# The Evolution of Statistical Induction Heads: In-Context Learning Markov Chains

**Anonymous Author(s)**
Affiliation
Address
email

## Abstract

Large language models have the ability to generate text that mimics patterns in their inputs. We introduce a simple Markov Chain (MC) sequence modeling task in order to study how this in-context learning (ICL) capability emerges. Transformers trained on this task (ICL-MC) form *statistical induction heads* which compute accurate next-token probabilities given the bigram statistics of the context. During the course of training, models pass through multiple phases: after an initial stage in which predictions are uniform, they learn to sub-optimally predict using in-context single-token statistics (unigrams); then, there is a rapid phase transition to the correct in-context bigram solution. We conduct an empirical and theoretical investigation of this multi-phase process, showing how successful learning results from the interaction between the transformer's layers, and uncovering evidence that the presence of simpler solutions delays formation of the final optimal solutions.

## 1 Introduction

Large language models (LLMs) exhibit a remarkable ability to perform *in-context learning* (ICL) from patterns in their input context [10, 14]. The ability of LLMs to adaptively learn from context is profoundly useful, yet the underlying mechanisms of this emergent capability are not fully understood.

In an effort to better understand ICL, some recent works propose to study ICL in controlled synthetic settings—in particular, training transformers on mathematically defined tasks which require learning from the input context. For example, a recent line of works studies the ability of transformers to perform ICL of standard supervised learning problems such as linear regression [2, 18, 23, 35]. Studying these well-understood synthetic learning tasks enables fine-grained control over the data distribution, allows for comparisons with established supervised learning algorithms, and facilitates the examination of the in-context "algorithm" implemented by the network.

The goal of this work is to propose and analyze a simple synthetic setting for studying ICL. To achieve this, we consider $n$-gram models [9, 13, 32], one of the simplest and oldest methods for language modeling. An $n$-gram language model predicts the probability of a token based on the preceding $n-1$ tokens, using fixed-size chunks ($n$-grams) of text data to capture linguistic patterns. Our work studies ICL of $n$-gram models, where the network needs to compute the conditional probability of the next token based on the statistics of the tokens observed in the input context, rather than on the statistics of the entire training data. We mainly focus on the simple case of $n = 2$; i.e., bigram models, which can be represented as Markov chains. We therefore consider ICL of Markov chains (ICL-MC): we train a two layer attention-only transformer on sequences of tokens, where each sequence is produced by a different Markov chain, generated using a different transition matrix (see Figure 1 (left)).

We summarize our key findings:

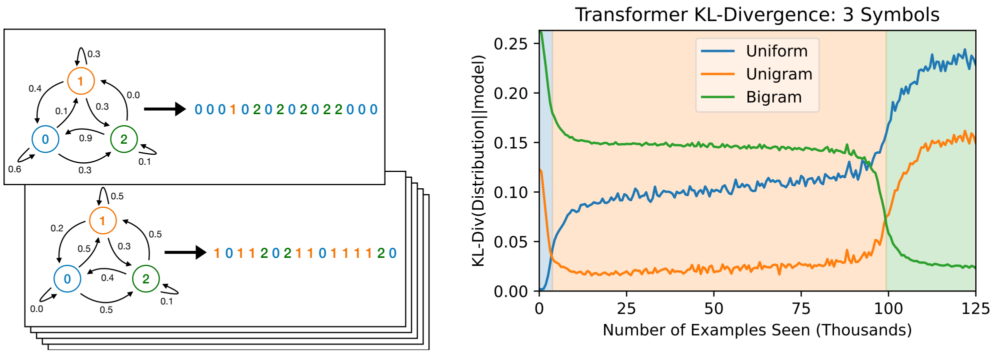


Figure 1: (*left*) We train small transformers to perform in-context learning of Markov chains (ICL-MC). Each training sequence is generated by sampling a transition matrix from a prior distribution, and then sampling a sequence from this Markov chain. (*right*) Distance of a transformer's output distribution to several well-defined strategies over the course of training. The model passes through three stages: (1) predicting a uniform distribution (*blue* region), (2) predicting based on in-context unigram statistics (*orange* region), (3) predicting based on in-context bigram statistics (*green* region).

**(1) Transformers learn statistical induction heads to optimally solve ICL-MC.** We show that in order to solve ICL-MC, transformers learn *statistical* induction heads [16] that are able to compute the correct *conditional (posterior) probability* of the next token given all previous occurrences of the prior token (see attention patterns in Figure 4). We show that these statistical induction heads lead to the transformer achieving performance approaching that of the Bayes-optimal predictor.

**(2) Transformers learn predictors of increasing complexity and undergo a phase transition when increasing complexity.** We observe that transformers display *phase transitions* when learning Markov chains—learning appears to be separated into phases, with fast drops in loss between the phases. We are able to show that different phases correspond to learning models of increased complexity—unigrams, then bigrams (see Figure 1)—and characterize the transition between the phases.

**(3) Simplicity bias may slow down learning.** We provide evidence that the model's inherent bias towards simpler solutions (in particular, in-context unigrams) causes learning of the optimal solution to be delayed. Changing the distribution of the in-context examples to remove the usefulness of in-context unigrams leads to faster convergence, even when evaluated on the original distribution.

**(4) Alignment of layers is crucial.** We show that the transition from a phase of learning the simple-but-inadequate solution to the complex-and-correct solution happens due to an alignment between the layers of the model: the learning signal for the first layer is tied to the extent to which the second layer approaches its correct weights.

Finally, in Appendix E we provide experiments with higher order Markov Chains, where we also observe a similar multi-stage learning process.

**Concurrent work.** In parallel to this work, there have been a number of papers devoted to the study of similar questions regarding in-context learning of Markov chains [3, 19, 25]. Perhaps closest to our work, [27] introduces a general family of in-context learning tasks with causal structure, a special case of which is in-context Markov chains, and shows that simplified transformers (similar to the ones we introduce in Section B.2) can learn to identify the causal relationships. The focus of our work, instead, is on the different stages of training and how they relate to specific, well-defined, strategies. See Section A for a detailed discussion on prior work.

## 2 Setup

**ICL-MC Task.** Our learning task consists of sequences generated from Markov Chains with random transition matrices. The goal is to in-context estimate the transition probabilities from

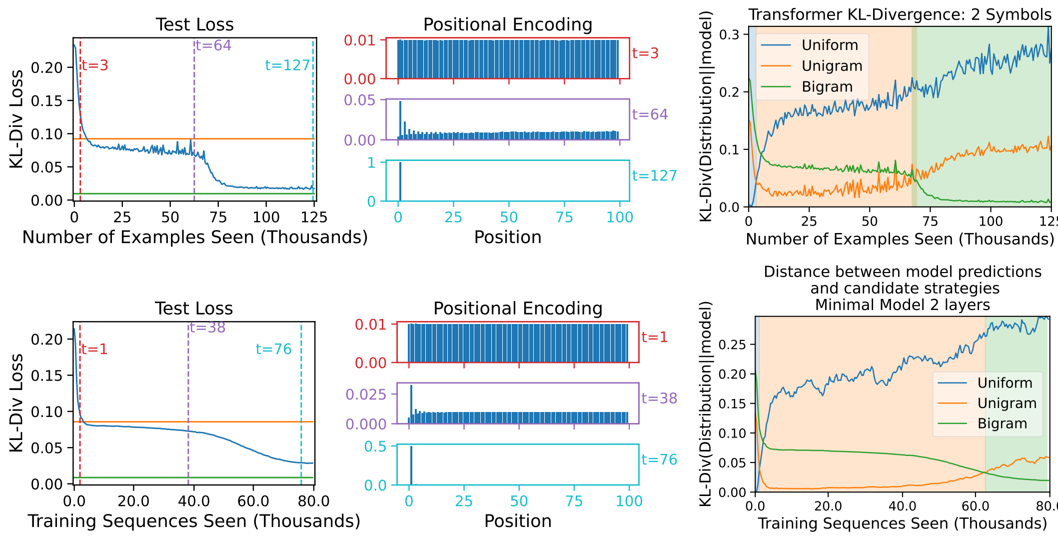


Figure 2: A two layer transformer (*top*) and a minimal model (*bottom*) trained on our in-context Markov Chain task. A comparison of the two layer attention-only transformer and minimal model (4). The graphs on the left are test loss measured by KL-Divergence from the underlying truth. The orange line shows the loss of the unigram strategy, and the green line shows the loss of the bigram strategy. The middle graph shows the effective positional encoding (for the transformer, these are for the first layer). The graph on the right shows the KL-divergence between the outputs of the models and three strategy. The lower the KL-divergence, the more similar the model is to that strategy.

sampled sequences, in order to predict the next state. Formally, each sample sequence is generated by a Markov Chain with state space $S = \{1, \ldots, k\}$ and a transition matrix $\mathcal{P}$ sampled from a prior distribution, with $x_1$ drawn from some other prior distribution (potentially dependent on $\mathcal{P}$), and the rest of $\boldsymbol{x} = (x_1, \ldots, x_t)$ drawn from the Markov Chain. We focus on the case where each row of the matrix is sampled from the Dirichlet distribution with concentration parameter $\boldsymbol{\alpha}$, i.e. $\mathcal{P}_{i,:} \sim \mathrm{Dir}(\boldsymbol{\alpha})$. We want to learn a predictor that, given context $x_1, \ldots, x_t$, predicts the next token, $x_{t+1}$.

**Strategies.** We consider two particular strategies that can be employed to solve the above task: a (suboptimal) *unigram* strategy which assumes tokens in each sequence are i.i.d. samples (and counts the frequency of the states in the sequence so far), and the *bigram* strategy which correctly takes into account dependencies among adjacent tokens (and counts frequency of pairs of tokens). See Section B in the Appendix for a detailed description of our learning setup.

# 3 Empirical Findings and Theoretical Validation

In this section, we present our empirical findings on how transformers succeed in in-context learning Markov Chains, we demonstrate the different learning stages during training and the sudden transitions between them, and draw analytical and empirical insights from a minimal model that we believe captures the behavior of transformers for this task.

## 3.1 Transformers In-Context Learn Markov Chains Hierarchically

We focus on attention-only transformers with 2 layers with causal masking and relative positional encodings and train them with the Adam optimizer on ICL-MC. As can be seen in Figure 2, all the models converge near the Bayes optimal solution, suggesting that they learn to implement the bigram strategy. Curiously, however, the learning seems to be happening in stages; there is an initial rapid drop and the model quickly finds a better than random solution. Afterwards, there is a long period of only slight improvement before a second rapid drop brings the model close to the Bayes optimal loss.

Interestingly, as can be seen from the horizontal lines in Figure 2, the intermediate plateau corresponds to a phase when the model reaches the unigram baseline. We provide evidence that this is not a coincidence, and that after the initial drop in loss, the model's strategy is very similar to the unigram strategy, before eventually being overtaken by the bigram strategy (see Figure 2). This final drop is what has been associated to prior work with *induction heads* formation [28]; special dedicated heads inside a transformer are suddenly being formed to facilitate in-context learning.

**Mechanistic evidence for solutions found by transformer.** To confirm how the two layer attention-only transformer solves ICL-MC, we inspected the attention in each layer throughout training. Figure 4 shows the attention for a particular input during different parts of training. We observe that, by the end of training, each token in the first layer is attending to the previous token. In the second layer, the last token, a "2", is attending to tokens that followed "2"s, allowing bigram statistics to be calculated. In Proposition B.2, we show how this behavior can be implemented in the transformer architecture.

**Varying the data distribution - Unigrams slow down learning.** Given the previous findings, one can ask the question: *is the unigram solution helpful for the eventual convergence of the model, or is it perhaps just a by-product of the learning procedure?* To answer these questions, we define distributions over Markov chains that are in between the distribution where unigrams is Bayes optimal, and the distribution where unigrams is as good as uniform. As we see in Figure 3, the transformers that are being trained on the distribution where there is no unigrams "signal" train much faster. It appears that this simplicity bias towards the unigrams solution actually slows down learning. See also Figure 10 in the Appendix that displays how the models perform on different parts of the distribution during training.

### 3.2 Theoretical Insights from the Minimal Model

We now provide theoretical insights on how training progresses stage by stage and how this is achieved by the synergy between the two layers. For this, we analyze the training dynamics of a minimal model which can be seen as a simplified 2-layer attention only transformer. Section D contains our main theoretical result. Here, we summarize our theoretical findings:

**Learning occurs in two phases.** Both in the theoretical and experimental models, training has two phases that work at very different speeds. The first phase is fast in both cases; in the theoretical setting, even a $O\left(\frac{1}{T}\right)$ step size is sufficient for learning the second layer. In the second phase, a much larger step size of $O(1)$ is needed in order to learn the positional encodings.

**Second layer is learned first.** It has been observed before in a similar bigram learning setting with a two-layer transformer that the model might be learning first the second layer [6]. We also make similar observations in our experiments with the minimal model and the transformers (see Figure 4). For the minimal model, the gradient calculations, clearly suggest that starting from a default initialization, it is only the second layer that quickly "picks up" the right solution.

**Even/odd pattern in positional encodings.** We notice in the experiments that the positional embeddings of the models displayed an intriguing even/odd oscillating pattern - see Figure 2 (*top, center*), Figure 3 (*right*). We believe that a careful analysis the gradient of $v$ in the second step will recover this pattern, which is likely related to the moments of the eigenvalues of the transition matrix.

## 4 Conclusion

In this work, we have introduced a simple learning problem which serves as a controlled setting for understanding in-context learning and the emergence of (statistical) induction heads. Through a combination of empirical investigation and theoretical analysis, we identify different stages during learning which we were able to precisely characterize. These validate similar observations from training large-scale language models.

It would be worthwhile to understand similar stage-wise learning with natural language data, and use insights from our minimal model to improve formation of induction heads. In particular, it would be great to understand if better data curriculum could remove the undesirable simplicity bias we observe from unigrams. Such simple but incomplete solutions may be commonplace in language modeling and other rich learning settings; for any such solution, one can ask to what extent its presence speeds up or slows down the formation of more complex circuits with higher accuracy.

## References


[1] Jacob D. Abernethy, Alekh Agarwal, Teodor V. Marinov, and Manfred K. Warmuth. A mechanism for sample-efficient in-context learning for sparse retrieval tasks. *CoRR*, abs/2305.17040, 2023.

[2] Ekin Akyürek, Dale Schuurmans, Jacob Andreas, Tengyu Ma, and Denny Zhou. What learning algorithm is in-context learning? investigations with linear models. *arXiv preprint arXiv:2211.15661*, 2022.

[3] Ekin Akyürek, Bailin Wang, Yoon Kim, and Jacob Andreas. In-context language learning: Architectures and algorithms. *CoRR*, abs/2401.12973, 2024. doi: 10.48550/ARXIV.2401.12973. URL `https://doi.org/10.48550/arXiv.2401.12973`.

[4] Jimmy Ba, Murat A. Erdogdu, Taiji Suzuki, Zhichao Wang, Denny Wu, and Greg Yang. High-dimensional asymptotics of feature learning: How one gradient step improves the representation. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022.

[5] Boaz Barak, Benjamin Edelman, Surbhi Goel, Sham Kakade, Eran Malach, and Cyril Zhang. Hidden progress in deep learning: Sgd learns parities near the computational limit. *Advances in Neural Information Processing Systems*, 35:21750–21764, 2022.

[6] Alberto Bietti, Vivien Cabannes, Diane Bouchacourt, Herve Jegou, and Leon Bottou. Birth of a transformer: A memory viewpoint, 2023.

