SETA INTERNATIONAL

WHITEPAPER

Performance Comparison of

Open-Source LLMs Across Multiple Domains

Nguyen Viet Anh

Abstract

Large Language Models (LLMs) have demonstrated strong performance across a wide range of natural language processing tasks, but many state-of-the-art models remain computationally expensive and difficult to deploy in resource-constrained environments. In recent years, lightweight open-source and source-available LLMs (1B–3B parameters) have emerged as practical alternatives for real-world applications.

This paper presents a comparative evaluation of five representative lightweight LLMs—Llama-3.2-1B, DeepSeek-R1-Distill-Qwen-1.5B, Gemma-2-2B-IT, Ministral-3-3B-Instruct, and Qwen-3.5-2B—using the MMLU benchmark, which covers 57 subjects across multiple knowledge domains. To improve interpretability, the evaluation results are grouped into four categories: STEM, Humanities, Social Sciences, and Other.

Our findings reveal a clear trade-off between model performance and inference efficiency. While Ministral-3B achieves the highest overall accuracy, it also incurs the highest computational cost. In contrast, Qwen-3.5-2B provides the most balanced performance-efficiency trade-off. Smaller models (≤1.5B parameters) consistently underperform across all domains, particularly in reasoning-intensive tasks.

These results highlight that selecting an appropriate LLM for deployment requires careful consideration of both performance and operational constraints, rather than relying solely on benchmark accuracy.

I. Introduction

Large Language Models (LLMs) have become a central component of modern artificial intelligence systems, enabling significant advances in natural language understanding, reasoning, and code generation. While proprietary models such as GPT-series and Claude have achieved strong performance, their reliance on closed APIs and high computational costs limits their applicability in many real-world scenarios.

At the same time, the open-source and source-available ecosystem of LLMs has rapidly evolved, with organizations such as Meta, Mistral AI, Alibaba, Google, and DeepSeek releasing competitive models with relatively small parameter sizes. These lightweight models, typically ranging from 1B to 3B parameters, are particularly attractive for practical deployment due to their lower hardware requirements, reduced latency, and greater flexibility for customization.

1.1 Motivation

Despite their growing popularity, selecting an appropriate lightweight LLM for deployment remains a challenging task. In real-world systems, model choice is rarely determined by benchmark performance alone. Instead, practitioners must consider a combination of factors, including accuracy, inference time, hardware constraints, licensing conditions, and domain-specific capabilities.

However, existing evaluations often focus either on large-scale models or on aggregate benchmark scores without providing a detailed analysis of performance across different knowledge domains. Furthermore, the trade-off between model quality and inference efficiency is often underexplored, especially for smaller models intended for local or enterprise deployment.

This gap creates practical uncertainty for developers and organizations seeking to integrate LLMs into production systems, particularly when operating under resource constraints.

1.2 Contributions

To address these challenges, this study provides a systematic comparison of several representative lightweight LLMs under a unified evaluation setup. Specifically, this paper makes the following contributions:

1. We benchmark five widely used lightweight open-source or source-available LLMs (1B–3B parameters) using the MMLU benchmark.

2. We group 57 MMLU subjects into four high-level categories—STEM, Humanities, Social Sciences, and Other—to enable more interpretable analysis.
3. We analyze the trade-off between model performance and inference efficiency, highlighting practical considerations for deployment.
4. We provide insights into domain-specific strengths and weaknesses of different models, supporting informed decision-making in real-world applications.

The findings of this study aim to bridge the gap between benchmark evaluation and practical deployment, offering guidance for selecting lightweight LLMs in resource-constrained and enterprise environments.

II. Overview of Selected Models

This study evaluates a set of lightweight open-source and source-available large language models (LLMs) with parameter sizes ranging from 1B to 3B. The selected models represent diverse design choices in terms of architecture, training strategy, and licensing conditions, which are important factors for real-world deployment.

2.1 Model Selection Criteria

The models are selected based on the following criteria:

- Lightweight scale: Models within the 1B–3B parameter range to ensure feasibility for local and resource-constrained environments.
- Public availability: Models must be accessible as open-source or source-available.
- Practical relevance: Models are chosen from widely adopted and actively developed LLM families.

2.2 Llama-3.2-1B

Llama-3.2-1B, developed by Meta, is part of the widely adopted Llama ecosystem and serves as a strong baseline for lightweight LLMs. It benefits from extensive community support and is commonly used in both research and industry applications.

