Studying Transformer Behavior Under Markovian Input

by

Negar Soltan Mohammadi

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Management Sciences

Waterloo, Ontario, Canada, 2026

© Negar Soltan Mohammadi 2026

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Transformers have achieved remarkable success in modeling sequential data, yet a principled theoretical understanding of their behavior remains limited. A recent framework has analyzed transformers through the lens of first-order Markov chains, providing four theorems that characterize the loss landscape and the conditions under which global minima and bad local minima arise. However, the theoretical analysis in that work was restricted to first-order processes, leaving open questions about the behavior of transformers on higher-order Markovian data.

This thesis extends the theoretical framework to the case of second-order Markov chains. Specifically, all four theorems originally established for first-order chains are formally proven for second-order chains, thereby broadening the mathematical foundation for analyzing transformers on sequential data. Also, we show that these theorems are also true on another type of transformer architecture (attention-only transformers), given first and second-order Markovian input data. Furthermore, experimental evaluations demonstrate that the empirical learning dynamics for second-order chains align closely with the simplified second-order model proposed in prior work, confirming that the theoretical predictions hold in practice.

By closing the gap between first-order theory and second-order behavior, this study contributes to a deeper understanding of transformers' sequential modeling capabilities. The findings highlight the conditions under which transformers correctly capture second-order dependencies and provide new insights into their limitations and potential extensions for higher-order processes.

Acknowledgements

I am deeply grateful to my supervisor, Professor Saeed Ghadimi, for his constant support, guidance, and belief in me throughout this journey. His mentorship made this work possible.

My sincere thanks also go to Professor Jangho Yang and Professor Sibel Alumur Alev for their time, insight, and thoughtful feedback.

My heartfelt gratitude goes to my mom and dad for their unconditional love and support. Their belief in me has always been the foundation of my motivation and the reason I strive to do my best.

I would also like to thank my friends for their support and for all the fun we shared during my master's degree at Waterloo.

Table of Contents

Author's Declaration ii

Abstract iii

Acknowledgements iv

List of Figures viii

1 Introduction 1

1.1 Background and Motivation 1

1.2 Problem Statement and Research Gap 1

1.3 Research Objectives 2

1.4 Significance 2

1.5 Thesis Outline 3

2 Literature Review 4

2.1 Related Work 4

2.1.1 Introduction 4

2.2 Conclusion 8

2.2.1 The First-Order Blueprint 8

2.2.2 The High-Order Capacity 8

2.2.3 The Missing Piece (Our Focus) 8

3 Theoretical Results **9**

3.1 Introduction . . . 9

3.2 Preliminaries . . . 10

3.3 Part One: Second-order Markov chain . . . 13

3.3.1 Transformer Model Setup . . . 13

3.3.2 Lemma 1 . . . 14

3.3.3 Lemma 2 . . . 16

3.3.4 Theorem 1 (Global minimum) . . . 18

3.3.5 Theorem 2 (Bad local minimum) . . . 24

3.3.6 Theorem 3 (Global minimum) . . . 33

3.3.7 Theorem 4 (Saddle point) . . . 34

3.4 Part Two: Attention-only transformers . . . 36

3.4.1 Transformer Architecture . . . 36

3.4.2 First Order MC . . . 36

3.4.3 Lemma 1 . . . 39

3.4.4 Lemma 2 . . . 41

3.4.5 Theorem 1 (Global minimum) . . . 43

3.4.6 Theorem 2 (Bad local minimum) . . . 45

3.4.7 Second Order MC . . . 52

4 Observations **55**

4.1 Experiments on Learning Second-Order Markov Structure . . . 55

4.1.1 Summary of Prior Results (Section 4 of [9]) . . . 56

4.1.2 General Second-Order Markov Chains . . . 56

4.1.3 Stationary Distribution . . . 57

4.1.4 Experimental Configuration . . . 57

4.1.5 Empirical Results: Effect of Attention Window Size . . . 58

4.1.6 Discussion . . . 60

5 Conclusion and Future Work **61**

5.1 Conclusion 61

5.2 Future Work 62

References **64**

List of Figures

3.1 First order Markov chain with states 0 and 1, and transition probabilities p and q. 10

3.2 First-order view of a second-order binary Markov chain over states $\{00, 01, 10, 11\}$ with transition probabilities (p, q, s, t). 01 represents the state where $x_{n-1} = 0$ and $x_n = 1$. 10

4.1 Predicted probability $P(X_{t+1} = 1)$ under each second-order context $(X_{t-2}, X_{t-1}) \in \{00, 01, 10, 11\}$ for different memory window sizes. Each panel corresponds to one context (00, 01, 10, or 11). For short windows (particularly memory = 2), the curves closely track the appropriate entry in the second column of the transition matrix T, indicating that the model correctly learns the second-order transition probabilities. For larger windows, the predicted probabilities fluctuate substantially across positions and deviate from the true values. 59

Chapter 1

Introduction

1.1 Background and Motivation

Transformers have become the dominant architecture for sequence modeling tasks in natural language processing, computer vision, and related domains. Their effectiveness is often attributed to their ability to capture long-range dependencies through the self-attention mechanism. Despite their empirical success, however, the theoretical foundations of transformers remain only partially understood.

A promising direction has been to study transformers on simplified sequential models such as Markov chains. In particular, recent work analyzed transformers trained on first-order Markov chains and established formal theorems that characterize the loss landscape, including conditions for global minima, the emergence of bad local minima under weight tying, the role of saddle points, and the importance of depth in escaping suboptimal solutions. These findings offered an important step toward understanding transformers theoretically, yet they remain restricted to first-order dependencies.

1.2 Problem Statement and Research Gap

Real-world sequential data often exhibit dependencies beyond the immediate past. Language, biological sequences, and time series processes typically involve higher-order structure. The existing theoretical framework, however, does not address whether the results established for first-order Markov chains extend to more complex settings. This limitation

leaves a gap in our understanding of how transformers learn longer-memory dependencies and whether the same guarantees and challenges arise in higher-order processes.

1.3 Research Objectives

The objective of this thesis is to extend the theoretical analysis of transformers to second-order Markov chains and to empirically validate the theoretical predictions. Specifically, the thesis aims to:

- Provide formal proofs of four central theorems originally shown for first-order Markov chains in the second-order case.
- Analyze how architectural factors such as weight tying and depth influence the loss landscape when modeling second-order dependencies.
- Design and implement controlled experiments on synthetic second-order Markov data to evaluate whether empirical training outcomes align with theoretical expectations.
- Identify conditions under which transformers recover the correct transition dynamics, as well as situations where they fail by collapsing to stationary distributions.
- Provide formal proofs characterizing the global and local minima of the loss landscape in attention-only transformer models.

1.4 Significance

By bridging the gap between first-order and second-order analyses, this work advances the theoretical foundations of transformers in sequential learning. It validates that key structural phenomena generalize to richer dependencies, while also exposing new limitations and open questions. The findings contribute to both the theoretical literature on deep learning and the practical understanding of how transformer architectures can be adapted to capture more complex sequential dynamics.

1.5 Thesis Outline

The remainder of this thesis is organized as follows:

- Chapter 2 (*Literature Review*) surveys related work on transformers, Markov processes, and theoretical approaches to sequence modeling.
- Chapter 3 (*Theoretical Results*) provides the formal proofs of the extended theorems and discusses their implications.
- Chapter 4 (*Observations*) reports experimental results, comparing empirical training dynamics with theoretical predictions.
- Chapter 5 (*Conclusion*) summarizes the contributions, discusses implications, and outlines directions for future research.

Chapter 2

Literature Review

2.1 Related Work

2.1.1 Introduction

Transformers have become the core architecture behind today's most powerful language models, showing exceptional performance in tasks such as next-token prediction and a wide range of natural language processing applications. Despite this remarkable experimental success, our understanding of *how* transformers actually learn the patterns they capture remains limited. This gap has motivated researchers to investigate simplified settings—using controlled input distributions or reduced architectural designs—in order to gain clearer insight into the learning behaviors underlying these models. These efforts represent a growing theoretical interest in uncovering the fundamental principles that drive transformer learning. A recent line of work has taken this idea even further by asking whether transformers can be taught to perform in-context learning directly, without relying on massive pretraining. Garg et al. [5] explore this by training standard Transformers from scratch on very simple function classes, such as linear regression. Interestingly, they find that the model learns to solve new tasks by internally carrying out a procedure that resembles an iterative regression algorithm based on the prompt examples. In other words, the Transformer learns a small meta-learning routine on its own. This shows that in-context learning is not just a byproduct of scaling, but something that can emerge from much simpler, targeted training. A complementary perspective comes from work showing that in-context learning can be interpreted as a form of gradient-based meta-learning. Von Oswald et al. [14] demonstrate that even a single self-attention layer can be constructed

to perform one step of gradient descent on a simple regression loss. In this view, a trained Transformer effectively becomes a "mesa-optimizer," carrying out an inner optimization procedure through its forward pass. This result provides a clear mathematical explanation for how Transformers can adapt to new tasks from context alone, without parameter updates, reinforcing the idea that ICL emerges from implementing learned optimization dynamics inside the model. Another useful perspective comes from work examining what kinds of structures are already hidden inside Transformer embeddings. Hewitt and Manning [6] show that a simple linear probe can recover full syntactic parse trees from models like BERT and ELMo. By applying a learned linear map, the distances between word vectors align with their distances in the parse tree, revealing that hierarchical syntax is implicitly encoded in the embedding space. In other words, these models learn surprisingly rich structure, even though they were never trained to predict syntax directly. While these results offer valuable insights, a precise characterization of the NTP objective on sequential learning capabilities of transformers is not well understood.

Attention with Markov: A Framework for Principled Analysis of Transformers via Markov Chains

A foundational contribution to the theoretical understanding of transformers trained on sequential data is presented by Makkuva et al.[9] The central problem addressed in their work is the lack of a **precise characterization of the next-token prediction (NTP) objective** and its implications for the transformer's ability to learn from sequential structures. The authors formulate the fundamental research question:

> *How do transformers learn from sequential data?*

To make this question analytically tractable, the authors model the input as a **Markov chain**, enabling a principled examination of the interplay between (i) the statistical properties of the data distribution, (ii) the transformer architecture, (iii) the learned conditional distribution, and (iv) overall model performance. This Markovian abstraction provides a clean framework within which the loss landscape can be exactly characterized.

Theoretical Findings. The analysis focuses on the **loss landscape of single-layer transformers** trained on **first-order binary Markov chains** with transition probabilities $P_{01} = p$ and $P_{10} = q$. The authors derive several key theoretical results:

- **Global Minimum (Theorem 1).**

  - For any first-order Markov chain, the transformer's loss admits a **global minimum** θ_*.
  - At this optimum, the model's predicted next-token distribution matches the **true Markov kernel**. That is, the transformer correctly uses only the current token x_n to predict x_{n+1}, consistent with the memoryless property of first-order chains.
  - The loss at the global minimum equals the **entropy rate** of the chain:

$$L(\theta_*) = H(x_{n+1} \mid x_n).$$

- **Bad Local Minimum and the Role of Weight Tying (Theorem 2).**
  - The architecture—specifically **weight tying** between embedding and output projection—interacts with the Markov chain's **switching factor** $(p + q)$ to shape the loss landscape.
  - When weight tying is used and $p+q > 1$, the loss function exhibits a **bad local minimum** θ_π.
  - At this local minimum, the transformer disregards both past and current tokens, predicting instead the **stationary marginal distribution** π_1 of the chain. Consequently,

$$L(\theta_\pi) > L(\theta_*).$$

Experimental Findings. The empirical results closely corroborate the theoretical predictions and extend the analysis to deeper and higher-order architectures.

- **Confirmation of the Loss Landscape Structure.**
  - When the switching factor satisfies $p+q < 1$, the transformer reliably converges to the **global minimum**, recovering the true transition probabilities.
  - When $p+q > 1$ and weight tying is used, the model converges to the **bad local minimum**, predicting only the stationary distribution.
  - When weight tying is removed—even with $p + q > 1$—the transformer successfully learns the correct Markov kernel and converges to the global optimum.
- **Effect of Depth.**

  - For the problematic case of $p+q>1$ with weight tying, increasing the depth of the transformer ($\ell \geq 2$) allows the model to **escape the bad local minimum**. Deep transformers therefore overcome the failure mode observed in single-layer models.

- **Higher-Order Markov Chains.**
  - When the data is generated by a **higher-order** Markov chain, a standard transformer fails to learn the correct transition dynamics unless a **masking window** corresponding to the true order is explicitly imposed.
  - Unlike the first-order case, simply increasing depth does **not** solve the learning problem for higher-order dependencies.

Overall, Makkuva et al.[9] provide one of the first rigorous, principled characterizations of how transformer architectures behave under Markovian sequential structure. Their results form the theoretical and conceptual foundation for subsequent work—including this thesis—which investigates the geometry of transformer loss landscapes, global and local minima, and the behavior of attention mechanisms under structured sequential data.

A recent and closely related result comes from Liu et al. [7], who study whether Transformers truly require increasing depth to model higher-order Markov processes. Prior theory suggested that representing a kth-order Markov source should require a depth that grows logarithmically with k. Surprisingly, their work shows that this is not necessary in practice: even a fixed, shallow Transformer (e.g., three layers) can learn kth-order Markov data with high accuracy when trained long enough. They further prove that a three-layer, single-head Transformer is expressive enough to represent the in-context conditional distribution of kth-order sources, relying on attention mechanisms that collectively track the required k prior tokens.

This finding was particularly interesting for my thesis, because although the model can successfully learn high-order chains with constant depth, the authors do not analyze the *loss landscape* of this architecture. In contrast, Makkuva et al. [9] provide a detailed characterization of global and local minima for single-layer Transformers trained on first-order Markov data. My goal is to investigate whether these theoretical insights also extend to the constant-depth architecture studied in Liu et al. [7], and whether similar convergence behaviors and critical points arise when modeling higher-order Markov chains.

2.2 Conclusion

This chapter reviewed the journey of sequential modeling, from early models like RNNs and LSTMs—which struggled with long-term memory—to the powerful, attention-based Transformers that dominate today.

While we see incredible performance from large language models (LLMs) every day, we still lack a complete understanding of how transformers learn the patterns in sequential data.

Recent theoretical work has made significant progress, primarily by studying simplified cases:

2.2.1 The First-Order Blueprint

Makkuva et al. (2024) gave us a clear map of the learning landscape for basic transformers on first-order Markov chains (where the future only depends on the immediate present). They found that architecture choices (like "weight tying") can make the model get stuck in a "bad" solution, failing to learn the true sequence rules.

2.2.2 The High-Order Capacity

Other work (Liu et al., 2024) confirmed that transformers, even shallow ones, are powerful enough to represent the rules of much more complex, higher-order Markov chains (where the future depends on multiple past steps).

2.2.3 The Missing Piece (Our Focus)

There's a critical gap between what we know they can represent (higher-order chains) and how they actually learn it (their loss landscape). The detailed theoretical map we have for the simple first-order case doesn't exist for the complex, higher-order chains that better resemble real-world data (like language).

This thesis will fill that gap. We aim to extend the rigorous theoretical analysis of the learning landscape—including finding the "bad" local solutions—to second-order Markov chains. By doing this, we connect the dots between the expressive power of the transformer architecture and the practical challenges of optimization, giving us a much clearer picture of how these models handle the complexity of real-world sequences.

Chapter 3

Theoretical Results

3.1 Introduction

In this chapter, we explore the main theoretical contributions of the thesis. Our work is divided into two parts. First, we build on the first-order analysis from previous research and extend it to second-order results. Specifically, we establish second-order analogues for four key findings:
(i) the existence of global minima that accurately recover the true second-order kernel,
(ii) the presence of bad local minima when weight tying is used,
(iii) how these bad local minima turn into saddle points once weight tying is removed, and
(iv) the influence of network depth in helping the model escape suboptimal stationary points.

Second, we investigate whether these results also apply to a different architecture—the attention-only transformer. We analyze how the loss behaves for both first-order and second-order Markov chain data and check if the patterns we observed in the first part still hold.

We start by reviewing the necessary background and then move on to state and prove the lemmas and theorems.

3.2 Preliminaries

Throughout this work we consider binary sequences for the input data sequence $\{x_n\}_{n=1}^N \in \{0,1\}^N$ generated by either a first-order Markov chain or a second-order Markov chain with transition probabilities same as 3.1 for the first-order chain and 3.2 for the second order chain.

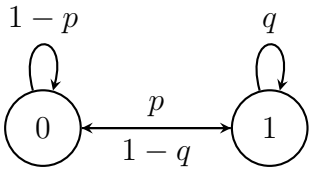


Figure 3.1: First order Markov chain with states 0 and 1, and transition probabilities p and q.

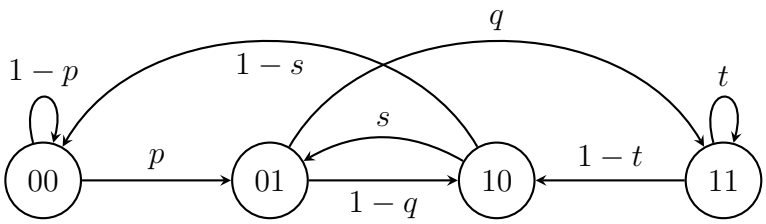


Figure 3.2: First-order view of a second-order binary Markov chain over states $\{00, 01, 10, 11\}$ with transition probabilities (p, q, s, t). 01 represents the state where $x_{n-1} = 0$ and $x_n = 1$.

