Cornell University

Text-Trained LLMs Can Zero-Shot Extrapolate PDE Dynamics

Revealing a Three-Stage In-Context Learning Mechanism

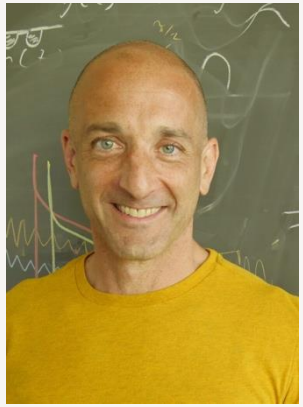
Jiajun Bao, Center for Applied Mathematics, Cornell University
Advisor: Prof. Christopher J. Earls
Tongji University, College of Civil Engineering, Group D ML & AI Seminar
Date: March 12, 2026

Collaborators

Nicolas Boullé
Assistant Professor
Imperial College London

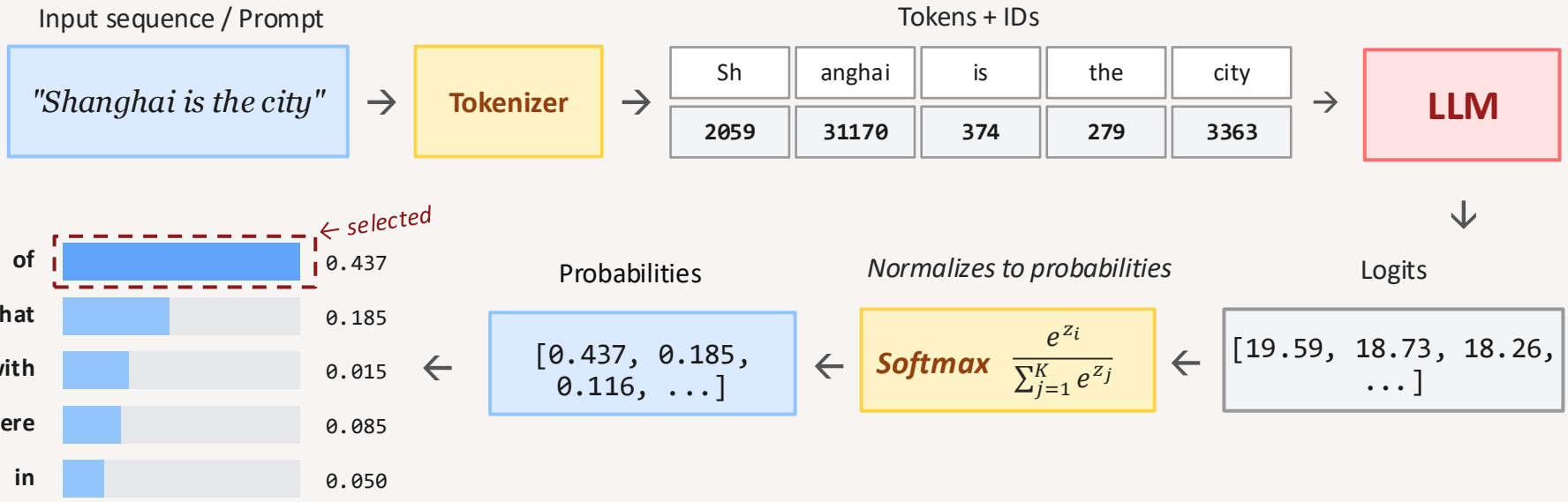
Toni J.B. Liu
PhD Candidate
Cornell University

Raphaël Sarfati
Member of Technical Staff
Goodfire AI



Christopher J. Earls
J. Preston Levis Professor
Cornell University

How do LLMs Generate Text?



1. Tokenize: Text → sub-word tokens (e.g., “Shanghai” → “Sh” + “anghai”).

2. Forward pass: LLM produces a logit vector over the full vocabulary; one score per token, representing next-token likelihood.

3. Sample: Softmax normalizes logits → probabilities; the next token is sampled from this distribution.

Autoregressive: Selected token is appended to input, process repeats — one token at a time, left to right.

The Rise of LLMs & In-Context Learning

In-Context Learning (ICL)

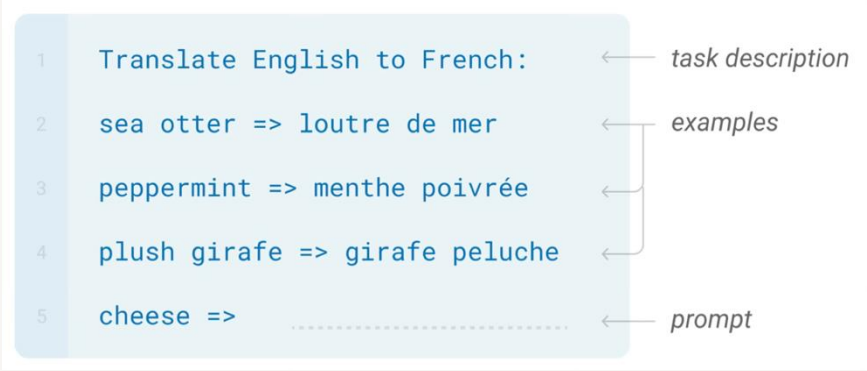
An emergent ability of large language models:

The model is given examples or instructions in the prompt.

