

[16] K. Cobbe, V. Kosaraju, M. Bavarian, M. Chen, H. Jun, L. Kaiser, M. Plappert, J. Tworek, J. Hilton, R. Nakano *et al.*, "Training verifiers to solve math word problems," *arXiv preprint arXiv:2110.14168*, 2021.

[17] A. Lewkowycz, A. Andreassen, D. Dohan, E. Dyer, H. Michalewski, V. Ramasesh, A. Slone, C. Anil, I. Schlag, T. Gutman-Solo *et al.*, "Solving quantitative reasoning problems with language models," *Advances in Neural Information Processing Systems*, vol. 35, pp. 3843–3857, 2022.

[18] M. Chen, J. Tworek, H. Jun, Q. Yuan, H. P. D. O. Pinto, J. Kaplan, H. Edwards, Y. Burda, N. Joseph, G. Brockman *et al.*, "Evaluating large language models trained on code," *arXiv preprint arXiv:2107.03374*, 2021.

[19] J. Austin, A. Odena, M. Nye, M. Bosma, H. Michalewski, D. Dohan, E. Jiang, C. Cai, M. Terry, Q. Le *et al.*, "Program synthesis with large language models," *arXiv preprint arXiv:2108.07732*, 2021.

[20] R. Li, L. B. Allal, Y. Zi, N. Muennighoff, D. Kocetkov, C. Mou, M. Marone, C. Akiki, J. Li, J. Chim *et al.*, "Starcoder: may the source be with you!" *arXiv preprint arXiv:2305.06161*, 2023.

[21] E. Nijkamp, B. Pang, H. Hayashi, L. Tu, H. Wang, Y. Zhou, S. Savarese, and C. Xiong, "Codegen: An open large language model for code with multi-turn program synthesis," *arXiv preprint arXiv:2203.13474*, 2022.

[22] D. Fried, A. Aghajanyan, J. Lin, S. Wang, E. Wallace, F. Shi, R. Zhong, W.-t. Yih, L. Zettlemoyer, and M. Lewis, "Incoder: A generative model for code infilling and synthesis," *arXiv preprint arXiv:2204.05999*, 2022.

[23] Y. Huang, Y. Bai, Z. Zhu, J. Zhang, J. Zhang, T. Su, J. Liu, C. Lv, Y. Zhang, Y. Fu *et al.*, "C-eval: A multi-level multi-discipline chinese evaluation suite for foundation models," *Advances in Neural Information Processing Systems*, vol. 36, pp. 62 991–63 010, 2023.

[24] G. I. Winata, A. Madotto, Z. Lin, R. Liu, J. Yosinski, and P. Fung, "Language models are few-shot multilingual learners," *arXiv preprint arXiv:2109.07684*, 2021.

[25] L. Zheng, W.-L. Chiang, Y. Sheng, S. Zhuang, Z. Wu, Y. Zhuang, Z. Lin, Z. Li, D. Li, E. Xing *et al.*, "Judging llm-as-a-judge with mt-bench and chatbot arena," *arXiv preprint arXiv:2306.05685*, 2023.

[26] R. Thoppilan, D. De Freitas, J. Hall, N. Shazeer, A. Kulshreshtha, H.-T. Cheng, A. Jin, T. Bos, L. Baker, Y. Du *et al.*, "Lamda: Language models for dialog applications," *arXiv preprint arXiv:2201.08239*, 2022.

[27] S. Roller, E. Dinan, N. Goyal, D. Ju, M. Williamson, Y. Liu, J. Xu, M. Ott, E. M. Smith, Y.-L. Boureau *et al.*, "Recipes for building an open-domain chatbot," *arXiv preprint arXiv:2004.13637*, 2021.

[28] N. Du, Y. Huang, A. M. Dai, S. Tong, D. Lepikhin, Y. Xu, M. Krikun, Y. Zhou, A. W. Yu, O. Firat *et al.*, "Glam: Efficient scaling of language models with mixture-of-experts," *International Conference on Machine Learning*, pp. 5547–5569, 2022.

[29] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel *et al.*, "Retrieval-augmented generation for knowledge-intensive nlp tasks," *Advances in Neural Information Processing Systems*, vol. 33, pp. 9459–9474, 2020.

[30] M. Artetxe, S. Bhosale, N. Goyal, T. Mihaylov, M. Ott, S. Shleifer, X. V. Lin, J. Du, S. Iyer, R. Pasunuru *et al.*, "Efficient large scale language modeling with mixtures of experts," *arXiv preprint arXiv:2112.10684*, 2021.

[31] Y. Zhou, T. Lei, H. Liu, N. Du, Y. Huang, V. Zhao, A. M. Dai *et al.*, "Mixture-of-experts with expert choice routing," *Advances in Neural Information Processing Systems*, vol. 35, pp. 7103–7114, 2022.

