Instruction Fine-Tuning Effects on Language Model Mathematical Reasoning Accuracy

Assignee Research

June 6, 2026

Abstract

This report synthesises findings from 12 peer-reviewed papers addressing the following research question: What is the effect of instruction fine-tuning on language model mathematical problem-solving accuracy v11. 19 claims were extracted from source literature; 1 was independently verified against retrieved documents. An automated multi-reviewer quality assessment produced a score of 4.2/10. This report is a machine-generated literature synthesis and does not constitute original research.

1 Introduction

This paper examines: Demystifying Instruction Mixing for Fine-tuning Large Language Models. Research question: What is the effect of instruction fine-tuning on language model mathematical problem-solving accuracy v11.

2 Methodology

Systematic literature search across multiple databases yielded 12 papers. Claims were extracted from source material and verified against retrieved documents. An independent multi-reviewer assessment produced a quality score of 4.2/10.

3 Results

12 papers retrieved. 19 claims extracted; 1 independently verified. Quality review score: 4.2/10.

4 Limitations

This report is a machine-generated literature synthesis and does not constitute original research. Automated retrieval and verification may introduce

errors or omissions. Review scores reflect automated assessment, not human peer review. Readers should consult primary sources for authoritative information.

5 Extracted Claims

Claim	Verified	Confidence
The study categorizes instructions into three primary types: NLP downstream tasks, coding, and general chat.	✓	0.39
The evaluation uses ARC, Winogrande, PIQA, MMLU, RACE, and HellaSwag for NLP benchmarks.	×	0.01
The evaluation uses HumanEval to test the pass rate of generated code.	×	0.05
The evaluation uses FLASK for alignment (chat ability) evaluation.	×	0.03
The Alpaca, CodeAlpaca, and P3 datasets are denoted as A, C, and P, respectively.	×	0.02
Eight data mixing strategies were compared: None, A, C, P, AC, AP, CP, and ACP.	×	0.03
In the no-mixture setting, models fine-tuned on P3 achieve the highest average score for NLP tasks.	×	0.07
In the no-mixture setting, models fine-tuned on CodeAlpaca excel in code generation benchmarks.	×	0.04
Alpaca fine-tuned models excel in RACE and HellaSwag tasks.	×	0.04
P3 fine-tuned models perform well on ARC and Winogrande tasks.	×	0.04
For the 7B model, the AC mixing strategy improves code benchmark performance by +1.28 compared to the C strategy.	×	0.04
For the 7B model, the AC mixing strategy improves code benchmark performance by +0.61 compared to the C strategy.	×	0.04
The Alpaca dataset contains 52K instruction–response pairs.	×	0.05
The P3 dataset was processed by randomly sampling 1K instances from each subtask, resulting in 660K samples.	×	0.03
The CodeAlpaca dataset contains 20K samples in different programming languages.	×	0.02
A 20K subset was randomly sampled from each dataset to ensure a balanced comparison.	×	0.02
The alignment evaluation selected the eight most frequent alignment skills, resulting in 1,180 samples.	×	0.04
GPT-4 was employed to assess model responses based on human-written principles.	×	0.05
LLaMA-2 7B and 13B models were fine-tuned for two epochs.	×	0.05

References

- http://arxiv.org/abs/2312.10793v3
- http://arxiv.org/abs/2310.04793v2
- http://arxiv.org/abs/2606.05868v1