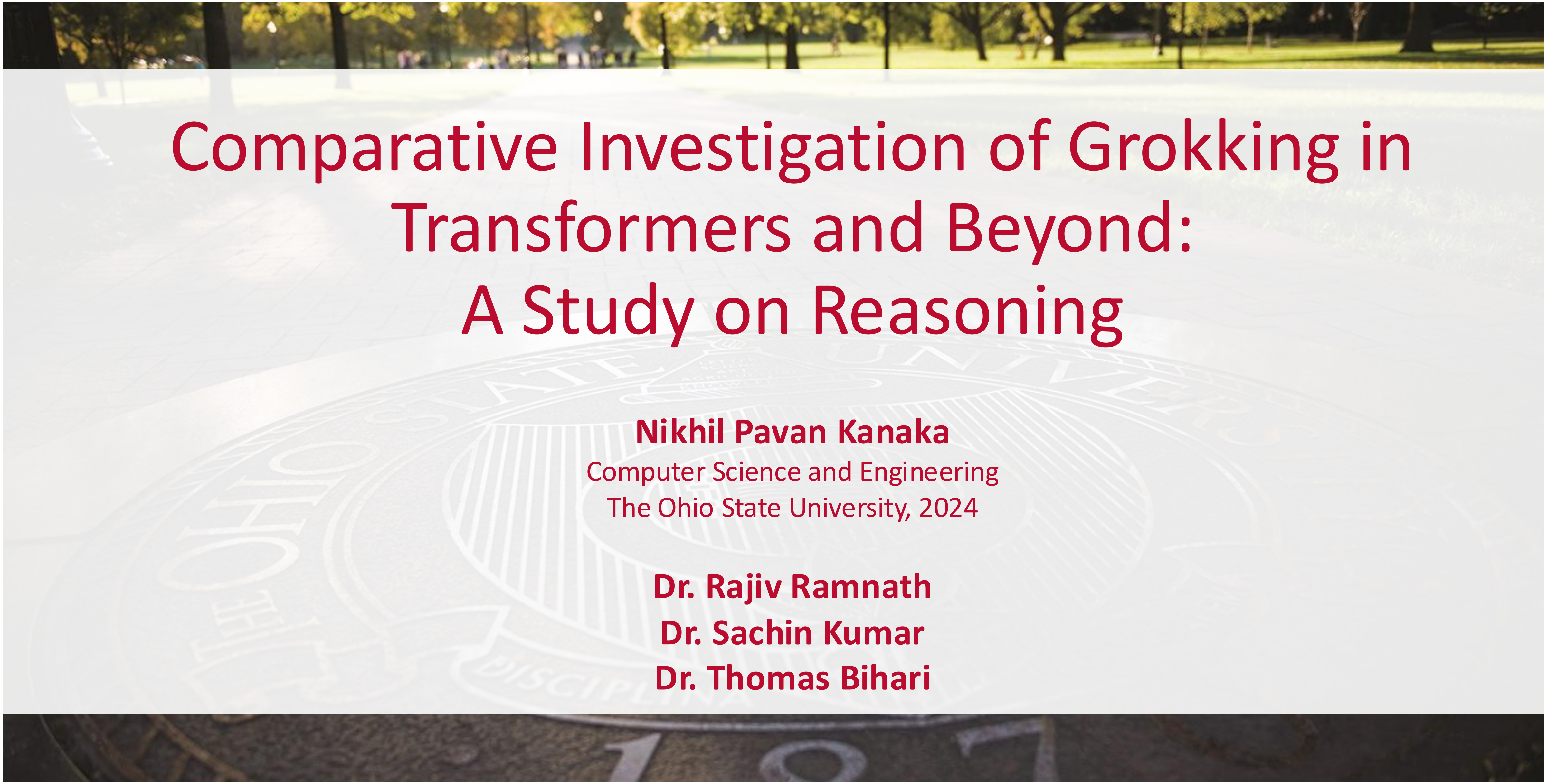


Comparative Investigation of Grokking in Transformers and Beyond: A Study on Reasoning

Nikhil Pavan Kanaka
Computer Science and Engineering
The Ohio State University, 2024

Dr. Rajiv Ramnath
Dr. Sachin Kumar
Dr. Thomas Bihari

THE OHIO STATE UNIVERSITY
COLLEGE OF ENGINEERING

Research Contributions

- **Identified Limitations of LLMs**: Focused on reasoning capabilities and their challenges.
- **Explored Reasoning Mechanisms and Grokking**: Current approaches and Insights.
- **Reasoning Dataset**: Centered on comparison tasks.
- **Conducted Experimental Evaluation**: Tested multiple models to assess grokking performance.
- **Analyzed Factors Influencing Grokking**: Dataset Parameters that impact generalization.
- **Identified Optimal Configurations for Reasoning**: Insights to enhance grokking.

LLMs: Revolutionizing NLP, Yet Facing Challenges

The Rise of LLMs:
LLMs have transformed natural language understanding and generation.

Challenges in Reasoning:

- Despite their successes, LLMs face limitations in tasks that demand **logical deduction**, **systematic reasoning**, and **generalization**.
- Reasoning tasks involve understanding complex relationships, making inferences, and applying knowledge to novel contexts.
- These tasks often expose weaknesses, leading to issues like hallucinations and inaccurate outputs, even when the language is fluent and coherent.

Research Motivation

Need for Systematic Exploration:
Addressing LLM limitations in reasoning is crucial for advancing their utility.

Research Goal:

- To systematically evaluate, improve, and understand reasoning capabilities in LLMs. This involves designing reasoning-specific datasets, experimenting with various models and analyzing performance across configurations.
- To investigate gaps in understanding reasoning within LLMs, this research will explore the nuances of grokking and generalization.

