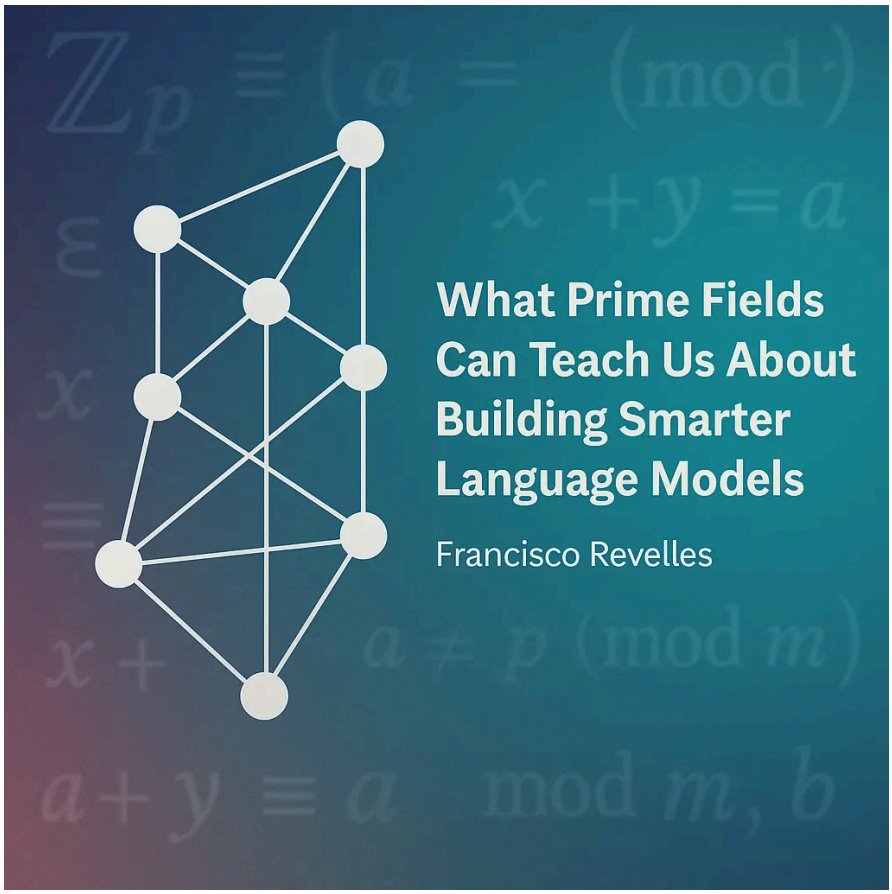
Open in app ↗

Medium Search Write

What Prime Fields Can Teach Us About Building Smarter Language Models

Francisco Revelles 3 min read · Jun 19, 2025

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 generate 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 not from more data, but from deeper mathematics?

This essay explores a novel idea: that the structure of finite fields — particularly those defined modulo a prime number (GF(p)) — can serve not only as inspiration, but as a **framework** for constructing and classifying LLMs.

> **Author's Note:** This is a working hypothesis. If you find this idea relevant to your work, please refrain from reusing or adapting it without attribution. This framework is part of an ongoing research effort, and formal publication is under consideration.

Framing the Problem: Intelligence Without a Map

Today's LLMs are often treated as black boxes. We throw more parameters at them, feed them more data, and hope for better performance. But while scale matters, it alone doesn't give us insight. We lack a language for describing *families* of LLMs. We don't have a principled way to ask: How are these models related? What operations connect them? What transformations preserve capability or introduce novel structure?

In short: we need a grammar for LLM systems. Finite fields might help.

Modular Arithmetic and the Power of Primes

Modular arithmetic is often introduced as "clock math," where numbers wrap around a modulus. But when that modulus is a *prime*, something beautiful happens: the resulting system ($\mathbb{N}/p\mathbb{N}$) forms a **finite field**. In such fields, you can add, subtract, multiply, and crucially — *divide*. Every nonzero element has an inverse. Algebra flourishes.

This field structure is at the heart of cryptography, error correction, and even the theory of elliptic curves. But it also has unexplored potential in structuring how we think about **computational architectures**.

The Core Idea: Constructing LLMs as Field Elements

What if we treated individual LLMs as *elements* of a field GF(p)?

Each model might correspond to an integer modulo p. Arithmetic operations between models could represent interpolation, modular composition, or representational alignment. Some LLMs might behave like multiplicative identities. Others, when combined, cancel out. We might even define derivative-like operations across model space.

In this view, training becomes not just optimization, but **construction** over a discrete, algebraic structure. It might allow us to:

- Define composability rules

- Organize models into algebraic families
- Predict properties of interpolated models
- Interpret scaling behaviors as transformations

Algebraic construction could enforce constraints on models' behavior, limiting drift or collapse.

1. **Efficiency**: Rather than retraining from scratch, we might "build" new models by operating on existing ones.
2. **Scientific Value**: It gives us a new mathematical lens through which to study learning systems.
3. **Human Insight**: This mirrors how many thinkers — from Gauss to Galois to Shannon — have used algebra not just to compute, but to *organize thought*.

Is This New?

This line of reasoning may have analogs in neural modularity, vector space operations, or meta-learning. But to my knowledge, no one has explicitly proposed applying **field theory over primes** to classify and construct LLMs in this way.

I offer this not as a finished theory, but as a working hypothesis — a bridge between mathematics and AI that deserves deeper exploration.

A Final Note

If this resonates with you — as a researcher, a builder, or simply a curious mioite you to engage. We need more ways to think about intelligence that are *structured* but not rigid, *abstract* but still grounded.

The future of AI won't be found just in bigger models. It may be discovered in **smarter frameworks.**

Large Model Languages | Llm | Galois Fields | AI | Abstract Agebra

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 Finite Fields of Size p Can Teach Us About Building Smarter...

By Francisco Revelles

Jun 20

Francisco Revelles

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

By Francisco Revelles

May 6

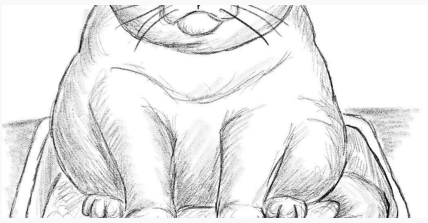
See all from Francisco Revelles

Recommended from Medium

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...



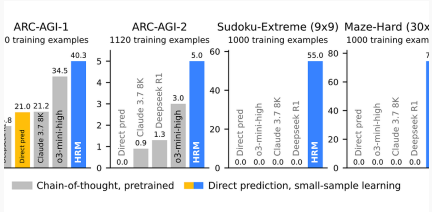
Mark Riedl

The Intuition Behind How Large Language Models Work

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

6d ago 365 4

5d ago 245 6



Allen Liang

Killing LMMs with a tiny 27M model: Hierarchical Reasoning...

Aug 3 28

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...

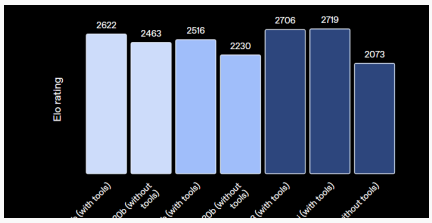
Aug 5 11

In Towards AI by MahendraMedapati

🚀 The Death of Vector Databases? How Agentic RAG is...

Imagine querying a 900-page legal document and getting precise answers in minutes—...

6d ago 592 22



DhanushKumar

GPT-OSS: OpenAI's Open-Weight GPT Models

OpenAI's gpt-oss models (gpt-oss-120b and gpt-oss-20b) are the first open-weight GPT-...

6d ago 50

See more recommendations