As the input is binary for the first-order case, we have $\mathbb{P}(x_{n+1} = 1 \mid x_n) + \mathbb{P}(x_{n+1} = 0 \mid x_n) = 1$, and for the second-order case, $\mathbb{P}(x_{n+1} = 1 \mid x_n, x_{n-1}) + \mathbb{P}(x_{n+1} = 0 \mid x_n, x_{n-1}) = 1$. Also for the first-order case, the probability of getting 1 on the next token can be written as (3.1)

$$\mathbb{P}(x_{n+1} = 1 \mid x_n) = x_n(1 - p - q) + p \tag{3.1}$$

Similarly, for the second-order case the probability of getting 1 on the next token can be written as (3.2)

$$\mathbb{P}(x_{n+1} = 1 \mid x_n, x_{n-1}) = x_{n-1}\left(x_n t + (1 - x_n)s\right) + (1 - x_{n-1})\left(x_n q + (1 - x_n)p\right) \tag{3.2}$$

Let the system be said to be in steady-state when the following condition is satisfied:

$$\pi_i = \sum_j \pi_j P_{ji}, \quad \forall i$$

where π_i represents the steady-state probability of being in state i.
Where:

- π_i is the steady-state probability of state i,
- P_{ji} is the transition probability from state j to state i,
- The sum runs over all possible states j of the Markov Chain.

Additionally, the steady-state probabilities must satisfy the normalization condition:

$$\sum_i \pi_i = 1$$

Therefore, the steady state probability for the first-order chain can be written as (3.3)

$$\begin{aligned} \pi_0 &= \frac{q}{p+q} \\ \pi_1 &= \frac{p}{p+q} \end{aligned} \tag{3.3}$$

And the steady state probability for the second-order chain can be written as (3.4)

$$\begin{aligned} \pi_{00} &= \frac{st - s - t + 1}{pq - 2pt + 2p + st - s - t + 1} \\ \pi_{01} &= \frac{-pt + p}{pq - 2pt + 2p + st - s - t + 1} \\ \pi_{10} &= \frac{-pt + p}{pq - 2pt + 2p + st - s - t + 1} \\ \pi_{11} &= \frac{pq}{pq - 2pt + 2p + st - s - t + 1} \end{aligned} \tag{3.4}$$

We can also calculate π_1 and π_0 as follows:

$$\begin{aligned} \pi_0 = \pi_{00} + \pi_{10} &= \frac{st - s - t + 1 - pt + p}{pq - 2pt + 2p + st - s - t + 1} \\ \pi_1 = \pi_{11} + \pi_{01} &= \frac{-pt + p + pq}{pq - 2pt + 2p + st - s - t + 1} \end{aligned} \tag{3.5}$$

We build on the transformer architecture proposed in [13] and modify this in each part of the thesis.

$$\boldsymbol{x_n} = x_n e + p_n \in \mathbb{R}^d \quad \text{(Uni-embedding)} \tag{3.6}$$

$$\boldsymbol{y_n} = \boldsymbol{x_n} + W_0 \sum_{i \in [n]} \text{att}_{n,i} \cdot W_V x_i \in \mathbb{R}^d \quad \text{(Attention)} \tag{3.7}$$

$$\boldsymbol{z_n} = \boldsymbol{y_n} + W_2 \text{ReLU}(W_1 y_n) \in \mathbb{R}^d \quad \text{(FF)} \tag{3.8}$$

$$\text{logit}_n = \langle a, \boldsymbol{z_n} \rangle + b \in \mathbb{R} \quad \text{(Linear)} \tag{3.9}$$

$$f_\theta(x_1^n) \triangleq \mathbb{P}_\theta(x_{n+1} = 1 \mid x_1^n) = \sigma(\text{logit}_n) \in [0, 1] \quad \text{(Prediction)} \tag{3.10}$$

Embedding: The binary input values $x_n = 1$ and $x_n = 0$ are mapped to their respective embeddings, e and 0, within a d-dimensional space $\mathbb{R}^d$. Here, d denotes the embedding dimension. Additionally, each token receives a positional encoding p_n, also within $\mathbb{R}^d$, which varies depending on the token's position in the sequence. The final input representation x_n is obtained by adding these two embeddings together.

Attention Mechanism: In the attention mechanism, a query is compared to a series of key-value pairs to produce an output. The output $y_n \in \mathbb{R}^d$ is calculated by taking the weighted sum of the values $v_i \triangleq W_V x_i$, where the weights $att_{n,i}$ are determined by how well the query vector $q_n \triangleq W_Q x_n$ aligns with the key vectors $k_i \triangleq W_K x_i$. The weight for each key is computed as:

$$att_{n,i} \triangleq \text{softmax}\left(\frac{\langle q_n, k_i \rangle}{\sqrt{d}}\right),$$

with the weight matrices $W_K, Q, V \in \mathbb{R}^{m \times d}$, and $W_O \in \mathbb{R}^{d \times m}$ as the projection matrix. In multi-head attention, this process is repeated across multiple heads, and the resulting outputs are summed together.

Feed-Forward Layer (FF): The feed-forward component consists of a skip connection and a hidden layer with a ReLU activation. This layer uses two weight matrices, $W_2 \in \mathbb{R}^{d\times r}$ and $W_1 \in \mathbb{R}^{r\times d}$, and is applied independently to each token's embedding $y_n \in \mathbb{R}^d$, producing an output $z_n \in \mathbb{R}^d$ with the same dimensionality.

Linear Transformation: The linear layer maps the output from the feed-forward network, z_n, to a scalar value $\text{logit}_n \in \mathbb{R}$, using a weight vector $a \in \mathbb{R}^d$ and a bias term $b \in \mathbb{R}$.

Prediction: The scalar output, logit_n, is passed through a sigmoid function to yield a probability for predicting the next token. For a binary vocabulary, the probability of the next token being 1 is given by:

$$f_\theta(x_1^n) \triangleq \mathbb{P}_\theta(x_{n+1} = 1 \mid x_1^n) = \sigma(\text{logit}_n) \in [0, 1].$$

For vocabularies larger than two tokens, the logits would be processed by a softmax function, but for binary cases, the sigmoid function is sufficient. The input tokens undergo multiple layers of multi-headed attention and feed-forward operations, alternating between these layers, to compute the final logits.

Finally, the Transformer's parameters, $\theta \triangleq (e, \{p_n\}_{n=1}^N, b, a) \in \mathbb{R}^d$, are optimized by minimizing the cross-entropy loss for predicting the next token:

$$L(\theta) \triangleq -\frac{1}{N} \sum_{n\in[N]} \mathbb{E}_{n+1} \left[x_{n+1} \cdot \log f_\theta(x_1^n) + (1 - x_{n+1}) \cdot \log(1 - f_\theta(x_1^n))\right] \tag{3.11}$$

3.3 Part One: Second-order Markov chain

In this section, we work with second-order Markovian data as input, using a similar architecture to the one in equations 3.6 to 3.10, with just a few adjustments. We then show that the theoretical results from [13] also apply to this case, building on their work, which was originally focused on first-order Markov chains.

3.3.1 Transformer Model Setup

In this part, we use a different attention layer than 3.7.:

$$\boldsymbol{y_n} = \boldsymbol{x_n} + \delta \boldsymbol{x_{n-1}} \quad \text{(Attention)} \tag{3.12}$$

We use the exact same loss function as 3.11.

3.3.2 Lemma 1

Lemma 1 (Loss as KL divergence). Let the input sequence be

$$\{x_n\}_{n=1}^N \sim (\pi(p,q,s,t), P(p,q,s,t), \mathbf{P}(p,q,s,t))$$

for some fixed $(p,q,s,t) \in (0,1)^2$, $\boldsymbol{\theta} = (e, \{p_n\}_{n=1}^N, \ldots, b, a) \in \mathbb{R}^D$ be the full list of the transformer parameters, and $L(\boldsymbol{\theta})$ be the corresponding cross-entropy loss in 3.11. Then the loss function $L(\cdot)$ is equivalent to the KL divergence between the Markov kernel and the predicted distribution, i.e.,

$$L(\boldsymbol{\theta}) = \frac{1}{N} \sum_{n \in [N]} \mathbb{E}_{x_1^n} \left[D_{KL}(\mathbb{P}(x_{n+1} = \cdot \mid x_n, x_{n-1}) \parallel \mathbb{P}_{\boldsymbol{\theta}}(x_{n+1} = \cdot \mid x_1^n)) \right] + H(x_{n+1} \mid x_n, x_{n-1}), \tag{3.13}$$

where $D_{KL}(P \parallel Q)$ is the KL divergence between two distributions P and Q, and $H(x_{n+1} \mid x_n)$ is the entropy rate of the Markov chain.

Remark. Consequently, 3.13 highlights that any parameter $\boldsymbol{\theta}$ with the predicted probability

$$f_{\boldsymbol{\theta}}(x_1^n) = \mathbb{P}(x_{n+1} = 1 \mid x_n)$$

is a global minimum for the loss L as $D_{KL}(\cdot \parallel \cdot) \geq 0$. We utilize this fact in the proof of Theorem 1 later.

Proof. Consider the loss function $L(\cdot)$ given in Eq. (3.1). We can rewrite it as follows:

$$
\begin{aligned}
L(\boldsymbol{\theta}) &= -\frac{1}{N}\sum_{n\in[N]} \mathbb{E}_{x_1^{n+1}}\Big[x_{n+1}\cdot\log f_{\boldsymbol{\theta}}(x_1^n) + (1-x_{n+1})\cdot\log(1-f_{\boldsymbol{\theta}}(x_1^n))\Big] \\
&= -\frac{1}{N}\sum_{n\in[N]} \mathbb{E}_{x_1^n}\Big[\mathbb{E}_{x_{n+1}|x_1^n}\Big(x_{n+1}\cdot\log f_{\boldsymbol{\theta}}(x_1^n) + (1-x_{n+1})\cdot\log(1-f_{\boldsymbol{\theta}}(x_1^n))\Big)\Big] \\
&= -\frac{1}{N}\sum_{n\in[N]} \mathbb{E}_{x_1^n}\Big[\mathbb{P}(x_{n+1}=1\mid x_n,x_{n-1})\log\frac{f_{\boldsymbol{\theta}}(x_1^n)}{\mathbb{P}(x_{n+1}=1\mid x_n,x_{n-1})} \\
&\quad + \mathbb{P}(x_{n+1}=0\mid x_n,x_{n-1})\log\frac{1-f_{\boldsymbol{\theta}}(x_1^n)}{\mathbb{P}(x_{n+1}=0\mid x_n,x_{n-1})}\Big] \\
&\quad - \frac{1}{N}\sum_{n\in[N]} \mathbb{E}_{x_1^n}\Big[\mathbb{P}(x_{n+1}=1\mid x_n,x_{n-1})\log\frac{1}{\mathbb{P}(x_{n+1}=1\mid x_n,x_{n-1})} \\
&\quad + \mathbb{P}(x_{n+1}=0\mid x_n,x_{n-1})\log\frac{1}{\mathbb{P}(x_{n+1}=0\mid x_n,x_{n-1})}\Big] \\
&= -\frac{1}{N}\sum_{n\in[N]} \mathbb{E}_{x_1^n}\Big[\mathbb{P}(x_{n+1}=1\mid x_n,x_{n-1})\log\frac{f_{\boldsymbol{\theta}}(x_1^n)}{\mathbb{P}(x_{n+1}=1\mid x_n,x_{n-1})} \\
&\quad + \mathbb{P}(x_{n+1}=0\mid x_n,x_{n-1})\log\frac{1-f_{\boldsymbol{\theta}}(x_1^n)}{\mathbb{P}(x_{n+1}=0\mid x_n,x_{n-1})}\Big] \\
&\quad - \mathbb{E}_{x_{n,n-1}}\Big[\mathbb{E}_{x_1^{n-2}|x_n,x_{n-1}}\Big[\mathbb{P}(x_{n+1}=1\mid x_n,x_{n-1})\log\frac{1}{\mathbb{P}(x_{n+1}=1\mid x_n,x_{n-1})} \\
&\quad + \mathbb{P}(x_{n+1}=0\mid x_n,x_{n-1})\log\frac{1}{\mathbb{P}(x_{n+1}=0\mid x_n,x_{n-1})}\Big]\Big] \\
&= -\frac{1}{N}\sum_{n\in[N]} \mathbb{E}_{x_1^n}\Big[\mathbb{P}(x_{n+1}=1\mid x_n,x_{n-1})\log\frac{f_{\boldsymbol{\theta}}(x_1^n)}{\mathbb{P}(x_{n+1}=1\mid x_n,x_{n-1})} \\
&\quad + \mathbb{P}(x_{n+1}=0\mid x_n,x_{n-1})\log\frac{1-f_{\boldsymbol{\theta}}(x_1^n)}{\mathbb{P}(x_{n+1}=0\mid x_n,x_{n-1})}\Big] \\
&\quad - \mathbb{E}_{x_{n,n-1}}\Big[\mathbb{P}(x_{n+1}=1\mid x_n,x_{n-1})\log\frac{1}{\mathbb{P}(x_{n+1}=1\mid x_n,x_{n-1})} \\
&\quad + \mathbb{P}(x_{n+1}=0\mid x_n,x_{n-1})\log\frac{1}{\mathbb{P}(x_{n+1}=0\mid x_n,x_{n-1})}\Big].
\end{aligned}
$$

We knew that:

$$f_{\boldsymbol{\theta}}(x_1^n) = \mathbb{P}_{\boldsymbol{\theta}}(x_{n+1}=1\mid x_n,x_{n-1}) \quad \text{according to the architecture.}$$

Also, we can say:

$$\begin{aligned}\mathbb{P}(x_{n+1}=1 \mid x_n, x_{n-1}) \log \frac{1}{\mathbb{P}(x_{n+1}=1 \mid x_n, x_{n-1})} \\ + \mathbb{P}(x_{n+1}=0 \mid x_n, x_{n-1}) \log \frac{1}{\mathbb{P}(x_{n+1}=0 \mid x_n, x_{n-1})} \\ = \mathbb{H}(x_{n+1} \mid x_n, x_{n-1}).\end{aligned}$$

And:

$$\begin{aligned}\mathbb{P}(x_{n+1}=1 \mid x_n, x_{n-1}) \log \frac{f_{\boldsymbol{\theta}}(x_1^n)}{\mathbb{P}(x_{n+1}=1 \mid x_n, x_{n-1})} \\ + \mathbb{P}(x_{n+1}=0 \mid x_n, x_{n-1}) \log \frac{1-f_{\boldsymbol{\theta}}(x_1^n)}{\mathbb{P}(x_{n+1}=0 \mid x_n, x_{n-1})} \\ = -D_{KL}\Big(\mathbb{P}(x_{n+1} \mid x_n, x_{n-1}) \Big\| \mathbb{P}_{\boldsymbol{\theta}}(x_{n+1}=\cdot \mid x_1^n)\Big).\end{aligned}$$

Therefore:

$$L(\boldsymbol{\theta}) = \frac{1}{N} \sum_{n\in[N]} \mathbb{E}_{x_1^n}\Big[D_{KL}\Big(\mathbb{P}(x_{n+1}=\cdot \mid x_n, x_{n-1}) \Big\| \mathbb{P}_{\boldsymbol{\theta}}(x_{n+1}=\cdot \mid x_1^n)\Big)\Big] + \mathbb{H}(x_{n+1} \mid x_n, x_{n-1}).$$

3.3.3 Lemma 2

Lemma 2 (Gradient computation). Consider the same data and parameter setting as in Lemma 1 and let $L(\boldsymbol{\theta})$ be the cross-entropy loss in 3.11. Then for any parameter $w \in \boldsymbol{\theta}$,

$$\begin{aligned}\nabla_w L(\boldsymbol{\theta}) &= -\frac{1}{N} \sum_{n\in[N]} \mathbb{E}_{x_1^{n+1}}\big[(x_{n+1} - f_{\boldsymbol{\theta}}(x_1^n)) \cdot \nabla_w (a^\top z_n + b)\big] \\ &= -\frac{1}{N} \sum_{n\in[N]} \mathbb{E}_{x_1^n}\big[(\mathbb{P}(x_{n+1}=1 \mid x_n, x_{n-1}) - f_{\boldsymbol{\theta}}(x_1^n)) \cdot \nabla_w (a^\top z_n + b)\big].\end{aligned} \tag{3.14}$$

Remark. 3.14 highlights that any parameter $\boldsymbol{\theta}$ with the predicted probability $f_{\boldsymbol{\theta}}(x_1^n) = \mathbb{P}(x_{n+1}=1 \mid x_n, x_{n-1})$ is also a stationary point for the loss L. We utilize this fact in the proof of Theorem 1 later.