Existing Methods

Focused on enhancing reasoning capabilities without delving into the model's internal reasoning mechanisms or explicitly improving its underlying components.

Can be categorized into:

- Parameter-Frozen Paradigm
- Parameter-Tuning Paradigm
- Hybrid Approaches

Parameter-Frozen Paradigm

Utilizes LLMs without altering internal parameters, focusing on strategic prompt engineering.

- **Zero-Shot Learning**:
    - Models perform tasks without prior task-specific training.
    - Zero-Shot Chain-of-Thought[1] (CoT) prompting generates step-by-step reasoning via simple prompts like "Let's think step by step”.
    - Demonstrated success in arithmetic and symbolic reasoning tasks.
- **Few-Shot Learning**:
    - Incorporates task-specific examples in prompts.
    - Few-Shot CoT Prompting[2] (Wei et al., 2022) provides guided reasoning processes.
    - Enhances performance in complex reasoning tasks, requiring careful selection of exemplars.

[1] Kojima, T., Gu, S., Reid, M., Matsuo, Y., & Iwasawa, Y. (2022). *Large Language Models are Zero-Shot Reasoners*

[2] Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E. H., Le, Q. V., & Zhou, D. (2022). Chain-of-thought prompting elicits reasoning in large language models.

Parameter-Tuning Paradigm

Adjusting the model parameters to enhance their performance on specific reasoning tasks.

- **Full-Parameter Tuning**: Modifying all the parameters of an LLM to specialize it for particular tasks.
  - WizardMath[1]: Fine-tuned for mathematical reasoning using Reinforcement Learning with Evol-Instruct Feedback (RLEIF).
  - MAmmoTH[2]: On MathInstruct, combining CoT and program-of-thought rationales.
- **Parameter-Efficient Tuning**: Aim to adapt LLMs with minimal changes to the model’s parameters.
  - Low-Rank Adaptation (LoRA) enables fine-tuning with limited resources.
  - LLM-Adapters[3]: Framework that integrates various adapters into LLMs.

[1] Luo, H., Sun, Q., Xu, C., Zhao, P., Lou, J., Tao, C., Geng, X., Lin, Q., Chen, S., & Zhang, D. (2023). *WizardMath: Empowering mathematical reasoning for large language models via reinforced evol-instruct*.

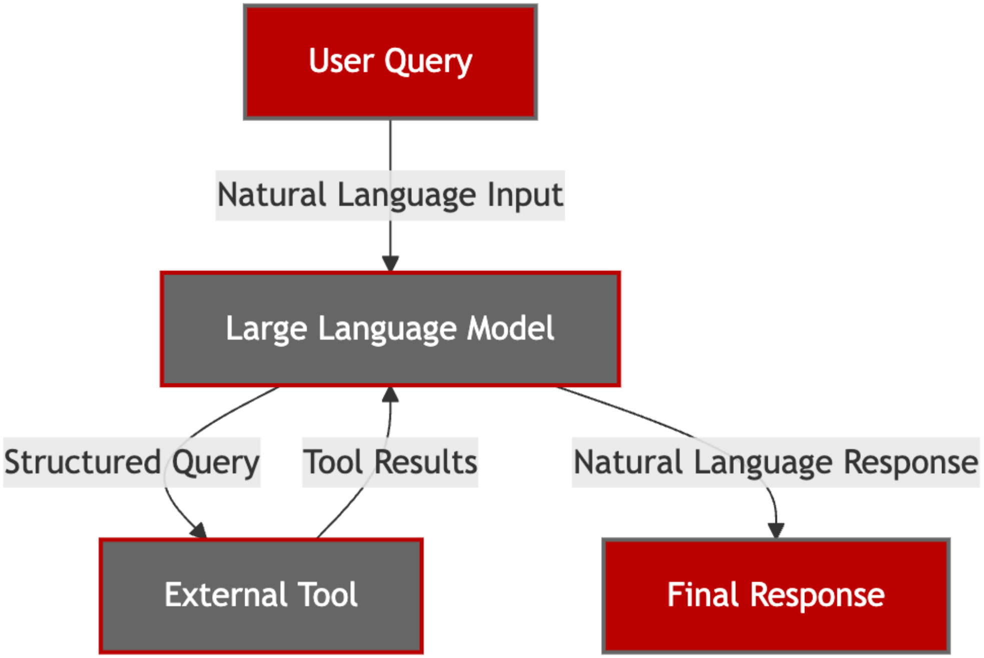
[2] Yue, X., Qu, X., Zhang, G., Fu, Y., Huang, W., Sun, H., Su, Y., & Chen, W. (2023). MAmmoTH: Building Math Generalist Models through Hybrid Instruction Tuning.

[3] Hu, Z., Wang, L., Lan, Y., Xu, W., Lim, E.-P., Bing, L., Xu, X., Poria, S., & Lee, R. K.-W. (2023). LLM-Adapters: An Adapter Family for Parameter-Efficient Fine-Tuning of Large Language Models.

Hybrid Approaches

Combines LLM generative capabilities with symbolic solvers[1] for precise logical reasoning.

- **LLMs as Translators**:
    - Converts natural language to symbolic representations for processing by tools.
    - Reduces errors like hallucinations and provides verifiable reasoning chains.
- **Challenges**:
    - Errors in translation or symbolic execution can lead to failures.
    - Symbolic solvers require explicit premises, limiting their ability to infer implicit relationships.