It learns to generate appropriate outputs for new instances, without any parameter updates.

No fine-tuning. No retraining.

Zero-shot



Few-shot

Figure from Brown et al., "Language Models are Few-Shot Learners" (NeurIPS 2020)

In-Context Learning Beyond Language

ICL has since been demonstrated beyond language — in domains involving mathematical and numerical reasoning:

- **Forecast time series** by treating numerical sequences as text, matching purpose-built forecasting models
  (Gruver et al., NeurIPS 2023; Jin et al., ICLR 2024)

- **Infer governing principles** of dynamical systems from observed trajectories
  (Liu et al., EMNLP 2024)

- **Perform density estimation** conditioned on in-context examples
  (Requeima et al., NeurIPS 2024; Liu et al. ICLR 2025)

Where does this capacity come from?

This is an emergent capability — it arises in sufficiently large models trained on text and code, and is not observed in smaller models.

Our Work: From 1D Time Series to Spatiotemporal PDEs

The Leap from 1D to PDEs

Previous work mostly focused on data that are 1D. We show pretrained LLMs can continue PDE dynamics directly from serialized solution data — no fine-tuning, no natural language prompting — effectively learning to infer both spatial structure and temporal dynamics.

Our Objective

We do not propose LLMs as a new kind of PDE solver. Instead, we use PDEs as a lens to investigate the inductive biases and numerical priors that emerge from large-scale pretraining.

PDEs Studied

Allen–Cahn	Phase separation in metal alloys (nonlinear)
Fisher–KPP	Population growth with spatial diffusion (nonlinear)
Heat Equation	Thermal diffusion in solids (linear parabolic)
Wave Equation	Vibration and wave propagation (linear hyperbolic)

LLMs Studied

Llama-3.1-8B	*8B parameters*
Llama-3.2-3B	*3B parameters*
Llama-3.2-1B	*1B parameters*
Phi-4-14B / SmolLM3-3B	*Cross-family generality check*

What We Found: Our Roadmap

Q1

Can it extrapolate the dynamics?

Zero-Shot Accuracy

Yes. Without fine-tuning or prompting, LLMs look at past time steps and accurately predict future states of PDE solutions.

Q2

Do its errors make sense?

In-Context Scaling Laws

Yes. Error behavior closely resembles truncation error accumulation in classical finite-difference methods — $\mathcal{O}(1/N_{\mathrm{T}})$ convergence, $\mathcal{O}(N_{\mathrm{X}})$ degradation.

Q3

How is it learning this?

The Learning Mechanism

By analyzing token-level uncertainty, we uncovered a three-stage ICL process: syntax mimicry → exploration → consolidation.

Extrapolation accuracy: LLMs behave like classical numerical PDE solvers.

Internal learning behavior: LLMs also show systematic, stage-wise progression.

Why Do We Need a Serialization Pipeline for LLMs?

① Dimensionality Mismatch

Standard PDE Methods:

Standard PDE methods operate directly on matrices/tensors, preserving spatial and temporal dimensions natively.

LLMs:

LLMs operate strictly on 1D sequences. They generate one token at a time (left to right) and cannot natively ingest multidimensional grids. A coupled space-time grid must be flattened into a token sequence.

② The Tokenizer Bottleneck

LLMs don't read raw numbers — they read sub-word tokens.

The Problem:

A floating-point value like 0.845126 will be split by the tokenizer:

