SOVEREIGN: How does the Tree of Reviews framework scale with context length on the MuSiQue benchmark when using Llama-3 w

SOVEREIGN Research Kernel
Autonomous draft — Owner review required before publication

May 28, 2026

Abstract

This paper introduces long-context Granite code models that support effective context windows of up to 128K tokens. Our solution for scaling context length of Granite 3B/8B code models from 2K/4K to 128K consists of a light-weight continual pretraining by gradually increasing its RoPE base frequency with repository-level file packing and length-upsampled long-context data. Additionally, we also release instruction-tuned models with long-context support which are derived by further finetuning the long context base models on a mix of permissively licensed short and long-context instruction-respo

1 Introduction

Analysis of: Scaling Granite Code Models to 128K Context. Research goal: How does the Tree of Reviews framework scale with context length on the MuSiQue benchmark when using Llama-3 with 128K context window, and what is the token efficiency trade-off compared to chain-based retrieval methods?.

2 Methodology

Multi-query arXiv search (4 parallel queries, Relevance-sorted). TF-IDF cosine semantic verification (bigrams, threshold=0.15). NIM nv-embedqa-e5-v5 (dim=1024) for semantic indexing. Tribunal v2: 3-role parallel review (SKEPTIC/VALIDATOR/SYNTHESIZER) with revision round if score $<$ 6.5.

3 Results

10 papers retrieved. 11 claims extracted, 1 verified. Tribunal: 1.8/10 → REJECT (revision_round=0). Policy: ESCALATE_TO_OWNER.

4 Uncertainties

NIM free tier latency varies. TF-IDF verification is a weak signal. arXiv Relevance ranking is query-dependent. Tribunal consensus is LLM-based and prompt-sensitive.

5 Extracted Claims

Claim	Verified	Confidence
Long-context Granite code models consistently outperform original base models at different input context from 4K to 32K	✓	0.16
The average Exact Match (EM) performance of long-context Granite code models on the LCC benchmark ranges from 61.6 at 4K	×	0.10
The absolute gap in average EM performance between long-context Granite code models and original base models on LCC rang	×	0.13
On the LCC benchmark, the average EM performance of long-context Granite code models for Python is 76.0 at 4K context an	×	0.11
On the LCC benchmark, the average EM performance of long-context Granite code models for C++ is 58.0 at 4K context and 1	×	0.12
On the LCC benchmark, the average EM performance of long-context Granite code models for Java is 59.0 at 4K context and	×	0.11
On the LCC benchmark, the average EM performance of long-context Granite code models for TypeScript is 58.0 at 4K contex	×	0.12
On the LCC benchmark, the average EM performance of long-context Granite code models for Rust is 57.0 at 4K context and	×	0.11
The average EM performance of long-context Granite code models on the LCC benchmark at 8K context is 33.8.	×	0.12
The average EM performance of long-context Granite code models on the LCC benchmark at 16K context is 26.2.	×	0.12
The average EM performance of long-context Granite code models on the LCC benchmark at 24K context is 20.8.	×	0.12

References

- http://arxiv.org/abs/2404.14464v1
- http://arxiv.org/abs/2411.00744v2
- http://arxiv.org/abs/2407.13739v1