The model is released under the Llama 3.2 Community License, which allows commercial usage, modification, and redistribution. However, it is not fully open-source and includes additional restrictions. For example, derivative models must follow specific naming conventions, products must display attribution (“Built with Llama”), and large-scale companies may require a separate license. These conditions make Llama flexible but more constrained compared to permissive open-source licenses.

2.3 DeepSeek-R1-Distill-Qwen-1.5B

DeepSeek-R1-Distill-Qwen-1.5B is a distilled reasoning-oriented model developed by DeepSeek. It transfers capabilities from larger models into a smaller architecture, making it particularly relevant for efficiency-focused deployments.

The model is released under the MIT License, which is highly permissive and allows unrestricted use, modification, redistribution, and commercial deployment. As the model is derived from the Qwen family, attribution requirements from the original Apache 2.0 license must also be preserved. This combination of permissive licensing and distilled design makes it a flexible option for experimentation and product integration.

2.4 Qwen-3.5-2B

Qwen-3.5-2B, developed by Alibaba, is designed as a general-purpose LLM with strong capabilities in reasoning, coding, and multilingual tasks. It represents a balanced approach between performance and efficiency within the lightweight model range.

The model is released under the Apache License 2.0, which permits modification, redistribution, and commercial use with minimal restrictions. Users are required to preserve attribution, include the license, and document modifications. Due to its permissive licensing and strong performance, Qwen is well-suited for both research and enterprise applications.

2.5 Ministral-3-3B-Instruct

Ministral-3-3B-Instruct, developed by Mistral AI, is designed to deliver strong performance relative to its parameter size. Mistral models are known for their efficiency and competitive benchmark results.

Similar to Qwen, this model is released under the Apache License 2.0, enabling flexible usage in both open and proprietary systems. Its combination of high performance and permissive licensing makes it particularly attractive for production environments.

2.6 Gemma-2-2B-IT

Gemma-2-2B-IT is a lightweight model developed by Google, derived from the research behind the Gemini family. It is designed to provide accessible language modeling capabilities with relatively small parameter sizes.

Gemma supports both research and commercial use, including deployment in local and hosted environments. However, it is subject to additional policy-based restrictions, such as compliance with Google’s Terms of Use and Prohibited Use Policy. Compared to Apache-licensed models, Gemma offers slightly less flexibility but remains a strong candidate for knowledge-intensive applications.

2.7 Summary

The selected models differ not only in architecture and parameter size, but also in licensing constraints and deployment flexibility. In particular, Apache-licensed models (Qwen, Mistral) provide the most permissive usage conditions, while Llama and Gemma introduce additional restrictions. DeepSeek offers a highly flexible alternative through MIT licensing combined with a distilled design.

This diversity enables a comprehensive comparison of both performance and practical deployment considerations in the lightweight LLM setting.

III. Related Work

3.1 Large Language Model Benchmarking

The evaluation of large language models has become an important research area as LLMs continue to improve across a wide range of tasks. One of the most widely used benchmarks is MMLU (Massive Multitask Language

Understanding) (Hendrycks et al., 2021), which assesses general knowledge and reasoning ability across 57 academic subjects. Due to its breadth and diversity, MMLU has become a standard benchmark for comparing both proprietary and open-source LLMs.

In addition to MMLU, several other benchmarks have been proposed to evaluate different aspects of LLM capabilities. For example, TruthfulQA (Lin et al., 2022) focuses on factual reliability and hallucination, while HumanEval (Chen et al., 2021) measures code generation ability using executable programming tasks. More recent benchmarks such as SWE-bench and LiveCodeBench further evaluate real-world software engineering capabilities. These benchmarks highlight that LLM performance is multi-dimensional and cannot be fully captured by a single metric.

3.2 Open-Source and Lightweight LLMs

Recent years have seen rapid progress in open-source and source-available LLMs. Model families such as LLaMA (Touvron et al., 2023), Mistral (Jiang et al., 2023), Qwen (Bai et al., 2023), Gemma (Google, 2024), and DeepSeek (DeepSeek-AI, 2024) have demonstrated that competitive performance can be achieved without relying on proprietary systems.

In particular, there is growing interest in lightweight models with parameter sizes ranging from 1B to 3B. These models are designed to balance performance and efficiency, making them suitable for deployment in resource-constrained environments such as edge devices, on-premise systems, and enterprise applications. Prior work has shown that while smaller models are generally less capable than larger ones, advances in training techniques, instruction tuning, and distillation have significantly improved their practical usefulness.

