CodeMMLU: A Multi-Task Benchmark for Assessing Code Understanding Capabilities of CodeLLMs

Dung Nguyen Manh[*], **Thang Phan Chau**[*], **Nam Le Hai**[*†], **Thong T. Doan**[*], **Nam V. Nguyen**[*], **Quang Pham**[*◇], **Nghi D. Q. Bui**[*]

[*]FPT Software AI Center, Viet Nam
[†]Hanoi University of Science and Technology
[◇]VNU-HCM - University of Science
dungnm31@fpt.com, thangpc13@fpt.com, namlh35@fpt.com, thongdt4@fpt.com, namnv78@fpt.com, quangp2808@gmail.com, bdqnghi@gmail.com

Recent advancements in Code Large Language Models (CodeLLMs) have predominantly focused on open-ended code generation tasks, often neglecting the critical aspect of code understanding and comprehension. To bridge this gap, we present CodeMMLU, a comprehensive multiple-choice question-answer benchmark designed to evaluate the depth of software and code understanding in LLMs. CodeMMLU includes over 10,000 questions sourced from diverse domains, encompassing tasks such as code analysis, defect detection, and software engineering principles across multiple programming languages. Unlike traditional benchmarks, CodeMMLU assesses models' ability to reason about code rather than merely generate it, providing deeper insights into their grasp of complex software concepts and systems. Our extensive evaluation reveals that even state-of-the-art models face significant challenges with CodeMMLU, highlighting deficiencies in comprehension beyond code generation. By underscoring the crucial relationship between code understanding and effective generation, CodeMMLU serves as a vital resource for advancing AI-assisted software development, ultimately aiming to create more reliable and capable coding assistants.

GitHub: https://github.com/FSoft-AI4Code/CodeMMLU

1. Introduction

Recent advancements in Code Large Language Models (CodeLLMs) have demonstrated impressive capabilities across various software engineering (SE) tasks (Allal et al., 2023; Bui et al., 2023; Feng et al., 2020; Guo et al., 2024; Li et al., 2023; Lozhkov et al., 2024b; Luo et al., 2023; Nijkamp et al., 2022; Pinnaparaju et al., 2024; Roziere et al., 2023; To et al., 2023; Wang et al., 2021, 2023b; Xu et al., 2022; Zheng et al., 2024c). However, existing benchmarks often fall short in providing rigorous evaluations due to outdated methodologies and potential data leakage (Matton et al., 2024). Moreover, practical applications of CodeLLMs reveal limitations such as bias and hallucination (Liu et al., 2024a; Rahman & Kundu, 2024) that current benchmarks fail to adequately address.

The predominant focus of coding-related benchmarks has been on open-ended, free-form generation tasks, such as code generation/code completion (Austin et al., 2021; Chen et al., 2021; Ding et al., 2023; Hendrycks et al., 2021; Iyer et al., 2018; Lai et al., 2023; Lu et al., 2021; Zhuo et al., 2024) and other SE tasks like program repair Ouyang et al. (2024); Xia et al. (2023) (Table 1). While appealing, these benchmarks struggle to discern whether CodeLLMs truly understand code or merely reproduce memorized training data (Carlini et al., 2022; Nasr et al., 2023). Additionally, the reliance on test cases and executability for evaluation limits the quantity and diversity of these benchmarks across domains, potentially leading to biased and limited generalizations. Recent efforts to improve evaluation through free-form question answering (Li et al., 2024; Liu & Wan, 2021) have introduced

Corresponding author(s): Nghi D. Q. Bui (bdqnghi@gmail.com)

