Floating-Point Transformer

FUNDED BY ANR

Detection of Numerical Bugs using Large Language Models (LLMs)

Ahmad Saeed & Lisa Taldir

Supervised by: David Defour, Eric Petit and Pablo de Oliveira

ESPACE-DEV
Université de Perpignan via Domitia

November 3rd, 2025

Overview

What is a Large Language Model (LLM)?

Definition

AI system trained to understand, generate and interact in human language.

Examples

- ChatGPT
- Gemini
- DeepSeek

Gemini deepseek ChatGPT

Inside an LLM

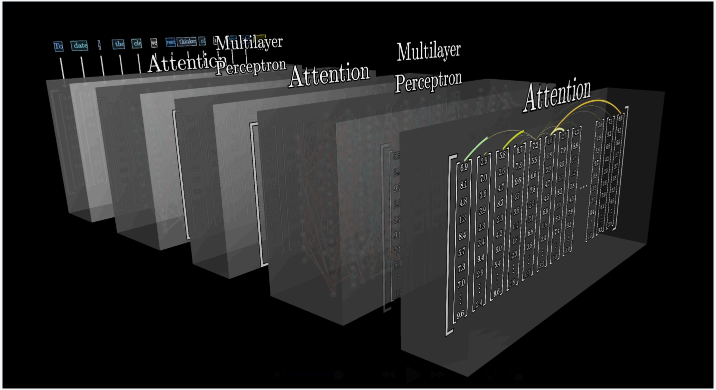


Figure: Source: *Transformers, the tech behind LLMs*, 3Blue1Brown

LLMs and their significance in code analysis

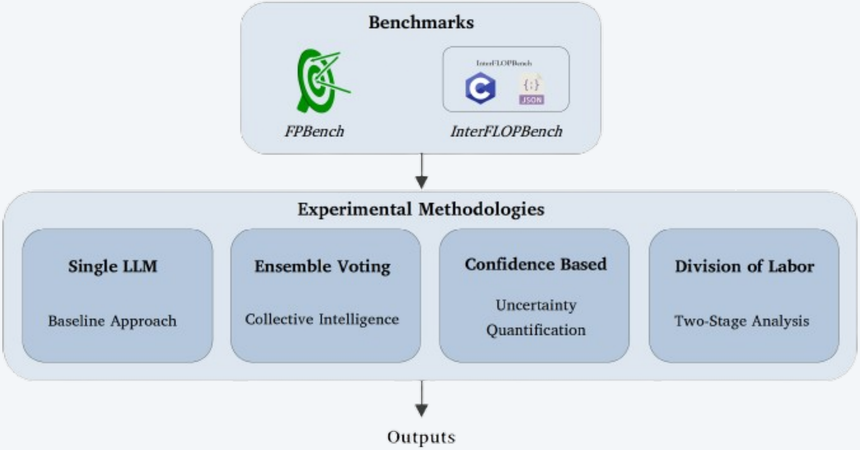
- **Static vs. Dynamic Analysis**: Static analysis examines code without execution, while dynamic analysis requires runtime testing
- **Tool Limitations**: Traditional static analysis struggles with input-dependent and complex numerical error patterns
- **Programming Proficiency**: LLMs show promise in coding and static analysis but their numerical error detection capabilities remain unexplored
- **Interpretation**: Returns outputs interpretable by humans

Research question

Can LLMs detect and classify floating-point errors in code?

$\Rightarrow$ Can it detect floating-point errors in small functions?

