Outcome-Based RL Provably Leads Transformers to Reason, but Only With the Right Data

Yuval Ran-Milo[*†] Tel Aviv University
Yotam Alexander[*] Tel Aviv University
Shahar Mendel Tel Aviv University
Nadav Cohen Tel Aviv University

Abstract

Transformers trained via Reinforcement Learning (RL) with outcome-based supervision can spontaneously develop the ability to generate intermediate reasoning steps (Chain-of-Thought). Yet the mechanism by which sparse rewards drive policy gradient to discover such systematic reasoning remains poorly understood. We address this by analyzing the policy gradient dynamics of single-layer linear Transformers on a synthetic graph traversal task that cannot be solved without Chain-of-Thought but admits a simple iterative solution. We prove that despite training solely on final-answer correctness, policy gradient drives the Transformer to converge to a structured, interpretable algorithm that iteratively traverses the graph vertex-by-vertex. We characterize the distributional properties required for this emergence, identifying the critical role of "simple examples": instances requiring fewer reasoning steps. When the training distribution places sufficient mass on these simpler examples, the Transformer learns a generalizable traversal strategy that extrapolates to longer chains; when this mass vanishes, policy gradient learning becomes infeasible. We corroborate our theoretical results through experiments on synthetic data and with real-world language models on mathematical reasoning tasks, validating that our theoretical findings carry over to practical settings.

1 Introduction

The Transformer architecture [66] has become the de facto standard for sequence modeling, powering advances in natural language processing, code generation, protein structure prediction, and mathematical problem solving [16, 41, 38]. While next-token pretraining on large corpora remains highly effective, it has been shown that reinforcement learning (RL) can further improve Transformer reasoning on tasks such as mathematics and code generation [62, 20]. In particular, models such as DeepSeek-R1 demonstrate that optimizing only for final-answer correctness with algorithms such as GRPO [62] or PPO [61] can induce detailed Chain-of-Thought (CoT) reasoning traces [19]. Yet the fundamental question of how sparse, outcome-based feedback at the end of generation guides policy gradient toward a systematic reasoning algorithm remains open. The need for such an explanation is sharpened by empirical evidence that reasoning emerges only under certain training distributions, often requiring examples of varying complexity [51, 54].

We study this question through graph traversal, a common setting for Transformer reasoning [58, 2, 64, 6, 59, 48]. In our task, the model must identify the terminal vertex of a chain in a directed graph. Under standard complexity-theoretic assumptions, this task requires iterative reasoning. Furthermore, when the model is allowed to generate intermediate reasoning steps, it admits multiple solutions, some of which are more efficient than others. Our analysis shows that, despite training solely on final-answer correctness, policy gradient can amplify a minimal level of competence at initialization, driving the Transformer to explicitly traverse the graph vertex-by-vertex. This occurs even though the

[*]Equal contribution.
[†]Correspondence to yuvalmilo@mail.tau.ac.il.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

model can implement alternative, less efficient solutions, revealing an *implicit bias* in the learning dynamics towards efficient reasoning algorithms.

We further characterize the distributional properties required for the above implicit bias, identifying a critical role of "simple examples"—instances requiring few reasoning steps—in guiding the optimization. We prove that when the training data distribution has sufficient mass on these simpler examples, policy gradient learns the efficient traversal algorithm; conversely, when this mass vanishes, policy gradient learning becomes infeasible as the chain length grows. We support our theoretical results with experiments in both synthetic and real-world settings, including RL post-training of Qwen-based models [56] on mathematical reasoning tasks.

Taken together, these findings suggest that deliberately introducing distribution shift during post-training can change the implicit bias of policy gradient and improve generalization: by training on simple examples, policy gradient can be steered toward efficient reasoning algorithms, despite the existence of less efficient solutions. Large-scale empirical results in the literature are directionally consistent with this mechanism [65, 31, 49, 37], and extending our simplified theory to more realistic settings, such as modern RL with verifiable rewards (RLVR) systems and large-scale reasoning benchmarks, is a direction we hope to spur.

1.1 Paper Organization

The remainder of the paper is organized as follows. Sections 2 and 3 introduce the Transformer model and chain identification task. Section 4 shows that multi-step reasoning is necessary and sufficient for the task, and that multiple algorithms can achieve high reward, with some being more efficient than others. Section 5 analyzes policy gradient dynamics. Section 6 reports experiments, Section 7 discusses the limitations of our work, and Section 8 concludes. Related work is discussed in Appendix A.

2 Preliminaries

2.1 Notation

Let $\mathbb{1}$ denote the indicator function and $[n] \coloneqq \{1, \dots, n\}$. For a sequence of tokens $s = (s_1, \dots, s_L)$, and for any $1 \le j \le l \le L$ we use the shorthand $s_{j:l} \coloneqq (s_j, \dots, s_l)$. Let Σ denote a finite vocabulary. Without loss of generality, we identify Σ with $[|\Sigma|]$, allowing us to index vocabulary tokens by integers. For any matrix $M \in \mathbb{R}^{|\Sigma|\times|\Sigma|}$ (or $M \in \mathbb{R}^{|\Sigma|\times d}$ for some $d \in \mathbb{N}$) and tokens $s, s' \in \Sigma$, we write $M_{s,s'}$ (or $M_{s,:}$) for the entry (or row) of M indexed by the tokens.

2.2 Transformer Architecture

Let Σ denote a finite vocabulary. We consider a Transformer architecture with d layers that maps sequences $s_{1:T} = (s_1, \dots, s_T) \in \Sigma^T$ of arbitrary length T to a probability distribution over the vocabulary Σ.

The Transformer first encodes the input sequence using one-hot embeddings: for each token $s_l \in \Sigma$, we define $x_l = e_{s_l} \in \mathbb{R}^{|\Sigma|\times 1}$ to be the one-hot vector corresponding to s_l. We denote the resulting embedding matrix by $H^{(0)} = [x_1, \dots, x_T]^\top \in \mathbb{R}^{T\times|\Sigma|}$.

The Transformer then applies d attention layers sequentially. We parameterize the Transformer by $\theta = \{(A_l, V_l)\}_{l=1}^d$, where each layer $l \in [d]$ has an *attention matrix* $A_l \in \mathbb{R}^{|\Sigma|\times|\Sigma|}$ and a *value matrix* $V_l \in \mathbb{R}^{|\Sigma|\times|\Sigma|}$. In the single-layer case ($d = 1$), we simplify notation by writing $A = A_1$ and $V = V_1$. At each layer $l \in [d]$, the hidden state, $H^{(l)} \in \mathbb{R}^{T\times|\Sigma|}$, is computed using an activation function σ and a causal masking operator MASK_σ which ensures that position i attends only to positions $j \le i$:

$$H^{(l)} = \sigma\Big(\mathrm{MASK}_\sigma\big(H^{(l-1)} A_l (H^{(l-1)})^\top\big)\Big) H^{(l-1)} V_l \,. \tag{1}$$

We consider two instantiations. In the *softmax Transformer*, $\sigma = \mathrm{softmax}$ (applied row-wise) and $\mathrm{MASK}_\sigma(Z)_{ij} = Z_{ij}$ if $j \le i$ and $-\infty$ otherwise. In the *linear Transformer*, σ is the identity function and $\mathrm{MASK}_\sigma(Z)_{ij} = Z_{ij}$ if $j \le i$ and 0 otherwise. Unless otherwise specified, our statements apply to both variants.

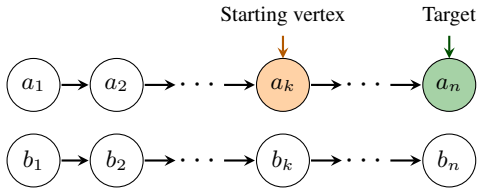


Figure 1: Illustration of the chain identification task. The input contains edges forming two disjoint chains $\mathcal{C}_a = (a_1, \ldots, a_n)$ and $\mathcal{C}_b = (b_1, \ldots, b_n)$, and a starting vertex v_{start} (here a_k). The task is to identify the terminal vertex of the chain containing v_{start} (here a_n).

Finally, the Transformer applies a softmax to the last position of the final hidden state to produce a probability distribution over the vocabulary:

$$\mathrm{TF}(s_{1:T}; \theta) = \mathrm{softmax}\big(H^{(d)}_{T,:}\big) \in \mathbb{R}^{1\times|\Sigma|}. \tag{2}$$

2.3 Autoregressive Generation

We use the Transformer architecture to generate sequences via autoregressive sampling. Given an initial prompt $s_{1:L_0}$, we define a set of *terminal tokens* $\Sigma_{\text{term}}(s_{1:L_0}) \subseteq \Sigma$. The Transformer generates tokens $s_{L_0+1}, s_{L_0+2}, \ldots$ sequentially. At each step $l \geq L_0$, the Transformer samples the next token $s_{l+1} \sim \mathrm{TF}(s_{1:l}; \theta)$. The Transformer continues sampling autoregressively until reaching a terminal token, *i.e.*, until $s_l \in \Sigma_{\text{term}}(s_{1:L_0})$ for some $l > L_0$. [3] We denote by $p_{\text{roll}}(\theta)$ the distribution over complete trajectories $\tau = (s_{1:L_0}, s_{L_0+1}, \ldots, s_{L_{\text{term}}})$ induced by the Transformer's parameters θ, where $s_{L_{\text{term}}} \in \Sigma_{\text{term}}(s_{1:L_0})$. We denote by $|\tau| = L_{\text{term}} - L_0$ the length of the generated trajectory (excluding the prompt). Given two parameter settings θ and θ', we measure the difference between their induced rollout distributions using the *total variation distance*:

$$\mathrm{TV}\big(p_{\text{roll}}(\theta), p_{\text{roll}}(\theta')\big) := \frac{1}{2}\sum_{\tau} \big|p_{\text{roll}}(\theta)[\tau] - p_{\text{roll}}(\theta')[\tau]\big|, \tag{3}$$

where the sum is over all possible rollouts τ.

2.4 Reinforcement Learning

We train the Transformer using reinforcement learning with a reward function over complete rollouts. Given a trajectory $\tau = (s_{1:L_0}, s_{L_0+1}, \ldots, s_{L_{\text{term}}})$, we denote a reward function $r(\tau)$ which evaluates the quality of the trajectory. For any distribution $\mathcal{D}$ over input sequences $s_{1:L_0}$, the training objective is defined as:

$$\mathcal{R}(\mathcal{D}, \theta) = \mathop{\mathbb{E}}_{s_{1:L_0}\sim\mathcal{D}} \mathop{\mathbb{E}}_{\tau\sim p_{\text{roll}}(\theta)} \big[r(\tau)\big].$$

Our interest lies in *outcome-based* reinforcement learning, where the reward $r(\tau)$ depends on the trajectory τ only through its final token $s_{L_{\text{term}}}$.

3 The Task: Chain Identification

We study a graph traversal task: given a set of *directed* edges forming two disjoint chains, and a starting vertex, the goal is to identify the terminal vertex of the chain containing the starting vertex (see Figure 1 for an illustration). This task captures the essential structural property of reasoning tasks: the answer cannot be produced in a single step (as we show in Section 4.1), but emerges naturally from a step-by-step iterative procedure.

Formally, let $n \in \mathbb{N}_{\geq 2}$ denote the task's chain length. We define the *vertex set* $\mathcal{V} = [2n]$ and the *edge vocabulary* $\mathcal{E} = \mathcal{V}\times\mathcal{V}$, where each ordered pair (u, v) denotes a directed edge from u to v. The full vocabulary is $\Sigma = \mathcal{V}\cup\mathcal{E}$. We define a *chain* for the task as a sequence of n distinct vertices $(v_1, \ldots, v_n)$ with $v_i \in \mathcal{V}$, which we identify with its directed edge set $\{(v_1, v_2), (v_2, v_3), \ldots, (v_{n-1}, v_n)\}$.

[3] It can be easily shown that, in the setting under investigation in this paper, the probability of reaching a terminal token is one.

3.1 Data Distribution

We define a general family of distributions $\mathcal{D}^{\mathcal{Q}}$ over input sequences, parameterized by a distribution $\mathcal{Q}$ over the index set $[n-1]$. To sample $s_{1:L_0} \sim \mathcal{D}^{\mathcal{Q}}$, we first partition $\mathcal{V}$ uniformly at random into two disjoint subsets of size n each, and uniformly order the vertices within each subset to form chains $\mathcal{C}_a = (a_1, \ldots, a_n)$ and $\mathcal{C}_b = (b_1, \ldots, b_n)$. We then sample an index $k \sim \mathcal{Q}$, choose a chain $\mathcal{C} \in \{\mathcal{C}_a, \mathcal{C}_b\}$ uniformly at random, and set the starting vertex v_{start} to be the k-th vertex of $\mathcal{C}$. The input sequence $s_{1:L_0}$ consists of an arbitrary ordering of all edges in $\mathcal{C}_a$ and $\mathcal{C}_b$, followed by v_{start}. [4] For an instance of the problem, *i.e.*, an input sequence $s_{1:L_0}$, we define the set of terminal tokens to include the final vertices of the two chains (a_n and b_n). Formally $\Sigma_{\text{term}}(s_{1:L_0}) = \{a_n, b_n\}$.

3.2 Task Reward

Given an input sequence $s_{1:L_0}$ containing edges forming two chains $\mathcal{C}_a = (a_1, \ldots, a_n)$ and $\mathcal{C}_b = (b_1, \ldots, b_n)$ with starting vertex v_{start}, let $\tau = (s_{1:L_0}, s_{L_0+1}, \ldots, s_{L_{\text{term}}})$ denote a rollout terminated at the first $s_{L_{\text{term}}} \in \Sigma_{\text{term}}(s_{1:L_0})$. We define $y^*(s_{1:L_0})$ as the terminal vertex of the chain containing v_{start}:

$$y^*(s_{1:L_0}) := \begin{cases} a_n & \text{if } v_{\text{start}} \in \mathcal{C}_a, \\ b_n & \text{if } v_{\text{start}} \in \mathcal{C}_b. \end{cases}$$

We use the reward $r(\tau) := \mathbb{1}\{s_{L_{\text{term}}} = y^*(s_{1:L_0})\}$.

3.3 Step Types and Chain Traversal

To characterize the Transformer's behavior during generation, we define three types of atomic operations. We say that the Transformer performs a *forward step* at time $t > L_0$ if the generated token s_t is the target of a forward edge from the previous token s_{t-1}, i.e., $(s_{t-1}, s_t) \in \mathcal{C}_a \cup \mathcal{C}_b$. A *backward step* occurs when s_t is the source of an edge to s_{t-1}, i.e., $(s_t, s_{t-1}) \in \mathcal{C}_a \cup \mathcal{C}_b$. Finally, a *switch step* occurs when s_t belongs to a different chain than s_{t-1}, that is, when $s_{t-1} \in \mathcal{C}_a$ and $s_t \in \mathcal{C}_b$, or vice versa.

A rollout $\tau = (s_{1:L_0}, s_{L_0+1}, \ldots, s_{L_{\text{term}}})$ is a *chain traversal* if it consists of a consecutive sequence of forward steps starting from v_{start}. Formally, we require that for all $l \in \{L_0 + 1, \ldots, L_{\text{term}}\}$, the token s_l is a forward step from s_{l-1} (where $s_{L_0} = v_{\text{start}}$).

4 Solutions to Chain Identification Task

In this section, we show that fixed-depth Transformers cannot solve chain identification without multi-step reasoning, but single-layer Transformers can solve it by autoregressively generating such steps. We also show that high reward can be achieved by multiple algorithms, ranging from efficient chain traversal to inefficient long trajectories, setting up the question of which solution policy gradient selects.

4.1 Reasoning is Necessary to Solve the Task

We first prove that multi-step reasoning is necessary: with any fixed depth Transformer, the task cannot be solved when the Transformer is required to output the answer in a single step only.

To formalize the result, we define the *single-step Reward*, which measures performance when the Transformer must output the answer immediately:

$$\mathcal{R}_{\text{single}}(\mathcal{D}^{\mathcal{Q}}, \theta) := \mathop{\mathbb{E}}_{s \sim \mathcal{D}^{\mathcal{Q}}} \mathop{\mathbb{E}}_{\hat{y} \sim \text{TF}(s;\theta)} \big[\mathbb{1}\{\hat{y} = y^*(s)\}\big]. \tag{4}$$

Theorem 1 shows that for Transformers of any fixed depth [5], there exists a distribution $\mathcal{D}^{\mathcal{Q}}$ where single-step generation achieves reward bounded away from 1 by a constant. This contrasts sharply

[4] While the order of edges is arbitrary (as we use no positional encodings), v_{start} must be the last token so that the Transformer can attend from it to the edges to predict the next vertex.

[5] The same analysis readily extends to architectures incorporating normalization layers and MLP layers of width polynomial in the input length alongside attention. Moreover, Theorem 1 extends to any constant number of chain-of-thought steps; see Appendix E.9 for both extensions.

with the results in Section 4.2, where autoregressive generation can achieve reward arbitrarily close to 1 on all distributions. This separation demonstrates the necessity of multi-step reasoning.

Theorem 1. *Assume transformer weights are represented with logarithmic precision. Then under standard complexity-theoretic assumptions (see Appendix E.2), for any fixed depth $d \in \mathbb{N}$, there exist $n \in \mathbb{N}_{\geq 2}$, a distribution $\mathcal{Q}$ over $[n-1]$, and $c \in \mathbb{R}_{>0}$ such that for all parameters $\theta = \{(A_i, V_i)\}_{i=1}^{d}$:*

$$\mathcal{R}_{\textit{single}}(\mathcal{D}^{\mathcal{Q}}, \theta) \leq 1 - c.$$

Proof sketch (full proof in Appendix E). Using results of Li et al. [42], we first note that constant-depth Transformers without CoT generation can only compute functions in TC^0, the class of constant-depth circuits with unbounded fan-in threshold gates. We then relate our chain identification task to standard problems in circuit complexity by giving reductions from the word problem on S_5 (the symmetric group on five elements) to the ORD problem (path reachability in directed graphs), and from ORD to the chain identification task. The word problem on S_5 is NC^1-complete, where NC^1 is the class of Boolean functions computable by polynomial-size, logarithmic-depth circuits with bounded fan-in, so these reductions show that solving chain identification in TC^0 would imply that an NC^1-complete problem lies in TC^0. Under the widely believed conjecture that $TC^0 \neq NC^1$, this is impossible, and hence the chain identification task is not in TC^0. □

4.2 Efficient and Inefficient Reasoning Algorithms Solve the Task

We next show that multi-step reasoning is sufficient: single-layer Transformers can achieve reward arbitrarily close to 1 by autoregressively generating intermediate steps. However, high reward does not identify a unique algorithm. Proposition 1 gives an efficient construction that follows the chain directly, while Proposition 2 shows that, because the reward penalizes only the final answer, arbitrarily long trajectories can also achieve high reward.

Proposition 1. *For any $n \in \mathbb{N}_{\geq 2}$, $\mathcal{Q}$ a distribution over $[n-1]$ and $\varepsilon, \delta \in \mathbb{R}_{>0}$, there exist parameters θ for single-layer Transformer such that $\mathcal{R}(\mathcal{D}^{\mathcal{Q}}, \theta) > 1 - \varepsilon$, and in addition, for every input sequence $s_{1:L_0}$ from $\mathcal{D}^{\mathcal{Q}}$, with probability at least $1 - \varepsilon$ over $\tau \sim p_{\textit{roll}}(\theta)$, the rollout τ is a chain traversal (Section 3.3).* [6]

Proof sketch (full proof in Appendix B). Assume by induction that the Transformer has correctly performed chain traversal up to a non-terminal vertex up to time l producing a sequence $s_{1:l}$ where $l \geq 2n - 1$. The prefix $s_{1:2n-2}$ contains the edges, and the last token s_l is a vertex. We construct the attention matrix A such that for any vertex $u \in \mathcal{V}$, the attention weight from u to any outgoing edge (u, v) is strictly larger than the weight to any other token. The value matrix V is constructed such that $V_{(u,v),v} = \alpha$ for some $\alpha \in \mathbb{R}_{>0}$ for all edges (u, v), while all other entries are zero. Since s_l is in some chain and is a non-terminal vertex, there exists some edge $(s_l, k) \in s_{1:2n-2}$. This construction ensures that the Transformer attends to (s_l, k) more strongly than to any other edge, and the output logits satisfy $\mathrm{TF}(s_{1:l})_k / \mathrm{TF}(s_{1:l})_v = \Omega(\alpha)$ for any $v \neq k$. Consequently, the probability of outputting the next vertex in the chain, k, becomes arbitrarily close to one for sufficiently large α, completing the induction. □

Proposition 2. *For any $n \in \mathbb{N}_{\geq 3}$, $\mathcal{Q}$ a distribution over $[n-1]$, $k \in \mathbb{N}$, and $\varepsilon, \delta \in \mathbb{R}_{>0}$, there exists parameters θ for a single-layer linear Transformer such that $\mathcal{R}(\mathcal{D}^{\mathcal{Q}}, \theta) > 1 - \varepsilon$, and for every input sequence $s_{1:L_0}$ from $\mathcal{D}^{\mathcal{Q}}$, the probability that $|\tau| > k$ is greater than $1 - \delta$.*

Proof sketch (full proof in Appendix C). We construct parameters similar to those in the proof of Proposition 1. However, instead of configuring the transformer's parameters to always move forward in the chain, we set the parameters such that from any vertex, the Transformer moves backward to its predecessor with high probability, forward to its successor with small probability, and to the other chain with negligible probability. This creates a random walk confined (with high probability) to the correct chain that frequently backtracks, producing arbitrarily long trajectories before eventually reaching the terminal vertex. □

[6] For simplicity, we present this result assuming infinite precision. However, one can show that under logarithmic precision, one can obtain reward strictly larger than the upper bound given by Theorem 1. For further details see Appendix B.

5 Dynamical Analysis

In this section, we analyze policy gradient dynamics for linear Transformers trained on $\mathcal{R}(\mathcal{D}^{\mathcal{Q}}, \theta)$ (Section 3), showing that success depends sharply on the training distribution. With sufficient mass on simple examples, policy gradient learns efficient chain traversal in polynomial time and the learned behavior generalizes to any test distribution, including unseen complex examples (Theorem 2). Since algorithms far less efficient than chain traversal also solve our chain identification task (Section 4.2), this demonstrates an *implicit bias* toward efficiency. [7] Without simple examples, learning is intractable: even after exponential time, the rollout distribution remains essentially unchanged from initialization (Theorem 3).

5.1 Data Distribution

We now describe the property of the data distribution that controls the dynamics. We characterize a data distribution based on the number of reasoning steps required to solve its examples. Recall from Section 3.1 that each example includes as starting vertex the k-th vertex of a chain, for some $k \in [n-1]$. Examples starting near the end of the chain (large k) require fewer autoregressive steps (Section 3.3), thus are regarded as simpler. We accordingly define a measure of simplicity for the data distribution.

Definition 1. *(Simplicity Mass).* Let $\mathcal{D}^{\mathcal{Q}}$ be a data distribution parameterized by a distribution $\mathcal{Q}$ over the index set $[n-1]$ (Section 3.1). For any $s \in [n-1]$ and $c \in [0,1]$, we say that $\mathcal{D}^{\mathcal{Q}}$ *has s-simplicity mass c* if $\sum_{i=n-s}^{n-1} \mathcal{Q}(i) = c$, i.e., if there is probability c that the starting vertex is within distance s from the end of its chain.

5.2 Technical Assumptions

We analyze policy gradient on the objective $\mathcal{R}(\mathcal{D}^{\mathcal{Q}}, \theta)$ under the technical assumptions described below. We make most of these assumptions for analytical tractability: the experiments in Appendix I.1 show that the main phenomena persist when they are relaxed.

Motivated by Section 4.2, which shows that single-layer Transformers can solve the task through autoregressive reasoning, we specialize the dynamics to such an architecture. In addition, we assume that the model outputs only vertices and never repeats the previous vertex. This condition can be enforced via output masking.

Assumption 1. (Architecture). *Throughout this section, we focus on single-layer linear Transformers, i.e., $d = 1$ and σ is the identity in Equation* (1)*. Thus, for $\theta = (A, V)$ and any prefix $s_{1:T}$, Equation* (1) *becomes*

$$H^{(1)} = \mathrm{MASK}_{\sigma}\left(H^{(0)} A (H^{(0)})^{\top}\right) H^{(0)} V, \qquad \mathrm{TF}(s_{1:T}; \theta) = \mathrm{softmax}\left(H^{(1)}_{T,:}\right).$$

As is common practice in Reinforcement Learning for Transformers, we consider policy gradient initialized from a base model that has already acquired a minimal level of task proficiency during pre-training [19, 53]. Concretely, we assume a weak initial preference for following the chain direction rather than reversing it, and also require non-negligible probability of switching chains so that the model is not initialized near either the efficient or inefficient solutions constructed in Section 4.2.

Assumption 2. (Pretraining). *For any input sequence $s_{1:L_0}$, let p_{fwd}, p_{bwd}, and p_{switch} denote the probabilities that the Transformer performs a forward, backward, or switch step, respectively, at the first step (time $t = L_0 + 1$). There exist constants $\zeta, \upsilon \in \mathbb{R}_{>0}$ such that, for any input sequence $s_{1:L_0}$,*

$$p_{fwd} > p_{bwd} + \zeta \qquad \text{and} \qquad p_{switch} > \upsilon.$$

We next specify the idealized optimization dynamics used in our analysis. We use gradient flow, the continuous-time limit of gradient-based optimization, which is known to closely approximate policy gradient with moderately small step sizes [21]. Following prior theoretical work on Transformers [52, 70], we also train only the attention matrix A while keeping the value matrix V fixed at initialization.

[7] While the constructions of Propositions 1 and 2 do not conform to the settings of Section 5.2, one can implement very similar constructions which do. For further details see Appendix D.

Assumption 3. (Gradient flow). *Training follows continuous-time gradient flow with exact expected gradients. Only the attention matrix A is trained, while the value matrix V is kept fixed at its initialization:*

$$\frac{dA(t)}{dt} = \nabla_A \mathcal{R}(\mathcal{D}, A(t)). \tag{5}$$

It remains to specify the initial parameterization of the attention and value matrices. The attention initialization below captures the idea that the Transformer begins without any preference toward specific vertices, treating all of them symmetrically.

Assumption 4. (Symmetric initialization). *The attention matrix A is initialized symmetrically.*[8] *Equivalently, there exist constants $\alpha, \beta, \gamma, \eta, \delta \in \mathbb{R}$ with $\alpha, \gamma > 0$ such that for any edge $(u, w) \in \mathcal{E}$ and vertices $v, w' \in \mathcal{V}$,*[9] *$A_{v,(u,w)} = \alpha$ if $v = u$, $A_{v,(u,w)} = \beta$ if $v = w$, $A_{v,(u,w)} = \gamma$ if $v \notin \{u, w\}$, $A_{v,w'} = \eta$ if $v = w'$, and $A_{v,w'} = \delta$ if $v \neq w'$.*

Finally, we fix a value matrix that encodes the endpoints of each edge token, analogous to the one-hot value initializations used in prior work [52, 70, 33].

Assumption 5. (Sparse Value). *We set the value vector of each edge token (u, w) to encode its defining vertices: $V_{(u,w),v} = \mathbb{1}\{v \in \{u, w\}\}$. Vertex tokens have zero value vectors: $V_{v,:} = 0$ for all $v \in \mathcal{V}$.*

Output masking, the fixed value matrix (Assumption 3 and Assumption 5), gradient flow (Assumption 3), and symmetric initialization (Assumption 4) are all empirically shown to be inessential (Appendix I.1), and removing them from the theory is an important direction for future work (see Section 7). The switch probability bound (Assumption 2) is necessary: without it, the initialization may already sit near a solution, and the question of which algorithm policy gradient selects is vacuous.

5.3 With Simple Examples Efficient Reasoning is Learned

Theorem 2 below essentially establishes that, when the training data distribution has a simplicity mass greater than zero for some *constant* distance s (Definition 1), the Transformer learns, in time polynomial in the chain length n, [10] to perform *forward steps* (Section 3.3) with arbitrarily high probability. The theorem additionally requires the same mass at distance $\lceil an \rceil$, which confines the support of training examples to two bands: sufficiently easy and sufficiently hard. This condition is likely an artifact of our proof technique: empirically, learning also succeeds when the training distribution includes examples of intermediate difficulty (Section 6 and Appendix I.1.1).

Theorem 2. *Let $s \in [n-1]$, $a \in (0, 1)$ and $c \in (0, 1]$. Suppose the s-simplicity mass and $\lceil an \rceil$-simplicity mass of the training data distribution equal c (Definition 1). Then, for any $\varepsilon > 0$, there exists $t = O(n^2 c^{-1} \varepsilon^{-2})$* [11] *such that the linear Transformer parameters at time t of optimization, i.e., $\theta(t) := (A(t), V)$, satisfy the following: for any input sequence $s_{1:L_0}$ drawn from any test data distribution (Section 3.1), at each step $l \geq L_0$ of the autoregressive generation $s_{l+1} \sim \mathrm{TF}(s_{1:l}; \theta(t))$ (Section 2.3), the probability of s_{l+1} being a forward step (Section 3.3) is at least $1 - \varepsilon$.* [12]

Proof sketch (full proof in Appendix F). By permutation symmetry of the initialization and data distribution, the rollout generated by $\theta(t)$ can be analyzed as a time-homogeneous Markov chain on a canonical two-chain graph. Its behavior is determined by three transition probabilities: forward, backward, and chain-switch steps, parameterized by α, β, γ. We decompose the same-side absorption probability $S(v)$, *i.e.*, the probability that the rollout terminates at the end of the chain it started on, according to whether the rollout makes a *long jump* (LJ), a transition that changes the distance to the terminal vertex by at least 2:

$$S(v) \;=\; (1 - h(v))\, \bar{S}^{NL}(v) \;+\; \tfrac{1}{2}\, h(v),$$

[8] More precisely, for any permutation π of the vertex set $\mathcal{V}$, extended naturally to edge tokens via $\pi((u, w)) = (\pi(u), \pi(w))$, the matrix A satisfies $A_{\pi(s),\pi(s')} = A_{s,s'}$ for all tokens $s, s' \in \Sigma$.

[9] Note that we do not specify the attention initialization between pairs of edge tokens; since the attending token is always a vertex, these values do not affect the dynamics.

[10] The bound applies to the time index of gradient flow; by discretizing with a fixed step size, this translates to a polynomial number of policy gradient iterations [21]. See Appendix F.10 for further details.

[11] The O-notation hides constants that depend on s, a, ζ and v.

[12] When $c = 1$, we can further show that at time $t = O(n^2 c^{-1} \varepsilon^{-2})$, for rollouts initiated from a starting vertex drawn from the training distribution $\mathcal{D}^{\mathcal{Q}}$, the probability of the entire rollout being a chain traversal is at least $1 - \varepsilon$.

Table 1: Test accuracy on *Hard*-n under various post-training distributions, in two real-world post-training experiments on a mathematical reasoning task (Section 6.1). Left: *real-world pre-training* using Qwen2.5-3B-Instruct and $n = 15$. Right: *synthetic pre-training* using a model pre-trained on chain-of-thought traces of simpler instances [73] and $n = 20$. Full details in Appendix I.2.

Real-World Pre-Training		**Synthetic Pre-Training**	
Training Dist.	Test Acc. (*Hard*-15)	Training Dist.	Test Acc. (*Hard*-20)
Hard-15	52.9%	*Hard*-20	43.15%
Mixed-15	96.2%	*Mixed*-20	84.77%
Mixed-10	91.6%	*Mixed*-17	78.68%

where $h(v)$ is the probability of at least one LJ and $\bar{S}^{NL}(v)$ is the same-side absorption probability conditional on no LJ. The factor $1/2$ follows because, after an LJ occurs, the symmetry and structure of the two-chain Markov chain make absorption on either side equally likely. We then use a source decomposition for derivatives of Markov-chain absorption probabilities, expressing each parameter derivative as the expected sum of local one-step sensitivities along the rollout. Applying this representation to the LJ-absorbing and no-LJ chains shows that gradient flow decreases $p_{\mathrm{bwd}}(t)$ and $p_{\mathrm{switch}}(t)$, while increasing $p_{\mathrm{fwd}}(t)$ at a sufficient rate. Choosing $t = O(n^2 c^{-1} \varepsilon^{-2})$ therefore ensures that each rollout step is forward with probability at least $1 - \varepsilon$. □

5.4 Without Simple Examples Learning is Intractable

Theorem 3 shows that if the training distribution has zero $\lceil an \rceil$-simplicity mass (Definition 1), so all examples require at least $\lceil an \rceil$ autoregressive steps, learning is intractable: even up to exponential training time, [13] the Transformer's rollout distribution remains essentially unchanged from initialization. Together with Theorem 2, this implies that simple examples are both sufficient and necessary for learning efficient reasoning.

Theorem 3. *Let $a \in (0, 1)$ and $\varepsilon \in \mathbb{R}_{>0}$. Suppose the $\lceil an \rceil$-simplicity mass of the training data distribution equals zero (Definition 1). For any $t \in \mathbb{R}_{\geq 0}$, denote by $\theta(t)$ the linear Transformer parameters at time t of optimization, i.e., $\theta(t) := (A(t), V)$. Then there exists $\kappa = \kappa(a, \upsilon, \zeta) > 0$, independent of n, such that for every $t < \varepsilon^2 \cdot e^{\kappa n}$, it holds that with any test data distribution (Section 3.1):*

$$\mathrm{TV}\big(p_{roll}(\theta(t)), p_{roll}(\theta(0))\big) < \varepsilon, \tag{6}$$

i.e., at time t of optimization, the distribution over trajectories autoregressively generated by the Transformer (Section 2.3) is ε-close (in total variation distance; Equation (3)) to what it is at initialization.

Proof sketch (full proof in Appendix G). We proceed as in the proof sketch of Theorem 2 by viewing the Transformer's rollouts as a Markov chain and decomposing the same-side absorption probability via the LJ decomposition above. We show, again using the source decomposition, that when the training distribution has $\lceil an \rceil$-simplicity mass zero, these derivatives are exponentially small in n. This implies that the one-step kernels (and hence the induced rollout distributions) stay ε-close in total variation distance up to the stated time scale. □

6 Experiments

In this section, we empirically validate our theoretical predictions in two settings, building on the framework of Zhang et al. [73]. We extend the chain identification task analyzed in our theory (Section 3) to a mathematical reasoning task in which variables can depend on, and affect, multiple other variables. Our experiments confirm two key predictions of our theory: *(i)* simple examples suffice for learning to reason (Section 5.3); and *(ii)* without simple examples, learning fails (Section 5.4).

[13] The bound applies to the time index of gradient flow; by discretizing with a fixed step size, this translates to an exponential number of policy gradient iterations [21].

Synthetic experiments with single-layer Transformers on the specific chain identification task, which more closely match our theoretical setting, are presented in Appendix I.1. [14]

6.1 Setting

We adopt a framework similar to that of Zhang et al. [73]. Each example is presented as a system of arithmetic equations defining a set of variables in terms of others, together with one or more constant assignments and a query for a specified target variable. Unlike the chain identification task analyzed in our theory (Section 3), each variable can depend on, and affect, multiple other variables. The model is post-trained from a base model using a method inspired by DeepSeek-R1 [19], with reward depending only on whether the final answer matches the ground truth (and adherence to a specified output format), with no supervision on intermediate reasoning steps. We characterize the difficulty of an example by a notion of *complexity* corresponding to the extent of reasoning required to compute its answer (defined precisely per setting in Appendix I.2). For some complexity n, we let *Hard-n* denote a training distribution containing only examples of complexity n, and *Mixed-n* denote a training distribution that uniformly mixes examples of complexity $1 \leq i \leq n$.

We conduct two experiments sharing the same task and training procedure, differing primarily in the base model used for RL. We refer to these as the *synthetic pre-training* and *real-world pre-training* settings. In the *synthetic pre-training* setting, following Zhang et al. [73], the base model is a Qwen2.5-style Transformer pre-trained on chain-of-thought traces of our specific mathematical reasoning task, with pre-training restricted to examples simpler than those encountered during RL and evaluation.[15] In the *real-world pre-training* setting, the base model is Qwen2.5-3B-Instruct [56], a standard LLM with broad language pre-training but no prior exposure to our mathematical task.[16] This second setting is more reflective of standard LLM post-training pipelines, where RL is applied directly to a general-purpose base model with no task-specific reasoning priors. Full experimental details are provided in Appendix I.2.

6.2 Results

Table 1 reports results across both settings. Consistent with our theory, including simple examples in the post-training distribution leads to strong generalization on *Hard-n*, even when the training distribution never includes complexity-n examples (*Mixed-k*, $k < n$).[17] On the other hand, excluding simple examples causes generalization to fail entirely.[18] Together, these results support the key conclusion of our theory: a distribution shift toward simpler examples is beneficial for learning to generalize to complex tasks.[19] In terms of our theory, *Hard-n* training corresponds to the regime of Theorem 3, where learning is intractable, whereas *Mixed-n* and *Mixed-k* correspond to the regime of Theorem 2, where the learned algorithm generalizes to unseen complex examples.

To test whether RL induces an implicit bias toward efficient reasoning, we also evaluate the real-world pre-training setting on a restricted version of the task in which the relevant variables form a single dependency chain. A natural efficient strategy for this restricted task is to implement the following procedure: first trace this chain backward, and then compute values forward along it.[20] Analysis via LLM annotation over 100 correct solutions generated by the trained model reveals that 98% of them

[14] Code is available at `https://github.com/shaharmendel/RLProvablyLeadsToReason`.

[15] While our experiments build on the framework of Zhang et al. [73], our setup isolates the role of simple examples: we show that training directly on the target distribution performs worse than training on a distribution shifted towards simpler examples. In contrast, their training distribution is always either harder or easier than their test distribution.

[16] Equivalent experiments with Llama-3.2-3B-Instruct [28] showing the same qualitative results are provided in Appendix I.2.

[17] To isolate the effect of simple examples from increased data diversity, we ran an ablation study in which we held the number of distinct complexity levels, the mass on the hardest level, and the training budget fixed and varied whether the post-training distribution includes simple examples. Including simple examples substantially improves accuracy on hard instances. See Appendix I.2.3 for details.

[18] The failure of *Hard-n* training is not due to insufficient training: the corresponding loss curves show that the training loss plateaus (Appendix I.2).

[19] We note that training on only very simple examples without any complex ones, also fails to generalize to *Hard-n*. Characterizing the minimal post-training distribution sufficient for generalization is an interesting direction for future work.

[20] Calculating values without first identifying the dependency path seems far more difficult, as the model cannot know *a priori* the order in which equations should be processed.

follow precisely this step-by-step algorithm (see Appendix I.2.2 for details), confirming a strong implicit bias toward efficient, structured reasoning.

7 Limitations

It is important to acknowledge several limitations of our work. First, our theoretical analysis focuses on a synthetic chain identification task, and while this task captures a core structural property of reasoning, it is still far simpler than the open-ended reasoning tasks encountered by real-world LLMs. Second, our theory is restricted to single-layer linear Transformers, leaving the extension to deeper, nonlinear architectures theoretically unexplored. Third, the theoretically analyzed training regime assumes continuous-time gradient flow with exact expected gradients, a symmetric initialization with a weak forward preference, output masking, and a fixed value matrix. Fourth, our positive result requires the training distribution to contain sufficient mass on simple examples, and in its current form also imposes an additional simplicity-mass condition (Theorem 2); we believe the latter may be removable. Fifth, our negative result establishes intractability for training distributions with no simple examples within our stylized distribution family, rather than characterizing all possible data distributions or curricula. Sixth, our real-world experiments are limited to a particular mathematical reasoning task, model family, and fine-tuning setup, and therefore should be viewed as evidence for the relevance of our mechanism rather than a comprehensive empirical characterization of outcome-based RL in LLMs.

Our experiments go beyond most of the theoretical limitations above. Indeed, the appendix experiments in Appendix I.1 demonstrate that the main phenomena persist when both attention and value matrices are trained jointly with sample-based policy gradient from a small Gaussian initialization, without the simplifying assumptions of Section 5.2. Appendix I.2 further demonstrates analogous behavior in a modern LLM fine-tuned on a natural-language mathematical reasoning task. Across these settings, Transformers trained with sufficiently simple examples learn efficient step-by-step reasoning algorithms, generalize to more complex examples, and fail to do so when simple examples are absent. Extending our theoretical analysis to match the generality of these empirical demonstrations is an important direction for future work.

8 Conclusion

In this paper, we established a theoretical footing for the emergence of CoT (Chain-of-Thought) reasoning in Transformers trained via outcome-based RL (Reinforcement Learning), and we corroborated our theory with experiments in synthetic and real-world settings. Our findings imply that, with the right data, policy gradient exhibits an implicit bias towards reasoning algorithms that are both efficient and capable of out-of-distribution generalization. Specifically, when the training data includes a sufficient amount of simple examples, policy gradient bypasses inefficient solutions, and converges to an efficient reasoning algorithm that generalizes to examples of arbitrary complexity. The condition on simple examples in the training data is necessary: without such examples, learning becomes intractable.

From a practical perspective, our findings suggest a counterintuitive message: when fine-tuning an LLM (Large Language Model), rather than training on the target distribution, it may be beneficial to deliberately introduce a distribution shift by including simple examples in training. Prior work has identified settings where introducing a distribution shift in training benefits learning (see Appendix A.3). However, to our knowledge, this paper is the first to establish said phenomenon for Transformers trained via outcome-based RL. Moreover, it identifies a specific type of examples whose inclusion in training can lead to efficient reasoning and out-of-distribution generalization. We hope that our work will serve as a foundation for developing practical post-training curricula for LLMs, thereby bolstering the efficiency and generalizability of their reasoning capabilities.

Acknowledgments and Disclosure of Funding

We thank Joey Rudoler, Noam Razin, and Yonatan Slutzky for illuminating discussions. This work was supported by the European Research Council (ERC) grant and NN4C 101164614, a Google Research Scholar Award, a Google Research Gift, Meta, the Yandex Initiative in Machine Learning,

the Israel Science Foundation (ISF) grant 1780/21, the Tel Aviv University Center for AI and Data Science, the Adelis Research Fund for Artificial Intelligence, Len Blavatnik and the Blavatnik Family Foundation, and Amnon and Anat Shashua.

References

[1] Emmanuel Abbe, Samy Bengio, Aryo Lotfi, and Kevin Rizk. Generalization on the unseen, logic reasoning and degree curriculum, 2024. URL https://arxiv.org/abs/2301.13105.

[2] Palaash Agrawal, Shavak Vasania, and Cheston Tan. Can llms perform structured graph reasoning?, 2024. URL https://arxiv.org/abs/2402.01805.

[3] Miklós Ajtai. σ_1^1-formulae on finite structures. In *Annals of Pure and Applied Logic*, volume 24, pages 1–48, 1983.

[4] Yotam Alexander, Nimrod De La Vega, Noam Razin, and Nadav Cohen. What makes data suitable for a locally connected neural network? a necessary and sufficient condition based on quantum entanglement, 2024. URL https://arxiv.org/abs/2303.11249.

[5] Sanjeev Arora, Hrishikesh Khandeparkar, Mikhail Khodak, Orestis Plevrakis, and Nikunj Saunshi. A theoretical analysis of contrastive unsupervised representation learning, 2019. URL https://arxiv.org/abs/1902.09229.