[7] Garrett Birkhoff. Three observations on linear algebra. *Univ. Nac. Tacuman, Rev. Ser. A*, 5: 147–151, 1946.

[8] Charles Bordenave, Pietro Caputo, and Djalil Chafai. Circular law theorem for random markov matrices. *Probability Theory and Related Fields*, 152, 08 2008.

[9] Peter F. Brown, Peter V. deSouza, Robert L. Mercer, Vincent J. Della Pietra, and Jenifer C. Lai. Class-based n-gram models of natural language. *Comput. Linguist.*, 18(4):467–479, dec 1992. ISSN 0891-2017.

[10] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[11] Stephanie Chan, Adam Santoro, Andrew Lampinen, Jane Wang, Aaditya Singh, Pierre Richemond, James McClelland, and Felix Hill. Data distributional properties drive emergent in-context learning in transformers. *Advances in Neural Information Processing Systems*, 35:18878–18891, 2022.

[12] Angelica Chen, Ravid Shwartz-Ziv, Kyunghyun Cho, Matthew L. Leavitt, and Naomi Saphra. Sudden drops in the loss: Syntax acquisition, phase transitions, and simplicity bias in mlms, 2023.

[13] Noam Chomsky. Three models for the description of language. *IRE Transactions on information theory*, 2(3):113–124, 1956.

[14] Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, and Zhifang Sui. A survey for in-context learning. *arXiv preprint arXiv:2301.00234*, 2022.

[15] Benjamin L. Edelman, Surbhi Goel, Sham Kakade, and Cyril Zhang. Inductive biases and variable creation in self-attention mechanisms, 2022.

[16] Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, et al. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 1, 2021.

[17] Robert Gallager. *Discrete Stochastic Processes (Draft of 2nd Edition)*. MIT OpenCouseWare, 2011.

[18] Shivam Garg, Dimitris Tsipras, Percy Liang, and Gregory Valiant. What can transformers learn in-context? A case study of simple function classes. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022.

[19] Jesse Hoogland, George Wang, Matthew Farrugia-Roberts, Liam Carroll, Susan Wei, and Daniel Murfet. The developmental landscape of in-context learning. *CoRR*, abs/2402.02364, 2024. doi: 10.48550/ARXIV.2402.02364. URL `https://doi.org/10.48550/arXiv.2402.02364`.

[20] Arthur Jacot, Clément Hongler, and Franck Gabriel. Neural tangent kernel: Convergence and generalization in neural networks. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 8580–8589, 2018. URL `https://proceedings.neurips.cc/paper/2018/hash/5a4be1fa34e62bb8a6ec6b91d2462f5a-Abstract.html`.

[21] Andrej Karpathy. Mingpt. `https://github.com/karpathy/minGPT/tree/master`, 2023.

[22] Tanishq Kumar, Blake Bordelon, Samuel J. Gershman, and Cengiz Pehlevan. Grokking as the transition from lazy to rich training dynamics. *CoRR*, abs/2310.06110, 2023. doi: 10.48550/ARXIV.2310.06110. URL `https://doi.org/10.48550/arXiv.2310.06110`.

[23] Yingcong Li, Muhammed Emrullah Ildiz, Dimitris Papailiopoulos, and Samet Oymak. Transformers as algorithms: Generalization and stability in in-context learning. In *International Conference on Machine Learning*, pages 19565–19594. PMLR, 2023.

[24] Kaifeng Lyu, Jikai Jin, Zhiyuan Li, Simon S. Du, Jason D. Lee, and Wei Hu. Dichotomy of early and late phase implicit biases can provably induce grokking. *CoRR*, abs/2311.18817, 2023.

[25] 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. *CoRR*, abs/2402.04161, 2024. doi: 10.48550/ARXIV.2402.04161.

[26] William Merrill, Nikolaos Tsilivis, and Aman Shukla. A tale of two circuits: Grokking as competition of sparse and dense subnetworks. *CoRR*, abs/2303.11873, 2023.

[27] Eshaan Nichani, Alex Damian, and Jason D Lee. How transformers learn causal structure with gradient descent. *arXiv preprint arXiv:2402.14735*, 2024.

[28] Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Scott Johnston, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. In-context learning and induction heads. *Transformer Circuits Thread*, 2022. https://transformer-circuits.pub/2022/in-context-learning-and-induction-heads/index.html.

[29] Alethea Power, Yuri Burda, Harri Edwards, Igor Babuschkin, and Vedant Misra. Grokking: Generalization beyond overfitting on small algorithmic datasets. *arXiv preprint arXiv:2201.02177*, 2022.

[30] Gautam Reddy. The mechanistic basis of data dependence and abrupt learning in an in-context classification task, 2023.

[31] Rylan Schaeffer, Brando Miranda, and Sanmi Koyejo. Are emergent abilities of large language models a mirage? *CoRR*, abs/2304.15004, 2023.

[32] Claude Elwood Shannon. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423, 1948.

[33] Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. Self-attention with relative position representations. In Marilyn A. Walker, Heng Ji, and Amanda Stent, editors, *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics:*

*Human Language Technologies, NAACL-HLT, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 2 (Short Papers)*, pages 464–468. Association for Computational Linguistics, 2018.

[34] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008, 2017.

[35] Jingfeng Wu, Difan Zou, Zixiang Chen, Vladimir Braverman, Quanquan Gu, and Peter L Bartlett. How many pretraining tasks are needed for in-context learning of linear regression? *arXiv preprint arXiv:2310.08391*, 2023.

[36] Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. An explanation of in-context learning as implicit bayesian inference. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*, 2022.

# A Related Work

**In-Context Learning.** In [11], the authors discuss how properties of the data distribution promote ICL. Xie et al. [36] suggest a Bayesian interpretation of ICL and studies how ICL emerges when the training distribution comes from a Hidden Markov Model (HMM). Abernethy et al. [1] study the ability of transformers to segment the context into pairs of examples and labels and provide learning guarantees when the labeling is of the form of a sparse function. Finally, the work of Bietti et al. [6] studies the dynamics of training transformers on a task that is reminiscent of our Markov chain setting but has additional complexities. Instead of drawing a fresh Markov chain for each sequence, in their task all sequences are sampled from the same Markov chain; after certain 'trigger' tokens, the following 'output' token is chosen deterministically within a sequence. Thus, successful prediction requires incorporating both global bigram statistics and in-context deterministic bigram copying, unlike in our setting where the patterns computed by *statistical* induction heads are necessary and sufficient. As in our work, the authors identify multiple distinct stages of training and show how multiple top-down gradient steps lead to a solution.

**Induction Heads.** Elhage et al. [16] relates ICL with the formation of induction heads, subcomponents of transformers that match previous occurrences of the current token, retrieving the token that succeeds the most recent occurrence. Reddy [30] studies the formation of induction heads and their role in ICL, showing empirically that a three layer network exhibits a sudden formation of induction heads towards solving some ICL problem of interest. Bietti et al. [6] study the effect of specific trigger tokens on the formation of induction heads.

**Phase Transitions.** It has been observed in different contexts that neural networks and language models display a sudden drop in loss during their training process. This phase transition is often related to emergence of new capabilities in the network. The work of Power et al. [29] observed the "grokking" phenomena, where the test loss of neural networks sharply drops, long after the network overfits the training data. Chen et al. [12] shows another example of a phase transition in language model training, where the formation of specific attention mechanisms happen suddenly in training, causing the loss to quickly drop. Barak et al. [5] observe that neural networks trained on complex learning problems display a phase transition when converging to the correct solution. Several works [22, 24] attribute these phase transitions to rapid changes in the inductive bias of networks, while Merrill et al. [26] argue that the models are sparser after the phase change. Schaeffer et al. [31] warn that phenomena in deep learning that seem to be discontinuous can actually be understood to evolve continuously once seen through the right lens.

**Concurrent works.** In parallel to this work, there have been a number of papers devoted to the study of similar questions regarding in-context learning or Markov chains: Akyürek et al. [3] empirically compare the ability of different architectures to perform in-context learning of regular languages. Their experiments with synthetic languages motivate architectural changes which improve natural language modeling in large scale datasets. Hoogland et al. [19] observe similar stage-wise learning behaviors on transformers trained on language or synthetic linear regression tasks. Makkuva et al. [25] study the loss landscape of transformers trained on sequences sampled from a single Markov Chain. Perhaps closest to our work, Nichani et al. [27] introduces a general family of in-context learning tasks with causal structure, a special case of which is in-context Markov chains. The authors prove that a simplified transformer architecture (similar to the one we introduce in Section B.2) can learn to identify the causal relationships by training via gradient descent, and also characterize the ability of the trained models to adapt to out-of-distribution data. The focus of our work, instead, is on the different stages of training and how they relate to specific, well-defined, strategies.

# B Setup

In this section, we provide further details on our learning problem and present the neural network architectures that we consider.

**Details on ICL-MC Task.** We focus on the case of the *flat* Dirichlet distribution, with $\boldsymbol{\alpha} = (1, \ldots, 1)^\top$, that corresponds to uniformly random transition probabilities between states. We draw the initial state $x_1$ from the stationary distribution $\boldsymbol{\pi}$ of the chain (which exists almost surely). We primarily consider the case where the number of states $k$ is 2 or 3. In subsection E, we consider the generalization of this setting to $n$-grams for $n > 2$. Instead of $\Pr(x_t)$ being determined by $x_{t-1}$,

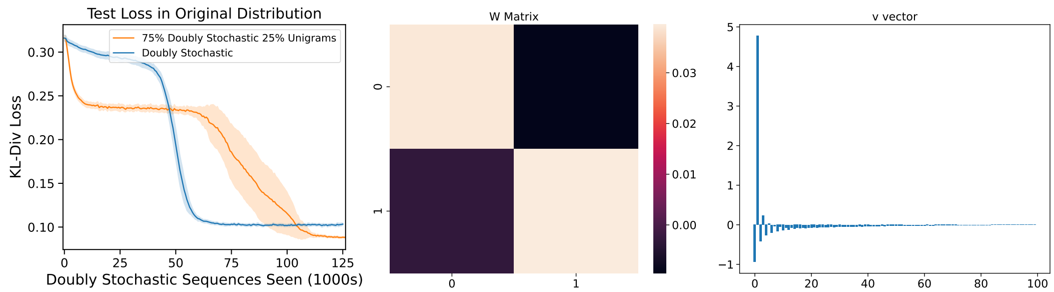


Figure 3: (*left*) Unigrams slow down optimization: Comparison of two-layer attention only transformers trained on two distributions; one with a uniformly random doubly stochastic transition matrix and another with a mixture of the doubly stochastic and unigrams distribution. We see that in absence of unigrams "signal" the model minimizes the loss (evaluated on the full distribution) much faster. (*center, right*) Training of the minimal model on ICL-MC with $k = 2$ states: (*center*) The heatmap of the second layer ($W$ matrix) that learns to be close to diagonal. (*right*) The values of the positional embeddings (1st layer) that display a curious even/odd pattern. This is before any softmax is applied to the positional embeddings.

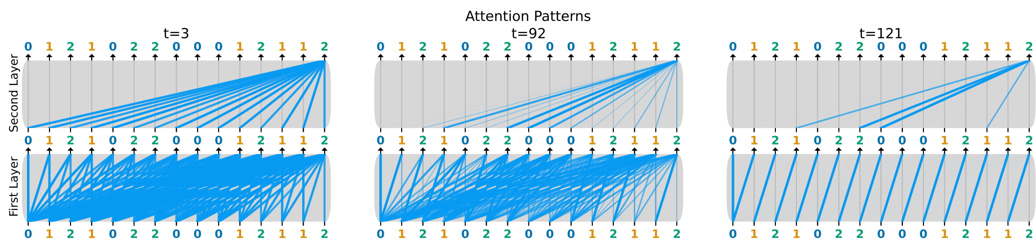


Figure 4: Attention patterns that correspond to the last token of the sequence for a transformer trained to perform ICL-MC. The intensity of each blue line signifies the strength of the corresponding attention value. As the model gets trained, we observe that the attention weights mimic the construction of Proposition B.2. Specifically, at the end of training (*right*), each token in the first layer is attending to the previous token. In the second layer, the last token, a "2", is attending to tokens that followed "2"s, allowing bigram statistics to be calculated. See Figure 7 for full attention matrices

we let $\Pr(x_t)$ be determined by $x_{t-n+1}, \ldots, x_{t-1}$, according to a conditional distribution $\mathcal{P}$ drawn from some prior. In particular, for each tuple of $n-1$ tokens, we sample the vector of conditional probabilities for the next state from a flat Dirichlet distribution.

### B.1 Potential Strategies for (Partially) Solving ICL-MC

**1st strategy: Unigrams.** Since we let the Markov chain reach its stationary distribution (which exists a.s.), the optimal strategy across unigrams is just to count frequency of states and form a posterior belief about the stationary distribution. Unfortunately, the stationary distribution of this random Markov chain does not admit a simple analytical characterization when there is a finite number of states, but it can be estimated approximately. At the limit of $k \to \infty$, the stationary distribution converges to the uniform distribution [8].

**2nd strategy: Bigrams.** For any pair of states $i$ and $j$, let $\mathcal{P}_{ij}$ be the probability of transitioning from $i$ to $j$. On each sample $\boldsymbol{x}$, we can focus on the transitions from the $i$-th state, which follow a categorical distribution with probabilities equal to $(\mathcal{P}_{i1}, \ldots, \mathcal{P}_{ik})$. If we observe the in-context empirical counts $\{c_{ij}\}_{j=1}^{k}$ of the transitions, then $\mathcal{P}_{ij}$ is given by: $(\mathcal{P}_{i1}, \ldots, \mathcal{P}_{ik})\,|\boldsymbol{x} \sim \mathrm{Dir}(k, c_{i1} + \alpha_1, \ldots, c_{ik} + \alpha_k)$, where $\alpha_1, \ldots, \alpha_k$ are the Dirichlet concentration parameters of the prior. Hence, each $\mathcal{P}_{ij}$ has a (marginal) distribution that is actually a Beta distribution: $\mathcal{P}_{ij}|\boldsymbol{x} \sim \mathrm{Beta}\left(c_{ij} + \alpha_j, \sum_j \alpha_j + N_i - \alpha_j - c_{ij}\right)$, where $N_i$ is the total number of observed transitions from state $i$. As such, our best (point) estimate for each state $j$ is given by: $\mathbb{E}\left[\mathcal{P}_{ij}|\boldsymbol{x}\right] = \frac{c_{ij}+\alpha_j}{N+\sum_i \alpha_i}$. For the uniform Dirichlet, $\boldsymbol{\alpha} = (1,\dots,1)^\top$, it is $\mathbb{E}\left[\mathcal{P}_{ij}|\boldsymbol{x}\right] = \frac{c_{ij}+1}{N_i+k}$.

*Remark* B.1. The bigram strategy implicitly assumes that the first token $x_1$ is sampled uniformly, as opposed to being sampled from the stationary distribution (which is used in our experiments and theoretical results). As the context length grows, the bigram statistics approach the Bayes optimal solution either way and this difference becomes negligible.