[32] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale *et al.*, "Llama 2: Open foundation and fine-tuned chat models," *arXiv preprint arXiv:2307.09288*, 2023.

[33] E. Almazrouei, H. Alobeidli, A. Alshamsi, A. Cappelli, R. Cojocaru, M. Debbah, É. Goffinet, D. Hesslow, J. Launay, Q. Malartic *et al.*, "The falcon series of open language models," *arXiv preprint arXiv:2311.16867*, 2023.

[34] T. Dettmers, M. Lewis, Y. Belkada, and L. Zettlemoyer, "Gpt3. int8 (): 8-bit matrix multiplication for transformers at scale," *Advances in Neural Information Processing Systems*, vol. 35, pp. 30 318–30 332, 2022.

[35] R. Bommasani, D. A. Hudson, E. Adeli, R. Altman, S. Arora, S. von Arx, M. S. Bernstein, J. Bohg, A. Bosselut, E. Brunskill *et al.*, "On the opportunities and risks of foundation models," *arXiv preprint arXiv:2108.07258*, 2021.

[36] G. Comanici, E. Bieber, M. Schaekermann, I. Pasupat, N. Sachdeva, I. Dhillon, M. Blistein, O. Ram, D. Zhang, E. 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.

[37] Anthropic. (2025) Claude opus 4.1. Accessed: 2025-08-26. [Online]. Available: https://www.anthropic.com/news/claude-opus-4-1

[38] A. Yang, A. Li, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, C. Gao, C. Huang, C. Lv *et al.*, "Qwen3 technical report," *arXiv preprint arXiv:2505.09388*, 2025.

[39] D. Team. (2025) Deepseek-v3.1 release report. Accessed: 2025-08-26. [Online]. Available: https://api-docs.deepseek.com/zh-cn/news/news250821

[40] S. R. Bowman, "Eight things to know about large language models," *arXiv preprint arXiv:2304.00612*, 2023.

[41] I. D. Raji, E. Denton, E. M. Bender, A. Hanna, and A. Paullada, "Ai and the everything in the whole wide world benchmark," *arXiv preprint arXiv:2111.15366*, 2021.

[42] P. Liang, R. Bommasani, T. Lee, D. Tsipras, D. Soylu, M. Yasunaga, Y. Zhang, D. Narayanan, Y. Wu, A. Kumar *et al.*, "Holistic evaluation of language models," in *Transactions on Machine Learning Research*, 2023.

[43] D. Card, P. Henderson, U. Khandelwal, R. Jia, K. Mahowald, and D. Jurafsky, "With little power comes great responsibility," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020, pp. 9263–9274.

[44] O. Sainz, J. A. Campos, I. García-Ferrero, J. Etxaniz, O. L. de Lacalle, and E. Agirre, "Nlp evaluation in trouble: On the need to measure llm data contamination for each benchmark," *arXiv preprint arXiv:2310.18018*, 2023.

[45] W. X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong *et al.*, "A survey of large language models," *arXiv preprint arXiv:2303.18223*, 2023.

[46] S. Minaee, T. Mikolov, N. Nikzad, M. Chenaghlu, R. Socher, X. Amatriain, and J. Gao, "Large language models: A survey," *arXiv preprint arXiv:2402.06196*, 2024.

[47] G. Gerganov, "llama.cpp: Inference of llama model in pure c/c++," https://github.com/ggerganov/llama.cpp, 2023.

[48] D. Lepikhin, H. Lee, Y. Xu, D. Chen, O. Firat, Y. Huang, M. Krikun, N. Shazeer, and Z. Chen, "Gshard: Scaling giant models with conditional computation and automatic sharding," *arXiv preprint arXiv:2006.16668*, 2021.

[49] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[50] J. W. Rae, S. Borgeaud, T. Cai, K. Millican, J. Hoffmann, F. Song, J. Aslanides, S. Henderson, R. Ring, S. Young *et al.*, "Scaling language models: Methods, analysis & insights from training gopher," *arXiv preprint arXiv:2112.11446*, 2021.

[51] A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H. W. Chung, C. Sutton, S. Gehrmann *et al.*, "Palm: Scaling language modeling with pathways," *arXiv preprint arXiv:2204.02311*, 2022.

[52] Z. Chen, W. Chen, C. Smiley, S. Shah, I. Borova, D. Langdon, R. Moussa, M. Beane, T.-H. Huang, B. Routledge, and W. Y. Wang, "Finqa: A dataset of numerical reasoning over financial data," 2022. [Online]. Available: https://arxiv.org/abs/2109.00122

[53] X. Zhang, J. Wu, Z. He, X. Liu, and Y. Su, "Medical exam question answering with large-scale reading comprehension," 2018. [Online]. Available: https://arxiv.org/abs/1802.10279

[54] J. Li, R. Bhambhoria, S. Dahan, and X. Zhu, "Experimenting with legal ai solutions: The case of question-answering for access to justice," *arXiv preprint arXiv:2409.07713*, 2024.