[6] Gregor Bachmann and Vaishnavh Nagarajan. The pitfalls of next-token prediction, 2024. URL https://arxiv.org/abs/2403.06963.

[7] Yu Bai, Fan Chen, Huan Wang, Caiming Xiong, and Song Mei. Transformers as statisticians: Provable in-context learning with in-context algorithm selection, 2023. URL https://arxiv.org/abs/2306.04637.

[8] David A Mix Barrington, Neil Immerman, and Howard Straubing. Bounded-width polynomial-size branching programs recognize exactly those languages in nc^1. *Journal of Computer and System Sciences*, 38(1):150–164, 1989.

[9] Laurent Bartholdi, Michael Figelius, Markus Lohrey, and Armin Weiß. Groups with alogtime-hard word problems and pspace-complete compressed word problems, 2020. URL https://arxiv.org/abs/1909.13781.

[10] Tina Behnia, Puneesh Deora, and Christos Thrampoulidis. Facts in stats: Impacts of pretraining diversity on language model generalization, 2025. URL https://arxiv.org/abs/2510.16096.

[11] Mikhail Belkin, Daniel Hsu, Siyuan Ma, and Soumik Mandal. Reconciling modern machine-learning practice and the classical bias–variance trade-off, July 2019. ISSN 1091-6490. URL http://dx.doi.org/10.1073/pnas.1903070116.

[12] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, page 41–48, New York, NY, USA, 2009. Association for Computing Machinery. ISBN 9781605585161. doi: 10.1145/1553374.1553380. URL https://doi.org/10.1145/1553374.1553380.

[13] Allan Borodin. On relating time and space to size and depth. *SIAM Journal on Computing*, 6(4):733–744, December 1977. doi: 10.1137/0206054.

[14] Dake Bu, Wei Huang, Andi Han, Atsushi Nitanda, Hau-San Wong, Qingfu Zhang, and Taiji Suzuki. Provable benefit of curriculum in transformer tree-reasoning post-training, 2025. URL https://arxiv.org/abs/2511.07372.

[15] Bo Chen, Xiaoyu Li, Yingyu Liang, Zhenmei Shi, and Zhao Song. Bypassing the exponential dependency: Looped transformers efficiently learn in-context by multi-step gradient descent, 2025. URL https://arxiv.org/abs/2410.11268.

[16] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir

Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating large language models trained on code, 2021. URL https://arxiv.org/abs/2107.03374.

[17] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations, 2020. URL https://arxiv.org/abs/2002.05709.

[18] Stephen A Cook. A taxonomy of problems with fast parallel algorithms. *Information and Control*, 64(1-3): 2–22, 1985.

[19] DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Shengfeng Ye, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanjia Zhao, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanhong Xu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025. URL https://arxiv.org/abs/2501.12948.

[20] Shihan Dou, Yan Liu, Haoxiang Jia, Limao Xiong, Enyu Zhou, Wei Shen, Junjie Shan, Caishuang Huang, Xiao Wang, Xiaoran Fan, Zhiheng Xi, Yuhao Zhou, Tao Ji, Rui Zheng, Qi Zhang, Xuanjing Huang, and Tao Gui. Stepcoder: Improve code generation with reinforcement learning from compiler feedback, 2024. URL https://arxiv.org/abs/2402.01391.

[21] Omer Elkabetz and Nadav Cohen. Continuous vs. discrete optimization of deep neural networks, 2021. URL https://arxiv.org/abs/2107.06608.

[22] K. Etessami. Counting quantifiers, successor relations, and logarithmic space. In *Proceedings of the 10th Annual Structure in Complexity Theory Conference (SCT'95)*, SCT '95, page 2, USA, 1995. IEEE Computer Society. ISBN 0818670525.

[23] William Fedus, Prajit Ramachandran, Rishabh Agarwal, Yoshua Bengio, Hugo Larochelle, Mark Rowland, and Will Dabney. Revisiting fundamentals of experience replay, 2020. URL https://arxiv.org/abs/2007.06700.

[24] Vitaly Feldman. Does learning require memorization? a short tale about a long tail, 2021. URL https://arxiv.org/abs/1906.05271.

[25] Guhao Feng, Bohang Zhang, Yuntian Gu, Haotian Ye, Di He, and Liwei Wang. Towards revealing the mystery behind chain of thought: A theoretical perspective, 2023. URL https://arxiv.org/abs/2305.15408.

[26] Merrick Furst, James B Saxe, and Michael Sipser. Parity, circuits, and the polynomial-time hierarchy. *Mathematical Systems Theory*, 17(1):13–27, 1984.

[27] Angeliki Giannou, Shashank Rajput, Jy yong Sohn, Kangwook Lee, Jason D. Lee, and Dimitris Papailiopoulos. Looped transformers as programmable computers, 2023. URL https://arxiv.org/abs/2301.13196.

[28] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vítor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso,

Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

[29] Alex Graves, Marc G. Bellemare, Jacob Menick, Remi Munos, and Koray Kavukcuoglu. Automated curriculum learning for neural networks, 2017. URL https://arxiv.org/abs/1704.03003.

[30] Tianyu Guo, Wei Hu, Song Mei, Huan Wang, Caiming Xiong, Silvio Savarese, and Yu Bai. How do transformers learn in-context beyond simple functions? a case study on learning with representations, 2023. URL https://arxiv.org/abs/2310.10616.

[31] Peter Hase, Mohit Bansal, Peter Clark, and Sarah Wiegreffe. The unreasonable effectiveness of easy training data for hard tasks. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL)*, 2024. URL https://arxiv.org/abs/2401.06751.

[32] Jianhao Huang, Zixuan Wang, and Jason D. Lee. Transformers learn to implement multi-step gradient descent with chain of thought, 2025. URL https://arxiv.org/abs/2502.21212.

[33] Yu Huang, Zixin Wen, Aarti Singh, Yuejie Chi, and Yuxin Chen. Transformers provably learn chain-of-thought reasoning with length generalization, 2025. URL https://arxiv.org/abs/2511.07378.

[34] Yu Huang, Zixin Wen, Yuejie Chi, Yuting Wei, Aarti Singh, Yingbin Liang, and Yuxin Chen. On the learning dynamics of rlvr at the edge of competence, 2026. URL https://arxiv.org/abs/2602.14872.

[35] Nirmit Joshi, Gal Vardi, Adam Block, Surbhi Goel, Zhiyuan Li, Theodor Misiakiewicz, and Nathan Srebro. A theory of learning with autoregressive chain of thought, 2025. URL https://arxiv.org/abs/2503.07932.

[36] Juno Kim and Taiji Suzuki. Transformers provably solve parity efficiently with chain of thought, 2025. URL https://arxiv.org/abs/2410.08633.

[37] Sunghwan Kim, Junhee Cho, Beong woo Kwak, Taeyoon Kwon, Liang Wang, Nan Yang, Xingxing Zhang, Furu Wei, and Jinyoung Yeo. On training large language models for long-horizon tasks: An empirical study of horizon length. In *International Conference on Machine Learning (ICML)*, 2026. URL https://arxiv.org/abs/2605.02572.

[38] Oleg Kovalevskiy, Juan Mateos-Garcia, and Kathryn Tunyasuvunakool. Alphafold two years on: validation and impact, 2024. URL https://arxiv.org/abs/2403.02124.

[39] Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning, 2020. URL https://arxiv.org/abs/2006.04779.

[40] Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems, 2020. URL https://arxiv.org/abs/2005.01643.

[41] Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, Yuhuai Wu, Behnam Neyshabur, Guy Gur-Ari, and Vedant Misra. Solving quantitative reasoning problems with language models, 2022. URL https://arxiv.org/abs/2206.14858.

[42] Zhiyuan Li, Hong Liu, Denny Zhou, and Tengyu Ma. Chain of thought empowers transformers to solve inherently serial problems, 2024. URL https://arxiv.org/abs/2402.12875.

[43] Licong Lin, Yu Bai, and Song Mei. Transformers as decision makers: Provable in-context reinforcement learning via supervised pretraining, 2024. URL https://arxiv.org/abs/2310.08566.

[44] Bochen Lyu, Yiyang Jia, Xiaohao Cai, and Zhanxing Zhu. Transformers with rl or sft provably learn sparse boolean functions, but differently, 2025. URL https://arxiv.org/abs/2511.17852.

[45] Marko Medvedev, Kaifeng Lyu, Zhiyuan Li, and Nathan Srebro. Shift is good: Mismatched data mixing improves test performance, 2025. URL https://arxiv.org/abs/2510.25108.

[46] William Merrill and Ashish Sabharwal. The parallelism tradeoff: Limitations of log-precision transformers, 2023. URL https://arxiv.org/abs/2207.00729.

[47] William Merrill and Ashish Sabharwal. The expressive power of transformers with chain of thought, 2024. URL https://arxiv.org/abs/2310.07923.

[48] Parsa Mirtaheri, Ezra Edelman, Samy Jelassi, Eran Malach, and Enric Boix-Adsera. Let me think! a long chain-of-thought can be worth exponentially many short ones, 2025. URL https://arxiv.org/abs/2505.21825.

[49] Sumeet Ramesh Motwani, Alesia Ivanova, Ziyang Cai, Philip Torr, Riashat Islam, Shital Shah, Christian Schroeder de Witt, and Charles London. h1: Bootstrapping LLMs to reason over longer horizons via reinforcement learning, 2025. URL https://arxiv.org/abs/2510.07312.

[50] Preetum Nakkiran, Gal Kaplun, Yamini Bansal, Tristan Yang, Boaz Barak, and Ilya Sutskever. Deep double descent: Where bigger models and more data hurt, 2019. URL https://arxiv.org/abs/1912.02292.

[51] Sanmit Narvekar, Bei Peng, Matteo Leonetti, Jivko Sinapov, Matthew E. Taylor, and Peter Stone. Curriculum learning for reinforcement learning domains: A framework and survey, 2020. URL https://arxiv.org/abs/2003.04960.

[52] Eshaan Nichani, Alex Damian, and Jason D. Lee. How transformers learn causal structure with gradient descent, 2024. URL https://arxiv.org/abs/2402.14735.

[53] Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback, 2022. URL https://arxiv.org/abs/2203.02155.

[54] Shubham Parashar, Shurui Gui, Xiner Li, Hongyi Ling, Sushil Vemuri, Blake Olson, Eric Li, Yu Zhang, James Caverlee, Dileep Kalathil, and Shuiwang Ji. Curriculum reinforcement learning from easy to hard tasks improves llm reasoning, 2025. URL https://arxiv.org/abs/2506.06632.

[55] Rémy Portelas, Cédric Colas, Lilian Weng, Katja Hofmann, and Pierre-Yves Oudeyer. Automatic curriculum learning for deep rl: A short survey, 2020. URL https://arxiv.org/abs/2003.04664.

[56] Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025. URL https://arxiv.org/abs/2412.15115.

[57] Leonardo Ranaldi, Giulia Pucci, and Fabio Massimo Zanzotto. Modeling easiness for training transformers with curriculum learning. In Ruslan Mitkov and Galia Angelova, editors, *Proceedings of the 14th International Conference on Recent Advances in Natural Language Processing*, pages 937–948, Varna, Bulgaria, September 2023. INCOMA Ltd., Shoumen, Bulgaria. URL https://aclanthology.org/2023.ranlp-1.101/.

[58] Clayton Sanford, Bahare Fatemi, Ethan Hall, Anton Tsitsulin, Mehran Kazemi, Jonathan Halcrow, Bryan Perozzi, and Vahab Mirrokni. Understanding transformer reasoning capabilities via graph algorithms, 2024. URL https://arxiv.org/abs/2405.18512.

[59] Clayton Sanford, Daniel Hsu, and Matus Telgarsky. Transformers, parallel computation, and logarithmic depth, 2024. URL https://arxiv.org/abs/2402.09268.

[60] Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. Prioritized experience replay, 2016. URL https://arxiv.org/abs/1511.05952.

[61] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017. URL https://arxiv.org/abs/1707.06347.

[62] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024. URL https://arxiv.org/abs/2402.03300.

[63] Petru Soviany, Radu Tudor Ionescu, Paolo Rota, and Nicu Sebe. Curriculum learning: A survey, 2022. URL https://arxiv.org/abs/2101.10382.

[64] Alex F. Spies, William Edwards, Michael I. Ivanitskiy, Adrians Skapars, Tilman Räuker, Katsumi Inoue, Alessandra Russo, and Murray Shanahan. Transformers use causal world models in maze-solving tasks, 2025. URL https://arxiv.org/abs/2412.11867.

[65] Zhiqing Sun, Longhui Yu, Yikang Shen, Weiyang Liu, Yiming Yang, Sean Welleck, and Chuang Gan. Easy-to-hard generalization: Scalable alignment beyond human supervision. In *Advances in Neural Information Processing Systems*, 2024. URL https://arxiv.org/abs/2403.09472.

[66] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017. URL https://arxiv.org/abs/1706.03762.

[67] Heribert Vollmer. *Introduction to Circuit Complexity: A Uniform Approach*. Springer Science & Business Media, 1999.

[68] Johannes von Oswald, Eyvind Niklasson, Ettore Randazzo, João Sacramento, Alexander Mordvintsev, Andrey Zhmoginov, and Max Vladymyrov. Transformers learn in-context by gradient descent, 2023. URL https://arxiv.org/abs/2212.07677.

[69] Jiuqi Wang, Ethan Blaser, Hadi Daneshmand, and Shangtong Zhang. Transformers can learn temporal difference methods for in-context reinforcement learning, 2025. URL https://arxiv.org/abs/2405.13861.

[70] Zixuan Wang, Eshaan Nichani, Alberto Bietti, Alex Damian, Daniel Hsu, Jason D. Lee, and Denny Wu. Learning compositional functions with transformers from easy-to-hard data, 2025. URL https://arxiv.org/abs/2505.23683.

[71] Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8:229–256, 1992. URL https://api.semanticscholar.org/CorpusID:2332513.

[72] Tong Yang, Yu Huang, Yingbin Liang, and Yuejie Chi. Multi-head transformers provably learn symbolic multi-step reasoning via gradient descent, 2025. URL https://arxiv.org/abs/2508.08222.

[73] Charlie Zhang, Graham Neubig, and Xiang Yue. On the interplay of pre-training, mid-training, and rl on reasoning language models, 2025. URL https://arxiv.org/abs/2512.07783.

[74] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization, 2017. URL https://arxiv.org/abs/1611.03530.

[75] Ruiqi Zhang, Spencer Frei, and Peter L. Bartlett. Trained transformers learn linear models in-context, 2023. URL https://arxiv.org/abs/2306.09927.

[76] Yifan Zhang, Wenyu Du, Dongming Jin, Jie Fu, and Zhi Jin. Finite state automata inside transformers with chain-of-thought: A mechanistic study on state tracking, 2025. URL https://arxiv.org/abs/2502.20129.

A Related Work

A.1 Problems which provably require CoT

A growing theoretical literature seeks to explain how CoT boosts Transformer performance by enhancing its expressivity. Several works show that CoT allows constant-depth Transformers to solve tasks such as parity, arithmetic, and regular languages that are inaccessible to single-step decoding [25, 47, 42]. We contribute to this literature by establishing a natural graph traversal problem that provably requires CoT under standard complexity-theoretic assumptions.

A.2 Training Dynamics of Transformers

A parallel line of work analyzes how gradient-based training shapes Transformer behavior. Several studies track the emergence of algorithmic heads, programmable computation, or compositional rules [36, 27, 76, 69, 70], while other results show that gradient descent leads Transformers to implement in-context learning algorithms [30, 7, 68, 75, 43, 52]. Complementary work analyzes when CoT trajectory supervision improves the sample and computational complexity of learning [35].

Recent analyses further show that gradient-based training can drive Transformers that employ CoT to implement explicit multi-step algorithms [72, 32, 15, 33]. These works differ from ours in that they rely on training with next-token prediction and do not characterize when CoT reasoning emerges under RL training.

Two results that analyze RL-trained Transformers are Lyu et al. [44] and Bu et al. [14], but both differ from our setting in crucial ways. The first [44] assumes a setting with dense rewards at every step along the CoT, relying on a much richer feedback signal than the sparse, outcome-based rewards we study. The second [14], although formally using an outcome-based reward, defines a curriculum over progressively longer prefixes, so the learner effectively receives feedback for partial progress; in contrast, our setting provides feedback only on the full final outcome. Furthermore, Bu et al. [14] analyze a simplified regime with only a single gradient update per curriculum stage, while we consider full training dynamics.

Follow-up work by Huang et al. [34], which appeared after our submission and cites our work, studies policy gradient dynamics in a related compositional setting. They provide analysis of policy gradient dynamics under less restrictive assumptions on the distribution of task difficulties.However, their setting can be viewed as a "single-chain" problem in which each step applies a group transformation, and the model is assumed to have access to a fixed MLP that already perfectly implements the atomic operation. Thus, the learning problem is to traverse the chain correctly rather than to acquire the stepwise transformation itself. By contrast, we do not assume such a fixed module. Moreover, our problem admits multiple minimizers corresponding to different algorithms; in this regime, we show that policy gradient exhibits a data-dependent implicit bias toward the most efficient solution.

A.3 Effect of Training Data Distribution on Learning

The composition and ordering of training data can significantly impact learning in neural networks. Empirical and theoretical work across supervised learning, unsupervised learning, and reinforcement learning demonstrates that training data distribution affects generalization, sample efficiency, and the emergence of specific learned behaviors [60, 74, 11, 50, 5, 17, 39, 23, 40, 24, 4]. Work on curriculum learning further suggests that ordering examples by complexity—determined by factors such as length, rarity, or comprehensibility—can significantly aid the learning process [12, 29, 51, 55, 63, 1, 14]. More specifically, both empirical and theoretical investigations into Transformers emphasize that training data diversity is essential for the effective learning of complex compositional tasks [57, 70, 10]. In a complementary line of work, Medvedev et al. [45] demonstrate that out-of-distribution data can improve in-distribution performance in general mixture settings.

Particularly relevant to our setting, the empirical study of Zhang et al. [73] finds that RL post-training improves reasoning only when training examples are not significantly harder than those seen during pretraining (what they call the model's edge of competence). This supports our finding that simple examples are necessary for learning, and our experiments in Section 6 build on this perspective in both synthetic and real-world settings.

B Proof of Proposition 1

Proposition 1 in the main text is stated for infinite precision regime, which is incompatible with the finite log-precision regime, in which Theorem 1 holds. In this section we prove a generalization of Proposition 1 that applies to the finite log-precision regime, in which Theorem 1 holds. Proposition 1 (as stated in Section 4.2) is an immediate corollary of the result in this section: taking the precision parameter $\xi \to \infty$ yields reward arbitrarily close to 1. Moreover, we will show that the reward achieved by the construction in this section is strictly larger than the bound obtained in Theorem 1.

First, we formalize what it means for a Transformer to have *b-bit precision*:

Definition 2. We say that a Transformer θ has *b-bit precision* if each scalar entry of the parameter matrices (A, V) (or $\{(A_i, V_i)\}_{i=1}^{d}$ in the multi-layer case) can take at values in $1, 2, \ldots, 2^b$

Definition 2 can be extended to other numerical representations, including signed integers and fixed-point or floating-point formats, as long as the maximum representable magnitude remains $O(2^b)$.

We work in the *logarithmic precision* regime: the number of bits scales logarithmically with the problem size n. Concretely, we set

$$b \;=\; \lceil \xi \, \log_2 n \rceil$$

for a positive integer $\xi \in \mathbb{N}_{>0}$ independent of n. (This is the same precision assumption as in Theorem 1 where ξ can be set to any positive integer.)

Now we are ready to state the theorem:

Theorem 4. *Fix any $n \geq 3$, any positive integer $\xi \geq 1$, and any distribution $\mathcal{Q}$ over $[n-1]$. There exists a single-layer Transformer $\theta = (A, V)$ with $\xi\lceil \log_2 n \rceil$-bit precision such that*

$$1 - \mathcal{R}(\mathcal{D}^{\mathcal{Q}}, \theta) \;\leq\; n\big(|\Sigma| - 1\big) \exp\big(-\tfrac{1}{2} n^{\xi}\big),$$

where $|\Sigma| = 2n + (2n)^2$ (see Section 3). Moreover, for every input sequence $s_{1:L_0}$ from $\mathcal{D}^{\mathcal{Q}}$, with probability at least $1 - n(|\Sigma| - 1)\exp(-\frac{1}{2}n^{\xi})$ over $\tau \sim p_{roll}(\theta)$, the rollout τ is a chain traversal (Section 3.3).

Proof. We prove the result simultaneously for the softmax and linear Transformer variants using the unified weight function $w(\cdot)$ from Lemma 68.

Let A satisfy the assumptions of Lemma 68 with

$$\alpha = n, \qquad \beta = 0, \qquad \gamma = 0,$$

and set all vertex–vertex entries of A equal to $\gamma = 0$. Let V satisfy the assumptions of Lemma 68 with

$$\alpha' = 0, \qquad \gamma' = 0, \qquad \beta' = n^{\xi}.$$

All nonzero entries of (A, V) are integers of magnitude at most $n^{\xi} \leq 2^b - 1$, hence representable with b bits.

Fix any time during autoregressive generation before termination, and let v be the last generated vertex (so the current input sequence s ends with v). Let u denote the (unique) successor of v in its chain if it exists. By Lemma 68 and our choice $\Delta_\alpha = \alpha' - \gamma' = 0$ and $\Delta_\beta = \beta' - \gamma' = \beta'$, the logit differences are controlled by the incoming-edge weights. For every nonterminal $v \in (\mathcal{C}_a \cup \mathcal{C}_b) \setminus \{a_n, b_n\}$, the successor u has an incoming edge $(v, u) \in G$ whose attention score from v is $A_{v,(v,u)} = \alpha = n$; every other vertex $k \neq u$ has either an incoming edge with attention score 0 from v (i.e., $A_{v,(\mathrm{pred}(k),k)} \in \{\beta, \gamma\} = \{0\}$) or has no incoming edge in G (start vertices). For edge token outputs $e' \in \mathcal{E}$, by Lemma 68 we have $o(v, G, Z_v)_{e'} = 0$. Hence there is a logit gap of the form

$$o(v, G, Z_v)_u - o(v, G, Z_v)_k \;\geq\; \beta' \big(w(n) - w(0)\big) \qquad \text{for all } k \neq u.$$

Since $w(\cdot)$ is monotone increasing, $w(n) > w(0)$ in both attention variants: for linear attention, $w(n) - w(0) = n$; for softmax attention, $w(n) - w(0) = \frac{e^n - 1}{Z_v}$ (where Z_v is defined in Lemma 68). Under our choice of A, all tokens other than the unique outgoing edge (v, u) have attention score 0

from v (including all vertex tokens, since we set the vertex–vertex block to $\gamma = 0$). Therefore, for softmax attention,

$$Z_v = \sum_{j=1}^{L} \exp(A_{v,s_j}) \le e^n + (L-1)\cdot e^0 = e^n + L - 1,$$

where L is the current prefix length. Condition on the "good" event E that the rollout has followed the correct chain successor at every previous step (equivalently, it is a chain traversal up to the current time). On E, the number of generated vertices before termination is at most n (the chain length), hence $L \le L_0 + n \le (2n-1) + n = 3n - 1$. Thus, on E,

$$w(n) - w(0) \;\ge\; \begin{cases} n, & \text{linear attention,} \\ \dfrac{e^n - 1}{3n - 2 + e^n}, & \text{softmax attention.} \end{cases}$$

Since $e^n \ge 3n$ for all $n \ge 2$, we have

$$\frac{e^n - 1}{3n - 2 + e^n} \;\ge\; \frac{1}{2},$$

and hence for both attention variants $w(n) - w(0) \ge \frac{1}{2}$ on E.

The next-token distribution is $\mathrm{TF}(s_{1:L};\theta) = \mathrm{softmax}\big(o(v, G, Z_v)\big)$. Using the logit gap and bounding the denominator by $|\Sigma|$ terms (including both vertices and edges), we obtain for every nonterminal vertex $v \in (\mathcal{C}_a \cup \mathcal{C}_b) \setminus \{a_n, b_n\}$:

$$\Pr\left[s_{L+1} = u \,\middle|\, s_L = v\right] \ge \frac{1}{1 + \sum_{k \ne u} \exp\big(-\frac{1}{2}\beta'\big)} \ge 1 - (|\Sigma| - 1)\exp\big(-\tfrac{1}{2}\beta'\big).$$

All rollouts start from a vertex in one of the chains; at most n nonterminal-to-successor transitions are required to reach the chain terminal. By a union bound, the probability that the rollout ever deviates from the correct successor is at most

$$n(|\Sigma| - 1)\exp\big(-\tfrac{1}{2}\beta'\big).$$

Whenever the rollout follows the chain edges until it reaches a terminal vertex, it terminates at the correct terminal $y^*(s_{1:L_0})$ and receives reward 1 (see Sections 3.2 and 3.3). Therefore,

$$1 - \mathcal{R}(\mathcal{D}^{\mathcal{Q}}, \theta) \;\le\; n(|\Sigma| - 1)\exp\big(-\tfrac{1}{2}\beta'\big) \;=\; n(|\Sigma| - 1)\exp\big(-\tfrac{1}{2}n^{\xi}\big),$$

as claimed.

□

Corollary 1. *For any $\xi > 1$ there exists a chain length n such that the reward achieved by the construction in Theorem 4 is strictly greater than the maximal single-step reward achievable by any Transformer under the setting of Theorem 1.*

Proof. By Theorem 4, the construction achieves reward at least $1 - n(|\Sigma| - 1)\exp(-\frac{1}{2}n^{\xi})$. By Lemma 1 and the analysis in Appendix E, there exists a chain length n such that any Transformer achieves single-step reward at most $1 - \frac{1}{(2n)!\cdot 2(n-1)}$. It suffices to show that for sufficiently large n:

$$n(|\Sigma| - 1)\exp\big(-\tfrac{1}{2}n^{\xi}\big) < \frac{1}{(2n)!\cdot 2(n-1)}.$$

Rearranging, this is equivalent to:

$$n(|\Sigma| - 1)\cdot(2n)!\cdot 2(n-1) < \exp\big(\tfrac{1}{2}n^{\xi}\big).$$

Taking logarithms, it suffices to show:

$$\log n + \log(|\Sigma| - 1) + \log((2n)!) + \log 2 + \log(n-1) < \tfrac{1}{2}n^{\xi}.$$

Since $|\Sigma| = 2n + (2n)^2$, we have $\log(|\Sigma| - 1) = O(\log n)$. By Stirling's approximation, $\log((2n)!) = 2n\log(2n) - 2n + O(\log n) = O(n\log n)$. Thus the left-hand side is $O(n\log n)$. Since $\xi > 1$, we have $n^{\xi}/(n\log n) \to \infty$ as $n \to \infty$, so $\frac{1}{2}n^{\xi} > O(n\log n)$ for sufficiently large n. □

C Proof of Proposition 2

Proof. Fix $n \geq 3$, a distribution Q over $[n-1]$, a target length threshold $k \in \mathbb{N}$, and a loss tolerance $\varepsilon > 0$. We'll assume w.l.o.g. that $\delta > \varepsilon$ (otherwise we can lower ε).

We use the notation and symmetry setup of Lemma 68, in particular the parameters α, β, γ in the attention matrix and α', β', γ' in the value matrix. We also reuse the notation $\Delta_\alpha := \alpha' - \gamma'$ and $\Delta_\beta := \beta' - \gamma'$. The formulas for the per-vertex logit contributions Ψ_k as a function of $(\alpha, \beta, \gamma, \Delta_\alpha, \Delta_\beta)$ are exactly those in Lemma 68. As the transformer is assumed to be linear, we have $w(\mu) = \mu$.

We construct a 1-layer transformer with attention parameters

$$\gamma = 0, \qquad \beta = -\alpha_0 - \beta_0, \qquad \alpha = \alpha_0.$$

for some free parameters $\alpha_0, \beta_0 \in \mathbb{R}$ to be chosen later, and with value parameters

$$\gamma' = 0, \qquad \alpha' = 1, \qquad \beta' = -1.$$

Thus $\Delta_\alpha = 1$ and $\Delta_\beta = -1$. We keep the vertex–vertex block of A equal to $\gamma = 0$, and set $V_{v,:} = 0$ for all vertex tokens $v \in V$, exactly as in Lemma 68.

Let v be any non-terminal vertex in one of the chains, and consider the logits at the last position when the current prefix ends with v. Applying Lemma 68 and substituting $\gamma = 0$, $w(\gamma) = 0$, $\Delta_\alpha = 1$, $\Delta_\beta = -1$, we obtain for an interior vertex (Case 1 in Lemma 68) that the only nonzero logit contributions on the two incident vertices are

$$\Psi_{\mathrm{forw}} = \Delta_\beta w(\alpha) = \beta'\,\alpha_0 = -\alpha_0, \qquad \Psi_{\mathrm{back}} = \Delta_\alpha w(\beta) = \alpha'\,\beta = \beta = -\alpha_0 - \beta_0.$$

The current vertex v itself has

$$\Psi_v = \Delta_\alpha w(\alpha) + \Delta_\beta w(\beta) = \alpha_0 - \beta = 2\alpha_0 + \beta_0,$$

and every other vertex has $\Psi_k = 0$ (since all those terms are proportional to $w(\gamma) = 0$ in Lemma 68), and all edge token outputs also have $\Psi_{e'} = 0$ by Lemma 68. Thus, up to some offset C, any vertex v which is "in the middle of the chain" (not the start, end, second or penultimate vertex of some chain) has a local logit pattern of the form

$$\ell_v = 2\alpha_0 + \beta_0, \quad \ell_{\mathrm{back}} = -\alpha_0 - \beta_0, \quad \ell_{\mathrm{forw}} = -\alpha_0, \quad \ell_{\mathrm{other}} = 0.$$

The same calculation applied to vertices v which are not the start of a chain (Cases 2,3,5 in Lemma 68) gives the same qualitative picture: for any non-terminal v on either chain the logits on the two incident chain vertices are linear functions of α_0 and β_0 of the form

$$\ell_{\mathrm{forw}}(\beta_0, \alpha_0) = \beta'\,\alpha_0 = -\alpha_0, \qquad \ell_{\mathrm{back}}(\beta_0, \alpha_0) = \alpha'\,\beta = -\alpha_0 - \beta_0,$$

while all other vertices have logit 0, and the current vertex v has logit $2\alpha_0 + \beta_0$. In particular the *only* dependence on β_0 in the *differences* between the forward and backward logits is

$$\ell_{\mathrm{forw}} - \ell_{\mathrm{back}} = \beta_0.$$

Finally, when the current vertex v is the start of a chain (Case 4 in Lemma 68), there is no backward neighbor within the chain. In this case, the forward logit is still $\ell_{\mathrm{forw}} = -\alpha_0$, but there is no ℓ_{back}, so as $\alpha_0 \to -\infty$, the forward probability to move forward limits to one.

We now use this to tune the random-walk behaviour in two stages.

1. Controlling the conditional walk along the chain (choosing β_0). Fix α_0 for the moment. Condition on the "good event" that at each step the model only chooses the forward or backward neighbour in the chain (i.e., we ignore the possibility of staying at v or jumping to any other vertex). Under this conditioning, from every interior vertex v the next-token distribution over $\{\mathrm{forw}, \mathrm{back}\}$ has the form

$$\mathbb{P}(\mathrm{forw} \mid \{\mathrm{forw}, \mathrm{back}\}) = \frac{e^{\ell_{\mathrm{forw}}}}{e^{\ell_{\mathrm{forw}}} + e^{\ell_{\mathrm{back}}}} = \frac{1}{1 + e^{-(\ell_{\mathrm{forw}} - \ell_{\mathrm{back}})}} = \frac{1}{1 + e^{-\beta_0}},$$
$$\mathbb{P}(\mathrm{back} \mid \{\mathrm{forw}, \mathrm{back}\}) = \frac{1}{1 + e^{\beta_0}}.$$

Thus the forward/backward bias of the induced nearest-neighbour walk along the chain depends *only* on β_0, via a smooth, strictly monotone function. As $\beta_0 \to 0$ the walk becomes approximately unbiased ($\mathbb{P}(\text{forw}) \to 1/2$); as $\beta_0 \to \infty$ the walk becomes almost deterministic in the forward direction; as $\beta_0 \to -\infty$ it becomes almost deterministic in the backward direction.

Let us denote by G the event that the rollout consists only of forward/backward moves along the chain. Standard random-walk estimates on a finite chain imply that, for any fixed chain length n, and starting index $k \in [n-1]$, there exists a choice of β_0 and some $L_{\max}$ (dependent on β_0) such that, *conditional on* G, the hitting time T of the terminal vertex satisfies

$$\mathbb{P}\big(L_{\max} > T > L \;\big|\; G\big) \geq 1 - \varepsilon/2$$

for any prescribed L. This implies that conditioned on G, the rollout will take more than L and less than $L_{\max}$ steps to reach the terminal vertex with probability at least $1 - \varepsilon/2$ (which assures loss smaller than $1 - \varepsilon/2$).

2. Making $\mathbb{P}(G^c) \leq \varepsilon/2$ (choosing α_0). We now choose α_0 to ensure that the model stays on the chain. Taking $\alpha_0 \to -\infty$, the logits for the forward and backward neighbors ($\ell_{\text{forw}} = -\alpha_0$, $\ell_{\text{back}} = -\alpha_0 - \beta_0$) tend to $+\infty$, while the logit for the current vertex ($\ell_v = 2\alpha_0 + \beta_0$) tends to $-\infty$, and other vertices as well as all edge token outputs remain at 0. Consequently, the probability mass concentrates on the forward and backward neighbors, with the ratio $p_{\text{forw}}/p_{\text{back}} = e^{\beta_0}$ preserved. The probability of outputting any other token (vertex or edge) vanishes relative to the move probabilities.

Together, these two points imply the theorem.

□

D Restatement of Propositions 1 and 2 Under the Setting of Section 5.2

The constructions establishing Propositions 1 and 2 do not adhere to the value-matrix parameterization of Section 5.2, which fixes $V_{(u,w),v} = \mathbb{1}\{v \in \{u, w\}\}$. Using this parameterization directly with variants of those constructions induces a high logit for the current vertex, preventing convergence to the desired behavior. However, under the assumption of Section 5.2 that the model only outputs vertices (edges are masked from the output) and never outputs the same vertex twice consecutively (the previous vertex is masked), we can construct single-layer linear Transformers achieving the same guarantees and adhering to the setting of Section 5.2. Appendix D.1 restates Proposition 1 and Appendix D.2 restates Proposition 2.

D.1 Restatement of Proposition 1 Under the Setting of Section 5.2

Proposition 3. *Consider the setting of Section 5.2, in which the value matrix satisfies $V_{(u,w),v} = \mathbb{1}\{v \in \{u, w\}\}$, the model only outputs vertices, and the model never outputs the same vertex twice consecutively (the previous vertex is masked). For any $n \in \mathbb{N}_{\geq 2}$, any distribution $\mathcal{Q}$ over $[n-1]$, and any $\varepsilon \in \mathbb{R}_{>0}$, there exist parameters $\theta = (A, V)$ for a single-layer linear Transformer such that $\mathcal{R}(\mathcal{D}^{\mathcal{Q}}, \theta) > 1 - \varepsilon$, and for every input sequence $s_{1:L_0}$ from $\mathcal{D}^{\mathcal{Q}}$, with probability at least $1 - \varepsilon$ over $\tau \sim p_{\text{roll}}(\theta)$, the rollout τ is a chain traversal (Section 3.3).*

Proof. We use the notation and symmetry setup of Lemma 68. As the transformer is assumed to be linear, we have $w(\mu) = \mu$.

Let A satisfy the assumptions of Lemma 68 with

$$\alpha = \alpha_0, \qquad \beta = 0, \qquad \gamma = 0,$$

for some free parameter $\alpha_0 \in \mathbb{R}_{>0}$ to be chosen later, and set all vertex–vertex entries of A equal to $\gamma = 0$. Let V satisfy the assumptions of Lemma 68 with

$$\alpha' = 1, \qquad \beta' = 1, \qquad \gamma' = 0,$$

as required by Section 5.2. Thus $\Delta_\alpha = \Delta_\beta = 1$.

Fix any time during autoregressive generation before termination, and let v be the last generated vertex (so the current input sequence s ends with v). Let u denote the (unique) successor of v in its chain if it exists. By Lemma 68, the successor u has logit contribution $\Psi_u = \Delta_\alpha w(\gamma) + \Delta_\beta w(\alpha) = \alpha_0$;

every other vertex $k \neq u, v$ has logit contribution $\Psi_k \in \{w(\beta), w(\gamma)\} = \{0\}$ or is a chain start with $\Psi_k = \Delta_\alpha w(\gamma) = 0$. The current vertex v has $\Psi_v = w(\alpha) + w(\beta) = \alpha_0$, but is masked out by assumption (as are all edge tokens). Hence there is a logit gap of the form

$$o(v, G)_u - o(v, G)_k \;\geq\; \alpha_0 \qquad \text{for all } k \neq u, v.$$

The next-token distribution is $\mathrm{TF}(s_{1:L}; \theta) = \mathrm{softmax}\big(o(v, G)\big)$. Using the logit gap and bounding the denominator by $|\Sigma|$ terms (excluding v), we obtain for every nonterminal v:

$$\Pr\left[s_{L+1} = u \,\middle|\, s_L = v\right] \geq \frac{e^{\alpha_0}}{e^{\alpha_0} + (|\Sigma| - 2) \cdot e^0} = \frac{1}{1 + (|\Sigma| - 2)e^{-\alpha_0}} \geq 1 - (|\Sigma| - 2)e^{-\alpha_0}.$$

All rollouts start from a vertex in one of the chains; at most n nonterminal-to-successor transitions are required to reach the chain terminal. By a union bound, the probability that the rollout ever deviates from the correct successor is at most

$$n(|\Sigma| - 2)e^{-\alpha_0}.$$

Whenever the rollout follows the chain edges until it reaches a terminal vertex, it terminates at the correct terminal $y^*(s_{1:L_0})$ and receives reward 1 (see Sections 3.2 and 3.3). Choosing $\alpha_0 = \log(n(|\Sigma| - 2)/\varepsilon)$ yields

$$1 - \mathcal{R}(\mathcal{D}^{\mathcal{Q}}, \theta) \;\leq\; n(|\Sigma| - 2)e^{-\alpha_0} \;=\; \varepsilon,$$

as claimed. □

D.2 Restatement of Proposition 2 Under the Setting of Section 5.2

Proposition 4. *Consider the setting of Section 5.2, in which the value matrix satisfies $V_{(u,w),v} = \mathbb{1}\{v \in \{u, w\}\}$, the model only outputs vertices, and the model never outputs the same vertex twice consecutively (the previous vertex is masked). For any $n \in \mathbb{N}_{\geq 3}$, any distribution $\mathcal{Q}$ over $[n-1]$, any $k \in \mathbb{N}$, and any $\varepsilon, \delta \in \mathbb{R}_{>0}$, there exist parameters $\theta = (A, V)$ for a single-layer linear Transformer such that $\mathcal{R}(\mathcal{D}^{\mathcal{Q}}, \theta) > 1 - \varepsilon$, and for every input sequence $s_{1:L_0}$ from $\mathcal{D}^{\mathcal{Q}}$, the probability that $|\tau| > k$ is greater than $1 - \delta$.*

Proof. Fix $n \geq 3$, a distribution Q over $[n-1]$, a target length threshold $k \in \mathbb{N}$, and a loss tolerance $\varepsilon > 0$. We assume w.l.o.g. that $\delta > \varepsilon$ (otherwise we can lower ε).

We use the notation and symmetry setup of Lemma 68, in particular the parameters α, β, γ in the attention matrix and α', β', γ' in the value matrix. We also reuse the notation $\Delta_\alpha := \alpha' - \gamma'$ and $\Delta_\beta := \beta' - \gamma'$. The formulas for the per-vertex logit contributions Ψ_k as a function of $(\alpha, \beta, \gamma, \Delta_\alpha, \Delta_\beta)$ are exactly those in Lemma 68. As the transformer is assumed to be linear, we have $w(\mu) = \mu$.

We construct a 1-layer transformer with attention parameters

$$\gamma = 0, \qquad \beta = \alpha_0 + \beta_0, \qquad \alpha = \alpha_0,$$

for some free parameters $\alpha_0, \beta_0 \in \mathbb{R}$ to be chosen later, and with value parameters

$$\gamma' = 0, \qquad \alpha' = 1, \qquad \beta' = 1,$$

as required by Section 5.2. Thus $\Delta_\alpha = 1$ and $\Delta_\beta = 1$. We keep the vertex–vertex block of A equal to $\gamma = 0$, and set $V_{v,:} = 0$ for all vertex tokens $v \in V$, exactly as in Lemma 68.

Let v be any non-terminal vertex in one of the chains, and consider the logits at the last position when the current prefix ends with v. Applying Lemma 68 and substituting $\gamma = 0$, $w(\gamma) = 0$, $\Delta_\alpha = 1$, $\Delta_\beta = 1$, we obtain for an interior vertex (Case 1 in Lemma 68) that the only nonzero logit contributions on the two incident vertices are

$$\Psi_{\mathrm{forw}} = \Delta_\alpha w(\gamma) + \Delta_\beta w(\alpha) = w(\alpha) = \alpha_0, \qquad \Psi_{\mathrm{back}} = \Delta_\alpha w(\beta) + \Delta_\beta w(\gamma) = w(\beta) = \alpha_0 + \beta_0.$$

The current vertex v itself has

$$\Psi_v = \Delta_\alpha w(\alpha) + \Delta_\beta w(\beta) = \alpha_0 + (\alpha_0 + \beta_0) = 2\alpha_0 + \beta_0,$$

but is masked out by assumption. Every other vertex has $\Psi_k = 0$ (since all those terms are proportional to $w(\gamma) = 0$ in Lemma 68); edge tokens are masked from the output. Thus, up to some offset C, any vertex v which is "in the middle of the chain" (not the start, end, second or penultimate vertex of some chain) has a local logit pattern of the form

$$\ell_{\mathrm{forw}} = \alpha_0, \quad \ell_{\mathrm{back}} = \alpha_0 + \beta_0, \quad \ell_{\mathrm{other}} = 0.$$

The same calculation applied to vertices v which are not the start of a chain (Cases 2,3,5 in Lemma 68) gives the same qualitative picture: for any non-terminal v on either chain the logits on the two incident chain vertices are

$$\ell_{\mathrm{forw}} = \alpha_0, \qquad \ell_{\mathrm{back}} = \alpha_0 + \beta_0,$$

while all other vertices have logit 0. In particular the *only* dependence on β_0 in the *differences* between the forward and backward logits is

$$\ell_{\mathrm{forw}} - \ell_{\mathrm{back}} = -\beta_0.$$

Finally, when the current vertex v is the start of a chain (Case 4 in Lemma 68), there is no backward neighbor within the chain. In this case, the forward logit is still $\ell_{\mathrm{forw}} = \alpha_0$, but there is no ℓ_{back}, so as $\alpha_0 \to \infty$, the probability to move forward limits to one.

We now use this to tune the random-walk behaviour in two stages.