### B.2 Architectures: Transformers and Simplifications

We are mainly interested in investigating how transformers [34] can succeed in in-context learning this task. We focus on attention-only transformers with 2 layers with causal masking which is a popular architecture for language modeling. Given an input sequence $\boldsymbol{x}$, the output of an $n$-layer attention-only transformer[1] is:

$$TF(E) = P \circ (Attn_n + I) \cdots \circ (Attn_1 + I) \circ E. \tag{1}$$

Where $E \in \mathbb{R}^{t\times d}$ is an embedding of $\boldsymbol{x}$, $P \in \mathbb{R}^{d\times k}$ is a linear projection to the output logits, and $Attn(\boldsymbol{x})$ is masked self attention with relative position embeddings [33], which is parameterized by $W_Q, W_K, W_V \in \mathbb{R}^{d\times d}, v \in \mathbb{R}^{t\times d}$:

$$Attn(z) = \text{softmax}(\text{mask}(A))zW_V, \qquad A_{i,j} = \frac{(z_i W_Q)(z_j W_K + v_{i-j+1})^\top}{\sqrt{d}}. \tag{2}$$

Transformers with more complicated components, such as MLPs, also display similar qualitative behavior (see Figure 8). During training, we minimize this loss:

$$L(\theta) = \mathop{\mathbb{E}}_{\substack{\boldsymbol{x}\sim\mathcal{P} \\ \mathcal{P}\sim\text{Dir}(\boldsymbol{\alpha})^k}} \left[ \frac{1}{t} \sum_{p=1}^{t} l\left(TF(\boldsymbol{x};\theta)_p, x_{p+1}\right) \right], \tag{3}$$

where $\theta$ denotes the parameters of the model and $l$ is the cross entropy loss.

We now show how a two-layer transformer can represent the optimal bigrams solution.

**Proposition B.2** (Transformer Construction)**.** *A single-head two layer attention-only transformer can find the bigram statistics in the in-context learning Markov chain task.*

Intuitively, the first layer of the transformer copies the previous token at each position, and in the second layer each token sums the embeddings of all the tokens whose output from the first layer matches itself. The full proof can be found in Appendix D.1.

**Simplified Transformer Architecture.** As we see from the construction, there are two main ingredients in the solution realized by the transformer; (1st layer) the ability to look one token back and (2nd layer) the ability to attend to itself. For this reason, we define a *minimal model* that is expressive enough to be able to represent such a solution, but also simple enough to be amenable to analysis. Let $e_{x_i}$ denote the one-hot embedding that corresponds to the state at position $i \in [T]$, and let $E$ be the $\mathbb{R}^{(T+1)\times k}$ one-hot embedding matrix. Then the model is parameterized by $W \in \mathbb{R}^{k\times k}$ and $v \in \mathbb{R}^{T+1}$ and defined as:

$$f(E) = \text{mask}(EW(\text{Softmax}(M)E)^\top)E, \quad M = \begin{pmatrix} v_0 & -\infty & \dots & -\infty \\ v_1 & v_0 & \dots & -\infty \\ \vdots & \vdots & \dots & \vdots \\ v_T & v_{T-1} & \dots & v_0 \end{pmatrix} \in \mathbb{R}^{(T+1)\times(T+1)}, \tag{4}$$

where mask $(\cdot)$ is a causal mask, and $\text{Softmax}(M)_{i,j} = \frac{\exp(M_{i,j})}{\sum_{t=0}^T \exp(M_{t,j})}$. Notice that the role of $W$ is to mimic the attention mechanism of the second layer and the role of $v$ is that of the relative positional embeddings. This model can be seen as a simplified version of a two-layer linear attention-only transformer. See also Appendix D.2 for a discussion.

[1]For simplicity of notation we assume embedding dimension equals the hidden dimension, but in general they can be different.

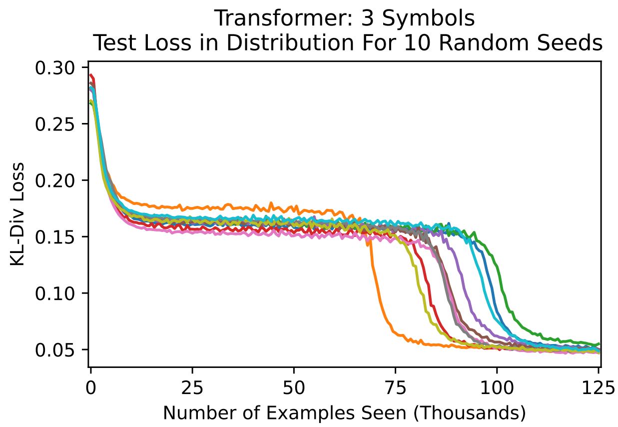


Figure 5: In distribution test loss for $10$ two layer attention only transformers, with random seeds $0, 1, \ldots 9$ (randomness affects initialization and the training data). The training dynamics are consistent for each model, though the exact position of the phase transitions changes.

*Fact* B.3. Both the bigrams strategy and the unigrams strategy can be expressed by the minimal model with a simple choice of weights.

- **Bigrams:** $v = (0, c, 0, \ldots, 0)^\top$ and $W = I_{k\times k}$, then $f(E)_{T,s} = \sum_{t'=2}^{T} \mathbb{1}\{x_{t'} = s\} \mathbb{1}\{x_{t'-1} = x_T\} + O\left(\frac{kT^2}{\exp(c)}\right)$.
- **Unigrams:** For $v = (0, 0, 0 \ldots, 0)^\top$, $W = 11^\top$, we have $f(E)_{T,s} = \sum_{t'=1}^{T} \mathbb{1}\{x_{t'} = s\}$.

# C Experimental Details and Additional Experiments

**Note on KL-divergence** In our experiments, we used KL divergence to measure the difference between the probabilities predicted by the model and other probability distributions. For test loss, this other distribution was the appropriate rows of the transition matrices used to generate the test examples.

Formally, let $f(\boldsymbol{x}_{1:T-1})$ be the softmax distribution of the transformer's output, given the input sequence $\boldsymbol{x}_{1:T-1}$. In our standard setting, we measured

$$d_{KL}(\mathcal{P}_{\boldsymbol{x}_{T-1}} || f(\boldsymbol{x}_{1:T-1}))$$

where $\mathcal{P}_{\boldsymbol{x}_{T-1}}$ is the true distribution of the next state $\boldsymbol{x}_T$ given the previous state, under the true Markov chain $\mathcal{P}$. Note that $\mathcal{P}$ varies from sequence to sequence (it is drawn from a prior over transition matrices) and is not directly observable by the learner—this is what needs to be learned in-context.

For measuring how close the model was to various strategies, we computed the predicted probabilities given by said strategies, and used those as the base distribution. Note that the output of the bigrams strategy (which is Bayes-optimal for our base setting) is different from the aforementioned ground-truth $\mathcal{P}_{\boldsymbol{x}_{T-1}}$). Instead, as described in Section B, it is a Bayesian posterior distribution of the next state given the observed sequence, with the prior determined by the prior distribution of transition matrices. Formally:

$$\mathbb{E}[\mathcal{P}_{\boldsymbol{x}_{T-1}} | \boldsymbol{x}_{1:T-1}]$$

where the expectation is taken over the draw of Markov chain transition matrix.

**Experimental details** We train transformers of the form (1) with the AdamW optimizer with learning rate $3e-5$ (for 3-grams a learning rate of $3e-2$ was used), batch size $64$, and hidden dimension $16$. The sequence length of the examples is $100$ tokens. The minimal model was trained with SGD, with batch size $64$, and learning rate $3e-4$. We use PyTorch 2.1.2.

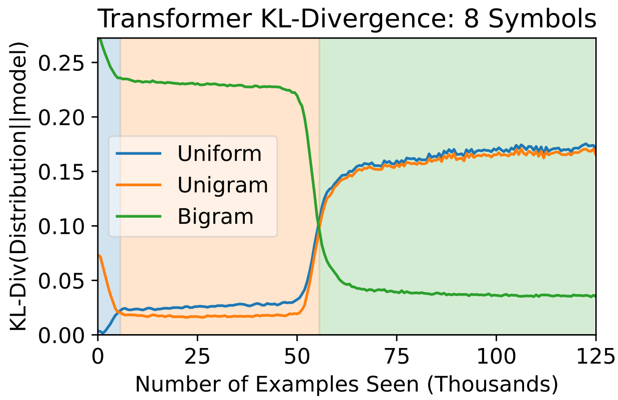


Figure 6: ICL-MC with $k = 8$ states - KL-divergence between the transformer and the various strategies over training. This required a sequence length greater than 100 (200 in this case) for the difference between unigrams and bigrams to be large enough for the unigram phase to be visible (in either case there was a plateau before the final drop in test loss).

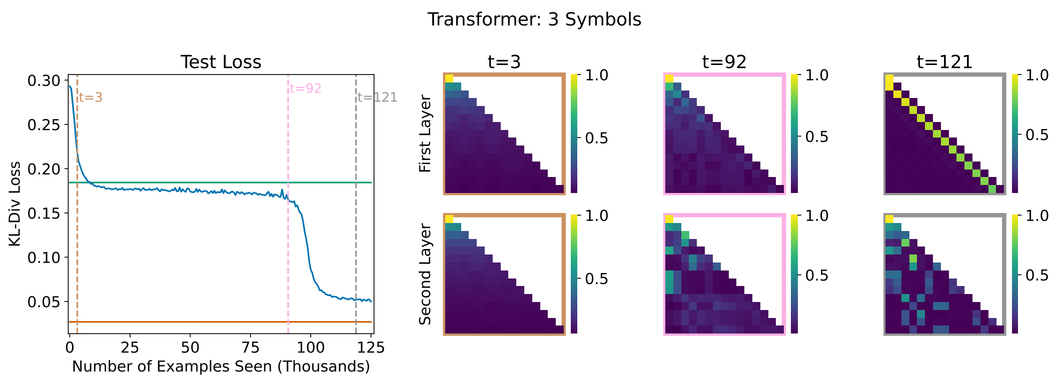


Figure 7: A two layer attention-only transformer trained with cross entropy loss on ICL-MC. The heatmaps on the right represent part of the attention for the transformer at various time steps, specifically the values of the matrix $A$ from (2). The top row are showing $A$ from the first layer, and the bottom row from the second layer.

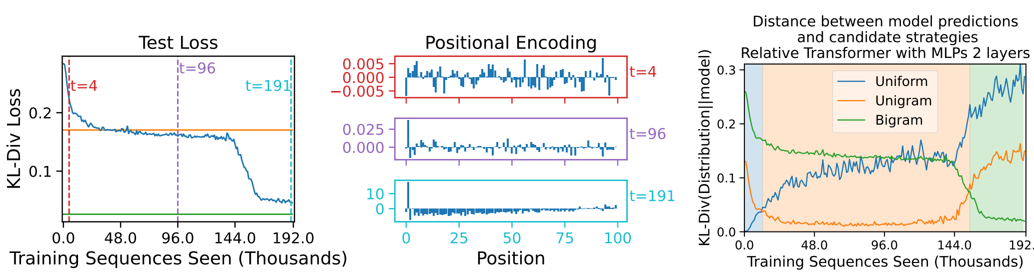


Figure 8: A two layer relative position encoding transformer with MLPs trained on ICL-MC with k=3 symbols. Notice while slightly noisier, the overall trend and observations made regarding the attention only transformer still hold.

The data was generated in an online fashion, using numpy.random.dirichlet to generate each row of the transition matrices. Both the model initialization (for the transformers) and the data were randomized based on the seed (in a perfectly reproducible manner).

Some of the training and model code was based on minGPT [21]. The experiments all measure the outputs of the models at the last token.

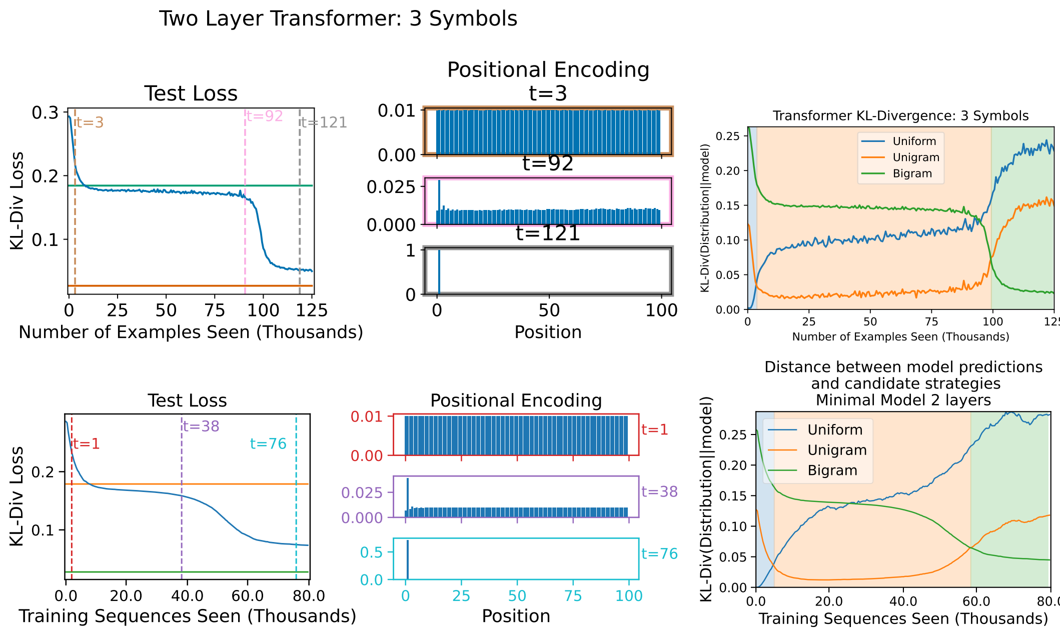


Figure 9: A comparison of the two layer attention only transformer and minimal model for $k = 3$ symbols.

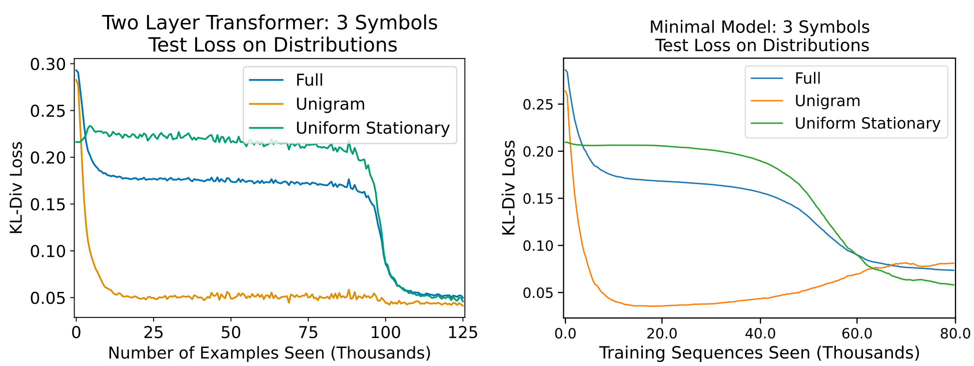


Figure 10: A two layer attention-only transformer (top) and minimal model (4) (bottom), trained on the main task with ICL-MC with cross entropy loss, test loss measured by KL-Divergence from the underlying truth (labels based on transition probabilities, not samples). The distributions test loss is measured in are (from left to right) in-distribution, a distribution where each token is sampled iid, and a distribution over uniformly random doubly stochastic transition matrices (equivalently, stationary distribution is identity, or unigram based guesses are as good as guessing uniform probability). For both models, the in distribution test loss quickly drops to the level of the unigram algorithm.

All of the experiments were performed with a single NVIDIA GeForce GTX 1650 Ti GPU with 4 gigabytes of vram with 32 gigabytes of system memory. Each training run took under ten minutes.

# D Proofs

In this section, we present our theoretical results on in-context learning Markov Chains of Section B.2.