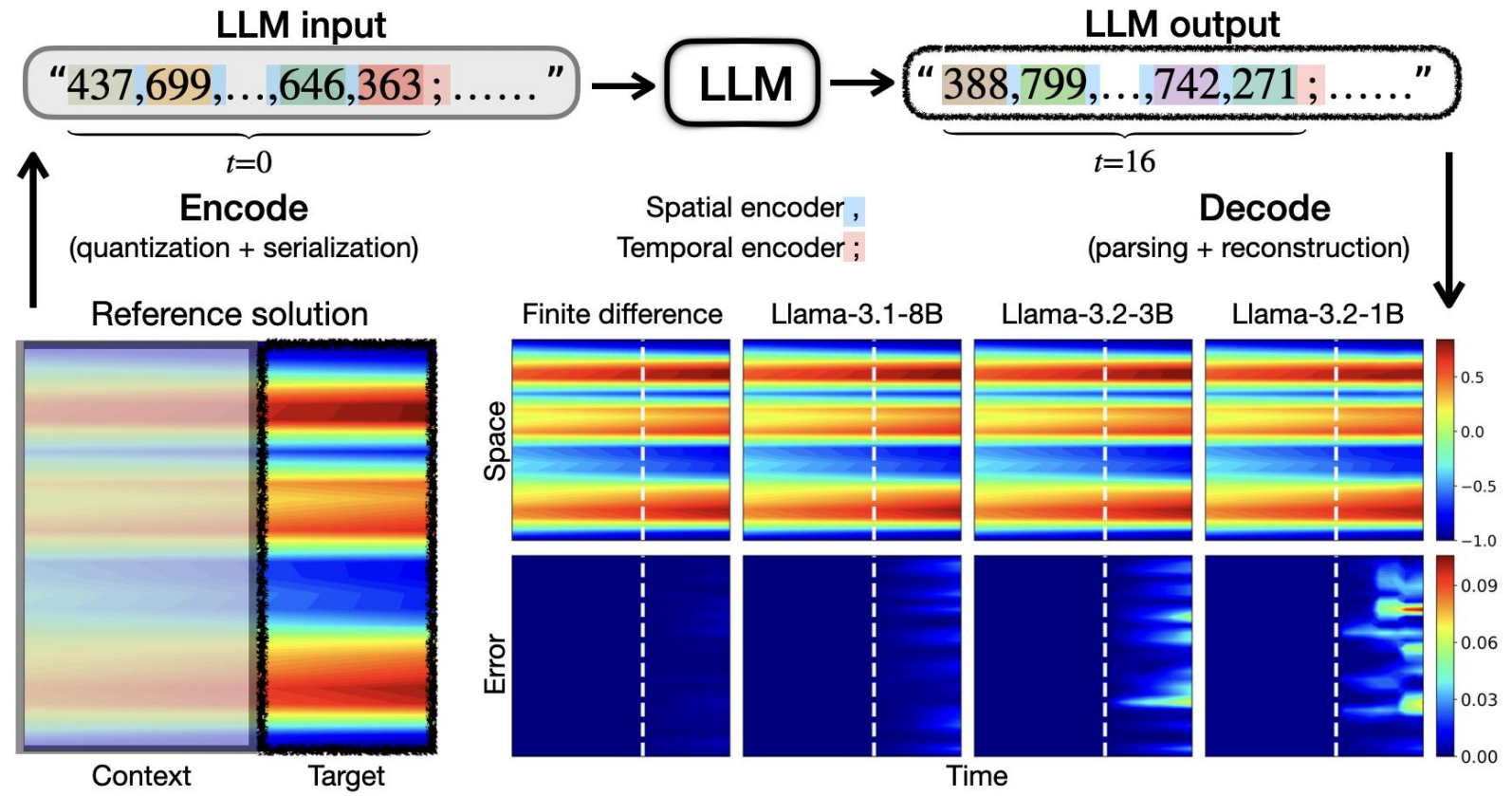
```
0.845126  →  [0] [.] [845] [126]
```

The Result: Mathematical meaning is destroyed, making numerical reasoning unstable.

Our Solution: Quantize to 3-digit integers, so each grid value maps to exactly one token:

```
0.845126  →  743  →  [743] ✓ one token
```

Method: Zero-Shot PDE Extrapolation Workflow



Demo with Allen–Cahn equation

Serialization: Encoding Space and Time

"437,699,…,646,363 ; 445,… ; …"

commas (,) = spatial delimiter semicolons (;) = temporal delimiter **3-digit integers** = quantized values

Direct Grid-Token Alignment

Each token position corresponds to exactly one spatial grid value, enabling precise error computation and uncertainty estimation per location.

Tokenizer Compatibility

We use models (GPT-4, Llama-3) whose tokenizers map every 3-digit number (000–999) to a single token. The framework generalizes to other tokenizer schemes.

