Open in app ↗

Medium Search Write

Francisco Revelles 3 min read · Jun 20, 2025

Copy link
Share on X
Share on Facebook
Share on LinkedIn

What Finite Fields of Size p Can Teach Us About Building Sma Language Models

By Francisco Revelles

Large Language Models (LLMs) have exploded into public awareness as tools of staggering generative power. They write poetry, summarize legal cases, translate medical jargon, and even write code. But for all their versatility, we still understand surprisingly little about how they internally represent structure, reason about numbers, or scale their capabilities. What if part of the answer comes from mathematics — not the kind used to build the model, but the kind it seems to understand?

This article explores how the behavior of LLMs might be illuminated by algebraic structures — specifically, finite fields of size p (also denoted as F_p or GF(p), where p is prime). These structures offer a clean, modular world where numbers wrap around after reaching p, making them ideal test beds for understanding generalization, abstraction, and modular reasoning in neural networks.

Modular Arithmetic: A Natural Language for LLMs?

Finite fields of size p are sets with exactly p elements (0, 1, 2, …, p−1) in which addition, subtraction, multiplication, and division (except by zero) are defined modulo p. These fields are fundamental in cryptography, error correction, and coding theory — but also make an ideal sandbox for probing how LLMs handle symbolic reasoning.

For example, a model can be asked to perform tasks such as:

- $3 + 5 \bmod 7 \rightarrow 1$
- Find x such that $4x \equiv 1 \bmod 7$

When tested across various p, LLMs often generalize surprisingly well, even to unfamiliar values of p or new algebraic identities — suggesting they may learn abstract structures rather than rote patterns.

Representation and Tokenization

To understand what LLMs are doing, we must also consider how they see the problem. Modular arithmetic often lacks natural tokenization in standard models. For instance, “mod” or congruences like “≡” might be tokenized into multiple parts or confounded with natural language. Encoding arithmetic in clearer mathematical notation or using structured prompts can improve performance, revealing the model’s latent competence.

Helix Representations and Modular Cycles

LLMs operate over embeddings in high-dimensional vector spaces. Some recent work proposes that they might internally learn “helix-like” or circular topologies to represent periodic structures — perfect for capturing arithmetic mod p. This maps surprisingly well to the cyclic nature of F_p, where counting wraps around.

Understanding these internal geometries may help us interpret LLM memory, discover emergent neurons, or even design specialized pretraining tasks that target algebraic structure.

Cryptographic Reasoning and Finite Fields

In cryptography, operations over F_p underpin algorithms like RSA, Diffie-Hellman, and elliptic curves. LLMs trained on open-source cryptographic code can develop partial competence in these areas — solving discrete log problems, modular inverses, or simulating key generation — even without formal symbolic solvers. Their ability to learn such structured computation from raw text alone hints at powerful inductive biases.

Applications and Open Questions

By examining LLMs over finite fields of size p, we can investigate:

- Abstraction: Do LLMs learn general rules like distributivity or Fermat’s little theorem?
- Limits of generalization: Can they transfer from small primes to large ones?
- Interpretable mechanisms: Which neurons encode field properties?
- Architectural tuning: Could models be trained to internalize algebra more efficiently?

This isn’t just about math — it’s about building smarter systems. Finite fields give us a precise, interpretable domain in which to reverse-engineer and potentially improve large models. And perhaps most exciting: they might help bridge symbolic and neural learning in a mathematically principled way.

—

Francisco Revelles is a computer scientist exploring intersections of algebra, language, and cognition. He writes about the inner lives of algorithms and what they reveal about us.

This article is part of an ongoing series on language models and symbolic reasoning. Feedback and collaborations welcome.

Finite Fiels LIm Correlation

Written by Francisco Revelles

1 follower · 34 following

Designing intelligent systems with clarity and purpose. AI, software development, and the art of elegant code.

Edit profile

No responses yet

Francisco Revelles

What are your thoughts?

More from Francisco Revelles

Francisco Revelles

What Prime Fields Can Teach Us About Building Smarter Languag...

By Francisco Revelles

Jun 19

Francisco Revelles

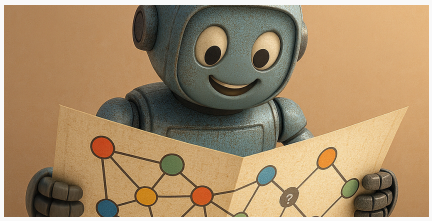
Francisco-Revelles–From System Builder to AI Explorer: My Journey...

By Francisco Revelles

May 6

See all from Francisco Revelles

Recommended from Medium

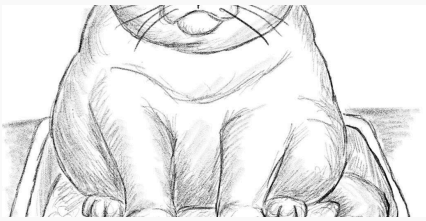


In Towards AI by Michael Shapiro MD MSc

Graph Neural Networks for Knowledge Graphs

A Practical Guide to training Graph Neural Networks with Knowledge Graphs for...

Aug 4 33 2



Mark Riedl

The Intuition Behind How Large Language Models Work

Large Language Models (LLMs) are fancy artificial neural networks. But you don't have...

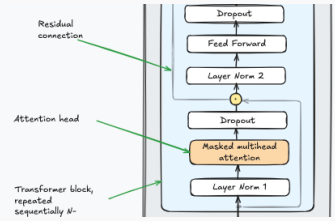
5d ago 245 6

Jose Gomez

OpenAI's Revolutionary Return to Open Source: The Complete Guid...

After six years since the release of GPT-2, OpenAI has made a groundbreaking return t...

Aug 5 11

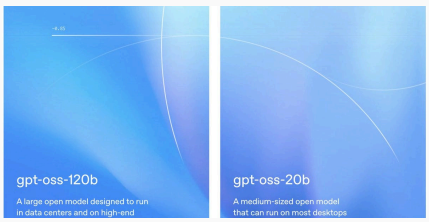


Mahadir Ahmad

Building Large Language Model (LLM) from Scratch in just 160 line...

My first exposure to the concepts of self-attention and Transformers came a few year...

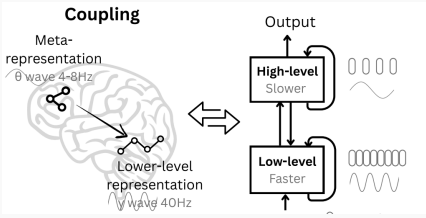
May 12



Isaak Kamau

Things to Know About OpenAI GPT-OSS: Run it Locally on Your...

OpenAI has finally open-sourced a GPT-class large language model for the first time since...



Arvind Nagaraj

The Loop is Back: Why HRM is the Most Exciting AI Architecture in...

Years ago, I sat in Jeremy Howard's FastAI class, right at the dawn of a new era. He was...

6d ago 365 4 Aug 4 771 13

See more recommendations