Proof. Let θ_j of $\boldsymbol{\theta} \in \mathbb{R}^D$:

$$\begin{aligned}\frac{\partial}{\partial \tilde{\theta}_j} L(\tilde{\theta}) &= -\frac{1}{N} \sum_{n \in [N]} \mathbb{E}_{x_1^{n+1}} \Big[\big(x_{n+1} - f_{\tilde{\theta}}(x_1^n) \big) \cdot \frac{\partial}{\partial \tilde{\theta}_j} \big(a^\top z_n + b \big) \Big] \\ &= -\frac{1}{N} \sum_{n \in [N]} \mathbb{E}_{x_1^n} \Big[\big(\mathbb{P}(x_{n+1} = 1 \mid x_n, x_{n-1}) - f_{\tilde{\theta}}(x_1^n) \big) \cdot \frac{\partial}{\partial \tilde{\theta}_j} \big(a^\top z_n + b \big) \Big].\end{aligned}$$

Also we had:

$$L(\tilde{\theta}) = -\frac{1}{N} \sum_{n \in [N]} \mathbb{E}_{x_1^{n+1}} \big[x_{n+1} \cdot \log f_{\tilde{\theta}}(x_1^n) + (1 - x_{n+1}) \cdot \log(1 - f_{\tilde{\theta}}(x_1^n)) \big],$$

which implies that

$$\begin{aligned}\frac{\partial}{\partial \tilde{\theta}_j} L(\tilde{\theta}) &= -\frac{1}{N} \sum_{n \in [N]} \mathbb{E}_{x_1^{n+1}} \Big[x_{n+1} \cdot \frac{\partial}{\partial \tilde{\theta}_j} \log f_{\tilde{\theta}}(x_1^n) + (1 - x_{n+1}) \cdot \frac{\partial}{\partial \tilde{\theta}_j} \log(1 - f_{\tilde{\theta}}(x_1^n)) \Big] \\ &= -\frac{1}{N} \sum_{n \in [N]} \mathbb{E}_{x_1^{n+1}} \Big[x_{n+1} \cdot \frac{1}{f_{\tilde{\theta}}(x_1^n)} \frac{\partial}{\partial \tilde{\theta}_j} f_{\tilde{\theta}}(x_1^n) + (1 - x_{n+1}) \cdot \frac{1}{1 - f_{\tilde{\theta}}(x_1^n)} \frac{\partial}{\partial \tilde{\theta}_j} (1 - f_{\tilde{\theta}}(x_1^n)) \Big].\end{aligned}$$

Since $f_{\tilde{\theta}}(x_1^n) = \sigma(a^\top z_n + b)$, we first note that the derivative of σ is given by

$$\sigma'(z) = \frac{e^{-z}}{(1 + e^{-z})^2} = \sigma(z)(1 - \sigma(z)).$$

Hence, the derivative $\frac{\partial}{\partial \tilde{\theta}_j} f_{\tilde{\theta}}(x_1^n)$ can be written as

$$\begin{aligned}\frac{\partial}{\partial \tilde{\theta}_j} f_{\tilde{\theta}}(x_1^n) &= \frac{\partial}{\partial \tilde{\theta}_j} \sigma(a^\top z_n + b) \\ &= \sigma(a^\top z_n + b) \big[1 - \sigma(a^\top z_n + b) \big] \cdot \frac{\partial}{\partial \tilde{\theta}_j} \big(a^\top z_n + b \big) \\ &= f_{\tilde{\theta}}(x_1^n) \big[1 - f_{\tilde{\theta}}(x_1^n) \big] \cdot \frac{\partial}{\partial \tilde{\theta}_j} \big(a^\top z_n + b \big).\end{aligned}$$

Plugging this into the above expression, we have

$$\begin{aligned}
\frac{\partial}{\partial \tilde{\theta}_j} L(\tilde{\theta}) &= -\frac{1}{N} \sum_{n\in[N]} \mathbb{E}_{x_1^{n+1}} \Big[x_{n+1}\big(1 - f_{\tilde{\theta}}(x_1^n)\big) \frac{\partial}{\partial \tilde{\theta}_j} \big(a^\top z_n + b\big) \\
&\qquad - (1 - x_{n+1}) f_{\tilde{\theta}}(x_1^n) \frac{\partial}{\partial \tilde{\theta}_j} \big(a^\top z_n + b\big) \Big] \\
&= -\frac{1}{N} \sum_{n\in[N]} \mathbb{E}_{x_1^{n+1}} \Big[\big(x_{n+1} - f_{\tilde{\theta}}(x_1^n)\big) \frac{\partial}{\partial \tilde{\theta}_j} \big(a^\top z_n + b\big) \Big] \\
&= -\frac{1}{N} \sum_{n\in[N]} \mathbb{E}_{x_1^{n}} \Big[\mathbb{E}_{x_{n+1}|x_1^n} \big[x_{n+1} - f_{\tilde{\theta}}(x_1^n) \big] \frac{\partial}{\partial \tilde{\theta}_j} \big(a^\top z_n + b\big) \Big] \\
&= -\frac{1}{N} \sum_{n\in[N]} \mathbb{E}_{x_1^{n}} \Big[\big(\mathbb{P}(x_{n+1} = 1 \mid x_n, x_{n-1}) - f_{\tilde{\theta}}(x_1^n) \big) \frac{\partial}{\partial \tilde{\theta}_j} \big(a^\top z_n + b\big) \Big],
\end{aligned}$$

and we are done.

3.3.4 Theorem 1 (Global minimum)

Theorem 1 (Global minimum).
Let the input sequence be $\{x_n\}_{n=1}^N \sim (\pi(p,q,s,t), P(p,q,s,t))$ for some fixed $(p,q,s,t) \in (0,1)^2$. Then for all (p,q,s,t), there exists a $\theta_* \in \mathbb{R}^{D-d}$ with an explicit construction such that it is a global minimum for the population loss $L(\cdot)$ in Eq. (3.1), i.e.,

1. $L(\theta) \geq L(\theta_*)$ for all $\theta \in \mathbb{R}^{D-d}$
2. $\mathbb{P}_{\theta_*}(x_{n+1} = 1 \mid x_1^n) = \mathbb{P}(x_{n+1} = 1 \mid x_n, x_{n-1})$, the Markov kernel.
3. $L(\theta_*) = H(x_{n+1} \mid x_n, x_{n-1})$, the entropy rate of the Markov chain.
4. $\nabla L(\theta_*) = 0$, i.e., θ_* is a stationary point.

Proof. We assume that we use weight tying, i.e., the list of parameters $\theta = (e = a, \{p_n\}_{n=1}^N, b) \in \mathbb{R}^{D-d}$. Thus in view of Lemma 1 and Lemma 2, it follows that any θ satisfying

$$f_\theta(x_1^n) = \mathbb{P}(x_{n+1} = 1 \mid x_n, x_{n-1})$$

is a global minimum with loss equalling the entropy rate, and is a stationary point. Hence it suffices to construct such a θ.

To build our intuition towards designing $\theta_\star$, recall that the Markov kernel $\mathbb{P}(x_{n+1} = 1 \mid x_n, x_{n-1})$ can be succinctly written as

$$\mathbb{P}(x_{n+1} = 1 \mid x_n, x_{n-1}) = x_{n-1}\left(x_n t + (1 - x_n)s\right) + (1 - x_{n-1})\left(x_n q + (1 - x_n)p\right)$$

To ensure that $f_\theta(x_1^n) = x_{n-1}\left(x_n t + (1 - x_n)s\right) + (1 - x_{n-1})\left(x_n q + (1 - x_n)p\right)$, it suffices for the transformer to utilize only the information from the current symbol x_n and the symbol before that x_{n-1} and ignore the past x_1^{n-2}. In view of the transformer architecture, if we set $w_2 = 0$, then:

$$x_n = x_n e + \rho_n \in \mathbb{R}^d \quad \text{(embedding)}$$

$$y_n = x_n + \delta x_{n-1} \in \mathbb{R}^d \quad \text{(attention)}$$

$$z_n = y_n + w_2 \mathrm{ReLU}(w_1 y_n) \in \mathbb{R}^d \quad \text{(FF layer)}$$

$$\mathrm{logit}_n = \langle a, z_n \rangle + b \in \mathbb{R} \quad \text{(linear layer)}$$

$$f_{\boldsymbol{\theta}}(x_1^n) \triangleq \mathbb{P}_{\boldsymbol{\theta}}(x_{n+1} = 1 \mid x_1^n) = \sigma(\mathrm{logit}_n) \in [0, 1] \quad \text{(prediction)}$$

Therefore:

$$z_n = y_n = x_n + \delta x_{n-1} = x_n e + p_n + \delta x_{n-1} + \delta p_{n-1}.$$

$$\mathrm{logit}_n = \langle a, z_n \rangle + b = \langle e, x_n \rangle + \delta \langle e, x_{n-1} \rangle + b$$

$$\sigma[\langle e, x_n \rangle + \delta \langle e, x_{n-1} \rangle + b] = x_{n-1}\left(x_n t + (1 - x_n)s\right) + (1 - x_{n-1})\left(x_n q + (1 - x_n)p\right)$$

$$\langle e, x_n \rangle + \delta \langle e, x_{n-1} \rangle + b = -\log\left(\frac{1}{x_{n-1}\left(x_n t + (1 - x_n)s\right) + (1 - x_{n-1})\left(x_n q + (1 - x_n)p\right)} - 1\right)$$

Therefore for every state we have:

x_{n-1}	x_n	**Relation**
0	0	$\langle e, p_n\rangle + \delta\langle e, p_{n-1}\rangle + b = -\log\left(\frac{1}{p} - 1\right)$
0	1	$\langle e, p_n\rangle + \lvert e\rvert^2 + b = -\log\left(\frac{1}{q} - 1\right)$
1	0	$\lvert e\rvert^2 + \delta\langle e, p_{n-1}\rangle + b = -\log\left(\frac{1}{s} - 1\right)$
1	1	$2\lvert e\rvert^2 + \langle e, p_n\rangle + \delta\langle e, p_{n-1}\rangle + b = -\log\left(\frac{1}{t} - 1\right)$

According to the table we have:

$$\Rightarrow 2\|e\|^2 = -\log\left(\frac{1}{t} - 1\right) + \log\left(\frac{1}{p} - 1\right)$$

$$b + \langle e, p_n\rangle = -\log\left(\frac{1}{q} - 1\right) - \frac{1}{2}\left[-\log\left(\frac{1}{t} - 1\right) + \log\left(\frac{1}{p} - 1\right)\right]$$

$$b + \delta\langle e, p_{n-1}\rangle = -\log\left(\frac{1}{s} - 1\right) - \frac{1}{2}\left[-\log\left(\frac{1}{t} - 1\right) + \log\left(\frac{1}{p} - 1\right)\right]$$

In order to make these equations hold true we should:

$$\|e\|^2 = \log\left(\frac{(1-p)t}{p(1-t)}\right) \quad \text{(should be +)}$$

$$\Rightarrow \frac{t - pt}{p - pt} \geq 1 \quad \Rightarrow t - pt \geq p - pt \quad \Rightarrow t \geq p \quad \text{(condition)}$$

$$\langle e, p_n\rangle + b = \log\left(\frac{q}{1-q} \cdot \sqrt{\frac{(1-t)p}{(1-p)t}}\right)$$

$$\delta\langle e, p_{n-1}\rangle + b = \log\left(\frac{s}{1-s} \cdot \sqrt{\frac{(1-t)p}{(1-p)t}}\right)$$

Case 1: If $\{p_n = 0\}_{n=1}^{N}$, $e = a$,

$$b_* = \log\left(\frac{q}{1-q} \cdot \sqrt{\frac{(1-t)p}{(1-p)t}}\right) = \log\left(\frac{s}{1-s} \cdot \sqrt{\frac{(1-t)p}{(1-p)t}}\right) \quad \Rightarrow q = s$$

$$e_* = a_* = \sqrt{\frac{1}{d}\log\left(\frac{(1-p)t}{p(1-t)}\right)}$$

Case 2: If $\{p_n = p_{n-1} = p_{n-2} = \dots\}$, $e = a$,

$$e_* = a_* = \sqrt{\frac{1}{d}\log\left(\frac{(1-p)t}{p(1-t)}\right)}$$

$$\delta\left[\log\left(\frac{q}{1-q}\cdot\sqrt{\frac{(1-t)p}{(1-p)t}}\right) - b\right] + b = \log\left(\frac{s}{1-s}\cdot\sqrt{\frac{(1-t)p}{(1-p)t}}\right)$$

$$b_* = \frac{1}{1-\delta}\left(\log\left(\frac{s}{1-s}\cdot\sqrt{\frac{(1-t)p}{(1-p)t}}\right) - \delta\log\left(\frac{q}{1-q}\cdot\sqrt{\frac{(1-t)p}{(1-p)t}}\right)\right)$$

Therefore, for the case $t \geq p$, Theorem 1 holds, given that θ_* is defined as follows:

$$\theta_* = \big(e = a = e_* = a_*,\ p_n,\ W_O = 0,\ W_{K,Q,V},\ W_1,\ W_2 = 0,\ b = b_*\big).$$

Now if $t < p$ given $e = a = 1$ and $p_n = -1\frac{1}{2}$:

$$f_\theta(x_1^n) = \sigma(\text{logit}_n) = \sigma(\langle a, (\boldsymbol{y}_n + \boldsymbol{W}_2\text{ReLU}(\boldsymbol{W}_1\boldsymbol{y}_n))\rangle)$$

$$= x_{n-1}\left(x_n t + (1-x_n)s\right) + (1-x_{n-1})\left(x_n q + (1-x_n)p\right)$$

$$\text{Let } p_n = -\frac{1}{2}, \quad e = a = 1$$

$$x_n = ex_n + p_n = ex_n - \frac{1}{2} = \begin{cases} -\frac{1}{2} & \text{if } x_n = 0 \\ \frac{1}{2} & \text{if } x_n = 1 \end{cases} = 1\alpha_n$$

$$y_n = x_n + \delta x_{n-1} = \alpha_n + \delta\alpha_{n-1}$$

$$\text{If } w_2\text{ReLU}(w_1 y_n) = \beta_n 1$$

$$\Rightarrow z_n = (\alpha_n + \delta\alpha_{n-1} + \beta_n)1$$

$$\Rightarrow \sigma\left(\langle a, z_n\rangle + b\right) = \sigma\left(d(\alpha_n + \delta\alpha_{n-1} + \beta_n) + b\right)$$

$$= x_{n-1}\left(x_n t + (1-x_n)s\right) + (1-x_{n-1})\left(x_n q + (1-x_n)p\right)$$

$$\Rightarrow d(\alpha_n+\delta\alpha_{n-1}+\beta_n)+b = -\log\left(\frac{1}{x_{n-1}\left(x_n t + (1-x_n)s\right) + (1-x_{n-1})\left(x_n q + (1-x_n)p\right)} - 1\right)$$

x_{n-1}	x_n	α_{n-1}	α_n	**Relation**
0	0	$-\frac{1}{2}$	$-\frac{1}{2}$	-log $\left(\frac{1}{p}-1\right) = d\left(-\frac{1}{2}-\frac{\delta}{2}+\beta_{00}\right)+b$
0	1	$-\frac{1}{2}$	$\frac{1}{2}$	-log $\left(\frac{1}{q}-1\right) = d\left(\frac{1}{2}-\frac{\delta}{2}+\beta_{01}\right)+b$
1	0	$\frac{1}{2}$	$-\frac{1}{2}$	-log $\left(\frac{1}{s}-1\right) = d\left(-\frac{1}{2}+\frac{\delta}{2}+\beta_{10}\right)+b$
1	1	$\frac{1}{2}$	$\frac{1}{2}$	-log $\left(\frac{1}{t}-1\right) = d\left(\frac{1}{2}+\frac{\delta}{2}+\beta_{11}\right)+b$

$$\beta_{00} - \beta_{01} = -\frac{1}{d}\log\left(\frac{(1-p)q}{p(1-q)}\right) + 1 \quad (1)$$
$$\beta_{10} - \beta_{11} = -\frac{1}{d}\log\left(\frac{(1-s)t}{s(1-t)}\right) + 1 \quad (2)$$

Beta Coefficients

$$W_2\,\mathrm{ReLU}(W_1 y_n) = \beta_n 1, \quad W_1 = w\mathbf{1}\mathbf{1}^\top, \quad W_2 = -W_1^\top$$

$$\beta_n\mathbf{1} = -W_1^\top\,\mathrm{ReLU}(w\mathbf{1}\mathbf{1}^\top(\alpha_n + \delta\alpha_{n-1}))$$
$$\beta_n\mathbf{1} = -wd\,\mathrm{ReLU}(wd(\alpha_n + \delta\alpha_{n-1})) = -w^2d^2\,\mathrm{ReLU}(\alpha_n + \delta\alpha_{n-1})$$

$$\beta_{00} = -w^2 d^2 \,\mathrm{ReLU}\left(-\frac{1}{2} - \frac{\delta}{2}\right)$$

$$\beta_{01} = -w^2 d^2 \,\mathrm{ReLU}\left(\frac{1}{2} - \frac{\delta}{2}\right)$$

$$\beta_{10} = -w^2 d^2 \,\mathrm{ReLU}\left(-\frac{1}{2} + \frac{\delta}{2}\right)$$

$$\beta_{11} = -w^2 d^2 \,\mathrm{ReLU}\left(\frac{1}{2} + \frac{\delta}{2}\right)$$

$$\text{if } -1 \le \delta \le 1, \quad \begin{cases} \beta_{00} = 0, \\ \beta_{01} = -w^2 d^2 \left(\frac{1-\delta}{2}\right), \\ \beta_{10} = 0, \\ \beta_{11} = -w^2 d^2 \left(\frac{1+\delta}{2}\right). \end{cases}$$

Then b_* will be:

$$b_* = -\log\left(\frac{1-p}{p}\right) + \frac{d}{2}(1+\delta)$$

Conclusion

$$\beta_{00} - \beta_{01} = -\frac{1}{d}\log\left(\frac{(1-p)q}{p(1-q)}\right) + 1 = w^2 d^2 \left(\frac{1-\delta}{2}\right). \quad (1)$$

$$\beta_{10} - \beta_{11} = -\frac{1}{d}\log\left(\frac{(1-s)t}{s(1-t)}\right) + 1 = w^2 d^2 \left(\frac{1+\delta}{2}\right). \quad (2)$$

$$\implies 1 - \frac{w^2 d^2}{2}(1-\delta) = \frac{1}{d}\log\left(\frac{(1-p)q}{p(1-q)}\right)$$

$$1 - \frac{w^2 d^2}{2}(1+\delta) = \frac{1}{d}\log\left(\frac{(1-s)t}{s(1-t)}\right)$$

$$w_* = \pm\sqrt{\frac{1}{d^3\,\delta}\left[\log\left(\frac{(1-p)q}{p(1-q)}\right) - \log\left(\frac{(1-s)t}{s(1-t)}\right)\right]}$$

Therefore, for the case $t < p$, Theorem 1 holds, given that θ_* is defined as follows:

$$\theta_* = (e = a = 1, p_n = (-1/2)1_{n=1}^N, W_O = 0, W_{K,Q,V}, W_1 = w_* 11^T, W_2 = -W_1^T, b = b_*)$$

3.3.5 Theorem 2 (Bad local minimum)

Let the input sequence be $\{x_n\}_{n=1}^N \sim (\pi(p,q,s,t), P(p,q,s,t))$ for some fixed $(p,q,s,t) \in (0,1)^2$. There exists an explicit $\theta_\pi \in \mathbb{R}^{D-d}$ such that it is a bad local minimum for the loss $L(\cdot)$, i.e.,

(i) there exists a neighborhood $\mathcal{B}(\theta_\pi, r)$ with $r > 0$ such that $L(\theta) \geq L(\theta_\pi)$ for all $\theta \in \mathcal{B}(\theta_\pi, r)$, with $L(\theta_\pi) > L_\star$.