Overall Testing Infrastructure



FPBench : A benchmark for validation of numerical accuracy in floating-point computations

- **Benchmark**: Include benchmarks sourced from recent papers on automatic floating-point verification and accuracy improvement
- **Format**: Automatic translation from FPCore to C

InterFLOPBench : A benchmark organized into different floating-point error categories

Format Structure

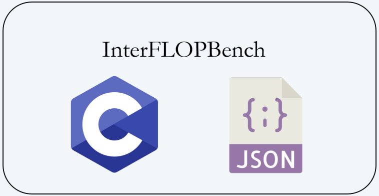


Figure: C source code with accompanying JSON metadata files

Error Categories

Error Category	Benchmark Files	Individual Samples	Total Occurrences
No_error	15	306	480
Comparison	16	247	246
Cancellation	17	181	208
Overflow	13	130	151
Underflow	10	117	83
Div_zero	10	79	70
Nan	9	70	50
Total	90	1130	1288

Experimental Methodology

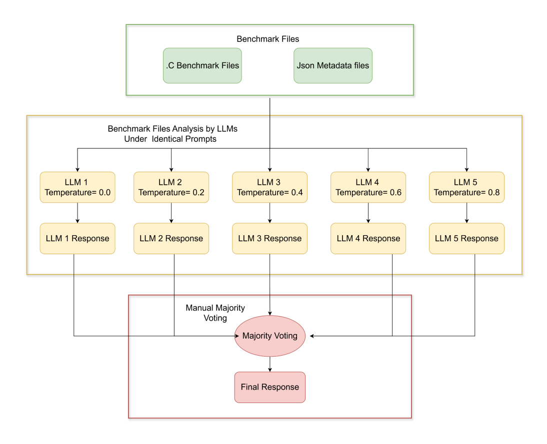
LLMs used

LLM	Developer	Release
Gemini-2.5-flash	Google	Jun 2025
Gemma-3-27B	Google	Mar 2025
Gemini-2.0-flash	Google	Feb 2025
DeepSeek-r1	DeepSeek	Jan 2025
GPT-4o	OpenAI	May 2024

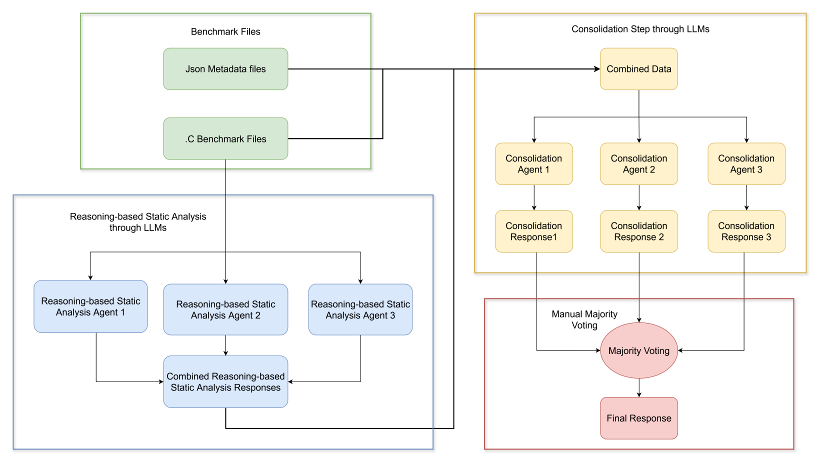
Agent Pipelines

- **Single LLM**: All five LLMs were considered separately
- **Ensemble Voting, Confidence Based, and Division of Labor**: Based on Gemini-2.0-flash

Ensemble Voting Pipeline



Division of Labor Pipeline



Evaluation Metrics and Validation

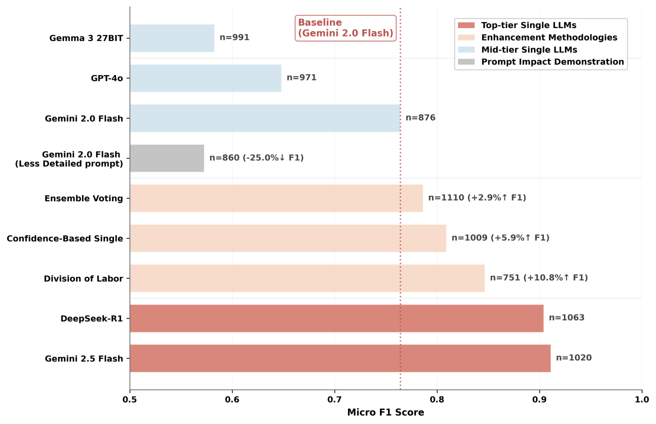
Validation with FPChecker

Figure: FPChecker (Floating-Point Checker) is a dynamic analysis tool to detect floating-point errors in HPC applications