[1] Zhang, Y., Chen, S., & Kambhampati, S. (2023). *A closer look at tool-based logical reasoning with LLMs: The choice of tool matters*

Challenges

Existing methodologies fall short of addressing fundamental questions about improving a model's capability to reason.

Key limitations:

- Opaque Reasoning Processes: Lack of Interpretability
- Over-Reliance on External Tools
- Dataset Constraints
- Focus on Fundamental Reasoning

Grokking

Grokking describes the phenomenon where transformers generalize effectively long after overfitting. Models leveraging grokking excel at out-of-distribution (OOD) tasks, demonstrating robust generalization capabilities.

Introduced by **Power et al. (2022)**[1],

Liu et al., (2022)[2]: Explored structured representations through prolonged training.

Murty et al., (2022)[3]: how transformer computations align with hierarchical encodings.

Nanda et al., (2023)[4]: Proposed metrics to quantify grokking progress.

Murty et al., (2023)[5]: Structural grokking, emphasizing hierarchical representations.

Furuta et al., (2024)[6]: Investigated modular arithmetic tasks.

Wang et al., (2024)[7]: Identified generalizing circuits as structured pathways within transformers.

1. Power, A., Burda, Y., Edwards, H., Babuschkin, I., & Misra, V. (2022). Grokking: Generalization Beyond Overfitting on Small Algorithmic Datasets.
2. Liu, Z., Kitouni, O., Nolte, N., Michaud, E. J., Tegmark, M., & Williams, M. (2022). Towards Understanding Grokking: An Effective Theory of Representation Learning
3. Murty, S., Sharma, P., Andreas, J., & Manning, C. D. (2022). Characterizing Intrinsic Compositionality in Transformers with Tree Projections.
4. Nanda, Neel, et al. Progress measures for grokking via mechanistic interpretability.
5. Murty, S., Sharma, P., Andreas, J., & Manning, C. D. (2023). Grokking of Hierarchical Structure in Vanilla Transformers.
6. Furuta, H., Minegishi, G., Iwasawa, Y., & Matsuo, Y. (2024). Interpreting Grokked Transformers in Complex Modular Arithmetic.
7. Wang, B., Yue, X., Su, Y., & Sun, H. (2024). Grokked Transformers are Implicit Reasoners: A Mechanistic Journey to the Edge of Generalization.

Objectives

- **Investigate Grokking in Reasoning Tasks**:
Examine the phenomenon of grokking in state-of-the-art LLMs, focusing on identifying the conditions that enable it.
- **Evaluate Reasoning Capabilities**:
Compare the reasoning and grokking performance of multiple LLMs, including GPT, LLaMA, RWKV, and Mamba, specifically for tasks requiring logical comparison and deduction.
- **Dataset Analysis**:
Explore the influence of dataset characteristics on grokking speed and effectiveness in reasoning.
- **Understand Generalization Limitations**:
Identify the constraints of current LLMs in reasoning tasks, particularly their ability to generalize beyond the training data. Provide insights for improving reasoning capabilities in LLMs.

Scope and Focus

- **Focused on Reasoning Tasks**:
The study is limited to reasoning tasks, emphasizing logical comparison and deduction.
- **Experimental Boundaries**:
The research is confined to analyzing grokking behavior under different dataset designs and model configurations. The study focuses on understanding how these factors influence grokking behavior.
- **Prioritizing Common Ground:**
When comparing both model performance and dataset design, the research prioritizes identifying a common ground to ensure fair and meaningful comparisons.

Methodology

Reasoning Dataset

- Custom datasets were curated with controlled variations.

Model Experiments

- Experiments were conducted using open-source LLMs, evaluating their architectural performance and behaviour under identical conditions to ensure fair comparisons.

Controlled Analysis

- Models were assessed across consistent experimental setups, highlighting differences in their grokking dynamics.
- Factors such as training duration, hyperparameter configurations, and architectural choices were analyzed individually to provide targeted insights into their respective contributions.

Dataset Design

- Comparison reasoning task is chosen as it effectively demonstrated grokking[1].
- Evaluate entities based on attribute values to determine relationships.

  Example: Compare "Alice (30 years)" and "Bob (25 years)" → Alice > Bob for attribute age.

Parameter Controls

- **Number of Entities:** Total entities in the dataset. Scales entity pairs quadratically.
- **Number of Attributes:** Total attributes in the dataset. More attributes increase dataset richness.
- **Values per Attribute:** Range of possible values for attributes.
- **Inferred to Atomic Ratio:** Ratio between inferred and atomic facts.
- **In-Domain to Out-of-Domain Ratio:** Ratio of in-domain to out-of-domain entities.

[1] Wang, B., Yue, X., Su, Y., & Sun, H. (2024). Grokked Transformers are Implicit Reasoners: A Mechanistic Journey to the Edge of Generalization.

Dataset Configuration

To enable a meaningful comparison of reasoning performance across models, experiments were conducted with consistent dataset configurations.

Number of Entities: **1000**

Number of Attributes: **20**

Values per Attribute: **20**

In-domain to Out-of-domain Ratio: **0.9**

Test Dataset Size: **3000**

Inferred to Atomic Fact Ratio: **12.6**

Model Configuration

Architecture: Comparable to GPT-2 Small (8 layers).

Batch Size: 512

Precision: fp16 (half-precision).

Sequence Length: Capped at 10 tokens.