1. Controlling the conditional walk along the chain (choosing β_0). Fix α_0 for the moment. Condition on the "good event" that at each step the model only chooses the forward or backward neighbour in the chain (i.e., we ignore the possibility of jumping to any other vertex). Under this conditioning, from every interior vertex v the next-token distribution over $\{\mathrm{forw}, \mathrm{back}\}$ has the form

$$\mathbb{P}(\mathrm{forw} \mid \{\mathrm{forw}, \mathrm{back}\}) = \frac{e^{\ell_{\mathrm{forw}}}}{e^{\ell_{\mathrm{forw}}} + e^{\ell_{\mathrm{back}}}} = \frac{1}{1 + e^{-(\ell_{\mathrm{forw}} - \ell_{\mathrm{back}})}} = \frac{1}{1 + e^{\beta_0}},$$
$$\mathbb{P}(\mathrm{back} \mid \{\mathrm{forw}, \mathrm{back}\}) = \frac{1}{1 + e^{-\beta_0}}.$$

Thus the forward/backward bias of the induced nearest-neighbour walk along the chain depends *only* on β_0, via a smooth, strictly monotone function. As $\beta_0 \to 0$ the walk becomes approximately unbiased ($\mathbb{P}(\mathrm{forw}) \to 1/2$); as $\beta_0 \to -\infty$ the walk becomes almost deterministic in the forward direction; as $\beta_0 \to +\infty$ it becomes almost deterministic in the backward direction.

Let us denote by G the event that the rollout consists only of forward/backward moves along the chain. Standard random-walk estimates on a finite chain imply that, for any fixed chain length n, and starting index $k \in [n-1]$, there exists a choice of β_0 and some $L_{\max}$ (dependent on β_0) such that, *conditional on* G, the hitting time T of the terminal vertex satisfies

$$\mathbb{P}\big(L_{\max} > T > L \;\big|\; G\big) \geq 1 - \varepsilon/2$$

for any prescribed L. This implies that conditioned on G, the rollout will take more than L and less than $L_{\max}$ steps to reach the terminal vertex with probability at least $1 - \varepsilon/2$ (which assures loss smaller than $1 - \varepsilon/2$).

2. Making $\mathbb{P}(G^c) \leq \varepsilon/2$ (choosing α_0). We now choose α_0 to ensure that the model stays on the chain. Taking $\alpha_0 \to \infty$, the logits for the forward and backward neighbors ($\ell_{\mathrm{forw}} = \alpha_0$, $\ell_{\mathrm{back}} = \alpha_0 + \beta_0$) tend to $+\infty$, while other vertices remain at 0 (edge tokens are masked). Consequently, the probability mass concentrates on the forward and backward neighbors, with the ratio $p_{\mathrm{forw}}/p_{\mathrm{back}} = e^{-\beta_0}$ preserved. The probability of outputting any other vertex vanishes relative to the move probabilities.

Together, these two points imply the theorem. □

E Proof of Theorem 1

The proof proceeds through a chain of reductions in circuit complexity. We first establish that if greedy decoding fails on any input, the single-step reward is bounded away from 1 (Appendix E.1). We then recall the relevant complexity classes and problems (Appendix E.2), and show that constant-depth Transformers without CoT can be simulated by TC^0 circuits (Lemma 10). The core technical work reduces the NC^1-complete word problem for S_5 to path reachability (ORD), and then to the two-chain endpoint problem. Since $\mathrm{TC}^0 \neq \mathrm{NC}^1$ under standard assumptions, no TC^0 circuit—and hence no constant-depth Transformer—can solve the endpoint problem on all inputs.

E.1 Relation between reward, complexity, and decision problems

In this section we connect the computational hardness of the chain identification task to an upper bound on the single-step reward. Our approach uses deterministic complexity classes: we show that for infinitely many chain lengths n, any constant-depth Transformer without CoT fails to compute the correct answer via greedy decoding on at least one input.

Lemma 1. *Let $\mathcal{Q}_{\text{unif}}$ denote the uniform distribution over the index set $[n-1]$, so that $\mathcal{D}^{\mathcal{Q}_{\text{unif}}}$ is the uniform distribution over all (graph, starting vertex) pairs (Section 3.1). If there exists at least one input $s \in \text{supp}(\mathcal{D}^{\mathcal{Q}_{\text{unif}}})$ on which greedy decoding does not output $y^*(s)$, then*

$$\mathcal{R}_{\text{single}}(\mathcal{D}^{\mathcal{Q}_{\text{unif}}}, \theta) \leq 1 - \frac{1}{(2n)! \cdot 2(n-1)}.$$

Proof. The support of $\mathcal{D}^{\mathcal{Q}_{\text{unif}}}$ consists of all pairs of a two-chain graph on $2n$ vertices and a non-terminal starting vertex. The number of such pairs is $(2n)! \cdot (n-1)$ (there are $(2n)!/2$ ordered two-chain graphs, each with $2(n-1)$ non-terminal starting vertices, but each unordered graph is counted twice).

Under the uniform distribution, the single-step reward is

$$\mathcal{R}_{\text{single}}(\mathcal{D}^{\mathcal{Q}_{\text{unif}}}, \theta) = \frac{1}{|\text{supp}(\mathcal{D}^{\mathcal{Q}_{\text{unif}}})|} \sum_{s \in \text{supp}(\mathcal{D}^{\mathcal{Q}_{\text{unif}}})} \Pr\left(\text{TF}(s;\theta) = y^*(s)\right).$$

If greedy decoding fails on input s^*, then $\arg\max_y \text{TF}(s^*;\theta)_y \neq y^*(s^*)$, which implies $\Pr(\text{TF}(s^*;\theta) = y^*(s^*)) \leq 1/2$ (since the correct answer does not have the maximal logit). Thus at least one term in the sum is at most $1/2$, yielding

$$\mathcal{R}_{\text{single}}(\mathcal{D}^{\mathcal{Q}_{\text{unif}}}, \theta) \leq 1 - \frac{1}{2 \cdot (2n)! \cdot (n-1)} = 1 - \frac{1}{(2n)! \cdot 2(n-1)}.$$

□

We also utilize the decision version of the two-chain endpoint problem (which returns true/false for a pair of vertices depending on whether the latter is the endpoint of the chain containing the former); this will be expanded upon in Appendix E.7.

E.2 Complexity-Theoretic Backgrounnd

Our hardness result relies on standard assumptions from circuit complexity theory regarding the computational power of constant-depth versus logarithmic-depth circuits. Circuit complexity characterizes computational problems by the resources required by boolean circuits to solve them. A circuit family consists of a circuit for each input length n, where each circuit is a directed acyclic graph with input nodes, internal gates (computing boolean functions), and output nodes. The key resources are the *size* (number of gates) and *depth* (longest path from input to output).

Definition 3. A language $L \subseteq \{0,1\}^*$ is in L if there exists a deterministic Turing machine M and a constant $c \in \mathbb{N}$ such that for every input $x \in \{0,1\}^n$:

- M halts on all inputs, and accepts x if and only if $x \in L$,
- On input x, the total number of work-tape cells that M ever scans is at most $c \log n$ (i.e., M uses $O(\log n)$ space).

We next define the circuit classes central to our analysis.

Definition 4. A language $L \subseteq \{0,1\}^*$ is in AC^0 if there exists a constant $d \in \mathbb{N}$, a polynomial $p(\cdot)$, and a family of boolean circuits $\{C_n\}_{n \in \mathbb{N}}$ such that for all $n \in \mathbb{N}$:

- C_n has depth at most d and size at most $p(n)$,
- C_n uses unbounded fan-in AND, OR, and NOT gates,
- For all $x \in \{0,1\}^n$, $C_n(x) = \mathbb{1}[x \in L]$.

It is known that simple problems like PARITY (computing the XOR of all input bits) provably lie outside AC^0 [26, 3].

Definition 5. A language $L \subseteq \{0,1\}^*$ is in TC^0 if there exists a constant $d \in \mathbb{N}$, a polynomial $p(\cdot)$, and a family of boolean circuits $\{C_n\}_{n\in\mathbb{N}}$ such that for all $n \in \mathbb{N}$:

- C_n has depth at most d and size at most $p(n)$,
- C_n uses unbounded fan-in AND, OR, NOT, and MAJORITY gates, where a MAJORITY gate outputs 1 if and only if the majority of its inputs are 1,
- For all $x \in \{0,1\}^n$, $C_n(x) = \mathbb{1}[x \in L]$.

The additional power of MAJORITY gates allows TC^0 to compute functions like PARITY which lie outside AC^0, and thus $\mathrm{AC}^0 \subsetneq \mathrm{TC}^0$ [8].

Definition 6. A language $L \subseteq \{0,1\}^*$ is in NC^1 if there exists a polynomial $p(\cdot)$ and a family of boolean circuits $\{C_n\}_{n\in\mathbb{N}}$ such that for all $n \in \mathbb{N}$:

- C_n has depth $O(\log n)$ and size at most $p(n)$,
- C_n uses bounded fan-in (typically fan-in 2) AND, OR, and NOT gates,
- For all $x \in \{0,1\}^n$, $C_n(x) = \mathbb{1}[x \in L]$.

This class corresponds to problems efficiently parallelizable with a logarithmic number of parallel steps. Examples include evaluating boolean formulas and computing certain regular languages [18].

Definition 7. Let $\mathcal{C}$ be a circuit class defined via families $\{C_n\}_{n\in\mathbb{N}}$ (e.g., $\mathrm{AC}^0, \mathrm{TC}^0, \mathrm{NC}^1$). A family $\{C_n\}_{n\in\mathbb{N}}$ is X-*uniform* if there exists a function $U \in \mathsf{X}$ such that, on input 1^n, U outputs a description of C_n.

A language $L \subseteq \{0,1\}^*$ is in the X-*uniform version of* $\mathcal{C}$ if it is decided by some X-uniform family $\{C_n\} \in \mathcal{C}$.

Throughout, all circuit classes (AC^0, TC^0, NC^1) are non–uniform unless explicitly marked as uniform.

Conjecture 1. *There exist problems in* NC^1 *that are not in* TC^0.

This separation- which is widely believed to hold- remains unproven and is considered a fundamental open problem in complexity theory [67]. Several natural problems are candidates for witnessing this separation, including certain composition operations on non-solvable groups and the iterated multiplication problem [42].

E.3 A Complexity–Theoretic Obstruction for No–CoT Transformers

In this section we formalize a conditional lower bound showing that, under the standard conjecture $\mathrm{TC}^0 \neq \mathrm{NC}^1$, a constant–depth transformer without chain–of–thought (CoT) steps cannot compute a simple "two–chain endpoint" function. The argument proceeds via a chain of reductions starting from an NC^1–complete word problem, passing through the logspace–complete problem ORD, and ending at a decision version of the two–chain endpoint problem.

E.4 Problems and Classes

We begin by fixing the problems we will use.

Definition 8. Let S_5 be the Symmetric group over 5 elements. The *word problem for* S_5, denoted WP_{S_5}, is the language

$$\mathrm{WP}_{S_5} \;=\; \{\, w \in \Sigma^* : \text{the product of the letters of } w \text{ equals the identity of } S_5 \,\}.$$

Definition 9. Fix $n \in \mathbb{N}$. An ORD *instance* on $[n] = \{1, \ldots, n\}$ consists of:

- a *successor function* $\mathsf{succ} : [n] \to [n] \cup \{\perp\}$, where $\mathsf{succ}(i) = j$ encodes a directed edge $i \to j$ and $\mathsf{succ}(i) = \perp$ means i has no outgoing edge;

- two distinguished vertices $s, t \in [n]$.

Equivalently, an ORD instance is a directed line graph (i.e a single chain), $P = ([n], E)$ with edges $E = \{(i, \mathsf{succ}(i)) : \mathsf{succ}(i) \neq \perp\}$ and two distinguished vertices $s, t \in [n]$.

The language ORD is:

$$\mathrm{ORD} = \{(\mathsf{succ}, s, t) : t \text{ is reachable from } s \text{ along the unique directed path on } [n]\}.$$

Definition 10. A directed graph $G = ([2n], E)$ is a *pair of equal-length directed chains* if G consists of exactly two connected components, each being a directed chain on n vertices.

- The *two–chain endpoint function*

$$F_{\mathrm{TwoChain}} : \{\, G, v \,\} \longrightarrow [n]$$

  outputs the unique terminal vertex of the (unique) directed chain of G that contains v. (If v is itself terminal, then $F_{\mathrm{TwoChain}}(G, v) = v$.)

- The associated *decision problem* is

$$\mathrm{TwoChainEndEq} = \left\{ (G, v, q) : \begin{array}{l} G \text{ is a pair of equal-length directed chains,} \\ F_{\mathrm{TwoChain}}(G, v) = q \end{array} \right\}.$$

Definition 11. Let $\mathcal{T}_0$ denote the class of functions computed (with no COT) by transformers with:

- Constant depth.
- Width, embedding dimension and number of attention heads at most $\mathrm{poly}(n)$ on length-n inputs.
- Each parameter is represented with $O(\log n)$ bits of precision in each scalar.

We view each such transformer family as a non–uniform circuit family (one transformer per input length).

E.5 Complexity–theoretic results

We collect the complexity–theoretic ingredients we need.

Lemma 2. $\mathrm{WP}_{S_5} \in \mathrm{NC}^1$ *and* WP_{S_5} *is* NC^1*–complete under (non–uniform)* AC^0 *many–one reductions. That is, for every language* $L \in \mathrm{NC}^1$ *there is a family of* AC^0 *circuits* (f_n) *such that for all* x*,*

$$x \in L \iff f_{|x|}(x) \in \mathrm{WP}_{S_5}.$$

Proof. See [42]. □

The next lemma shows that this completeness extends to the uniform setting.

Lemma 3. $\mathrm{WP}_{S_5} \in \mathrm{NC}^1$ *and* WP_{S_5} *is DLOGTIME–uniform* NC^1*–complete.*

Proof. See [9]. □

This lemma establishes a fundamental containment between uniform circuit classes and space complexity.

Lemma 4. *Let* $\mathrm{NC}^1_{\mathsf{u}}$ *denote DLOGTIME–uniform* NC^1*, and let* L *denote deterministic logspace. Then*

$$\mathrm{NC}^1_{\mathsf{u}} \subseteq \mathsf{L}.$$

Proof. See [13]. □

The Ordering/Path Reachability Problem serves as a canonical complete problem for logspace.

Lemma 5. $\mathrm{ORD} \in \mathsf{L}$ *and* ORD *is* L*–complete under DLOGTIME–uniform* AC^0 *reductions (equivalently, first–order interpretations). That is, for every* $A \in \mathsf{L}$ *there is a DLOGTIME–uniform* AC^0 *function* f *with*

$$x \in A \iff f(x) \in \mathrm{ORD}.$$

Proof. See [22]. □

Combining the previous lemmas, we obtain an AC^0 reduction from the word problem to ORD.

Lemma 6. *Fix a group* G *as in Lemma 2. Then there exists a DLOGTIME–uniform* AC^0 *many–one reduction* f *such that*

$$w \in \mathrm{WP}_G \iff f(w) \in \mathrm{ORD}.$$

Equivalently, $\mathrm{WP}_G \leq_m^{\mathrm{AC}^0} \mathrm{ORD}$.

Proof. By Lemma 2 and Lemma 4, the (uniform) language WP_G is in L. By Lemma 5, there is a DLOGTIME–uniform AC^0 reduction f such that $w \in \mathrm{WP}_G \iff f(w) \in \mathrm{ORD}$. This is precisely an $\mathrm{AC}^0_{\mathsf{u}}$ (hence AC^0) many–one reduction as claimed. □

We next state the transformer-to-TC^0 upper bound and a basic closure property. We start with some lemmas from Merrill–Sabharwal [46]

Lemma 7. *Let* $c > 0$ *and* $f : \{0,1\}^{c \log N} \to \{0,1\}^m$. *There is a depth-3* $\{\wedge, \vee, \neg\}$*-circuit of size* $O(N^c + \log N + m)$ *computing* f.

Proof. see [46] (Lemma 1) □

The next lemma extends this to iterated addition with finite precision.

Lemma 8. *Let* $p = c \log N$. *There exists a constant* $d_\oplus$ *such that for each* $n \leq N$ *the* p*-precision sum of* $x_1, \ldots, x_n \in \{0,1\}^p$ *is computed by a depth-*$d_\oplus$*, polynomial-size threshold circuit.*

Proof. see [46] (Appendix A) □

Directly from 7 we can obtain the following lemma:

Lemma 9. *Let* $p = c \log N$. *The truncated/rounded product* $(u, v) \in \{0,1\}^p \times \{0,1\}^p \mapsto u \cdot v \in \{0,1\}^p$ *,and has a depth-*$d_\times$ *implementation with* $d_\times = 3$ *and polynomial size.*

Proof. This follows directly from 7. □

The following lemma is the key connection between no-CoT Transformers and circuit complexity.

Lemma 10. *For every function family* F *computed by a family of no–CoT transformers in* $\mathcal{T}_0$*, there is a non–uniform* TC^0 *circuit family computing* F. *In other words,*

$$\mathcal{T}_0 \subseteq \mathrm{TC}^0.$$

Proof. We define a family of models $\{T_n\}$, where a model T_n accepts input from vocabulary Σ_n of size $|\Sigma_n| = n^2$.

Fix $n \in \mathbb{N}$. The input is a sequence of 1-hot vectors of length $l = n - 2$, $(h_1^{(0)}, \ldots, h_l^{(0)})$ with $h_i^{(0)} \in \{e_1, ..., e_{|\Sigma_n|}, 0\}$. Hence the total input bit-length is

$$N \;=\; l \cdot |\Sigma_n| \;=\; n^2 * (n-2). \tag{7}$$

Let $L \geq 1$ be a fixed constant. For each layer $\ell \in \{0, \ldots, L-1\}$ there are shared matrices

$$Q^{(\ell)},\; K^{(\ell)},\; V^{(\ell)} \;\in\; \mathbb{R}^{|\Sigma_n| \times |\Sigma_n|} \tag{8}$$

with p-bit entries, where

$$p \;=\; c \log N \tag{9}$$

for a fixed constant $c > 0$. Hidden vectors at layer ℓ are $h_i^{(\ell)} \in \mathbb{R}^{|\Sigma_n|}$ (with p-bit coordinates); for $\ell = 0$, $h_i^{(0)}$ is the 1-hot input.

For all positions i, j the layer computes

$$q_i^{(\ell)} = Q^{(\ell)}\, h_i^{(\ell)}, \tag{10}$$

$$k_j^{(\ell)} = K^{(\ell)}\, h_j^{(\ell)}, \tag{11}$$

$$v_j^{(\ell)} = V^{(\ell)}\, h_j^{(\ell)}, \tag{12}$$

$$r_{i,j}^{(\ell)} = \sum_{t=1}^{|\Sigma_n|} q_i^{(\ell)}[t]\; k_j^{(\ell)}[t], \tag{13}$$

$$s_{i,j}^{(\ell)} = \begin{cases} r_{i,j}^{(\ell)} & \text{(Linear)}, \\ \dfrac{\exp\big(r_{i,j}^{(\ell)}\big)}{\sum_{j'=1}^{l} \exp\big(r_{i,j'}^{(\ell)}\big)} & \text{(Softmax)}, \end{cases} \tag{14}$$

$$a_i^{(\ell)} = \sum_{j=1}^{l} s_{i,j}^{(\ell)}\, v_j^{(\ell)}, \qquad h_i^{(\ell+1)} := a_i^{(\ell)}. \tag{15}$$

Every scalar multiplication returns a p-bit result under a fixed truncation/rounding rule; all sums use the p-precision addition operator from Appendix A of Merrill–Sabharwal[47].

We will now show that the Forward Pass of the described model is in TC^0.

All steps below are performed in parallel over the relevant indices and coordinates.

For each possible i,j, equations 10, 11, and 12 are computed in the following way:

$$q_i^{(\ell)}[t] = \sum_{r=1}^{|\Sigma_n|} Q^{(\ell)}[t, r]\; h_i^{(\ell)}[r], \tag{16}$$

$$k_j^{(\ell)}[t] = \sum_{r=1}^{|\Sigma_n|} K^{(\ell)}[t, r]\; h_j^{(\ell)}[r], \tag{17}$$

$$v_j^{(\ell)}[t] = \sum_{r=1}^{|\Sigma_n|} V^{(\ell)}[t, r]\; h_j^{(\ell)}[r]. \tag{18}$$

Each product is a p-bit multiply (Lemma 9, depth $d_\times$); each sum is an iterated p-precision sum on $|\Sigma_n| = O(N)$ elements (Lemma 8, depth $d_\oplus$). All three MV products are computed in parallel, so this stage costs $d_\times + d_\oplus$ depth.

We now analyze softmax and linear attention separately.

For linear attention, we have the following Similarities computation:

$$s_{i,j}^{(\ell)} \;=\; \sum_{t=1}^{|\Sigma_n|} q_i^{(\ell)}[t]\; k_j^{(\ell)}[t], \tag{19}$$

computed by p-bit multiplies (depth $d_\times$) followed by a p-precision sum (depth $d_\oplus$) for a total of $d_\times + d_\oplus$.

For softmax attention, we have the following computation:
Given the scores $r_{i,j}^{(\ell)}$ from the model definition, compute $u_{i,j}^{(\ell)} := \exp_p(r_{i,j}^{(\ell)})$. By Lemma 7, $\exp_p : \{0,1\}^p \to \{0,1\}^p$ has a depth-3, polynomial-size implementation. Next, compute $Z_i^{(\ell)} := \sum_{j'=1}^{l} u_{i,j'}^{(\ell)}$ using Lemma 8 (depth $d_\oplus$). Finally output $s_{i,j}^{(\ell)} := \div_p(u_{i,j}^{(\ell)}, Z_i^{(\ell)})$ (where $\div_p$ is p-precision division); since $\div_p$ depends on $2p = O(\log N)$ input bits, Lemma 7 gives depth 3 and polynomial size for a total of $d_\oplus + 6$.

Finally, we analyze the remaining step in 15:
For each (i, j) and coordinate k, we compute the sum element

$$(s_{i,j}^{(\ell)}\, v_j^{(\ell)})[k] \;=\; s_{i,j}^{(\ell)}\, v_j^{(\ell)}[k], \tag{20}$$

with depth $d_\times$ and polynomial size via Lemma 9.

We then sum for each i and coordinate k,

$$a_i^{(\ell)}[k] \;=\; \sum_{j=1}^{l} \big(s_{i,j}^{(\ell)}\, v_j^{(\ell)}[k]\big), \tag{21}$$

which is an iterated p-precision sum of l p-bit values (Lemma 8) for a total of $d_\oplus + d_\times$.

We have shown that each layer has constant depth, and we parallelize multiple fixed-depth polynomial threshold circuits over a polynomial number of indices which yealds us a polynomial total size. Therefore, Stacking $L = O(1)$ layers gives overall constant depth and polynomial size. Parameters are hard-wired, so the family is non-uniform. Hence $T_n \in \mathrm{TC}^0$. □

Remarks: This model represents the model described in the main text by setting $L = 1$, the number of heads as 1, and $|\Sigma|_n$ to be the vocabulary of the model for graphs with n nodes. The input size to the model is consistent as any 2 chain graph on n nodes can be represente with exactly $n - 2$ edges.

Lemma 11. *Let L_1, L_2 be languages with $L_1 \leq_m^{\mathrm{AC}^0} L_2$. If $L_2 \in \mathrm{TC}^0$, then $L_1 \in \mathrm{TC}^0$ as well.*

Proof sketch. Let (f_n) be the AC^0 reduction and (C_n) a TC^0 circuit family for L_2. For each input length n, define a circuit $D_n(x) = C_{|f_n(x)|}(f_n(x))$. The D_n have constant depth, polynomial size, and threshold gates: they are obtained by composing an AC^0 circuit with a TC^0 circuit. Thus (D_n) is a TC^0 family deciding L_1. □

E.6 From ORD to Two–Chain Endpoint

We now formalize the reduction from ORD to the decision version of the two–chain endpoint problem.

We fix the following encodings.

Definition 12. An ORD instance (succ, s, t) on $[n]$ is encoded as:

- For each $i \in [n]$, the value $\mathsf{succ}(i) \in [n+1]$, where we identify $\perp$ with 0, written in binary using $\lceil \log(n+1) \rceil$ bits;
- the vertices $s, t \in [n]$, written in binary using $\lceil \log n \rceil$ bits each.

The encoding of a TwoChainEndEq instance (G, v, q) uses the same representation for G via a successor function succ_G on $[n]$, together with $v, q \in [n]$ in binary.

Lemma 12. *There is a DLOGTIME–uniform AC^0 many–one reduction*

$$(\mathsf{succ}, s, t) \mapsto (\widetilde{\mathsf{succ}}, \tilde{v}, \tilde{q})$$

such that

$$(\mathsf{succ}, s, t) \in \mathrm{ORD} \iff (\widetilde{\mathsf{succ}}, \tilde{v}, \tilde{q}) \in \mathrm{TwoChainEndEq},$$

and moreover the directed graph induced by $\widetilde{\mathsf{succ}}$ is a disjoint union of exactly two *directed chains of the same length.*

Proof. Let (succ, s, t) be an ORD instance on $[n]$ satisfying the promise that $P = ([n], E)$ is a single directed path.

Encoding convention. We will build an output instance on $2n$ vertices, which we identify with the set $[n] \times \{0, 1\}$. We encode a vertex (i, b) by writing the copy bit b (one bit) followed by the binary encoding of $i \in [n]$ using $\lceil \log n \rceil$ bits. The special value $\perp$ is encoded as all zeros. This is just a fixed representation choice for this reduction.

(1) Construction of $(\widetilde{\mathsf{succ}}, \tilde{v}, \tilde{q})$. Define $\widetilde{\mathsf{succ}} : [n] \times \{0,1\} \to ([n] \times \{0,1\}) \cup \{\bot\}$ as follows. For every $i \in [n]$ and $b \in \{0,1\}$,

$$\widetilde{\mathsf{succ}}(i,b) \;=\; \begin{cases} \bot, & \text{if } \mathsf{succ}(i) = \bot, \\ (\mathsf{succ}(i),\, b), & \text{if } \mathsf{succ}(i) \neq \bot \text{ and } i \neq t, \\ (\mathsf{succ}(t),\, 1-b), & \text{if } \mathsf{succ}(t) \neq \bot \text{ and } i = t. \end{cases}$$

Set $\tilde{v} := (s, 0)$.

Let $\mathrm{end} \in [n]$ be the (unique) terminal vertex of the input path, i.e. the unique vertex with $\mathsf{succ}(\mathrm{end}) = \bot$. Define the target vertex $\tilde{q}$ by

$$\tilde{q} := \begin{cases} (\mathrm{end}, 1), & \text{if } \mathsf{succ}(t) \neq \bot, \\ (\mathrm{end}, 0), & \text{if } \mathsf{succ}(t) = \bot. \end{cases}$$

(2) Structure: two equal–length chains. Consider the input path P. For each copy bit b, the edges $(i,b) \to (\mathsf{succ}(i), b)$ follow P within copy b except at the single vertex (t,b), where (when $\mathsf{succ}(t) \neq \bot$) the edge is redirected to the other copy. Therefore the graph on $[n] \times \{0,1\}$ induced by $\widetilde{\mathsf{succ}}$ has out-degree at most one everywhere.

If $\mathsf{succ}(t) = \bot$, then $\widetilde{\mathsf{succ}}(t,b) = \bot$ and the output graph is simply two disjoint copies of the same directed path, hence exactly two directed chains, each on n vertices.

If $\mathsf{succ}(t) \neq \bot$, let $u := \mathsf{succ}(t)$. Cutting the input path at the edge $t \to u$ decomposes it into a prefix ending at t and a suffix starting at u and ending at end. The redirections $(t,0) \to (u,1)$ and $(t,1) \to (u,0)$ "swap" the suffixes between the two copies. Hence the output graph is the disjoint union of exactly the two chains:

$$\big(\text{prefix in copy } 0\big) \;\cdot\; \big(\text{suffix in copy } 1\big), \qquad \big(\text{prefix in copy } 1\big) \;\cdot\; \big(\text{suffix in copy } 0\big).$$

Each chain contains exactly one copy of every original vertex, so each has exactly n vertices. Thus the output is a union of exactly two directed chains of the same length.

(3) Correctness. We show that $(\mathsf{succ}, s, t) \in \mathrm{ORD} \iff (\widetilde{\mathsf{succ}}, \tilde{v}, \tilde{q}) \in \mathrm{TwoChainEndEq}$.

- If $\mathsf{succ}(t) = \bot$, then $t = \mathrm{end}$ is the terminal of the unique directed path on $[n]$. Since the input graph is a single directed path, t is reachable from every s, so $(\mathsf{succ}, s, t) \in \mathrm{ORD}$. In the output graph, the chain containing $\tilde{v} = (s,0)$ is exactly copy 0 of the path, whose terminal is $(\mathrm{end}, 0) = \tilde{q}$. Hence $(\widetilde{\mathsf{succ}}, \tilde{v}, \tilde{q}) \in \mathrm{TwoChainEndEq}$.
- Suppose $\mathsf{succ}(t) \neq \bot$. Consider the unique directed walk from s in the input path. If t is reachable from s, then the walk reaches t, so in the output the walk from $(s,0)$ reaches $(t,0)$ and then crosses to copy 1, after which it follows the suffix in copy 1 until terminating at $(\mathrm{end}, 1) = \tilde{q}$. Thus $F_{\mathrm{TwoChain}}(\widetilde{G}, \tilde{v}) = \tilde{q}$. If t is not reachable from s, then the walk from $(s,0)$ never reaches $(t,0)$, hence never crosses copies, and it terminates at $(\mathrm{end}, 0) \neq (\mathrm{end}, 1) = \tilde{q}$. Therefore $F_{\mathrm{TwoChain}}(\widetilde{G}, \tilde{v}) \neq \tilde{q}$.

This proves the desired equivalence.

(4) AC^0 *and uniformity.* We argue that the mapping is computable by a DLOGTIME–uniform AC^0 circuit family.

First, $\tilde{v} = (s,0)$ is obtained by copying the bits of s and setting the leading copy bit to 0.

Next, the bits of end are selectable in AC^0 under the promise that there is a unique terminal vertex. For each $i \in [n]$ let $\mathrm{IsEnd}(i)$ be the predicate "$\mathsf{succ}(i) = \bot$", which is a depth–2 AC^0 check that all bits of the encoding of $\mathsf{succ}(i)$ are 0. Then each bit of end can be written as an unbounded OR over $i \in [n]$:

$$\mathrm{end}_r \;=\; \bigvee_{i=1}^{n} \Big(\mathrm{IsEnd}(i) \;\wedge\; \mathrm{bit}_r(i) \Big),$$

where $\mathrm{bit}_r(i)$ is the r-th bit of the (fixed) binary encoding of the constant index i. The copy bit of $\tilde{q}$ is 1 iff $\mathsf{succ}(t) \neq \bot$, which is again a depth–2 AC^0 predicate on the bits of $\mathsf{succ}(t)$. Thus $\tilde{q}$ is computable in constant depth.

Finally, each output entry $\widetilde{\mathsf{succ}}(i, b)$ is obtained from the input bits of $\mathsf{succ}(i)$ and a depth–2 AC^0 equality test between the constant index i and the binary input t, together with the predicate $\mathsf{succ}(t) \neq \bot$. The rule above is a constant-depth case distinction implemented by AND/OR/NOT gates. DLOGTIME uniformity follows since the circuit only uses fixed wiring patterns plus the standard uniform equality gadgets.

This completes the proof. □

E.7 From the Function Problem to the Decision Problem

We now relate the function F_{TwoChain} to the decision problem $\mathrm{TwoChainEndEq}$.

Lemma 13. *Let F_{TwoChain} be the two–chain endpoint function. Define the associated decision language*

$$L_{F_{\mathrm{TwoChain}}} = \{\, (x, q) : F_{\mathrm{TwoChain}}(x) = q \,\},$$

where x encodes (G, v) and q encodes a vertex in $[n]$. Then $\mathrm{TwoChainEndEq} = L_{F_{\mathrm{TwoChain}}}$, and the mapping $(G, v, q) \mapsto ((G, v), q)$ is computable in DLOGTIME–uniform AC^0.

Proof. By definition, $\mathrm{TwoChainEndEq}$ is precisely the set of triples (G, v, q) such that $F_{\mathrm{TwoChain}}(G, v) = q$. The encoding of instances in $L_{F_{\mathrm{TwoChain}}}$ is just the concatenation of the encoding of (G, v) and the encoding of q, which matches the encoding of (G, v, q) used above, up to a trivial re-ordering of fields. The re-ordering is implemented by wiring and does not require any gates, so it is computable in depth 1. Thus $\mathrm{TwoChainEndEq} = L_{F_{\mathrm{TwoChain}}}$ and the identity mapping is a DLOGTIME–uniform AC^0 transformation between the two encodings. □

The next lemma shows that computing the function in TC^0 implies deciding the equality problem in TC^0.

Lemma 14. *Suppose F_{TwoChain} is computable by a non–uniform TC^0 circuit family. Then $\mathrm{TwoChainEndEq} \in \mathrm{TC}^0$.*

Proof. Let (C_n) be TC^0 circuits computing F_{TwoChain} on inputs of length n. To decide whether $F_{\mathrm{TwoChain}}(x) = q$, we can:

- feed x into $C_{|x|}$ to obtain an output y encoding a vertex in $[n]$;
- compare y and q bitwise using an AC^0 equality circuit: $y = q$ iff each bit matches.

Equality on $O(\log n)$ bits is in AC^0, and the composition of a TC^0 circuit with an AC^0 post–processing is still a TC^0 circuit by the closure Lemma 11. Thus $\mathrm{TwoChainEndEq} = L_{F_{\mathrm{TwoChain}}} \in \mathrm{TC}^0$. □

E.8 Main Theorem: No–CoT Transformers Cannot Solve Two–Chain Endpoint

We are now ready to state and prove our main result.

Theorem 5. *Assume $\mathrm{TC}^0 \neq \mathrm{NC}^1$. Then there is no family of no–CoT transformers in $\mathcal{T}_0$ that computes F_{TwoChain} on all valid inputs. Equivalently, $F_{\mathrm{TwoChain}} \notin \mathcal{T}_0$.*

Proof. Suppose, for contradiction, that there is a family of no–CoT transformers computing F_{TwoChain} exactly on all valid inputs (graphs that are unions of at most two chains).

By Lemma 10, we have $\mathcal{T}_0 \subseteq \mathrm{TC}^0$, so $F_{\mathrm{TwoChain}} \in \mathrm{TC}^0$. By Lemma 14, this implies $\mathrm{TwoChainEndEq} \in \mathrm{TC}^0$.

By Lemma 12, we have $\mathrm{ORD} \leq_m^{\mathrm{AC}^0} \mathrm{TwoChainEndEq}$. By the closure Lemma 11, it follows that $\mathrm{ORD} \in \mathrm{TC}^0$.

By Lemma 6, we also have $\mathrm{WP}_G \leq_m^{\mathrm{AC}^0} \mathrm{ORD}$ for some fixed finite group G. Combining this with $\mathrm{ORD} \in \mathrm{TC}^0$ and applying Lemma 11 again, we obtain $\mathrm{WP}_G \in \mathrm{TC}^0$.

Finally, by Lemma 2, WP_G is NC^1–complete under AC^0 reductions. Hence for every language $L \in \mathrm{NC}^1$, there is an AC^0 reduction $L \leq_m^{\mathrm{AC}^0} \mathrm{WP}_G$. By Lemma 11, this implies $L \in \mathrm{TC}^0$. Therefore $\mathrm{NC}^1 \subseteq \mathrm{TC}^0$.

Since we always have $\mathrm{TC}^0 \subseteq \mathrm{NC}^1$, we conclude $\mathrm{TC}^0 = \mathrm{NC}^1$, contradicting the assumption that these classes are different. Thus our initial assumption that a no–CoT transformer can compute F_{TwoChain} must be false. □

E.9 Extensions

We justify the two extensions claimed in Footnote 5 (Section 4.1): first, that Theorem 5 extends to Transformer architectures incorporating MLP and normalization layers; second, that it extends to Transformers that are allowed a constant number of chain-of-thought steps.

E.9.0.1 MLP and normalization layers. The proof of Theorem 5 relies on the inclusion $\mathcal{T}_0 \subseteq \mathrm{TC}^0$ (Lemma 10). This containment continues to hold when the architecture is augmented with MLP layers of width polynomial in n (composed of linear maps and element-wise activations such as ReLU or GELU) and normalization layers (LayerNorm or RMSNorm), provided all parameters use $O(\log n)$-bit precision. The polynomial-width condition is essential for the resulting circuit family to have polynomial size. Indeed, each such MLP layer computes a polynomial-size composition of threshold and arithmetic operations, and each normalization layer computes a ratio of linear functions followed by element-wise scaling, all of which are implementable in TC^0 [46]. Adding a constant number of such layers therefore preserves the TC^0 upper bound, and the remainder of the proof (the chain of reductions to the NC^1-complete word problem) applies without modification.

E.9.0.2 Constant number of chain-of-thought steps. We now argue that Theorem 5 extends to Transformers that perform any constant number $T = O(1)$ of autoregressive chain-of-thought steps before producing the final answer. Formally, define $\mathcal{T}_T$ analogously to $\mathcal{T}_0$, except that the model is permitted T autoregressive generation steps: at each step, the model appends a token to the sequence and conditions on the extended sequence for the next step.

A Transformer with T CoT steps can be simulated by a TC^0 circuit of size $\mathrm{poly}(n) \cdot |\Sigma|^T$, where $|\Sigma|$ is the vocabulary size. The simulation proceeds as follows: enumerate all $|\Sigma|^T$ possible sequences of T intermediate tokens. For each candidate sequence $(z_1, \ldots, z_T)$, construct a *verification subcircuit* that (i) evaluates the no-CoT Transformer on the input concatenated with $(z_1, \ldots, z_{t-1})$ for each $t = 1, \ldots, T$ *in parallel*, (ii) checks that the output at step t equals z_t under greedy decoding, and (iii) takes the AND of all T consistency checks to gate the final prediction. Since the T evaluations are independent (each operates on a fixed candidate prefix), they can be computed in parallel; each evaluation is a TC^0 circuit and each equality check is in AC^0, so the entire verification subcircuit has the same constant depth as a single no-CoT Transformer evaluation (plus a single AND gate). The overall circuit takes the OR over all $|\Sigma|^T$ branches; since exactly one branch is consistent with greedy decoding, this OR recovers the correct output.

The circuit has constant depth for any T (since all branches and all evaluations within each branch run in parallel). The constraint $T = O(1)$ is needed for the *size*: when $|\Sigma| = \mathrm{poly}(n)$, the total number of branches is $|\Sigma|^T = \mathrm{poly}(n)$ only if $T = O(1)$, ensuring polynomial circuit size and hence containment in TC^0. Therefore $\mathcal{T}_{O(1)} \subseteq \mathrm{TC}^0$, and the same chain of reductions used in the proof of Theorem 5 yields $F_{\mathrm{TwoChain}} \notin \mathcal{T}_{O(1)}$ under the assumption $\mathrm{TC}^0 \neq \mathrm{NC}^1$.

F Proof of Theorem 2

This section proves Theorem 2 and is organized as follows. In Appendix F.1, we show that the loss can be expressed as the same-side absorption probability of a Markov chain, which we call the canonical chain. Appendix F.2 introduces the key definitions (depth, long jumps) and two surrogate chains—the long-jump-absorbing chain and the no-long-jump chain—along with the source function representation that relates parameter derivatives to expected sources along trajectories.

Appendix F.3 and Appendix F.4 compute the source functions and derivative estimates for these two chains respectively. Appendix F.6 combines these estimates to establish that $\partial_\alpha S \geq 0$, $\partial_\beta S \leq 0$, and $\partial_\gamma S \leq 0$ for easy examples, implying monotonic parameter evolution under gradient flow. Appendix F.7 shows that hard examples contribute negligibly to the gradient. Finally, Appendix F.9 assembles these components to complete the proof.

We establish the derivative estimates at initialization, but they in fact hold throughout the learning process as long as the forward transition probability remains non-decreasing and the long-jump probability stays bounded away from zero. Since our analysis shows that both conditions are maintained for all $t \leq t_0$, these estimates in fact hold for all $t \leq t_0$.

Furthermore, under the assumptions of Theorem 2, the training distribution decomposes into *easy examples*—starting vertices within distance s of the terminal—and *hard examples*—starting vertices at distance at least $\lceil an \rceil$ from the terminal. We analyze the gradient contributions from easy and hard examples separately, showing that easy examples drive learning while hard examples contribute negligibly.

F.1 Loss can be written as absorption probability of a Markov chain

Lemma 68 implies that the loss from Section 3, under the assumptions of Section 5.2, is equivalent to the probability of same-side absorption in a Markov chain on the state space of two identical directed chains A and B of equal length n, with absorbing endpoints a_n, b_n, and nonterminal states $a_1, \ldots, a_{n-1}, b_1, \ldots, b_{n-1}$. We will therefore analyze this absorption probability throughout the subsequent sections, which we term the *canonical Markov chain*.

The transition probabilities of the canonical chain are determined by a logit structure with parameters $\alpha, \beta, \gamma \in \mathbb{R}$. Without loss of generality, we subtract $\gamma(t)$ from all logits (which does not affect the softmax probabilities).

If v is in the middle of the chain, we have that:

$$o(v, C)_k = \begin{cases} \alpha(t), & k \text{ is one ahead of } v \text{ in the chain} \\ \beta(t), & k \text{ is one back of } v \text{ in the chain} \\ 0, & k \text{ start of some chain (not necessary that with } v) \\ 0, & k \text{ end of some chain (not necessary that with } v) \\ \gamma(t), & \text{otherwise} \end{cases}$$

If v is at the penultimate vertex of the chain, we have that:

$$o(v, C)_k = \begin{cases} (\alpha(t) - \gamma(t)), & k \text{ is one ahead of } v \text{ in the chain} \\ \beta(t), & k \text{ is one back of } v \text{ in the chain} \\ 0, & k \text{ start of some chain (not necessary that with } v) \\ 0, & k \text{ is the end of chain without } v \\ \gamma(t), & \text{otherwise} \end{cases}$$

If v is at the second vertex of the chain, we have that:

$$o(v, C)_k = \begin{cases} \alpha(t), & k \text{ is one ahead of } v \text{ in the chain} \\ (\beta(t) - \gamma(t)), & k \text{ is one back of } v \text{ in the chain} \\ 0, & k \text{ start of some chain without } v \\ 0, & k \text{ end of some chain (not necessary that with } v) \\ \gamma(t), & \text{otherwise} \end{cases}$$

If v is at the first vertex of the chain, we have that:

$$o(v, C)_k = \begin{cases} \alpha(t), & k \text{ is one ahead of } v \text{ in the chain} \\ 0, & k \text{ start of some chain without } v \\ 0, & k \text{ is the end of some chain (not necessary that with } v) \\ \gamma(t), & \text{otherwise} \end{cases}$$

The transition probabilities are then given by softmax:

$$\pi(v \to k) = \frac{\exp(o(v, C)_k)}{\sum_{k'} \exp(o(v, C)_{k'})}.$$

F.2 Preliminaries

This subsection establishes the foundational definitions and tools for analyzing the canonical Markov chain. We begin by defining key quantities such as hitting probabilities, depth, and long jumps (Definitions 14 and 15). We then introduce two surrogate chains (Definitions 16 and 17) that decompose the canonical chain's behavior, enabling separate analysis of long-jump and local-step dynamics. Finally, we present the source function representation (Lemma 16) that expresses parameter derivatives in terms of expected sources along trajectories, which forms the basis for our derivative analysis in subsequent sections.