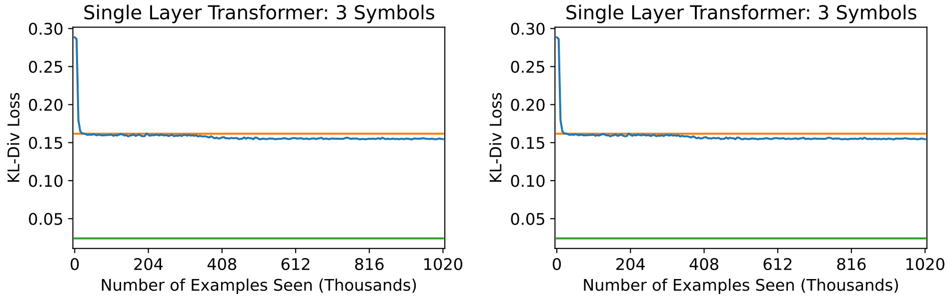


Figure 11: Graphs of test loss showing that a single layer transformer can not achieve good performance on ICL-MC. This result holds for transformers with or without MLPs, and with absolute or relative positional encodings. These graphs show that even trained 8 times longer, there is no notable increase in performance beyond the unigrams strategy (orange line).

### D.1 Transformer Construction

*Proof of Proposition B.2.* Set the internal dimension $d = 3k$, and choose $\mathbf{e}_{\boldsymbol{x}}$ to be one-hot embeddings—that is, $\mathbf{e}_{\boldsymbol{x}_i} = \delta_{\boldsymbol{x}_i}$, where $\delta$ is the Kronecker delta. We will call the parameters of attention layer $i$, $W_Q^{(i)}, W_K^{(i)}, W_V^{(i)}, v^{(i)}$. Let

$$v^{(1)} = \begin{pmatrix} \delta_2 \mathbf{1}_k^\top \\ \mathbf{0} \\ \mathbf{0} \end{pmatrix} \quad W_Q^{(1)} = \begin{pmatrix} cI^{k\times k} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{pmatrix} \quad W_K^{(1)} = \mathbf{0} \quad W_V^{(1)} = \begin{pmatrix} \mathbf{0} & I^{k\times k} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{pmatrix}$$

So,

$$A_{i,j}^{(1)} = \frac{(e_i W_Q^{(1)})(v_{i-j+1}^{(1)})^\top}{\sqrt{d}}.$$

Notice that $A_{i,j}^{(1)} = c\mathbb{1}[j = i-1]$. So, $\text{softmax}(\text{mask}(A))_{i,j}^{(1)} \approx \mathbb{1}[j = i-1]$ for large enough $c$. So, for any $2 \le i < T, 1 \le j < k$, $Attn_1(e)_{i,j+k} = e_{i-1,j}$. Effectively, the first layer appends the embedding of the previous token after the embedding of the current token, so that the output at position $i$ is approximately $(e_{x_i} \quad e_{x_{i-1}} \quad \mathbf{0})$.

The second layer is defined as follows:

$$v^{(2)} = \mathbf{0} \quad W_Q^{(2)} = \begin{pmatrix} cI^{k\times k} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{pmatrix} \quad W_K^{(2)} = \begin{pmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ I^{k\times k} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{pmatrix} \quad W_V^{(2)} = \begin{pmatrix} \mathbf{0} & \mathbf{0} & I^{k\times k} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{pmatrix}$$

Note that $z = e + Attn_1(e)$, then

$$A_{i,j}^{(2)} = \frac{(z_i W_Q^{(2)})(z_j W_K^{(2)})^\top}{\sqrt{d}} = \frac{c e_{x_i} (e_{x_{j-1}})^\top}{\sqrt{d}} = \frac{c}{\sqrt{d}} \mathbb{1}[x_{j-1} = x_i].$$

So, for all $j < i$, $\text{softmax}(\text{mask}(A))_{i,j} \approx \frac{\mathbb{1}[x_{j-1}=x_i]}{\sum_{h=1}^i \mathbb{1}[x_{h-1}=x_i]}$ for large enough $c$. For any $2 \le i < T, 1 \le j < k$,

$$Attn_2(e)_{i,j+2k} = \sum_{h=1}^{3k} \frac{\mathbb{1}[x_{h-1} = x_i]}{\sum_{g=1}^i \mathbb{1}[x_{g-1} = x_i]} (z W_V^{(2)})_{h,j} = \frac{\sum_{h=1}^k \mathbb{1}[x_{h-1} = x_i] \mathbb{1}[x_h = j]}{\sum_{g=1}^i \mathbb{1}[x_{g-1} = x_i]}.$$

Which is exactly the empircal bigram statistics (that is, the number of times $x_i \to j$ appears before position $i$), so to make this the output, $P = \begin{pmatrix} \mathbf{0} \\ \mathbf{0} \\ I^{k\times k} \end{pmatrix}$ [2] $\square$

### D.2 ICL-MC with Minimal Model

To abstract away some of the many complicated components from the transformer architecture, we focus our attention now to the minimal model of Section B.2. We train minimal models of eq. (4), starting from a deterministic constant initialization, by minimizing the cross entropy loss with SGD. Full experimental details can be found in the Appendix. Figure 2 (bottom) displays the training curves for the minimal model.

**Lemma D.1.** *Let the model defined as in eq.* (4) *and initialized with* $W = \mathbf{0}, v = \mathbf{0}$. *If the random transition matrices are either*

- *uniformly random from* $2 \times 2$ *stochastic matrices*
- *With some constant probability* $0 < \alpha < 1$, *a uniformly random doubly stochastic matrices, and otherwise* $\mathbb{1}^\top v$ *where* $v$ *is a uniformly random vector on the* $k$*-simplex.*

*after one step of full batch gradient descent with step size* $\eta$ *we have:*

$$W = \eta(T+1)\left(AI + B\,\mathbf{1}^\top\mathbf{1}\right) + \eta\, O(\log T) \text{ and } v^{(1)} = \mathbf{0},$$

*where* $A, B \in \mathbb{R}^+$.

*Assuming in the first step* $\eta = O\left(\frac{1}{T^2}\right)$, *after the second step of gradient descent, it holds:*

$$W = (\eta + \eta_W)(T+1)\left(AI + B\,\mathbf{1}^\top\mathbf{1}\right) + (\eta + \eta_W)\, O(\log T)$$

*where the step size on* $W$ *in the second step is* $\eta_W$. *Furthermore,*

$$v_1 = \eta_v C \log T, \text{ and } v_1 - v_n = \eta_v \Omega(\log T)\ \forall n \neq 1,$$

*where* $\eta_v$ *is the step size for* $v$ *in the second step, and* $C \approx 0.0114$.

*If* $\eta_v = O(T)$, *and* $\eta_W = \frac{1}{T(A+B)}$ *then the output of the model will be a weighted sum of bigrams and unigrams. Formally,*

$$f(E)_{T,s} = \frac{A}{A+B}\sum_{i=1}^{T} \mathbb{1}\left[x_{i-1} = x_t, x_i = s\right] + \frac{B}{A+B}\sum_{i=1}^{T} \mathbb{1}\left[x_i = s\right] + O(\log T)$$

*Note that in the first distribution (uniformly random* $2 \times 2$*) or the second distribution with* $k > 6$, $A > B$, *so at the end of the two steps, the weight on bigrams is greater than that of the weight on unigrams.*

*Proof Overview.* The idea of the proof is that a first step of gradient descent with a small learning rate can align the second layer, while a second step can learn to identify the correct relative positional embedding. The identity bias of $W$ in the second layer ensures there is a strong signal in the gradient to look back one in the first layer. Without a bias in $W$, the gradient for the positional encodings, $v$, turns out to be zero.

We get additional intuition from looking at the proof for just the second distribution: in the first step, effectively all of the gradient comes from the examples where the unigram strategy is optimal, while in the second step effectively all of the gradient comes from the examples where the bigram strategy is optimal.

*Remark* D.2. It is worth noting that, while this is a simplified setting, the analysis goes beyond NTK-based [20] analyses where the representations do not change much and it crucially involves more than one step which has been a standard tool in the analysis of feature learning [4].

[2]Technically, the output of this construction is not the log probabilities as generally cross-entropy loss assumes. These can be approximated linearly by setting $P = \begin{pmatrix} b\mathbf{1}^\top\mathbf{1} \\ \mathbf{0} \\ aI^{k\times k} \end{pmatrix}$ to change the output from $x$ to $ax + b$. In practice, this approximation can achieve close to Bayes optimal loss.

**Setup and notation** Our data consists of sequences of length $T+1$, $\boldsymbol{x} = (x_0, \ldots, x_T)$, drawn from a Markov Chain with state space $S = \{1, \ldots, k\}$ (i.e., $x_j \in \{1, \ldots, k\}$ for all $j \in [T]$), and a random transition matrix $P$. Each row of the matrix is sampled from a flat Dirichlet distribution, i.e. $P_i \sim \mathrm{Dir}(\mathbf{1})$, corresponding drawing the row from a uniform distribution over the simplex. Let $E \in \{0,1\}^{(T+1)\times k}$ be the one hot embedding matrix of $x$, that is, $E_{i,x_i} = 1$ and for all $s \neq x_i$ $E_{i,s} = 0$.

**Model** We define our model as a simplified sequence to sequence transformer $f : \mathbb{R}^{T\times k} \to \mathbb{R}^{(T+1)\times k}$ with $f(E) = \mathrm{mask}(EW(\mathrm{Softmax}(M)E)^\top)E$. The trained parameters are $W \in \mathbb{R}^{k\times k}$ and $v \in \mathbb{R}^{T+1}$. We define $M \in \mathbb{R}^{(T+1)\times(T+1)}$ as $M = \begin{pmatrix} v_0 & -\infty & \ldots & -\infty \\ v_1 & v_0 & \ldots & -\infty \\ \vdots & \vdots & \ldots & \vdots \\ v_T & v_{T-1} & \ldots & v_0 \end{pmatrix}$, that is, for all $T \geq i \geq j \geq 0$, $M_{i,j} = v_{i-j}$ and if $i > j$, $M_{j,i} = -\infty$. Furthermore, $v = [v_0, v_2, \ldots, v_T] \in \mathbb{R}^{tT+1}$. Softmax is defined as follows:

$$\mathrm{Softmax}(M)_{i,j} = \frac{\exp(M_{i,j})}{\sum_{T=1}^{T} \exp(M_{i,j})}$$

The logit for symbol $s$ at position $T$ for our model is:

$$f(E)_{T,s} = \sum_{u=1}^{k} \sum_{i=0}^{T} \sum_{j=0}^{i} W_{x_T,u} \mathbb{1}[x_{i-j} = u \wedge x_i = s] \frac{\exp(v_j)}{\sum_{\ell=0}^{i} \exp(v_\ell)}. \tag{5}$$

The model can represent the unigrams and bigrams solutions as following:

- Construction for bigrams: $v = (0, c, 0, \ldots, 0)^\top$ and $W = I_{k\times k}$, then $f(E)_{T,s} = \sum_{i=0}^{T} \mathbb{1}[x_i = s \wedge x_{i-1} = x_T] + O\left(\frac{T^3}{\exp(c)}\right)$. As $c$ tends to infinity, this becomes bigrams.
- Construction for unigrams: $v = \mathbf{0}$ and $W = \mathbf{1}^\top \mathbf{1}$, then $f(E)_{T,s} = \sum_{i=0}^{T} \mathbb{1}[x_i = s]$.

**Proof of Fact B.3**

*Proof.* We will first prove the unigrams construction.

$$\begin{aligned}
f(E)_{T,s} &= \sum_{u=1}^{k} \sum_{i=0}^{T} \sum_{j=0}^{i} W_{x_T,u} \mathbb{1}[x_{i-j} = u \wedge x_i = s] \frac{\exp(v_j)}{\sum_{\ell=0}^{i} \exp(v_\ell)} \\
&= \sum_{u=1}^{k} \sum_{i=0}^{T} \sum_{j=0}^{i} \mathbb{1}[x_{i-j} = u \wedge x_i = s] \frac{1}{i} \\
&= \sum_{i=0}^{T} \sum_{j=0}^{i} \mathbb{1}[x_i = s] \frac{k}{i} \\
&= k \sum_{i=0}^{T} \mathbb{1}[x_i = s]
\end{aligned}$$

Which is exactly unigrams.

Now consider the bigrams construction. As $c$ grows, the softmax of $v$ very quickly becomes one hot. Formally, by lemma B.7 in [15], for any $i > 0$,

$$\frac{\exp(v_j)}{\sum_{\ell=0}^{i} \exp(v_\ell)} = \mathbb{1}[j = 1] + O\left(\frac{T}{\exp(c)}\right)$$

So,

$$f(E)_{T,s} = \sum_{u=1}^{k}\sum_{i=0}^{T}\sum_{j=0}^{i} W_{x_T,u}\mathbb{1}[x_{i-j} = u \wedge x_i = s]\frac{\exp(v_j)}{\sum_{\ell=0}^{i}\exp(v_\ell)}$$
$$= \sum_{i=0}^{T}\sum_{j=0}^{i}\mathbb{1}[x_{i-j} = x_T \wedge x_i = s]\frac{\exp(v_j)}{\sum_{\ell=0}^{i}\exp(v_\ell)}$$

We will take out the term $i = 0$,

$$= \mathbb{1}[x_T = x_0 = s] + \sum_{i=1}^{T}\sum_{j=0}^{i}\mathbb{1}[x_T = x_{i-j} \wedge x_i = s]\frac{\exp(v_j)}{i-1+\exp(c))}$$

Then apply the softmax approximation mentioned earlier,

$$= \mathbb{1}[x_T = x_0 = s] + \sum_{i=1}^{T}\sum_{j=0}^{i}\mathbb{1}[x_T = x_{i-j} \wedge x_i = s]\left(\mathbb{1}\left[j=1\right] + O\left(\frac{2T}{\exp(c)}\right)\right)$$
$$= \mathbb{1}[x_T = x_0 = s] + \sum_{i=1}^{T}\mathbb{1}[x_T = x_{i-1} \wedge x_i = s] + \sum_{i=1}^{T}\sum_{j=0}^{i}\mathbb{1}[x_T = x_{i-j} \wedge x_i = s]O\left(\frac{2T}{\exp(c)}\right)$$
$$= \sum_{i=1}^{T}\mathbb{1}[x_T = x_{i-1} \wedge x_i = s] + \sum_{i=1}^{T}O\left(\frac{T^3}{\exp(c)}\right)$$

□

This simplified model was constructed by taking a two layer transformer with relative positional encodings and simplifying it. Our construction for how transformers would form induction heads (corroborated with experiments such as the viewing of attention patterns in figure 4) implies that the MLPs and the value matrices could just be identity functions, and the first layer query matrix, and the second layer positional embeddings were zero matricies, so in the simplified model we froze these parameters to there final states. We also remove the softmax on the attention in the first layer. Despite these changes, the training dynamics, our main interest, stay remarkably similar.