Further, θ_π satisfies:

(ii) $\mathbb{P}_{\theta_\pi}(x_{n+1} = 1 \mid x_1^n) = \mathbb{P}(x_{n+1} = 1) = \pi_1$, the marginal distribution.

(iii) $L(\theta_\pi) = H(x_{n+1}) = H(\pi)$, the entropy of the marginal.

(iv) $\nabla L(\theta_\pi) = 0$, i.e., θ_π is a stationary point.

Proof (Sketch). The main idea behind constructing θ_π is that if we set $e = a = 0$ in the `Linear` layer, the model ignores the inputs altogether and outputs a constant probability, and in particular π_1 by choosing the bias b appropriately, i.e., $f_{\theta_\pi}(x_1^n) = \pi_1$ for all x_1^n. For this θ_π, it's easy to show that $L(\theta_\pi) = H(\pi)$ and that it's a stationary point. Further, we show that the Hessian at θ_π follows the structure

$$H = \begin{bmatrix} H_\alpha & 0 \\ 0 & 0 \end{bmatrix},$$

where $H_\alpha \succ 0$, and that it implies the local minimality of θ_π. □

Proof. Given the following assumption, we start with property ii:

$$\theta_\pi = (e = a = o)\, \{P_n\}_{n=1}^N, W_0 = 0, W_{k,Q,v}, W_2 = 0, W_1,$$

$$b = \log\left(\frac{p}{q}\right)$$

$$\{P_n\}_{n=1}^{N} \subset \mathbb{R}^d, \quad W_{k,Q,v} \in \mathbb{R}^{m\times d}, \quad W_1 \in \mathbb{R}^{r\times d}$$

ii)

$$f_{\theta_\pi}(x_1^n) = P_{\theta_\pi}(x_{n+1} = 1 | x_1^n) = P(x_{n+1} = 1) = \pi_1$$

Since $a = 0$

$$\sigma(b) = \sigma\left(\log\left(\frac{P}{q}\right)\right) = \frac{P}{P+q} = \pi_1$$

The model ignores all the input and outputs a constant π_1.

iii)

$$L(\theta_\pi) = H(x_{n+1}) = H(\pi)$$

$$L(\theta_\pi) = -\frac{1}{N}\sum_{n\in[N]} \mathbb{E}_{x_{n+1}}\left[x_{n+1}\log f_{\theta_\pi}(x_1^n) + (1-x_{n+1})\log(1-f_{\theta_\pi}(x_1^n))\right]$$

$$= -\frac{1}{N}\sum_{n\in[N]} \mathbb{E}_{x_{n+1}}\left[x_{n+1}\log \pi_1 + (1-x_{n+1})\log \pi_0\right]$$

$$= \frac{1}{N}\sum_{n\in[N]} \left[-\pi_1 \log \pi_1 - \pi_0 \log \pi_0\right]$$

$$= H(\pi) = H(x_{n+1}).$$

iv)

$$\nabla L(\theta_\pi) = 0$$

$$\nabla_a L = -\frac{1}{N}\sum_{n\in[N]} \mathbb{E}_{x_1^{n+1}}\left[(x_{n+1} - f_{\theta_\pi}(x_1^n)) \cdot \nabla_a (a^T z_n + b)\right]$$

$$= -\frac{1}{N}\sum_{n\in[N]} \mathbb{E}_{x_1^{n+1}}\left[(x_{n+1} - \pi_1)(z_n + \nabla_a z_n \cdot a)\right]$$

For $a = 0$:

$$= -\frac{1}{N} \sum_{n\in[N]} \mathbb{E}_{x_{n+1}} \left[(x_{n+1} - \pi_1) \cdot (p_n + \delta p_{n-1)}) \right] = 0.$$

$$\nabla_b L = -\frac{1}{N} \sum_{n\in[N]} \mathbb{E}_{x_1^{n+1}} \left[(x_{n+1} - f_{\theta_\pi}(x_1^n)) \cdot \nabla_b (a^T z_n + b) \right]$$

$$= -\frac{1}{N} \sum_{n\in[N]} \mathbb{E}_{x_{n+1}} \left[x_{n+1} - \pi_1 \right] = 0.$$

$$\nabla_w L \to \nabla_w (a^T z_n) = (\nabla_w z_n) \cdot a \to 0.$$

i) Here we assume two sets of parameters, α and β, and we calculate the Hessian with respect to these sets.

$$\alpha = (b, a)$$

$$\beta = \{P_n\}_{n=1}^N, W_0, W_{K,Q,V}, W_2, W_1$$

$$\theta = (\alpha, \beta)$$

$$H_\pi = \nabla^2 L(\theta)\big|_{\theta=\theta_\pi}$$

Towards establishing this, we first let $\alpha = (b, a)$ and $\beta = (\{p_n\}_{n=1}^N, W_O, W_{K,Q,V}, W_2, W_1)$ be two different sets of parameters comprising θ, i.e. $\theta = (\alpha, \beta)$. We compute the Hessian $H_\pi \triangleq \nabla^2 L(\theta)\big|_{\theta=\theta_\pi}$, and show that it has the following block-diagonal structure:

$$H_\pi = \begin{bmatrix} H_\alpha & 0 \\ 0 & 0 \end{bmatrix},$$

where H_α corresponds to the Hessian with respect to the parameters a and b in α. Further, we show that H_α is positive-definite. This helps us establish that θ_π is a local minimum. Now we start with the Hessian computation.

$$H_\pi = \begin{bmatrix} H_\alpha & 0 \\ 0 & 0 \end{bmatrix}$$

Gradient Computations:

$$\nabla_b^2 L = \nabla_b(\nabla_b L) = \nabla_b \left(-\frac{1}{N} \sum_{n\in[N]} \mathbb{E}_{x_1^{n+1}}[x_{n+1} - f_\theta(x_1^n)] \right)$$

$$= \nabla_b \left(-\frac{1}{N} \sum_{n\in[N]} \mathbb{E}_{x_1^{n+1}}[x_{n+1} - \nabla_b(a^T z_n + b)] \right)$$

For $a = 1$:

$$= \frac{1}{N} \sum_{n\in[N]} \mathbb{E}[f_\theta(x_1^n)(1 - f_\theta(x_1^n)) \cdot \nabla_b(a^T z_n + b)]$$

For $\theta = \theta_\pi$:

$$= \frac{1}{N} \sum_{n\in[N]} \mathbb{E}\left[\pi_1 \pi_0\right]$$

$$= \pi_1 \pi_0 > 0.$$

$$\nabla_\alpha^2 L = \nabla_\alpha(\nabla_\alpha L) = \nabla_\alpha \left(-\frac{1}{N} \sum_{n\in[N]} \mathbb{E}_{x_1^{n+1}}[(x_{n+1} - f_\theta(x_1^n)) \cdot \nabla_\alpha(a^T z_n)] \right)$$

$$= \nabla_\alpha \left(-\frac{1}{N} \sum_{n\in[N]} \mathbb{E}[(x_{n+1} - f_\theta(x_1^n))(z_n + (\nabla_\alpha z_n) \cdot a)] \right)$$

$$= -\frac{1}{N} \sum_{n\in[N]} \mathbb{E}\left[f_\theta(1 - f_\theta)(z_n + (\nabla_\alpha z_n) \cdot a)(z_n + (\nabla_\alpha z_n) \cdot a)^T \right]$$

$$-\frac{1}{N}\sum_{n\in[N]}\mathbb{E}[(x_{n+1}-f_\theta(x_1^n))(2\nabla_\alpha z_n)].$$

$$\nabla_a^2 L=\frac{1}{N}\sum_{n\in[N]}\left(\mathbb{E}[\pi_1\pi_0\cdot(p_n+\delta p_{n-1})(p_n+\delta p_{n-1})^T]-2\mathbb{E}[(x_{n+1}-\pi_1)(x_n+\delta x_{n-1})I]\right)$$

$$=\frac{1}{N}\sum_{n\in[N]}\left((\pi_0\pi_1)\cdot(p_n+\delta p_{n-1})(p_n+\delta p_{n-1})^T-2\mathbb{E}[x_n(1-p-q)+p-\pi_1)x_nI]\right)$$

$$=\frac{1}{N}\sum_{n\in[N]}\left((\pi_0\pi_1)\cdot p_np_n^T-2\pi_1(\pi_0-q)I\right)$$

$$=\frac{1}{N}\sum\left((\pi_0\pi_1)(p_n+\delta p_{n-1})(p_n+\delta p_{n-1})^T-2(\pi_{11}(t-\pi_1)(1+\delta)+\pi_{01}(q-\pi_1)+\pi_{10}(s-\pi_1))\right)$$

$$\nabla_{ab}L=\nabla_a(\nabla_bL)=\nabla_a\left(-\frac{1}{N}\sum_{n\in[N]}\mathbb{E}_{x_1^{n+1}}[x_{n+1}-f_\theta(x_1^n)]\right)$$

$$=\frac{1}{N}\sum_{n\in[N]}\mathbb{E}[f_\theta(1-f_\theta)(z_n+(\nabla_a z_n)\cdot a)]$$

For $\theta=\theta_\pi$:

$$=\frac{1}{N}\sum_{n\in[N]}\mathbb{E}[\pi_1\pi_0\cdot p_n]$$

$$=\pi_0\pi_1\left(\sum_{n\in[N]}\frac{p_n+\delta p_{n-1}}{N}\right)$$

$$
\begin{aligned}
H_\alpha &\triangleq \nabla^{(2)}_\alpha L(\alpha_\pi, \beta_\pi) = \Pi_0 \Pi_1 \begin{bmatrix} 1 & u^T \\ u & v \end{bmatrix}, \\
v &= \nabla^2_a, u = \nabla_a b, 1 = \nabla^2_b \\
u &\triangleq \frac{\sum_{n\in\mathcal{N}} (\rho_n + \delta\rho_{n-1})}{N}, \\
v &\triangleq \frac{1}{N} \sum_{n\in\mathcal{N}} (\pi_0\pi_1)(\rho_n + \delta\rho_{n-1})(\rho_n + \delta\rho_{n-1}) \\
&\quad - 2\big(\pi_{11}(t-\pi_1)(1+\delta) + \pi_{01}(q-\pi_1) + \pi_{10}(s-\pi_1)\big), \\
H_\beta &\triangleq \nabla^2_\beta L(\alpha_\pi, \beta_\pi) = 0, \\
\beta &= \left(\{\rho_n\}_{n=1}^N, W_0, W_{K,Q,v}, W_2, W_1 \right), \quad W_1, W_2 \in \beta,
\end{aligned}
$$

$$
\begin{aligned}
\nabla_{w_1,w_2} L = \nabla_{w_1}(\nabla_{w_2} L) &= \nabla_{w_1} \left(-\frac{1}{N} \sum_{n\in\mathcal{N}} \mathbb{E}\big[(x_{n+1} - f_\theta(x_1^n))(\nabla_{w_2} z_n \cdot a)\big] \right), \\
&\overset{(a)}{=} \frac{1}{N} \sum_{n\in\mathcal{N}} \mathbb{E}\big[f_\theta(1-f_\theta)(\nabla_{w_2} z_n \cdot a)(\nabla_{w_1} z_n \cdot a)^T\big], \\
(\theta = \theta_\pi) &\Rightarrow 0.
\end{aligned}
$$

$$
\begin{aligned}
\nabla_{w_1} f_\theta(x_1^n) = \nabla_{w_1} \sigma(a^T z_n + b) &= f_\theta(x_1^n)(1 - f_\theta(x_1^n))(\nabla_{w_1} z_n \cdot a), \\
H_\beta = 0, \quad H_{\alpha\beta} = \nabla_{\alpha\beta} L = 0, \quad &\text{and} \quad H_{\beta\alpha} = H^T_{\alpha\beta} = 0, \\
H_\pi = \nabla^2_\pi L(\theta_\pi) &= \begin{bmatrix} H_\alpha & 0 \\ 0 & 0 \end{bmatrix},
\end{aligned}
$$

Positive-Definiteness of H_α

$$
\begin{aligned}
v \cdot uu^T &= \frac{1}{N}\sum_{n\in\mathcal{N}} \pi_0\pi_1(\rho_n + \delta\rho_{n-1})(\rho_n + \delta\rho_{n-1})^T \\
&\quad - 2\big(\pi_{11}(t-\pi_1)(1+\delta) + \pi_{01}(q-\pi_1) + \pi_{10}(s-\pi_1)\big) \\
&\quad - \left[\frac{\sum_{n\in\mathcal{N}}(\rho_n + \delta\rho_{n-1})}{N}\right]\left[\frac{\sum_{n\in\mathcal{N}}(\rho_n + \delta\rho_{n-1})}{N}\right]^T, \\
&= \frac{-2}{\pi_0\pi_1}\Bigg(\pi_{11}(t-\pi_1)(1+\delta) + \pi_{01}(q-\pi_1) + \pi_{10}(s-\pi_1)\Bigg) \\
&\quad + \frac{1}{N}\sum_{n\in\mathcal{N}}(\rho_n + \delta\rho_{n-1})(\rho_n + \delta\rho_{n-1})^T \\
&\quad - \left[\frac{\sum_{n\in\mathcal{N}}(\rho_n + \delta\rho_{n-1})}{N}\right]\left[\frac{\sum_{n\in\mathcal{N}}(\rho_n + \delta\rho_{n-1})}{N}\right]^T, \\
&= \frac{\sum YY^T}{N} - \frac{\sum Y}{N}\frac{\sum Y^T}{N}.
\end{aligned}
$$

$$
\begin{aligned}
&= \frac{-2}{\pi_0\pi_1}\Big(\pi_{11}(t-\pi_1)(1+\delta) + \pi_{01}(q-\pi_1) + \pi_{10}(s-\pi_1)\Big) \\
&\quad + \mathrm{Cov}\left(\rho_n + \delta\rho_{n-1}\right), \\
\pi_0 &= \frac{s+q-2t+1}{2(-p+s+q-t)} \\
&\quad - \frac{\sqrt{-4Pt+4P+s^2+2Sq-2s+q^2-2q+1}}{2(-p+s+q-t)}, \\
\pi_1 &= 1-\pi_0, \\
\pi_0+\pi_1 &= 1, \\
\pi_0 &= (1-p)\pi_0\pi_0 + (1-q)\pi_0\pi_1 + (1-s)\pi_0\pi_1 + (1-t)\pi_1\pi_1, \\
\pi_1 &= p\pi_0\pi_0 + q\pi_0\pi_1 + s\pi_0\pi_1 + t\pi_1\pi_1, \\
\pi_{00} &= (1-p)\pi_{00} + (1-s)\pi_{10}, \\
\pi_{01} &= p\pi_{00} + s\pi_{10}, \\
\pi_{10} &= (1-q)\pi_{01} + (1-t)\pi_{11}, \\
\pi_{11} &= q\pi_{01} + t\pi_{11}, \\
\pi_{00}+\pi_{01}+\pi_{10}+\pi_{11} &= 1, \\
\pi_{00} &= \frac{st-s-t+1}{pq-2pt+2p+st-s-t+1}, \\
\pi_{01} &= \frac{-pt+p}{pq-2pt+2p+st-s-t+1}, \\
\pi_{10} &= \frac{-pt+p}{pq-2pt+2p+st-s-t+1}, \\
\pi_{11} &= \frac{pq}{pq-2pt+2p+st-s-t+1}.
\end{aligned}
$$

where

$$
\mathrm{Cov}\left(\rho_n + \delta\rho_{n-1}\right) = -\left[\frac{\sum_{n\in\mathcal{N}}(\rho_n+\delta\rho_{n-1})}{N}\right]\left[\frac{\sum_{n\in\mathcal{N}}(\rho_n+\delta\rho_{n-1})}{N}\right]^T
$$

is the covariance matrix of the set $\{\rho_n\}_{n=1}^N$ and hence positive semi-definite. We obtain that $V - uu^T \succ 0$. Hence $H_\alpha \succ 0$. Now it remains to show that θ_π is a local minimum.

***H*$_\alpha$ is positive-definite implies θ_π is a local minimum**. Since $H_\alpha \succ 0$, let $H_\alpha \succ \lambda I$ for some $\lambda > 0$s. Since $\theta = (\alpha, \beta) \in \mathbb{R}^{D-d}$, interpret $L(\theta) = L(\alpha, \beta)$ as a function of two variables α and β with appropriate dimensions. We know the following facts about $L(\cdot, \cdot)$:

- Fact 1. $\alpha \mapsto L(\alpha, \beta_\pi)$ has a local minimum (as a function of one variable) at $\alpha = \alpha_\pi$ (since $H_\alpha \succ 0$).
- Fact 2. $\beta \mapsto L(\alpha, \beta)$ is constant in β (since $\alpha_\pi = 0$, the probability $f_\theta(x_1^\pi)$ is constant w.r.t. z_n and hence w.r.t. β (Linear)).
- Fact 3. $\nabla L(\alpha_\pi, \beta_\pi) = 0$ and $H_\pi = \nabla^{(2)} L(\alpha_\pi, \beta_\pi) = \begin{bmatrix} H_\alpha & 0 \\ 0 & 0 \end{bmatrix}$ with $H_\alpha \succ \lambda I$.