F.2.1 Useful definitions

Definition 13. For any state v (including absorbing states), let

$$h_A(v) := \Pr_v[X_\tau = a_n], \qquad h_B(v) := \Pr_v[X_\tau = b_n],$$

where τ is the (almost surely finite) hitting time of $\{a_n, b_n\}$.

For $v \in \{a_1, \ldots, a_{n-1}\}$ on chain A, the "same–side absorption probability" is $h_A(v)$; for $v \in \{b_1, \ldots, b_{n-1}\}$ on chain B, it is $h_B(v)$. Denote these by

$$S(v) := \begin{cases} h_A(v), & v \in \{a_1, \ldots, a_{n-1}\}, \\ h_B(v), & v \in \{b_1, \ldots, b_{n-1}\}. \end{cases}$$

Definition 14. Fix a chain length $n \geq 2$ and consider the state space

$$V := \{a_1, \ldots, a_n, b_1, \ldots, b_n\}.$$

Define the *depth* (distance to the terminal on the same chain) by the function

$$d : V \to \{0, 1, \ldots, n-1\}, \qquad d(a_i) := n - i, \quad d(b_i) := n - i.$$

Definition 15. A transition $x \to y$ is a *long jump* if the depth changes by 2 or more, i.e.

$$|d(y) - d(x)| \geq 2.$$

F.2.2 Surrogate chains

To analyze the canonical chain, we introduce two auxiliary Markov chains: the long-jump-absorbing chain and the no-long-jump chain. Lemma 15 below shows that the derivative of the same-side absorption probability can be decomposed into contributions from these two surrogate chains. This decomposition is fundamental to our analysis, as it separates the effect of long-range transitions from local transitions.

Definition 16. The *long–jump–absorbing chain* is the Markov chain obtained from the two–chain model by *redirecting every long jump* $x \to y$ to a single new absorbing state e: all non–long–jump transition probabilities are unchanged, and the total probability mass of long jumps out of x is assigned to e. The absorbing set is $T := \{a_n, b_n, e\}$.

Definition 17. The *no–long–jump (NL) chain* is the Markov chain obtained from the two–chain model by *disallowing* all long jumps: from any nonterminal state x, the only allowed transitions are to states $y \neq x$ with $|d(y) - d(x)| \leq 1$, and the transition probabilities are obtained from the original chain by conditioning on this. (As in the original chain, the terminal states a_n, b_n are absorbing).

Proposition 5. *Fix parameters (hence a fixed transition kernel) and a start state v. Let*

$$\tau := \inf\{t \geq 0 : \ X_t \in \{a_n, b_n\}\}, \qquad J := \inf\{t \geq 0 : \ (X_t \to X_{t+1}) \textit{ is a long jump}\},$$

and write $LJ := \{J \leq \tau\}$ for the event that a long jump occurs before absorption. Then

$$\Pr_v[X_\tau = a_n \mid LJ] = \tfrac{1}{2}.$$

Proof. Let φ be the chain-swap map on states: $\varphi(a_i) = b_i$ and $\varphi(b_i) = a_i$. For any state y, by chain-swap symmetry of the model we have

$$\Pr_y[X_\tau = a_n] + \Pr_{\varphi(y)}[X_\tau = a_n] = 1,$$

since swapping chains interchanges the absorbing events $\{X_\tau = a_n\}$ and $\{X_\tau = b_n\}$.

Now condition on the time and landing state of the first long jump. On LJ, let $Y := X_{J+1}$ be the landing state of the first long jump. By definition of a long jump and the logit structure, the distribution of Y is symmetric under φ (targets come in φ-paired states with equal one-step probability from any fixed source), so $\Pr_v[Y = y \mid LJ] = \Pr_v[Y = \varphi(y) \mid LJ]$ for every y. Therefore,

$$\begin{aligned}
\Pr_v[X_\tau = a_n \mid LJ] &= \sum_y \Pr_v[Y = y \mid LJ]\ \Pr_y[X_\tau = a_n] \\
&= \frac{1}{2}\sum_y \Pr_v[Y = y \mid LJ]\Big(\Pr_y[X_\tau = a_n] + \Pr_{\varphi(y)}[X_\tau = a_n]\Big) \\
&= \frac{1}{2}\sum_y \Pr_v[Y = y \mid LJ]\cdot 1 = \tfrac{1}{2},
\end{aligned}$$

as claimed. □

Lemma 15. *Fix $\theta \in \{\alpha, \beta, \gamma\}$ and a start state v. Let*

$$\tau := \inf\{t \geq 0 : X_t \in \{a_n, b_n\}\}, \qquad J := \inf\{t \geq 0 :\ (X_t \to X_{t+1}) \text{ is a long jump}\},$$

and define $NL := \{J > \tau\}$ and $LJ := \{J \leq \tau\}$. Set

$$\begin{aligned}
S_\theta(v) &:= \overset{\theta}{\Pr_v}[X_\tau = a_n], & h_\theta(v) &:= \overset{\theta}{\Pr_v}(LJ), \\
\bar{S}_\theta^{NL}(v) &:= \overset{\theta}{\Pr_v}(X_\tau = a_n \mid NL), & \bar{S}_\theta^{LJ}(v) &:= \overset{\theta}{\Pr_v}(X_\tau = a_n \mid LJ).
\end{aligned}$$

Then

$$S_\theta(v) = (1 - h_\theta(v))\bar{S}_\theta^{NL}(v) + h_\theta(v)\bar{S}_\theta^{LJ}(v),$$

and

$$\partial_\theta S_\theta(v) = (\partial_\theta h_\theta(v))\big(\bar{S}_\theta^{LJ}(v) - \bar{S}_\theta^{NL}(v)\big) + (1 - h_\theta(v))\,\partial_\theta \bar{S}_\theta^{NL}(v) + h_\theta(v)\,\partial_\theta \bar{S}_\theta^{LJ}(v).$$

By Proposition 5, $\bar{S}_\theta^{LJ}(v) = \frac{1}{2}$, hence

$$\begin{aligned}
S_\theta(v) &= \tfrac{1}{2} + (1 - h_\theta(v))\Big(\bar{S}_\theta^{NL}(v) - \tfrac{1}{2}\Big), \\
\partial_\theta S_\theta(v) &= -(\partial_\theta h_\theta(v))\Big(\bar{S}_\theta^{NL}(v) - \tfrac{1}{2}\Big) + (1 - h_\theta(v))\,\partial_\theta \bar{S}_\theta^{NL}(v).
\end{aligned}$$

Proof. The argument is identical for $\theta \in \{\alpha, \beta, \gamma\}$, so we fix an arbitrary θ throughout.

Since $\{NL, LJ\}$ is a partition of the sample space,

$$S_\theta(v) = \overset{\theta}{\Pr_v}(X_\tau = a_n, NL) + \overset{\theta}{\Pr_v}(X_\tau = a_n, LJ).$$

Writing $\Pr_v^\theta(X_\tau = a_n, NL) = \Pr_v^\theta(NL)\Pr_v^\theta(X_\tau = a_n \mid NL) = (1 - h_\theta(v))\bar{S}_\theta^{NL}(v)$ and $\Pr_v^\theta(X_\tau = a_n, LJ) = \Pr_v^\theta(LJ)\Pr_v^\theta(X_\tau = a_n \mid LJ) = h_\theta(v)\bar{S}_\theta^{LJ}(v)$ yields

$$S_\theta(v) = (1 - h_\theta(v))\bar{S}_\theta^{NL}(v) + h_\theta(v)\bar{S}_\theta^{LJ}(v).$$

Differentiate the preceding identity with respect to θ and apply the product rule:

$$\begin{aligned}
\partial_\theta S_\theta(v) &= -(\partial_\theta h_\theta(v))\bar{S}_\theta^{NL}(v) + (1 - h_\theta(v))\partial_\theta \bar{S}_\theta^{NL}(v) + (\partial_\theta h_\theta(v))\bar{S}_\theta^{LJ}(v) + h_\theta(v)\partial_\theta \bar{S}_\theta^{LJ}(v) \\
&= (\partial_\theta h_\theta(v))\big(\bar{S}_\theta^{LJ}(v) - \bar{S}_\theta^{NL}(v)\big) + (1 - h_\theta(v))\partial_\theta \bar{S}_\theta^{NL}(v) + h_\theta(v)\,\partial_\theta \bar{S}_\theta^{LJ}(v).
\end{aligned}$$

By Proposition 5, $\bar{S}_\theta^{LJ}(v) = \frac{1}{2}$; substituting this constant into the value identity and its derivative identity yields the displayed simplifications. □

F.2.3 Source function of surrogate chains and derivative form

To analyze how the absorption probabilities change under gradient flow, we require a tractable representation of their parameter derivatives $\partial_\theta u(v)$ where $\theta \in \{\alpha, \beta, \gamma\}$ and $u(v)$ is an absorption probability of interest. Direct computation of these derivatives is challenging due to the complex dependency of multi-step trajectories on the parameters. The source function presented below is advantageous because sources depend only on single-step transition sensitivities and can be computed explicitly from our logit structure, enabling the quantitative bounds developed in Appendices F.3 and F.4.

Lemma 16. *Let $(X_t)_{t\geq 0}$ be a finite Markov chain with transition matrix P_θ smoothly parameterized by $\theta \in \mathbb{R}$. Let T be a nonempty absorbing set and $\tau := \inf\{t \geq 0 : X_t \in T\}$. Fix a target state $t^\star \in T$ and define the absorption probability*

$$u_\theta(x) := \overset{\theta}{\Pr_x}[X_\tau = t^\star], \qquad x \in \mathcal{S},$$

so $u_\theta(t^\star) = 1$ and $u_\theta(t) = 0$ for $t \in T \setminus \{t^\star\}$. Then for every nonterminal state $x \notin T$, u_θ is P_θ–harmonic:

$$u_\theta(x) = \sum_{y\in\mathcal{S}} P_\theta(x, y)\, u_\theta(y).$$

Define the source *for $\partial_\theta u_\theta$ by*

$$s_\theta(x) := \sum_{y\in\mathcal{S}} \big(\partial_\theta P_\theta(x, y)\big)\, u_\theta(y), \qquad x \notin T.$$

Then $g_\theta := \partial_\theta u_\theta$ satisfies, for all $x \notin T$,

$$g_\theta(x) = \sum_y P_\theta(x, y) g_\theta(y) + s_\theta(x), \qquad g_\theta(t) = 0, \;\; t \in T,$$

and admits the trajectory representation

$$\partial_\theta u_\theta(x) = \mathbb{E}^\theta_x\left[\sum_{t=0}^{\tau-1} s_\theta(X_t)\right], \qquad x \in \mathcal{S}.$$

Proof. Let $\mathcal{S}$ be the finite state space and let $T \subset \mathcal{S}$ be the absorbing set. Write $U := \mathcal{S} \setminus T$ for the transient states. Reorder states so that transient states come first, and write the transition matrix in block form

$$P_\theta = \begin{pmatrix} Q_\theta & R_\theta \\ 0 & I \end{pmatrix},$$

where Q_θ is the $|U| \times |U|$ transient-to-transient submatrix. Fix a target $t^\star \in T$ and define the absorption probability

$$u_\theta(x) := \overset{\theta}{\Pr_x}[X_\tau = t^\star], \qquad \tau := \inf\{t \geq 0 : X_t \in T\}.$$

Then $u_\theta(t^\star) = 1$ and $u_\theta(t) = 0$ for $t \in T \setminus \{t^\star\}$, and for $x \in U$, the harmonic equations read

$$u_\theta(x) = \sum_{y\in U} Q_\theta(x, y) u_\theta(y) + \sum_{t\in T} R_\theta(x, t) u_\theta(t).$$

Equivalently, in vector form on U,

$$u_\theta = Q_\theta u_\theta + r_\theta, \qquad r_\theta(x) := \sum_{t\in T} R_\theta(x, t) u_\theta(t) = R_\theta(x, t^\star).$$

Hence

$$(I - Q_\theta) u_\theta = r_\theta. \tag{22}$$

Differentiate (22) with respect to θ:

$$-(\partial_\theta Q_\theta) u_\theta + (I - Q_\theta)\, \partial_\theta u_\theta = \partial_\theta r_\theta.$$

Let $g_\theta := \partial_\theta u_\theta$. Rearranging gives the linear system on U,

$$(I - Q_\theta) g_\theta = s_\theta, \qquad s_\theta := \partial_\theta r_\theta + (\partial_\theta Q_\theta) u_\theta. \tag{23}$$

Noting that u_θ is constant on T, we also have the boundary condition $g_\theta|_T \equiv 0$.

Because the chain is absorbing, the spectral radius satisfies $\rho(Q_\theta) < 1$, hence $(I - Q_\theta)$ is invertible and

$$(I - Q_\theta)^{-1} = \sum_{k=0}^{\infty} Q_\theta^k \quad \text{(absolutely convergent).}$$

Therefore from (23),

$$g_\theta = (I - Q_\theta)^{-1} s_\theta = \sum_{k=0}^{\infty} Q_\theta^k s_\theta. \tag{24}$$

For $x \in U$, the vector $(Q_\theta^k s_\theta)(x)$ equals the expected source at time k before absorption:

$$(Q_\theta^k s_\theta)(x) = \sum_{y \in U} Q_\theta^k(x, y)\, s_\theta(y) = \mathcal{E}_x^\theta\big[s_\theta(X_k)\, \mathbf{1}\{k < \tau\}\big].$$

Summing over k and justifying the exchange of sum and expectation by dominated convergence (since $|s_\theta|$ is bounded on the finite state space and $\mathcal{E}_x^\theta[\tau] < \infty$ for every transient x in a finite absorbing chain) gives

$$g_\theta(x) = \sum_{k=0}^{\infty} \mathcal{E}_x^\theta\big[s_\theta(X_k)\, \mathbf{1}\{k < \tau\}\big] = \mathcal{E}_x^\theta\left[\sum_{k=0}^{\tau-1} s_\theta(X_k)\right].$$

Since $g_\theta = \partial_\theta u_\theta$ and $g_\theta(t) = 0$ for $t \in T$, this holds for all $x \in \mathcal{S}$ (trivially on T). This is the desired representation. □

F.3 Analysis of long-jump-absorbing chain

In this subsection, we analyze the long-jump-absorbing Markov chain defined in Definition 16. Our goal is to compute the source functions at each transient state x, which by Lemma 16 determine the derivatives of the long-jump absorption probability. We establish in Lemma 17 that these sources exhibit symmetry between the two chains, and then derive explicit formulas and bounds for the α, β, and γ sources in subsequent lemmas. These results feed into Appendix F.3.1 where we obtain quantitative derivative estimates, and ultimately into Appendix F.6 where we show monotonicity of the loss as a function of the parameters.

Lemma 17. *Work in the two–chain model and in the long–jump–absorbing chain of Definition 16. Let*

$$T := \{a_n, b_n, e\}, \qquad \tau := \inf\{t \geq 0 : X_t \in T\}, \qquad u_\theta(x) := \overset{\theta}{\Pr_x}[X_\tau = e].$$

For $\theta \in \{\alpha, \beta, \gamma\}$ define the local source on transient states

$$s_\theta(x) := \sum_y \big(\partial_\theta P_\theta(x, y)\big)\, u_\theta(y), \qquad x \notin T,$$

where P_θ is the transition matrix of the long–jump–absorbing chain.

Then for every $i \in \{1, \ldots, n-1\}$ and every $\theta \in \{\alpha, \beta, \gamma\}$,

$$s_\theta(a_i) = s_\theta(b_i).$$

Proof. Fix θ and let φ be the chain-swap involution: $\varphi(a_i) = b_i$, $\varphi(b_i) = a_i$ for $i \leq n$, and $\varphi(e) = e$. By symmetry of the logits, the long–jump redirection, and the softmax, the kernel P_θ is φ-invariant:

$$P_\theta(\varphi(x), \varphi(y)) = P_\theta(x, y) \qquad (\forall x, y).$$

Hence $\partial_\theta P_\theta(\varphi(x), \varphi(y)) = \partial_\theta P_\theta(x, y)$, and since the definition of e is symmetric with respect to the two chains, also $u_\theta(\varphi(x)) = u_\theta(x)$. Therefore, for any transient $x \notin T$,

$$s_\theta(\varphi(x)) = \sum_y \partial_\theta P_\theta(\varphi(x), y)\, u_\theta(y) = \sum_y \partial_\theta P_\theta(x, y)\, u_\theta(y) = s_\theta(x),$$

where we reindexed $y \mapsto \varphi(y)$ and used φ-invariance of $\partial_\theta P_\theta$ and u_θ. □

Lemma 18. *Work in the long–jump–absorbing chain (Definition 16) with terminal index n (so $T = \{a_n, b_n, e\}$ is absorbing) and depth function $d(\cdot)$ as in Definition 14. Let $\tau := \inf\{t \geq 0 : X_t \in T\}$ and define $u(v) := \Pr_v[X_\tau = e]$. By chain–swap symmetry $u(a_i) = u(b_i)$; write $u_i := u(a_i) = u(b_i)$, so $u_n = 0$.*

For $i = 1, \ldots, n-1$ define the depth-grouped probabilities (aggregating by the effect on depth):

$$q_i := \Pr(X_{t+1} = e \mid X_t \in \{a_i, b_i\}), \quad f_i := \Pr(d(X_{t+1}) = d(X_t) - 1 \mid X_t \in \{a_i, b_i\}),$$

$$b_i := \Pr(d(X_{t+1}) = d(X_t) + 1 \mid X_t \in \{a_i, b_i\}), \quad r_i := \Pr(d(X_{t+1}) = d(X_t) \mid X_t \in \{a_i, b_i\}),$$

where the probabilities aggregate all allowed transitions by their effect on depth (ignoring which chain the next state lies in). These satisfy $q_i + f_i + b_i + r_i = 1$, with the boundary convention $b_1 := 0$, and $f_i > 0$.

Define forward differences $d_i := u_i - u_{i+1}$ for $i = 1, \ldots, n-1$, and set $d_0 := 0$. Then $d_i \geq 0$ for all i, hence

$$u_1 \geq u_2 \geq \cdots \geq u_n = 0.$$

Proof. Fix $i \in \{1, \ldots, n-1\}$. Since u is harmonic on transient states and depends only on depth by symmetry, the harmonic equation at depth i can be written using the depth–grouped probabilities:

$$u_i = q_i \cdot u(e) + f_i u_{i+1} + b_i u_{i-1} + r_i u_i = q_i + f_i u_{i+1} + b_i u_{i-1} + r_i u_i, \tag{25}$$

with $u_n = 0$ and $b_1 = 0$. Rearranging gives

$$(1 - r_i)u_i = q_i + f_i u_{i+1} + b_i u_{i-1}.$$

Substitute $u_{i+1} = u_i - d_i$ and $u_{i-1} = u_i + d_{i-1}$ to obtain

$$(1 - r_i)u_i = q_i + f_i(u_i - d_i) + b_i(u_i + d_{i-1}) = q_i + (f_i + b_i)u_i - f_i d_i + b_i d_{i-1}.$$

Using $1 - r_i = q_i + f_i + b_i$, cancel $(f_i + b_i)u_i$ from both sides, yielding

$$q_i u_i = q_i - f_i d_i + b_i d_{i-1},$$

and hence

$$f_i d_i = b_i d_{i-1} + q_i(1 - u_i). \tag{26}$$

Since $f_i > 0$, $b_i \geq 0$, $q_i \geq 0$, and $1 - u_i \geq 0$, the identity (26) implies $d_i \geq 0$ whenever $d_{i-1} \geq 0$. For $i = 1$, we have $b_1 = 0$ and $d_0 = 0$, so

$$f_1 d_1 = q_1(1 - u_1) \geq 0 \quad \Rightarrow \quad d_1 \geq 0.$$

Inductively, $d_i \geq 0$ for all i, hence $u_i \geq u_{i+1}$ for all i, proving monotonicity. □

Corollary 2. *Under the assumptions of Lemma 18, for every $i = 1, \ldots, n-1$,*

$$d_i \;\geq\; \frac{q_i}{f_i}\,(1 - u_i).$$

Moreover, since the depth at state a_i is $n - i$, reaching the terminal requires $n - i$ consecutive forward steps (each decreasing depth by 1), and

$$1 - u_i \;\geq\; \Pr((n-i) \text{ consecutive forward steps starting from state } a_i),$$

hence

$$d_i \;\geq\; \frac{q_i}{f_i}\,\Pr((n-i) \text{ consecutive forward steps starting from state } a_i).$$

In particular, if $f_k \geq f_{\min} > 0$ for all $k \in \{i, i+1, \ldots, n-1\}$, then

$$\Pr((n-i) \text{ consecutive forward steps starting from state } a_i) \;\geq\; \prod_{j=0}^{n-i-1} f_{i+j} \;\geq\; (f_{\min})^{n-i},$$

and therefore

$$d_i \;\geq\; \frac{q_i}{f_i} \prod_{j=0}^{n-i-1} f_{i+j} \;\geq\; \frac{q_i}{f_i}(f_{\min})^{n-i}.$$

Proof. From (26) and $b_i d_{i-1} \geq 0$ we get $f_i d_i \geq q_i(1-u_i)$, proving $d_i \geq \frac{q_i}{f_i}(1-u_i)$.

Next, $1-u_i = \Pr(\tau \neq \tau_e) = \Pr(\tau_e > \tau)$ is the probability of not hitting e before absorption. Since the depth at state a_i is $n-i$, reaching the terminal a_n (depth 0) requires exactly $n-i$ consecutive forward steps. The event that the chain makes $n-i$ consecutive forward steps starting from a_i implies that e is not hit during those steps, hence is contained in $\{\tau_e > \tau\}$, giving $1-u_i \geq \Pr((n-i)$ consecutive forward steps from $a_i)$. The remaining inequalities follow by multiplying the corresponding one–step lower bounds and using $f_k \geq f_{\min}$ on the indicated range $k \in \{i, \dots, n-1\}$. □

Lemma 19. *For every start state $x \notin T$,*

$$u(x) \;\leq\; p(e)_{\max}\, \mathcal{E}_x[\tau].$$

In particular, whenever $\mathcal{E}_x[\tau] \leq D(x)/\zeta$ (e.g. by applying Lemma 64 to the depth process under a drift gap $\zeta > 0$),

$$u(x) \;\leq\; \frac{D(x)}{\zeta}\, p(e)_{\max}.$$

Proof. Let $\mathcal{F}_t := \sigma(X_0, \dots, X_t)$ and write

$$N_e := \sum_{t=0}^{\tau-1} \mathbf{1}\{X_{t+1} = e\}.$$

Since e is absorbing and $e \in T$, the event $\{X_\tau = e\}$ occurs iff the chain makes a transition to e before absorption, so $u(x) = \Pr_x[X_\tau = e] = \mathcal{E}_x[N_e]$. By the tower property and Markov property,

$$\begin{aligned}
\mathcal{E}_x[N_e] &:= \mathcal{E}_x\left[\sum_{t=0}^{\tau-1} \mathcal{E}_x[\mathbf{1}\{X_{t+1} = e\} \mid \mathcal{F}_t]\right] \\
&= \mathcal{E}_x\left[\sum_{t=0}^{\tau-1} \widetilde{P}(X_t, e)\right] = \mathcal{E}_x\left[\sum_{t=0}^{\tau-1} p(e; X_t)\right] \\
&\leq p(e)_{\max}\, \mathcal{E}_x[\tau],
\end{aligned}$$

which proves the first display. The final inequality follows by substituting the upper bound on $\mathcal{E}_x[\tau]$ from Lemma 64. □

Lemma 20. *Let $\widetilde{P}$ be the transition kernel of the long–jump–absorbing chain and let $u(x) := \Pr_x[X_\tau = e]$ denote the absorption probability at e (so $u(e) = 1$ and $u(a_n) = u(b_n) = 0$). Let $F(x) \in S(x)$ denote the forward neighbor when it exists and define*

$$p_f(x) := \widetilde{P}(x, F(x)) \quad (\text{set } p_f(x) := 0 \text{ if } F(x) \text{ does not exist}).$$

The α–source in the long–jump–absorbing chain satisfies for every transient $x \notin T$,

$$\boxed{s_\alpha(x) = p_f(x)\big(u(F(x)) - u(x)\big),}$$

with the convention that $s_\alpha(x) = 0$ if $F(x)$ does not exist.

Proof. Fix a transient state $x \notin T$. Let $L(x)$ be the set of long–jump targets in the original chain. Since $\widetilde{P}(x, y) = P(x, y)$ for $y \in S(x)$ and $\widetilde{P}(x, e) = \sum_{z \in L(x)} P(x, z)$, we have

$$s_\alpha(x) = \sum_{y \in S(x)} (\partial_\alpha P(x, y))\, u(y) + \left(\sum_{z \in L(x)} \partial_\alpha P(x, z)\right) u(e),$$

and $u(e) = 1$. Because $P(x, \cdot)$ is a softmax over the logits $\{o(x)_y\}$ in the original chain and only the forward logit depends on α with derivative 1, Lemma 62 gives, for each original target y,

$$\partial_\alpha P(x, y) = P(x, y)\big(\mathbf{1}\{y = F(x)\} - P(x, F(x))\big) = P(x, y)\big(\mathbf{1}\{y = F(x)\} - p_f(x)\big).$$

Introduce the auxiliary successor value

$$w_x(y) := \begin{cases} u(y), & y \in S(x), \\ 1, & y \in L(x). \end{cases}$$

Then the definition of $s_\alpha(x)$ can be rewritten as

$$s_\alpha(x) = \sum_y (\partial_\alpha P(x,y))\, w_x(y),$$

where the sum ranges over all original targets y from x. Substituting the softmax derivative yields

$$\begin{aligned} s_\alpha(x) &:= \sum_y P(x,y)\big(\mathbf{1}\{y = F(x)\} - p_f(x)\big) w_x(y) \\ &= P(x,F(x)) w_x(F(x)) - p_f(x) \sum_y P(x,y) w_x(y). \end{aligned}$$

Since $w_x(F(x)) = u(F(x))$ and $P(x,F(x)) = p_f(x)$, this becomes

$$s_\alpha(x) = p_f(x)\Big(u(F(x)) - \sum_y P(x,y) w_x(y)\Big).$$

Finally, harmonicity of u for the long–jump–absorbing chain gives the aggregated identity

$$\begin{aligned} u(x) &= \sum_{y \in S(x)} \widetilde{P}(x,y) u(y) + \widetilde{P}(x,e) u(e) \\ &= \sum_{y \in S(x)} P(x,y) u(y) + \sum_{z \in L(x)} P(x,z) \cdot 1 \\ &= \sum_y P(x,y) w_x(y), \end{aligned}$$

so $\sum_y P(x,y) w_x(y) = u(x)$, and hence

$$s_\alpha(x) = p_f(x)\big(u(F(x)) - u(x)\big).$$

If $F(x)$ does not exist then $\partial_\alpha o(x)_y \equiv 0$, hence $\partial_\alpha P(x,y) = 0$ for all y, which implies $\partial_\alpha \widetilde{P}(x,y) = 0$ for all $y \in S(x) \cup \{e\}$ and thus $s_\alpha(x) = 0$, consistent with the stated convention. □

Lemma 21. *Let $\widetilde{P}$ be the transition kernel of the long–jump–absorbing chain and let $u(x) := \Pr_x[X_\tau = e]$ (so $u(e) = 1$ and $u(a_n) = u(b_n) = 0$). Let $B(x) \in S(x)$ denote the backward neighbor when it exists and define*

$$p_b(x) := \widetilde{P}(x, B(x)) \quad (\text{set } p_b(x) := 0 \text{ if } B(x) \text{ does not exist}).$$

The β–source in the long–jump–absorbing chain satisfies for every transient $x \notin T$,

$$\boxed{s_\beta(x) = p_b(x)\big(u(B(x)) - u(x)\big),}$$

with the convention that $s_\beta(x) = 0$ if $B(x)$ does not exist.

Proof. Fix a transient state $x \notin T$. Let $L(x)$ be the set of long–jump targets in the original chain. Since $\widetilde{P}(x,y) = P(x,y)$ for $y \in S(x)$ and $\widetilde{P}(x,e) = \sum_{z \in L(x)} P(x,z)$, we have

$$s_\beta(x) = \sum_{y \in S(x)} (\partial_\beta P(x,y))\, u(y) + \left(\sum_{z \in L(x)} \partial_\beta P(x,z) \right) u(e),$$

and $u(e) = 1$. Because $P(x,\cdot)$ is a softmax over the logits $\{o(x)_y\}$ in the original chain and only the backward logit depends on β with derivative 1, Lemma 62 gives, for each original target y,

$$\partial_\beta P(x,y) = P(x,y)\big(\mathbf{1}\{y = B(x)\} - P(x,B(x))\big) = P(x,y)\big(\mathbf{1}\{y = B(x)\} - p_b(x)\big).$$

Introduce the auxiliary successor value

$$w_x(y) := \begin{cases} u(y), & y \in S(x), \\ 1, & y \in L(x). \end{cases}$$

Then the definition of $s_\beta(x)$ can be rewritten as

$$s_\beta(x) = \sum_y (\partial_\beta P(x,y))\, w_x(y),$$

where the sum ranges over all original targets y from x. Substituting the softmax derivative yields

$$\begin{aligned} s_\beta(x) &:= \sum_y P(x,y)\big(\mathbf{1}\{y = B(x)\} - p_b(x)\big) w_x(y) \\ &= P(x,B(x)) w_x(B(x)) - p_b(x) \sum_y P(x,y) w_x(y). \end{aligned}$$

Since $w_x(B(x)) = u(B(x))$ and $P(x,B(x)) = p_b(x)$, this becomes

$$s_\beta(x) = p_b(x)\Big(u(B(x)) - \sum_y P(x,y) w_x(y)\Big).$$

Finally, harmonicity of u for the long–jump–absorbing chain gives the aggregated identity

$$\begin{aligned} u(x) &= \sum_{y \in S(x)} \widetilde{P}(x,y) u(y) + \widetilde{P}(x,e) u(e) \\ &= \sum_{y \in S(x)} P(x,y) u(y) + \sum_{z \in L(x)} P(x,z) \cdot 1 \\ &= \sum_y P(x,y) w_x(y), \end{aligned}$$

so $\sum_y P(x,y) w_x(y) = u(x)$, and hence

$$s_\beta(x) = p_b(x)\big(u(B(x)) - u(x)\big).$$

If $B(x)$ does not exist then $\partial_\beta o(x)_y \equiv 0$, hence $\partial_\beta P(x,y) = 0$ for all y, which implies $\partial_\beta \widetilde{P}(x,y) = 0$ for all $y \in S(x) \cup \{e\}$ and thus $s_\beta(x) = 0$, consistent with the stated convention. □

Lemma 22. *Work in the long–jump–absorbing chain (Definition 16) with terminal index n (so $T := \{a_n, b_n, e\}$ is absorbing) and assume self–transitions are disallowed. Fix parameters α, β, γ. Let*

$$u(v) := \Pr_v[X_\tau = e], \qquad \tau := \inf\{t \geq 0 : \ X_t \in T\},$$

so $u(e) = 1$ and $u(a_n) = u(b_n) = 0$. By chain–swap symmetry, $u(a_i) = u(b_i)$; write $u_i := u(a_i) = u(b_i)$ and $u_n = 0$.

For a transient state $x \notin T$, define the local γ–source

$$s_\gamma(x) := \sum_z (\partial_\gamma \widetilde{P}(x,z))\, u(z),$$

where $\widetilde{P}$ is the transition kernel of the long–jump–absorbing chain. Then the following explicit formulas hold.

(A) Deep middle $x = a_i$, $2 < i < n-1$. *Let*

$$Z_{\mathrm{mid}} := e^\alpha + e^\beta + 4 + (2n-7)e^\gamma.$$

Then

$$s_\gamma(a_i) = \frac{e^\gamma}{Z_{\mathrm{mid}}}\Big((2n-10)(1-u_i) \ + \ u_{i-1} + u_{i+1} - 2u_i\Big).$$

(B) First vertex $x = a_1$. *Let*

$$Z_1 := e^{\alpha} + 3 + (2n-5)e^{\gamma}.$$

Then

$$s_\gamma(a_1) = \frac{e^{\gamma}}{Z_1}\Big((2n-6)(1-u_1) \;+\; (u_2 - u_1)\Big).$$

(C) Second vertex $x = a_2$. *Let*

$$Z_2 := e^{\alpha} + e^{(\beta-\gamma)} + 3 + (2n-6)e^{\gamma}.$$

Then

$$s_\gamma(a_2) = \frac{(2n-8)e^{\gamma}}{Z_2}(1-u_2) + \frac{e^{\gamma}}{Z_2}(u_3 - u_2) + \frac{e^{(\beta-\gamma)}}{Z_2}(u_2 - u_1).$$

(D) Penultimate vertex $x = a_{n-1}$. *Let*

$$Z_{n-1} := e^{(\alpha-\gamma)} + e^{\beta} + 3 + (2n-6)e^{\gamma}.$$

Then

$$s_\gamma(a_{n-1}) = \frac{(2n-8)e^{\gamma}}{Z_{n-1}}(1-u_{n-1}) + \frac{e^{\gamma}}{Z_{n-1}}(u_{n-2} - u_{n-1}) + \frac{e^{(\alpha-\gamma)}}{Z_{n-1}}\,u_{n-1}.$$

Proof. We argue as in Lemmas 20 and 21.

Fix a transient state $x \notin T$. Let $L(x)$ be the set of long–jump targets in the original chain. Since $\widetilde{P}(x,y) = P(x,y)$ for $y \in S(x)$ and $\widetilde{P}(x,e) = \sum_{z\in L(x)} P(x,z)$, we have

$$s_\gamma(x) = \sum_{y\in S(x)} (\partial_\gamma P(x,y))\,u(y) + \left(\sum_{z\in L(x)} \partial_\gamma P(x,z)\right) u(e),$$

and $u(e) = 1$. Introduce the auxiliary successor value

$$w_x(y) := \begin{cases} u(y), & y \in S(x), \\ 1, & y \in L(x). \end{cases}$$

Then the definition of $s_\gamma(x)$ can be rewritten as

$$s_\gamma(x) = \sum_y (\partial_\gamma P(x,y))\,w_x(y), \tag{27}$$

where the sum ranges over all original targets y from x. Harmonicity of u for the long–jump–absorbing chain gives $u(x) = \sum_y P(x,y)\,w_x(y)$.

Because $P(x,\cdot)$ is a softmax over the logits $\{o(x)_y\}$ in the original chain, Lemma 62 gives, for each original target y,

$$\partial_\gamma P(x,y) = P(x,y)\big(\partial_\gamma o(x,y) - \sum_z P(x,z)\partial_\gamma o(x,z)\big).$$

In each of the four cases below, $\partial_\gamma o(x,y)$ is constant on the γ–dependent classes of targets, so (27) reduces to a finite sum over those classes. We now enumerate the γ–dependent targets and whether they are short/long in each case; all remaining targets have $\partial_\gamma o(x,y) = 0$ and contribute nothing to $s_\gamma(x)$.

(A) Deep middle $x = a_i$, $2 < i < n-1$. The γ–dependent logits are exactly the "otherwise" targets, which have logit γ, hence $\partial_\gamma o = 1$. There are $2n-7$ such targets: the three short switches b_{i-1}, b_i, b_{i+1} and $2n-10$ others. Among these, b_{i-1} and b_{i+1} are short with values u_{i-1}, u_{i+1}, while b_i is short with value u_i (hence contributes zero after centering), and the remaining $2n-10$ are long jumps (value 1). All these have probability $e^{\gamma}/Z_{\mathrm{mid}}$, where $Z_{\mathrm{mid}} = e^{\alpha} + e^{\beta} + 4 + (2n-7)e^{\gamma}$. Substituting into (27) yields the stated formula for $s_\gamma(a_i)$.

(B) First vertex $x = a_1$. The unique α–target is a_2 (logit α); the special zero–logit targets are b_1, a_n, b_n; all remaining $2n-5$ targets have logit γ and thus $\partial_\gamma o = 1$. Among the γ–targets, the only short one is b_2, with value u_2; the other $2n-6$ γ–targets are long jumps (value 1). All γ–targets have probability e^γ/Z_1 with $Z_1 = e^\alpha + 3 + (2n-5)e^\gamma$. Plugging into (27) gives the stated expression for $s_\gamma(a_1)$.

(C) Second vertex $x = a_2$. Here a_3 has logit α, a_1 has logit $(\beta-\gamma)$ so $\partial_\gamma o(a_2, a_1) = -1$, the special zero–logit targets are b_1, a_n, b_n, and all remaining $2n-6$ targets have logit γ and $\partial_\gamma o = 1$. Among these γ–targets, the short ones are b_2 (value u_2, hence cancels) and b_3 (value u_3); the remaining $2n-8$ γ–targets are long jumps (value 1). The softmax denominator is $Z_2 = e^\alpha + e^{(\beta-\gamma)} + 3 + (2n-6)e^\gamma$, so $P(a_2, a_1) = e^{(\beta-\gamma)}/Z_2$ and each γ–target has probability e^γ/Z_2. Substituting these contributions into (27) yields the stated formula for $s_\gamma(a_2)$.

(D) Penultimate vertex $x = a_{n-1}$. Here a_n has logit $(\alpha-\gamma)$ so $\partial_\gamma o(a_{n-1}, a_n) = -1$, a_{n-2} has logit β, the special zero–logit targets are a_1, b_1, b_n, and all remaining $2n-6$ targets have logit γ and $\partial_\gamma o = 1$. Among these γ–targets, the short ones are b_{n-2} (value u_{n-2}) and b_{n-1} (value u_{n-1}, hence cancels); the remaining $2n-8$ γ–targets are long jumps (value 1). The denominator is $Z_{n-1} = e^{(\alpha-\gamma)} + e^\beta + 3 + (2n-6)e^\gamma$, so $P(a_{n-1}, a_n) = e^{(\alpha-\gamma)}/Z_{n-1}$ and each γ–target has probability e^γ/Z_{n-1}. Plugging these into (27) gives the stated formula for $s_\gamma(a_{n-1})$. □

Lemma 23. *Assume the setting and notation of Lemma 22 and Lemma 18. Let τ be the absorption time into $T = \{a_n, b_n, e\}$. For a deep-middle index i with $2 < i < n-1$, define the deep-middle one-step absorption probability*

$$q_{\mathrm{mid}} := \widetilde{P}(a_i, e) = \frac{(2n-10)e^\gamma}{Z_{\mathrm{mid}}}.$$

Assume a forward-drift condition at the depths under consideration:

$$f_j \geq b_j \qquad \text{for all relevant depths } j. \tag{28}$$

Then there exist constants $n_0 \in \mathbb{N}$ and $C_\star \geq 1$, depending only on (α, β, γ), such that for all $n \geq n_0$ the following bounds hold.

(A) Deep middle ($x = a_j$, $2 < j < n-1$).

$$s_\gamma(a_j) \;\geq\; \frac{1}{2}\, q_j\,(1-u_j) \;>\; 0, \qquad q_j := \widetilde{P}(a_j, e).$$

In particular, since $q_j \geq C_\star^{-1} q_{\mathrm{mid}}$ for $n \geq n_0$,

$$s_\gamma(a_j) \;\geq\; \frac{1}{2C_\star}\, q_{\mathrm{mid}}\,(1-u_j) \;>\; 0.$$

(B) First vertex ($x = a_1$).

$$s_\gamma(a_1) \;\geq\; \frac{1}{2}\, q_1\,(1-u_1) \;>\; 0, \qquad q_1 := \widetilde{P}(a_1, e).$$

(C) Penultimate vertex ($x = a_{n-1}$).

$$s_\gamma(a_{n-1}) \;\geq\; q_{n-1}\,(1-u_{n-1}) \;\geq\; 0, \qquad q_{n-1} := \widetilde{P}(a_{n-1}, e).$$

(D) Second vertex ($x = a_2$). *Let $q_2 := \widetilde{P}(a_2, e)$. Then*

$$s_\gamma(a_2) \;\geq\; -(d_1 + d_2), \qquad d_1 := u_1 - u_2,\;\; d_2 := u_2 - u_3,$$

and moreover, for all $n \geq n_0$,

$$d_1 + d_2 \;\leq\; 3C_\star\, q_{\mathrm{mid}}\, \mathcal{E}_{a_2}[\tau], \qquad \textit{hence} \qquad s_\gamma(a_2) \;\geq\; -3C_\star\, q_{\mathrm{mid}}\, \mathcal{E}_{a_2}[\tau].$$

Moreover, for all $n \geq n_0$,

$$|s_\gamma(a_2)| \;\leq\; q_2 + (d_1 + d_2) \;\leq\; 4C_\star\, q_{\mathrm{mid}}\, \mathcal{E}_{a_2}[\tau],$$

using $q_2 \le C_\star q_{\mathrm{mid}}$ from (29) *and $\mathcal{E}_{a_2}[\tau] \ge 1$. By symmetry, the same bounds hold with a_2 replaced by b_2.*

Finally, under (28) *one has the crude polynomial bound*

$$\mathcal{E}_{a_2}[\tau] \;\le\; \mathrm{poly}(n),$$

and under the stronger drift condition $\inf_j(f_j - b_j) \ge \Delta > 0$ one has

$$\mathcal{E}_{a_2}[\tau] \;\le\; \frac{C}{\Delta}\, n$$

for a constant C depending only on the (logit-determined) depth-grouped probabilities.

Proof. We use the explicit formulas of Lemma 22 and the difference recurrence of Lemma 18.

Step 1: Comparability of one-step e-absorption probabilities. For each transient depth j, the quantity $q_j = \widetilde{P}(a_j, e)$ equals the total probability of the long-jump class, whose numerator is $(2n + O(1))e^\gamma$ and whose denominator equals $(2n + O(1))e^\gamma + O(1)$, where the $O(1)$ terms depend only on (α, β, γ) and come from the finitely many non-γ logits and the finitely many special zero-logit targets. Consequently, there exist n_0 and $C_\star \ge 1$ depending only on (α, β, γ) such that for all $n \ge n_0$,

$$C_\star^{-1} q_{\mathrm{mid}} \;\le\; q_j \;\le\; C_\star q_{\mathrm{mid}} \qquad \text{for all transient } j. \tag{29}$$

Step 2: Positive lower bound in the deep middle. Fix j with $2 < j < n-1$. Lemma 22 gives

$$s_\gamma(a_j) = \frac{e^\gamma}{Z_{\mathrm{mid}}}\Big((2n-10)(1-u_j)+u_{j-1}+u_{j+1}-2u_j\Big) = \frac{e^\gamma}{Z_{\mathrm{mid}}}\Big((2n-10)(1-u_j)+d_{j-1}-d_j\Big).$$

By the recurrence (26) at depth j,

$$d_j = \frac{b_j}{f_j} d_{j-1} + \frac{q_j}{f_j}(1-u_j),$$

hence

$$d_{j-1} - d_j = d_{j-1}\Big(1 - \frac{b_j}{f_j}\Big) - \frac{q_j}{f_j}(1-u_j) \;\ge\; -\frac{q_j}{f_j}(1-u_j),$$

where we used $f_j \ge b_j$ so that $1 - \frac{b_j}{f_j} \ge 0$. Therefore

$$(2n-10)(1-u_j) + d_{j-1} - d_j \;\ge\; \Big((2n-10) - \frac{q_j}{f_j}\Big)(1-u_j).$$

Using $\frac{e^\gamma}{Z_{\mathrm{mid}}} = \frac{q_j}{2n-10}$ yields

$$s_\gamma(a_j) \;\ge\; q_j(1-u_j)\Big(1 - \frac{q_j}{f_j(2n-10)}\Big).$$

For all n large enough (depending only on fixed parameters through f_j and the bound $q_j \le 1$), the parenthesis is at least $1/2$, giving $s_\gamma(a_j) \ge \frac{1}{2} q_j(1-u_j) > 0$. The refinement in terms of q_{mid} follows from (29).