3.3 Performance-Efficiency Trade-offs

A key challenge in LLM deployment is the trade-off between model performance and computational efficiency. Larger models tend to achieve higher accuracy but require more memory, longer inference time, and higher operational cost. Conversely, smaller models offer faster inference and lower resource consumption but may struggle with complex reasoning tasks.

Several recent studies have explored this trade-off, emphasizing that model selection should be guided by application-specific constraints rather than benchmark scores alone. However, many existing works focus primarily on large-scale models or do not provide detailed analysis across different knowledge domains.

3.4 Positioning of This Work

In contrast to prior work, this study focuses specifically on lightweight LLMs (1B–3B parameters) and evaluates them under a unified benchmarking setup. By grouping MMLU tasks into high-level categories, we provide a more interpretable analysis of domain-specific performance. Furthermore, we explicitly incorporate inference time into the evaluation, enabling a clearer understanding of the trade-off between model accuracy and efficiency in practical deployment scenarios.

IV. Methodology

4.1 Benchmark Selection

The evaluation is conducted using the Massive Multitask Language Understanding (MMLU) benchmark, a widely adopted benchmark for measuring general knowledge and reasoning ability in large language models. MMLU consists of multiple-choice questions spanning 57 academic and professional subjects, including mathematics, physics, computer science, law, history, and medicine.

MMLU is selected in this study because it provides a comprehensive and standardized evaluation of both factual knowledge and reasoning capability across diverse domains, making it particularly suitable for comparing general-purpose language models.

4.2 Evaluation Metric

Model performance is measured using accuracy, defined as the proportion of correctly answered multiple-choice questions. Each question contains four options (A, B, C, D), and a prediction is considered correct only if it exactly matches the ground-truth answer.

To provide a more interpretable analysis, performance is reported at both the category level and the overall level. The overall score is computed as the average accuracy across all categories.

4.3 Category-Based Analysis

To better understand domain-specific behavior, the 57 MMLU subjects are grouped into four high-level categories:

- *STEM:* mathematics, physics, chemistry, computer science, and engineering
- *Humanities:* history, philosophy, law, and related disciplines
- *Social Sciences:* psychology, economics, politics, and geography
- *Other:* medicine, business, and miscellaneous knowledge domains

This grouping enables a more structured comparison by highlighting strengths and weaknesses of each model across different types of knowledge and reasoning tasks.

4.4 Evaluation Framework

The evaluation is implemented using the DeepEval framework, which provides a standardized interface for running benchmarks and collecting performance metrics. A unified evaluation framework ensures that all models are assessed under consistent conditions, improving the fairness and comparability of results.

V. Experimental Setup and Result

5.1 Experimental Setup

5.1.1. Assumptions

To ensure consistency and comparability across all evaluated models, several assumptions are made throughout the experimental design and evaluation process:

1. **Consistent inference conditions**
   All models are evaluated under identical settings, including the same prompting strategy (2-shot), batch size (1), and unified prompt template. This ensures that observed differences primarily reflect model capability rather than variations in evaluation setup.

2. **Deterministic decoding for reproducibility**
   The evaluation adopts deterministic decoding (temperature = 0, do_sample = False), ensuring that models always produce the most probable output. This removes stochastic variation and guarantees reproducibility across runs. However, this assumption may not fully reflect real-world usage, where sampling-based decoding can improve diversity and reasoning performance.

3. **MMLU as a proxy for general capability**
   The benchmark assumes that MMLU provides a representative measure of general knowledge and reasoning ability. While comprehensive, it does not fully capture other real-world scenarios such as open-ended generation or interactive tasks.

4. **Accuracy as the primary performance metric**
   Model performance is evaluated using exact-match accuracy on multiple-choice questions. This assumes that accuracy is a sufficient proxy for capability, although it does not account for explanation quality, robustness, or generative fluency.

5. **Throughput as the efficiency metric**
   Computational efficiency is measured using throughput (items per second), which reflects processing capacity under fixed conditions. This assumption does not capture other aspects such as per-request latency or energy consumption.

6. **Single-hardware evaluation environment**
   All models are evaluated on the same hardware setup (NVIDIA Tesla P100), assuming that relative comparisons remain valid. However, absolute performance and efficiency may differ under other hardware configurations.