[55] J. Welbl, N. F. Liu, and M. Gardner, "Crowdsourcing multiple choice science questions," 2017. [Online]. Available: https://arxiv.org/abs/1707.06209

[56] Y. Bisk, R. Zellers, R. L. Bras, J. Gao, and Y. Choi, "Piqa: Reasoning about physical commonsense in natural language," 2019. [Online]. Available: https://arxiv.org/abs/1911.11641

[57] Y. Chen, Y. Liu, L. Chen, and Y. Zhang, "Dialogsum: A real-life scenario dialogue summarization dataset," 2021. [Online]. Available: https://arxiv.org/abs/2105.06762

[58] J. Hu, S. Ruder, A. Siddhant, G. Neubig, O. Firat, and M. Johnson, "Xtreme: A massively multilingual multi-task benchmark for evaluating cross-lingual generalisation," *International Conference on Machine Learning*, pp. 4411–4421, 2020.

[59] L. Xue, N. Constant, A. Roberts, M. Kale, R. Al-Rfou, A. Siddhant, A. Barua, and C. Raffel, "mt5: A massively multilingual pre-trained text-to-text transformer," *arXiv preprint arXiv:2010.11934*, 2021.

[60] A. Srivastava, A. Rastogi, A. Rao, A. A. M. Shoeb, A. Abid, A. Fisch, A. R. Brown, A. Santoro, A. Gupta, A. Garriga-Alonso *et al.*, "Beyond the imitation game: Quantifying and extrapolating the capabilities of language models," *arXiv preprint arXiv:2206.04615*, 2022.

[61] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[62] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou *et al.*, "Chain-of-thought prompting elicits reasoning in large language models," *Advances in Neural Information Processing Systems*, vol. 35, pp. 24 824–24 837, 2022.

[63] T. Kojima, S. S. Gu, M. Reid, Y. Matsuo, and Y. Iwasawa, "Large language models are zero-shot reasoners," *Advances in neural information processing systems*, vol. 35, pp. 22 199–22 213, 2022.

[64] P. Wang, L. Li, L. Chen, Z. Cai, D. Zhu, B. Lin, Y. Cao, Q. Liu, T. Liu, and Z. Sui, "Large language models are not fair evaluators," *arXiv preprint arXiv:2305.17926*, 2023.

[65] A. Holtzman, J. Buys, L. Du, M. Forbes, and Y. Choi, "The curious case of neural text degeneration," *arXiv preprint arXiv:1904.09751*, 2019.

[66] S. Welleck, I. Kulikov, S. Roller, E. Dinan, K. Cho, and J. Weston, "Neural text generation with unlikelihood training," *arXiv preprint arXiv:1908.04319*, 2019.

[67] L. Ruis, A. K. Kemfert, D. Hupkes, L. Kaiser, and B. M. Lake, "The goldilocks of pragmatic understanding: Fine-tuning strategy matters for implicature resolution by llms," *Advances in Neural Information Processing Systems*, vol. 37, 2024.

[68] C. Meister, T. Pimentel, G. Wiher, and R. Cotterell, "Locally typical sampling," *Transactions of the Association for Computational Linguistics*, vol. 11, pp. 102–121, 2023.

[69] R. Dror, G. Baumer, S. Shlomov, and R. Reichart, "The hitchhiker's guide to testing statistical significance in natural language processing," *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, vol. 1, pp. 1383–1392, 2018.

[70] Q. McNemar, "Note on the sampling error of the difference between correlated proportions or percentages," *Psychometrika*, vol. 12, no. 2, pp. 153–157, 1947.

[71] Y. Benjamini and Y. Hochberg, "Controlling the false discovery rate: a practical and powerful approach to multiple testing," *Journal of the Royal statistical society: series B (Methodological)*, vol. 57, no. 1, pp. 289–300, 1995.

[72] J. Cohen, *Statistical power analysis for the behavioral sciences*. Lawrence Erlbaum Associates, 1988.

[73] B. Efron and R. J. Tibshirani, *An introduction to the bootstrap*. CRC press, 1994.

[74] P. Koehn, "Statistical significance tests for machine translation evaluation," in *Proceedings of the 2004 conference on empirical methods in natural language processing*, 2004, pp. 388–395.

[75] T. Pimentel, J. Valvoda, R. Hall Maudslay, R. Zmigrod, A. Williams, and R. Cotterell, "Information-theoretic probing for linguistic structure," *arXiv preprint arXiv:2004.03061*, 2020.

[76] D. Deutsch, R. Dror, and D. Roth, "A statistical analysis of summarization evaluation metrics using resampling methods," *Transactions of the Association for Computational Linguistics*, vol. 9, pp. 1132–1146, 2021.