Step 3: Positive lower bound at the first vertex. Lemma 22 gives

$$s_\gamma(a_1) = \frac{e^\gamma}{Z_1}\Big((2n-6)(1-u_1) + (u_2 - u_1)\Big) = \frac{e^\gamma}{Z_1}\Big((2n-6)(1-u_1) - d_1\Big).$$

At $j = 1$, the recurrence (26) with $b_1 = 0$ gives $f_1 d_1 = q_1(1-u_1)$, hence $d_1 = \frac{q_1}{f_1}(1-u_1)$. Thus

$$s_\gamma(a_1) = \frac{e^\gamma}{Z_1}\Big((2n-6) - \frac{q_1}{f_1}\Big)(1-u_1) = q_1(1-u_1)\Big(1 - \frac{q_1}{f_1(2n-6)}\Big),$$

where we used $\frac{e^\gamma}{Z_1} = \frac{q_1}{2n-6}$. For n large enough the parenthesis is at least $1/2$, yielding $s_\gamma(a_1) \ge \frac{1}{2} q_1(1-u_1) > 0$.

Step 4: Nonnegativity at the penultimate vertex. Lemma 22 gives

$$s_\gamma(a_{n-1}) = \frac{(2n-8)e^\gamma}{Z_{n-1}}(1-u_{n-1}) + \frac{e^\gamma}{Z_{n-1}}(u_{n-2}-u_{n-1}) + \frac{e^{(\alpha-\gamma)}}{Z_{n-1}}u_{n-1}.$$

Each term in brackets is nonnegative ($1-u_{n-1} \geq 0$, $u_{n-2}-u_{n-1} = d_{n-2} \geq 0$, and $u_{n-1} \geq 0$), so dropping the last two terms yields $s_\gamma(a_{n-1}) \geq q_{n-1}(1-u_{n-1}) \geq 0$.

Step 5: Second-vertex negativity and control by $q_{\mathrm{mid}} \cdot \mathcal{E}[\tau]$. From Lemma 22,

$$s_\gamma(a_2) = q_2(1-u_2) - P(a_2,b_3)d_2 - P(a_2,a_1)d_1.$$

Since $q_2(1-u_2) \geq 0$ and the probabilities are at most 1, we obtain the crude lower bound

$$s_\gamma(a_2) \;\geq\; -(d_1+d_2).$$

Also, using $|q_2(1-u_2) - P(a_2,b_3)d_2 - P(a_2,a_1)d_1| \leq q_2(1-u_2) + d_1 + d_2 \leq q_2 + d_1 + d_2$, we have the crude absolute bound

$$|s_\gamma(a_2)| \;\leq\; q_2 + (d_1+d_2).$$

Next, $d_2 = u_2 - u_3 \leq u_2$ and $d_1 = u_1 - u_2 \leq u_1$, hence $d_1 + d_2 \leq u_1 + u_2$.

We now use the hazard identity for absorption at e in the long-jump-absorbing chain. Let $q(X_t) := \widetilde{P}(X_t, e)$. Since the chain can enter e at most once,

$$u(x) = \Pr_x(X_\tau = e) = \mathcal{E}_x\Big[\sum_{t=0}^{\tau-1} \mathbf{1}\{X_{t+1} = e\}\Big] = \mathcal{E}_x\Big[\sum_{t=0}^{\tau-1} q(X_t)\Big].$$

In particular,

$$u(x) \;\leq\; q_{\max}\,\mathcal{E}_x[\tau], \qquad q_{\max} := \sup_{y \notin T} \widetilde{P}(y,e).$$

By the comparability (29), $q_{\max} \leq C_\star q_{\mathrm{mid}}$ for $n \geq n_0$, hence

$$u(x) \;\leq\; C_\star q_{\mathrm{mid}}\,\mathcal{E}_x[\tau] \qquad (n \geq n_0).$$

Applying this to $x = a_2$ gives $u_2 \leq C_\star q_{\mathrm{mid}} \mathcal{E}_{a_2}[\tau]$. For u_1, observe that from a_1 in one step one either absorbs or moves to depth 2, so

$$\mathcal{E}_{a_1}[\tau] \leq 1 + \sup_{y \in \{a_2, b_2\}} \mathcal{E}_y[\tau] = 1 + \mathcal{E}_{a_2}[\tau] \leq 2\mathcal{E}_{a_2}[\tau],$$

using $\mathcal{E}_{a_2}[\tau] \geq 1$. Therefore

$$u_1 \leq C_\star q_{\mathrm{mid}} \mathcal{E}_{a_1}[\tau] \leq 2C_\star q_{\mathrm{mid}} \mathcal{E}_{a_2}[\tau].$$

Combining yields

$$d_1 + d_2 \leq u_1 + u_2 \leq 3C_\star q_{\mathrm{mid}} \mathcal{E}_{a_2}[\tau],$$

and hence

$$s_\gamma(a_2) \;\geq\; -3C_\star q_{\mathrm{mid}}\,\mathcal{E}_{a_2}[\tau].$$

Finally, the stated polynomial (or linear under strict drift) upper bounds on $\mathcal{E}_{a_2}[\tau]$ follow from standard biased (or nonnegatively drifted) one-dimensional random-walk estimates applied to the depth process under (28) (and are used only as coarse growth controls). □

F.3.1 Derivative estimates

In this subsubsection we derive quantitative bounds on the derivatives of the absorption probability at the long–jump state e in the long–jump–absorbing chain (Definition 16). We leverage the source computations from the preceding subsubsections with the source representation of the derivative (Lemma 16). Throughout, $h_\theta(v)$ denotes this absorption probability, i.e., $h_\theta(v) := \Pr_v[X_\tau = e]$.

Lemma 24. *In the* long–jump–absorbing chain*, let*

$$\tau := \inf\{t \geq 0 : \ X_t \in \{a_n, b_n, e\}\},$$
$$v_{\mathrm{pen}} := \begin{cases} a_{n-1}, & v \in \{a_1, \dots, a_{n-1}\}, \\ b_{n-1}, & v \in \{b_1, \dots, b_{n-1}\}, \end{cases}$$
$$\sigma := \inf\{t \geq 0 : \ X_t = v_{\mathrm{pen}}\}.$$

Then the derivative $\partial_\alpha h_\alpha(v)$ satisfies

$$\partial_\alpha h_\alpha(v) = \mathcal{E}_v\left[\sum_{t=0}^{\tau-1} s_\alpha(X_t)\right] \ \leq \ s_\alpha(v) \ + \ s_\alpha(v_{\mathrm{pen}}) \Pr_v(0 < \sigma < \tau),$$

where

$$s_\alpha(x) = p_f(x)\big(h_\alpha(F(x)) - h_\alpha(x)\big) \leq 0,$$

with the convention $s_\alpha(x) = 0$ if the (same–chain) forward neighbor $F(x)$ does not exist.

In particular, if v is k steps from the terminal on its chain, i.e. $v \in \{a_{n-k}, b_{n-k}\}$ with $2 \leq k \leq n-1$, then

$$\Pr_v(0 < \sigma < \tau) \ \geq \ \Pr_v\Big(X_i = F^i(v) \ \ \forall i = 1, \dots, k-1\Big),$$

and hence

$$\partial_\alpha h_\alpha(v) \ \leq \ s_\alpha(v) \ + \ s_\alpha(v_{\mathrm{pen}}) \Pr_v\Big(X_i = F^i(v) \ \ \forall i = 1, \dots, k-1\Big).$$

Proof. Let $u(\cdot) := h_\alpha(\cdot)$ be the absorption probability at e in the long–jump–absorbing chain, so $u(x) = \Pr_x[X_\tau = e]$ with $\tau := \inf\{t \geq 0 : X_t \in \{a_n, b_n, e\}\}$. Apply the source/trajectory representation (Lemma 16) with $\theta = \alpha$:

$$\partial_\alpha h_\alpha(v) = \partial_\alpha u(v) = \mathcal{E}_v\left[\sum_{t=0}^{\tau-1} s_\alpha(X_t)\right],$$

where $s_\alpha(x) := \sum_y (\partial_\alpha P(x,y))\, u(y)$ is the local α–source.

By the explicit α–source identity in the long–jump–absorbing chain (Lemma 20),

$$s_\alpha(x) = p_f(x)\big(u(F(x)) - u(x)\big),$$

with the convention $s_\alpha(x) = 0$ if $F(x)$ does not exist. By monotonicity of u along depth (Lemma 18), we have $u(F(x)) \leq u(x)$ whenever $F(x)$ exists, hence $s_\alpha(x) \leq 0$ for all transient x. Therefore

$$\begin{aligned} \partial_\alpha h_\alpha(v) &= \mathcal{E}_v\left[\sum_{t=0}^{\tau-1} s_\alpha(X_t)\right] \\ &\leq \mathcal{E}_v[s_\alpha(X_0) + s_\alpha(X_\sigma)\, \mathbf{1}\{0 < \sigma < \tau\}] \\ &= s_\alpha(v) + s_\alpha(v_{\mathrm{pen}}) \Pr_v(0 < \sigma < \tau), \end{aligned}$$

where $\sigma := \inf\{t \geq 0 : \ X_t = v_{\mathrm{pen}}\}$ and v_{pen} is the penultimate state on the same chain as v. Indeed, on $\{0 < \sigma < \tau\}$ we have $X_\sigma = v_{\mathrm{pen}}$, and for all other times $t \in \{0, \dots, \tau-1\} \setminus \{0, \sigma\}$ we have $s_\alpha(X_t) \leq 0$, so discarding those terms only increases the sum. □

Lemma 25. *In the* long–jump–absorbing chain*, with $\tau, \sigma, v_{\mathrm{pen}}$ as in Lemma 24, the derivative $\partial_\beta h_\beta(v)$ satisfies*

$$\partial_\beta h_\beta(v) = \mathcal{E}_v\left[\sum_{t=0}^{\tau-1} s_\beta(X_t)\right] \ \geq \ s_\beta(v) \ + \ s_\beta(v_{\mathrm{pen}}) \Pr_v(0 < \sigma < \tau),$$

where

$$s_\beta(x) = p_b(x)\big(h_\beta(B(x)) - h_\beta(x)\big) \geq 0,$$

with the convention $s_\beta(x) = 0$ if the (same–chain) backward neighbor $B(x)$ does not exist.

In particular, if v is k steps from the terminal on its chain, i.e. $v \in \{a_{n-k}, b_{n-k}\}$ with $2 \le k \le n-1$, then

$$\Pr_v(0 < \sigma < \tau) \;\ge\; \Pr_v\Big(X_i = F^i(v) \;\; \forall i = 1, \dots, k-1\Big),$$

and hence

$$\partial_\beta h_\beta(v) \;\ge\; s_\beta(v) \;+\; s_\beta(v_{\mathrm{pen}}) \Pr_v\Big(X_i = F^i(v) \;\; \forall i = 1, \dots, k-1\Big).$$

Proof. The argument is analogous to that of Lemma 24; we include it here for completeness.

Let $u(\cdot) := h_\beta(\cdot)$ be the absorption probability at e in the long–jump–absorbing chain, so $u(x) = \Pr_x[X_\tau = e]$ with $\tau := \inf\{t \ge 0 : X_t \in \{a_n, b_n, e\}\}$. Apply the source/trajectory representation (Lemma 16) with $\theta = \beta$:

$$\partial_\beta h_\beta(v) = \mathcal{E}_v\left[\sum_{t=0}^{\tau-1} s_\beta(X_t)\right],$$

where $s_\beta(x) := \sum_y (\partial_\beta P(x,y))\, u(y)$ is the local β–source.

By the explicit β–source identity in the long–jump–absorbing chain (Lemma 21),

$$s_\beta(x) = p_b(x)\big(u(B(x)) - u(x)\big),$$

with the convention $s_\beta(x) = 0$ if $B(x)$ does not exist. By monotonicity of u along depth (Lemma 18), we have $u(B(x)) \ge u(x)$ whenever $B(x)$ exists, hence $s_\beta(x) \ge 0$ for all transient x. Therefore the random sum in the trajectory representation is almost surely nonnegative, and so for $\sigma := \inf\{t \ge 0 : X_t = v_{\mathrm{pen}}\}$,

$$\begin{aligned}
\partial_\beta h_\beta(v) &= \mathcal{E}_v\left[\sum_{t=0}^{\tau-1} s_\beta(X_t)\right] \\
&\ge \mathcal{E}_v[s_\beta(X_0) + s_\beta(X_\sigma)\,\mathbf{1}\{0 < \sigma < \tau\}] \\
&= s_\beta(v) + s_\beta(v_{\mathrm{pen}}) \Pr_v(0 < \sigma < \tau),
\end{aligned}$$

since $X_0 = v$ and on $\{0 < \sigma < \tau\}$ we have $X_\sigma = v_{\mathrm{pen}}$. □

Lemma 26. *Assume the forward-drift condition* (28) *and the setting of Lemma 23. Fix a start state $v_0 = a_{n-K}$ at distance $K \ge 1$ from the terminal, where K is a fixed constant independent of n. Let*

$$v_{\mathrm{pen}} := a_{n-1}, \qquad \tau := \inf\{t \ge 0 :\; X_t \in \{a_n, b_n, e\}\}, \qquad \sigma := \inf\{t \ge 0 :\; X_t = v_{\mathrm{pen}}\}.$$

Then for all sufficiently large n (depending on K, α, β, γ),

$$\partial_\gamma h_\gamma(v_0) \;\ge\; \Big(s_\gamma(v_{\mathrm{pen}})\Big) \Pr_{v_0}(0 < \sigma < \tau) \;+\; \frac{1}{2}\, s_\gamma(v_0) \;>\; 0,$$

and

$$\Pr_{v_0}(0 < \sigma < \tau) \;\ge\; \Pr_{v_0}\Big(X_i = F^i(v_0) \;\; \forall i = 1, \dots, K-1\Big).$$

where $s_\gamma(v_0)$ is the local γ–source at the initial state (as given explicitly in Lemma 22 and lower-bounded in Lemma 23).

Proof. Let $u(v) := \Pr_v[X_\tau = e]$ denote the absorption probability at e in the long–jump–absorbing chain, with $\tau := \inf\{t \ge 0 : X_t \in T\}$ and $T := \{a_n, b_n, e\}$. By the source/trajectory representation (Lemma 16),

$$\partial_\gamma u(v_0) = \mathcal{E}_{v_0}\left[\sum_{t=0}^{\tau-1} s_\gamma(X_t)\right]. \tag{30}$$

Since $v_0 = a_{n-K}$ with $K \ge 1$ fixed, Lemma 23 applies (using case (C) when $K = 1$, and case (A) when $K \ge 2$) and gives

$$s_\gamma(v_0) \;\ge\; \frac{1}{2}\, q_{n-K}\,(1 - u(v_0)) \;>\; 0,$$

where $q_{n-K} = \widetilde{P}(a_{n-K}, e)$ is the one-step absorption probability. The factor $1 - u(v_0) \geq \eta > 0$ is bounded below by a positive constant depending only on K and the fixed parameters (since the event of K consecutive forward moves has probability at least $(f_{\min})^K > 0$).

Let $S_2 := \{a_2, b_2\}$. By Lemma 23(A)–(C), for all states $x \notin S_2$ with $x \notin T$, we have $s_\gamma(x) \geq 0$. Thus

$$\mathcal{E}_{v_0}\left[\sum_{t=0}^{\tau-1} s_\gamma(X_t)\right] \;\geq\; s_\gamma(v_0) + \mathcal{E}_{v_0}[s_\gamma(X_\sigma)\,\mathbf{1}\{0 < \sigma < \tau\}] \;+\; \mathcal{E}_{v_0}\left[\sum_{t:\, X_t \in S_2} s_\gamma(X_t)\right].$$

On the event $\{0 < \sigma < \tau\}$ we have $X_\sigma = v_{\text{pen}} = a_{n-1}$, hence

$$\mathcal{E}_{v_0}[s_\gamma(X_\sigma)\,\mathbf{1}\{0 < \sigma < \tau\}] = s_\gamma(v_{\text{pen}})\,\Pr_{v_0}(0 < \sigma < \tau).$$

Moreover, since $v_0 = a_{n-K}$, the event of $K-1$ consecutive forward steps implies $\{0 < \sigma < \tau\}$, so

$$\Pr_{v_0}(0 < \sigma < \tau) \;\geq\; \Pr_{v_0}\Big(X_i = F^i(v_0)\;\; \forall i = 1, \ldots, K-1\Big) = \prod_{j=0}^{K-2} p_f\big(F^j(v_0)\big).$$

Under the forward-drift condition $\alpha > \beta$, there exists $\rho \in (0,1)$ (depending only on α, β, γ) such that $p_b(x)/p_f(x) \leq \rho$ for all deep–middle states. By the gambler's–ruin estimate (Lemma 63), the probability of reaching the second vertex $S_2 = \{a_2, b_2\}$ from a start state at depth K before absorption satisfies

$$\Pr_{v_0}(H_2) \;\leq\; \rho^{\,n-K-2},$$

where $H_2 := \{\exists\, t < \tau : X_t \in S_2\}$.

To control the (potentially negative) contribution from visits to S_2, we use the absolute bound from Lemma 23(D): for each $x \in S_2$,

$$|s_\gamma(x)| \;\leq\; 4C_\star\, q_{\text{mid}}\, \mathcal{E}_x[\tau].$$

Let $N_2 := \sum_{t<\tau} \mathbf{1}\{X_t \in S_2\}$. Then

$$\left|\mathcal{E}_{v_0}\left[\sum_{t:\, X_t \in S_2} s_\gamma(X_t)\right]\right| \;\leq\; \sup_{x \in S_2} |s_\gamma(x)| \cdot \mathcal{E}_{v_0}[N_2] \;\leq\; 4C_\star q_{\text{mid}}\Big(\sup_{x \in S_2} \mathcal{E}_x[\tau]\Big)\mathcal{E}_{v_0}[N_2],$$

where we used $\mathcal{E}_{v_0}[N_2] \leq \Pr_{v_0}(H_2) \cdot \sup_{x \in S_2} \mathcal{E}_x[\tau]$ and the polynomial bound on hitting times from Lemma 23. Combining with $\Pr_{v_0}(H_2) \leq \rho^{\,n-K-2}$, we obtain

$$\left|\mathcal{E}_{v_0}\left[\sum_{t:\, X_t \in S_2} s_\gamma(X_t)\right]\right| \;\leq\; C'\, q_{\text{mid}}\, \text{poly}(n) \cdot \rho^{\,n-K-2},$$

for a constant $C' > 0$ depending only on $(\alpha, \beta, \gamma, K)$.

Combining the above bounds, we have

$$\partial_\gamma u(v_0) \;\geq\; s_\gamma(v_0) + s_\gamma(v_{\text{pen}})\,\Pr_{v_0}(0 < \sigma < \tau) - C'\, q_{\text{mid}}\, \text{poly}(n) \cdot \rho^{\,n-K-2}.$$

Finally, note that the main term is itself proportional to q_{mid}: by the lower bound above together with (29),

$$s_\gamma(v_0) \;\geq\; \frac{1}{2}\, q_{n-K}\, \eta \;\geq\; \frac{\eta}{2C_\star}\, q_{\text{mid}}.$$

Thus both the positive contribution $s_\gamma(v_0)$ and the worst-case negative correction from S_2 carry a factor q_{mid}, but the latter is suppressed by the exponentially small prefactor $\text{poly}(n)\rho^{\,n-K-2}$. Since $\rho < 1$, for all sufficiently large n we have

$$C'\, q_{\text{mid}}\, \text{poly}(n) \cdot \rho^{\,n-K-2} \;\leq\; \frac{1}{2}\, s_\gamma(v_0),$$

and hence

$$\partial_\gamma h_\gamma(v_0) \;\geq\; s_\gamma(v_{\text{pen}})\,\Pr_{v_0}(0 < \sigma < \tau) + \frac{1}{2}\, s_\gamma(v_0) \;>\; 0.$$

□

F.4 Analysis of no-long-jump chain

In this subsection, we analyze the no-long-jump (NL) chain defined in Definition 17, which captures the canonical chain's behavior conditioned on never making a long jump before absorption. The analysis mirrors that of Appendix F.3: we compute the source functions for the NL chain, establish symmetry properties, and derive derivative estimates in Appendix F.5.1.

We recall the base model assumptions from Section 5.2:

Assumption 6. *For any input sequence $s_{1:L_0}$, let p_{fwd}, p_{bwd}, and p_{switch} denote the probabilities that the model performs a forward, backward, or switch step respectively at the first step (time $t = L_0 + 1$). We assume that for any $s_{1:L_0}$, $p_{fwd} > p_{bwd} + \zeta$ for some constant $\zeta > 0$, and additionally $p_{switch} > \upsilon$ for some $\upsilon > 0$.*

Setup (NL chain). Throughout this section we work with the *no–long–jump (NL) chain* on the two–chain graph of length n, with absorbing set $\{a_n, b_n\}$. We write $p_{\mathrm{NL}}(v \to k)$ for its one–step transition probabilities. Let $D(\cdot)$ denote depth (distance from the absorbing endpoint on its chain) and let

$$\tau := \inf\{t \geq 0 : \ X_t \in \{a_n, b_n\}\}, \qquad h(v) := \Pr_v^{NL}[X_\tau = a_n].$$

For any parameter $\theta \in \{\alpha, \beta, \gamma\}$ define the local NL source

$$s_{\theta,\mathrm{NL}}(v) := \sum_{k \in \mathcal{N}(v)} \big(\partial_\theta p_{\mathrm{NL}}(v \to k)\big)\, h(k), \qquad \mathcal{N}(v) := \{k \in V \setminus \{v\} : \ |D(k) - D(v)| \leq 1\}.$$

We also write $p_{\mathrm{sw}}(v) := \sum_{k \in \mathcal{S}(v)} p_{\mathrm{NL}}(v \to k)$ for the single–step switch probability (where $\mathcal{S}(v) \subseteq \mathcal{N}(v)$ are the cross–chain neighbors) and $p_{\mathrm{sw}}^{\max} := \sup_{v \notin \{a_n, b_n\}} p_{\mathrm{sw}}(v)$. Finally let

$$\begin{aligned} p_f(v) &:= \Pr\big(D(X_1) = D(v) - 1 \mid X_0 = v\big), \\ p_b(v) &:= \Pr\big(D(X_1) = D(v) + 1 \mid X_0 = v\big), \\ \delta &:= \min_{v \notin \{a_n, b_n\}} \big(p_f(v) - p_b(v)\big) > 0, \end{aligned}$$

where probabilities are under the NL chain.

Lemma 27. *Consider the* no–long–jump (NL) chain *on the two–chain graph of length n (Definition 17), and write $p_{\mathrm{sw}}(v)$ for the one-step switch probability in the NL chain (as in paragraph F.4 below). Under Assumption 6 and the logit structure of Appendix F.1, there exists a constant $C > 0$ (depending only on fixed parameters) such that for all nonterminal states v,*

$$p_{\mathrm{sw}}(v) \ \leq \ \frac{C}{n}.$$

Proof. Fix a nonterminal state v, and let f be the logit of the forward probability. Because the forward probability is bounded below by a constant $\zeta > 0$, we have that $e^f \geq \Theta(n)\zeta \cdot e^\gamma$. Because the logits of the switch targets are are all at most γ (recall that $\gamma > 0$), and in the No Long jump chain, there are at most 3 such targets, it follows that the switch probability is bounded above $3 \cdot \frac{e^\gamma}{e^f} \leq 3 \cdot \frac{e^\gamma}{\Theta(n)\zeta \cdot e^\gamma} \leq \frac{C}{n}$. □

Lemma 28. *Let $p(v \to k)$ denote the one-step transition probability in the* original *(unconditioned) chain, and let $p_{\mathrm{NL}}(v \to k)$ denote the one-step transition probability in the* no–long–jump (NL) chain*. Fix any nonterminal state v and any state $k \neq v$ such that the transition $v \to k$ is* not *a long jump (equivalently, $|D(k) - D(v)| \leq 1$). Then*

$$p_{\mathrm{NL}}(v \to k) \ \leq \ \frac{1}{\zeta}\, p(v \to k),$$

where ζ is the constant from Assumption 6.

Proof. Let $\mathcal{N}(v) := \{y \in V \setminus \{v\} : \ |D(y) - D(v)| \leq 1\}$ be the set of non–long–jump targets from v. By Definition 17, the NL kernel is obtained by conditioning on the event $\{X_1 \in \mathcal{N}(v)\}$, hence

$$p_{\mathrm{NL}}(v \to k) = \Pr(X_1 = k \mid X_0 = v, \ X_1 \in \mathcal{N}(v)) = \frac{p(v \to k)}{\Pr(X_1 \in \mathcal{N}(v) \mid X_0 = v)}.$$

Moreover, the forward move $v \to F(v)$ is never a long jump, so $\Pr(X_1 \in \mathcal{N}(v) \mid X_0 = v) \geq p(v \to F(v))$. By Assumption 6, $p(v \to F(v)) \geq \zeta$, hence $\Pr(X_1 \in \mathcal{N}(v) \mid X_0 = v) \geq \zeta$, which yields the claimed bound. □

Lemma 29. *Fix $k \geq 1$. In the notation of the NL setup above (paragraph F.4), for $v = a_{n-k}$ we have*

$$1 - h(v) \;\leq\; p_{\mathrm{sw}}^{\max} \cdot \frac{k}{\zeta}.$$

Consequently, for fixed k, the same-side absorption probability tends to 1 as n tends to infinity in the NL chain.

Proof. Let $D(\cdot)$ be the depth–distance to the end and define, for each transient state $x \in V \setminus \{a_n, b_n\}$,

$$\begin{aligned} p_f(x) &:= \Pr\big(D(X_1) = D(x) - 1 \mid X_0 = x\big), \\ p_b(x) &:= \Pr\big(D(X_1) = D(x) + 1 \mid X_0 = x\big), \\ \delta &:= \min_{x \in V \setminus \{a_n, b_n\}} \big(p_f(x) - p_b(x)\big) > 0. \end{aligned}$$

If $X_\tau = b_n$ starting from a_{n-k}, then the trajectory must switch from chain A to chain B at least once before absorption. Let E be the event that at least one switch occurs before τ, and let N_{sw} be the number of switches before τ. Then

$$1 - S_n^{NL}(a_{n-k}) = \Pr(X_\tau = b_n) \leq \Pr(E) \leq \Pr(N_{\mathrm{sw}} \geq 1) \leq \mathcal{E}[N_{\mathrm{sw}}].$$

Moreover,

$$\mathcal{E}[N_{\mathrm{sw}}] = \mathcal{E}\Big[\sum_{t=0}^{\tau-1} \mathbf{1}\{\text{switch at } t\}\Big] \leq \mathcal{E}\Big[\sum_{t=0}^{\tau-1} p_{\mathrm{sw}}(X_t)\Big] \leq p_{\mathrm{sw}}^{\max}\, \mathcal{E}[\tau].$$

By Lemma 64 applied to the depth process (using that $p_f(x) - p_b(x) \geq \delta$ for all transient x), $\mathcal{E}[\tau] \leq k/\delta$. Combining yields the stated bound. The consequence now follows from lemma 27. □

Lemma 30. *Fix $k \geq 2$. In the notation of the NL setup above (paragraph F.4), for every $j \in \{1, \ldots, k-1\}$,*

$$\begin{aligned} |h(a_{n-j}) - h(a_{n-(j+1)})| &\leq \max\Big(1 - S_n^{NL}(a_{n-j}),\, 1 - S_n^{NL}(a_{n-(j+1)})\Big) \\ &\leq p_{\mathrm{sw}}^{\max}\, \max\Big(\mathcal{E}^{NL}_{a_{n-j}}[\tau],\, \mathcal{E}^{NL}_{a_{n-(j+1)}}[\tau]\Big) \\ &\leq p_{\mathrm{sw}}^{\max}\, \max\Big(\tfrac{D(a_{n-j})}{\delta},\, \tfrac{D(a_{n-(j+1)})}{\delta}\Big) \;=\; p_{\mathrm{sw}}^{\max}\, \frac{j+1}{\delta}, \end{aligned}$$

where $D(a_{n-j}) = j$.

Proof. Let $u_j := S_n^{NL}(a_{n-j}) \in [0, 1]$. Then

$$|u_j - u_{j+1}| = |(1 - u_j) - (1 - u_{j+1})| \leq \max(1 - u_j,\, 1 - u_{j+1}).$$

Moreover, for any start state $x \in \{a_1, \ldots, a_{n-1}\}$, the event $\{X_\tau = b_n\}$ implies that the trajectory switches chains at least once before absorption, hence

$$1 - \underset{x}{\overset{NL}{\Pr}}(X_\tau = a_n) = \underset{x}{\overset{NL}{\Pr}}(X_\tau = b_n) \leq \mathcal{E}_x^{NL}[N_{\mathrm{sw}}] \leq p_{\mathrm{sw}}^{\max}\, \mathcal{E}_x^{NL}[\tau],$$

where the last inequality follows from $\mathcal{E}_x^{NL}[N_{\mathrm{sw}}] \leq \mathcal{E}_x^{NL}\big[\sum_{t=0}^{\tau-1} p_{\mathrm{sw}}(X_t)\big] \leq p_{\mathrm{sw}}^{\max} \mathcal{E}_x^{NL}[\tau]$. Applying this with $x = a_{n-j}$ and $x = a_{n-(j+1)}$ and combining with the first display gives the stated bound. For the final inequality, apply Lemma 64 to the depth process (with drift gap δ) to obtain $\mathcal{E}^{NL}_{a_{n-j}}[\tau] \leq D(a_{n-j})/\delta = j/\delta$ and $\mathcal{E}^{NL}_{a_{n-(j+1)}}[\tau] \leq D(a_{n-(j+1)})/\delta = (j+1)/\delta$. □

F.5 Source computations in the no–long–jump (NL) chain

Lemma 31. *In the notation of the NL setup above (paragraph F.4), consider the local γ–source $s_{\gamma,\mathrm{NL}}(v)$ and the switch probability $p_{\mathrm{sw}}(v)$. Define $\mathrm{osc}(h) := \sup_{x,y\in V} |h(x) - h(y)| \le 1$. Under the logit structure of Appendix F.1 (after the global shift by $-\gamma$), the only γ–dependence among allowed NL moves is:*

- *If $v \notin \{a_2, b_2, a_{n-1}, b_{n-1}\}$, then $\partial_\gamma o(v,k) = +1$ for $k \in \mathcal{S}(v)$ and $\partial_\gamma o(v,k) = 0$ for $k \in \mathcal{N}(v) \setminus \mathcal{S}(v)$.*
- *If $v \in \{a_2, b_2\}$, then in addition $\partial_\gamma o(v, v^-) = -1$ for the unique same–chain backward neighbor $v^- \in \mathcal{N}(v)$.*
- *If $v \in \{a_{n-1}, b_{n-1}\}$, then in addition $\partial_\gamma o(v, v^+) = -1$ for the unique same–chain forward neighbor $v^+ \in \mathcal{N}(v)$.*

Then:

(A) Generic vertices $v \notin \{a_2, b_2, a_{n-1}, b_{n-1}\}$.

$$s_{\gamma,\mathrm{NL}}(v) = \sum_{k\in\mathcal{S}(v)} p_{\mathrm{NL}}(v \to k)\big(h(k) - h(v)\big), \qquad \textit{hence} \qquad |s_{\gamma,\mathrm{NL}}(v)| \le p_{\mathrm{sw}}(v)\,\mathrm{osc}(h) \le p_{\mathrm{sw}}(v).$$

(B) Second vertex $v \in \{a_2, b_2\}$.

$$s_{\gamma,\mathrm{NL}}(v) = \sum_{k\in\mathcal{S}(v)} p_{\mathrm{NL}}(v \to k)\big(h(k) - h(v)\big) \;-\; p_{\mathrm{NL}}(v \to v^-)\big(h(v^-) - h(v)\big).$$

Consequently,

$$|s_{\gamma,\mathrm{NL}}(v)| \le p_{\mathrm{sw}}(v)\,\mathrm{osc}(h) + p_{\mathrm{NL}}(v \to v^-)\,|h(v^-) - h(v)| \le p_{\mathrm{sw}}(v) + p_{\mathrm{NL}}(v \to v^-)\,|h(v^-) - h(v)|.$$

(C) Penultimate vertex $v \in \{a_{n-1}, b_{n-1}\}$.

$$s_{\gamma,\mathrm{NL}}(v) = \sum_{k\in\mathcal{S}(v)} p_{\mathrm{NL}}(v \to k)\big(h(k) - h(v)\big) \;-\; p_{\mathrm{NL}}(v \to v^+)\big(h(v^+) - h(v)\big).$$

Consequently,

$$|s_{\gamma,\mathrm{NL}}(v)| \le p_{\mathrm{sw}}(v)\,\mathrm{osc}(h) + p_{\mathrm{NL}}(v \to v^+)\,|h(v^+) - h(v)| \le p_{\mathrm{sw}}(v) + p_{\mathrm{NL}}(v \to v^+)\,|h(v^+) - h(v)|.$$

Proof. Fix $v \in U$. Abbreviate $p_k := p_{\mathrm{NL}}(v \to k)$ and $o_k := o(v,k)$ for $k \in \mathcal{N}(v)$. By the (restricted) softmax derivative identity (Lemma 62),

$$\partial_\gamma p_k = p_k\Big(\partial_\gamma o_k - \sum_{k'\in\mathcal{N}(v)} p_{k'}\,\partial_\gamma o_{k'}\Big).$$

Therefore

$$\begin{aligned} s_{\gamma,\mathrm{NL}}(v) &= \sum_{k\in\mathcal{N}(v)} \big(\partial_\gamma p_k\big)\,h(k) \\ &= \sum_{k\in\mathcal{N}(v)} p_k\,h(k)\Big(\partial_\gamma o_k - \sum_{k'\in\mathcal{N}(v)} p_{k'}\,\partial_\gamma o_{k'}\Big) \\ &= \sum_{k\in\mathcal{N}(v)} p_k\,h(k)\,\partial_\gamma o_k - \Big(\sum_{k\in\mathcal{N}(v)} p_k\,h(k)\Big)\Big(\sum_{k'\in\mathcal{N}(v)} p_{k'}\,\partial_\gamma o_{k'}\Big). \end{aligned}$$

Since h is harmonic at transient states for the NL chain, $h(v) = \mathcal{E}_v[h(X_1)]$, we can rewrite this in centered form as

$$s_{\gamma,\mathrm{NL}}(v) = \sum_{k\in\mathcal{N}(v)} p_k\,\big(\partial_\gamma o_k\big)\,\big(h(k) - h(v)\big).$$

Now apply the case split on $\partial_\gamma o(v,\cdot)$ listed in the statement:

- If $v \notin \{a_2, b_2, a_{n-1}, b_{n-1}\}$, then $\partial_\gamma o(v,k) = 1$ on $k \in \mathcal{S}(v)$ and zero otherwise, giving (A).
- If $v \in \{a_2, b_2\}$, then in addition $\partial_\gamma o(v, v^-) = -1$, giving (B).
- If $v \in \{a_{n-1}, b_{n-1}\}$, then in addition $\partial_\gamma o(v, v^+) = -1$, giving (C).

The displayed upper bounds follow from the triangle inequality together with $|h(k) - h(v)| \le \operatorname{osc}(h) \le 1$. □

Lemma 32. *In the notation of the NL setup above (paragraph F.4), let v^+ be the same–chain forward neighbor of v when it exists. Let v^+ denote the unique same–chain forward neighbor of v (so $v^+ = a_{i+1}$ if $v = a_i$ with $i \le n-1$, and similarly on chain B; in particular $a^+_{n-1} = a_n$ and $b^+_{n-1} = b_n$). Then*

$$s_{\alpha,\mathrm{NL}}(v) = p_{\mathrm{NL}}(v \to v^+)\Big(h(v^+) - h(v)\Big). \tag{31}$$

Moreover,

$$\begin{aligned} |s_{\alpha,\mathrm{NL}}(v)| &\le p_{\mathrm{NL}}(v \to v^+)\operatorname{osc}_v(h) \le p_{\mathrm{NL}}(v \to v^+) \le 1, \\ \operatorname{osc}_v(h) &:= \max_{k \in \mathcal{N}(v)} h(k) - \min_{k \in \mathcal{N}(v)} h(k) \le 1. \end{aligned} \tag{32}$$

Proof. Fix $v \in U$. Abbreviate $p_k := p_{\mathrm{NL}}(v \to k)$ and $o_k := o(v,k)$ for $k \in \mathcal{N}(v)$. For restricted softmax,

$$\partial_\alpha p_k = p_k\Big(\partial_\alpha o_k - \sum_{k' \in \mathcal{N}(v)} p_{k'}\, \partial_\alpha o_{k'}\Big).$$

By the specified logits, α appears only on the forward edge $k = v^+$ with coefficient $+1$, and nowhere else among allowed neighbors. Hence

$$\partial_\alpha o_k = \begin{cases} 1, & k = v^+, \\ 0, & k \in \mathcal{N}(v) \setminus \{v^+\}. \end{cases}$$

Therefore,

$$\sum_{k' \in \mathcal{N}(v)} p_{k'}\, \partial_\alpha o_{k'} = p_{\mathrm{NL}}(v \to v^+), \qquad \sum_{k \in \mathcal{N}(v)} p_k\, h(k)\, \partial_\alpha o_k = p_{\mathrm{NL}}(v \to v^+)\, h(v^+).$$

Now expand

$$s_{\alpha,\mathrm{NL}}(v) = \sum_{k \in \mathcal{N}(v)} (\partial_\alpha p_k)\, h(k) = \sum_{k \in \mathcal{N}(v)} p_k h(k) \partial_\alpha o_k - \Big(\sum_{k \in \mathcal{N}(v)} p_k h(k)\Big)\Big(\sum_{k' \in \mathcal{N}(v)} p_{k'} \partial_\alpha o_{k'}\Big),$$

and substitute the two identities above to obtain (31).

For (32), note that h is harmonic for the NL chain at transient v, so $h(v) = \mathcal{E}_v[h(X_1)] = \sum_{k \in \mathcal{N}(v)} p_k\, h(k)$ is a convex combination of $\{h(k) : k \in \mathcal{N}(v)\}$. Hence $|h(v^+) - h(v)| \le \operatorname{osc}_v(h) \le 1$. Multiply by $p_{\mathrm{NL}}(v \to v^+)$. □

Lemma 33. *In the notation of the NL setup above (paragraph F.4), let v^- be the same–chain backward neighbor of v when it exists. Let v^- denote the unique same–chain backward neighbor of v when it exists (so $v^- = a_{i-1}$ if $v = a_i$ with $i \ge 2$, and similarly on chain B). Then for all $v \in U$,*

$$s_{\beta,\mathrm{NL}}(v) = \begin{cases} p_{\mathrm{NL}}(v \to v^-)\Big(h(v^-) - h(v)\Big), & \text{if } v^- \text{ exists}, \\ 0, & \text{if } v \in \{a_1, b_1\}. \end{cases} \tag{33}$$

Moreover, whenever v^- exists,

$$|s_{\beta,\mathrm{NL}}(v)| \le p_{\mathrm{NL}}(v \to v^-)\operatorname{osc}_v(h) \le p_{\mathrm{NL}}(v \to v^-) \le 1, \qquad \operatorname{osc}_v(h) \le 1. \tag{34}$$

Proof. Fix $v \in U$. If $v \in \{a_1, b_1\}$, the logits do not contain β on any allowed edge, so $\partial_\beta o(v,k) = 0$ for all $k \in \mathcal{N}(v)$ and hence $s_{\beta,\mathrm{NL}}(v) = 0$.

Otherwise, the specified logits place β only on the single backward edge $k = v^-$, with coefficient $+1$ (including the boundary case $v \in \{a_2, b_2\}$ where the back logit is $(\beta - \gamma)$). Thus

$$\partial_\beta o_k = \begin{cases} 1, & k = v^-, \\ 0, & k \in \mathcal{N}(v) \setminus \{v^-\}. \end{cases}$$

The rest of the argument is identical to the proof of Lemma 32: use the restricted softmax derivative, expand $s_{\beta,\mathrm{NL}}(v)$, and substitute

$$\sum_{k' \in \mathcal{N}(v)} p_{\mathrm{NL}}(v \to k')\, \partial_\beta o_{k'} = p_{\mathrm{NL}}(v \to v^-),$$

$$\sum_{k \in \mathcal{N}(v)} p_{\mathrm{NL}}(v \to k)\, h(k)\, \partial_\beta o_k = p_{\mathrm{NL}}(v \to v^-)\, h(v^-),$$

to get (33). The bound (34) follows from $|h(v^-) - h(v)| \le \mathrm{osc}_v(h) \le 1$ (and $\mathcal{E}_v[h(X_1)] = h(v)$ by harmonicity). □

F.5.1 Derivative estimates

Having computed the source functions for the NL chain in the preceding lemmas, we now derive quantitative bounds on the derivatives of the same-side absorption probability.