No Language Prompting

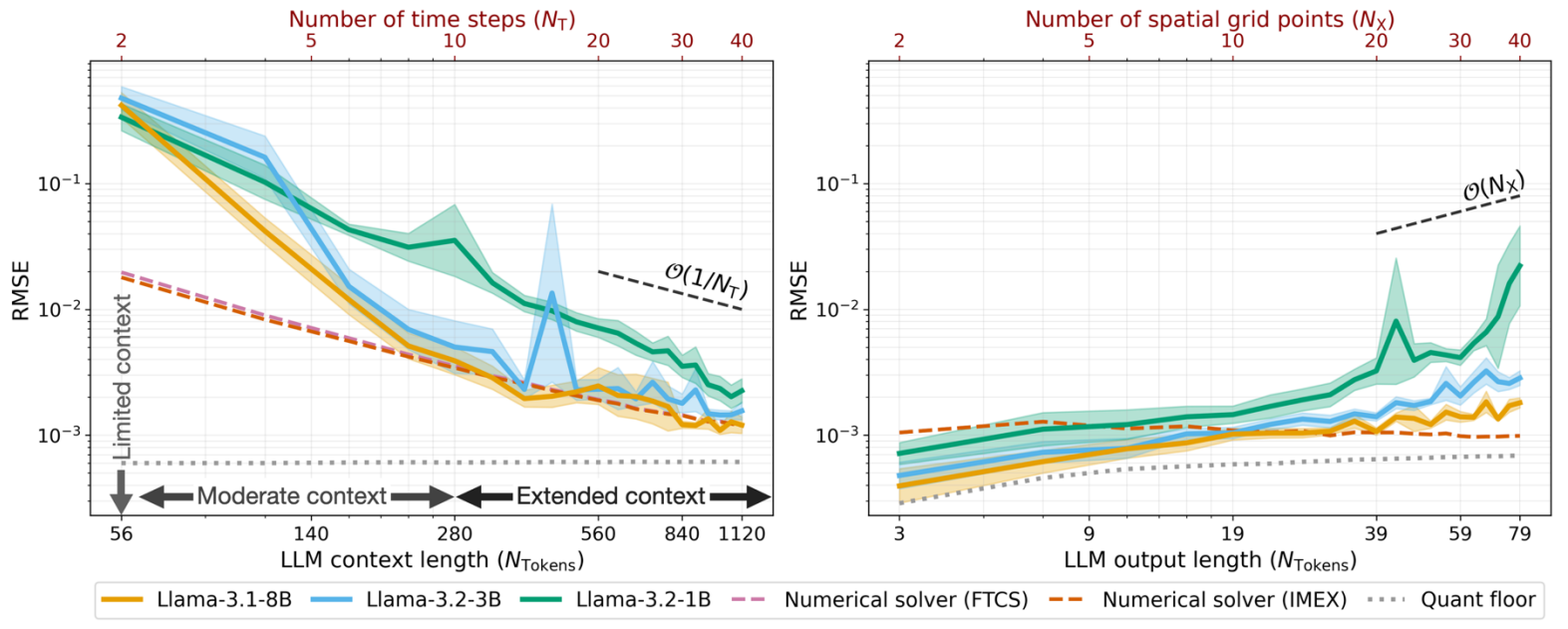
Serialized data is fed directly — no system prompt, no instructions. The model infers structure from numerical data alone.

Two Inference Modes

One-step: predict the next time slice.
Multi-step: recursively roll out by appending each prediction as new context.

One-Step Prediction: In-Context Scaling Laws

Setup: Given past time steps of a discretized PDE solution $\{u(x_i, t_j)\}_{i=1,j=0}^{N_X, N_T-1}$, the model predicts the full spatial state at the next time step $\{u(x_i, t_{N_T})\}_{i=1}^{N_X}$.

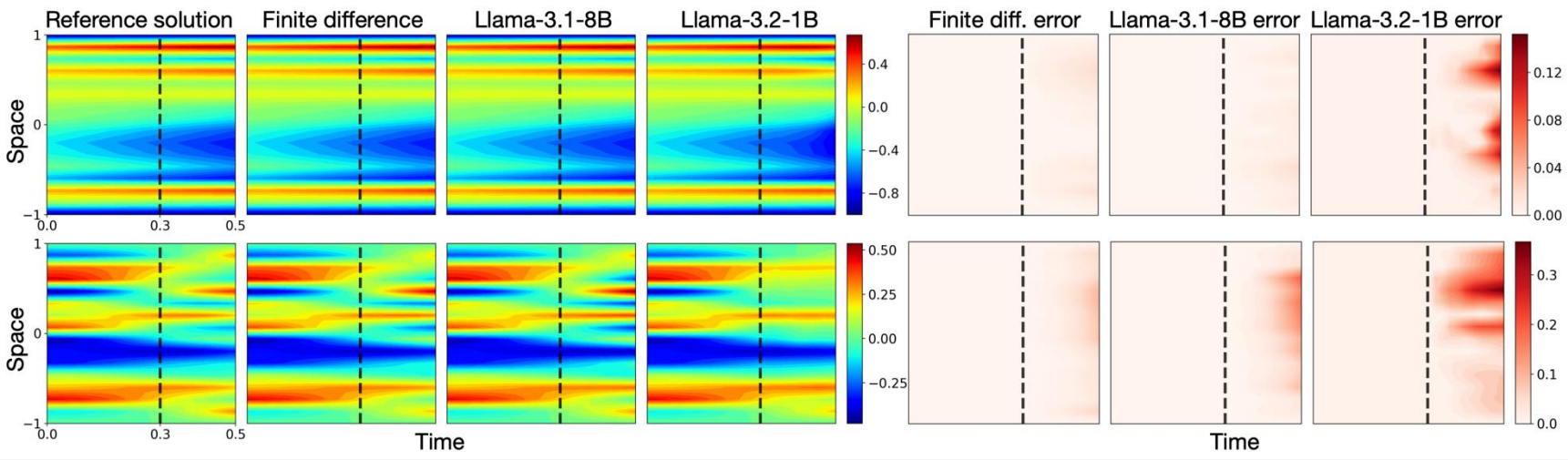


Longer context length improves prediction accuracy
Convergence matches first-order-in-time solvers: Forward Time, Centered Space (FTCS) and Implicit-Explicit (IMEX)

Longer output length degrades prediction accuracy
Finer spatial discretization produces longer outputs, taxing ICL capacity. Larger models are more robust.