7. **Prompt and answer extraction neutrality**
   A standardized prompt format and answer extraction strategy are applied across all models. It is assumed that this does not introduce systematic bias. Invalid outputs are handled using a predefined fallback mechanism to ensure robustness.

5.1.2 Evaluation Framework and Model Integration

The evaluation is conducted using the DeepEval framework, which provides a standardized interface for benchmarking large language models on the MMLU dataset. To ensure compatibility between HuggingFace models and the evaluation pipeline, each model is encapsulated within a custom adapter that inherits from a unified base interface.

Specifically, the adapter is responsible for handling prompt construction, text generation, and answer extraction. For example, in the case of Llama-based models, the adapter utilizes the HuggingFace AutoTokenizer and AutoModelForCausalLM APIs for inference. The model is configured to run on GPU if available, otherwise falling back to CPU execution. Additionally, half-precision (float16) is enabled to reduce memory consumption, and caching is disabled to ensure stable generation behavior across sequential evaluations.

The generation process follows a deterministic setup (do_sample=False) to guarantee reproducibility. Each input prompt is augmented with an explicit instruction requiring the model to return only a single answer choice (A, B, C, or D). The generated output is then post-processed using a regular expression to extract the final answer.

5.1.3 Benchmark Configuration

The evaluation is conducted using the Massive Multitask Language Understanding (MMLU) benchmark under a multiple-choice question answering setting. Each question consists of four candidate answers (A, B, C, D), and the model is required to select the correct option.

The benchmark is configured using the DeepEval framework with the following setup:

- **Prompting strategy:** Few-shot prompting (2-shot)
- **Batch size:** 1
- **Decoding strategy:** Greedy decoding (do_sample = False)
- **Maximum generation length:** 5 tokens

To ensure consistent and structured outputs across models, a standardized prompt template is applied. For each question, the original MMLU prompt (including few-shot examples handled internally by DeepEval) is augmented with an explicit instruction that constrains the output format:

"""

Answer the following multiple choice question.

Respond with ONLY one letter: A, B, C, or D.

{question}

Answer:

"""

This prompt design enforces a strict output space, ensuring that models produce a single-character response corresponding to one of the valid answer choices. Such constraint is particularly important for lightweight models, which may otherwise generate verbose or unstructured outputs.

During inference, the model generates a short continuation limited to 5 tokens. The final prediction is extracted by applying a regular expression that identifies the first valid occurrence of {A, B, C, D} in the generated text. If no valid answer is detected, a default fallback option ("A") is assigned to maintain evaluation robustness and avoid missing predictions.

The use of greedy decoding combined with a constrained output format ensures deterministic behavior across runs, improving reproducibility and fairness in model comparison.

5.1.4 Task Grouping and Category Design

Instead of evaluating all MMLU subjects as a single aggregate benchmark, the tasks are explicitly grouped into four high-level categories: **STEM, Humanities, Social Sciences, and Other**.

This grouping is implemented programmatically via a predefined mapping from subject names to categories. Each subject is mapped to a corresponding MMLUTask object, and tasks are aggregated into category-specific lists before evaluation.

This design enables category-level benchmarking, allowing the analysis to capture domain-specific strengths and weaknesses rather than relying solely on a single overall score.

5.1.5 Execution Procedure

The evaluation is executed sequentially for each model to ensure controlled resource usage and fair comparison. The procedure follows these steps:

1. A model adapter is instantiated via a model factory abstraction.
2. GPU peak memory statistics are reset prior to evaluation.
3. For each category:
    - A separate MMLU benchmark instance is created with the corresponding subset of tasks.
    - The model is evaluated using batch_size = 1.
    - Execution time is recorded.
    - Resource usage metrics (GPU memory, peak GPU memory, and RAM usage) are logged.
4. After all categories are evaluated:
    - Total inference time and peak resource usage are computed.
    - Category-level scores are stored for further aggregation.

To prevent memory accumulation across runs, the model is explicitly deleted after evaluation, followed by garbage collection and CUDA cache clearing.

5.1.6 Resource and Efficiency Tracking

In addition to accuracy, the experimental setup explicitly tracks system-level resource usage to analyze efficiency trade-offs between models. The following metrics are recorded:

- **Inference time per category and total runtime**
- **GPU memory usage (allocated and peak)**
- **RAM usage**

These measurements are collected during runtime using psutil and PyTorch CUDA utilities and are stored in a structured JSON file for post-hoc analysis.