Lemma 34. *In the notation of the NL setup above (paragraph F.4), let*

$$u(x) := \Pr_x^{NL}[X_\tau = a_n], \qquad u_\beta(x) := \partial_\beta u(x),$$

and fix $v = a_{n-k}$ with k independent of n. Assume the base model assumptions of Assumption 6 (in particular, a uniform drift gap $\zeta > 0$). Then there exist constants $C, c > 0$ (depending only on fixed parameters and ζ) such that for all n large enough,

$$|u_\beta(v)| \;\le\; C\,\frac{\sqrt{n}}{\zeta}\, p_b^{(2\lfloor n^{1/4} \rfloor)}\, p_{\mathrm{sw}}^{\max} \;+\; C\, n\, \exp\!\left(-\frac{c^2}{8}\, n^{1/4}\right) \cdot \sup_{x \notin \{a_n, b_n\}} |s_{\beta,\mathrm{NL}}(x)|.$$

Proof. Fix $K := \lfloor n^{1/4} \rfloor$. By the trajectory/source representation (Lemma 16) for the NL chain,

$$u_\beta(v) = \mathcal{E}_v\left[\sum_{t=0}^{\tau-1} s_{\beta,\mathrm{NL}}(X_t)\right].$$

Decompose on $\{\tau \le K\}$ and $\{\tau > K\}$. On $\{\tau \le K\}$, the sum has at most K terms and the depth can increase by at most one per step, so every visited state satisfies $D(X_t) \le D(v) + t \le k + K \le 2K$ for all large n. Therefore,

$$|u_\beta(v)| \le K \cdot \sup_{x:\, D(x) \le 2K} |s_{\beta,\mathrm{NL}}(x)|.$$

By Lemma 33, $|s_{\beta,\mathrm{NL}}(x)| \le p_b(x)\, |u(B(x)) - u(x)|$. By Lemma 30, $|u(B(x)) - u(x)| \le p_{\mathrm{sw}}^{\max} \mathcal{E}_x[\tau]$. Under the base model forward-bias gap $\zeta > 0$, the depth-to-terminal process has $\mathcal{E}_x[\tau] \le D(x)/\zeta \le 2K/\zeta$ on the band $D(x) \le 2K$. Combining yields

$$\sup_{x:\, D(x) \le 2K} |s_{\beta,\mathrm{NL}}(x)| \;\le\; C\, p_b^{(2K)}\, p_{\mathrm{sw}}^{\max} \cdot \frac{2K}{\zeta},$$

for a constant $C > 0$, hence $|u_\beta(v)| \lesssim \frac{K^2}{\zeta}\, p_b^{(2K)} p_{\mathrm{sw}}^{\max}$. For $\{\tau > K\}$, write $\sum_{t=0}^{\tau-1} = \sum_{t=0}^{K-1} + \sum_{t=K}^{\tau-1}$. The first part is bounded identically by the same K-step band argument above. For the tail, since $\sup_x \mathcal{E}_x[\tau] \le n/\zeta$ under the drift gap and $|s_{\beta,\mathrm{NL}}(x)| \le \sup_y |s_{\beta,\mathrm{NL}}(y)|$, the contribution after time K is at most on the order of $n \cdot \sup_y |s_{\beta,\mathrm{NL}}(y)|$. Multiplying by $\Pr_v(\tau > K)$ and using Lemma 65 (with $t = \lfloor n^{1/4} \rfloor$) gives

$$\Pr_v(\tau > K) \cdot n \cdot \sup_y |s_{\beta,\mathrm{NL}}(y)| \;\le\; n \cdot \exp\!\left(-\frac{c^2}{8}\, n^{1/4}\right) \cdot \sup_y |s_{\beta,\mathrm{NL}}(y)|,$$

□

The next lemma is completely analogous to Lemma 34 and is included for completeness.

Lemma 35. *In the notation of the NL setup above (paragraph F.4), let*

$$u(x) := \overset{NL}{\Pr_x}[X_\tau = a_n], \qquad u_\alpha(x) := \partial_\alpha u(x),$$

and fix $v = a_{n-k}$ with k independent of n. Assume the base model assumptions of Assumption 6 (in particular, a uniform drift gap $\zeta > 0$). Then there exist constants $C, c > 0$ (depending only on fixed parameters and ζ) such that for all n large enough,

$$|u_\alpha(v)| \;\le\; C\,\frac{\sqrt{n}}{\zeta}\,p_f^{(2\lfloor n^{1/4}\rfloor)}\,p_{\mathrm{sw}}^{\max} \;+\; C\,n\,\exp\!\left(-\frac{c^2}{8}\,n^{1/4}\right)\cdot \sup_{x\notin\{a_n,b_n\}} |s_{\alpha,\mathrm{NL}}(x)|,$$

where $p_f^{(m)} := \sup\{p_{\mathrm{NL}}(x \to x^+) :\; x \notin \{a_n, b_n\},\; D(x) \le m\}$.

Proof. Fix $K := \lfloor n^{1/4} \rfloor$. By the trajectory/source representation (Lemma 16) for the NL chain,

$$u_\alpha(v) = \mathcal{E}_v\left[\sum_{t=0}^{\tau-1} s_{\alpha,\mathrm{NL}}(X_t)\right].$$

Decompose on $\{\tau \le K\}$ and $\{\tau > K\}$. On $\{\tau \le K\}$, the sum has at most K terms and the depth can increase by at most one per step, so every visited state satisfies $D(X_t) \le D(v) + t \le k + K \le 2K$ for all large n. Therefore,

$$|u_\alpha(v)| \le K \cdot \sup_{x:\, D(x)\le 2K} |s_{\alpha,\mathrm{NL}}(x)|.$$

By Lemma 32, $|s_{\alpha,\mathrm{NL}}(x)| \le p_{\mathrm{NL}}(x \to x^+)\,|u(x^+) - u(x)|$. By Lemma 30 (applied to the appropriate chain), $|u(x^+) - u(x)| \le p_{\mathrm{sw}}^{\max}\mathcal{E}_x[\tau]$. Under the base model forward-bias gap $\zeta > 0$, the depth-to-terminal process has $\mathcal{E}_x[\tau] \le D(x)/\zeta \le 2K/\zeta$ on the band $D(x) \le 2K$. Combining yields

$$\sup_{x:\, D(x)\le 2K} |s_{\alpha,\mathrm{NL}}(x)| \;\le\; C\,p_f^{(2K)}\,p_{\mathrm{sw}}^{\max}\cdot\frac{2K}{\zeta},$$

for a constant $C > 0$, hence $|u_\alpha(v)| \lesssim \frac{K^2}{\zeta}\,p_f^{(2K)}p_{\mathrm{sw}}^{\max}$. For $\{\tau > K\}$, write $\sum_{t=0}^{\tau-1} = \sum_{t=0}^{K-1} + \sum_{t=K}^{\tau-1}$. The first part is bounded identically by the same K-step band argument above. For the tail, since $\sup_x \mathcal{E}_x[\tau] \le n/\zeta$ under the drift gap and $|s_{\alpha,\mathrm{NL}}(x)| \le \sup_y |s_{\alpha,\mathrm{NL}}(y)|$, the contribution after time K is at most on the order of $n \cdot \sup_y |s_{\alpha,\mathrm{NL}}(y)|$. Multiplying by $\Pr_v(\tau > K)$ and using Lemma 65 (with $t = \lfloor n^{1/4} \rfloor$) gives

$$\Pr_v(\tau > K)\cdot n \cdot \sup_y |s_{\alpha,\mathrm{NL}}(y)| \;\le\; n\cdot\exp\!\left(-\frac{c^2}{8}\,n^{1/4}\right)\cdot\sup_y |s_{\alpha,\mathrm{NL}}(y)|,$$

as needed. □

The next lemma is completely analogous to Lemmas 34 and 35 and is included for completeness.

Lemma 36. *In the notation of the NL setup above (paragraph F.4), let*

$$u(x) := \overset{NL}{\Pr_x}[X_\tau = a_n], \qquad u_\gamma(x) := \partial_\gamma u(x),$$

and fix $v = a_{n-k}$ with k independent of n. Assume the base model assumptions of Assumption 6 (in particular, a uniform drift gap $\zeta > 0$). Then there exist constants $C, c > 0$ (depending only on fixed parameters and ζ) such that for all n large enough,

$$|u_\gamma(v)| \;\le\; C\,n^{1/4}\,p_{\mathrm{sw}}^{\max} \;+\; C\,\frac{\sqrt{n}}{\zeta}\,p_f^{(2\lfloor n^{1/4}\rfloor)}\,p_{\mathrm{sw}}^{\max} \;+\; C\,n\,\exp\!\left(-\frac{c^2}{8}\,n^{1/4}\right)\cdot \sup_{x\notin\{a_n,b_n\}} |s_{\gamma,\mathrm{NL}}(x)|,$$

where $p_f^{(m)} := \sup\{p_{\mathrm{NL}}(x \to x^+) :\; x \notin \{a_n, b_n\},\; D(x) \le m\}$.

Proof. Fix $K := \lfloor n^{1/4} \rfloor$. By the trajectory/source representation (Lemma 16) for the NL chain,

$$u_\gamma(v) = \mathcal{E}_v \left[\sum_{t=0}^{\tau-1} s_{\gamma,\mathrm{NL}}(X_t) \right].$$

Decompose on $\{\tau \le K\}$ and $\{\tau > K\}$. On $\{\tau \le K\}$, the sum has at most K terms and the depth can increase by at most one per step, so every visited state satisfies $D(X_t) \le D(v) + t \le k + K \le 2K$ for all large n. Therefore,

$$|u_\gamma(v)| \le K \cdot \sup_{x:\, D(x) \le 2K} |s_{\gamma,\mathrm{NL}}(x)|.$$

For x with $D(x) \le 2K$, Lemma 31 gives a uniform upper bound: by part (A) we have $|s_{\gamma,\mathrm{NL}}(x)| \le p_{\mathrm{sw}}(x)$, while at the boundary $x \in \{a_{n-1}, b_{n-1}\}$ part (C) yields

$$|s_{\gamma,\mathrm{NL}}(x)| \le p_{\mathrm{sw}}(x) + p_{\mathrm{NL}}(x \to x^+)\, |u(x^+) - u(x)|.$$

By Lemma 30 (applied to the appropriate chain), $|u(x^+) - u(x)| \le p_{\mathrm{sw}}^{\max} \mathcal{E}_x[\tau]$. Under the base model forward-bias gap $\zeta > 0$, the depth-to-terminal process has $\mathcal{E}_x[\tau] \le D(x)/\zeta \le 2K/\zeta$ on the band $D(x) \le 2K$. Combining yields

$$\sup_{x:\, D(x) \le 2K} |s_{\gamma,\mathrm{NL}}(x)| \;\le\; C\, p_{\mathrm{sw}}^{\max} \;+\; C\, p_f^{(2K)}\, p_{\mathrm{sw}}^{\max} \cdot \frac{2K}{\zeta},$$

and hence $|u_\gamma(v)| \lesssim K\, p_{\mathrm{sw}}^{\max} + \frac{K^2}{\zeta}\, p_f^{(2K)} p_{\mathrm{sw}}^{\max}$. For $\{\tau > K\}$, write $\sum_{t=0}^{\tau-1} = \sum_{t=0}^{K-1} + \sum_{t=K}^{\tau-1}$. The first part is bounded identically by the same K-step band argument above. For the tail, since $\sup_x \mathcal{E}_x[\tau] \le n/\zeta$ under the drift gap and $|s_{\gamma,\mathrm{NL}}(x)| \le \sup_y |s_{\gamma,\mathrm{NL}}(y)|$, the contribution after time K is at most on the order of $n \cdot \sup_y |s_{\gamma,\mathrm{NL}}(y)|$. Multiplying by $\Pr_v(\tau > K)$ and using Lemma 65 (with $t = \lfloor n^{1/4} \rfloor$) gives

$$\Pr_v(\tau > K) \cdot n \cdot \sup_y |s_{\gamma,\mathrm{NL}}(y)| \;\le\; n \cdot \exp\!\left(-\frac{c^2}{8}\, n^{1/4} \right) \cdot \sup_y |s_{\gamma,\mathrm{NL}}(y)|,$$

as needed. □

F.6 Monotonicity of α, β, γ and derivative estimates

We establish monotonicity and estimates for the derivativesof α, β, γ at initialization, but they in fact hold throughout the learning process as long as the forward transition probability remains non-decreasing and the long-jump probability stays bounded away from zero. Since our analysis shows that both conditions are maintained for all $t \le t_0$, the estimates below in fact hold for all $t \le t_0$.

Lemma 37. *Assume $\gamma > 0$ and the logit structure of Appendix F.1 (after the global shift by $-\gamma$), and consider the original chain (equivalently, the long–jump–absorbing chain before redirection of long jumps). For a transient state $x \notin \{a_n, b_n\}$, let*

$$p_b(x) := \Pr(X_1 = B(x) \mid X_0 = x) \quad (\textit{set } p_b(x) := 0 \textit{ if } B(x) \textit{ does not exist}),$$
$$p(e; x) := \Pr(\textit{a long jump at time } 1 \mid X_0 = x),$$

where $B(x)$ is the same–chain backward neighbor when it exists. Then there exists $n_0 \in \mathbb{N}$ (depending only on fixed parameters α, β, γ) such that for all $n \ge n_0$,

$$\max_{x \notin \{a_n, b_n\}} p_b(x) \;\le\; 2\, p_b(a_{n-1}), \qquad \max_{x \notin \{a_n, b_n\}} p(e; x) \;\le\; 2\, p(e; a_{n-1}),$$

and by chain-swap symmetry the same bounds hold with a_{n-1} replaced by b_{n-1}.

Proof. Fix a chain (say A); the statement for chain B follows by chain-swap symmetry. Write the one-step kernel as a softmax over logits: for each transient x and each allowed target y,

$$P(x, y) = \frac{e^{o(x,y)}}{Z(x)}, \qquad Z(x) := \sum_z e^{o(x,z)},$$

where the sum ranges over all original one-step targets z from x (including long-jump targets). Therefore

$$p_b(x) = \frac{e^{o(x,B(x))}}{Z(x)}, \qquad p(e;x) = \sum_{z \in L(x)} \frac{e^{o(x,z)}}{Z(x)} = \frac{\sum_{z \in L(x)} e^{o(x,z)}}{Z(x)}.$$

Under the logit specification of Appendix F.1 (after the harmless global shift by $-\gamma$), the logits take only finitely many values, and both the numerator $\sum_{z \in L(x)} e^{o(x,z)}$ and the normalizer $Z(x)$ can be written explicitly in each of the four boundary regimes. In particular (see the enumeration underlying Lemma 22 for the corresponding target counts and denominators):

- Deep middle $x = a_i$, $2 < i < n-1$:

$$Z_{\mathrm{mid}} = e^{\alpha} + e^{\beta} + 4 + (2n-7)e^{\gamma}, \qquad p_b(a_i) = \frac{e^{\beta}}{Z_{\mathrm{mid}}}, \qquad p(e;a_i) = \frac{(2n-10)e^{\gamma}}{Z_{\mathrm{mid}}}.$$

- Second vertex $x = a_2$:

$$Z_2 = e^{\alpha} + e^{(\beta-\gamma)} + 3 + (2n-6)e^{\gamma}, \qquad p_b(a_2) = \frac{e^{(\beta-\gamma)}}{Z_2}, \qquad p(e;a_2) = \frac{(2n-8)e^{\gamma}}{Z_2}.$$

- First vertex $x = a_1$:

$$Z_1 = e^{\alpha} + 3 + (2n-5)e^{\gamma}, \qquad p_b(a_1) = 0, \qquad p(e;a_1) = \frac{(2n-6)e^{\gamma}}{Z_1}.$$

- Penultimate vertex $x = a_{n-1}$:

$$Z_{n-1} = e^{(\alpha-\gamma)} + e^{\beta} + 3 + (2n-6)e^{\gamma}, \qquad p_b(a_{n-1}) = \frac{e^{\beta}}{Z_{n-1}}, \qquad p(e;a_{n-1}) = \frac{(2n-8)e^{\gamma}}{Z_{n-1}}.$$

Set $C_{\max} := e^{\alpha} + e^{\beta} + 4$. For every transient x on chain A, the normalizer has the form

$$Z(x) = C_x + c_x e^{\gamma}, \qquad 0 \le C_x \le C_{\max}, \qquad c_x \in \{2n-7,\, 2n-6,\, 2n-5\},$$

and at the penultimate vertex $Z_{n-1} = C_{n-1} + (2n-6)e^{\gamma}$ with $0 \le C_{n-1} \le C_{\max}$. Choose n_0 so that $(2n_0 - 6)e^{\gamma} \ge 2C_{\max}$. Then for all $n \ge n_0$,

$$Z_{n-1} \;\le\; (2n-6)e^{\gamma} + C_{\max} \;\le\; \tfrac{3}{2}(2n-6)e^{\gamma}, \qquad Z(x) \;\ge\; (2n-7)e^{\gamma}.$$

We first bound p_b. The only vertices with a backward move are a_i for $i \ge 2$. Among these, the backward weight is e^{β} except at a_2, where it is $e^{(\beta-\gamma)} \le e^{\beta}$. Hence for all x on chain A,

$$p_b(x) \;\le\; \frac{e^{\beta}}{(2n-7)e^{\gamma}}.$$

On the other hand,

$$p_b(a_{n-1}) = \frac{e^{\beta}}{Z_{n-1}} \;\ge\; \frac{e^{\beta}}{\frac{3}{2}(2n-6)e^{\gamma}}.$$

Therefore, for all $n \ge n_0$,

$$\frac{p_b(x)}{p_b(a_{n-1})} \;\le\; \frac{\frac{3}{2}(2n-6)}{(2n-7)} \;\le\; 2,$$

which proves $\max_x p_b(x) \le 2p_b(a_{n-1})$ on chain A.

Next we bound the long-jump mass. In each of the cases above we have $p(e;x) = N_x e^{\gamma}/Z(x)$ with $N_x \in \{2n-6, 2n-8, 2n-10\}$. Thus for all x on chain A and all $n \ge n_0$,

$$p(e;x) \;\le\; \frac{(2n-6)e^{\gamma}}{(2n-7)e^{\gamma}} = \frac{2n-6}{2n-7}.$$

Meanwhile,

$$p(e;a_{n-1}) = \frac{(2n-8)e^{\gamma}}{Z_{n-1}} \;\ge\; \frac{(2n-8)e^{\gamma}}{\frac{3}{2}(2n-6)e^{\gamma}} = \frac{2}{3} \cdot \frac{2n-8}{2n-6}.$$

For n large enough (in particular, for $n \ge n_0$ after possibly increasing n_0 by a fixed constant), the ratio $\frac{2n-6}{2n-7} \Big/ \left(\frac{2}{3} \cdot \frac{2n-8}{2n-6}\right)$ is at most 2, and therefore $\max_x p(e;x) \le 2p(e;a_{n-1})$ on chain A. The same argument applies on chain B. $\square$

Recall the derivative decomposition from Lemma 15. For a parameter $\theta \in \{\alpha, \beta, \gamma\}$ and a start state v, we have

$$\partial_\theta S_\theta(v) = -(\partial_\theta h_\theta(v))\Big(\bar{S}_\theta^{NL}(v) - \tfrac{1}{2}\Big) + (1 - h_\theta(v))\,\partial_\theta \bar{S}_\theta^{NL}(v),$$

where:

- $S_\theta(v) := \Pr_v^\theta[X_\tau = a_n]$ is the same–side absorption probability in the original chain.
- $h_\theta(v)$ is the probability of a long jump before absorption, which by definition equals the absorption probability at S_{LJ} in the long–jump–absorbing chain where long jumps are redirected to the absorbing state S_{LJ}.
- $\bar{S}_\theta^{NL}(v) := \Pr_v^\theta(X_\tau = a_n \mid NL)$ is the same–side absorption probability conditional on no long jump occurring (i.e. in the no–long–jump chain).
- By symmetry, $\bar{S}_\theta^{LJ}(v) = \frac{1}{2}$, so the long–jump conditional absorption probability equals $1/2$.

We will prove monotonicity results for each of the three parameters $\theta \in \{\alpha, \beta, \gamma\}$ by showing that the term involving the long–jump–absorbing chain, $-(\partial_\theta h_\theta(v))\Big(\bar{S}_\theta^{NL}(v) - \frac{1}{2}\Big)$, dominates the remaining term $(1 - h_\theta(v))\,\partial_\theta \bar{S}_\theta^{NL}(v)$ in the derivative decomposition.

Lemma 38. *Fix a constant $K \geq 1$ independent of n and initialize at $v_0 := a_{n-K}$. For all n large enough, the long–jump term in the decomposition of Lemma 15 dominates the NL term, in the sense that*

$$\partial_\alpha S_\alpha(v_0) \;\geq\; \frac{1}{8}(\partial_\alpha h_\alpha(v_0)) \geq 0.$$

Moreover, $\partial_\alpha h_\alpha(v_0) \leq 0$, so the right-hand side is nonnegative. In particular, $\partial_\alpha S_\alpha(v_0) \geq 0$, so the same–side absorption probability is (near-terminal) monotonically increasing in α.

Proof. Recall the derivative decomposition (Lemma 15):

$$\partial_\alpha S_\alpha(v_0) = -(\partial_\alpha h_\alpha(v_0))\Big(\bar{S}_\alpha^{NL}(v_0) - \tfrac{1}{2}\Big)(1 - h_\alpha(v_0))\,\partial_\alpha \bar{S}_\alpha^{NL}(v_0).$$

We first lower bound the long–jump contribution. By Lemma 24, together with Corollary 2 and the base model assumption $p_{\mathrm{fwd}} \geq \zeta$ (Assumption 6), there exists a constant $c_{\mathrm{LJ}} > 0$ (depending only on fixed parameters and K) such that

$$|\partial_\alpha h_\alpha(v_0)| \;\geq\; c_{\mathrm{LJ}} \cdot p(e), \qquad p(e) := P^{\text{LJ-abs}}(v_0, e).$$

Moreover, by Lemma 29 applied in the NL chain at depth K, for n large enough we have $\bar{S}_\alpha^{NL}(v_0) \geq \frac{3}{4}$, hence $\bar{S}_\alpha^{NL}(v_0) - \frac{1}{2} \geq \frac{1}{4}$.

We next upper bound the NL contribution. By the truncation bound in Lemma 35 (completely analogous to Lemma 34), we have

$$\big|\partial_\alpha \bar{S}_\alpha^{NL}(v_0)\big| \;\leq\; C\,\frac{\sqrt{n}}{\zeta}\,p_f^{(2\lfloor n^{1/4}\rfloor)}\,p_{\mathrm{sw}}^{\max} \;+\; C\,n\,\exp\!\left(-\frac{c^2}{8}\,n^{1/4}\right)\cdot \sup_{x \notin \{a_n, b_n\}} |s_{\alpha,\mathrm{NL}}(x)|.$$

Using Lemma 30, $\sup_x |s_{\alpha,\mathrm{NL}}(x)| \lesssim p_{\mathrm{sw}}^{\max} \sup_x \mathcal{E}_x^{NL}[\tau] \lesssim (n/\zeta)\,p_{\mathrm{sw}}^{\max}$ and the exponential tail $\exp(-\Omega(n^{1/4}))$, the last term is a negligible fraction of $p_{\mathrm{sw}}^{\max}$. Since $p_f^{(2\lfloor n^{1/4}\rfloor)} \leq 1$, we obtain $\big|\partial_\alpha \bar{S}_\alpha^{NL}(v_0)\big| \leq C\sqrt{n}\,p_{\mathrm{sw}}^{\max}$ for all n large enough. Finally, note that the relevant comparison is with the one-step long-jump probability at the initialization, $p(e) := P^{\text{LJ-abs}}(v_0, e)$, which depends on v_0 through the local softmax denominator. Nevertheless, the same $1/n$ ratio holds uniformly. Indeed, from any state x, the NL switch set has only $O(1)$ available targets (at most three), each with logit at most γ, while the long-jump class consists of $\Theta(n)$ targets at the same γ-logit scale. One can check this separately in the deep middle (where there are $2n + O(1)$ long-jump targets) and at the penultimate vertex (where there are again $2n + O(1)$ long-jump targets); in both cases the ratio of the switch numerator to the long-jump numerator is $O(1/n)$, and conditioning on NL only rescales by a factor bounded away from 0 (since the forward move has probability at least ζ by Assumption 6). Consequently, for all n large enough we have the domination $p_{\mathrm{sw}}^{\max} \leq C\,p(e)/n$. Therefore,

$$\big|(1 - h_\alpha(v_0))\,\partial_\alpha \bar{S}_\alpha^{NL}(v_0)\big| \;\leq\; \frac{1}{2}\Big(-(\partial_\alpha h_\alpha(v_0))\Big(\bar{S}_\alpha^{NL}(v_0) - \tfrac{1}{2}\Big)\Big),$$

and substituting into the decomposition yields the claimed factor-$\frac{1}{2}$ dominance and $\partial_\alpha S_\alpha(v_0) \geq 0$. □

Lemma 39. *Fix a constant $K \geq 1$ independent of n and initialize at $v_0 := a_{n-K}$. For all n large enough, the long–jump term in the decomposition of Lemma 15 dominates the NL term, in the sense that*

$$\partial_\beta S_\beta(v_0) \;\leq\; -\frac{1}{8}\Big((\partial_\beta h_\beta(v_0))\Big) \leq 0.$$

Moreover, $\partial_\beta h_\beta(v_0) \geq 0$, so the right-hand side is nonpositive. In particular, $\partial_\beta S_\beta(v_0) \leq 0$, so the same–side absorption probability is (near-terminal) monotonically decreasing in β.

Proof. Recall the derivative decomposition (Lemma 15):

$$\partial_\beta S_\beta(v_0) = -(\partial_\beta h_\beta(v_0))\Big(\bar{S}_\beta^{NL}(v_0) - \tfrac{1}{2}\Big) + (1 - h_\beta(v_0))\,\partial_\beta \bar{S}_\beta^{NL}(v_0).$$

We first lower bound the long–jump contribution. By Lemma 25,

$$\partial_\beta h_\beta(v_0) \;\geq\; s_\beta(v_0) \;+\; s_\beta(v_{\mathrm{pen}})\,\Pr_{v_0}(0 < \sigma < \tau),$$

where $v_{\mathrm{pen}} := a_{n-1}$ and $\sigma := \inf\{t \geq 0 :\; X_t = v_{\mathrm{pen}}\}$. We may drop the one of the two terms and get that regardless of whther $n = 1$ or $n > 1$,

$$\partial_\beta h_\beta(v_0) \;\geq\; s_\beta(v_{\mathrm{pen}})\,\Pr_{v_0}(0 < \sigma < \tau).$$

Define the one-step long-jump probability at a state x in the long–jump–absorbing chain by

$$p(e; x) := P^{\mathrm{LJ\text{-}abs}}(x, e).$$

By the β-source identity $s_\beta(x) = p_b(x)\big(u(B(x)) - u(x)\big)$ (Lemma 21), Corollary 2 (applied at $B(v_{\mathrm{pen}}) = a_{n-1}$), and the base model lower bound $p_{\mathrm{fwd}} \geq \zeta$ (Assumption 6), we obtain that

$$s_\beta(v_{\mathrm{pen}}) \;\geq\; \zeta^K \cdot p_b(v_{\mathrm{pen}})\,p(e; v_{\mathrm{pen}}).$$

Moreover, if $v_0 = a_{n-K}$ then the event of $K-1$ consecutive forward steps implies $\{0 < \sigma < \tau\}$, hence

$$\Pr_{v_0}(0 < \sigma < \tau) \;\geq\; \zeta^{K-1}.$$

Combining,

$$\partial_\beta h_\beta(v_0) \;\geq\; \zeta^{2K-1} \cdot p_b(v_{\mathrm{pen}})\,p(e; v_{\mathrm{pen}}).$$

Moreover, by Lemma 29 applied in the NL chain at depth K, for n large enough we have $\bar{S}_\beta^{NL}(v_0) \geq \frac{3}{4}$, hence $\bar{S}_\beta^{NL}(v_0) - \frac{1}{2} \geq \frac{1}{4}$.

We next upper bound the NL contribution. By Lemma 34,

$$\big|\partial_\beta \bar{S}_\beta^{NL}(v_0)\big| \;\leq\; C\,\frac{\sqrt{n}}{\zeta}\,p_b^{(2\lfloor n^{1/4}\rfloor)}\,p_{\mathrm{sw}}^{\max} \;+\; C\,n\,\exp\!\left(-\frac{c^2}{8}\,n^{1/4}\right) \cdot \sup_{x \notin \{a_n, b_n\}} |s_{\beta,\mathrm{NL}}(x)|.$$

We now control the NL source term. By the NL β–source identity (Lemma 33) and the consecutive–value bound (Lemma 30), for every nonterminal x with backward neighbor $B(x)$,

$$|s_{\beta,\mathrm{NL}}(x)| \leq p_{\mathrm{NL}}(x \to B(x))\,|h(B(x)) - h(x)| \leq p_{\mathrm{NL}}(x \to B(x))\,p_{\mathrm{sw}}^{\max}\,\mathcal{E}_x^{NL}[\tau].$$

Therefore,

$$\sup_x |s_{\beta,\mathrm{NL}}(x)| \leq \Big(\sup_x p_{\mathrm{NL}}(x \to B(x))\Big)\,p_{\mathrm{sw}}^{\max}\,\Big(\sup_x \mathcal{E}_x^{NL}[\tau]\Big) \leq \frac{n}{\zeta}\Big(\sup_x p_{\mathrm{NL}}(x \to B(x))\Big)\,p_{\mathrm{sw}}^{\max},$$

using $\sup_x \mathcal{E}_x^{NL}[\tau] \leq n/\zeta$ under the drift gap.

Next we bound both $\sup_x p_{\mathrm{NL}}(x \to B(x))$ and $p_{\mathrm{sw}}^{\max}$ in terms of penultimate-vertex probabilities in the original / LJ-absorbing chain. By Lemma 28, NL transition probabilities are inflated by at most $1/\zeta$ relative to the original chain, so in particular

$$\sup_x p_{\mathrm{NL}}(x \to B(x)) \leq \frac{1}{\zeta}\max_x p_b(x), \qquad p_{\mathrm{sw}}^{\max} \leq \frac{1}{\zeta}\max_x p_{\mathrm{sw}}^{\mathrm{orig}}(x).$$

Moreover, in the original chain the switch class consists of $O(1)$ γ-logit targets while the long-jump class has $\Theta(n)$ γ-logit targets, hence $p_{\mathrm{sw}}^{\mathrm{orig}}(x) \le C\, p(e;x)/n$ for all x. Combining with Lemma 37 gives, for all n large enough,

$$\sup_x p_{\mathrm{NL}}(x \to B(x)) \;\le\; C\, p_b(v_{\mathrm{pen}}), \qquad p_{\mathrm{sw}}^{\max} \;\le\; C\, \frac{p(e;v_{\mathrm{pen}})}{n}.$$

Also, $p_b^{(2\lfloor n^{1/4} \rfloor)} \le \sup_x p_{\mathrm{NL}}(x \to B(x)) \le C\, p_b(v_{\mathrm{pen}})$. Substituting these bounds into the two-term upper bound from Lemma 34 yields

$$\left|\partial_\beta \bar{S}_\beta^{NL}(v_0)\right| \le C\, \frac{\sqrt{n}}{\zeta}\, p_b(v_{\mathrm{pen}}) \cdot \frac{p(e;v_{\mathrm{pen}})}{n} \le \frac{C'}{\zeta\sqrt{n}}\, p_b(v_{\mathrm{pen}})\, p(e;v_{\mathrm{pen}})$$

for all n large enough.

Therefore, for n large enough we have

$$\left|(1-h_\beta(v_0))\, \partial_\beta \bar{S}_\beta^{NL}(v_0)\right| \;\le\; \frac{1}{2}\Big((\partial_\beta h_\beta(v_0))\Big(\bar{S}_\beta^{NL}(v_0) - \tfrac{1}{2}\Big)\Big),$$

and substituting into the decomposition yields $\partial_\beta S_\beta(v_0) \le \frac{1}{2}\big(- (\partial_\beta h_\beta(v_0))(\bar{S}_\beta^{NL}(v_0) - \frac{1}{2})\big) \le 0$. □

Lemma 40. *Fix a constant $K \ge 1$ independent of n and initialize at $v_0 := a_{n-K}$. For all n large enough, the long–jump term in the decomposition of Lemma 15 dominates the NL term, in the sense that*

$$\partial_\gamma S_\gamma(v_0) \;\le\; -\frac{1}{8}\Big((\partial_\gamma h_\gamma(v_0))\Big) \le 0.$$

Moreover, $\partial_\gamma h_\gamma(v_0) \ge 0$, so the right-hand side is nonpositive. In particular, $\partial_\gamma S_\gamma(v_0) \le 0$, so the same–side absorption probability is (near-terminal) monotonically decreasing in γ.

Proof. We analyze the derivative decomposition (Lemma 15):

$$\partial_\gamma S_\gamma(v_0) = -(\partial_\gamma h_\gamma(v_0))\Big(\bar{S}_\gamma^{NL}(v_0) - \tfrac{1}{2}\Big) + (1-h_\gamma(v_0))\, \partial_\gamma \bar{S}_\gamma^{NL}(v_0).$$

We first lower bound the long–jump contribution in the same way as in the β case. Let $v_{\mathrm{pen}} := a_{n-1}$ and $\sigma := \inf\{t \ge 0 :\ X_t = v_{\mathrm{pen}}\}$ in the long–jump–absorbing chain. By Lemma 26,

$$\partial_\gamma h_\gamma(v_0) \;\ge\; s_\gamma(v_{\mathrm{pen}}) \Pr_{v_0}(0 < \sigma < \tau) \;+\; \frac{1}{2}\, s_\gamma(v_0) \;\ge\; s_\gamma(v_{\mathrm{pen}}) \Pr_{v_0}(0 < \sigma < \tau),$$

since $s_\gamma(v_0) \ge 0$. Moreover, if $v_0 = a_{n-K}$ then the event of $K-1$ consecutive forward steps implies $\{0 < \sigma < \tau\}$, hence

$$\Pr_{v_0}(0 < \sigma < \tau) \;\ge\; \zeta^{K-1}$$

by the base model lower bound $p_{\mathrm{fwd}} \ge \zeta$ (Assumption 6). Define the one-step long-jump probability at a state x in the long–jump–absorbing chain by $p(e;x) := P^{\text{LJ-abs}}(x,e)$. At the penultimate vertex, Lemma 23(C) gives

$$s_\gamma(v_{\mathrm{pen}}) \;\ge\; p(e;v_{\mathrm{pen}})\,\big(1 - u(v_{\mathrm{pen}})\big) \;\ge\; \zeta \cdot p(e;v_{\mathrm{pen}}),$$

where $u(x) = \Pr_x[X_\tau = e]$ and the last inequality uses 2. Combining these and considering cases where $n = 1$ or $n > 1$ yields $\partial_\gamma h_\gamma(v_0) \ge 0.5 \cdot \cdot p(e;v_{\mathrm{pen}})\, \zeta^K$. Moreover, by Lemma 29 applied in the NL chain at depth K, for n large enough we have $\bar{S}_\gamma^{NL}(v_0) \ge \frac{3}{4}$, hence $\bar{S}_\gamma^{NL}(v_0) - \frac{1}{2} \ge \frac{1}{4}$.

We next upper bound the NL contribution. By Lemma 36,

$$\left|\partial_\gamma \bar{S}_\gamma^{NL}(v_0)\right| \;\le\; C\, n^{1/4}\, p_{\mathrm{sw}}^{\max} + C\, \frac{\sqrt{n}}{\zeta}\, p_f^{(2\lfloor n^{1/4} \rfloor)}\, p_{\mathrm{sw}}^{\max} + C\, n\, \exp\!\left(-\frac{c^2}{8}\, n^{1/4}\right) \cdot \sup_{x \notin \{a_n, b_n\}} |s_{\gamma,\mathrm{NL}}(x)|.$$

We now control the three terms on the right-hand side using the same comparison-to-penultimate-vertex strategy as in the β case.

First, by Lemma 28, NL probabilities are inflated by at most $1/\zeta$ relative to the original chain. In the original chain, the switch class consists of $O(1)$ γ-logit targets while the long-jump class has $\Theta(n)$

γ-logit targets, so $p_{\mathrm{sw}}^{\mathrm{orig}}(x) \le C\, p(e;x)/n$ for all x. Combining with Lemma 37 gives, for all n large enough,

$$p_{\mathrm{sw}}^{\max} \;\le\; \frac{1}{\zeta} \max_x p_{\mathrm{sw}}^{\mathrm{orig}}(x) \;\le\; C\, \frac{\max_x p(e;x)}{n} \;\le\; C\, \frac{p(e;v_{\mathrm{pen}})}{n}.$$

Next, we bound the NL γ-source. By the NL γ–source bounds (Lemma 31), each $s_{\gamma,\mathrm{NL}}(x)$ is a linear combination of $p_{\mathrm{sw}}(x)$ and (at the boundary) one additional term involving a same–chain neighbor difference $|h(x^{\pm}) - h(x)|$. By Lemma 30 and $\sup_x \mathcal{E}_x^{NL}[\tau] \le n/\zeta$ under the drift gap, these neighbor differences are at most $p_{\mathrm{sw}}^{\max} \sup_x \mathcal{E}_x^{NL}[\tau] \le (n/\zeta)\, p_{\mathrm{sw}}^{\max}$. Therefore,

$$\sup_x |s_{\gamma,\mathrm{NL}}(x)| \;\le\; C\, \frac{n}{\zeta}\, p_{\mathrm{sw}}^{\max}.$$

Substituting the bound on $p_{\mathrm{sw}}^{\max}$ into the first two terms (using $p_f^{(2\lfloor n^{1/4} \rfloor)} \le 1$), and the source bound into the exponentially-suppressed third term, yields

$$\big|\partial_\gamma \bar{S}_\gamma^{NL}(v_0)\big| \;\le\; C\, \frac{n^{1/4}}{n}\, p(e;v_{\mathrm{pen}}) + C\, \frac{\sqrt{n}}{\zeta} \cdot \frac{1}{n}\, p(e;v_{\mathrm{pen}}) + C\, n\, \exp\!\left(-\frac{c^2}{8}\, n^{1/4}\right) \cdot \frac{n}{\zeta} \cdot \frac{1}{n}\, p(e;v_{\mathrm{pen}}).$$

In particular, for all n large enough,

$$\big|\partial_\gamma \bar{S}_\gamma^{NL}(v_0)\big| \;\le\; \frac{C'}{\zeta\sqrt{n}}\, p(e;v_{\mathrm{pen}}).$$

Therefore, for n large enough we have

$$\big|(1 - h_\gamma(v_0))\, \partial_\gamma \bar{S}_\gamma^{NL}(v_0)\big| \;\le\; \frac{1}{2}\Big((\partial_\gamma h_\gamma(v_0)) \Big(\bar{S}_\gamma^{NL}(v_0) - \tfrac{1}{2} \Big) \Big),$$

and substituting into the decomposition yields $\partial_\gamma S_\gamma(v_0) \le \frac{1}{2}\big(- (\partial_\gamma h_\gamma(v_0))(\bar{S}_\gamma^{NL}(v_0) - \frac{1}{2})\big) \le 0$. □

Lemma 41. *Fix a constant $K \ge 1$ independent of n, and initialize at*

$$v_0 := a_{n-K}.$$

Write $p(e)$ for the one–step probability of a long jump out of v_0 in the original chain (equivalently, $p(e) = P^{\mathrm{LJ\text{-}abs}}(v_0, e)$ in the long–jump–absorbing chain). Write $p(e;x) := P^{\mathrm{LJ\text{-}abs}}(x, e)$ and $p(e)_{\max} := \sup_{x \notin \{a_n, b_n\}} p(e;x)$. Then

$$\big|\partial_\gamma S_\gamma(v_0)\big| \;\le\; C_{1,\gamma}\, \frac{\sqrt{n}}{\zeta}\, p(e)_{\max}.$$

Proof (upper bound). Write the derivative decomposition (Lemma 15) at v_0:

$$\partial_\gamma S_\gamma(v_0) = -(\partial_\gamma h_\gamma(v_0)) \Big(\bar{S}_\gamma^{NL}(v_0) - \tfrac{1}{2} \Big) + (1 - h_\gamma(v_0))\, \partial_\gamma \bar{S}_\gamma^{NL}(v_0).$$

In the proof of Lemma 40 (the domination step), for all n large enough we showed that

$$\big|(1 - h_\gamma(v_0))\, \partial_\gamma \bar{S}_\gamma^{NL}(v_0)\big| \;\le\; \tfrac{1}{2}\, \big|(\partial_\gamma h_\gamma(v_0))\big| \cdot \Big|\bar{S}_\gamma^{NL}(v_0) - \tfrac{1}{2}\Big| \;\le\; \tfrac{1}{2}\, \big|\partial_\gamma h_\gamma(v_0)\big|.$$

Therefore,

$$\big|\partial_\gamma S_\gamma(v_0)\big| \;\le\; \tfrac{3}{2}\, \big|\partial_\gamma h_\gamma(v_0)\big|. \tag{35}$$

It remains to bound $|\partial_\gamma h_\gamma(v_0)|$ in the long–jump–absorbing chain. Let $u(x) := \Pr_x[X_\tau = e]$ denote the absorption probability at e (so $h_\gamma(\cdot) = u(\cdot)$), with $\tau := \inf\{t \ge 0 : X_t \in \{a_n, b_n, e\}\}$. By the trajectory/source representation (Lemma 16),

$$\partial_\gamma h_\gamma(v_0) = \partial_\gamma u(v_0) = \mathcal{E}_{v_0}\left[\sum_{t=0}^{\tau-1} s_\gamma(X_t) \right],$$

where s_γ is the local γ–source in the long–jump–absorbing chain. We use the standard band decomposition with $M := \lfloor n^{1/4} \rfloor$.

On $\{\tau \le M\}$, all visited states satisfy $D(X_t) \le D(v_0) + t \le 2M$ for n large enough (since $D(v_0) = K$ is fixed), hence

$$\left|\sum_{t=0}^{\tau-1} s_\gamma(X_t)\right| \mathbf{1}\{\tau \le M\} \;\le\; M \cdot \sup_{x:\, D(x)\le 2M} |s_\gamma(x)|.$$

To bound the band supremum, use the explicit γ–source formulas in the long–jump–absorbing chain (Lemma 22). In each of the finitely many cases (deep middle / first / second / penultimate), $s_\gamma(x)$ is a finite linear combination of terms of the form

$$\widetilde{P}(x,y) \cdot w_x(y),$$

where either $y = e$ (a long jump in the original chain, redirected to e) so $\widetilde{P}(x,y) = p(e;x) \le p(e)_{\max}$, or else y is one of the $O(1)$ non–long–jump γ–dependent neighbors and then $\widetilde{P}(x,y) \le 1$ and the coefficient multiplies a value term $w_x(y)$ which is a difference of absorption probabilities (or a centered version thereof). On the band $D(\cdot) \le 2M$, Lemma 19 together with the drift-gap bound $\mathcal{E}[\tau] \lesssim D/\zeta$ gives

$$u(y) = \Pr_y[X_\tau = e] \;\lesssim\; \frac{M}{\zeta}\, p(e)_{\max},$$

and hence every such value term $w_x(y)$ appearing in the case formulas is $O\big((M/\zeta)\, p(e)_{\max}\big)$. Since each case has only $O(1)$ such terms, we conclude that

$$\sup_{D(x)\le 2M} |s_\gamma(x)| \;\le\; C\, \frac{M}{\zeta}\, p(e)_{\max}$$

for a constant C. Therefore the contribution on $\{\tau \le M\}$ is at most $C\, \frac{M^2}{\zeta}\, p(e)_{\max}$.

On $\{\tau > M\}$, use the crude bound $\big|\sum_{t<\tau} s_\gamma(X_t)\big| \le \tau \sup_x |s_\gamma(x)|$ together with the tail bound from Lemma 65 (with $t = M$). The same reasoning as above, with $D(\cdot) \le n$ instead of $2M$, gives the global bound

$$\sup_x |s_\gamma(x)| \;\le\; C\, \frac{n}{\zeta}\, p(e)_{\max}.$$

Multiplying by $\Pr_{v_0}(\tau > M) = \exp(-\Omega(M))$ (with $M = \lfloor n^{1/4} \rfloor$) shows this tail contribution is negligible. Thus,

$$|\partial_\gamma h_\gamma(v_0)| \;\le\; C\, \frac{M^2}{\zeta}\, p(e)_{\max} \;=\; C\, \frac{\sqrt{n}}{\zeta}\, p(e)_{\max}.$$

Combining with (35) yields the claimed upper bound. □

F.7 Hard example contributions to derivative are negligible

Recall the derivative decomposition from Lemma 15. For a parameter $\theta \in \{\alpha, \beta, \gamma\}$ and a start state v, we have

$$\partial_\theta S_\theta(v) = -(\partial_\theta h_\theta(v))\Big(\bar{S}_\theta^{NL}(v) - \tfrac{1}{2}\Big) + (1 - h_\theta(v))\, \partial_\theta \bar{S}_\theta^{NL}(v),$$

where:

- $S_\theta(v) := \Pr_v^\theta[X_\tau = a_n]$ is the same–side absorption probability in the original chain.
- $h_\theta(v)$ is the probability of a long jump before absorption, which by definition equals the absorption probability at S_{LJ} in the long–jump–absorbing chain where long jumps are redirected to the absorbing state S_{LJ}.
- $\bar{S}_\theta^{NL}(v) := \Pr_v^\theta(X_\tau = a_n \mid NL)$ is the same–side absorption probability conditional on no long jump occurring (i.e. in the no–long–jump chain).
- By symmetry, $\bar{S}_\theta^{LJ}(v) = \frac{1}{2}$, so the long–jump conditional absorption probability equals $1/2$.