Qualitative Multi-Step Rollouts

Setup: Given 16 context steps of a PDE solution, $\{u(x_i, t_j)\}_{i=1,j=0}^{N_X,15}$, the LLM autoregressively generates 10 prediction steps $\{u(x_i, t_{N_T})\}_{i,j=16}^{N_X,25}$. (2:1 context-to-prediction ratio)

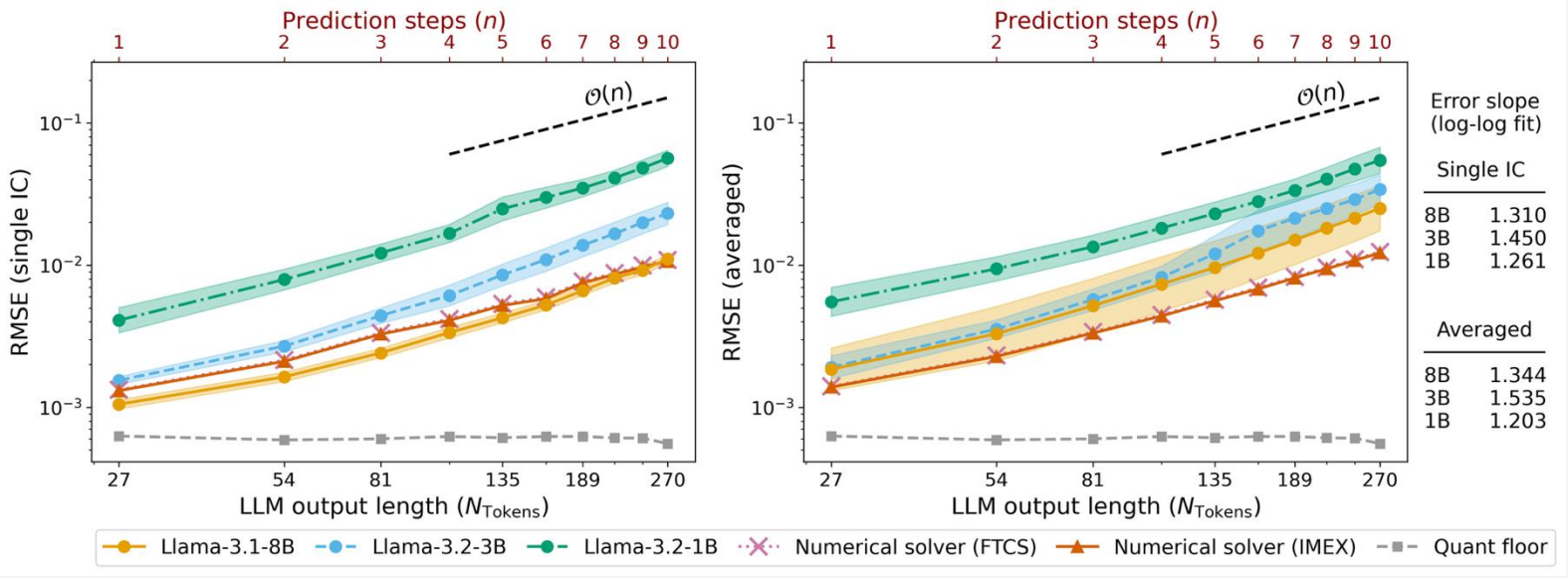


Multi-step rollouts: Allen–Cahn equation (top) and wave equation (bottom)

Coherent predictions: Llama-3.1-8B closely tracks nonlinear reaction–diffusion dynamics (Allen–Cahn) and finite-speed wave propagation, without access to governing equations. Initial conditions are randomly sampled at inference time.

Model size matters: Llama-3.2-1B shows larger deviations and fails to preserve spatiotemporal structure over extended rollouts, highlighting the role of model capacity in sustaining coherent dynamics.

Quantitative Multi-Step Error Growth



Left: rollout from a single random initial condition. Right: average over 20 random initial conditions

Algebraic, not divergent. RMSE grows as $\mathcal{O}(n)$, resembling global error accumulation in classical finite-difference solvers under stable discretizations. Error remains bounded across the 10-step horizon — a nontrivial result given only in-context information, without prompting or access to governing equations.