Using these facts, we now show that $(\alpha_\pi, \beta_\pi) = \theta_\pi$ is also a local minimum in two variables. We prove this by contradiction. Suppose that (α_π, β_π) is not a local minimum for $L(\cdot, \cdot)$. Without loss of generality, by a shift of coordinates treat θ_π as the origin, i.e., $(\alpha_\pi = 0, \beta_\pi = 0)$ is not a local minimum for $L(\cdot, \cdot)$. Then there exists a unit direction $d = (u, v) \in \mathbb{R}^{D-d}$ with $\|d\|^2 = \|u\|^2 + \|v\|^2 = 1$ and a $0 < \varepsilon_0 < 1$ such that

$$L(\varepsilon d) < L(0), \quad \forall 0 < \varepsilon \leq \varepsilon_0.$$

Clearly $\|u\| > 0$, otherwise it will contradict Fact 2. Using the definition of directional-derivative, we have that

$$\begin{aligned} d^T \nabla^{(2)} L(0,0) d &= \lim_{\varepsilon \to 0^+} \frac{\langle \nabla L(\varepsilon d), d \rangle - \langle \nabla L(0), d \rangle}{\varepsilon} \\ &= \lim_{\varepsilon \to 0^+} \frac{\langle \nabla L(\varepsilon d), d \rangle}{\varepsilon}. \end{aligned}$$

On the other hand, using the Hessian structure, the LHS equals

$$d^T \nabla^{(2)} L(0,0) d \geq \lambda \|u\|^2 \triangleq K_1 > 0.$$

Thus

$$\lim_{\varepsilon \to 0^+} \frac{\langle \nabla L(\varepsilon d), d \rangle}{\varepsilon} = K_1 > 0.$$

Thus, there exists an $\varepsilon_1 > 0$ and $K > 0$ such that

$$\frac{\langle \nabla L(\varepsilon d), d\rangle}{\varepsilon} \geq K, \quad \forall 0 < \varepsilon \leq \varepsilon_1.$$

which implies

$$\langle \nabla L(\varepsilon d), d\rangle \geq K\varepsilon, \quad \forall 0 < \varepsilon \leq \varepsilon_1.$$

Defining the function $g : \mathbb{R}_+ \to \mathbb{R}$ as $g(\varepsilon) = L(\varepsilon d)$, we obtain that $g'(\varepsilon) = \langle \nabla L(\varepsilon d), d\rangle \geq K\varepsilon$ for $0 \leq \varepsilon \leq \varepsilon_1$. Using the fundamental theorem of Calculus, we have that for any $0 \leq \varepsilon \leq \varepsilon_1$,

$$g(\varepsilon) - g(0) = \int_0^\varepsilon g'(t)dt \geq \int_0^\varepsilon Ktdt = \frac{K\varepsilon^2}{2}. \tag{3.15}$$

Thus $g(\varepsilon) \geq L(0) + \frac{K\varepsilon^2}{2}$ for all $0 \leq \varepsilon \leq \varepsilon_1$, whereas $L(\varepsilon d) < L(0)$ for all $0 < \varepsilon < \varepsilon_0$ from Eq. (13). Choosing $\varepsilon_* = \min(\varepsilon_0, \varepsilon_1)$, we have a contradiction for $0 < \varepsilon < \varepsilon_*$. Thus $0 \equiv \theta_\pi = (\alpha_\pi, \beta_\pi)$ is a local minimum. □

3.3.6 Theorem 3 (Global minimum)

Consider the same setting as in Thm. 1. Then for all (p, q, s, t), if $\theta_\star = (e_\star = a_\star, \ldots, b_\star) \in \mathbb{R}^{D-d}$ is a global minimum for the loss $L(\cdot)$ in the weight-tied scenario, then its extension $\bar{\theta}_\star \triangleq (\theta_\star, a_\star) \in \mathbb{R}^D$ is also a global minimum for $L(\cdot)$ in $\mathbb{R}^D$ in the non-weight-tied case.

Further, $\bar{\theta}_\star$ satisfies the same properties (ii)–(iv) as in Thm. 1.

Proof

By extending $\theta_\star \in \mathbb{R}^{D-d}$ to $\bar{\theta}_\star \triangleq (\theta_\star, a_\star) \in \mathbb{R}^D$, it follows from the Transformer architecture that

$$\mathbb{P}_{\bar{\theta}_\star}(x_{n+1} = 1 \mid x_1^n) = \mathbb{P}_{\theta_\star}(x_{n+1} = 1 \mid x_1^n) = \mathbb{P}(x_{n+1} = 1 \mid x_n, x_{n-1}),$$

the Markov kernel. As the proof of Thm. 1, prediction probability equalling the kernel is a sufficient condition for global optimality. Hence $\bar{\theta}_\star$ is a global minimum for $L(\cdot)$ in $\mathbb{R}^D$. □

3.3.7 Theorem 4 (Saddle point)

Consider the same setting as in Thm. 2 , let $\theta_\pi = (e_\pi = a_\pi, \ldots, b_\pi) \in \mathbb{R}^{D-d}$ be the corresponding bad local minimum for the loss $L(\cdot)$ in the weight-tied scenario. Then its extension $\bar{\theta}_\pi \triangleq (\theta_\pi, a_\pi) \in \mathbb{R}^D$ is a saddle point for $L(\cdot)$ in $\mathbb{R}^D$ in the non-weight-tied case.

Further, $\bar{\theta}_\pi$ satisfies the same properties (ii)–(iv) as in Thm. 2.

Proof. Since $\bar{\theta}_\pi \triangleq (\theta_\pi, a_\pi) \in \mathbb{R}^D$ is a canonical extension of $\theta_\pi = (e_\pi = a_\pi, \ldots, b_\pi) \in \mathbb{R}^{D-d}$, which is a local minimum for $L(\cdot)$ in $\mathbb{R}^{D-d}$, following the same-steps for the gradient computation and probability evaluation as in proof of Thm. 2, it immediately follows that $\bar{\theta}_\pi$ also satisfies properties (ii)–(iv), i.e., it's a stationary point, it captures the marginal, since $\mathbb{P}_{\bar{\theta}_\pi}(x_{n+1} = 1 \mid x_1^n) = \mathbb{P}_{\theta_\pi}(x_{n+1} = 1 \mid x_1^n) = \mathbb{P}(x_{n+1} = 1 \mid x_n, x_{n-1}) = \pi$, and hence its loss equals the entropy of stationary distribution $H(\pi)$. In a similar fashion, the Hessian computation is essentially the same except for a slight difference in the Hessian structure, i.e.,

$$H(\bar{\theta}_\pi) \triangleq \nabla^2 L(\bar{\theta}_\pi) = \begin{bmatrix} H_\alpha & 0 \\ 0 & 0 \end{bmatrix}, \quad H_\alpha = \pi_0\pi_1 \begin{bmatrix} \nabla_b^2 & \nabla_{ba} & \nabla_{be} \\ \nabla_{ab} & \nabla_a^2 & \nabla_{ae} \\ \nabla_{eb} & \nabla_{ea} & \nabla_e^2 \end{bmatrix} = \begin{bmatrix} 1 & u^\top & 0 \\ u & V & \nabla_{ae} \\ 0 & p+q-1 & 0 \end{bmatrix},$$

where

$$\begin{aligned} u &\triangleq \frac{\sum_{n\in\mathcal{N}}(\rho_n + \delta\rho_{n-1})}{N}, \\ v &\triangleq \frac{1}{N}\sum_{n\in\mathcal{N}} \pi_0\pi_1(\rho_n + \delta\rho_{n-1})(\rho_n + \delta\rho_{n-1})^T \\ &\quad - 2\big(\pi_{11}(t-\pi_1)(1+\delta) + \pi_{01}(q-\pi_1) + \pi_{10}(s-\pi_1)\big). \end{aligned}$$

In the weight-tied case, we observe that the matrix V also contains the $(p+q-1)I$ terms which in the non-weight-tied case get decoupled (due to separate e and a parameters). In fact, H_α corresponds to the Hessian with respect to the parameters $\alpha = (b, a, e)$, i.e.,

$$H_\alpha = \nabla^2 L\big|_{\alpha=\alpha_\star}.$$

Now it remains to show that H_α is indefinite.$\bar{\theta}_\pi$ a saddle point.

Clearly, H_α cannot be negative definite since with $d = (1, 0, \ldots, 0)$, we have $d^\top H_\alpha d = \pi\sigma_\pi > 0$ for $(p, q) \in (0, 1)$. Now we show that it cannot be positive definite either. Denoting

$$H_\alpha = \pi_0\pi_1 \begin{bmatrix} 1 & b^\top \\ b & C \end{bmatrix} - (p+q-1)I,$$

$$b \triangleq \begin{bmatrix} \sum_{n\in[N]} \frac{p_n}{N} \\ 0 \end{bmatrix}, \quad C \triangleq \begin{bmatrix} \sum_{n\in[N]} \frac{p_n p_n^\top}{N} & (p+q-1)I \\ (p+q-1)I & 0 \end{bmatrix}.$$

Using the characterization of positive-definiteness by Schur's complement (Horn & Johnson, 2012), we have that $H_\alpha \succ 0 \iff 1 > 0$ and $C - bb^\top \succ 0$. This can be further simplified to

$$M \triangleq C - bb^\top = \begin{bmatrix} \sum_{n\in[N]} \frac{p_n p_n^\top}{N} & (p+q-1)I \\ (p+q-1)I & 0 \end{bmatrix} - \begin{bmatrix} \sum_{n\in[N]} \frac{p_n}{N} \\ 0 \end{bmatrix} \begin{bmatrix} \sum_{n\in[N]} \frac{p_n^\top}{N} & 0 \end{bmatrix}.$$

$$= \begin{bmatrix} \mathrm{Cov}(\{p_n\}_{n=1}^N) & (p+q-1)I \\ (p+q-1)I & 0 \end{bmatrix}$$

where $\mathrm{Cov}(\{p_n\}_{n=1}^N) = \sum_{n\in[N]} \frac{1}{N} \left(p_n - \sum_{n\in[N]} \frac{p_n}{N}\right) \left(p_n - \sum_{n\in[N]} \frac{p_n}{N}\right)^\top$ is the covariance matrix of the set $\{p_n\}_{n=1}^N$. Now we show that M cannot be positive definite. Suppose not. Then there exists a $\lambda > 0$ such that $v^\top M v \geq \lambda ||v||^2$ for all $v = (v_1, v_2) \in \mathbb{R}^{2d}$. This further implies that

$$v_1^\top \mathrm{Cov}(\{p_n\}_{n=1}^N) v_1 + 2(p+q-1)\langle v_1, v_2\rangle + \|v_2\|^2 \geq \|v_2\|^2, \quad \forall v_1, v_2 \in \mathbb{R}^d.$$

Taking $v_1 = 0$ the above inequality implies that $\lambda||v||^2 \leq 0$ for all $v_2 \in \mathbb{R}^d$, which is a contradiction. Hence M cannot be positive definite and consequently neither can H_α. □

with parameters $\boldsymbol{\theta} = (e, \{p_n\}_{n=1}^N, W_1, W_2, a, b, \delta)$. We will sometimes analyze the *weight-tied* case $a = e$; the *untied* case treats them independently. For some constructive proofs, we set $W_2 = 0$ (so $z_n = y_n$) or use the rank-1 FF setting $W_1 = w\,\mathbf{1}\mathbf{1}^\top$, $W_2 = -W_1^\top$.

3.4 Part Two: Attention-only transformers

In this section, we check if the theorems of the previous section are also true for the architecture proposed in [7]. This section consists of two parts; First, we check the theorems for the first-order Markov chain and then for the second-order Markov chain.

3.4.1 Transformer Architecture

In this part, we use the architecture proposed in [7]:

for $n = 1, 2, \ldots, T$ **do**	
$\quad \mathbf{x}_n^{(1)} = \mathrm{Emb}(\mathbf{x}_n) \in \mathbb{R}^d.$	*Embeddings*
for $\ell = 1, 2, \ldots, L$ **do**	
$\quad$ **for** $n = 1, 2, \ldots, T$ **do**	
$\quad\quad \tilde{\mathbf{x}}_n^{(\ell)} = \sum_{i \in [n]} \mathrm{att}_{n,i}^{(\ell)} \cdot \mathbf{W}_V^{(\ell)} \left(\mathbf{x}_i^{(\ell)} + \mathbf{p}_{n-i}^{(\ell,V)} \right) \in \mathbb{R}^d,$	*Attention*
$\quad\quad \mathbf{x}_n^{(\ell+1)} = \mathbf{x}_n^{(\ell)} + \tilde{\mathbf{x}}_n^{(\ell)},$	*Residual*
$\mathrm{logit}_T = \mathbf{a}\mathbf{x}_T^{(L+1)} + \mathbf{b} \in \mathbb{R},$	*Linear*
$\Pr_\theta(\cdot \mid \mathbf{x}_1, \ldots, \mathbf{x}_T) = f(\mathrm{logit}_T)$	*Prediction*

Architecture 1: Attention-only transformer.

We use cross cross-entropy loss function similar to 3.11.
In the linear layer, there is a slight modification; the output of this layer in this part of our work is a scalar, while in [7] they use vectors. Also, the prediction is calculated using a sigmoid function.

3.4.2 First Order MC

First, we assume the input data comes from a first-order Markov chain. For the embedding layer, we follow the idea proposed in [7] for the proof of their Theorem 4.1. Also we only

consider one layer of attention (L=1), so based on the same idea from [7], the output of the model after the residual layer is as follows:

$$x_n^{(1)} = \mathrm{Emb}(x_n) = \kappa \begin{bmatrix} 1_{1\times 2} & e_{x_n}^S & 0_{1\times 2S} \end{bmatrix}^T \in \mathbb{R}^d.$$

for a constant $\kappa > 0$ to be chosen later and $d = 2S + 2$. The relative position encodings will essentially be supported on the first two coordinates, the middle S coordinates are a one-hot encoding of the symbol x_n and the last $2S$ coordinates are 0. The relative position encodings in the first layer are chosen to be $p_{n-i}^{(1),K} = \kappa(-1 + \mathbb{I}(n - i = 1))\, e_1^d \in \mathbb{R}^d$ and $p_{n-i}^{(1),V} = 0 \in \mathbb{R}^d$. Choose $W_K^{(1)}$ and $W_Q^{(1)}$ to be $e_1^d (e_1^d)^T \in \mathbb{R}^{d\times d}$. With this choice,

$$\left\langle W_K^{(1)} \left(x_i^{(1)} + p_{n-i}^{(1),K} \right), W_Q^{(1)} x_n^{(1)} \right\rangle = \kappa \mathbb{I}(n - i = 1)$$

As $\kappa \to \infty$, the attention pattern (which takes the softmax over of these inner products over $i \in [n]$) computes,

$$\mathrm{att}_{n,i}^{(1)} = \mathbb{I}(i = n - 1)$$

for any $n > 1$. Choose the value matrix as,

$$W_V^{(1)} = \begin{bmatrix} 0_{(2+S)\times 2} & 0 & 0 \\ 0 & I_{S\times S} & 0 \\ 0 & 0 & 0 \end{bmatrix} \in \mathbb{R}^{d\times d}$$

And with this choice and the residual connection, we get,

$$x_n^{(2)} = \kappa \begin{bmatrix} 1_{1\times 2} & e_{x_n}^S & e_{x_{n-1}}^S & 0 \end{bmatrix} \in \mathbb{R}^d$$

which serves as the input to the 2nd transformer layer. Here, since we are studying binary tokens, the vocabulary size (S) is 2.

$$e_0 = [1 \;\; 0], \qquad e_1 = [0 \;\; 1],$$

$$x_n^{(1)} = \mathrm{Emb}(x_n) = k \begin{bmatrix} 1 \\ 1 \\ e_{x_n} \\ 0 \\ 0 \\ 0 \end{bmatrix}^\top, \qquad d = 8$$

$$x_n^{(2)} = k \begin{bmatrix} 1 \\ 1 \\ e_{x_n} \\ e_{x_{n-1}} \\ 0 \\ 0 \end{bmatrix}^\top$$

We can write $x_n^{(2)}$ as a function of x_n and x_{n-1}:

$$x_n^{(2)} = \beta\, x_n + \alpha\, x_{n-1} + \xi.$$

$$\beta = \begin{bmatrix} 0 \\ 0 \\ -1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \qquad \alpha = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ -1 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \qquad \xi = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Given these assumptions, we first prove Lemma 1 and Lemma 2 as in the previous section and then move forward to the theorems.