[77] J. Pineau, P. Vincent-Lamarre, K. Sinha, V. Larivière, A. Beygelzimer, F. d'Alché Buc, E. Fox, and H. Larochelle, "Improving reproducibility in machine learning research: a report from the neurips 2019 reproducibility program," *The Journal of Machine Learning Research*, vol. 22, no. 1, pp. 7459–7478, 2021.

[78] J. Dodge, S. Gururangan, D. Card, R. Schwartz, and N. A. Smith, "Show your work: Improved reporting of experimental results," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019, pp. 2185–2194.

[79] I. Magar and R. Schwartz, "Data contamination: From memorization to exploitation," *arXiv preprint arXiv:2203.08242*, 2022.

[80] J. Dekoninck, M. Whitehouse, M. Suppanen, S. Cullen, M. Kuchnik, and J. Mitchell, "Evading data contamination detection: Exploring test-time preprocessing methods," *arXiv preprint arXiv:2411.02643*, 2024.

[81] J. Wei, N. Hou, A. Lampinen, X. Chen, D. Huang, Y. Tay, X. Chen, Y. Lu, D. Zhou, T. Ma *et al.*, "Inverse scaling can become u-shaped," *arXiv preprint arXiv:2211.02011*, 2023.

[82] I. R. McKenzie, A. Lyzhov, M. Pieler, A. Parrish, A. Mueller, A. Prabhu, E. McLean, A. Kirtland, A. Ross, A. Liu *et al.*, "Inverse scaling: When bigger isn't better," *arXiv preprint arXiv:2306.09479*, 2023.

[83] D. Ganguli, D. Hernandez, L. Lovitt, N. DasSarma, T. Henighan, A. Jones, N. Joseph, J. Kernion, B. Mann, A. Askell *et al.*, "Predictability and surprise in large generative models," *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, pp. 1747–1764, 2022.

[84] N. Kandpal, H. Deng, A. Roberts, E. Wallace, and C. Raffel, "Large language models struggle to learn long-tail knowledge," *International Conference on Machine Learning*, pp. 15 696–15 707, 2023.

[85] X. Wang, J. Wei, D. Schuurmans, Q. Le, E. Chi, S. Narang, A. Chowdhery, and D. Zhou, "Self-consistency improves chain of thought reasoning in language models," *arXiv preprint arXiv:2203.11171*, 2023.

[86] R. Burnell, W. Schellaert, J. Burden, T. D. Ullman, F. Martinez-Plumed, J. B. Tenenbaum, D. Rutar, L. G. Cheke, J. Sohl-Dickstein, M. Mitchell *et al.*, "Rethink reporting of evaluation results in ai," *Science*, vol. 380, no. 6641, pp. 136–138, 2023.

[87] Y. Razeghi, R. L. Logan IV, M. Gardner, and S. Singh, "Impact of pretraining term frequencies on few-shot numerical reasoning," *arXiv preprint arXiv:2202.07206*, 2022.

[88] F. F. Xu, U. Alon, G. Neubig, and V. J. Hellendoorn, "A systematic evaluation of large language models of code," *Proceedings of the 6th ACM SIGPLAN International Symposium on Machine Programming*, pp. 1–10, 2022.

[89] E. Strubell, A. Ganesh, and A. McCallum, "Energy and policy considerations for deep learning in nlp," *Proceedings of the 57th annual meeting of the association for computational linguistics*, pp. 3645–3650, 2019.

[90] D. Patterson, J. Gonzalez, Q. Le, C. Liang, L.-M. Munguia, D. Rothchild, D. So, M. Texier, and J. Dean, "Carbon emissions and large neural network training," *arXiv preprint arXiv:2104.10350*, 2021.

[91] Y. Tay, M. Dehghani, J. Rao, W. Fedus, S. Abnar, H. W. Chung, S. Narang, D. Yogatama, A. Vaswani, and D. Metzler, "Scale efficiently: Insights from pretraining and finetuning transformers," *arXiv preprint arXiv:2109.10686*, 2022.

[92] C. Bonferroni, "Teoria statistica delle classi e calcolo delle probabilita," *Pubblicazioni del R Istituto Superiore di Scienze Economiche e Commericiali di Firenze*, vol. 8, pp. 3–62, 1936.

[93] S. S. Sawilowsky, "New effect size rules of thumb," *Journal of modern applied statistical methods*, vol. 8, no. 2, p. 26, 2009.

[94] J. R. Landis and G. G. Koch, "The measurement of observer agreement for categorical data," *biometrics*, pp. 159–174, 1977.

[95] D. Ulmer, C. Hardmeier, and J. Frellsen, "Prior and posterior networks: A survey on evidential deep learning methods for uncertainty estimation," *arXiv preprint arXiv:2110.03051*, 2022.