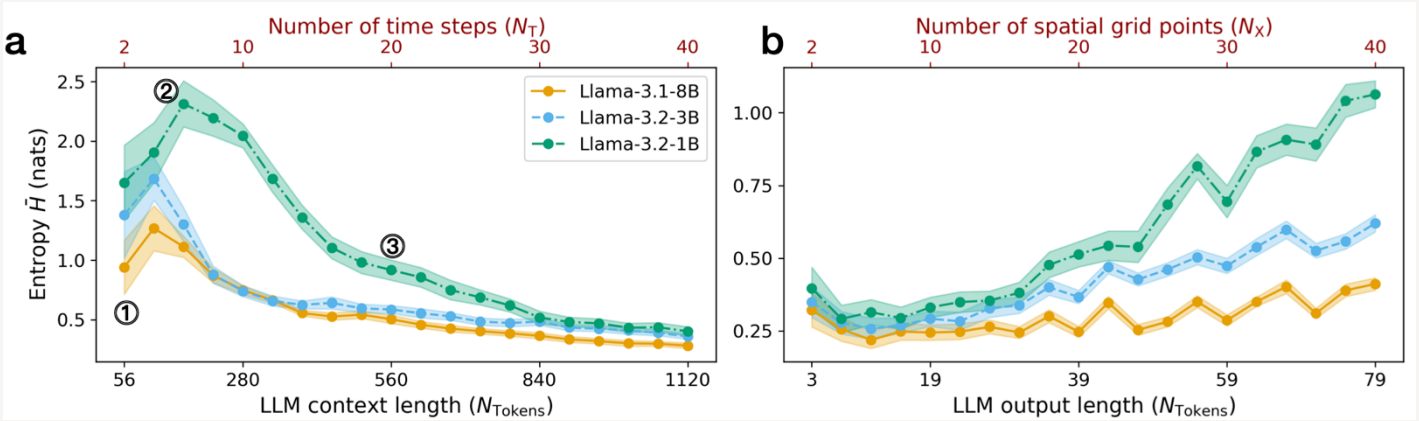
Probing the Learning Mechanism: Predictive Entropy

Measuring Uncertainty via Entropy

At each spatial value token, we compute the Shannon entropy of the model's softmax distribution and average across space:

$$\bar{H}(N_{\mathrm{T}}, N_{\mathrm{X}}) = -\frac{1}{N_{\mathrm{X}}} \sum_{i=1}^{N_{\mathrm{X}}} \sum_{y \in \mathcal{V}} p(\, y \mid x_i, N_{\mathrm{T}}\,) \, \log p(\, y \mid x_i, N_{\mathrm{T}}\,) \,,$$

where $\mathcal{V}$ denotes the tokenizer's vocabulary, and $p(\, y \mid x_i, N_{\mathrm{T}}\,)$is the predicted probability of token y at spatial location x_i, given N_T prior time steps in the serialized input.



Mean spatial entropy vs. context length

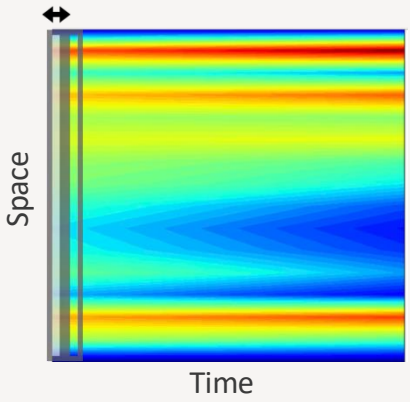
Mean spatial entropy vs. output length

Low entropy
model is confident
(sharp distribution)

High entropy
model is uncertain
(spread across candidates)

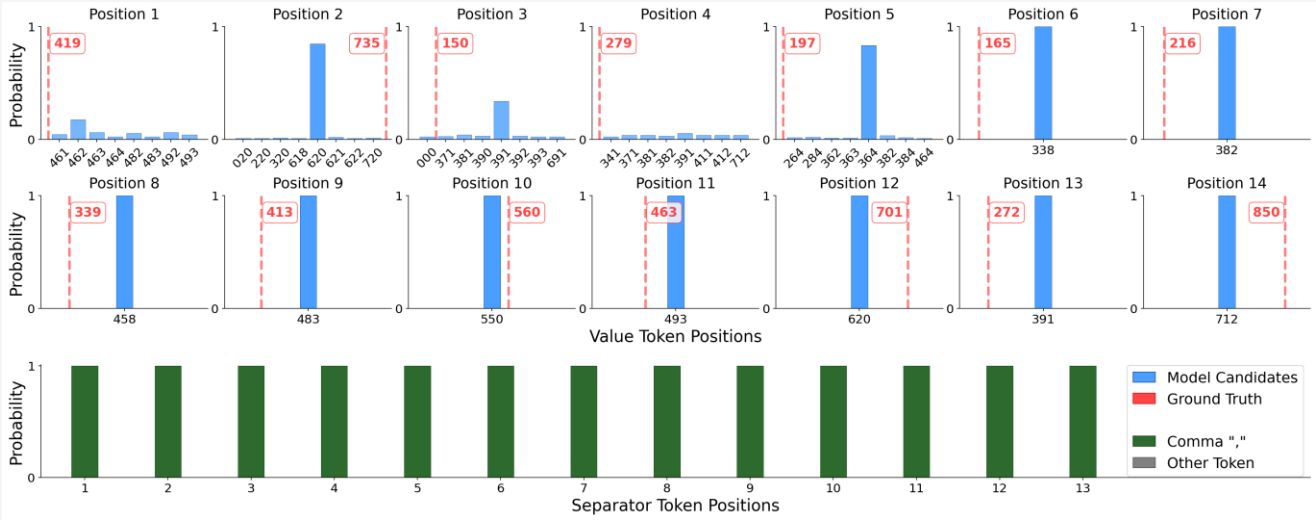
Three-Stage ICL Mechanism: ① Syntax-Only