We will now argue using this decomposition that the addition of hard examples preserves the properties of the derivatives argued above. Specifically, we will show that the monotonicity and sign properties established for the derivatives $\partial_\theta S_\theta(v)$ continue to hold when hard examples are included in the training distribution.

F.7.1 Upper bound on derivative for hard examples

We will now bound the derivatives of the same-side absorption probability for hard examples.

Lemma 42. *Let S be finite and let $T \subseteq S$ be absorbing. Fix $A \subseteq T$ and a starting state $v \in S$. For $\theta \in \mathbb{R}$, let P_θ be a time-homogeneous Markov kernel on S with absorbing set T. Let*

$$\tau := \inf\{t \geq 0 : \ X_t \in T\}, \qquad R := \mathbf{1}\{X_\tau \in A\}, \qquad u_\theta(v) := \mathcal{E}_v^\theta[R].$$

For $x \notin T$ and $P_\theta(x, y) > 0$, define the one-step score

$$\phi_\theta(x, y) := \partial_\theta \log P_\theta(x, y),$$

and the stopped cumulative score

$$L := \sum_{t=0}^{\tau-1} \phi_\theta(X_t, X_{t+1}).$$

Assume that all differentiations and exchanges of derivative and expectation below are valid, and that $\mathcal{E}_v^\theta[L^2] < \infty$. Then $u_\theta(v)$ is differentiable and

$$\partial_\theta u_\theta(v) = \mathcal{E}_v^\theta[R\, L] = \mathcal{E}_v^\theta\big[(R - u_\theta(v))\, L\big] = \overset{\theta}{\underset{v}{\mathrm{Cov}}}(R, L).$$

Moreover, by Cauchy–Schwarz,

$$\big|\partial_\theta u_\theta(v)\big| \leq \sqrt{u_\theta(v)\big(1 - u_\theta(v)\big)}\ \sqrt{\mathcal{E}_v^\theta[L^2]}.$$

Proof. By definition $u_\theta(v) = \mathcal{E}_v^\theta[R]$ where $R = \mathbf{1}\{X_\tau \in A\}$ depends on the trajectory only through the absorbed terminal state. Using the log-derivative trick (score-function identity),

$$\partial_\theta u_\theta(v) = \partial_\theta \mathcal{E}_v^\theta[R] = \mathcal{E}_v^\theta\big[R\, \partial_\theta \log \overset{\theta}{\underset{v}{\Pr}}(X_0, \ldots, X_\tau)\big].$$

Since the chain is time-homogeneous and T is absorbing, the (stopped) path likelihood factors as

$$\overset{\theta}{\underset{v}{\Pr}}(X_0, \ldots, X_\tau) = \prod_{t=0}^{\tau-1} P_\theta(X_t, X_{t+1}).$$

Therefore

$$\partial_\theta \log \overset{\theta}{\underset{v}{\Pr}}(X_0, \ldots, X_\tau) = \sum_{t=0}^{\tau-1} \partial_\theta \log P_\theta(X_t, X_{t+1}) = \sum_{t=0}^{\tau-1} \phi_\theta(X_t, X_{t+1}) = L.$$

Plugging this back gives the REINFORCE form

$$\partial_\theta u_\theta(v) = \mathcal{E}_v^\theta[R\, L].$$

Next we show that L has mean zero (baseline fact). Condition on $X_t = x$ (with $x \notin T$). Then

$$\mathcal{E}_v^\theta[\phi_\theta(X_t, X_{t+1}) \mid X_t = x] = \sum_{y \in S} P_\theta(x, y)\, \partial_\theta \log P_\theta(x, y) = \sum_{y \in S} \partial_\theta P_\theta(x, y) = \partial_\theta\Big(\sum_{y \in S} P_\theta(x, y)\Big) = 0.$$

Summing over $t = 0, \ldots, \tau - 1$ and using linearity of expectation yields

$$\mathcal{E}_v^\theta[L] = \mathcal{E}_v^\theta\left[\sum_{t=0}^{\tau-1} \phi_\theta(X_t, X_{t+1})\right] = 0.$$

Hence, for the constant baseline $b := u_\theta(v) = \mathcal{E}_v^\theta[R]$,

$$\mathcal{E}_v^\theta[(R - b)L] = \mathcal{E}_v^\theta[R\, L] - b\, \mathcal{E}_v^\theta[L] = \mathcal{E}_v^\theta[R\, L].$$

Therefore

$$\partial_\theta u_\theta(v) = \mathcal{E}_v^\theta[R\, L] = \mathcal{E}_v^\theta[(R - u_\theta(v))\, L].$$

Since $\mathcal{E}_v^\theta[L] = 0$, this is exactly the covariance:

$$\mathcal{E}_v^\theta[(R - u_\theta(v))\, L] = \mathcal{E}_v^\theta[R\, L] - \mathcal{E}_v^\theta[R]\mathcal{E}_v^\theta[L] = \operatorname*{Cov}_v^\theta(R, L).$$

Finally, apply Cauchy–Schwarz:

$$\big|\partial_\theta u_\theta(v)\big| = \big|\mathcal{E}_v^\theta[(R - u_\theta(v))\, L]\big| \le \sqrt{\mathcal{E}_v^\theta[(R - u_\theta(v))^2]}\, \sqrt{\mathcal{E}_v^\theta[L^2]}.$$

Because R is Bernoulli with mean $u_\theta(v)$,

$$\mathcal{E}_v^\theta[(R - u_\theta(v))^2] = \operatorname*{Var}_v^\theta(R) = u_\theta(v)\big(1 - u_\theta(v)\big).$$

Substituting yields

$$\big|\partial_\theta u_\theta(v)\big| \le \sqrt{u_\theta(v)\big(1 - u_\theta(v)\big)}\, \sqrt{\mathcal{E}_v^\theta[L^2]}.$$

□

We have the following decomposition for $E_\theta[L^2]$:

Lemma 43. *Let S be finite and let $T \subseteq S$ be absorbing. Fix $\theta \in \mathbb{R}$ and a starting state $v \in S$. Let $(X_t)_{t\ge 0}$ be the time-homogeneous Markov chain with kernel P_θ and absorption time*

$$\tau := \inf\{t \ge 0 :\ X_t \in T\}.$$

For $x \notin T$ and $P_\theta(x, y) > 0$ define the one-step score

$$\phi_\theta(x, y) := \partial_\theta \log P_\theta(x, y),$$

and define the stopped cumulative score

$$L := \sum_{t=0}^{\tau-1} \phi_\theta(X_t, X_{t+1}).$$

Define the one-step Fisher-type quantity

$$I_\theta(x) := \sum_{y \in S} P_\theta(x, y)\, \phi_\theta(x, y)^2 \qquad (x \in S \setminus T),$$

and set $I_\theta(x) := 0$ for $x \in T$. Assume $\mathcal{E}_v^\theta[L^2] < \infty$ and $\mathcal{E}_v^\theta[L] = 0$. Then

$$\operatorname*{Var}_v^\theta(L) = \mathcal{E}_v^\theta[L^2] = \mathcal{E}_v^\theta\left[\sum_{t=0}^{\tau-1} I_\theta(X_t)\right].$$

In particular, if $I_\theta^\star := \max_{x \in S\setminus T} I_\theta(x)$, then

$$\operatorname*{Var}_v^\theta(L) \le I_\theta^\star\, \mathcal{E}_v^\theta[\tau].$$

Proof. Let $\mathcal{F}_t := \sigma(X_0, \dots, X_t)$ and define the stopped increments

$$D_t := \phi_\theta(X_t, X_{t+1})\, \mathbf{1}\{t < \tau\}, \qquad t \ge 0,$$

so that $L = \sum_{t\ge 0} D_t$.

For each $t \ge 0$ we have $\mathcal{E}_v^\theta[D_t \mid \mathcal{F}_t] = 0$. Indeed, on the event $\{t \ge \tau\}$ we have $D_t = 0$. On the event $\{t < \tau\}$ we have $X_t \notin T$, and conditioning on $X_t = x$ gives

$$\mathcal{E}_v^\theta[D_t \mid \mathcal{F}_t] = \mathcal{E}_v^\theta[\phi_\theta(X_t, X_{t+1}) \mid X_t = x] = \sum_{y \in S} P_\theta(x, y)\, \partial_\theta \log P_\theta(x, y) = \sum_{y \in S} \partial_\theta P_\theta(x, y) = 0.$$

Hence $(D_t)_{t\ge 0}$ is a martingale-difference sequence.

Expanding L^2 gives

$$L^2 = \left(\sum_{t\geq 0} D_t\right)^2 = \sum_{t\geq 0} D_t^2 + 2\sum_{0\leq s<t} D_s D_t.$$

We claim $\mathcal{E}_v^\theta[D_s D_t] = 0$ for all $0 \leq s < t$. Since D_s is $\mathcal{F}_t$-measurable when $s < t$, and using the tower property,

$$\mathcal{E}_v^\theta[D_s D_t] = \mathcal{E}_v^\theta\big[\mathcal{E}_v^\theta[D_s D_t \mid \mathcal{F}_t]\big] = \mathcal{E}_v^\theta\big[D_s\, \mathcal{E}_v^\theta[D_t \mid \mathcal{F}_t]\big] = 0,$$

by the martingale-difference property. Therefore

$$\mathcal{E}_v^\theta[L^2] = \sum_{t\geq 0} \mathcal{E}_v^\theta[D_t^2].$$

For each t,

$$\mathcal{E}_v^\theta[D_t^2] = \mathcal{E}_v^\theta\big[\mathcal{E}_v^\theta[D_t^2 \mid \mathcal{F}_t]\big] = \mathcal{E}_v^\theta\big[\mathbf{1}\{t<\tau\}\, \mathcal{E}_v^\theta[\phi_\theta(X_t, X_{t+1})^2 \mid X_t]\big].$$

On $\{t < \tau\}$, $X_t \notin T$, and conditioning on $X_t = x$ yields

$$\mathcal{E}_v^\theta[\phi_\theta(X_t, X_{t+1})^2 \mid X_t = x] = \sum_{y\in S} P_\theta(x,y)\, \phi_\theta(x,y)^2 = I_\theta(x).$$

Hence $\mathcal{E}_v^\theta[D_t^2] = \mathcal{E}_v^\theta[\mathbf{1}\{t<\tau\} I_\theta(X_t)]$, and summing over t gives

$$\mathcal{E}_v^\theta[L^2] = \sum_{t\geq 0} \mathcal{E}_v^\theta[\mathbf{1}\{t<\tau\} I_\theta(X_t)] = \mathcal{E}_v^\theta\left[\sum_{t=0}^{\tau-1} I_\theta(X_t)\right].$$

Since $\mathcal{E}_v^\theta[L] = 0$, we have $\mathrm{Var}_v^\theta(L) = \mathcal{E}_v^\theta[L^2]$, proving the main identity.

Finally, $I_\theta(X_t) \leq I_\theta^\star$ for $t < \tau$, so

$$\sum_{t=0}^{\tau-1} I_\theta(X_t) \leq I_\theta^\star\, \tau,$$

and taking expectations yields $\mathrm{Var}_v^\theta(L) \leq I_\theta^\star\, \mathcal{E}_v^\theta[\tau]$. □

We now bound these single step terms for the LJ absorbing chain:

Lemma 44. *Work in the long-jump absorbing (LJ) chain with the softmax logit structure. For each $\theta \in \{\alpha, \beta, \gamma\}$ and each nonterminal state $x \notin T$, let*

$$I_\theta(x) \;:=\; \sum_{a\in\mathcal{A}(x)} P_{\alpha,\beta,\gamma}(x,a)\Big(\partial_\theta \log P_{\alpha,\beta,\gamma}(x,a)\Big)^2$$

denote the one-step Fisher-type term. Then

$$I_\theta(x) \leq 1 \qquad \textit{for all } \theta \in \{\alpha,\beta,\gamma\} \textit{ and all } x \notin T.$$

Consequently, $I_\theta^\star := \max_{x\notin T} I_\theta(x) \leq 1$ for each $\theta \in \{\alpha, \beta, \gamma\}$.

Proof. Fix $\theta \in \{\alpha, \beta, \gamma\}$ and a nonterminal state $x \notin T$. Write the allowed moves out of x as $\mathcal{A}(x)$ and write the outgoing probabilities in softmax form

$$p_a(\theta) := P_{\alpha,\beta,\gamma}(x,a) = \frac{e^{\ell_a(\theta)}}{Z(\theta)}, \qquad Z(\theta) := \sum_{b\in\mathcal{A}(x)} e^{\ell_b(\theta)}.$$

In the LJ logit structure, after the usual normalization for γ (subtracting a common γ from all logits, which does not change the softmax probabilities), the dependence on θ is affine with coefficient bounded by 1:

$$\ell_a(\theta) = \ell_a^{(0)} + k_a\,\theta \qquad \text{with} \qquad |k_a| \leq 1.$$

Define $\bar{k}(\theta) := \sum_{b \in \mathcal{A}(x)} p_b(\theta)\, k_b$. Differentiating $\log p_a(\theta) = \ell_a(\theta) - \log Z(\theta)$ gives

$$\partial_\theta \log p_a(\theta) = \partial_\theta \ell_a(\theta) - \frac{Z'(\theta)}{Z(\theta)} = k_a - \bar{k}(\theta),$$

where we used

$$\frac{Z'(\theta)}{Z(\theta)} = \frac{\sum_b e^{\ell_b(\theta)} \partial_\theta \ell_b(\theta)}{\sum_b e^{\ell_b(\theta)}} = \sum_b \frac{e^{\ell_b(\theta)}}{Z(\theta)}\, k_b = \bar{k}(\theta).$$

Therefore

$$I_\theta(x) = \sum_{a \in \mathcal{A}(x)} p_a(\theta) \big(\partial_\theta \log p_a(\theta)\big)^2 = \sum_{a \in \mathcal{A}(x)} p_a(\theta)\, (k_a - \bar{k}(\theta))^2 = \operatorname*{Var}_{a \sim p(\theta)}(k_a).$$

Finally, $\mathrm{Var}(k) \le \mathcal{E}[k^2]$ and $|k_a| \le 1$ imply

$$\operatorname*{Var}_{a \sim p(\theta)}(k_a) \le \mathcal{E}_{a \sim p(\theta)}[k_a^2] \le 1,$$

hence $I_\theta(x) \le 1$, as claimed. Taking the maximum over $x \notin T$ yields $I_\theta^\star \le 1$. □

Lemma 45. *Work in the long–jump–absorbing (LJ) chain (Definition 16) with absorbing set $T = \{a_n, b_n, e\}$. Fix $\theta \in \{\alpha, \beta, \gamma\}$ and define the long-jump absorption probability*

$$u_\theta(v) := \overset{\theta}{\Pr_v}(X_\tau = e), \qquad \tau := \inf\{t \ge 0 :\ X_t \in T\}.$$

Assume that for all starting states $v \notin T$,

$$\mathcal{E}_v^\theta[\tau] \ \le\ \frac{n}{\zeta},$$

for some constant $\zeta > 0$ independent of n. Then, for every $v \notin T$,

$$\big|\partial_\theta u_\theta(v)\big| \ \le\ \sqrt{u_\theta(v)\big(1 - u_\theta(v)\big)} \cdot \sqrt{\frac{n}{\zeta}}$$

Proof. Apply Lemma 42 with $A = \{e\}$. This gives

$$|\partial_\theta u_\theta(v)| \le \sqrt{u_\theta(v)\big(1 - u_\theta(v)\big)}\ \sqrt{\mathcal{E}_v^\theta[L^2]}.$$

By Lemma 43,

$$\mathcal{E}_v^\theta[L^2] = \mathcal{E}_v^\theta\left[\sum_{t=0}^{\tau-1} I_\theta(X_t)\right] \le I_\theta^\star\, \mathcal{E}_v^\theta[\tau].$$

Finally, Lemma 44 gives $I_\theta^\star \le 1$, and by assumption $\mathcal{E}_v^\theta[\tau] \le n/\zeta$. Combining yields

$$|\partial_\theta u_\theta(v)| \le \sqrt{u_\theta(v)\big(1 - u_\theta(v)\big)}\ \sqrt{\frac{n}{\zeta}}.$$

□

Lemma 46. *Fix a constant $a \in (0,1)$ and consider the original two–chain model of length n, together with its long–jump–absorbing chain (Definition 16). Assume there exists a constant $q_{\min} \in (0,1)$, independent of n, such that the one-step long-jump probability satisfies*

$$p(e; x) := P^{\text{LJ-abs}}(x, e) \ \ge\ q_{\min} \qquad \textit{for all nonterminal } x \notin \{a_n, b_n\}.$$

Let v be any start state with depth $D(v) \ge an$. Then for each parameter $\theta \in \{\alpha, \beta, \gamma\}$ there exist constants $C, c > 0$ (depending only on fixed parameters, a, and $q_{\min}$, but not on n) such that

$$\big|\partial_\alpha S_\alpha(v)\big| \ \le\ C\, \frac{n}{\zeta}\, p_f^{\max}\, e^{-cn}, \qquad \big|\partial_\beta S_\beta(v)\big| \ \le\ C\, \frac{n}{\zeta}\, p_b^{\max}\, e^{-cn}, \qquad \big|\partial_\gamma S_\gamma(v)\big| \ \le\ C\, \frac{n}{\zeta}\, p(e)_{\max}\, e^{-cn},$$

where $\zeta > 0$ is the drift-gap constant from Assumption 6,

$$p_f^{\max} := \sup_{x \notin \{a_n, b_n\}} \Pr(X_1 = F(x) \mid X_0 = x),$$
$$p_b^{\max} := \sup_{x \notin \{a_n, b_n\}} \Pr(X_1 = B(x) \mid X_0 = x),$$
$$p(e)_{\max} := \sup_{x \notin \{a_n, b_n\}} p(e; x),$$

and $F(x), B(x)$ denote the same–chain forward/backward neighbors when they exist (with the convention that the corresponding probability is 0 if the neighbor does not exist).

Proof. Let $\tau := \inf\{t \geq 0 : \ X_t \in \{a_n, b_n\}\}$ be the absorption time in the original chain and let $NL := \{J > \tau\}$ be the event of no long jump before absorption (with J as in Lemma 15). In the long–jump–absorbing chain (Definition 16) with absorption time

$$\tau_{LJ} := \inf\{t \geq 0 : \ X_t \in \{a_n, b_n, e\}\},$$

one has $NL = \{X_{\tau_{LJ}} \in \{a_n, b_n\}\}$ and $LJ = \{X_{\tau_{LJ}} = e\}$.

We use the long–jump / no–long–jump *derivative* decomposition (Lemma 15): for each $\theta \in \{\alpha, \beta, \gamma\}$,

$$\partial_\theta S_\theta(v) = -(\partial_\theta h_\theta(v))\Big(\bar{S}_\theta^{NL}(v) - \tfrac{1}{2}\Big) + (1 - h_\theta(v))\, \partial_\theta \bar{S}_\theta^{NL}(v),$$

where $h_\theta(v) = \Pr_v^\theta(LJ)$ and $1 - h_\theta(v) = \Pr_v^\theta(NL)$.

Since $D(v) \geq an$, on NL the depth can decrease by at most 1 per step, so $\tau \geq an$. Under the assumption $p(e; x) \geq q_{\min}$ for every nonterminal x, this gives

$$\overset{\theta}{\Pr_v}(NL) \leq \overset{\theta}{\Pr_v}(\text{no hit of } e \text{ in the first } an \text{ steps}) \leq (1 - q_{\min})^{an} \leq e^{-cn}$$

for some $c > 0$ depending only on $a, q_{\min}$.

Using $|\bar{S}_\theta^{NL}(v) - \frac{1}{2}| \leq \frac{1}{2}$, we have

$$\big|\partial_\theta S_\theta(v)\big| \leq \tfrac{1}{2}\,\big|\partial_\theta h_\theta(v)\big| + \overset{\theta}{\Pr_v}(NL) \cdot \big|\partial_\theta \bar{S}_\theta^{NL}(v)\big|.$$

Now $\partial_\theta h_\theta(v) = \partial_\theta \Pr_v^\theta(LJ)$ is the derivative of the long-jump absorption probability in the long–jump–absorbing chain (with absorbing set $T = \{a_n, b_n, e\}$ and target $A = \{e\}$). Therefore, applying Lemma 45 yields

$$|\partial_\theta h_\theta(v)| \leq c\sqrt{h_\theta(v)\big(1 - h_\theta(v)\big)} \cdot \sqrt{\frac{n}{\zeta}} = c\sqrt{\overset{\theta}{\Pr_v}(LJ)\,\overset{\theta}{\Pr_v}(NL)} \cdot \sqrt{\frac{n}{\zeta}} \leq C\sqrt{\frac{n}{\zeta}}\, e^{-c'n},$$

where we used $\Pr_v^\theta(NL) \leq e^{-cn}$ and absorbed constants by possibly decreasing c.

It remains to control $|\partial_\theta \bar{S}_\theta^{NL}(v)|$ by the relevant one-step probability scale. By the source identities in the NL chain (Lemmas 32, 33, and the corresponding γ-source bound Lemma 31), the NL sources satisfy $|s_{\alpha,NL}(x)| \leq C\, p_f(x)$, $|s_{\beta,NL}(x)| \leq C\, p_b(x)$, and $|s_{\gamma,NL}(x)| \leq C\, p_{\mathrm{sw}}(x) \leq C\, p(e)_{\max}$. Therefore, by the trajectory/source representation (Lemma 16) and the drift-gap absorption time bound $\mathcal{E}_v^{NL}[\tau] \leq n/\zeta$, we have

$$|\partial_\alpha \bar{S}_\alpha^{NL}(v)| \leq C\,\frac{n}{\zeta}\, p_f^{\max}, \qquad |\partial_\beta \bar{S}_\beta^{NL}(v)| \leq C\,\frac{n}{\zeta}\, p_b^{\max}, \qquad |\partial_\gamma \bar{S}_\gamma^{NL}(v)| \leq C\,\frac{n}{\zeta}\, p(e)_{\max}.$$

Multiplying by $\Pr_v^\theta(NL) \leq e^{-cn}$ and combining with the bound on $|\partial_\theta h_\theta(v)|$ yields the stated result. □

F.8 Easy+Hard examples derivative estimates

We now apply the results of the previous subsections to derive our three main derivative bounds that will be used under the training-distribution assumptions of Theorem 2.

Lemma 47. *Assume the training distribution $\mathcal{Q}$ satisfies the assumptions of Theorem 2, and that there exists a constant $q_{\min} \in (0,1)$, independent of n, such that $P^{\text{LJ-abs}}(x,e) \geq q_{\min}$ for all nonterminal x. Let $v_i := a_i$ (equivalently, by symmetry one may take $v_i := b_i$). Define*

$$p(e)_{\max} := \max_{x \text{ nonterminal}} P^{\text{LJ-abs}}(x,e).$$

Then for all n large enough,

$$\mathcal{E}_{i\sim\mathcal{Q}}\big[\partial_\gamma S_\gamma(v_i)\big] \leq 0,$$

and there exist constants $C_{1,\gamma}, C_{2,\gamma} > 0$ (independent of n and c) such that

$$c \cdot C_{2,\gamma}\, p(e)_{\max} \;\leq\; \Big|\mathcal{E}_{i\sim\mathcal{Q}}\big[\partial_\gamma S_\gamma(v_i)\big]\Big| \;\leq\; c \cdot C_{1,\gamma}\,\sqrt{n}\, p(e)_{\max}.$$

Proof. Let $v_i := a_i$. Define the index sets

$$\mathcal{E} := \{n-k, \ldots, n-1\} \quad \text{(easy)}, \qquad \mathcal{H} := \{1, \ldots, \lfloor (1-a)n \rfloor\} \quad \text{(hard)}.$$

Since $\mathcal{Q}$ has s-simplicity mass c we have $\mathcal{Q}(\mathcal{E}) = c$, and since $\mathcal{Q}$ also has $\lceil an \rceil$-simplicity mass c, we can decompose the expected derivative as

$$\mathcal{E}_{i\sim\mathcal{Q}}[\partial_\gamma S_\gamma(v_i)] = \mathcal{Q}(\mathcal{E}) \cdot \mathcal{E}_{i\sim\mathcal{Q}}[\partial_\gamma S_\gamma(v_i) \mid i \in \mathcal{E}] + \mathcal{Q}(\mathcal{H}) \cdot \mathcal{E}_{i\sim\mathcal{Q}}[\partial_\gamma S_\gamma(v_i) \mid i \in \mathcal{H}]. \tag{36}$$

By Lemma 40, $\partial_\gamma S_\gamma(v_i) \leq 0$ for all n large enough, and moreover

$$C_{2,\gamma}\, p(e)_{\max} \;\leq\; |\partial_\gamma S_\gamma(v_i)| \;\leq\; C_{1,\gamma}\,\sqrt{n}\, p(e)_{\max},$$

where the constants $C_{1,\gamma}, C_{2,\gamma} > 0$ depend only on k (not on n or c). Therefore, the easy-example conditional expectation satisfies

$$C_{2,\gamma}\, p(e)_{\max} \;\leq\; \Big|\mathcal{E}_{i\sim\mathcal{Q}}[\partial_\gamma S_\gamma(v_i) \mid i \in \mathcal{E}]\Big| \;\leq\; C_{1,\gamma}\,\sqrt{n}\, p(e)_{\max}. \tag{37}$$

For each $i \in \mathcal{H}$, the depth satisfies $D(v_i) = n - i \geq an$. By Lemma 46, for all n large enough,

$$\big|\mathcal{Q}(\mathcal{H}) \cdot \mathcal{E}_{i\sim\mathcal{Q}}[\partial_\gamma S_\gamma(v_i) \mid i \in \mathcal{H}]\big| \leq C\,\frac{n}{\zeta}\, p(e)_{\max}\, e^{-cn}. \tag{38}$$

Combining (37) and (38) via the triangle inequality, and redefining the constants $C_{1,\gamma}, C_{2,\gamma}$, we obtain the stated bounds. □

Lemma 48. *Assume the training distribution $\mathcal{Q}$ satisfies the assumptions of Theorem 2, and that there exists a constant $q_{\min} \in (0,1)$, independent of n, such that $P^{\text{LJ-abs}}(x,e) \geq q_{\min}$ for all nonterminal x. Let $v_i := a_i$ (equivalently, by symmetry one may take $v_i := b_i$). Define*

$$p(e)_{\max} := \max_{x \text{ nonterminal}} P^{\text{LJ-abs}}(x,e), \qquad p(b)_{\max} := \max_{x \text{ nonterminal}} p_b(x),$$

where $p_b(x)$ denotes the one–step same–chain backward probability from x in the original chain. Then for all n large enough,

$$\mathcal{E}_{i\sim\mathcal{Q}}\big[\partial_\beta S_\beta(v_i)\big] \leq 0,$$

and there exists a constant $C_{2,\beta} > 0$ (independent of n) such that

$$\Big|\mathcal{E}_{i\sim\mathcal{Q}}\big[\partial_\beta S_\beta(v_i)\big]\Big| \;\geq\; c \cdot C_{2,\beta}\, p(e)_{\max}\, p(b)_{\max}.$$

Proof. As in Lemma 47, we decompose

$$\mathcal{E}_{i\sim\mathcal{Q}}[\partial_\beta S_\beta(v_i)] = \mathcal{Q}(\mathcal{E}) \cdot \mathcal{E}[\partial_\beta S_\beta(v_i) \mid i \in \mathcal{E}] + \mathcal{Q}(\mathcal{H}) \cdot \mathcal{E}[\partial_\beta S_\beta(v_i) \mid i \in \mathcal{H}].$$

For easy examples, Lemmas 39 and 25 give $\partial_\beta S_\beta(v_i) \leq 0$ and $|\partial_\beta S_\beta(v_i)| \geq C_{2,\beta}\, p(e)_{\max}\, p(b)_{\max}$. For hard examples, Lemma 46 gives $|\partial_\beta S_\beta(v_i)| \leq C\,\frac{n}{\zeta}\, p_b^{\max}\, e^{-cn}$. Since this is exponentially small, combining via the triangle inequality and redefining $C_{2,\beta}$ yields the stated bounds. □

Lemma 49. *Assume the training distribution $\mathcal{Q}$ satisfies the assumptions of Theorem 2, and that there exists a constant $q_{\min} \in (0,1)$, independent of n, such that $P^{\text{LJ-abs}}(x,e) \geq q_{\min}$ for all nonterminal x. Let $v_i := a_i$ (equivalently, by symmetry one may take $v_i := b_i$). Define*

$$p(e)_{\max} := \max_{x \text{ nonterminal}} P^{\text{LJ-abs}}(x,e), \qquad p(f)_{\min} := \min_{x \text{ nonterminal}} p_f(x),$$

where $p_f(x)$ denotes the one–step same–chain forward probability from x in the original chain. Then for all n large enough,

$$\mathcal{E}_{i\sim\mathcal{Q}}\big[\partial_\alpha S_\alpha(v_i)\big] \geq 0,$$

and there exists a constant $C_{2,\alpha} > 0$ (independent of n) such that

$$\Big|\mathcal{E}_{i\sim\mathcal{Q}}\big[\partial_\alpha S_\alpha(v_i)\big]\Big| \geq c \cdot\ C_{2,\alpha}\, p(e)_{\max}\, p(f)_{\min}.$$

Proof. As in Lemma 47, we decompose

$$\mathcal{E}_{i\sim\mathcal{Q}}[\partial_\alpha S_\alpha(v_i)] = \mathcal{Q}(\mathcal{E}) \cdot \mathcal{E}[\partial_\alpha S_\alpha(v_i) \mid i \in \mathcal{E}] + \mathcal{Q}(\mathcal{H}) \cdot \mathcal{E}[\partial_\alpha S_\alpha(v_i) \mid i \in \mathcal{H}].$$

For easy examples, Lemmas 38 and 24 give $\partial_\alpha S_\alpha(v_i) \geq 0$ and $|\partial_\alpha S_\alpha(v_i)| \geq C_{2,\alpha}\, p(e)_{\max}\, p(f)_{\min}$. For hard examples, Lemma 46 gives $|\partial_\alpha S_\alpha(v_i)| \leq C\, \frac{n}{\zeta}\, p_f^{\max}\, e^{-cn}$. Since this is exponentially small, combining via the triangle inequality and redefining $C_{2,\alpha}$ yields the stated bounds. □

F.9 Convergence rate

Main proof We now present the proof of Theorem 2, bringing together the components developed in the previous subsections.

Proof of Theorem 2. By Lemma 61, the dynamics of the attention parameters are governed by the gradients of the reduced loss $S := \tilde{L}(\alpha,\beta,\gamma)$. Since the loss S is an expectation over the starting vertex location v (sampled from Q), we write $S = \mathcal{E}_{v\sim Q}[S_v(\alpha,\beta,\gamma)]$, where S_v is the loss term for a specific v. Substituting this into the dynamics from Lemma 61 and taking absolute values, we have with $|V| = 2n$:

$$\begin{aligned}
|\dot{\alpha}(t)| &= \frac{1}{2n(2n-1)}\ \left|\mathcal{E}_v[\partial_\alpha S_v]\right|, \\
|\dot{\beta}(t)| &= \frac{1}{2n(2n-1)}\ \left|\mathcal{E}_v[\partial_\beta S_v]\right|, \\
|\dot{\gamma}(t)| &= \frac{1}{(2n-2)\cdot 2n(2n-1)}\ \left|\mathcal{E}_v[\partial_\gamma S_v]\right|.
\end{aligned}$$

We apply Lemmas 47 to 49 to bound the derivatives. These lemmas require the assumption that $p(e)_{\max}(t) \geq q_{\min}$ for some constant $q_{\min} > 0$ independent of n. Under this assumption, they imply $\mathcal{E}_{v\sim Q}[\partial_\alpha S_v] \geq 0$, $\mathcal{E}_{v\sim Q}[\partial_\beta S_v] \leq 0$, and $\mathcal{E}_{v\sim Q}[\partial_\gamma S_v] \leq 0$. Since Lemma 61 gives $\dot{\theta} = -c_\theta \partial_\theta S$ for positive constants c_θ, we have $\dot{\alpha}(t) \geq 0$, $\dot{\beta}(t) \leq 0$, and $\dot{\gamma}(t) \leq 0$.

We will from now assume that $p(e)_{\max}(t) \geq q_{\min}$ while analyzing the dynamics of the parameters α,β,γ until some time t_0 (to be determined later). Later, we will show that for all $t \leq t_0$ indeed $p(e)_{\max}(t) \geq q_{\min}$.

We define the extremal probabilities at time t:

$$p(e)_{\max}(t) := \max x \text{ nonterminal} P^{\text{LJ-abs}}(x,e;t), \qquad p(f)_{\min}(t) := \min_{x \text{ nonterminal}} p_f(x;t),$$

where the dependence on t comes through the parameters $\alpha(t),\beta(t),\gamma(t)$.

From Lemmas 47 and 49, the expected derivatives satisfy:

- $\left|\mathcal{E}_{v\sim Q}[\partial_\alpha S_v]\right| \geq c \cdot C_{2,\alpha}\, p(e)_{\max}(t)\, p(f)_{\min}(t)$,
- $c \cdot C_{2,\gamma}\, p(e)_{\max}(t) \leq \left|\mathcal{E}_{v\sim Q}[\partial_\gamma S_v]\right| \leq c \cdot C_{1,\gamma}\, \sqrt{n}\, p(e)_{\max}(t)$.

Next, we derive a lower bound on $|\dot{\alpha}(t)|$. From the dynamics and the lower bound on the expected α-derivative:

$$|\dot{\alpha}(t)| \geq \frac{c \cdot C_{2,\alpha}}{2n(2n-1)} \cdot p(e)_{\max}(t)\, p(f)_{\min}(t).$$

By the softmax structure of the attention logits (Lemma 68), there exist constants $D > 0$ and $C > 0$ (independent of n) such that

$$p(f)_{\min}(0) = \frac{e^{\alpha(0)-D}}{e^{\alpha(0)-D} + C}.$$

Moreover, since $\dot{\beta}, \dot{\gamma} \leq 0$, we have that

$$p(f)_{\min}(t) \geq \frac{e^{\alpha(t)-D}}{e^{\alpha(t)-D} + C} =: g(t).$$

By assumption (Section 5.2), at initialization we have $p(f)_{\min}(0) \geq \zeta$.

Define

$$q_{\min} := \frac{1}{8} \cdot \frac{\varepsilon\, \zeta}{(1-\varepsilon)(1-\zeta)} \cdot \upsilon,$$

where $\upsilon > 0$ is the lower bound on the initial switch probability from Section 5.2.

Under the assumption that $p(e)_{\max}(t) \geq q_{\min}$, we have

$$|\dot{\alpha}(t)| \geq \frac{c \cdot C_{2,\alpha}\, q_{\min}}{2n(2n-1)} \cdot p(f)_{\min}(t).$$

Since $p(f)_{\min}(t) \geq g(t)$, we may apply Lemma 67 to conclude that there exists a time

$$t_0 < \frac{16\, n(2n-1)}{c \cdot C_{2,\alpha}\, \upsilon} \cdot \frac{(1-\varepsilon)(1-\zeta)}{\varepsilon\, \zeta} \cdot \log\left(\frac{1-\varepsilon}{\varepsilon} \cdot \frac{1-\zeta}{\zeta}\right) \tag{39}$$

$$< O(\frac{n^2}{c \cdot \varepsilon^2}) \tag{40}$$

such that $p(f)_{\min}(t_0) \geq 1 - \varepsilon$ and

$$\alpha(t_0) - \alpha(0) < \log\left(\frac{1-\varepsilon}{\varepsilon} \cdot \frac{1-\zeta}{\zeta}\right). \tag{41}$$

All that's left to show is that indeed for any $t \leq t_0$, $p(e)_{\max}(t) \geq q_{\min}$ for some constant $q_{\min} > 0$ independent of n. Assume the contrary, then there exists a minimal time $0 \leq t_1 < t_0$ such that $p(e)_{\max}(t_1) < q_{\min}$. First, we now show that for any constant $1 > a_1 > 0$ independent of n, to be determined later, for sufficiently large n, for any $t \in [0, t_1]$, $|\dot{\gamma}(t)| < a_1 |\dot{\alpha}(t)|$.

Using the bounds from the lemmas:

$$|\dot{\alpha}(t)| \geq \frac{c \cdot C_{2,\alpha}}{2n(2n-1)} \cdot p(e)_{\max}(t)\, p(f)_{\min}(t),$$

and

$$|\dot{\gamma}(t)| \leq \frac{c \cdot C_{1,\gamma}\, \sqrt{n}}{(2n-2) \cdot 2n(2n-1)} \cdot p(e)_{\max}(t).$$

Taking the ratio and using that the $p(e)_{\max}(t)$ terms cancel:

$$\frac{|\dot{\gamma}(t)|}{|\dot{\alpha}(t)|} \leq \frac{C_{1,\gamma}}{C_{2,\alpha}(2n-2)} \cdot \frac{\sqrt{n}}{p(f)_{\min}(t)}.$$

By assumption (Section 5.2), the model is initialized such that $p(f)_{\min}(0) \geq \zeta$, together with the logit structure (Lemma 68) and the fact that $\dot{\alpha} \geq 0$ and $\dot{\beta}, \dot{\gamma} \leq 0$ implies that $p(f)_{\min}(t) \geq \zeta$ for all t if n is sufficiently large.

Consequently, for sufficiently large n, we have $|\dot{\gamma}(t)| < a_1 |\dot{\alpha}(t)|$.

This inequality allows us to apply Lemma 66 with $x(t) = \alpha(t), y(t) = \gamma(t)$ and $c = 1/a_1$. Note that $\dot{\alpha} \geq 0, \dot{\gamma} \leq 0$, and $\dot{\alpha} \geq -\dot{\gamma}/a_1$. This yields

$$\alpha(t) - \alpha(0) \geq \frac{1}{a_1}(\gamma(0) - \gamma(t)).$$

Rearranging and exponentiating gives

$$e^{\gamma(t)} \geq e^{\gamma(0)+a_1(\alpha(0)-\alpha(t))}. \tag{42}$$

By Equation (41), for all $t \leq t_1$ we have that

$$\alpha(t) - \alpha(0) < \log\left(\frac{1-\varepsilon}{\varepsilon} \cdot \frac{1-\zeta}{\zeta}\right),$$

which is a constant independent of n. Therefore, we can choose $a_1 > 0$ sufficiently small (independent of n) such that for all $t \leq t_0$,

$$e^{\gamma(t)} \geq e^{\gamma(0)+a_1(\alpha(0)-\alpha(t))} \geq e^{\gamma(0)-a_1\log\left(\frac{1-\varepsilon}{\varepsilon}\cdot\frac{1-\zeta}{\zeta}\right)} \geq \frac{1}{2}e^{\gamma(0)}.$$

By the logit structure (Lemma 68), and the fact that $\alpha(t) > \beta(t), \gamma(t), 0$ at init and throughout the run (by signs of the derivatives), we have that for any $t \leq t_1$:

$$\begin{aligned} p(e)_{\max}(t) &\geq \frac{1}{2} \cdot \frac{ne^{\gamma(t)}}{2ne^{\gamma(t)} + e^{\beta(t)} + e^{\alpha(t)}} \\ &\geq \frac{1}{4} \cdot \frac{ne^{\gamma(0)}}{2ne^{\gamma(0)} + e^{\beta(0)} + e^{\alpha(0)+(\alpha(t)-\alpha(0))}} \end{aligned}$$

By Equation (41), we have $\alpha(t) - \alpha(0) < \log\left(\frac{1-\varepsilon}{\varepsilon} \cdot \frac{1-\zeta}{\zeta}\right)$:

$$\begin{aligned} p(e)_{\max}(t) \geq& \frac{1}{4} \cdot \frac{\varepsilon\,\zeta}{(1-\varepsilon)(1-\zeta)} \cdot \frac{ne^{\gamma(0)}}{2ne^{\gamma(0)} + e^{\beta(0)} + e^{\alpha(0)}} \\ \geq& \frac{1}{8} \cdot \frac{\varepsilon\,\zeta}{(1-\varepsilon)(1-\zeta)} \cdot p(e)_{\max}(0) \\ \geq& \frac{1}{8} \cdot \frac{\varepsilon\,\zeta}{(1-\varepsilon)(1-\zeta)} \cdot \upsilon \;=\; q_{\min}. \end{aligned}$$

Thus $p(e)_{\max}(t) \geq q_{\min}$ for all $t \leq t_1$, which contradicts the assumption that $p(e)_{\max}(t_1) < q_{\min}$, as required. □

F.10 Extension to discrete-time gradient descent

We briefly justify the claim in Footnote 10 (Section 5.3) that the gradient flow guarantees of Theorem 2 extend to discrete-time policy gradient with a fixed step size $\eta > 0$.

By Lemma 61, the full attention-matrix dynamics reduce to dynamics on the three scalar parameters (α, β, γ). The analysis in Appendices F.6 and F.8 establishes that the expected gradients satisfy definite sign conditions—$\partial_\alpha S \geq 0$, $\partial_\beta S \leq 0$, $\partial_\gamma S \leq 0$ (Lemmas 38 to 40)—together with quantitative lower bounds on their magnitudes (Lemmas 47 to 49), provided two conditions hold: (i) the forward bias assumption $p_{\text{fwd}} > p_{\text{bwd}} + \zeta$ (Assumption 6), and (ii) the long-jump probability remains bounded away from zero, $p(e)_{\max}(t) \geq q_{\min}$. Crucially, these sign conditions are properties of the *loss landscape* at the current parameter values, not of the continuous-time parametrization. Therefore they apply equally to discrete gradient descent with any step size $\eta > 0$: at each iteration, $\alpha_{k+1} = \alpha_k + \eta\,\nabla_\alpha \mathcal{R} \geq \alpha_k$, and similarly $\beta_{k+1} \leq \beta_k$ and $\gamma_{k+1} \leq \gamma_k$. The monotonicity of the parameters is thus an immediate corollary of the gradient sign conditions, with no ODE approximation required. Moreover, the same self-reinforcing structure used in the gradient flow proof (Appendix F.9) carries over: since α is non-decreasing and β, γ are non-increasing, condition (i) is maintained (in fact strengthened) along the discrete trajectory; and the verification that condition (ii) holds throughout $[0, t_0]$ (carried out in the final part of the proof of Theorem 2 in Appendix F.9) depends only on the monotonicity of the parameters and the bounded total change in α, both of which transfer identically to the discrete setting. The quantitative lower bounds on gradient magnitudes from Lemmas 47 and 49 then imply that the number of discrete iterations to reach the target accuracy is $O(n^2 c^{-1} \varepsilon^{-2} \eta^{-1})$—polynomial in all relevant quantities.