3.4.3 Lemma 1

Lemma 1 (Loss as KL divergence). Let the input sequence be

$$\{x_n\}_{n=1}^N \sim (\pi(p,q), P(p,q), \mathbf{P}(p,q))$$

for some fixed $(p,q) \in (0,1)^2$, $\boldsymbol{\theta} = (e, \{p_n\}_{n=1}^N, \ldots, b, a) \in \mathbb{R}^D$ be the full list of the transformer parameters, and $L(\boldsymbol{\theta})$ be the corresponding cross-entropy loss in 3.11. Then the loss function $L(\cdot)$ is equivalent to the KL divergence between the Markov kernel and the predicted distribution, i.e.,

$$L(\boldsymbol{\theta}) = \frac{1}{N} \sum_{n \in [N]} \mathbb{E}_{x_1^n} \left[D_{KL}(\mathbb{P}(x_{n+1} = \cdot \mid x_n) \parallel \mathbb{P}_{\boldsymbol{\theta}}(x_{n+1} = \cdot \mid x_1^n)) \right] + H(x_{n+1} \mid x_n), \quad (3.16)$$

where $D_{KL}(P \parallel Q)$ is the KL divergence between two distributions P and Q, and $H(x_{n+1} \mid x_n)$ is the entropy rate of the Markov chain.

Remark. Consequently, 3.16 highlights that any parameter $\boldsymbol{\theta}$ with the predicted probability

$$f_{\boldsymbol{\theta}}(x_1^n) = \mathbb{P}(x_{n+1} = 1 \mid x_n)$$

is a global minimum for the loss L as $D_{KL}(\cdot \parallel \cdot) \geq 0$. We utilize this fact in the proof of Theorem 1 later.

Proof. Consider the loss function $L(\cdot)$ given in 3.11. We can rewrite it as follows:

$$L(\theta) = -\frac{1}{N} \sum_{n \in [N]} \mathbb{E}_{x_1^n x_{n+1}} \left[x_{n+1} \cdot \log f_\theta(x_1^n) + (1 - x_{n+1}) \cdot \log(1 - f_\theta(x_1^n)) \right]$$

$$= -\frac{1}{N} \sum_{n \in [N]} \Big(\mathbb{E}_{x_1^n | x_n}[x_{n+1}] \cdot \log f_\theta(x_1^n) + \mathbb{E}_{x_{n+1} | x_1^n}[1 - x_{n+1}] \cdot \log(1 - f_\theta(x_1^n)) \Big)$$

$$= -\frac{1}{N} \sum_{n \in [N]} \mathbb{E}_{x_1^n} \Big[\mathbb{P}(x_{n+1} = 1 \mid x_n) \log \frac{f_\theta(x_1^n)}{\mathbb{P}(x_{n+1} = 1 \mid x_n)} + \mathbb{P}(x_{n+1} = 0 \mid x_n) \log \frac{1 - f_\theta(x_1^n)}{\mathbb{P}(x_{n+1} = 0 \mid x_n)} \Big]$$

$$- \Big[\mathbb{P}(x_{n+1} = 1 \mid x_n) \log \frac{1}{\mathbb{P}(x_{n+1} = 1 \mid x_n)} - \mathbb{P}(x_{n+1} = 0 \mid x_n) \log \frac{1}{\mathbb{P}(x_{n+1} = 0 \mid x_n)} \Big].$$

$$= -\frac{1}{N} \sum_{n \in [N]} \mathbb{E}_{x_1^n} \Big[\mathbb{P}(x_{n+1} = 1 \mid x_n) \log \frac{f_\theta(x_1^n)}{\mathbb{P}(x_{n+1} = 1 \mid x_n)} + \mathbb{P}(x_{n+1} = 0 \mid x_n) \log \frac{1 - f_\theta(x_1^n)}{\mathbb{P}(x_{n+1} = 0 \mid x_n)} \Big]$$

$$- \mathbb{E}_{x_n} \Big[\mathbb{E}_{x_1^{n-1} | x_n} \Big[\mathbb{P}(x_{n+1} = 1 \mid x_n) \log \frac{1}{\mathbb{P}(x_{n+1} = 1 \mid x_n)} \Big] + \mathbb{P}(x_{n+1} = 0 \mid x_n) \log \frac{1}{\mathbb{P}(x_{n+1} = 0 \mid x_n)} \Big].$$

$$= -\frac{1}{N} \sum_{n \in [N]} \mathbb{E}_{x_1^n} \Big[\mathbb{P}(x_{n+1} = 1 \mid x_n) \log \frac{f_\theta(x_1^n)}{\mathbb{P}(x_{n+1} = 1 \mid x_n)} + \mathbb{P}(x_{n+1} = 0 \mid x_n) \log \frac{1 - f_\theta(x_1^n)}{\mathbb{P}(x_{n+1} = 0 \mid x_n)} \Big]$$

$$- \mathbb{E}_{x_n} \Big[\mathbb{P}(x_{n+1} = 1 \mid x_n) \log \frac{1}{\mathbb{P}(x_{n+1} = 1 \mid x_n)} + \mathbb{P}(x_{n+1} = 0 \mid x_n) \log \frac{1}{\mathbb{P}(x_{n+1} = 0 \mid x_n)} \Big].$$

Since $f_\theta(x_1^n) = \mathbb{P}_\theta(x_{n+1} = 1 \mid x_n)$, we have that the first term is

$$\mathbb{P}(x_{n+1} = 1 \mid x_n) \log \frac{f_\theta(x_1^n)}{\mathbb{P}(x_{n+1} = 1 \mid x_n)} + \mathbb{P}(x_{n+1} = 0 \mid x_n) \log \frac{1 - f_\theta(x_1^n)}{\mathbb{P}(x_{n+1} = 0 \mid x_n)}$$

$$= -D_{\mathrm{KL}}\big(\mathbb{P}(x_{n+1} = \cdot \mid x_n) \parallel \mathbb{P}_\theta(x_{n+1} = \cdot \mid x_1^n)\big).$$

Further, observe that the second term is exactly the entropy rate $H(x_{n+1} \mid x_n)$.

Hence, the loss reduces to

$$L(\theta) = \frac{1}{N} \sum_{n \in [N]} \mathbb{E}_{x_1^n} \Big[D_{\mathrm{KL}}(\mathbb{P}(x_{n+1} = \cdot \mid x_n) \parallel \mathbb{P}_\theta(x_{n+1} = \cdot \mid x_1^n)) + H(x_{n+1} \mid x_n) \Big],$$

and we are done.

3.4.4 Lemma 2

Lemma 2 (Gradient computation). Consider the same data and parameter setting as in Lemma 1 and let $L(\boldsymbol{\theta})$ be the cross-entropy loss in 3.11. Then for any parameter $w \in \boldsymbol{\theta}$,

$$\begin{aligned} \nabla_w L(\boldsymbol{\theta}) &= -\frac{1}{N} \sum_{n \in [N]} \mathbb{E}_{x_1^{n+1}} \big[(x_{n+1} - f_{\boldsymbol{\theta}}(x_1^n)) \cdot \nabla_w (a^\top z_n + b) \big] \\ &= -\frac{1}{N} \sum_{n \in [N]} \mathbb{E}_{x_1^n} \big[(\mathbb{P}(x_{n+1} = 1 \mid x_n) - f_{\boldsymbol{\theta}}(x_1^n)) \cdot \nabla_w (a^\top z_n + b) \big]. \end{aligned} \tag{3.17}$$

Remark. 3.17 highlights that any parameter $\boldsymbol{\theta}$ with the predicted probability $f_{\boldsymbol{\theta}}(x_1^n) = \mathbb{P}(x_{n+1} = 1 \mid x_n)$ is also a stationary point for the loss L. We utilize this fact in the proof of Theorem 1 later.

Proof. Let θ_j of $\boldsymbol{\theta} \in \mathbb{R}^D$:

$$\begin{aligned}\frac{\partial}{\partial \tilde{\theta}_j} L(\tilde{\theta}) &= -\frac{1}{N} \sum_{n \in [N]} \mathbb{E}_{x_1^{n+1}} \Big[\big(x_{n+1} - f_{\tilde{\theta}}(x_1^n) \big) \cdot \frac{\partial}{\partial \tilde{\theta}_j} \big(a^\top z_n + b \big) \Big] \\ &= -\frac{1}{N} \sum_{n \in [N]} \mathbb{E}_{x_1^n} \Big[\big(\mathbb{P}(x_{n+1} = 1 \mid x_n) - f_{\tilde{\theta}}(x_1^n) \big) \cdot \frac{\partial}{\partial \tilde{\theta}_j} \big(a^\top z_n + b \big) \Big].\end{aligned}$$

Also we had:

$$L(\tilde{\theta}) = -\frac{1}{N} \sum_{n \in [N]} \mathbb{E}_{x_1^{n+1}} \big[x_{n+1} \cdot \log f_{\tilde{\theta}}(x_1^n) + (1 - x_{n+1}) \cdot \log(1 - f_{\tilde{\theta}}(x_1^n)) \big],$$

which implies that

$$\begin{aligned}\frac{\partial}{\partial \tilde{\theta}_j} L(\tilde{\theta}) &= -\frac{1}{N} \sum_{n \in [N]} \mathbb{E}_{x_1^{n+1}} \Big[x_{n+1} \cdot \frac{\partial}{\partial \tilde{\theta}_j} \log f_{\tilde{\theta}}(x_1^n) + (1 - x_{n+1}) \cdot \frac{\partial}{\partial \tilde{\theta}_j} \log(1 - f_{\tilde{\theta}}(x_1^n)) \Big] \\ &= -\frac{1}{N} \sum_{n \in [N]} \mathbb{E}_{x_1^{n+1}} \Big[x_{n+1} \cdot \frac{1}{f_{\tilde{\theta}}(x_1^n)} \frac{\partial}{\partial \tilde{\theta}_j} f_{\tilde{\theta}}(x_1^n) + (1 - x_{n+1}) \cdot \frac{1}{1 - f_{\tilde{\theta}}(x_1^n)} \frac{\partial}{\partial \tilde{\theta}_j} (1 - f_{\tilde{\theta}}(x_1^n)) \Big].\end{aligned}$$

Since $f_{\tilde{\theta}}(x_1^n) = \sigma(a^\top z_n + b)$, we first note that the derivative of σ is given by

$$\sigma'(z) = \frac{e^{-z}}{(1 + e^{-z})^2} = \sigma(z)(1 - \sigma(z)).$$

Hence, the derivative $\frac{\partial}{\partial \tilde{\theta}_j} f_{\tilde{\theta}}(x_1^n)$ can be written as

$$\begin{aligned}\frac{\partial}{\partial \tilde{\theta}_j} f_{\tilde{\theta}}(x_1^n) &= \frac{\partial}{\partial \tilde{\theta}_j} \sigma(a^\top z_n + b) \\ &= \sigma(a^\top z_n + b) \big[1 - \sigma(a^\top z_n + b) \big] \cdot \frac{\partial}{\partial \tilde{\theta}_j} \big(a^\top z_n + b \big) \\ &= f_{\tilde{\theta}}(x_1^n) \big[1 - f_{\tilde{\theta}}(x_1^n) \big] \cdot \frac{\partial}{\partial \tilde{\theta}_j} \big(a^\top z_n + b \big).\end{aligned}$$

Plugging this into the above expression, we have

$$\begin{aligned}
\frac{\partial}{\partial \tilde{\theta}_j} L(\tilde{\theta}) &= -\frac{1}{N} \sum_{n \in [N]} \mathbb{E}_{x_1^{n+1}} \Big[x_{n+1} \big(1 - f_{\tilde{\theta}}(x_1^n)\big) \frac{\partial}{\partial \tilde{\theta}_j} \big(a^\top z_n + b\big) \\
&\qquad - (1 - x_{n+1}) f_{\tilde{\theta}}(x_1^n) \frac{\partial}{\partial \tilde{\theta}_j} \big(a^\top z_n + b\big) \Big] \\
&= -\frac{1}{N} \sum_{n \in [N]} \mathbb{E}_{x_1^{n+1}} \Big[\big(x_{n+1} - f_{\tilde{\theta}}(x_1^n)\big) \frac{\partial}{\partial \tilde{\theta}_j} \big(a^\top z_n + b\big) \Big] \\
&= -\frac{1}{N} \sum_{n \in [N]} \mathbb{E}_{x_1^n} \Big[\mathbb{E}_{x_{n+1} | x_1^n} \big[x_{n+1} - f_{\tilde{\theta}}(x_1^n) \big] \frac{\partial}{\partial \tilde{\theta}_j} \big(a^\top z_n + b\big) \Big] \\
&= -\frac{1}{N} \sum_{n \in [N]} \mathbb{E}_{x_1^n} \Big[\big(\mathbb{P}(x_{n+1} = 1 \mid x_n) - f_{\tilde{\theta}}(x_1^n) \big) \frac{\partial}{\partial \tilde{\theta}_j} \big(a^\top z_n + b\big) \Big],
\end{aligned}$$

and we are done.

3.4.5 Theorem 1 (Global minimum)

Let the input sequence be $\{x_n\}_{n=1}^N \sim \big(\pi(p,q), \mathbb{P}(p,q)\big)$ for some fixed $(p,q) \in (0,1)^2$. Then for all (p,q), there exists a $\theta_\star \in \mathbb{R}^{D-d}$ with an explicit construction such that it is a global minimum for the population loss $L(\cdot)$ in Eq3.11, i.e.

(i) $L(\theta) \geq L(\theta_\star)$ for all $\theta \in \mathbb{R}^{D-d}$.

Further, $\theta_\star$ satisfies:

(ii) $\mathbb{P}_{\theta_\star}(x_{n+1} = 1 \mid x_1^n) = \mathbb{P}(x_{n+1} = 1 \mid x_n)$, the Markov kernel.

(iii) $L(\theta_\star) = H(x_{n+1} \mid x_n)$, the entropy rate of the Markov chain.

(iv) $\nabla L(\theta_\star) = 0$, i.e. $\theta_\star$ is a stationary point.

Remark 1. While there could exist other global minima in principle, Thm. 1 highlights a specific way to realize this.

Proof (Sketch). The key idea here is to show that any θ satisfying $f_\theta(x_1^n) = \mathbb{P}(x_{n+1} = 1 \mid x_n)$ is a global minimum and is a stationary point with the loss being the entropy rate

(Lemmas. 1 and 2). To construct such a θ, we utilize the fact the Markov kernel is only a function of x_n and thus we can ignore the past information in the Attention layer using only the skip.

Proof.

In view of Lemma 1 and Lemma 2, it follows that any θ satisfying $f_\theta(x_1^n) = \mathbb{P}(x_{n+1} = 1 \mid x_n)$ is a global minimum with loss equalling the entropy rate, and is a stationary point. Hence it suffices to construct such a θ.

To build our intuition towards designing $\theta_\star$, recall that the Markov kernel $\mathbb{P}(x_{n+1} = 1 \mid x_n)$ can be succinctly written as

$$\mathbb{P}(x_{n+1} = 1 \mid x_n) = x_n(1-p-q) + p.$$

$$\sigma\big(a(\alpha x_{n-1} + \beta x_n + \xi) + b\big) = x_n(1-p-q) + p$$

$$a(\alpha x_{n-1} + \beta x_n + \xi) + b = \log\left(\frac{x_n(1-p-q)+p}{(1-x_n)(1-p)+qx_n}\right)$$

$$\begin{aligned}
&x_n = 0 \qquad x_{n-1} = 0 \quad (00) \quad a\xi + b = \log\left(\frac{p}{1-p}\right), \\
&x_n = 1 \qquad x_{n-1} = 0 \quad (01) \quad a\beta + a\xi + b = \log\left(\frac{1-q}{q}\right), \\
&x_n = 0 \qquad x_{n-1} = 1 \quad (10) \quad a\alpha + a\xi + b = \log\left(\frac{p}{1-p}\right), \\
&x_n = 1 \qquad x_{n-1} = 1 \quad (11) \quad a(\alpha+\beta) + a\xi + b = \log\left(\frac{1-q}{q}\right).
\end{aligned}$$

$$a(\alpha - \beta) = \log\left(\frac{p}{1-p}\right) - \log\left(\frac{1-q}{q}\right)$$

$$a(\alpha + \beta) = \log\left(\frac{1-q}{q}\right) - \log\left(\frac{p}{1-p}\right)$$

$$\Rightarrow \; a\alpha = 0$$

$$a\beta = \log\left(\frac{1-q}{q}\right) - \log\left(\frac{p}{1-p}\right)$$

$$a\xi = \log\left(\frac{p}{1-p}\right) - b$$

$$a = [a_1 \ \dots \ a_8], \qquad b = scalar$$

$$\beta = \begin{bmatrix} 0 \\ 0 \\ -1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \qquad \alpha = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ -1 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \qquad \xi = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Therefore we have:

$$a_1 + a_2 + a_3 + a_5 = \log\left(\frac{p}{1-p}\right) - b$$

$$a_6 - a_5 = 0 \quad \Rightarrow \quad a_6 = a_5$$

$$a_4 - a_3 = \log\left(\frac{1-q}{q}\right) - \log\left(\frac{p}{1-p}\right)$$

So we can say that as long as b and a meet these three conditions, we have a global minimum in θ_*.

3.4.6 Theorem 2 (Bad local minimum)

Let the input sequence be $\{x_n\}_{n=1}^N \sim (\pi(p,q), P(p,q))$ for some fixed $(p,q) \in (0,1)^2$. There exists an explicit $\theta_\pi \in \mathbb{R}^{D-d}$ such that it is a bad local minimum for the loss $L(\cdot)$, i.e.,

(i) there exists a neighborhood $\mathcal{B}(\theta_\pi, r)$ with $r > 0$ such that $L(\theta) \geq L(\theta_\pi)$ for all $\theta \in \mathcal{B}(\theta_\pi, r)$, with $L(\theta_\pi) > L_\star$.

Further, θ_π satisfies:

(ii) $\mathbb{P}_{\theta_\pi}(x_{n+1} = 1 \mid x_1^n) = \mathbb{P}(x_{n+1} = 1) = \pi_1$, the marginal distribution.

(iii) $L(\theta_\pi) = H(x_{n+1}) = H(\pi)$, the entropy of the marginal.

(iv) $\nabla L(\theta_\pi) = 0$, i.e., θ_π is a stationary point.