$N_T = 2$ — Stage 1 of 3



Given data $u(x, t_{0:1})$

Predict $\hat{u}(x, t_2)$



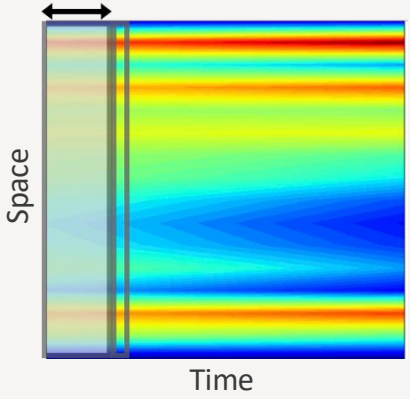
Token-level softmax distributions (Llama-3.1-8B)

Low entropy, high error. Separator tokens (commas) are predicted with near-perfect confidence. Spatial value tokens act as generic placeholders — the format is correct, but values bear no relation to the true PDE dynamics.

Key insight: Syntax is acquired before any meaningful understanding of the dynamics emerges.

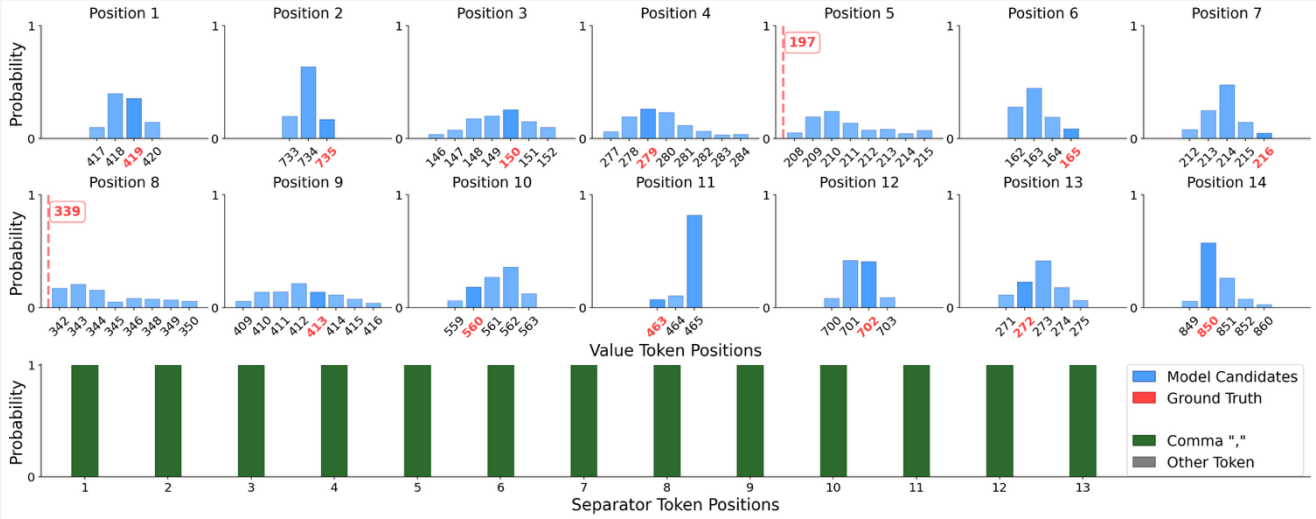
Three-Stage ICL Mechanism: ② Exploratory

$2 < N_T < 10$ — Stage 2 of 3



Given data $u(\boldsymbol{x}, t_{0:4})$

Predict $\hat{u}(\boldsymbol{x}, t_5)$



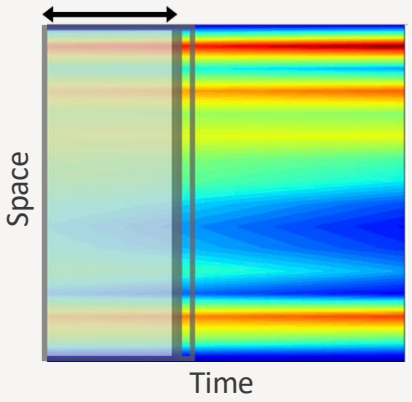
Token-level softmax distributions (Llama-3.1-8B)

Peak entropy, rapidly improving accuracy. Broad token distributions reflect the model exploring plausible continuations. Partial alignment with ground truth emerges across spatial positions.