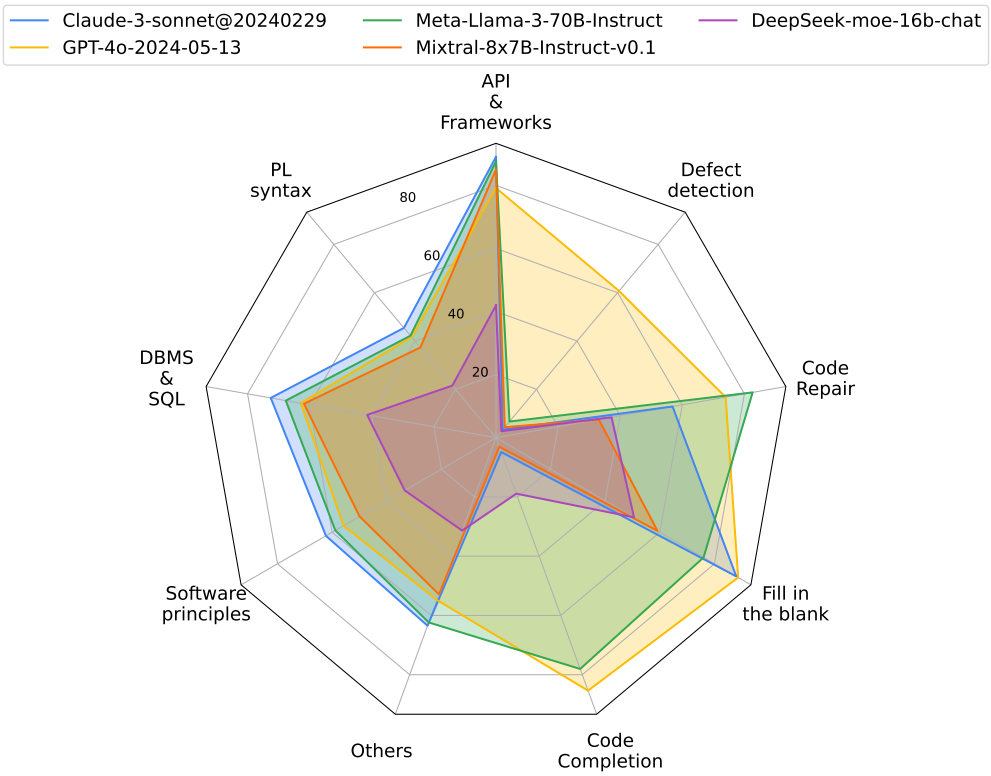

Figure 1 | **Summary performance of LLMs on the CodeMMLU benchmark.** This radar chart presents the evaluation results (accuracy %) of different models across various CodeMMLU tasks.

new challenges, often requiring less rigorous metrics or LLMs-as-a-judge approaches (Zheng et al., 2023). However, LLMs-as-a-judge methods are susceptible to adversarial attacks (Raina et al., 2024), raising concerns about the reliability of such evaluation pipelines for coding tasks.

To address these shortcomings, we introduce CodeMMLU, a novel benchmark designed to evaluate CodeLLMs' ability to understand and comprehend code through multi-choice question answering (MCQA). This approach enables a deeper assessment of how CodeLLMs grasp coding concepts, moving beyond mere generation capabilities. Inspired by the MMLU dataset (Hendrycks et al., 2020) from natural language understanding, CodeMMLU offers a robust and easily evaluable methodology with the following key features:

- **Comprehensiveness:** CodeMMLU comprises over 10,000 questions curated from diverse, high-quality sources, mitigating potential bias from limited evaluation data.
- **Diversity in task, domain, and language:** The dataset covers a wide spectrum of software knowledge, including general QA, code generation, defect detection, and code repair across various domains and more than 10 programming languages.

CodeMMLU enables us to assess LLMs' capabilities in coding and software tasks from a novel perspective, extending beyond traditional code generation and completion. Our analysis reveals several notable findings: (1) previously unexplored bias issues in CodeLLMs, aligning with those observed in natural language MCQA tasks; (2) GPT-4 consistently achieving the highest average performance among closed-source models, while (3) the Meta-Llama family demonstrated the greatest accuracy among open-source models; (4) scaling laws related to model size were partially observed within the same model family but not across different families, suggesting the significant influence of pretraining