Proof (Sketch). For this θ_π, it's easy to show that $L(\theta_\pi) = H(\pi)$ and that it's a stationary point. Further, we show that the Hessian at θ_π follows the structure

$$H = \begin{bmatrix} H_\alpha & 0 \\ 0 & 0 \end{bmatrix},$$

where $H_\alpha \succ 0$, and that it implies the local minimality of θ_π. □

Proof. Given the following assumption, we start with property ii:

$$\begin{aligned} x_n^{(2)} &= k\big[\, 1_{1\times 2} \;\; e^{S_{x_n}} \;\; e^{S_{x_{n-1}}} \;\; 0\big] \in \mathbb{R}^d \\ &= \alpha x_{n-1} + \beta x_n + \xi \end{aligned}$$

$$\pi_1 = \frac{p}{q+p}, \qquad \pi_0 = \frac{q}{p+q}$$

$$\text{Assumption: } \{a = 0,\ b = \log(p/q),\ L = 2\}$$

starting with ii)

$$\text{given } a = 0 \;\Rightarrow\; f_{\theta_\pi}(x_1^n) = \sigma(b) = \sigma\big(\log(p/q)\big) = \frac{p}{p+q} = \pi_1$$

iii) $\mathrm{L}(\theta_\pi) = H(x_{n+1}) = H(\pi)$

$$f_{\theta_\pi}(\cdot) = \pi_1 = E[x_{n+1}]$$

$$L(\bar{\theta}) \triangleq -\frac{1}{N}\sum_{n\in[N]} E_{x_1^{n+1}}\Big[x_{n+1}\log f_{\bar{\theta}}(x_1^n) + (1-x_{n+1})\log\big(1-f_{\bar{\theta}}(x_1^n)\big)\Big]$$

$$L(\theta_\pi) = -\frac{1}{N}\sum_{n\in[N]} E_{x_1^{n+1}}\Big[x_{n+1}\log f_{\theta_\pi}(x_1^n) + (1-x_{n+1})\log\big(1-f_{\theta_\pi}(x_1^n)\big)\Big]$$

$$= -\frac{1}{N}\sum_{n\in[N]} E_{x_1^{n+1}}\Big[x_{n+1}\log(\pi_1) + (1-x_{n+1})\log(\pi_0)\Big]$$

$$\frac{1}{N}\sum_{n\in[N]}\big[-\pi_1\log(\pi_1) - \pi_0\log(\pi_0)\big] = H(\pi) = H(x_{n+1})$$

iv) $\nabla L(\theta_\pi) = 0$ (lemma 2)

$$\nabla_a L = -\frac{1}{N}\sum_{n\in N} E_{x_1^{n+1}}\Big[\big(x_{n+1} - f_{\theta_\pi}(x_1^n)\big)\cdot\nabla_a\big(a^\top\bar{z}_n + b\big)\Big]$$

$$-\frac{1}{N}\sum_{n\in N} E_{x_1^{n+1}}\Big[\big(x_{n+1}-\pi_1\big)\big(\bar{z}_n + \nabla_a\bar{z}_n\cdot a\big)\Big] =$$

$$a = 0 \quad \Rightarrow \quad -\frac{1}{N}\sum_{n\in N} E_{x_1^{n+1}}\Big[\big(x_{n+1}-\pi_1\big)\cdot\bar{z}_n\Big] = 0$$

$$\nabla_b L = -\frac{1}{N}\sum_{n\in N} E_{x_1^{n+1}}\Big[\big(x_{n+1} - f_{\theta_\pi}(x_1^n)\big)\nabla_b\big(a^\top\bar{z}_n + b\big)\Big]$$

$$= -\frac{1}{N}\sum_{n\in N} E_{x_1^{n+1}}\big[x_{n+1}-\pi_1\big] = 0$$

$$\text{same for other } w \;\Rightarrow\; \nabla L = 0$$

(i) θ_π is a bad local min

Towards establishing this, we first let

$$\alpha = (b, a) \quad \text{and} \quad \beta = (\{p_n\}_{n=1}^N, W_{K,Q,V})$$

be two different sets of parameters comprising θ, i.e.

$$\theta = (\alpha, \beta)$$

and compute the Hessian

$$H_\pi \triangleq \nabla^{(2)} L(\theta)\big|_{\theta=\theta_\pi}$$

and show that it has the following block-diagonal structure:

$$H_\pi = \begin{bmatrix} H_\alpha & 0 \\ 0 & 0 \end{bmatrix},$$

where H_α corresponds to the Hessian with respect to the parameters a and b in α. Further we show that $H_\alpha \succ 0$, i.e. it is positive-definite. This helps us in establishing that θ_π a local minimum. Now we start with the Hessian computation.

Hessian computation. We first compute the Hessian with respect to α.

$$\nabla_b^2 L = \nabla_b(\nabla_b L) = \nabla_b \left(-\frac{1}{N} \sum_{n \in N} E_{x_1^{n+1}} \left[x_{n+1} - f_\theta(x_1^n) \right] \right)$$

$$= \frac{1}{N} \sum_{n \in N} E_{x_1^{n+1}} \Big[f_\theta(x_1^n)\big(1 - f_\theta(x_1^n)\big) \cdot \nabla_b (a^\top z_n + b) \Big]_{\theta=\theta_\pi} = \frac{1}{N} \sum_{n \in N} E(\pi_1 \pi_0) = \pi_1 \pi_0.$$

$$\nabla_a^2 L = \nabla_a(\nabla_a L) = \nabla_a \left(-\frac{1}{N} \sum_{n \in N} E_{x_1^{n+1}} \left[(x_{n+1} - f_\theta(x_1^n)) \cdot \nabla_a (a^\top z_n) \right] \right)$$

$$= \nabla_a \left(-\frac{1}{N} \sum_{n \in N} E_{x_1^{n+1}} \left[(x_{n+1} - f_\theta(x_1^n))(z_n + (\nabla_a z_n)\, a) \right] \right)$$

$$= \frac{1}{N} \sum_{n \in N} E_{x_1^{n+1}} \Big[f_\theta(x_1^n)\big(1 - f_\theta(x_1^n)\big)(z_n + (\nabla_a z_n) a)(z_n + (\nabla_a z_n) a)^\top \Big]$$

$$-\frac{1}{N} \sum_{n \in N} E_{x_1^{n+1}} \left[(x_{n+1} - f_\theta(x_1^n))(2 \nabla_a z_n) \right]$$

$$z_n = \alpha x_{n-1} + \beta x_n + \xi \quad \Rightarrow \quad \nabla_a z_n = 0$$

$$= \frac{1}{N} \sum_{n \in N} E\big(\pi_0 \pi_1 z_n z_n^\top\big)$$

$$\pi_1 = \frac{p}{p+q}, \qquad \pi_0 = \frac{q}{p+q}$$

$$\pi_{00} = \pi_0(1-p) = \frac{q(1-p)}{p+q}, \qquad \pi_{01} = \frac{pq}{p+q}, \qquad \pi_{10} = \frac{pq}{p+q}, \qquad \pi_{11} = \frac{p(1-q)}{p+q}$$

x_{n-1}	x_n	$\pi_0 \pi_1 z_n z_n^\top$
0	0	$[\pi_0 \pi_1 (\xi \xi^\top)]$
0	1	$[\pi_0 \pi_1 (\beta + \xi)(\beta + \xi)^\top]$
1	0	$[\pi_0 \pi_1 (\alpha + \xi)(\alpha + \xi)^\top]$
1	1	$[\pi_0 \pi_1 (\alpha + \beta + \xi)(\alpha + \beta + \xi)^\top]$

$$= (\pi_0 \pi_1) \Big[\pi_{00} \, (\xi \xi^\top) + \pi_{01} \, (\beta + \xi)(\beta + \xi)^\top + \pi_{10} \, (\alpha + \xi)(\alpha + \xi)^\top + \pi_{11} \, (\alpha + \beta + \xi)(\alpha + \beta + \xi)^\top \Big]$$

$$H_\alpha \triangleq \nabla_\alpha^2 L(\alpha_\pi, \beta_\pi) = \pi_0 \pi_1 \begin{bmatrix} 1 & u^\top \\ u & v \end{bmatrix}$$

$$u = \nabla_{ab} L, \qquad v = \nabla_b^2 L$$

For the Hessian with respect to β : $H_\beta \triangleq \nabla_\beta^2 L(\alpha_\pi, \beta_\pi) = 0$?

$$\beta = \big(\{p_n\}_{n=1}^N, \, W_{K,Q,V} \big) \;\; \Rightarrow \;\; z \text{ is independent of all these } \Rightarrow \; \nabla_\beta^2 L = 0$$

$$H_{\alpha\beta} = \nabla_{\alpha\beta}^2 L = 0 \; \Rightarrow \; H_{\beta\alpha} = 0$$

$$H_\pi = \nabla^2 L(\theta_\pi) = \begin{bmatrix} H_\alpha & 0 \\ 0 & 0 \end{bmatrix}, \qquad \text{if } H_\alpha \text{ is PSD } \Rightarrow \text{ local min}$$

$$H_\alpha = \begin{bmatrix} 1 & u^\top \\ u & v \end{bmatrix}$$

Definition of weighted covariance matrix: $\sum_i \omega_i x_i x_i^\top - \Big(\sum_i \omega_i x_i\Big)\Big(\sum_i \omega_i x_i\Big)^\top \Rightarrow$ PSD

Based on this definition, we see the determinant of the Hessian matrix shows exactly the same formula, therefore it's PSD.

$$v - uu^\top = \Big(\pi_{00}(\xi\xi^\top) + \pi_{01}(\beta+\xi)(\beta+\xi)^\top + \pi_{10}(\alpha+\xi)(\alpha+\xi)^\top + \pi_{11}(\alpha+\beta+\xi)(\alpha+\beta+\xi)^\top\Big)$$

$$-\Big[\pi_{00}\xi + \pi_{01}(\beta+\xi) + \pi_{10}(\alpha+\xi) + \pi_{11}(\alpha+\beta+\xi)\Big]\Big[\pi_{00}\xi + \pi_{01}(\beta+\xi) + \pi_{10}(\alpha+\xi) + \pi_{11}(\alpha+\beta+\xi)\Big]^\top$$

Now it remains to show that θ_π is a local minimum.

***H*$_\alpha$ is positive-definite implies θ_π is a local minimum**. Since $H_\alpha \succ 0$, let $H_\alpha \succ \lambda I$ for some $\lambda > 0$s. Since $\theta = (\alpha, \beta) \in \mathbb{R}^{D-d}$, interpret $L(\theta) = L(\alpha, \beta)$ as a function of two variables α and β with appropriate dimensions. We know the following facts about $L(\cdot, \cdot)$:

- Fact 1. $\alpha \mapsto L(\alpha, \beta_\pi)$ has a local minimum (as a function of one variable) at $\alpha = \alpha_\pi$ (since $H_\alpha \succ 0$).
- Fact 2. $\beta \mapsto L(\alpha, \beta)$ is constant in β (since $\alpha_\pi = 0$, the probability $f_\theta(x_1^\pi)$ is constant w.r.t. z_n and hence w.r.t. β (Linear)).
- Fact 3. $\nabla L(\alpha_\pi, \beta_\pi) = 0$ and $H_\pi = \nabla^{(2)} L(\alpha_\pi, \beta_\pi) = \begin{bmatrix} H_\alpha & 0 \\ 0 & 0 \end{bmatrix}$ with $H_\alpha \succ \lambda I$.

Using these facts, we now show that $(\alpha_\pi, \beta_\pi) = \theta_\pi$ is also a local minimum in two variables. We prove this by contradiction. Suppose that (α_π, β_π) is not a local minimum for $L(\cdot, \cdot)$. Without loss of generality, by a shift of coordinates treat θ_π as the origin, i.e., $(\alpha_\pi = 0, \beta_\pi = 0)$ is not a local minimum for $L(\cdot, \cdot)$. Then there exists a unit direction $d = (u, v) \in \mathbb{R}^{D-d}$ with $\|d\|^2 = \|u\|^2 + \|v\|^2 = 1$ and a $0 < \varepsilon_0 < 1$ such that

$$L(\varepsilon d) < L(0), \quad \forall 0 < \varepsilon \leq \varepsilon_0.$$

Clearly $\|u\| > 0$, otherwise it will contradict Fact 2. Using the definition of directional-derivative, we have that

$$\begin{aligned} d^T \nabla^{(2)} L(0,0) d &= \lim_{\varepsilon \to 0^+} \frac{\langle \nabla L(\varepsilon d), d \rangle - \langle \nabla L(0), d \rangle}{\varepsilon} \\ &= \lim_{\varepsilon \to 0^+} \frac{\langle \nabla L(\varepsilon d), d \rangle}{\varepsilon}. \end{aligned}$$

On the other hand, using the Hessian structure, the LHS equals

$$d^T \nabla^{(2)} L(0,0) d \geq \lambda \|u\|^2 \triangleq K_1 > 0.$$

Thus

$$\lim_{\varepsilon \to 0^+} \frac{\langle \nabla L(\varepsilon d), d \rangle}{\varepsilon} = K_1 > 0.$$

Thus, there exists an $\varepsilon_1 > 0$ and $K > 0$ such that

$$\frac{\langle \nabla L(\varepsilon d), d \rangle}{\varepsilon} \geq K, \quad \forall 0 < \varepsilon \leq \varepsilon_1.$$

which implies

$$\langle \nabla L(\varepsilon d), d \rangle \geq K\varepsilon, \quad \forall 0 < \varepsilon \leq \varepsilon_1.$$

Defining the function $g : \mathbb{R}_+ \to \mathbb{R}$ as $g(\varepsilon) = L(\varepsilon d)$, we obtain that $g'(\varepsilon) = \langle \nabla L(\varepsilon d), d \rangle \geq K\varepsilon$ for $0 \leq \varepsilon \leq \varepsilon_1$. Using the fundamental theorem of Calculus, we have that for any $0 \leq \varepsilon \leq \varepsilon_1$,

$$g(\varepsilon) - g(0) = \int_0^\varepsilon g'(t) dt \geq \int_0^\varepsilon K t dt = \frac{K\varepsilon^2}{2}.$$

Thus $g(\varepsilon) \geq L(0) + \frac{K\varepsilon^2}{2}$ for all $0 \leq \varepsilon \leq \varepsilon_1$, whereas $L(\varepsilon d) < L(0)$ for all $0 < \varepsilon < \varepsilon_0$ from Eq. (13). Choosing $\varepsilon_* = \min(\varepsilon_0, \varepsilon_1)$, we have a contradiction for $0 < \varepsilon < \varepsilon_*$. Thus $0 \equiv \theta_\pi = (\alpha_\pi, \beta_\pi)$ is a local minimum. □

3.4.7 Second Order MC

In this section, we assume the input data comes from a second-order Markov chain. For the embedding layer, we follow the idea proposed in [7] for the proof of their Theorem 4.1. Also we only consider one layer of attention (L=1), so based on the same idea from [7], the output of the model after the residual layer is as follows:

$$x_n^{(1)} = \text{Emb}(x_n) = \kappa \begin{bmatrix} 1_{1\times 2} & e_{x_n}^S & 0_{1\times 2S} \end{bmatrix}^T \in \mathbb{R}^d.$$

for a constant $\kappa > 0$ to be chosen later and $d = 2S + 2$. The relative position encodings will essentially be supported on the first two coordinates, the middle S coordinates are a one-hot encoding of the symbol x_n and the last $2S$ coordinates are 0. The relative position encodings in the first layer are chosen to be $p_{n-i}^{(1),K} = \kappa(-1 + \mathbb{I}(n - i = 1))\, e_1^d \in \mathbb{R}^d$ and $p_{n-i}^{(1),V} = 0 \in \mathbb{R}^d$. Choose $W_K^{(1)}$ and $W_Q^{(1)}$ to be $e_1^d (e_1^d)^T \in \mathbb{R}^{d\times d}$. With this choice,

$$\left\langle W_K^{(1)} \left(x_i^{(1)} + p_{n-i}^{(1),K} \right), W_Q^{(1)} x_n^{(1)} \right\rangle = \kappa \mathbb{I}(n - i = 1)$$

As $\kappa \to \infty$, the attention pattern (which takes the softmax over of these inner products over $i \in [n]$) computes,

$$\text{att}_{n,i}^{(1)} = \mathbb{I}(i = n - 1)$$

for any $n > 1$. Choose the value matrix as,

$$W_V^{(1)} = \begin{bmatrix} 0_{(2+S)\times 2} & 0 & 0 \\ 0 & I_{S\times S} & 0 \\ 0 & 0 & 0 \end{bmatrix} \in \mathbb{R}^{d\times d}$$

And with this choice and the residual connection, we get,

$$x_n^{(2)} = \kappa \begin{bmatrix} 1_{1\times 2} & e_{x_n}^S & e_{x_{n-1}}^S & 0 \end{bmatrix} \in \mathbb{R}^d$$

which serves as the input to the 2nd transformer layer. Here, since we are studying binary tokens, the vocabulary size (S) is 2.

$$e_0 = [1 \;\; 0], \qquad e_1 = [0 \;\; 1],$$

$$x_n^{(1)} = \mathrm{Emb}(x_n) = k \begin{bmatrix} 1 \\ 1 \\ e_{x_n} \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}^\top, \qquad d = 8$$

$$x_n^{(2)} = k \begin{bmatrix} 1 \\ 1 \\ e_{x_n} \\ e_{x_{n-1}} \\ 0 \\ 0 \end{bmatrix}^\top$$

We can write $x_n^{(2)}$ as a function of x_n and x_{n-1}:

$$x_n^{(2)} = \beta\, x_n + \alpha\, x_{n-1} + \xi.$$

$$\beta = \begin{bmatrix} 0 \\ 0 \\ -1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \qquad \alpha = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ -1 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \qquad \xi = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Given these assumptions, this part is very similar to the other parts previously discussed. Lemma 1 is true the same way as Section 3.4.3, and Lemma 2 is also the same as Section 3.4.4. For the proof of Theorem 1, the proof is very similar to Section 3.4.5, but here we have:

$$\sigma\left(ax_n^2 + b\right) = x_{n-1}\left(x_n t + (1 - x_n)s\right) + (1 - x_{n-1})\left(x_n q + (1 - x_n)p\right)$$

Therefore, the final value of θ_* should satisfy the following conditions:

$$\begin{cases} a\xi + b = -\log\left(\dfrac{1}{p} - 1\right) \\ a(\xi + \beta) + b = -\log\left(\dfrac{1}{q} - 1\right) \\ a(\xi + \alpha) + b = -\log\left(\dfrac{1}{s} - 1\right) \\ a(\xi + \alpha + \beta) + b = -\log\left(\dfrac{1}{t} - 1\right) \end{cases}$$

Theorem 2 is also true with the same definition as in Section 3.3.5. The proof is a combination of the approaches in Sections 3.3.5 and 3.4.6.