**Training** We analyze gradient descent with the cross entropy loss $L_T(f, E, x_{T+1}) = -\sum_{s=1}^{k}\log\text{Softmax}\left(f(E)\right)_{T,s} P_{X_T,s}$[3]

### D.3 Gradient Calculations

For use in the proofs, here we show the calculations of the gradients of the model with respect to the parameters, and the loss with respect to the model.

$$\frac{\partial f(E)_{T,s}}{\partial W_{a,b}} = \sum_{u=1}^{k}\sum_{i=0}^{T}\sum_{j=0}^{i}\mathbb{1}[x_T = a \wedge b = u]\mathbb{1}[x_{i-j} = u \wedge x_i = s]\frac{\exp(v_j)}{\sum_{\ell=0}^{i}\exp(v_\ell)}$$
$$= \sum_{i=0}^{T}\sum_{j=0}^{i}\mathbb{1}[x_T = a]\mathbb{1}[x_{i-j} = b \wedge x_i = s]\frac{\exp(v_j)}{\sum_{\ell=0}^{i}\exp(v_\ell)}$$

$$\frac{\partial f(E)_{T,s}}{\partial v_a}$$
$$= \sum_{u=1}^{k}\sum_{i=0}^{T}\sum_{j=0}^{i}W_{x_T,u}\mathbb{1}[x_{i-j} = u \wedge x_i = s]\left(\mathbb{1}[j=a]\frac{\exp(v_a)}{\sum_{\ell=0}^{i}\exp(v_\ell)} - \mathbb{1}[a \le i]\frac{\exp(v_j)}{\sum_{\ell=0}^{i}\exp(v_\ell)}\frac{\exp(v_a)}{\sum_{\ell=0}^{i}\exp(v_\ell)}\right)$$

[3]In practice, one would often use the empircal value of $x_{T+1}$ rather than its distribution $P_{X_T,s}$, but in full batch gradient descent this is in fact equivalent in our setting. This is because conditional on $x_T$ and $P$, $x_{T+1}$ is independent of $x_1, \ldots x_{T-1}$.

$$= \sum_{u=1}^{k}\sum_{i=0}^{T}\frac{\exp(v_a)}{\sum_{\ell=0}^{i}\exp(v_\ell)}\sum_{j=0}^{i}W_{x_T,u}\left(\mathbb{1}[x_{i-a}=u\wedge x_i=s]\mathbb{1}[j=a]-\mathbb{1}[x_{i-j}=u\wedge x_i=s]\mathbb{1}[a\le i]\frac{\exp(v_j)}{\sum_{\ell=0}^{i}\exp(v_\ell)}\right)$$
$$= \sum_{u=1}^{k}\sum_{i=0}^{T}\frac{\exp(v_a)}{\sum_{\ell=0}^{i}\exp(v_\ell)}W_{x_T,u}\left(\mathbb{1}[x_{i-a}=u\wedge x_i=s]\mathbb{1}[a\le i]-\sum_{j=0}^{i}\mathbb{1}[x_{i-j}=u\wedge x_i=s]\mathbb{1}[a\le i]\frac{\exp(v_j)}{\sum_{\ell=0}^{i}\exp(v_\ell)}\right)$$
$$= \sum_{u=1}^{k}\sum_{i=a}^{T}\frac{\exp(v_a)}{\sum_{\ell=0}^{i}\exp(v_\ell)}W_{x_T,u}\left(\mathbb{1}[x_{i-a}=u\wedge x_i=s]-\sum_{j=0}^{i}\mathbb{1}[x_{i-j}=u\wedge x_i=s]\frac{\exp(v_j)}{\sum_{\ell=0}^{i}\exp(v_\ell)}\right)$$
$$= \sum_{u=1}^{k}\sum_{i=a}^{T}\frac{\exp(v_a)}{\sum_{\ell=0}^{i}\exp(v_\ell)}W_{x_T,u}\mathbb{1}[x_i=s]\left(\mathbb{1}[x_{i-a}=u]-\sum_{j=0}^{i}\mathbb{1}[x_{i-j}=u]\frac{\exp(v_j)}{\sum_{\ell=0}^{i}\exp(v_\ell)}\right)$$

$$\frac{\partial L_T}{\partial f(E)_{T,s}} = \text{Softmax}(f(E))_{T,s} - P_{x_T,s}$$

## D.4 Proof of lemma D.1

*Proof.* Recall that at initialization, $v = \mathbf{0}$ and $W = \mathbf{0}$, implying further that $f(E) = \mathbf{0}$.

**First step.**

First consider the gradient of the loss with respect to $W$. By chain rule,

$$\frac{\partial L_T(E)}{\partial W_{a,b}} = \sum_{s=1}^{k}\frac{\partial L_T}{\partial f(E)_{T,s}}\frac{\partial f(E)_{T,s}}{\partial W_{a,b}}$$
$$= \sum_{s=1}^{k}(\text{Softmax}(f(E))_{T,s}-P_{x_T,s})\sum_{i=0}^{T}\sum_{j=0}^{i}\mathbb{1}[x_T=a]\mathbb{1}[x_{i-j}=b\wedge x_i=s]\frac{\exp(v_j)}{\sum_{\ell=0}^{i}\exp(v_\ell)}$$
$$= \sum_{s=1}^{k}\left(\frac{1}{k}-P_{x_T,s}\right)\mathbb{1}[x_T=a]\sum_{i=0}^{T}\sum_{j=0}^{i}\mathbb{1}[x_{i-j}=b\wedge x_i=s]\frac{1}{i+1}$$
$$= \frac{1}{k}\mathbb{1}[x_T=a]\sum_{i=0}^{T}\left(\sum_{j=0}^{i}\mathbb{1}[x_{i-j}=b]\frac{1}{i+1}-\sum_{s=1}^{k}P_{a,s}\mathbb{1}[x_T=a]\sum_{j=0}^{i}\mathbb{1}[x_{i-j}=b\wedge x_i=s]\frac{1}{i+1}\right)$$
$$= \frac{1}{k}\mathbb{1}[x_T=a]\sum_{i=0}^{T}\left(\mathbb{1}[x_i=b]-\sum_{s=1}^{k}P_{a,s}\mathbb{1}[x_T=a\sum_{j=0}^{i}\mathbb{1}[x_{i-j}=b\wedge x_i=s]\frac{1}{i+1}\right)$$
$$= \frac{1}{k}\sum_{i=0}^{T}\left(\mathbb{1}[x_i=b\wedge x_T=a]-\sum_{s=1}^{k}P_{a,s}\sum_{j=0}^{i}\mathbb{1}[x_{i-j}=b\wedge x_i=s\wedge x_T=a]\frac{1}{i+1}\right)$$
$$= \frac{1}{k}\sum_{i=0}^{T}\left(\mathbb{1}[x_0=b\wedge x_{T-i}=a]-\sum_{s=1}^{k}P_{a,s}\sum_{j=0}^{i}\mathbb{1}[x_0=b\wedge x_j=s\wedge x_{T-i+j}=a]\frac{1}{i+1}\right)$$

Where the last line follows from the markov property.

Now we take the expectation over $x, x_{T+1}$ conditioned on the transition matrix $P$,

$$\mathbb{E}_{x|P}\left[\frac{\partial L_T}{\partial W_{a,b}}\right] = \pi_b\sum_{i=0}^{T}\left(\frac{1}{k}\left(P^{T-i}\right)_{b,a}-\sum_{s=1}^{k}P_{a,s}\frac{1}{i+1}\left(P^{T-i}\right)_{s,a}\sum_{j=0}^{i}\left(P^j\right)_{b,s}\right)$$
$$= \pi_b\sum_{i=0}^{T}\left(\frac{1}{k}\left(P^{i}\right)_{b,a}-\sum_{s=1}^{k}P_{a,s}\frac{1}{T-i+1}\left(P^{i}\right)_{s,a}\sum_{j=0}^{T-i}\left(P^j\right)_{b,s}\right)$$

$$= \pi_b \pi_a (T+1) \left( \frac{1}{k} - \sum_{s=1}^{k} P_{a,s} \pi_s \right) + O(\log T)$$

Where the last step follows from Lemma D.8. Then, by applying Lemma D.3 or lemmas D.5 and D.6 (depending on the distribution assumption on $P$), there exist positive constants (potentially depending on $k$, but not $T$) $A, B$ such that for all $a$

$$\mathbb{E}_{x|P} \left[ \frac{\partial L_T}{\partial W_{a,a}} \right] = -(A+B)T + O(\log T)$$

and for all $a \neq b$,

$$\mathbb{E}_{x|P} \left[ \frac{\partial L_T}{\partial W_{a,b}} \right] = -BT + O(\log T)$$

The updated $W_{a,b}$ after the gradient step is just $-\eta \mathbb{E}_{x|P} \left[ \frac{\partial L_T}{\partial W_{a,b}} \right]$ (because $W$ is initialized at $\mathbf{0}$). Choose $\eta = \Theta\left(\frac{1}{T}\right)$, so that $W$ will be $O(1)$ with respect to $T$ after the first step.

For the gradient with respect to $v$, since $W = \mathbf{0}$,

$$\begin{aligned} \frac{\partial F(E)_{T,s}}{\partial v} &= \sum_{u=1}^{k} \sum_{i=a}^{T} \frac{\exp(v_a)}{\sum_{\ell=0}^{i} \exp(v_\ell)} W_{x_T,u} \left( \mathbb{1}[x_{i-a} = u \wedge x_i = s] - \sum_{j=0}^{i} \mathbb{1}[x_{i-j} = u \wedge x_i = s] \frac{\exp(v_j)}{\sum_{\ell=0}^{i} \exp(v_\ell)} \right) \\ &= 0 \end{aligned}$$

So,

$$\frac{\partial L_T(E)}{\partial v} = \sum_{s=1}^{k} \frac{\partial L_T(E)}{\partial f(E)_{T,s}} \frac{\partial F(E)_{T,s}}{\partial v} = 0$$

Completing the first step calculations.

**Second step.**

After the first step, $W = \eta\left(AI + B\mathbf{1}^\top \mathbf{1}\right)$. Now let us bound the output of the model,

$$\begin{aligned} |f(E)_{T,s}| &= \left| \sum_{u=1}^{k} \sum_{i=0}^{T} \sum_{j=0}^{i} W_{x_T,u} \mathbb{1}[x_{i-j} = u \wedge x_i = s] \frac{\exp(v_j)}{\sum_{\ell=0}^{i} \exp(v_\ell)} \right| \\ &= \left| \sum_{u=1}^{k} \sum_{i=0}^{T} \sum_{j=0}^{i} \eta \left(AI + B\mathbf{1}^\top \mathbf{1}\right)_{x_T,u} \mathbb{1}[x_{i-j} = u \wedge x_i = s] \frac{1}{i} \right| \\ &\leq \eta \left| \sum_{u=1}^{k} \sum_{i=0}^{T} \sum_{j=0}^{i} (A+B)\, \mathbb{1}[x_{i-j} = u \wedge x_i = s] \frac{1}{i} \right| \\ &\leq \eta \left| \sum_{u=1}^{k} \sum_{i=0}^{T} \sum_{j=0}^{i} (A+B)\, \mathbb{1}[x_{i-j} = u] \frac{1}{i} \right| \\ &\leq \eta \left| \sum_{i=0}^{T} \sum_{j=0}^{i} (A+B) \frac{1}{i} \right| \\ &\leq \eta T \left| A+B \right| \end{aligned}$$

So, using the first order approximation of softmax,

$$\frac{\partial L_T(E)}{\partial f(E)_{T,s}} = \mathrm{Softmax}(f(E))_{T,s} - \mathbb{1}\left[x_{T+1} = s\right]$$

$$
\begin{aligned}
&= \frac{1}{k} + \frac{f(E)_{T,s}}{k} - \frac{\sum_{u=1}^{k} f(E)_{T,u}}{k^2} + O(f(E)_{T,s}^2) - \mathbb{1}\left[x_{T+1} = s\right] \\
&= \frac{1}{k} + O\left(\eta \frac{T}{k}(A+B)\right) + O(\eta^2 T^2 (A+B)^2) - \mathbb{1}\left[x_{T+1} = s\right] \\
&= \frac{1}{k} + O\left(\eta \frac{T}{k}(A+B)\right) + O(\eta^2 T^2 (A+B)^2) - \mathbb{1}\left[x_{T+1} = s\right] \\
&= \frac{1}{k} - \mathbb{1}\left[x_{T+1} = s\right] + O\left(\frac{1}{T}\right)
\end{aligned}
$$

Where the last step follows since $\eta = O\left(\frac{1}{T^2}\right)$.

Now we can begin to analyze the gradients with respect to the parameters. For $W$, the gradient is approximately the same as in the last step. Notice that $\frac{\partial f(E)_{T,s}}{\partial W_{a,b}}$ does not depend on $W$, and $v$ is unchanged, so $\frac{\partial f(E)_{T,s}}{\partial W_{a,b}}$ is unchanged. Furthermore,

$$
\begin{aligned}
\frac{\partial f(E)_{T,s}}{\partial W_{a,b}} &= \sum_{s=1}^{k} \sum_{i=0}^{T} \sum_{j=0}^{i} \mathbb{1}[x_T = a] \mathbb{1}[x_{i-j} = b \wedge x_i = s] \frac{\exp(v_j)}{\sum_{\ell=0}^{i} \exp(v_\ell)} \\
&= \sum_{i=0}^{T} \sum_{j=0}^{i} \mathbb{1}[x_T = a] \mathbb{1}[x_{i-j} = b] \frac{1}{i} \\
&\le \sum_{i=0}^{T} \sum_{j=0}^{i} \frac{1}{i} \\
&= T
\end{aligned}
$$

We will now show that the gradient is approximately the same as in the first gradient step:

$$
\begin{aligned}
\frac{\partial L_T(E)}{\partial W_{a,b}} &= \sum_{s=1}^{k} \frac{\partial L_T}{\partial f(E)_{T,s}} \frac{\partial f(E)_{T,s}}{\partial W_{a,b}} \\
&= \sum_{s=1}^{k} \left(\frac{1}{k} - \mathbb{1}\left[x_{T+1} = s\right] + O\left(\frac{1}{T}\right)\right) \frac{\partial f(E)_{T,s}}{\partial W_{a,b}} \\
&= \sum_{s=1}^{k} \left(\frac{1}{k} - \mathbb{1}\left[x_{T+1} = s\right]\right) \frac{\partial f(E)_{T,s}}{\partial W_{a,b}} + O\left(\frac{1}{T}\right) \frac{\partial f(E)_{T,s}}{\partial W_{a,b}} \\
&= \pi_b \pi_a (T+1) \left(\frac{1}{k} - \sum_{s=1}^{k} P_{a,s} \pi_s\right) + O(\log T)
\end{aligned}
$$

Where the last lines follows from the gradient calculations in the first step.