Weight Decay: 0.1

Dropout: No

Gradient Accumulation: Single-step

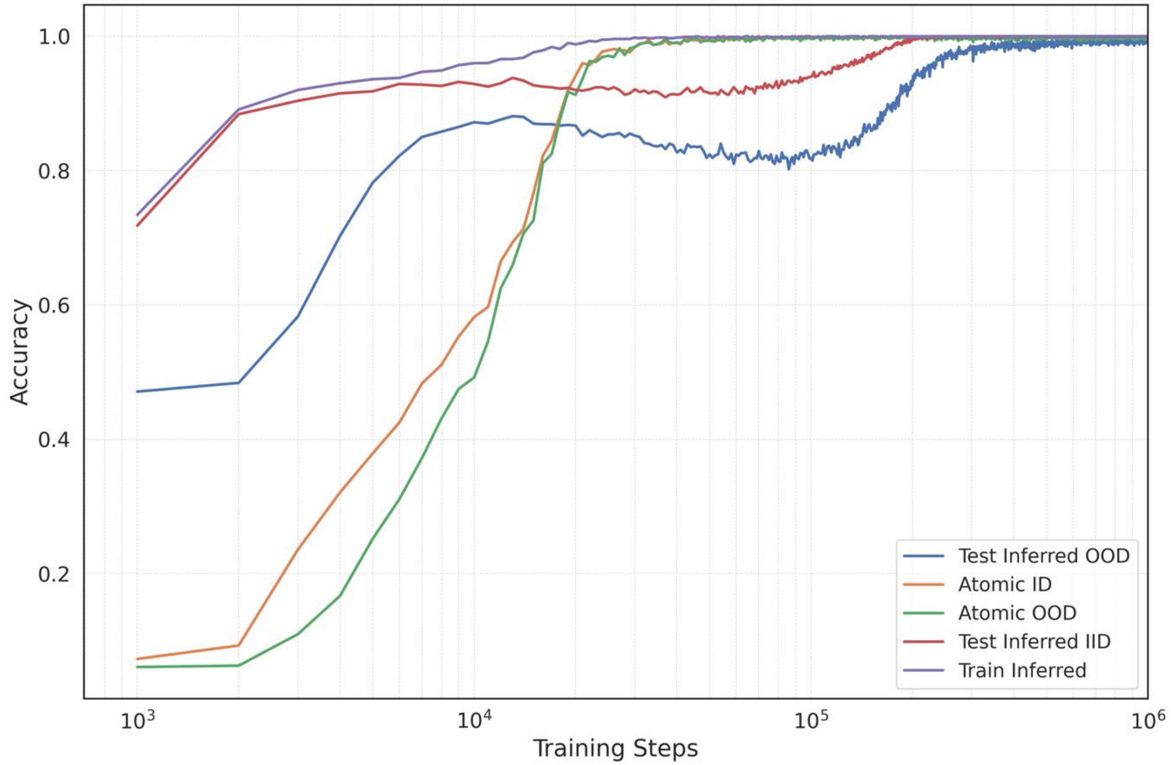
GPT2

Demonstrated outstanding performance on the comparison task.

Key Observations

• Achieved high accuracy in both in-distribution (ID) and out-of-distribution (OOD) inference, after extensive overfitting. (~2,270e)

• Highlights a nuanced progression where overfitting enhances OOD generalization rather than impeding it.



LLama

LLaMA is also a transformer-based large language model. It employs multi-head self-attention mechanisms and modular optimizations.

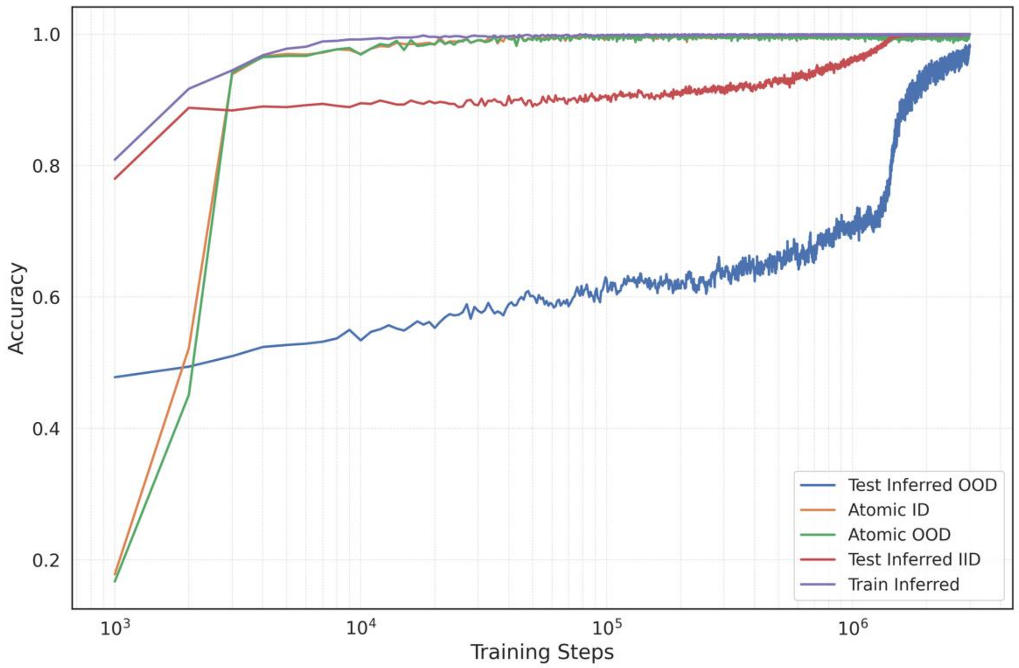
Key Architectural Differences:

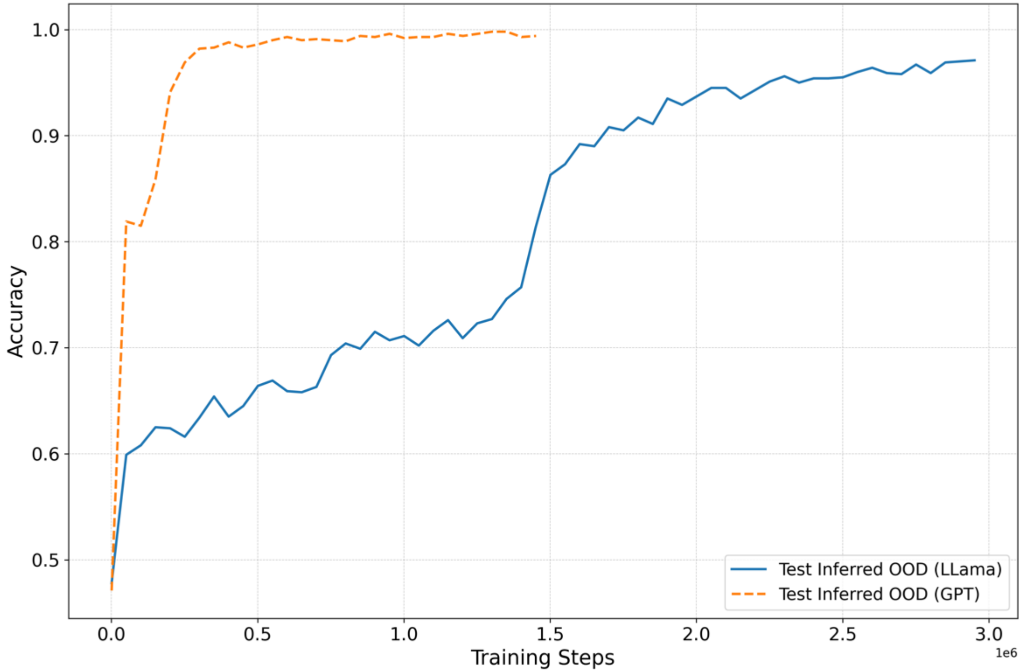
- **Tokenization:** SentencePiece vs. GPT's Byte Pair Encoding (BPE).
- **Normalization:** Post-layer normalization instead of GPT's pre-layer normalization.
- **Positional Encoding:** Rotary Positional Embeddings (RoPE) for positional context.

Multi-layer attention and residual connections allow efficient flow of information across layers, enabling compositional reasoning.

Llama Results

- While not so different from GPT architecture, LLaMA exhibited **slower convergence**.
- Delayed spike in OOD accuracy way after perfect ID accuracy was observed. (~8,072e)
- Highlights architectural components that may influence reasoning and generalization.
- Further study of LLaMA vs GPT can identify components critical for grokking.
- Incorporating elements from GPT might address observed limitations in speed.





RWKV

RWKV is a hybrid large language model that combines elements of RNNs and Transformers

Key Architectural Features:

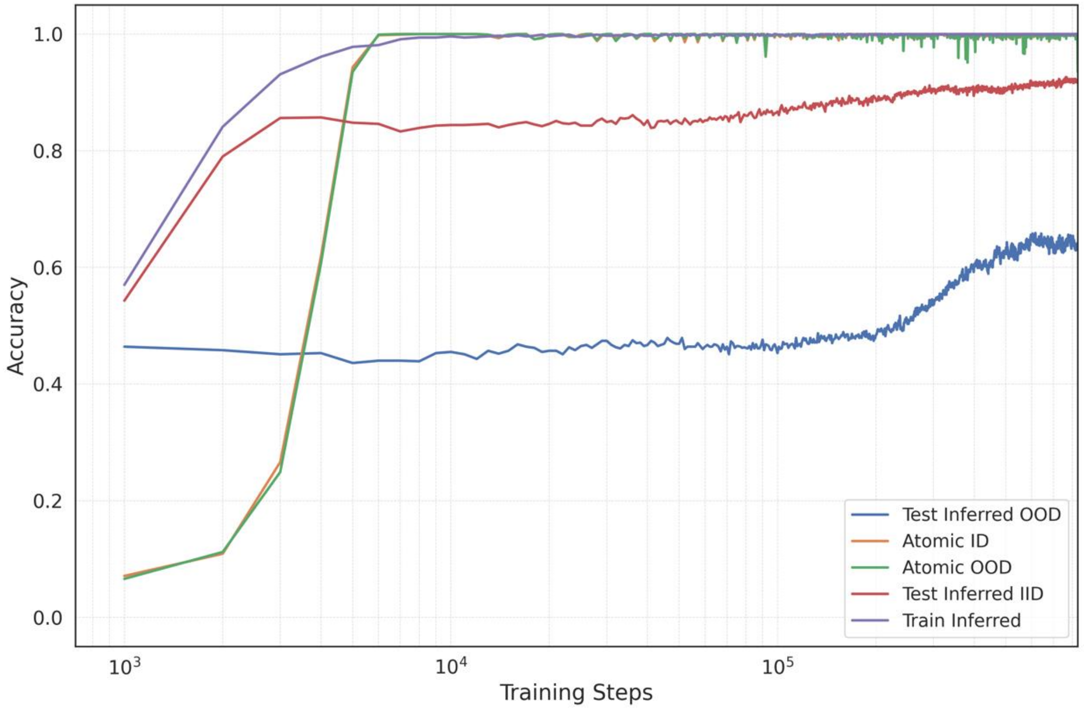
- Sequential Token Processing, while maintaining a memory-like state for long-term dependencies.
    - *Time-Mixing*: Captures sequential relationships across tokens.
    - *Channel-Mixing*: Extracts meaningful features from the input.
- Log-Space Attention: Mimics the long-range capabilities of transformers but more efficient.