This design enables a more comprehensive evaluation that considers not only model performance but also deployment cost and hardware efficiency.

5.1.7 Evaluation Metric

The primary evaluation metric is **accuracy**, defined as the proportion of correctly answered questions. For each category, accuracy is computed independently, and the overall score is obtained by averaging across all categories.

Unlike generative evaluation settings, correctness is determined strictly by exact match between the extracted answer (A/B/C/D) and the ground-truth label. This strict evaluation protocol ensures consistency and comparability across all models.

5.2 Results

5.2.1 Evaluation Output

The benchmark produces one score for each category and one overall score for each model. Category-level results provide a clearer picture of the relative strengths of each model across different knowledge areas, while the overall score summarizes general benchmark performance.

The final result format is as follows:

- **STEM score**
- **Humanities score**
- **Social Sciences score**
- **Other score**
- **Overall average score**

This structure is useful because lightweight models often do not perform uniformly across all domains. Some models may perform better in quantitative reasoning tasks, while others may be more stable in general knowledge or language-heavy subjects.

5.2.2 Result Table

Models	STEM	Humanities	Social Science	Other	Overall	Speed (item/s)	GPU Memory (MB)	GPU Peak (MB)	RAM (MB)
meta-llama/Llama-3.2-1B	0.2905	0.3015	0.3233	0.3170	0.3081	**5.2**	2366	2392	1778
deepseek-ai/DeepSeek-R1-Distill-Qwen-1.5B	0.3767	0.3192	0.3750	0.3710	0.3605	4.0	3398	3415	3913
google/gemma-2-2b-it	0.4811	0.4964	0.6506	0.6175	0.5614	2.7	4995	5029	4009
mistralai/Ministral-3-3B-Instruct-2512	**0.6295**	**0.5579**	**0.7871**	**0.7202**	**0.6737**	2.0	7372	7424	3339
Qwen/Qwen3.5-2B	0.5265	0.4726	0.6246	0.6295	0.5633	2.3	3598	3645	3372

Table 1. Category-wise MMLU Performance and Inference Time of Evaluated Models

5.2.3 Discussions

The experimental results reveal several important patterns regarding the capabilities and limitations of lightweight large language models.

First, there exists a substantial performance gap among the evaluated models, despite their relatively similar parameter scales (1B–3B). Ministral-3-3B-Instruct achieves the highest overall accuracy (0.6666), significantly outperforming smaller models such as Llama-3.2-1B (0.3003) and DeepSeek-R1-Distill-Qwen-1.5B (0.3411). This suggests that even within the "lightweight" regime, architectural design and training strategy play a critical role beyond parameter count alone. In particular, the strong performance of Ministral indicates that efficiency-oriented architectures can still maintain competitive reasoning and knowledge capabilities when properly optimized.

Second, the results highlight a clear trade-off between performance and inference efficiency. Models with higher accuracy consistently require longer inference time, with Ministral exhibiting both the best performance and the highest latency (12,048 seconds), while Llama-3.2-1B shows the opposite trend. This trade-off reflects an inherent constraint in lightweight model deployment: improving reasoning capability often comes at the cost of increased computational complexity. However, the relationship is not strictly linear. For example, Qwen3.5-2B achieves competitive performance (0.5948) with significantly lower inference time than Ministral, suggesting that certain model families are more efficient in utilizing their parameters. This makes Qwen a particularly strong candidate for real-world applications where both latency and accuracy are critical.

Third, domain-specific performance variations reveal that lightweight models are not uniformly capable across knowledge areas. Stronger models such as Ministral, Qwen, and Gemma consistently achieve higher scores in Social Science and Other categories compared to STEM. This pattern suggests that these models are more effective in tasks requiring contextual understanding, language comprehension, and factual recall, rather than strict analytical or mathematical reasoning. In contrast, lower performance in STEM domains indicates that lightweight models still struggle with structured reasoning tasks, which often require multi-step logical inference and precise symbolic manipulation.

Fourth, the relatively modest performance of DeepSeek-R1-Distill-Qwen-1.5B provides insight into the limitations of distillation for small-scale models. While distillation is intended to transfer reasoning capabilities from larger models, the results suggest that this process may not fully preserve general knowledge performance, particularly in diverse benchmark settings such as MMLU. This indicates that distillation may be more effective for specialized reasoning tasks than for broad-domain knowledge evaluation.