Key insight: The model transitions from surface-level syntax to beginning to internalize spatiotemporal dynamics.

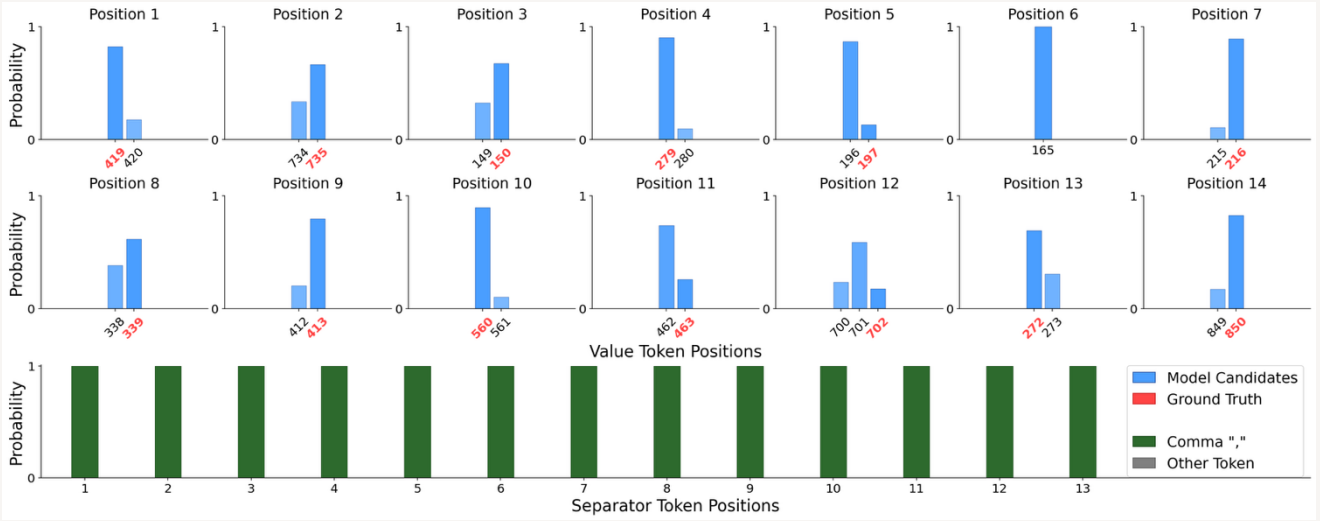
Three-Stage ICL Mechanism: ③ Consolidation

$N_T \geq 10$ — Stage 3 of 3



Given data $u(\boldsymbol{x}, t_{0:19})$

Predict $\hat{u}(\boldsymbol{x}, t_{20})$



Token-level softmax distributions (Llama-3.1-8B)

Declining entropy, continued accuracy gains. Sharp, confident distributions converge toward true PDE values. Predictions reflect coherent, physically meaningful dynamics.

Key insight: The model's predictions increasingly reflect accurate, physically grounded PDE dynamics.

Summary & Discussion

Zero-Shot Accuracy

Pretrained LLMs extrapolate PDE dynamics accurately from serialized numerical data — no fine-tuning, no prompting, no task-specific training.

In-Context Scaling Laws

Error behavior mirrors classical numerical analysis: $\mathcal{O}(1/N_{\mathrm{T}})$ convergence with context, $\mathcal{O}(N_{\mathrm{X}})$ degradation with spatial resolution, algebraic error growth in rollouts.

Three-Stage Mechanism

Entropy analysis reveals: syntax acquisition → exploratory numerical behavior → convergence to accurate, physically grounded predictions.

Open Questions & Future Directions

Higher dimensions: Do new ICL behaviors emerge as spatial complexity increases?

Physics-aware prompting: How does symbolic PDE knowledge shape in-context reasoning?

Internal representations: What compositional structures support generalization over spatiotemporal dynamics?

Thank You!

Scan for paper & code

Questions & Discussion

Jiajun Bao, Center for Applied Mathematics, Cornell University
jb2777@cornell.edu

Paper: *"Text-Trained LLMs Can Zero-Shot Extrapolate PDE Dynamics, Revealing a Three-Stage In-Context Learning Mechanism",* ICLR Workshop on AI & PDE – Oral (2026)

Code: github.com/Jiajun-Bao/LLM-PDE-Dynamics