Motivation:

- **Hybrid Design**: Allows parameter sharing across time steps focusing on temporal dependencies.
- **Potential for Generalization**: Implicit memory mechanisms may retain and reuse knowledge.
- **Scalability**: Its sequential and efficient nature offers promise for scaling up models while managing computational costs. Understanding RWKV's performance highlights critical trade-offs between efficiency and reasoning depth.

RWKV Results

- RWKV did **not** achieve perfect grokking under tested conditions, unlike GPT and LLaMA.
- OOD accuracy improved marginally at later training stages but quickly plateaued.
- Increased training duration may enable better pattern recognition for reasoning tasks.
- Enhancing cross-layer knowledge sharing may improve systematic generalization.
- RWKV’s design offers a computationally efficient alternative to transformers. It serves as a valuable platform for exploring reasoning and implicit memory.



Mamba

Mamba is a large language model (LLM) designed to achieve lightweight and efficient computations.

Introduces a simplified attention mechanism that reduces computational complexity. Unlike GPT’s dense self-attention layers, Mamba employs mechanisms inspired by or state-space models.

Key architectural features:

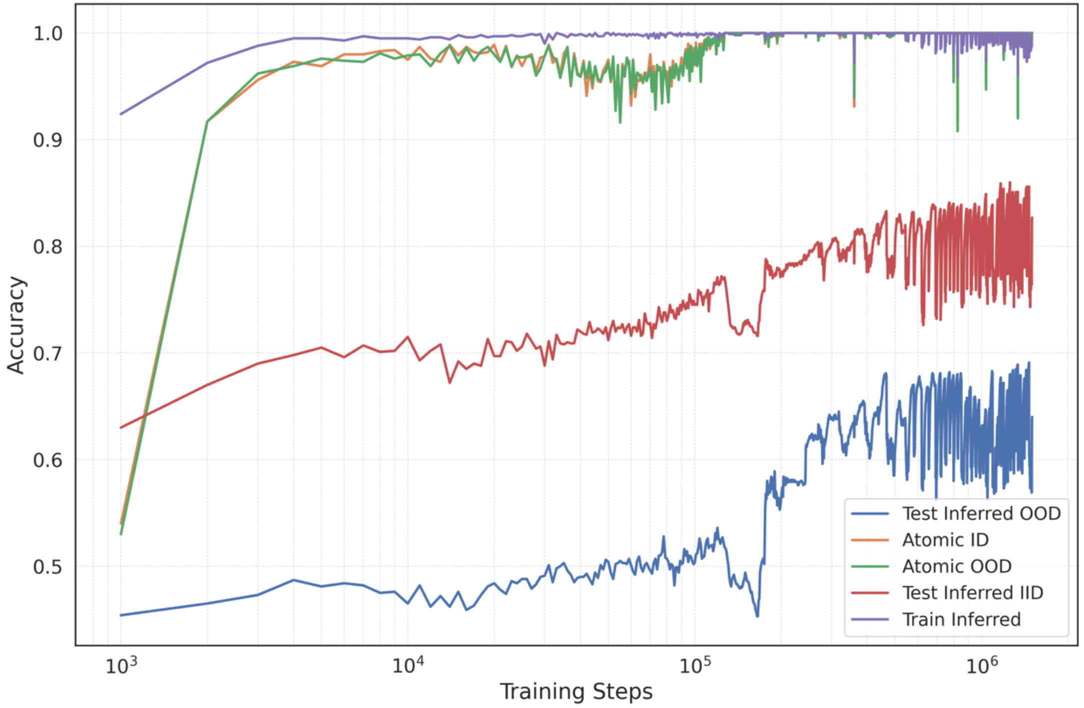
- **Tokenization**: Similar to WordPiece but opts for simplicity over GPT's BPE or LLAMA’s SentencePiece.
- **Positional Encodings**: Computationally simpler but less effective.
- **Lightweight feed-forward networks** (FFNs): Designed for efficiency but can limit learning stability.

Motivation:

- Mamba incorporates recurrent-inspired designs and modular connections to facilitate efficient information sharing across layers and improved memory utilization**.**
- The lightweight design of Mamba also offers practical advantages for overtraining.

Mamba Results

- Mamba achieved rapid improvement in atomic fact accuracy. However, performance fluctuated before stabilizing, indicating instability in learning basic patterns. Unlike atomic facts, inference accuracy **never stabilized**, highlighting limitations.
- The findings suggest that while Mamba excels at lightweight tasks, its design struggles to handle deep reasoning and systematic generalization.
- Mamba is a relatively new architecture, and its configuration may not be fully optimized.
- Further experiments with a mature codebase and optimized training strategies are essential to assess its full potential.