Finally, from a practical deployment perspective, the results emphasize that the "best" model depends strongly on application requirements rather than benchmark score alone. Ministral is most suitable for scenarios where accuracy is the primary concern, such as analytical assistants or high-quality enterprise systems. In contrast, Qwen offers a more favorable balance between performance and efficiency, making it well-suited for general-purpose deployment. Gemma provides stable performance in knowledge-intensive domains, while smaller models such as Llama and DeepSeek remain relevant in highly resource-constrained environments.

Overall, these findings suggest that lightweight LLMs have reached a level of maturity where meaningful trade-offs between performance, efficiency, and domain capability can be systematically evaluated. However, they also highlight that significant gaps remain, particularly in reasoning-intensive domains, indicating an important direction for future model optimization and research.

VI. Conclusion

This report presented a comparative evaluation of five lightweight open-source or source-available large language models: Llama-3.2-1B, DeepSeek-R1-Distill-Qwen-1.5B, Gemma-2-2b-it, Ministral-3-3B-Instruct-2512, and Qwen3.5-2B. The evaluation was conducted using the MMLU benchmark with tasks grouped into four categories: STEM, Humanities, Social Science, and Other. This category-based setup provided a more interpretable view of model behavior across different knowledge domains.

The experimental results showed that Ministral-3-3B-Instruct-2512 achieved the strongest overall benchmark performance, while Qwen3.5-2B offered the most attractive balance between accuracy and inference cost. Gemma-2-2b-it also demonstrated competitive and stable performance, particularly in knowledge-intensive categories. In contrast, DeepSeek-R1-Distill-Qwen-1.5B and Llama-3.2-1B were more limited in benchmark accuracy, but remained relevant for lightweight or resource-constrained deployment scenarios. These findings confirm that even among relatively small models, there are meaningful differences in both capability and efficiency.

A key conclusion of this study is that there is no single best model for every practical scenario. Instead, model selection should depend on the intended application, available hardware resources, acceptable inference latency, and required output quality. Larger lightweight models tend to perform better, but they also require significantly more computation time. Smaller models are faster and easier to deploy, but their performance may be insufficient for tasks requiring stronger reasoning or broader academic knowledge.

Overall, this benchmark demonstrates that lightweight LLMs can still provide useful performance for real-world applications, especially when deployment constraints make larger models impractical. The results also show the importance of evaluating models not only by overall accuracy, but also by domain-specific performance and operational efficiency.

For future work, the evaluation can be extended by incorporating additional benchmarks such as TruthfulQA for factual reliability and HumanEval for code generation. This would provide a broader understanding of model behavior beyond multiple-choice reasoning and support more informed model selection for enterprise use cases.

References

[1] Touvron, Hugo and Lavril, Thibaut and Izacard, Gautier and Martinet, Xavier and Lachaux, Marie-Anne and Lacroix, Timothe and others, "LLaMA: Open and Efficient Foundation Language Models," *arXiv,* 2023.

[2] DeepSeek-AI, "DeepSeek LLM: Scaling Open-Source Language Models with Long Context and Reasoning Capabilities," *arXiv preprint arXiv:2401.02954,* 2024.

[3] Bai, Jinze and Bai, Shiyue and Yang, Sheng and Wang, Shuai and Tan, Shuo and Wang, Kai and others, "Qwen Technical Report," *arXiv preprint arXiv:2309.16609,* 2023.

[4] Jiang, Albert Q. and Sablayrolles, Alexandre and Roux, Arthur and Mensch, Arthur and Bamford, Chris and Chaplot, Devendra Singh and others, "Mistral 7B," *arXiv preprint arXiv:2310.06825,* 2023.

[5] G. DeepMind, "Gemma: Open Models Based on Gemini Research," Google, 2024. [Online]. Available: https://ai.google.dev/gemma.

[6] Hendrycks, Dan and Burns, Collin and Basart, Steven and Zou, Andy and Mazeika, Mantas and Song, Dawn and Steinhardt, Jacob, "Measuring Massive Multitask Language Understanding," *ICLR,* 2021.

[7] Lin, Stephanie and Hilton, Jacob and Evans, Owain, "TruthfulQA: Measuring How Models Mimic Human Falsehoods," 2022.

[8] Chen, Mark and Tworek, Jerry and Jun, Heewoo and Yuan, Qiming and Pinto, Henrique P. and Kaplan, Jared and others, "Evaluating Large Language Models Trained on Code," *arXiv,* 2021.