Now we will consider the gradient with respect to $v$. First, notice that the uniform component of $W$, $B\mathbf{1}^\top \mathbf{1}$, has no affect on the gradient of $v$:

$$
\begin{aligned}
\frac{\partial f(E)_{T,s}}{\partial v_a} &= \sum_{u=1}^{k} \sum_{i=a}^{T} W_{x_T,u} \frac{\exp(v_a)}{\sum_{\ell=0}^{i} \exp(v_\ell)} \mathbb{1}[x_i = s] \left(\mathbb{1}[x_{i-a} = u] - \sum_{j=0}^{i} \frac{\exp(v_j)}{\sum_{\ell=0}^{i} \exp(v_\ell)} \mathbb{1}[x_{i-j} = u]\right) \\
&= \sum_{u=1}^{k} \sum_{i=a}^{T} \left(mI + B\mathbf{1}^\top \mathbf{1}\right)_{x_T,u} \frac{1}{i+1} \mathbb{1}[x_i = s] \left(\mathbb{1}[x_{i-a} = u] - \sum_{j=0}^{i} \frac{1}{i+1} \mathbb{1}[x_{i-j} = u]\right) \\
&= \sum_{u=1}^{k} \sum_{i=a}^{T} \left(A\mathbb{1}[x_T = u] + B\right) \frac{1}{i+1} \mathbb{1}[x_i = s] \left(\mathbb{1}[x_{i-a} = u] - \sum_{j=0}^{i} \frac{1}{i+1} \mathbb{1}[x_{i-j} = u]\right)
\end{aligned}
$$

$$= A\sum_{u=1}^{k}\sum_{i=a}^{T}\mathbb{1}[x_T = u]\frac{1}{i+1}\mathbb{1}[x_i = s]\left(\mathbb{1}[x_{i-a} = u] - \sum_{j=0}^{i}\frac{1}{i+1}\mathbb{1}[x_{i-j} = u]\right)$$
$$+ B\sum_{u=1}^{k}\sum_{i=a}^{T}\frac{1}{i+1}\mathbb{1}[x_i = s]\left(\mathbb{1}[x_{i-a} = u] - \sum_{j=0}^{i}\frac{1}{i+1}\mathbb{1}[x_{i-j} = u]\right)$$
$$= A\sum_{u=1}^{k}\sum_{i=a}^{T}\mathbb{1}[x_T = u]\frac{1}{i+1}\mathbb{1}[x_i = s]\left(\mathbb{1}[x_{i-a} = u] - \sum_{j=0}^{i}\frac{1}{i+1}\mathbb{1}[x_{i-j} = u]\right)$$
$$+ B\sum_{i=a}^{T}\frac{1}{i+1}\mathbb{1}[x_i = s]\left(\sum_{u=1}^{k}\mathbb{1}[x_{i-a} = u] - \sum_{j=0}^{i}\frac{1}{i+1}\sum_{u=1}^{k}\mathbb{1}[x_{i-j} = u]\right)$$
$$= A\sum_{u=1}^{k}\sum_{i=a}^{T}\mathbb{1}[x_T = u]\frac{1}{i+1}\mathbb{1}[x_i = s]\left(\mathbb{1}[x_{i-a} = u] - \sum_{j=0}^{i}\frac{1}{i+1}\mathbb{1}[x_{i-j} = u]\right)$$
$$+ B\sum_{i=a}^{T}\frac{1}{i+1}\mathbb{1}[x_i = s]\left(1 - \sum_{j=0}^{i}\frac{1}{i+1}\right)$$
$$= A\sum_{u=1}^{k}\sum_{i=a}^{T}\mathbb{1}[x_T = u]\frac{1}{i+1}\mathbb{1}[x_i = s]\left(\mathbb{1}[x_{i-a} = u] - \sum_{j=0}^{i}\frac{1}{i+1}\mathbb{1}[x_{i-j} = u]\right)$$

By chain rule,

$$\begin{aligned}
\frac{\partial L_T}{\partial v_a} &= \sum_{s=1}^{k}\frac{\partial L_T}{\partial f(E)_{T,s}}\frac{\partial f(E)_{T,s}}{\partial v_a} \\
&= \sum_{s=1}^{k}\left(\frac{1}{k} - P_{x_T,s} + O\left(\frac{1}{T}\right)\right)\sum_{u=1}^{k}\sum_{i=a}^{T}\mathbb{1}[x_T = u]\frac{1}{i+1}\mathbb{1}[x_i = s]\left(\mathbb{1}[x_{i-a} = u] - \sum_{j=0}^{i}\frac{1}{i+1}\mathbb{1}[x_{i-j} = u]\right) \\
&= \sum_{s=1}^{k}\left(\frac{1}{k} - P_{x_T,s}\right)\sum_{u=1}^{k}\sum_{i=a}^{T}\mathbb{1}[x_T = u]\frac{1}{i+1}\mathbb{1}[x_i = s]\left(\mathbb{1}[x_{i-a} = u] - \sum_{j=0}^{i}\frac{1}{i+1}\mathbb{1}[x_{i-j} = u]\right) + O\left(\frac{\log T}{T}\right)
\end{aligned}$$

Where the last step follows because

$$\begin{aligned}
\left|\sum_{u=1}^{k}\sum_{i=a}^{T}\mathbb{1}[x_T = u]\frac{1}{i+1}\mathbb{1}[x_i = s]\left(\mathbb{1}[x_{i-a} = u] - \sum_{j=0}^{i}\frac{1}{i+1}\mathbb{1}[x_{i-j} = u]\right)\right| &\leq \left|\sum_{u=1}^{k}\sum_{i=a}^{T}\mathbb{1}[x_T = u]\frac{1}{i+1}\right| \\
&= \left|\sum_{i=a}^{T}\frac{1}{i+1}\right| \\
&\leq \log T
\end{aligned}$$

In expectation over the values of $x$, conditioned on the choice of $P$:

$$\begin{aligned}
\mathbb{E}_{x|P}\left[\frac{\partial L_T}{\partial v_a}\right] &= \sum_{s=1}^{k}\sum_{u=1}^{k}\left(\frac{1}{k} - P_{u,s}\right)\sum_{i=a}^{T}\frac{\pi_u}{i+1}\left(P^{T-i}\right)_{s,u}\left(\left(P^a\right)_{u,s} - \frac{1}{i+1}\sum_{j=0}^{i}\left(P^{i-j}\right)_{u,s}\right) + O\left(\frac{\log T}{T}\right) \\
&= \sum_{s=1}^{k}\sum_{u=1}^{k}\left(\frac{1}{k} - P_{u,s}\right)\sum_{i=a}^{T}\frac{\pi_u}{T-i+1}\left(P^{i}\right)_{s,u}\left(\left(P^a\right)_{u,s} - \frac{1}{T-i+1}\sum_{j=0}^{T-i}\left(P^{j}\right)_{u,s}\right) + O\left(\frac{\log T}{T}\right) \\
&= \left(\log\left(T+1\right) - \log\left(a+1\right)\right)\sum_{s=1}^{k}\sum_{u=1}^{k}\pi_u^2 P_{u,s}\left(\pi_s - \left(P^a\right)_{u,s}\right) + O(1)
\end{aligned}$$

Where the last step follows from lemma D.9. Then, by applying Lemma D.4 or lemmas D.5 and D.6 (depending on the distribution assumption on $P$),

$$\mathbb{E}_{x|P}\left[\frac{\partial L_T}{\partial v_1}\right] < \mathbb{E}_{x|P}\left[\frac{\partial L_T}{\partial v_a}\right]$$
$$\mathbb{E}_{x|P}\left[\frac{\partial L_T}{\partial v_1}\right] < 0$$

Therefore, after the step is taken,

$$v_1 = \Theta(\eta_v \log T)$$
$$v_1 - v_n = \eta_v \Omega(\log T)$$

Finally, we can consider the state of the model after the second step. Assume that the step size for $v$ in the second step is $O(T)$, and the step size for $W$ is $\frac{1}{T(A+B)}$

$$\begin{aligned}
f(E)_{T,s} &= \sum_{u=1}^{k}\sum_{i=0}^{T}\sum_{j=0}^{i} W_{x_T,u}\mathbb{1}[x_{i-j} = u \wedge x_i = s]\frac{\exp(v_j)}{\sum_{\ell=0}^{i}\exp(v_\ell)} \\
&= \frac{1}{A+B}\sum_{u=1}^{k}\sum_{i=0}^{T}\sum_{j=0}^{i}\left(AI + B\mathbf{1}^\top\mathbf{1} + O\left(\frac{\log T}{T}\right)\right)_{x_T,u}\mathbb{1}[x_{i-j} = u \wedge x_i = s]\left(\mathbb{1}\left[j=1\right] + O\left(\frac{2T}{\exp(\log(T))}\right)\right) \\
&= \frac{1}{A+B}\sum_{u=1}^{k}\sum_{i=0}^{T}\sum_{j=0}^{i}\left(AI + B\mathbf{1}^\top\mathbf{1} + O\left(\frac{\log T}{T}\right)\right)_{x_T,u}\mathbb{1}[x_{i-j} = u \wedge x_i = s]\left(\mathbb{1}\left[j=1\right] + O\left(\frac{1}{T}\right)\right) \\
&= \frac{1}{A+B}\sum_{u=1}^{k}\sum_{i=0}^{T}\left(AI + B\mathbf{1}^\top\mathbf{1}\right)_{x_T,u}\mathbb{1}[x_{i-1} = u \wedge x_i = s] + O(\log T) \\
&= \frac{A}{A+B}\sum_{i=0}^{T}\mathbb{1}[x_{i-1} = x_T \wedge x_i = s] + \frac{B}{A+B}\sum_{u=1}^{k}\sum_{i=0}^{T}\mathbb{1}[x_{i-1} = u \wedge x_i = s] + O(\log T) \\
&= \frac{A}{A+B}\sum_{i=0}^{T}\mathbb{1}[x_{i-1} = x_T \wedge x_i = s] + \frac{B}{A+B}\sum_{i=0}^{T}\mathbb{1}[x_i = s] + O(\log T)
\end{aligned}$$

This completes the proof. □

### D.5 Inequality lemmas for $k = 2$

**Lemma D.3.** *If $P$ is a uniformly random stochastic $2 \times 2$ matrix, and $\pi$ is the stationary distribution of $P$, then*

$$\mathbb{E}\left[\pi_a^2\left(\frac{1}{k} - \sum_{s=1}^{k} P_{a,s}\pi_s\right)\right] = \frac{5}{12} - \frac{2}{3}\log(2) \approx -0.045$$

*and for any $b \neq a$*

$$\mathbb{E}\left[\pi_a\pi_b\left(\frac{1}{k} - \sum_{s=1}^{k} P_{a,s}\pi_s\right)\right] = -\frac{7}{6} + \frac{5}{3}\log(2) \approx -0.011$$

*Proof.* We have:

$$\begin{aligned}
\mathbb{E}\left[\pi_a^2\left(\frac{1}{k} - \sum_{s=1}^{k} P_{a,s}\pi_s\right)\right] &= \mathbb{E}_{a,b}\left[\frac{(b-1)^2}{(a+b-2)^2}\left[\frac{1}{2} - \frac{a(b-1)}{a+b-2} - \frac{(1-a)(a-1)}{a+b-2}\right]\right] \\
&= \frac{1}{2}\int_0^1\int_0^1 \frac{(b-1)^2}{(a+b-2)^2}dadb - \int_0^1\int_0^1 \frac{a(b-1)^3}{(a+b-2)^3}dadb + \int_0^1\int_0^1 \frac{(b-1)^2(a-1)^2}{(a+b-2)^3}dadb \\
&= \frac{1}{2}(1-\ln 2) - \frac{1}{2}(1-\ln 2) + \frac{5}{12}(5 - 8\ln 2) = \frac{5}{12} - \frac{2}{3}\ln 2.
\end{aligned} \tag{6}$$

For the non-diagonal elements, it holds:

$$
\begin{aligned}
&\mathbb{E}\left[\pi_a\pi_b\left(\frac{1}{k}-\sum_{s=1}^{k}P_{a,s}\pi_s\right)\right]\\
&\quad=\frac{1}{2}\int_0^1\int_0^1\frac{(b-1)(a-1)}{(a+b-2)^2}dadb-\int_0^1\int_0^1\frac{a(b-1)^2(a-1)}{(a+b-2)^3}dadb+\int_0^1\int_0^1\frac{(b-1)(a-1)^3}{(a+b-2)^3}dadb\\
&\quad=\frac{1}{2}\left(\ln 2-\frac{1}{2}\right)-\frac{1}{6}\left(1-\ln 2\right)+\left(\ln 2-\frac{3}{4}\right)=\frac{5}{3}\ln 2-\frac{7}{6}.
\end{aligned}
\tag{7}
$$

□

**Lemma D.4.** *If $P$ is a uniformly random stochastic $2\times 2$ matrix, and $\pi$ is the stationary distribution of $P$, then,*

$$\mathbb{E}\left[\sum_{s=1}^{k}\sum_{u=1}^{k}\pi_u^2P_{u,s}\left(\pi_s-P_{u,s}\right)\right]=-7/2+5\log(2)\approx-0.034$$

*and for any $n\neq 1$*

$$\mathbb{E}\left[\sum_{s=1}^{k}\sum_{u=1}^{k}\pi_u^2P_{u,s}\left(\pi_s-P_{u,s}\right)\right]\leq\mathbb{E}\left[\sum_{s=1}^{k}\sum_{u=1}^{k}\pi_u^2P_{u,s}\left(\pi_s-\left(P^n\right)_{u,s}\right)\right]$$

*Proof.* We have:

$$
\begin{aligned}
&\mathbb{E}\left[\sum_{s=1}^{k}\sum_{u=1}^{k}\pi_u^2P_{u,s}\left(\pi_s-P_{u,s}\right)\right]=\\
&\left[\frac{1/6x^4+(x+y)(6xy(-4x^2+2x+1)+6y^4+y^3(20-24x)}{12(x+y)}\right.\\
&\quad\left.+\frac{y^2(12x^2-12x-3)+\log((x+y)^{6x^2(4x^2+2x-1)}(x+y)^{6y^2(4y^2+2y-1)}))}{12(x+y)}\right]_0^1\\
&=-7/2+5log(2)
\end{aligned}
\tag{8}
$$

For the inequality, we have an intuition that doesn't depend on $k$, notice that:

$$
\begin{aligned}
\sum_{s=1}^{k}\sum_{u=1}^{k}\pi_u^2P_{u,s}\left(\pi_s-\left(P^n\right)_{u,s}\right)&\geq-\sum_{s=1}^{k}\sum_{u=1}^{k}\pi_u^2P_{u,s}\left|\pi_s-\left(P^a\right)_{u,s}\right|\\
&\geq-\sum_{s=1}^{k}\sum_{u=1}^{k}\pi_u^2P_{u,s}\alpha^n\\
&=-\sum_{u=1}^{k}\pi_u^2\alpha^n\\
&\leq\alpha^n
\end{aligned}
$$

As long as $\alpha$ isn't concentrated around 1, then this shows that the magnitude of the RHS is bounded by a term that shrinks exponentially in $n$. For $k=2$, we will find a similar bound, and then show separately that for all $n$ for which the bound fails, the inequality still holds true.

$$\sum_{s=1}^{k}\sum_{u=1}^{k}\pi_u^2P_{u,s}\left(\pi_s-\left(P^n\right)_{u,s}\right)=\frac{P_{1,2}P_{2,1}(4P_{1,2}P_{2,1}-P_{1,2}-P_{2,1})}{(P_{1,2}+P_{2,1})^3}(1-P_{1,2}-P_{2,1})^n$$

We can show that for any choice of $P_{1,2}$ and $P_{2,1}$ on the unit square,

$$\left|\frac{P_{1,2}P_{2,1}(4P_{1,2}P_{2,1}-P_{1,2}-P_{2,1})}{(P_{1,2}+P_{2,1})^3}\right|\leq\frac{1}{4}$$

To see why this is true, observe that,

$$
\begin{aligned}
&(4P_{1,2}P_{2,1} - P_{1,2} - P_{2,1})^2 \\
&= 16P_{1,2}^2P_{2,1}^2 + (P_{1,2} + P_{2,1})^2 - 8(P_{1,2} + P_{2,1})P_{1,2}P_{2,1} \\
&\leq 16P_{1,2}^2P_{2,1}^2 + (P_{1,2} + P_{2,1})^2 - 4(P_{1,2} + P_{2,1})^2P_{1,2}P_{2,1} && \text{since } P_{1,2} + P_{2,1} \leq 2 \\
&= 16P_{1,2}^2P_{2,1}^2 + (P_{1,2} + P_{2,1})^2 - 4P_{1,2}P_{2,1}((P_{1,2} + P_{2,1})^2 - 4P_{1,2}P_{2,1}) \\
&= (P_{1,2} + P_{2,1})^2 - 4P_{1,2}P_{2,1}(P_{1,2} - P_{2,1})^2 \\
&\leq (P_{1,2} + P_{2,1})^2
\end{aligned}
$$