Key Dataset Parameters Influencing Grokking

To enable meaningful comparison of reasoning performance across various dataset-related parameters, experiments were conducted using a consistent model configuration: GPT-2 (8 layers)

Validated Factors (from Prior Studies):

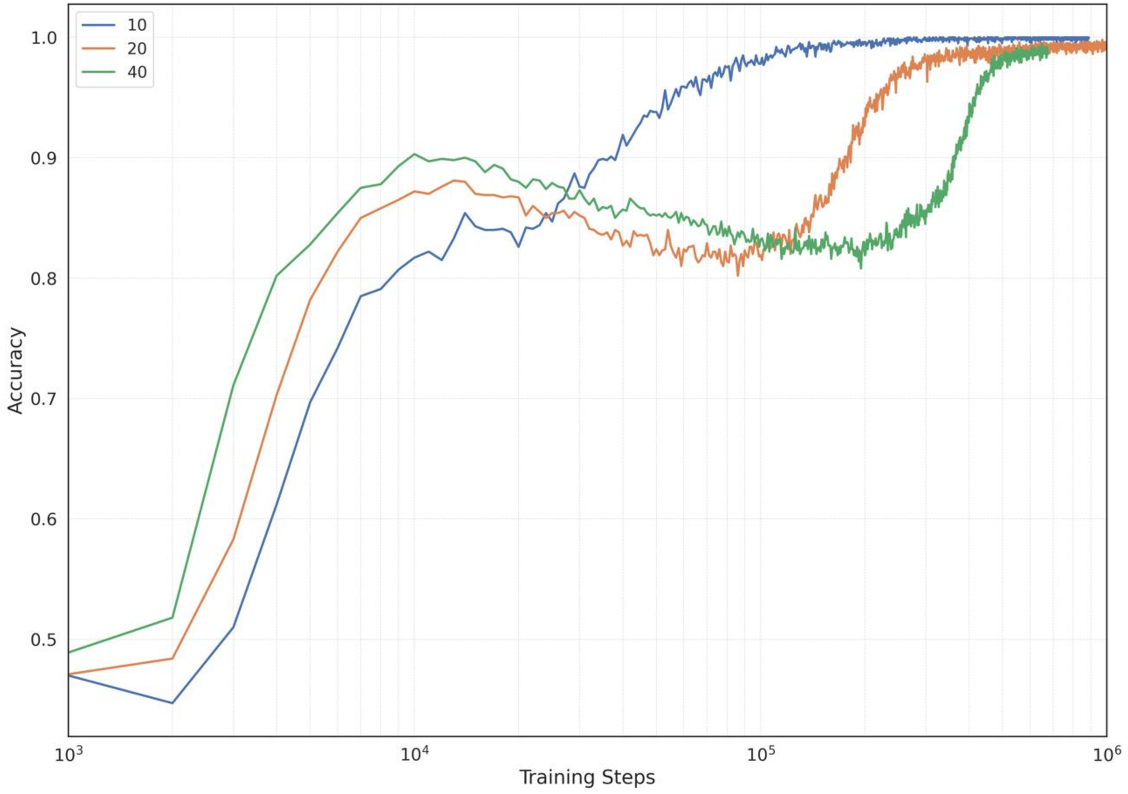
- **Dataset Size:** The absolute size of the training dataset has minimal impact.
- **Atomic to Inferred ratio:** The speed at which a transformer model generalizes correlates strongly with the ratio of inferred facts to atomic facts in the training data. A higher ratio accelerates the grokking process, enabling the model to achieve generalization more rapidly.

Novel Contributions

- **Attribute values:** The sparsity of attribute values affect the model’s ability to reason.
- **Number of Entities:** Increased complexity challenges the model's ability to comprehend.