Evaluation Metrics

- **Micro F1**: Aggregates all true/false positives globally before computing F1
- **Macro F1**: Computes F1 per error type independently, then averages
- **Weighted F1**: Weights F1 scores by label frequency in dataset
- **Sample-wise F1**: Averages F1 scores computed per test sample

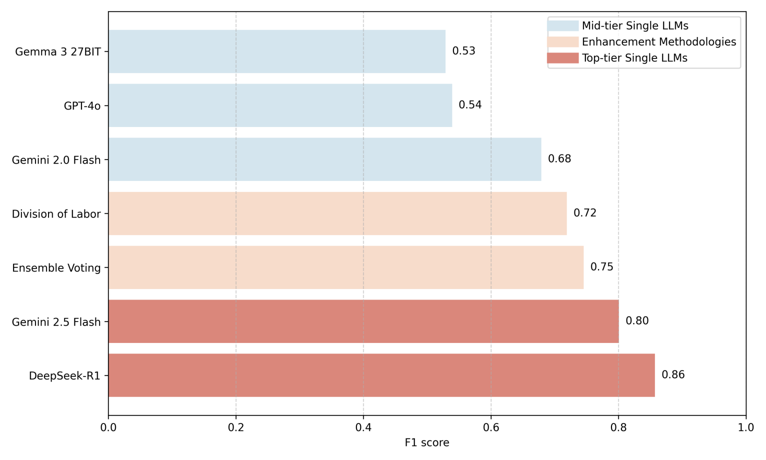
InterFLOPBench – Overall Performance



InterFLOPBench - Per-Category Ranking

Error Category	Avg F1 Score	Difficulty Rank	Best Method F1
Comparison	0.9274	1 (Easiest)	1.0000 (Division of Labor)
Division by Zero	0.9265	2	1.0000 (Division of Labor)
Overflow	0.818	3	0.9524 (Gemini 2.5 Flash)
No Error	0.774	4	0.9402 (Gemini 2.5 Flash)
NaN	0.7122	5	0.9901 (Gemini 2.5 Flash)
Cancellation	0.6521	6	0.8564 (Gemini 2.5 Flash)
Underflow	0.4729	7 (Hardest)	0.9940 (DeepSeek-R1)

FPBench - Overall Performance



1. LLMs Overview
2. Testing Infrastructure
3. Performance Summary
4. **Conclusion and Future Research Directions**

Conclusion - LLM already have some basic numerical knowledge

- **Model selection dominates performance** (7.6% gap between tiers)
- **Sophisticated methodologies enhance mid-tier models**
- **Error-specific capabilities**: Explicit operations vs. subtle numerical phenomena
- **Prompt engineering is crucial**: Simplified prompts cause substantial degradation

Future Research Directions

- **How to improve their capability ?** (more sophisticated reasoning scheme, coupling with tools ?)
- **How to make them cheaper to use?** (get similar quality with small LLM + finetuning)
- **How to get a much larger set of FP benchmarks?** (synthetic data generation thanks to Floq/HOL)

Floating-Point Transformer

FUNDED BY ANR

Thank you for your attention

Ahmad Saeed & Lisa Taldir

Supervised by: David Defour, Eric Petit and Pablo de Oliveira

ESPACE-DEV
Université de Perpignan via Domitia

November 3rd, 2025

References

Nasrine Damouche et al. (2017)
Toward a Standard Benchmark Format and Suite for Floating-Point Analysis
Numerical Software Verification, 63–77.

Laguna, Ignacio (2019)
FPChecker: Detecting Floating-point Exceptions in GPU Applications
ACM International Conference on Automated Software Engineering (ASE), 1126–1129.