Documentation of the Language Models as Markov Chains

First Semester of 2025-2026

Branzea Malina Alexandra Niculae Cosmin Marian

Abstract

Large language models (LLMs) have achieved remarkable empirical success on a variety of tasks, yet a comprehensive theoretical understanding of their inference mechanisms remains elusive. In this work, we explore the formal connection between autoregressive LLMs and Markov chains, following recent theoretical advances that interpret LLM generation as a Markov process over finite state spaces. We empirically evaluate the in-context learning behavior of two transformer models—Gemma-2B and Llama-2-7B—on synthetic three-state Markov sources, measuring prediction accuracy as total variation distance between model outputs and true transition distributions. Our results demonstrate that both models improve their predictive accuracy with increasing context length and exhibit power-law scaling consistent with theoretical predictions.

1 Introduction

Large Language Models (LLMs) generate text autoregressively, predicting each new token from a fixed context window. Although highly effective, their internal inference mechanisms remain difficult to characterize. Recent theoretical work proposes viewing LLMs as Markov processes, where the model's next-token distribution acts as the transition function of a finite-state Markov chain. This perspective offers a probabilistic interpretation of in-context learning (ICL) and connects LLM behaviour to classical statistical estimation.

In this project, we empirically test whether real LLMs behave like Markov predictors on synthetic data. We construct a simple three-state Markov chain and evaluate two open-source models—Gemma-2B and Llama-2-7B—on their ability to infer transition probabilities from context alone. By measuring prediction error as total variation distance and studying how it scales with context length, we assess whether empirical behaviour aligns with theoretical expectations such as the $\mathcal{O}(N^{-1/2})$ rate commonly associated with ICL.

1.1 Related Work

Zekri et al. (2024) introduce a formal framework that treats autoregressive LLMs as Markov chains over sequences of tokens, and derive generalization bounds for pre-training and in-context learning under realistic assumptions. Their work is the main theoretical reference for this project.

Ahn et al. and Garg et al. investigate what transformers can learn in context and show that they are able to approximate classical probabilistic estimators, including those for Markov chains, based solely on observed examples in the prompt. They also study scaling laws, observing that the error in in-context learning often decays as $\mathcal{O}(N^{-1/2})$ as the context length increases.

Von Oswald et al. and Li et al. provide complementary theoretical perspectives by interpreting transformers as implicit Bayesian or gradient-based learners that update internal representations in a way analogous to statistical estimators. Overall, these works motivate our empirical study of LLMs on synthetic Markov data.

2 Approach

2.1 Prior Work

Our project builds on the recent paper *Large Language Models as Markov Chains* (Zekri et al., 2024), which provides the first formal proof that an autoregressive LLM with a finite context window can be modeled as a finite-state Markov chain. Earlier research typically treated LLMs as black-box systems, relying on empirical observations and i.i.d. assumptions that are unrealistic for natural language. Other works showed that transformers can learn simple probabilistic structures through in-context learning. However, Zekri et al. introduce the first rigorous mathematical connection

between LLMs and Markov processes, motivating our empirical validation.

2.2 Data and Exploratory Analysis

We use fully synthetic data generated from a controlled 3-state Markov chain. This approach guarantees access to exact ground-truth transition probabilities, unlike real text datasets. Our exploratory analysis includes:

- visualizing the transition matrix via heatmaps,
- inspecting sparsity and dominant transitions,
- verifying stationary distribution through long-run simulation.

These checks confirm that the generated sequences behave as a proper Markov process, ensuring a controlled evaluation environment.

2.3 Models and Computational Requirements

We evaluate two open-source transformer LLMs:

- **Gemma-2B** in fp16 precision,
- **Llama-2-7B** in 4-bit NF4 quantized form.

All experiments are run on a Google Colab T4 GPU (16GB VRAM). Gemma-2B fits easily in fp16 (5GB VRAM), whereas Llama-2-7B requires quantization to reduce memory usage from 14GB to 6–7GB. This setup allows a fair comparison between models of different capacities under identical computational constraints.

2.4 Dataset Preparation

The labeled data was loaded from train.csv, while test.csv was used for submission. We split the training data into 80 percent training and 20 percent validation. Features were extracted using custom Python functions, and training pipelines were built using scikit-learn.

2.5 Tools and Libraries

We use the following software components throughout the project:

PyTorch. Main deep learning framework used for tensor operations, GPU execution, and probability computations (e.g., `softmax`).

Hugging Face Ecosystem.

- **Transformers**: loading pretrained models (Gemma-2B, Llama-2-7B) and tokenizers.
- **HuggingFace Hub**: authentication via `login()` for gated model access.
- **Accelerate**: automatic GPU memory management (`device_map="auto"`).

BitsAndBytes. Used for 4-bit quantization of Llama-2-7B, reducing VRAM usage from ∼14GB to ∼6GB, enabling inference on a T4 GPU.

NumPy. Handles numerical computation, Markov chain generation, random sampling, and evaluation metrics such as Total Variation Distance.

Matplotlib & Seaborn. Used for visualization: log–log scaling-law plots and heatmaps of transition matrices.

Google Colab (T4 GPU). Execution environment providing a 16GB NVIDIA T4 GPU. Manual memory cleanup is performed using Python's `gc` module between model runs.

3 Model Evaluation

We evaluate performance using **Total Variation Distance (TVD)** between:

1. the true Markov transition distribution, and
2. the model's predicted distribution over tokens {0,1,2}.

We then examine how TVD scales with context length N, testing the theoretical prediction that error decreases as $\mathcal{O}(N_{\text{icl}}^{-1/2})$. Preliminary results show a reduction in TVD from ≈ 0.14 at $N = 10$ to ≈ 0.017 at $N = 100$, closely matching theoretical expectations. We compare Gemma and Llama to determine which model learns the Markov structure more efficiently.

4 Model Training

—

5 Conclusions and Future Work

—