Using the above, we have

$$
\begin{aligned}
\left(\frac{P_{1,2}P_{2,1}(4P_{1,2}P_{2,1} - P_{1,2} - P_{2,1})}{(P_{1,2} + P_{2,1})^3}\right)^2 &\leq \frac{P_{1,2}^2P_{2,1}^2(P_{1,2} + P_{2,1})^2}{(P_{1,2} + P_{2,1})^6} \\
&= \frac{P_{1,2}^2P_{2,1}^2}{(P_{1,2} + P_{2,1})^4} \\
&\leq \frac{P_{1,2}^2P_{2,1}^2}{16P_{1,2}^2P_{2,1}^2} && \text{using } (P_{1,2} + P_{2,1})^2 \geq 4P_{1,2}P_{2,1} \\
&= \frac{1}{16}.
\end{aligned}
$$

So,

$$
\| \sum_{s=1}^{k} \sum_{u=1}^{k} \pi_u^2 P_{u,s} \left(\pi_s - (P^n)_{u,s}\right) \| \leq \frac{1}{4} \left|1 - P_{1,2} - P_{2,1}\right|^n
$$

Now,

$$
\begin{aligned}
\mathbb{E}\left[-1\frac{1}{4}\left|1 - P_{1,2} - P_{2,1}\right|^n\right] &= -\frac{1}{4}\int_0^1\int_0^1 \left|1 - x - y\right|^n \\
&= -\frac{1}{4}\frac{2}{(n+1)(n+2)} \\
&= -\frac{1}{2(n+1)(n+2)}
\end{aligned}
$$

Notice that this decreases in $n$, and at $n = 3$, $\frac{1}{2(3+1)(3+2)} = \frac{1}{40} = 0.025$ which is less in magnitude than the value we proved at $n = 1$, $| -7/2 + 5\log 2 \approx 0.034$. So, solving for $n = 2$ (verified by a symbolic algebra program)

$$
\mathbb{E}\left[\frac{P_{1,2}P_{2,1}\left(-P_{1,2} - P_{2,1} + 1\right)^2 \cdot \left(2P_{1,2}P_{2,1} + P_{1,2}\left(P_{2,1} - 1\right) + P_{2,1}\left(P_{1,2} - 1\right)\right)}{\left(P_{1,2} + P_{2,1}\right)^3}\right] = -\frac{413}{60} + \frac{149\log(2)}{15} \approx 0.002
$$

Which is not only greater than $-7/2 + 5\log 2$, but positive. Lastly, we simply need to show that the inequality holds at $n = 0$, and we are done.

$$
\begin{aligned}
&\mathbb{E}\left[\frac{P_{1,2}P_{2,1}\left(-P_{1,2} - P_{2,1} + 1\right)^0 \cdot \left(2P_{1,2}P_{2,1} + P_{1,2}\left(P_{2,1} - 1\right) + P_{2,1}\left(P_{1,2} - 1\right)\right)}{\left(P_{1,2} + P_{2,1}\right)^3}\right] \\
&= -\mathbb{E}\left[\frac{P_{1,2}P_{2,1} \cdot \left(2P_{1,2}P_{2,1} + P_{1,2}\left(P_{2,1} - 1\right) + P_{2,1}\left(P_{1,2} - 1\right)\right)}{\left(P_{1,2} + P_{2,1}\right)^3}\right] \\
&= -7/6 + 5 * log(2)/3 \approx -0.0114
\end{aligned}
$$

Which is greater than $-7/2 + 5\log 2$, completing our proof. $\square$

**Lemma D.5.** *If $P$ is a uniformly random doubly stochastic matrix, then,*

$$\mathbb{E}\left[\pi_a^2\left(\frac{1}{k}-\sum_{s=1}^{k}P_{a,s}\pi_s\right)\right]=\mathbb{E}\left[\pi_a\pi_b\left(\frac{1}{k}-\sum_{s=1}^{k}P_{a,s}\pi_s\right)\right]$$

*for all $a, b$ and*

$$\mathbb{E}\left[\sum_{s=1}^{k}\sum_{u=1}^{k}\pi_u^2P_{u,s}\left(\pi_s-P_{u,s}\right)\right]<\mathbb{E}\left[\sum_{s=1}^{k}\sum_{u=1}^{k}\pi_u^2P_{u,s}\left(\pi_s-\left(P^a\right)_{u,s}\right)\right]$$

*For all non-negative $a \neq 1$. and,*

$$\mathbb{E}\left[\sum_{s=1}^{k}\sum_{u=1}^{k}\pi_u^2P_{u,s}\left(\pi_s-P_{u,s}\right)\right]<0$$

*Proof.* We will use the fact that for doubly stochastic matrices, the stationary distribution is the uniform vector $\frac{1}{k}\mathbf{1}$.

The first equality follows directly from $\pi_a = \frac{1}{k} = \pi_b$. Now we will prove the inequality.

$$\begin{aligned}\mathbb{E}\left[\sum_{s=1}^{k}\sum_{u=1}^{k}\pi_u^2P_{u,s}\left(\pi_s-\left(P^a\right)_{u,s}\right)\right]&=\mathbb{E}\left[\sum_{s=1}^{k}\sum_{u=1}^{k}\frac{1}{k^2}P_{u,s}\left(\frac{1}{k}-\left(P^a\right)_{u,s}\right)\right]\\&=\frac{1}{k^2}-\frac{1}{k^2}\sum_{s=1}^{k}\sum_{u=1}^{k}\mathbb{E}\left[P_{u,s}\left(P^a\right)_{u,s}\right]\end{aligned}$$

By Cauchy Schwartz,

$$\begin{aligned}&=\frac{1}{k^2}-\frac{1}{k^2}\mathbb{E}\left[\langle P,P^a\rangle_F\right]\\&>\frac{1}{k^2}-\frac{1}{k^2}\mathbb{E}\left[\|P\|_F\|P^a\|_F\right]\end{aligned}$$

We can make the above inequality strict because Cauchy Schwartz is only tight when the vectorizations of $P$ and $P^a$ are linearly dependent, since both are still doubly stochastic, this can only happen when $P = P^a$, which occurs only when each row of $P$ is identical, which happens with probability zero. For now assume $a > 0$, then,

$$\begin{aligned}&=\frac{1}{k^2}-\frac{1}{k^2}\mathbb{E}\left[\|P\|_F\|P^a\|_F\right]\\&=\frac{1}{k^2}-\frac{1}{k^2}\mathbb{E}\left[\|P\|_F\|PP^{a-1}\|_F\right]\\&=\frac{1}{k^2}-\frac{1}{k^2}\mathbb{E}\left[\|P\|_F\|\sum_i\alpha_i\Lambda_iP^{a-1}\|_F\right]\\&\geq\frac{1}{k^2}-\frac{1}{k^2}\sum_i\alpha_i\mathbb{E}\left[\|P\|_F\|\Lambda_iP^{a-1}\|_F\right]\\&=\frac{1}{k^2}-\frac{1}{k^2}\sum_i\alpha_i\mathbb{E}\left[\|P\|_F\|P^{a-1}\|_F\right]\\&=\frac{1}{k^2}-\frac{1}{k^2}\mathbb{E}\left[\|P\|_F\|P^{a-1}\|_F\right]\end{aligned}$$

The third step used the well known Birkhoff-Von Nuemann Theorem [7] that any doubly stochastic matrix $P$ is the convex combination of permutation matrices, so $P = \sum_i \alpha_i\Lambda_i$ for some permutation matrices $\Lambda_i$ and constants $\alpha_i > 0$ with $\sum_i \alpha_i = 1$. The inequality step uses Jensen's inequality. Induction on positive $a$ yields the desired inequality for positive $a$.

Now consider the remaining case, $a = 0$,

$$\frac{1}{k^2} - \frac{1}{k^2}\sum_{s=1}^{k}\sum_{u=1}^{k}\mathbb{E}\left[P_{u,s}\left(P^0\right)_{u,s}\right] = \frac{1}{k^2} - \frac{1}{k^2}\sum_{s=1}^{k}\mathbb{E}\left[P_{s,s}\right]$$

Since cycling each row or column by 1 in a doubly stochastic matrix results in a doubly stochastic matrix, by symmetry the marginal distributions of any two entries in $P$ are identical, so,

$$\frac{1}{k^2} - \frac{1}{k^2}\sum_{s=1}^{k}\mathbb{E}\left[P_{s,s}\right] = \frac{1}{k^2} - \frac{1}{k^2} = 0$$

While at $a = 1$, we have

$$\frac{1}{k^2} - \frac{1}{k^2}\sum_{s=1}^{k}\sum_{u=1}^{k}\mathbb{E}\left[P_{u,s}^2\right] < 0$$

Note that equality only occurs when $P = \mathbb{1}\mathbb{1}^\top$, which occurs with probability 0, hence why the inequality is strict.

$\square$

**Lemma D.6.** *If $P$ is a uniformly random $k$ by $k$ stochastic matrix subject to each row being the same, then,*

$$\mathbb{E}\left[\pi_a^2\left(\frac{1}{k} - \sum_{s=1}^{k}P_{a,s}\pi_s\right)\right] < \mathbb{E}\left[\pi_a\pi_b\left(\frac{1}{k} - \sum_{s=1}^{k}P_{a,s}\pi_s\right)\right] < 0$$

*and*

$$\frac{\mathbb{E}\left[\pi_a^2\left(\frac{1}{k} - \sum_{s=1}^{k}P_{a,s}\pi_s\right)\right]}{\mathbb{E}\left[\pi_a\pi_b\left(\frac{1}{k} - \sum_{s=1}^{k}P_{a,s}\pi_s\right)\right]} \geq \frac{8}{5}$$

*for all $a$ and $b$ and*

$$\mathbb{E}\left[\sum_{s=1}^{k}\sum_{u=1}^{k}\pi_u^2 P_{u,s}\left(\pi_s - \left(P^a\right)_{u,s}\right)\right] = 0$$

*For all $a$.*

*Proof.* The equality statement follows from the facts that for such transition matrices, $P^a = P$ for all natural $a > 0$, and that the stationary distribution matches the rows, that is, for any $a, b$, $\pi_b = P_{a,b}$,

$$\mathbb{E}\left[\sum_{s=1}^{k}\sum_{u=1}^{k}\pi_u^2 P_{u,s}\left(\pi_s - \left(P^a\right)_{u,s}\right)\right] = \mathbb{E}\left[\sum_{s=1}^{k}\sum_{u=1}^{k}\pi_u^2 P_{u,s}\left(\pi_s - P_{u,s}\right)\right] \quad = \mathbb{E}\left[\sum_{s=1}^{k}\sum_{u=1}^{k}\pi_u^2 P_{u,s}\left(\pi_s - \pi_s\right)\right] = 0$$

Now we will do the inequalities. We will also use the following facts derived from the moments of the Dirichlet distribution,

$$E\left[\|\pi\|_2^2\right] = \frac{2}{k+1}$$
$$E\left[\|\pi\|_2^4\right] = \frac{4(k+5)}{(k+1)(k+2)(k+3)}$$

So,

$$\mathbb{E}\left[\pi_a^2\left(\frac{1}{k} - \sum_{s=1}^{k}P_{a,s}\pi_s\right)\right] = \mathbb{E}\left[\pi_a^2\left(\frac{1}{k} - \sum_{s=1}^{k}\pi_s^2\right)\right]$$
$$= \frac{1}{k}\mathbb{E}\left[\|\pi\|_2^2\left(\frac{1}{k} - \|\pi\|_2^2\right)\right]$$

$$= \frac{1}{k^2}\mathbb{E}\left[\|\pi\|_2^2\right] - \frac{1}{k}\mathbb{E}\left[\left(\|\pi\|_2^4\right)\right]$$
$$= \frac{2}{k^2(k+1)} - \frac{4(k+5)}{k(k+1)(k+2)(k+3)}$$

Which is negative for all $k \geq 2$. And,

$$\begin{aligned}\mathbb{E}\left[\pi_a\pi_b\left(\frac{1}{k} - \sum_{s=1}^{k} P_{a,s}\pi_s\right)\right] =& \mathbb{E}\left[\pi_a\pi_b\left(\frac{1}{k} - \sum_{s=1}^{k}\pi_s^2\right)\right] \\ =& \frac{1}{k^2}\mathbb{E}\left[\left(\frac{1}{k} - \|\pi\|_2^2\right)\right] \\ =& \frac{1}{k^3} - \frac{1}{k^2}\mathbb{E}\left[\|\pi\|_2^2\right] \\ =& \frac{1}{k^3} - \frac{2}{k^2(k+1)}\end{aligned}$$

Which is also negative for all $k \geq 2$. Finally, notice that

$$\frac{\frac{2}{k^2(k+1)} - \frac{4(k+5)}{k(k+1)(k+2)(k+3)}}{\frac{1}{k^3} - \frac{2}{k^2(k+1)}} \geq \frac{8}{5}$$

For all $k \geq 2$. □

### D.6 Approximation Lemmas

The following lemma is a well known property of stochastic matricies, (see Lemma 3.3.2 Gallager [17] for example).

**Lemma D.7.** *Let $\alpha = 1 - 2\min_{i,j} P_{i,j}$. Then, for any $i, j$*

$$\left|\left(P^n\right)_{i,j} - \pi_j\right| \leq \alpha^n$$

Lemma D.8 and lemma D.9 both share similar intuitions and proofs. They largely rely on lemma D.7, which shows that $\left(P^n\right)_{i,j}$ approaches $\pi_j$ exponentially fast with respect to $n$, to show that over the course of summations over $n$ the stationary distribution dominates, allowing us to simplify the expressions.