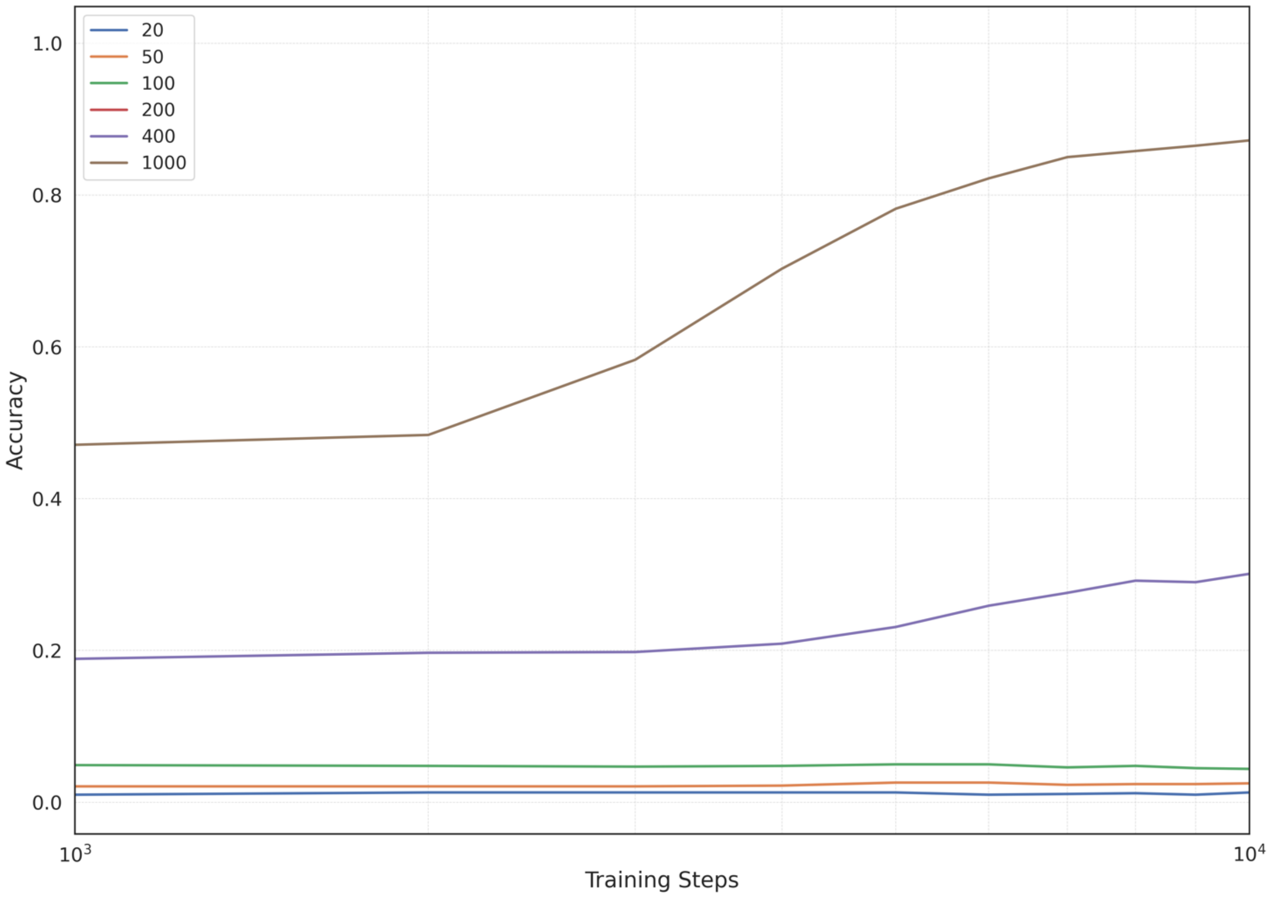
Attribute Values

- Refers to the possible range of values that attributes can take.
- For instance, if attribute age range is 30, the age attribute could range between different values, such as 0-29.
- Higher attribute value ranges led to a faster spike in Out-Of-Distribution (OOD) accuracy but delayed grokking.
- Lower attribute value ranges enable faster generalization, while larger ranges slow down the grokking process.



Entities

- The study was conducted by keeping the dataset size constant while varying the entities-to-attributes ratio.
- When the number of entities was less than 400, the model consistently failed to grok.
- This indicates that a minimum threshold of entities is required for the model to effectively identify patterns and relationships, enabling successful learning and generalization.



Summary

This study explores how various model architectures respond to reasoning tasks.

- GPT and LLaMA demonstrated grokking capabilities, LLaMA at a slower rate than GPT.
- RWKV and Mamba failed to grok the task entirely.

Also explored the dataset factors influencing grokking.

- Entities: Minimum threshold of entities is required for the model to grok.
- Attribute values: Lower attribute value ranges enabled faster generalization.

By exploring various configurations of these parameters within GPT, the study identifies optimal conditions for successful grokking, offering insights into the mechanisms driving reasoning and generalization in language models.

Future Research

Architectural and Training Innovations:
Develop methods to accelerate grokking while maintaining reasoning quality.

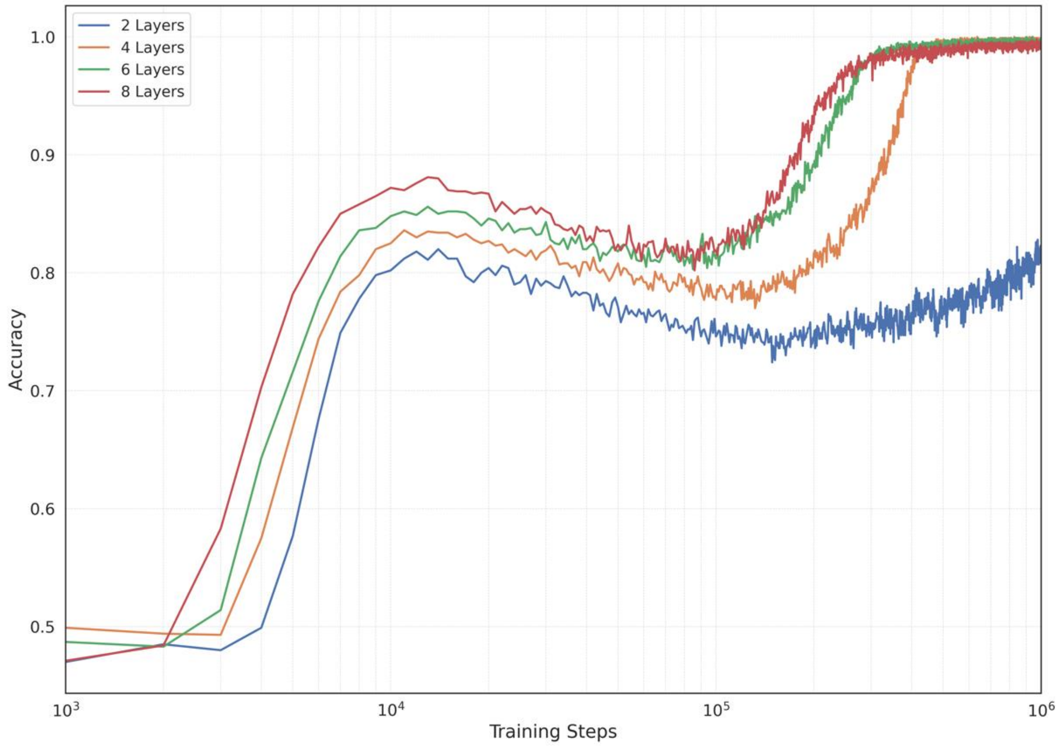
Mechanistic Interpretability:
Identify components or layers in LLMs that contribute to reasoning.

Scaling to Real-World Applications:
Adapt grokked models to effectively handle real-world datasets and benchmarks.

In-Context Learning on Grokked Systems:
Investigate the effectiveness of in-context learning for reasoning tasks in grokked models.

Additional Experiments: Model Size

- Experiments conducted using various configurations of GPT-2 (2, 4, 6, and 8 layers) demonstrated that smaller models grokked slower.
- As the number of layers increased, the rate of grokking slightly improved.
- Larger models exhibit better capacity to identify patterns and achieve generalization.



Thank you!