Chapter 4

Observations

4.1 Experiments on Learning Second-Order Markov Structure

This chapter presents a detailed empirical investigation of how transformer-based sequence models learn higher-order dependencies in Markovian data. While recent theoretical work has clarified that transformers have sufficient expressive capacity to represent higher-order Markov structure, considerably less is known about the conditions under which they successfully *learn* such structure in practice. We focus on the second-order Markov case, the simplest setting in which the conditional distribution depends on more than one previous token.

Our work extends the analysis of Section 4 in [**?**]. The authors investigate transformers trained on a special form of second-order Markov data produced by interleaving two independent first-order chains. They show that although transformers can theoretically represent such dependencies, learning fails when the model is given a full causal attention mask; the model collapses to predicting the stationary distribution. When the attention window is restricted to match the true second-order structure (e.g., window $W = 2$), the model correctly learns the underlying transition probabilities.

In contrast, we study transformers trained on a *general* second-order Markov chain whose conditional distribution genuinely depends on the ordered pair (X_{t-1}, X_{t-2}). This setting removes the structural simplification of the interleaving construction and provides a more representative test of transformer learning on higher-order temporal dependencies. Our experiments confirm that attention window size plays a critical role: when the win-

dow matches the true Markov order, the model successfully learns the transition kernel, whereas large windows produce unstable predictions that fail to track the true conditional probabilities. Importantly, and unlike [**?**], our large-window experiments do **not** converge to stationary probabilities; instead, they exhibit persistent instability.

4.1.1 Summary of Prior Results (Section 4 of [9])

Section 4 of [9] constructs a second-order chain by alternating tokens from two independent first-order chains. This yields a sequence where

$$X_{t+1} \text{ depends only on } X_{t-1},$$

a highly structured but technically second-order dependency.

Their findings include:

- With a full causal mask, learning fails and the model converges to the stationary distribution.
- With a short window (e.g., $W = 2$), the model correctly learns the true transition probabilities.

This shows that excessively long attention windows introduce harmful interference, preventing the transformer from isolating the relevant context needed for correct prediction.

4.1.2 General Second-Order Markov Chains

To study a more realistic setting, we consider a true second-order Markov chain where the distribution of X_{t+1} depends on the ordered pair (X_t, X_{t-1}). The state space is

$$\{00, 01, 10, 11\},$$

and the transition matrix is:

$$T = \begin{pmatrix} 0.8 & 0.2 \\ 0.1 & 0.9 \\ 0.2 & 0.8 \\ 0.4 & 0.6 \end{pmatrix}.$$

Each row corresponds to transitions out of the states $00, 01, 10$, and 11 respectively.

This model exhibits:

- multiple distinct conditional patterns,
- non-redundant use of both previous tokens, and
- heterogeneity across contexts.

4.1.3 Stationary Distribution

Solving $\pi T = \pi$, we obtain the stationary distribution:

$$\pi = \left(\tfrac{4}{21}, \tfrac{4}{21}, \tfrac{4}{21}, \tfrac{9}{21}\right) \approx (0.1905, 0.1905, 0.1905, 0.4286).$$

The marginal stationary distribution is:

$$P(X_t = 0) = \tfrac{8}{21} \approx 0.381, \qquad P(X_t = 1) = \tfrac{13}{21} \approx 0.619.$$

These values are *not* the convergent behavior in our experiments, but they serve as useful reference points when evaluating deviations from the true transition structure.

4.1.4 Experimental Configuration

We train a single-layer decoder-only transformer using the configuration below. The key hyperparameters are:

- batch size: 16, sequence length: 200,
- optimizer: AdamW with learning rate 0.01,
- number of heads: 1, embedding dimension: 8,
- iteration count: 200, cosine scheduler,
- vocabulary size: 2, dataset drawn from the second-order chain,

All experiments are repeated across multiple random seeds for robustness.

4.1.5 Empirical Results: Effect of Attention Window Size

To study how the memory window influences learning, we evaluate the transformer's predicted probability $P(X_{t+1} = 1)$ under a fixed context and track it across sequence positions. Figure 4.1 shows results for the context $(X_{t-2}, X_{t-1}) = (0, 1)$.

We compare memory windows:

$$W \in \{2, 4, 40, 80, 100\}.$$

Memory $= 2$ (matches true Markov order). The model converges to the true transition probability for the context. The predictions are stable and closely match the corresponding entry in T. This confirms that transformers can correctly learn second-order dependencies when the attention span matches the true memory of the process.

Memory $= 4$. Even a slight increase beyond the true dependency length introduces moderate instability. Predictions fluctuate across positions but still remain roughly near the correct value.

Memory $= 40, 80, 100$. For large windows, instability becomes severe: predictions oscillate widely and show no consistent trend toward the true transition probability. Unlike the collapse observed in [**?**], the model does *not* converge to the stationary distribution; instead, it produces highly variable outputs influenced by irrelevant historical tokens.

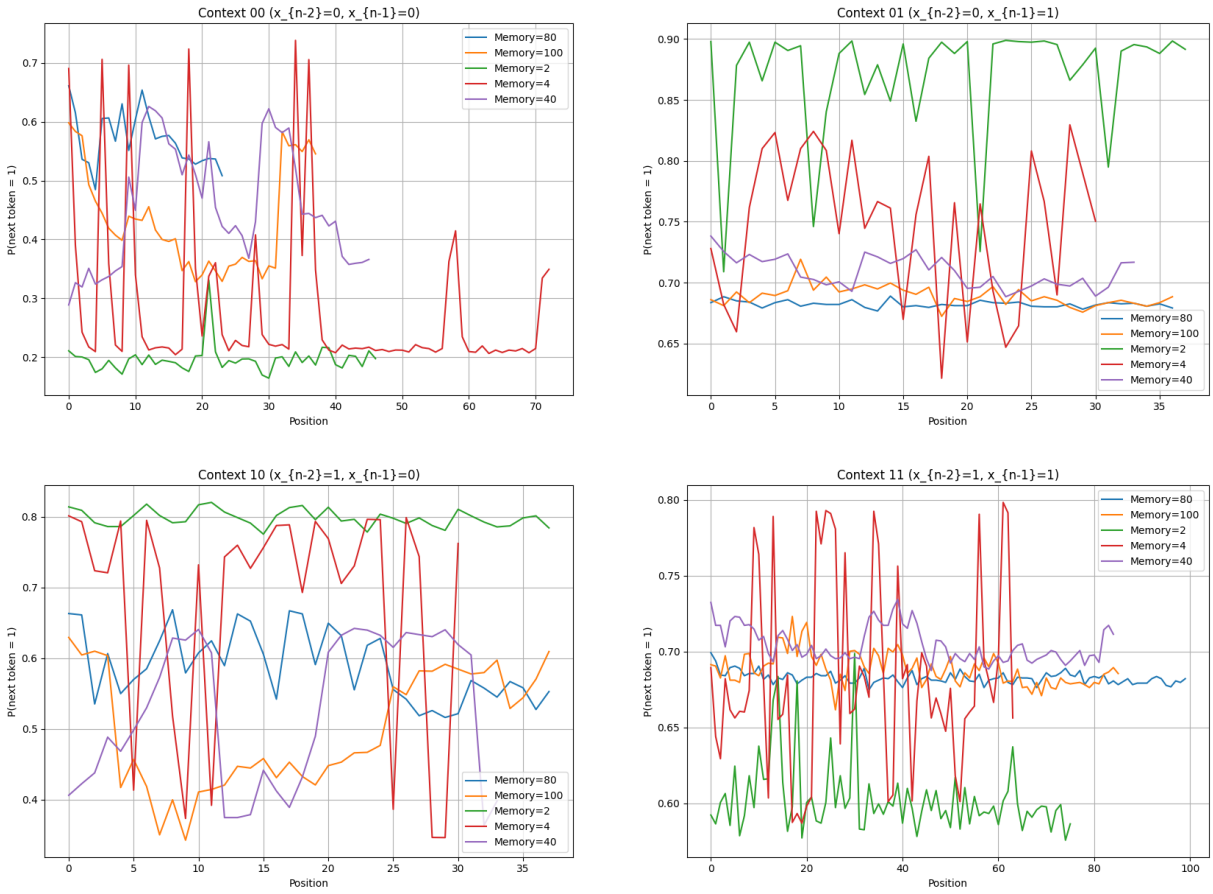


Figure 4.1: Predicted probability $P(X_{t+1} = 1)$ under each second-order context $(X_{t-2}, X_{t-1}) \in \{00, 01, 10, 11\}$ for different memory window sizes. Each panel corresponds to one context (00, 01, 10, or 11). For short windows (particularly memory = 2), the curves closely track the appropriate entry in the second column of the transition matrix T, indicating that the model correctly learns the second-order transition probabilities. For larger windows, the predicted probabilities fluctuate substantially across positions and deviate from the true values.

Similar patterns arise across all four contexts $(00, 01, 10, 11)$: small windows recover the true transitions, while large windows produce unstable behavior.

4.1.6 Discussion

Our experiments align with and extend the insights of [9]. In the special interleaving second-order setting, large windows cause the model to collapse to the stationary distribution. In our general second-order chain:

- Small windows (matching the true order) lead to successful learning of the transition kernel.
- Larger windows do not produce stationary collapse but instead generate substantial instability and failure to reproduce the correct context-specific probabilities.
- Thus, excessive context harms optimization in both settings, but the failure modes differ.

Overall, our findings highlight that correct learning of higher-order dependencies requires careful alignment between model architecture (attention span) and data structure (Markov order). Transformers benefit from—but also depend on—precisely constrained context windows when learning multi-step dependencies.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

The goal of this thesis was to deepen our theoretical understanding of how transformers learn from structured sequential data—particularly when the underlying process exhibits memory beyond the immediate past. While transformers have demonstrated remarkable empirical success, their behavior under controlled and mathematically well-defined input distributions remains only partially understood. Prior theoretical work provided a complete characterization of the loss landscape for first-order Markov chains, but real-world sequences often involve richer forms of dependency. This gap motivated the core contributions of the present work.

In this thesis, we showed that the full set of theoretical results established for first-order Markov chains also extends to second-order Markov chains. This includes explicit constructions of global minima that recover the true second-order transition kernel, as well as the identification of bad local minima that arise under weight tying, where the model collapses to predicting stationary-distribution solutions. We further demonstrated that these undesirable local minima disappear once weight tying is removed, and the model becomes able to represent and learn the correct transition dynamics under more flexible parameterizations. These findings closely parallel the phenomena observed in the first-order setting, indicating that the geometry and challenges of the transformer loss landscape persist under higher-order dependencies.

We then explored whether these theoretical patterns persist in an alternative architectural setting: attention-only transformers. Somewhat remarkably, we found that the same

global optima and failure modes appear in this architecture as well, suggesting that the behaviors studied in this thesis reflect deeper interactions between attention mechanisms and Markovian structure rather than artifacts of a particular modeling choice.

Finally, we validated our theoretical predictions using controlled experiments on synthetic second-order Markov chains. The empirical findings reinforce our theoretical analysis: when the transformer's attention window aligns with the true order of the chain, the model correctly learns the second-order transition structure, but when the window is unnecessarily large, the model's predictions drift toward averaged or stationary-like behaviors. This provides an intuitive explanation for why higher-order dependencies can be more fragile in practice, despite the architecture being expressive enough to represent them.

Taken together, these contributions provide a more comprehensive theoretical foundation for understanding how transformers learn from sequential structure. By extending first-order results to the second-order setting—and confirming these patterns in both standard and attention-only architectures—this thesis advances the mathematical understanding of transformer behavior on sequential data.

5.2 Future Work

The results presented in this thesis open several natural extensions for future research. A first direction involves studying transformer behavior on Markov chains of order higher than two. While this thesis establishes the theoretical guarantees for second-order processes, extending these analyses to general k-order Markov chains would provide a broader understanding of how the loss landscape scales with the memory of the underlying data-generating process. This would clarify whether the same forms of global minima and undesirable local minima continue to appear as the chain order grows.

A second promising direction is to examine more expressive transformer architectures. In this work, several architectural components were intentionally simplified: we used single-head attention, a minimal feedforward layer, and in some cases set specific parameters to zero to make the analysis tractable. Future studies could incorporate

- multi-head attention mechanisms,
- deeper or wider feedforward networks,
- more flexible embedding structures, and

- relaxed assumptions on parameters that were fixed or removed in our setting.

Exploring how these richer architectural elements reshape the loss landscape may reveal new behaviors or eliminate certain failure modes observed under simplified assumptions.

Another direction is to investigate the sensitivity of the loss landscape with respect to architectural hyperparameters such as depth, dimensionality, activation functions, and positional encodings. Since this thesis focused on a controlled and restricted parameterization, studying how these additional choices influence convergence and representation could yield practically relevant insights.

Finally, moving beyond Markovian input data presents an important opportunity for future work. Markov chains provide a clean probabilistic structure that enables exact theoretical analysis, but many real-world sequences exhibit dependencies that are non-Markovian or long-range. Future research could examine transformer behavior on more general sequential sources, such as renewal processes, hidden-state models, long-memory processes, or even empirically derived datasets. Understanding whether global minima, local minima, or other forms of structured critical points arise under these alternative input distributions would help bridge the gap between theoretical models and practical transformer applications.

Overall, these directions offer valuable opportunities to generalize, enrich, and test the theoretical framework developed in this thesis.

References

[1] Siyu Chen, Heejune Sheen, Tianhao Wang, and Zhuoran Yang. Training dynamics of multi-head softmax attention for in-context learning: Emergence, convergence, and optimality. *arXiv preprint arXiv:2402.19442*, 2024.

[2] Benjamin L. Edelman. A probabilistic approach to the theoretical analysis of transformers on markovian data. In *Advances in Neural Information Processing Systems*, 2020.

[3] Benjamin L. Edelman, Ezra Edelman, Surbhi Goel, Eran Malach, and Nikolaos Tsilivis. The evolution of statistical induction heads: In-context learning markov chains. In *Advances in Neural Information Processing Systems*, 2024. NeurIPS 2024.

[4] Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Jones, Tom Kernion, Ben Mann, Amanda Askell, Anna Chen, David Krueger, Jared Kaplan, Samuel R. Bowman, Dario Amodei, Sam McCandlish, and Jacob Steinhardt. A mathematical framework for transformer circuits. *arXiv preprint arXiv:2104.08696*, 2021.

[5] Shivam Garg, Dimitris Tsipras, Percy Liang, and Gregory Valiant. What can transformers learn in-context? a case study of simple function classes. *arXiv preprint arXiv:2208.01066*, 2022.

[6] John Hewitt and Christopher D. Manning. A structural probe for finding syntax in word representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT)*, pages 4129–4138, 2019.

[7] Fangyuan Liu, Simon S. Du, Rong Ge, Runzhe Wang, and Yisen Wei. Transformers on markov data: Constant depth suffices. *arXiv preprint arXiv:2407.17686*, 2024.

[8] Ashok Vardhan Makkuva, Marco Bondaschi, Chanakya Ekbote, Adway Girish, Alliot Nagle, Hyeji Kim, and Michael Gastpar. Local to global: Learning dynamics and effect of initialization for transformers. *arXiv preprint arXiv:2406.03072*, 2024.

[9] Ashok Vardhan Makkuva, Marco Bondaschi, Adway Girish, Alliot Nagle, Martin Jaggi, Hyeji Kim, and Michael Gastpar. Attention with markov: A framework for principled analysis of transformers via markov chains. *arXiv preprint arXiv:2402.04161*, 2024.

[10] J. R. Norris. *Markov Chains*. Cambridge University Press, 1997.

[11] Maithra Raghu, Ben Poole, Jon Kleinberg, Surya Ganguli, and Jascha Sohl-Dickstein. On the expressive power of deep neural networks. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, 2017.

[12] Nived Rajaraman, Jiantao Jiao, and Kannan Ramchandran. Toward a theory of tokenization in large language models. *arXiv preprint arXiv:2404.08335*, 2024.

[13] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017.

[14] Johannes von Oswald, Giambattista Parascandolo, Evgeny Burnaev, Dominik Grewe, Christian Henning, Jonathan A. Schwarz, Nikita Galashov, Sebastian Flennerhag, Yee Whye Teh, Laurence Aitchison, and Simon Osindero. Transformers learn in-context by gradient descent. *arXiv preprint arXiv:2212.07677*, 2022.

[15] Yair Weiss. Practical approaches to higher-order markov models. *Journal of Machine Learning Research*, 19:1–37, 2018.