**Lemma D.8.** *Let $P$ be a stochastic matrix with all positive entries, and let $a, b$ be states. Assume that $\min_{i,j} P_{i,j}$ is positive and doesn't dependend on $T$. Then,*

$$\pi_b \sum_{i=0}^{T}\left(\frac{1}{k}\left(P^i\right)_{b,a} - \sum_{s=1}^{k} P_{a,s}\frac{1}{T-i+1}\left(P^i\right)_{s,a}\sum_{j=0}^{T-i}\left(P^j\right)_{b,s}\right)$$
$$= \pi_b\pi_a(T+1)\left(\frac{1}{k} - \sum_{s=1}^{k} P_{a,s}\pi_s\right) + O(\log T).$$

*Proof.* Let us bound the magnitude of the difference between the two expressions.

$$\begin{aligned}&\left|\pi_b \sum_{i=0}^{T}\left(\frac{1}{k}\left(P^i\right)_{b,a} - \sum_{s=1}^{k} P_{a,s}\frac{1}{T-i+1}\left(P^i\right)_{s,a}\sum_{j=0}^{T-i}\left(P^j\right)_{b,s}\right) - \pi_b\pi_a(T+1)\left(\frac{1}{k} - \sum_{s=1}^{k} P_{a,s}\pi_s\right)\right| \\ =& \left|\pi_b \sum_{i=0}^{T}\left(\frac{1}{k}\left(\left(P^i\right)_{b,a} - \pi_a\right) - \sum_{s=1}^{k} P_{a,s}\left(\frac{1}{T-i+1}\left(P^i\right)_{s,a}\sum_{j=0}^{T-i}\left(P^j\right)_{b,s} - \pi_s\pi_a\right)\right)\right| \\ \leq& \pi_b \sum_{i=0}^{T}\left(\frac{1}{k}\left|\left(P^i\right)_{b,a} - \pi_a\right| + \sum_{s=1}^{k} P_{a,s}\frac{1}{T-i+1}\sum_{j=0}^{T-i}\left|\left(P^i\right)_{s,a}\left(P^j\right)_{b,s} - \pi_s\pi_a\right|\right)\end{aligned}$$

$$
\begin{aligned}
&\leq \pi_b \sum_{i=0}^{T} \left( \frac{1}{k}\alpha^i + \sum_{s=1}^{k} P_{a,s} \frac{1}{T-i+1} \sum_{j=0}^{T-i} \left| \left(P^i\right)_{s,a} \left(P^j\right)_{b,s} - \pi_s \pi_a \right| \right) \\
&= \pi_b \sum_{i=0}^{T} \left( \frac{1}{k}\alpha^i + \sum_{s=1}^{k} P_{a,s} \frac{1}{T-i+1} \sum_{j=0}^{T-i} \left| (\left(P^j\right)_{b,s} \left(P^i\right)_{s,a} - \pi_a) + \pi_a (\left(P^j\right)_{b,s} - \pi_s) \right| \right) \\
&\leq \pi_b \sum_{i=0}^{T} \left( \frac{1}{k}\alpha^i + \sum_{s=1}^{k} P_{a,s} \frac{1}{T-i+1} \sum_{j=0}^{T-i} \left( \left(P^j\right)_{b,s} \left| \left(P^i\right)_{s,a} - \pi_a \right| + \pi_a \left| \left(P^j\right)_{b,s} - \pi_s \right| \right) \right) \\
&\leq \pi_b \sum_{i=0}^{T} \left( \frac{1}{k}\alpha^i + \sum_{s=1}^{k} P_{a,s} \frac{1}{T-i+1} \sum_{j=0}^{T-i} \left( \left(P^j\right)_{b,s} \alpha^i + \pi_a \alpha^j \right) \right) \\
&\leq \pi_b \sum_{i=0}^{T} \left( \frac{1}{k}\alpha^i + \sum_{s=1}^{k} P_{a,s} \frac{1}{T-i+1} \sum_{j=0}^{T-i} \left( \alpha^i + \alpha^j \right) \right) \\
&\leq \pi_b \sum_{i=0}^{T} \left( \frac{1}{k}\alpha^i + \sum_{s=1}^{k} P_{a,s} \left( \alpha^i + \frac{1}{T-i+1} \frac{1-\alpha^{T-i+1}}{1-\alpha} \right) \right) \\
&\leq \pi_b \sum_{i=0}^{T} \left( \frac{1}{k}\alpha^i + \sum_{s=1}^{k} P_{a,s} \left( \alpha^i + \frac{1}{T-i+1} \frac{1}{1-\alpha} \right) \right) \\
&\leq \pi_b \sum_{i=0}^{T} \left( \frac{1}{k}\alpha^i + \alpha^i + \frac{1}{T-i+1} \frac{1}{1-\alpha} \right) \\
&\leq \pi_b \sum_{i=0}^{T} \left( \frac{1}{k}\alpha^i + \alpha^i + \frac{1}{T-i+1} \frac{1}{1-\alpha} \right) \\
&\leq \pi_b \left( \left(1 + \frac{1}{k}\right) \frac{1-\alpha^{T+1}}{1-\alpha} + \frac{\log(T+1)+1}{1-\alpha} \right) \\
&\leq \pi_b \frac{2 + \frac{1}{k} + \log(T+1)}{1-\alpha} \\
&\leq \frac{2\log T}{1-\alpha} \\
&= \frac{\log T}{\min_{i,j} P_{i,j}} \\
&= O(\log T)
\end{aligned}
$$

The last step follows from our assumption, completing the proof. $\square$

**Lemma D.9.** *Let $P$ be a stochastic matrix with all positive entries, and let $a, b$ be states. Assume that $\min_{i,j} P_{i,j}$ is positive and doesn't depend on $T$. Then,*

$$
\begin{aligned}
&\sum_{s=1}^{k} \sum_{u=1}^{k} \left( \frac{1}{k} - P_{u,s} \right) \sum_{i=a}^{T} \frac{\pi_u}{T-i+1} \left(P^i\right)_{s,u} \left( \left(P^a\right)_{u,s} - \frac{1}{T-i+1} \sum_{j=0}^{T-i} \left(P^j\right)_{u,s} \right) \\
&\qquad = \left(\log\left(T+1\right) - \log\left(a+1\right)\right) \sum_{s=1}^{k} \sum_{u=1}^{k} \pi_u^2 P_{u,s} \left( \pi_s - \left(P^a\right)_{u,s} \right) + O(1)
\end{aligned}
$$

*Proof.* First notice that,

$$
\sum_{s=1}^{k} \sum_{u=1}^{k} \left( \frac{1}{k} - P_{u,s} \right) \pi_u^2 \left( \left(P^a\right)_{u,s} - \pi_s \right) = \sum_{s=1}^{k} \sum_{u=1}^{k} \pi_u^2 \left( \frac{1}{k} \left( \left(P^a\right)_{u,s} - \pi_s \right) - P_{u,s} \left( \left(P^a\right)_{u,s} - \pi_s \right) \right)
$$

$$
\begin{aligned}
&= \sum_{u=1}^{k} \pi_u^2 \left( \sum_{s=1}^{k} \frac{1}{k} \left( (P^a)_{u,s} - \pi_s \right) - \sum_{s=1}^{k} P_{u,s} \left( (P^a)_{u,s} - \pi_s \right) \right) \\
&= \sum_{u=1}^{k} \pi_u^2 \left( \frac{1}{k} (1-1) - \sum_{s=1}^{k} P_{u,s} \left( (P^a)_{u,s} - \pi_s \right) \right) \\
&= \sum_{u=1}^{k} \pi_u^2 \sum_{s=1}^{k} P_{u,s} \left( \pi_s - (P^a)_{u,s} \right)
\end{aligned}
$$

571 We will bound the distance between $\sum_{s=1}^{k} \sum_{u=1}^{k} \left( \frac{1}{k} - P_{u,s} \right) \pi_u^2 \left( (P^a)_{u,s} - \pi_s \right)$ and $\mathbb{E}_{x|P} \left[ \frac{\partial L_T}{\partial v_a} \right]$.
572 Define $\alpha = 1 - 2 \min_{i,j} P_{i,j}$ as in lemma D.7.

$$
\begin{aligned}
&= \Bigg| \sum_{s=1}^{k} \sum_{u=1}^{k} \left( \frac{1}{k} - P_{u,s} \right) \sum_{i=a}^{T} \frac{\pi_u}{T-i+1} \left(P^i\right)_{s,u} \left( (P^a)_{u,s} - \frac{1}{T-i+1} \sum_{j=0}^{T-i} \left(P^j\right)_{u,s} \right) \\
&\qquad - \sum_{s=1}^{k} \sum_{u=1}^{k} \left( \frac{1}{k} - P_{u,s} \right) \sum_{i=a}^{T} \pi_u^2 \frac{1}{T-i+1} \left( (P^a)_{u,s} - \pi_s \right) \Bigg| \\
&= \left| \sum_{s=1}^{k} \sum_{u=1}^{k} \left( \frac{1}{k} - P_{u,s} \right) \sum_{i=a}^{T} \frac{\pi_u}{T-i+1} \left( \left(P^i\right)_{s,u} \left( (P^a)_{u,s} - \frac{1}{T-i+1} \sum_{j=0}^{T-i} \left(P^j\right)_{u,s} \right) - \pi_u \left( (P^a)_{u,s} - \pi_s \right) \right) \right| \\
&\le \sum_{s=1}^{k} \sum_{u=1}^{k} \sum_{i=a}^{T} \frac{\pi_u}{T-i+1} \left| \left(P^i\right)_{s,u} \left( (P^a)_{u,s} - \frac{1}{T-i+1} \sum_{j=0}^{T-i} \left(P^j\right)_{u,s} \right) - \pi_u \left( (P^a)_{u,s} - \pi_s \right) \right| \\
&\le \sum_{s=1}^{k} \sum_{u=1}^{k} \sum_{i=a}^{T} \frac{\pi_u}{T-i+1} \left| \left(P^i\right)_{s,u} \left( \pi_s - \frac{1}{T-i+1} \sum_{j=0}^{T-i} \left(P^j\right)_{u,s} \right) - \left( (P^a)_{u,s} - \pi_s \right) \left( \left(P^i\right)_{s,u} - \pi_u \right) \right| \\
&\le \sum_{s=1}^{k} \sum_{u=1}^{k} \sum_{i=a}^{T} \frac{\pi_u}{T-i+1} \left| \left(P^i\right)_{s,u} \frac{1}{T-i+1} \sum_{j=0}^{T-i} \left( \pi_s - \left(P^j\right)_{u,s} \right) - \left( (P^a)_{u,s} - \pi_s \right) \left( \left(P^i\right)_{s,u} - \pi_u \right) \right| \\
&\le \sum_{s=1}^{k} \sum_{u=1}^{k} \sum_{i=a}^{T} \frac{\pi_u}{T-i+1} \left( \left(P^i\right)_{s,u} \frac{1}{T-i+1} \sum_{j=0}^{T-i} \left| \pi_s - \left(P^j\right)_{u,s} \right| + \left( (P^a)_{u,s} - \pi_s \right) \left| \left(P^i\right)_{s,u} - \pi_u \right| \right) \\
&\le \sum_{s=1}^{k} \sum_{u=1}^{k} \sum_{i=a}^{T} \frac{\pi_u}{T-i+1} \left( \left(P^i\right)_{s,u} \frac{1}{T-i+1} \sum_{j=0}^{T-i} \alpha^j + \left( (P^a)_{u,s} - \pi_s \right) \alpha^i \right) \qquad \text{By lemma D.7} \\
&= \sum_{s=1}^{k} \sum_{u=1}^{k} \sum_{i=a}^{T} \frac{\pi_u}{T-i+1} \left( \left(P^i\right)_{s,u} \frac{1}{T-i+1} \frac{1-\alpha^{T-i+1}}{1-\alpha} + \left( (P^a)_{u,s} - \pi_s \right) \alpha^i \right) \\
&\le \sum_{s=1}^{k} \sum_{u=1}^{k} \sum_{i=a}^{T} \frac{\pi_u}{T-i+1} \left( \frac{1}{T-i+1} \frac{1}{1-\alpha} + \left( (P^a)_{u,s} - \pi_s \right) \alpha^i \right) \\
&\le \sum_{i=a}^{T} \frac{1}{T-i+1} \left( \sum_{s=1}^{k} \frac{1}{T-i+1} \frac{1}{1-\alpha} + \sum_{s=1}^{k} \left( \sum_{u=1}^{k} \pi_u \left(P^a\right)_{u,s} - \pi_s \right) \alpha^i \right) \\
&\le \sum_{i=a}^{T} \frac{1}{T-i+1} \left( \sum_{s=1}^{k} \frac{1}{T-i+1} \frac{1}{1-\alpha} + \sum_{s=1}^{k} \left( \pi_s - \pi_s \right) \alpha^i \right) \\
&\le \sum_{i=a}^{T} \frac{1}{T-i+1} \left( \frac{k}{T-i+1} \frac{1}{1-\alpha} \right)
\end{aligned}
$$

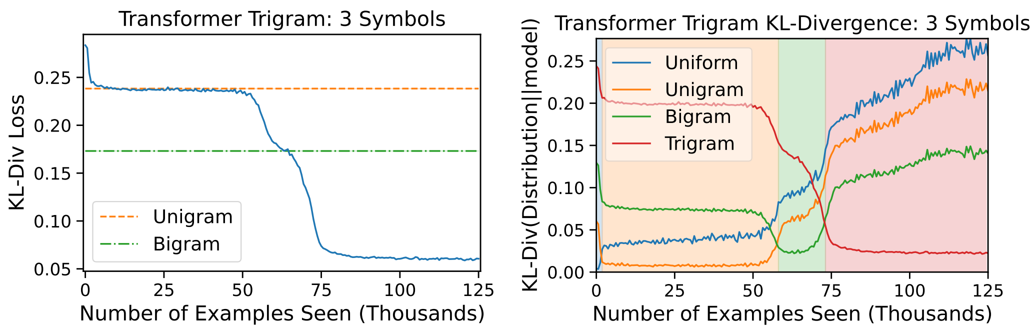


Figure 12: Three-headed transformer trained on In-Context Learning 3-grams (trigrams), with context length 200. (*left*) Loss during training. The model hierarchically converges close to the Bayes optimal solution. (*right*) KL divergence between the model and different strategies during training. As we observe, there are 4 stages of learning, each of them corresponding to a different algorithm implemented by the model.

$$\begin{aligned}
&\leq \sum_{i=a}^{T} \frac{1}{(T-i+1)^2} \frac{k}{1-\alpha} \\
&\leq \frac{2k}{1-\alpha} \\
&= \frac{k}{\min_{i,j} P_{i,j}} \\
&= O(1)
\end{aligned}$$

The last step follows from our assumption, and the fact that $k$ does not depend on $T$. $\square$

## E Beyond Bigrams: $n$-gram Statistics

Finally, we investigate the performance of transformers on learning in-context $n$-grams for $n > 2$; in particular, 3-grams. We train attention-only transformers with three heads in each layer by minimizing the in-context cross entropy loss with the Adam optimizer. As can be seen in Figure 12 (left), the model eventually converges to the Bayes optimal solution. Interestingly, as in the case of Markov Chains, the model displays a "hierarchical learning" behavior characterized by long plateaus and sudden drops. In this setup, the different strategies correspond to unigrams, bigrams and trigrams, respectively. This is presented clearly on the right of Figure 12, where we plot the similarity of the model with the different strategies and it exhibits the same clear pattern as in the case of $n = 2$. Curiously, single attention headed models could not achieve better performance than bigrams. We leave a more thorough investigation of $n$-grams for future work. This behaviour is much less stable for different number of heads and tokens. With two heads or four heads, there is sometimes no bigram phase and faster convergence.

## NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: All claims made in our paper are supported by theoretical proofs or rigorous empirical evaluations.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.

- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The purpose of this work is to introduce a novel task to study in-context learning in transformers. Since it focuses on a theoretical understanding, certain assumptions must be made. We mention in the introduction that our theoretical results apply to a simplified model of a transformer, while we precisely define our experimental setup.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: Our main paper may have some informal theorem statements for ease of readability, we give formal versions of all the statements with full proofs in the appendix. We also make sure to clarify the assumptions under which our results hold.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.

- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide all training details in Appendix C, and information on the data generating process and architecture in Appendix B.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: We plan to release the code prior to publication. However, we believe our experiments are easy to reproduce with the provided information.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.

- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: The details are available in Appendix C.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: Our main results regard the multiphase nature of training, and we include Figure 5 which shows that for the seeds $0, 1, \ldots 9$ the model each time has the same patterns in the training curve. The curves were shown individually instead of through error bars since the main purpose of the loss curves is to show the shape, but because the phase transition doesn't occur at a consistent time, adding error bars smooths out the curve making the phase transition look less sharp.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The resources are described in Appendix C. Only a single computer was used, and none of the training runs took longer than ten minutes.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We read the code of ethics in its entirety and strongly believe that our research abides by the stated code.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: Our work focuses on understanding the internal mechanisms of Transformer models on synthetic tasks. We do not foresee any direct societal impact of this work.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: Our work does not release any data or models that poses safety risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We cite the Github repository we use as the codebase for our Transformer models.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: We do not introduce any new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: Our work does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: Our work does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.