G Proof of Theorem 3

As in Appendix F.9, we define the loss as $\mathcal{L} := 1 - S$, where S is the same-side absorption probability (the reward). Gradient flow on $\mathcal{L}$ is equivalent to gradient ascent on S.

The proof proceeds in four stages. First, we establish general total-variation bounds showing that small parameter changes induce small changes in trajectory distributions (Lemma 50–Corollary 3). Second, we bound the absorption time of the Markov chain (Lemma 56–Lemma 59). Third, we show that under the hard-example assumption, the loss gradient is exponentially small in the chain length n (Lemma 60) at initialization. Finally, we combine these ingredients, showing that for exponentially long times, the derivative bounds remain valid and hence the induced distribution over rollouts remains statistically indistinguishable from the rollout distribution at initialization.

G.1 Total-variation bounds for trajectory distributions

The first lemma records a data processing (pushforward contraction) property of total variation on finite spaces.

Lemma 50. *Let $\Omega, \mathcal{S}$ be finite sets. Let P, Q be probability measures on Ω, and let $f : \Omega \to \mathcal{S}$ be any map. Denote the pushforward (image) measures by*

$$(P \circ f^{-1})(B) := P\big(f^{-1}(B)\big), \qquad (Q \circ f^{-1})(B) := Q\big(f^{-1}(B)\big), \qquad B \subseteq \mathcal{S}.$$

Then

$$\|P \circ f^{-1} - Q \circ f^{-1}\|_{\mathrm{TV}} \;\le\; \|P - Q\|_{\mathrm{TV}}.$$

Proof. By definition of total variation on a finite space,

$$\|P\circ f^{-1} - Q\circ f^{-1}\|_{\mathrm{TV}} = \sup_{B\subseteq\mathcal{S}} \big|(P\circ f^{-1})(B) - (Q\circ f^{-1})(B)\big| = \sup_{B\subseteq\mathcal{S}} \big|P(f^{-1}(B)) - Q(f^{-1}(B))\big|.$$

Since $f^{-1}(B) \subseteq \Omega$ for every $B \subseteq \mathcal{S}$, we have

$$\sup_{B\subseteq\mathcal{S}} \big|P(f^{-1}(B)) - Q(f^{-1}(B))\big| \;\le\; \sup_{A\subseteq\Omega} \big|P(A) - Q(A)\big| = \|P - Q\|_{\mathrm{TV}},$$

which proves the claim. □

The next lemma bounds the total variation between joint distributions sharing a common marginal.

Lemma 51. *Let $\mathcal{X}, \mathcal{Y}$ be finite sets. Let Π, Π' be probability measures on $\mathcal{X} \times \mathcal{Y}$ with the same first marginal μ on $\mathcal{X}$, i.e.*

$$\Pi(x, \mathcal{Y}) = \Pi'(x, \mathcal{Y}) = \mu(x) \qquad \forall\, x \in \mathcal{X}.$$

For each x with $\mu(x) > 0$, let

$$K_x(\cdot) := \Pi(\cdot \mid X = x), \qquad L_x(\cdot) := \Pi'(\cdot \mid X = x)$$

denote the conditional distributions on $\mathcal{Y}$. (For x with $\mu(x) = 0$, define $K_x = L_x$ arbitrarily.) Then

$$\|\Pi - \Pi'\|_{\mathrm{TV}} \;\le\; \sum_{x\in\mathcal{X}} \mu(x)\, \|K_x - L_x\|_{\mathrm{TV}}.$$

Proof. For any event $E \subseteq \mathcal{X} \times \mathcal{Y}$, write the x-section

$$E_x := \{y \in \mathcal{Y} :\ (x, y) \in E\} \subseteq \mathcal{Y}.$$

We have that

$$\Pi(E) = \sum_{x\in\mathcal{X}} \Pi(\{x\} \times E_x) = \sum_{x\in\mathcal{X}} \Pi(X = x)\, \Pi(E_x \mid X = x) = \sum_{x\in\mathcal{X}} \mu(x)\, K_x(E_x),$$

and similarly

$$\Pi'(E) = \sum_{x\in\mathcal{X}} \mu(x)\, L_x(E_x),$$

since the first marginal is the same μ for both measures. Therefore

$$\Pi(E) - \Pi'(E) = \sum_{x \in \mathcal{X}} \mu(x) \big(K_x(E_x) - L_x(E_x)\big).$$

Taking absolute values and using $|\sum a_x| \le \sum |a_x|$ gives

$$|\Pi(E) - \Pi'(E)| \le \sum_{x \in \mathcal{X}} \mu(x) \big|K_x(E_x) - L_x(E_x)\big|.$$

For each fixed x, the set $E_x \subseteq \mathcal{Y}$ is an admissible event in the definition of total variation on $\mathcal{Y}$, hence

$$\big|K_x(E_x) - L_x(E_x)\big| \le \sup_{B \subseteq \mathcal{Y}} \big|K_x(B) - L_x(B)\big| = \|K_x - L_x\|_{\mathrm{TV}}.$$

Substituting yields

$$|\Pi(E) - \Pi'(E)| \le \sum_{x \in \mathcal{X}} \mu(x) \, \|K_x - L_x\|_{\mathrm{TV}}.$$

Finally, take a supremum over all events $E \subseteq \mathcal{X} \times \mathcal{Y}$ to obtain

$$\|\Pi - \Pi'\|_{\mathrm{TV}} = \sup_{E \subseteq \mathcal{X} \times \mathcal{Y}} |\Pi(E) - \Pi'(E)| \le \sum_{x \in \mathcal{X}} \mu(x) \, \|K_x - L_x\|_{\mathrm{TV}},$$

as claimed. □

We now extend the above lemma to the case of trajectories sampled from Markov chains.

Lemma 52. *Let $\mathcal{X}$ be a finite state space and let P, Q be Markov kernels on $\mathcal{X}$. Fix an initial distribution μ on $\mathcal{X}$. For $T \ge 1$, let $\mathbb{P}^{(T)}_{\mu,P}$ and $\mathbb{P}^{(T)}_{\mu,Q}$ denote the laws on $\mathcal{X}^{T+1}$ of the length-$(T+1)$ trajectories $(X_0, \dots, X_T)$ with $X_0 \sim \mu$ and one-step transitions given by P and Q, respectively. Then*

$$\big\|\mathbb{P}^{(T)}_{\mu,P} - \mathbb{P}^{(T)}_{\mu,Q}\big\|_{\mathrm{TV}} \;\le\; \sum_{t=0}^{T-1} \sup_{x \in \mathcal{X}} \big\|P(x,\cdot) - Q(x,\cdot)\big\|_{\mathrm{TV}} \;=\; T \cdot \sup_{x \in \mathcal{X}} \big\|P(x,\cdot) - Q(x,\cdot)\big\|_{\mathrm{TV}}.$$

Proof. For $t = 0, 1, \dots, T$, let $R^{(t)}$ be the law on $\mathcal{X}^{T+1}$ of the trajectory $(X_0, \dots, X_T)$ obtained by starting from $X_0 \sim \mu$, using P for the first t transitions, and using Q for the remaining transitions. Thus $R^{(0)} = \mathbb{P}^{(T)}_{\mu,Q}$ and $R^{(T)} = \mathbb{P}^{(T)}_{\mu,P}$. By the triangle inequality,

$$\big\|\mathbb{P}^{(T)}_{\mu,P} - \mathbb{P}^{(T)}_{\mu,Q}\big\|_{\mathrm{TV}} \le \sum_{t=0}^{T-1} \|R^{(t+1)} - R^{(t)}\|_{\mathrm{TV}}.$$

Fix $t \in \{0, \dots, T-1\}$. Define $Y := (X_{t+1}, \dots, X_T) \in \mathcal{X}^{T-t}$. Under both $R^{(t)}$ and $R^{(t+1)}$, the first t transitions are governed by P, hence the marginal law of X_t is the same. Therefore conditioning on X_t and averaging yields

$$\|R^{(t+1)} - R^{(t)}\|_{\mathrm{TV}} \le \sum_{x \in \mathcal{X}} \Pr_{R^{(t)}}(X_t = x) \Big\|R^{(t+1)}(\cdot \mid X_t = x) - R^{(t)}(\cdot \mid X_t = x)\Big\|_{\mathrm{TV}}.$$

Define a Markov kernel M from $\mathcal{X}$ to $\mathcal{X}^{T-t}$ as follows: for each $y \in \mathcal{X}$, $M(y,\cdot)$ is the trajectory distribution on $\mathcal{X}^{T-t}$ of length $(T-t)$ for the Q-chain $(Z_0, \dots, Z_{T-t-1})$ started from $Z_0 = y$. We use the standard composition notation: if ν is a distribution on $\mathcal{X}$, then νM denotes the induced distribution on $\mathcal{X}^{T-t}$ obtained by first sampling $Y_0 \sim \nu$ and then sampling a trajectory $Y \sim M(Y_0, \cdot)$; equivalently, for any $A \subseteq \mathcal{X}^{T-t}$,

$$(\nu M)(A) = \sum_{y \in \mathcal{X}} \nu(y) \, M(y, A).$$

Then, conditional on $X_t = x$, under $R^{(t)}$ we have $X_{t+1} \sim Q(x,\cdot)$ and Y has distribution $Q(x,\cdot)M$, whereas under $R^{(t+1)}$ we have $X_{t+1} \sim P(x,\cdot)$ and Y has distribution $P(x,\cdot)M$. Hence

$$\Big\|R^{(t+1)}(\cdot \mid X_t = x) - R^{(t)}(\cdot \mid X_t = x)\Big\|_{\mathrm{TV}} = \|P(x,\cdot)M - Q(x,\cdot)M\|_{\mathrm{TV}}.$$

Finally, total variation is contractive under a common Markov kernel (by Lemma 50): for any distributions ν, η on $\mathcal{X}$,

$$\|\nu M - \eta M\|_{\mathrm{TV}} \le \|\nu - \eta\|_{\mathrm{TV}}.$$

Applying this with $\nu = P(x,\cdot)$ and $\eta = Q(x,\cdot)$ yields

$$\|P(x,\cdot)M - Q(x,\cdot)M\|_{\mathrm{TV}} \le \|P(x,\cdot) - Q(x,\cdot)\|_{\mathrm{TV}}.$$

Substituting into the bound on $\|R^{(t+1)} - R^{(t)}\|_{\mathrm{TV}}$ gives

$$\|R^{(t+1)} - R^{(t)}\|_{\mathrm{TV}} \le \sum_{x\in\mathcal{X}} \Pr_{R^{(t)}}(X_t = x)\,\|P(x,\cdot) - Q(x,\cdot)\|_{\mathrm{TV}} \le \sup_{x\in\mathcal{X}} \|P(x,\cdot) - Q(x,\cdot)\|_{\mathrm{TV}}.$$

Summing over $t = 0, \dots, T-1$ completes the proof:

$$\left\|\mathbb{P}^{(T)}_{\mu,P} - \mathbb{P}^{(T)}_{\mu,Q}\right\|_{\mathrm{TV}} \le T \cdot \sup_{x\in\mathcal{X}} \|P(x,\cdot) - Q(x,\cdot)\|_{\mathrm{TV}}.$$

□

We now prove a slight extension of the above lemma to the case of trajectories sampled from Markov chains with absorbing states, and hence the trajectory length is not fixed.

Lemma 53. *Let $\mathcal{X}$ be finite. Let P, Q be Markov kernels on $\mathcal{X}$ and let $T \subseteq \mathcal{X}$ be absorbing for both P and Q. Let $\tau := \inf\{t \ge 0 : X_t \in T\}$. For an integer $M \ge 0$, define the truncated stopping time $\tau \wedge M$ and consider the corresponding stopped paths $X_{0:\tau}$ and $X_{0:\tau\wedge M}$. Let $\mathcal{L}_P(X_{0:\tau})$ and $\mathcal{L}_Q(X_{0:\tau})$ denote the laws of $X_{0:\tau}$ under P and Q (respectively), for the same initial law μ. Then*

$$\left\|\mathcal{L}_P(X_{0:\tau}) - \mathcal{L}_Q(X_{0:\tau})\right\|_{\mathrm{TV}} \;\le\; M \cdot \sup_{x\in\mathcal{X}} \|P(x,\cdot) - Q(x,\cdot)\|_{\mathrm{TV}} \;+\; \Pr_P(\tau > M) \;+\; \Pr_Q(\tau > M).$$

Proof. Let $\mathbb{P}^{(M)}_{\mu,P}$ and $\mathbb{P}^{(M)}_{\mu,Q}$ be the laws of $(X_0, \dots, X_M)$ under kernels P and Q, respectively. The truncated stopped path $X_{0:\tau\wedge M}$ is a deterministic function of $(X_0, \dots, X_M)$, hence by the data processing inequality for total variation (Lemma 50),

$$\left\|\mathcal{L}_P(X_{0:\tau\wedge M}) - \mathcal{L}_Q(X_{0:\tau\wedge M})\right\|_{\mathrm{TV}} \;\le\; \left\|\mathbb{P}^{(M)}_{\mu,P} - \mathbb{P}^{(M)}_{\mu,Q}\right\|_{\mathrm{TV}}.$$

By Lemma 52,

$$\left\|\mathbb{P}^{(M)}_{\mu,P} - \mathbb{P}^{(M)}_{\mu,Q}\right\|_{\mathrm{TV}} \;\le\; M \cdot \sup_{x\in\mathcal{X}} \|P(x,\cdot) - Q(x,\cdot)\|_{\mathrm{TV}},$$

and therefore

$$\left\|\mathcal{L}_P(X_{0:\tau\wedge M}) - \mathcal{L}_Q(X_{0:\tau\wedge M})\right\|_{\mathrm{TV}} \;\le\; M \cdot \sup_{x\in\mathcal{X}} \|P(x,\cdot) - Q(x,\cdot)\|_{\mathrm{TV}}.$$

It remains to compare the true stopped path $X_{0:\tau}$ with the truncated one $X_{0:\tau\wedge M}$. On the event $\{\tau \le M\}$ we have the identity $X_{0:\tau} = X_{0:\tau\wedge M}$. Hence for any event A in the stopped-path sigma-field,

$$\Pr_P(X_{0:\tau} \in A) \le \Pr_P(X_{0:\tau\wedge M} \in A) \;+\; \Pr_P(\tau > M), \tag{43}$$

and similarly

$$\Pr_Q(X_{0:\tau} \in A) \le \Pr_Q(X_{0:\tau\wedge M} \in A) \;+\; \Pr_Q(\tau > M). \tag{44}$$

Applying (43) to the complement A^c and using $\Pr(\cdot) = 1 - \Pr(\cdot^c)$ yields the corresponding reverse inequality:

$$\Pr_P(X_{0:\tau} \in A) \ge \Pr_P(X_{0:\tau\wedge M} \in A) \;-\; \Pr_P(\tau > M),$$

and likewise

$$\Pr_Q(X_{0:\tau} \in A) \ge \Pr_Q(X_{0:\tau\wedge M} \in A) \;-\; \Pr_Q(\tau > M).$$

Combining upper and lower bounds, we obtain for every A,

$$\left|\Pr_P(X_{0:\tau} \in A) - \Pr_Q(X_{0:\tau} \in A)\right| \le \left|\Pr_P(X_{0:\tau\wedge M} \in A) - \Pr_Q(X_{0:\tau\wedge M} \in A)\right| + \Pr_P(\tau > M) + \Pr_Q(\tau > M).$$

Taking a supremum over A gives

$$\left\|\mathcal{L}_P(X_{0:\tau}) - \mathcal{L}_Q(X_{0:\tau})\right\|_{\mathrm{TV}} \;\le\; \left\|\mathcal{L}_P(X_{0:\tau\wedge M}) - \mathcal{L}_Q(X_{0:\tau\wedge M})\right\|_{\mathrm{TV}} + \Pr_P(\tau > M) + \Pr_Q(\tau > M).$$

Substituting the earlier bound on the truncated stopped-path TV completes the proof. □

The following lemma is a standard result about the total variation distance between distributions induced by softmax over logits.

Lemma 54. *Let $\mathcal{Y}$ be finite. For logits $o, o' \in \mathbb{R}^{\mathcal{Y}}$, define*

$$\sigma(o)(y) := \frac{e^{o(y)}}{\sum_{z\in\mathcal{Y}} e^{o(z)}}.$$

Then

$$\|\sigma(o) - \sigma(o')\|_{\mathrm{TV}} \;\le\; \|o - o'\|_\infty.$$

Proof. Let $o_t := (1-t)o + to'$ and $p_t := \sigma(o_t)$. Then

$$\sigma(o') - \sigma(o) = \int_0^1 \frac{d}{dt} p_t \, dt.$$

By the softmax derivative identity (Lemma 62) with parameter t, each coordinate satisfies

$$\Big|\frac{d}{dt}p_t(y)\Big| = p_t(y)\Big|(o'(y) - o(y)) - \sum_z p_t(z)(o'(z) - o(z))\Big| \le 2\, p_t(y)\, \|o' - o\|_\infty.$$

Summing over y gives

$$\Big\|\frac{d}{dt}p_t\Big\|_1 \le 2\|o' - o\|_\infty,$$

hence

$$\|\sigma(o') - \sigma(o)\|_{\mathrm{TV}} = \frac{1}{2}\|\sigma(o') - \sigma(o)\|_1 \le \frac{1}{2}\int_0^1 \Big\|\frac{d}{dt}p_t\Big\|_1 dt \le \|o' - o\|_\infty.$$

□

We now apply the above lemmas to the case of Markov kernels induced by the logit structure (Appendix F.1).

Lemma 55. *Let $P_{\alpha,\beta,\gamma}$ and $P_{\alpha',\beta',\gamma'}$ be Markov kernels induced by the logit structure with parameters (α,β,γ) and (α',β',γ') (see Appendix F.1). Then*

$$\sup_x \big\|P_{\alpha,\beta,\gamma}(x,\cdot) - P_{\alpha',\beta',\gamma'}(x,\cdot)\big\|_{\mathrm{TV}} \;\le\; |\alpha-\alpha'| + |\beta-\beta'| + |\gamma-\gamma'|.$$

Proof. Fix x. By the logit structure, $P_{\alpha,\beta,\gamma}(x,\cdot)$ is a softmax over logits $o_{\alpha,\beta,\gamma}(x,\cdot)$. By inspection of the logit structure (including boundary cases), each coordinate of $o(x,\cdot)$ changes by at most $|\alpha-\alpha'| + |\beta-\beta'| + |\gamma-\gamma'|$ when the parameters are changed, hence

$$\|o_{\alpha,\beta,\gamma}(x,\cdot) - o_{\alpha',\beta',\gamma'}(x,\cdot)\|_\infty \le |\alpha-\alpha'| + |\beta-\beta'| + |\gamma-\gamma'|.$$

Lemma 54 implies the stated bound after taking a supremum over x.

□

Combining the kernel TV bound with the trajectory TV lemmas yields the following corollary.

Corollary 3. *Let P_θ and $P_{\theta'}$ be Markov kernels (induced by the logit structure) at $\theta = (\alpha,\beta,\gamma)$ and $\theta' = (\alpha',\beta',\gamma')$, with the same initial distribution μ. Then for every $T \ge 1$,*

$$\big\|\mathbb{P}^{(T)}_{\mu,P_\theta} - \mathbb{P}^{(T)}_{\mu,P_{\theta'}}\big\|_{\mathrm{TV}} \;\le\; T\,\|\theta - \theta'\|_1.$$

If $T \subseteq \mathcal{X}$ is absorbing and τ is the corresponding hitting time, then for every integer $M \ge 0$,

$$\big\|p_{roll}(\theta) - p_{roll}(\theta')\big\|_{\mathrm{TV}} \;\le\; M\,\|\theta - \theta'\|_1 + \Pr_\theta(\tau > M) + \Pr_{\theta'}(\tau > M).$$

Proof. Combine Lemma 52 with Lemma 55 for the fixed-horizon bound, and combine Lemma 53 with Lemma 55 for the stopped-path bound. □

G.2 Absorption time bounds

The stopped-path TV bound in Corollary 3 involves tail probabilities $\Pr(\tau > M)$, which we control via a geometric bound on the absorption time.

Lemma 56. *Let P be a Markov kernel on $\mathcal{X}$ with terminal states $\{a_n, b_n\}$, and let $\tau := \inf\{t \geq 0 : X_t \in \{a_n, b_n\}\}$. If for every non-terminal state x, the two-step absorption probability satisfies $P^2(x, \{a_n, b_n\}) \geq p$ for some $p \in (0, 1]$, then for every integer $m \geq 0$,*

$$\Pr(\tau > 2m) \leq (1-p)^m, \qquad \textit{and hence} \qquad \mathbb{E}[\tau] \leq \frac{2}{p}.$$

Proof. For each $t \geq 0$, on the event $\{\tau > 2t\}$ we have $X_{2t} \notin \{a_n, b_n\}$ and therefore

$$\Pr(\tau > 2(t+1) \mid \tau > 2t) \leq \Pr(X_{2(t+1)} \notin \{a_n, b_n\} \mid X_{2t} \notin \{a_n, b_n\}) \leq 1 - p.$$

Thus $\Pr(\tau > 2(t+1)) \leq (1-p)\Pr(\tau > 2t)$, and iterating gives $\Pr(\tau > 2m) \leq (1-p)^m$. The expectation bound follows from $\mathbb{E}[\tau] \leq 2\sum_{m \geq 0} \Pr(\tau > 2m)$. □

To apply Lemma 56, we must verify that the two-step absorption probability is lower bounded at initialization. The next lemma establishes this via a two-step argument: from any non-terminal state, one can first perform a long jump to a penultimate state, then move forward to the terminal.

Lemma 57. *Under the base model assumptions of Section 5.2—specifically that $p_{\mathrm{fwd}} > p_{\mathrm{bwd}} + \zeta$ and $p_{\mathrm{switch}} > \upsilon$—the minimum two-step absorption probability satisfies*

$$p^{(2)}_{\mathrm{abs}}(\theta_0) \;:=\; \min_{x \notin \{a_n, b_n\}} P^2_{\theta_0}(x, \{a_n, b_n\}) \;\geq\; \frac{c}{n}$$

for some constant $c = c(\upsilon, \zeta) > 0$ and all n sufficiently large.

Proof. Fix a non-terminal vertex x. We show that with probability $\Omega(1/n)$, the chain reaches a terminal state in exactly two steps via the following path: first, a long jump to a penultimate state a_{n-1} or b_{n-1}; then, a forward step to the corresponding terminal a_n or b_n.

Step 1: Long jump to penultimate state. By the logit structure (Appendix F.1), all distant targets share the same logit γ. The penultimate states $\{a_{n-1}, b_{n-1}\}$ are among the $\Theta(n)$ distant targets from any non-terminal x. Since the switch probability $p_{\mathrm{switch}}(x) > \upsilon$ and all distant targets have equal logit, the probability of jumping to $\{a_{n-1}, b_{n-1}\}$ is at least $\Omega(\upsilon/n)$.

Step 2: Forward step to terminal. From a penultimate state a_{n-1} (or b_{n-1}), the forward step leads to the terminal a_n (or b_n). By the base model assumption, the forward probability satisfies $p_{\mathrm{fwd}} > p_{\mathrm{bwd}} + \zeta$, which implies $p_{\mathrm{fwd}} \geq \zeta/2$ (since probabilities are non-negative).

Combining both steps, the two-step absorption probability is at least

$$P^2_{\theta_0}(x, \{a_n, b_n\}) \;\geq\; \Omega\Big(\frac{\upsilon}{n}\Big) \cdot \Omega(\zeta) \;=\; \Omega\Big(\frac{1}{n}\Big).$$

Taking the minimum over all non-terminal x yields the result. □

Having established that $p^{(2)}_{\mathrm{abs}}(\theta_0) > 0$ at initialization, we need to ensure it remains positive throughout gradient flow. The next lemma shows that $p^{(2)}_{\mathrm{abs}}$ is Lipschitz in the parameters, so it cannot drop too quickly as θ evolves.

Lemma 58. *Let P_θ be the Markov kernel (induced by the logit structure) at parameters $\theta = (\alpha, \beta, \gamma)$, and define*

$$p^{(2)}_{\mathrm{abs}}(\theta) \;:=\; \min_{x \notin \{a_n, b_n\}} P^2_\theta(x, \{a_n, b_n\}).$$

Then for all θ, θ',

$$p^{(2)}_{\mathrm{abs}}(\theta) \;\geq\; p^{(2)}_{\mathrm{abs}}(\theta') \;-\; C\|\theta - \theta'\|_1$$

for some constant $C > 0$.

Proof. Fix $x \notin \{a_n, b_n\}$. The two-step transition probability can be written as

$$P_\theta^2(x, \{a_n, b_n\}) = \sum_y P_\theta(x, y) P_\theta(y, \{a_n, b_n\}).$$

By the triangle inequality and Lemma 55, changing θ to θ' affects each one-step kernel by at most $\|\theta - \theta'\|_1$ in total variation. Since the two-step probability is a composition of two kernels, the change in $P^2(x, \{a_n, b_n\})$ is bounded by $C\|\theta - \theta'\|_1$ for some constant C. Taking the minimum over all non-terminal x yields the result. □

For the gradient bounds below, we also need stability of the *long-jump probability*—the probability of jumping to a distant state in one step.

Lemma 59. *Let P_θ be the Markov kernel (induced by the logit structure) at parameters $\theta = (\alpha, \beta, \gamma)$. For each non-terminal state x, let $\mathrm{LJ}(x) := \{y : |D(y) - D(x)| \geq 2\}$ denote the set of long-jump targets, and define*

$$p_{\mathrm{lj}}(\theta) \;:=\; \min_{x \notin \{a_n, b_n\}} P_\theta(x, \mathrm{LJ}(x)).$$

Then for all θ, θ',

$$p_{\mathrm{lj}}(\theta) \;\geq\; p_{\mathrm{lj}}(\theta') \;-\; \|\theta - \theta'\|_1.$$

Proof. Fix $x \notin \{a_n, b_n\}$. By definition of total variation,

$$\big|P_\theta(x, \mathrm{LJ}(x)) - P_{\theta'}(x, \mathrm{LJ}(x))\big| \leq \big\|P_\theta(x, \cdot) - P_{\theta'}(x, \cdot)\big\|_{\mathrm{TV}}.$$

By Lemma 55, the right-hand side is at most $\|\theta - \theta'\|_1$. Taking the minimum over all non-terminal x yields the result. □

G.3 Exponentially small gradients for hard examples

The final ingredient shows that when all starting states are far from the terminal, the loss gradient is exponentially small.

Lemma 60. *Fix constants $a \in (0, 1)$ and $q_0 \in (0, 1)$. Let $\mathcal{H} \subseteq \mathcal{X}$ be a set of start states with $D(v) \geq an$ for all $v \in \mathcal{H}$, let $(w_v)_{v \in \mathcal{H}}$ be nonnegative weights summing to 1, and define*

$$\mathcal{L}(\theta) \;:=\; \sum_{v \in \mathcal{H}} w_v \, S(v; \theta), \qquad \mathcal{U}_{q_0} \;:=\; \Big\{\theta :\; p_{\mathrm{lj}}(\theta) \geq q_0\Big\}.$$

Then there exist constants $C, c > 0$ (independent of n) such that for every $\theta \in \mathcal{U}_{q_0}$,

$$\|\nabla \mathcal{L}(\theta)\|_1 \;\leq\; C \, \frac{n}{\zeta} \, e^{-cn}.$$

Proof. Fix $\theta \in \mathcal{U}_{q_0}$, so $p_{\mathrm{lj}}(\theta) \geq q_0$. By Lemma 46 (with the long-jump probability lower bounded by q_0) and the trivial bounds $p_f^{\max}, p_b^{\max}, p_{\mathrm{lj}}^{\max} \leq 1$, for each $\vartheta \in \{\alpha, \beta, \gamma\}$,

$$\sup_{v \in \mathcal{H}} \big|\partial_\vartheta S(v; \theta)\big| \;\leq\; C \, \frac{n}{\zeta} \, e^{-cn}.$$

Since $\mathcal{L}$ is a convex combination, the triangle inequality gives $|\partial_\vartheta \mathcal{L}| \leq \sup_v |\partial_\vartheta S(v; \theta)|$. Summing over ϑ yields the stated bound. □

G.4 Proof of Theorem 3

We now assemble the above lemmas to prove the main result.

Proof of Theorem 3. By Lemma 57, the base model assumptions ensure that $q_0 := p_{\mathrm{abs}}^{(2)}(\theta_0) \geq c/n > 0$ for some constant $c = c(\upsilon, \zeta) > 0$ and n sufficiently large. By an analogous argument (with the same υ bound on switch probability), the long-jump probability satisfies $p_{\mathrm{lj}}(\theta_0) \geq c'/n$ for some constant $c' > 0$; without loss of generality, assume $c' \geq c$ so that $p_{\mathrm{lj}}(\theta_0) \geq q_0$.

Define the safe region

$$\mathcal{U} \;:=\; \{\theta :\; p^{(2)}_{\mathrm{abs}}(\theta) \ge q_0/2 \text{ and } p_{\mathrm{lj}}(\theta) \ge q_0/2\},$$

and the exit time

$$t^\star \;:=\; \inf\{t \ge 0 :\; \theta(t) \notin \mathcal{U}\},$$

with the convention $t^\star = \infty$ if the set is empty. By definition, for every $s \in [0, t^\star)$ we have $\theta(s) \in \mathcal{U}$, and hence Lemma 60 (applied with $q_0/2$ in place of q_0) yields

$$\|\dot\theta(s)\|_1 = \|\nabla \mathcal{L}(\theta(s))\|_1 \;\le\; C\,\frac{n}{\zeta}\,e^{-cn} \qquad (0 \le s < t^\star).$$

Consequently, for every $s \ge 0$,

$$\|\theta(s \wedge t^\star) - \theta(0)\|_1 \le \int_0^{s\wedge t^\star} \|\dot\theta(u)\|_1\,du \le (s \wedge t^\star)\cdot\Big(C\,\frac{n}{\zeta}\,e^{-cn}\Big).$$

Applying Lemmas 58 and 59 with $\theta = \theta(s \wedge t^\star)$ and $\theta' = \theta(0)$ gives

$$p^{(2)}_{\mathrm{abs}}(\theta(s \wedge t^\star)) \ge q_0 - (s \wedge t^\star)\cdot\Big(C' \cdot C\,\frac{n}{\zeta}\,e^{-cn}\Big),$$
$$p_{\mathrm{lj}}(\theta(s \wedge t^\star)) \ge q_0 - (s \wedge t^\star)\cdot\Big(C\,\frac{n}{\zeta}\,e^{-cn}\Big).$$

In particular, if $s \le \frac{q_0}{2}\cdot\frac{\zeta}{C''n}\,e^{cn}$ for $C'' := \max(C'C, C)$, then both bounds remain at least $q_0/2$, so $\theta(s \wedge t^\star) \in \mathcal{U}$. If $t^\star \le s$, then by definition of $t^\star$ we would have $\theta(t^\star) \notin \mathcal{U}$, a contradiction. Therefore $t^\star > s$.

Setting

$$t_{\mathrm{boot}}(n) \;:=\; \frac{q_0}{4}\cdot\frac{\zeta}{C''n}\,e^{cn} \;=\; \frac{c\,\zeta}{4C''n^2}\,e^{cn},$$

we conclude that for every $t \in [0, t_{\mathrm{boot}}(n)]$ one has $t < t^\star$, and hence

$$p^{(2)}_{\mathrm{abs}}(\theta(u)) \;\ge\; \frac{q_0}{2}, \quad p_{\mathrm{lj}}(\theta(u)) \;\ge\; \frac{q_0}{2} \qquad \forall\, u \in [0, t].$$

Now fix $t \in [0, t_{\mathrm{boot}}(n)]$ and let $\tau := \inf\{t \ge 0 :\; X_t \in \{a_n, b_n\}\}$. Because $p^{(2)}_{\mathrm{abs}}(\theta(0)) = q_0$ and $p^{(2)}_{\mathrm{abs}}(\theta(t)) \ge q_0/2$, Lemma 56 yields

$$\Pr_{\theta(0)}(\tau > 2M) \le (1 - q_0/2)^M, \qquad \Pr_{\theta(t)}(\tau > 2M) \le (1 - q_0/2)^M \qquad \forall\, M \ge 0.$$

Finally, apply the stopped-path bound in Corollary 3 with $\theta = \theta(0)$ and $\theta' = \theta(t)$:

$$\big\|p_{\text{roll}}(\theta(t)) - p_{\text{roll}}(\theta(0))\big\|_{\mathrm{TV}} \;\le\; M\,\|\theta(t) - \theta(0)\|_1 \;+\; \Pr_{\theta(t)}(\tau > M) + \Pr_{\theta(0)}(\tau > M).$$

Using the tail estimates (noting that $\Pr(\tau > M) \le \Pr(\tau > 2\lfloor M/2\rfloor) \le (1 - q_0/2)^{\lfloor M/2\rfloor}$) and the drift bound $\|\theta(t) - \theta(0)\|_1 \le t\cdot\big(C\frac{n}{\zeta}e^{-cn}\big)$ gives, for all $M \ge 0$,

$$\big\|p_{\text{roll}}(\theta(t)) - p_{\text{roll}}(\theta(0))\big\|_{\mathrm{TV}} \;\le\; M\,t\cdot\Big(C\,\frac{n}{\zeta}\,e^{-cn}\Big) \;+\; 2(1 - q_0/2)^{M/2}.$$

We choose M to (approximately) minimize the right-hand side. Treat $M \ge 0$ as a real variable and write

$$f(M) \;:=\; A\,M + 2e^{-\lambda M},$$

where $A := t\big(C\frac{n}{\zeta}e^{-cn}\big)$ and $\lambda := -\frac{1}{2}\log(1 - q_0/2) > 0$. Differentiating gives

$$f'(M) \;=\; A - 2\lambda e^{-\lambda M}, \qquad f''(M) \;=\; 2\lambda^2 e^{-\lambda M} \;>\; 0,$$

so f is convex and has a unique minimizer whenever it exists in $[0, \infty)$. If $A < 2\lambda$, the minimizer satisfies $f'(M^\star) = 0$, i.e.

$$e^{-\lambda M^\star} \;=\; \frac{A}{2\lambda} \qquad\Longleftrightarrow\qquad M^\star \;=\; \frac{1}{\lambda}\log\Big(\frac{2\lambda}{A}\Big).$$

Substituting $e^{-\lambda M^\star} = A/(2\lambda)$ yields the optimized value

$$f(M^\star) \;=\; A \cdot \frac{1}{\lambda} \log\Big(\frac{2\lambda}{A}\Big) \;+\; 2 \cdot \frac{A}{2\lambda} \;=\; \frac{A}{\lambda}\Big(1 + \log\Big(\frac{2\lambda}{A}\Big)\Big).$$

Returning to integer M, set $M_{\mathrm{opt}} := \lceil M^\star \rceil$. Since f is convex and f' is A-Lipschitz, one has $f(M_{\mathrm{opt}}) \le f(M^\star) + A$. In particular, for all $A < 2\lambda$,

$$\big\|p_{\mathrm{roll}}(\theta(t)) - p_{\mathrm{roll}}(\theta(0))\big\|_{\mathrm{TV}} \;\le\; f(M_{\mathrm{opt}}) \;\le\; \frac{A}{\lambda}\Big(1 + \log\Big(\frac{2\lambda}{A}\Big)\Big) \;+\; A.$$

Equivalently, writing $x := A/(2\lambda) \in (0, 1)$, this becomes

$$f(M_{\mathrm{opt}}) \;\le\; 2x\big(1 + \log(1/x)\big) \;+\; 2\lambda x.$$

Consequently, to guarantee the total-variation bound is at most ε, it suffices to enforce

$$\frac{A}{\lambda}\Big(1 + \log\Big(\frac{2\lambda}{A}\Big)\Big) \;+\; A \;\le\; \varepsilon,$$

where $A = t\big(C\frac{n}{\zeta}e^{-cn}\big)$ and $\lambda = -\frac{1}{2}\log(1 - q_0/2)$.

Since $q_0 = c/n$, we have $\lambda = -\frac{1}{2}\log(1 - c/(2n)) \approx c/(4n)$ for large n, i.e., $\lambda = \Theta(1/n)$.

To match the bound in Theorem 3, we verify that $t < \varepsilon^2 \cdot e^{\kappa n}$ implies the above bound. For $t = \varepsilon^2 \cdot e^{\kappa n}$, we have

$$A \;=\; \varepsilon^2 \cdot e^{\kappa n} \cdot \frac{Cn}{\zeta}\, e^{-cn} \;=\; \varepsilon^2 \cdot \frac{Cn}{\zeta}\, e^{-(c-\kappa)n}.$$

The logarithmic factor satisfies

$$\log\Big(\frac{2\lambda}{A}\Big) \;=\; \log(2\lambda) - \log(\varepsilon^2) - \log\Big(\frac{Cn}{\zeta}\Big) + (c-\kappa)n \;=\; (c-\kappa)n + O\big(\log n + \log(1/\varepsilon)\big).$$

Substituting into the bound gives

$$\frac{A}{\lambda}\Big(1 + \log\Big(\frac{2\lambda}{A}\Big)\Big) + A \;\le\; \frac{A}{\lambda} \cdot (c-\kappa)n \;+\; O\Big(\frac{A}{\lambda}\big(\log n + \log(1/\varepsilon)\big)\Big) \;+\; A.$$

Since $\lambda = \Theta(1/n)$, we have $A/\lambda = \Theta(n) \cdot A$. The dominant term is thus

$$\frac{A}{\lambda}(c-\kappa)n = O(1) \cdot \varepsilon^2 \cdot \frac{C(c-\kappa)n^3}{\zeta}\, e^{-(c-\kappa)n}.$$

Since $n^3 e^{-(c-\kappa)n} \to 0$ as $n \to \infty$ for any $\kappa < c$, all terms vanish exponentially. Specifically, for any $\kappa \in (0, c)$, there exists $n_0 = n_0(a, \upsilon, \varepsilon)$ such that for all $n \ge n_0$, the left-hand side is at most ε. Setting $\kappa := c/2$ yields the claimed result.

Finally, we verify that the time scale $t < \varepsilon^2 e^{\kappa n}$ lies within the bootstrap regime $t \le t_{\mathrm{boot}}(n)$. Since $t_{\mathrm{boot}}(n) = \frac{q_0 \zeta}{4C'Cn}\, e^{cn}$ and $q_0 = c/n$, we have $t_{\mathrm{boot}}(n) = \frac{c\,\zeta}{4C'Cn^2}\, e^{cn}$. With $\kappa = c/2$,

$$\frac{\varepsilon^2 e^{\kappa n}}{t_{\mathrm{boot}}(n)} \;=\; \frac{4C'C\varepsilon^2 n^2}{c\,\zeta}\, e^{-(c-\kappa)n} \;=\; \frac{4C'C\varepsilon^2 n^2}{c\,\zeta}\, e^{-(c/2)n} \;\to\; 0$$

as $n \to \infty$. Thus, for $n \ge n_0$ (possibly enlarging n_0), we have $\varepsilon^2 e^{\kappa n} < t_{\mathrm{boot}}(n)$, so the analysis above is valid for all $t < \varepsilon^2 e^{\kappa n}$. □

H Supporting Lemmas

Lemma 61. *Under the assumptions and notation of Sections 3 and 5. In particular, the distribution D_m^Q is invariant under vertex permutations, and the value matrix $A(t)$ evolves by gradient flow*

$$\frac{d}{dt}A(t) = -\nabla_A L(D_m^Q, A(t)),$$

For $\alpha, \beta, \gamma \in \mathbb{R}$, define $A(\alpha, \beta, \gamma)$ on the vertex–edge block by

$$A(\alpha,\beta,\gamma)_{v,(u,w)} = \begin{cases} \alpha & \text{if } v = u, \\ \beta & \text{if } v = w, \\ \gamma & \text{if } v \notin \{u, w\}, \end{cases} \qquad v, u, w \in V,\ u \neq w,$$

with all other entries defined as in Section 5.

Let $A(t)$ solve gradient flow with $A(0) = A(\alpha_0, \beta_0, \gamma_0)$ for some $\alpha_0, \beta_0, \gamma_0$. Then for all $t \geq 0$ there exist scalars $\alpha(t), \beta(t), \gamma(t)$ such that

$$A_{v,(u,w)}(t) = \begin{cases} \alpha(t) & \text{if } v = u, \\ \beta(t) & \text{if } v = w, \\ \gamma(t) & \text{if } v \notin \{u, w\}, \end{cases} \qquad \forall\, v \in V,\ (u,w) \in E,\ u \neq w.$$

Define the reduced loss

$$\tilde{L}(\alpha, \beta, \gamma) := L\big(D_m^Q, A(\alpha, \beta, \gamma)\big).$$

The scalars $\alpha(t), \beta(t), \gamma(t)$ satisfy

$$\begin{aligned} \dot{\alpha}(t) &= -\frac{1}{2n(2n-1)}\, \partial_\alpha \tilde{L}\big(\alpha(t), \beta(t), \gamma(t)\big), \\ \dot{\beta}(t) &= -\frac{1}{2n(2n-1)}\, \partial_\beta \tilde{L}\big(\alpha(t), \beta(t), \gamma(t)\big), \\ \dot{\gamma}(t) &= -\frac{1}{(2n-2)\cdot 2n(2n-1)}\, \partial_\gamma \tilde{L}\big(\alpha(t), \beta(t), \gamma(t)\big). \end{aligned}$$

Proof. For a permutation π of the vertex set V, define an operator T_π on matrices A by

$$\big(T_\pi A\big)_{v,(u,w)} := A_{\pi^{-1}(v),\,(\pi^{-1}(u),\pi^{-1}(w))}, \quad v, u, w \in V.$$

Equivalently, $(T_\pi A)_{\pi(v),(\pi(u),\pi(w))} = A_{v,(u,w)}$. This corresponds to consistently relabelling all vertices by π.

By construction of D_m^Q and the RL loss, relabelling vertices does not change the distribution over inputs or the loss, hence

$$L(D_m^Q, T_\pi A) = L(D_m^Q, A) \quad \text{for all } A, \pi.$$

Differentiating gives gradient equivariance

$$\nabla_A L(D_m^Q, T_\pi A) = T_\pi\, \nabla_A L(D_m^Q, A).$$

Now let $A(t)$ solve gradient flow. Then

$$\frac{d}{dt}\big(T_\pi A(t)\big) = T_\pi \frac{d}{dt} A(t) = -T_\pi \nabla_A L(D_m^Q, A(t)) = -\nabla_A L(D_m^Q, T_\pi A(t)).$$

So $T_\pi A(t)$ is also a solution of the same ODE.

At $t = 0$, the symmetric initialization from Section 5.2 gives

$$A_{v,(u,w)}(0) = \begin{cases} \alpha_0 & v = u, \\ \beta_0 & v = w, \\ \gamma_0 & v \notin \{u, w\}, \end{cases}$$

which is clearly invariant under any permutation of vertex labels, hence $T_\pi A(0) = A(0)$ for all π. By uniqueness of solutions to the ODE, we must have

$$T_\pi A(t) = A(t) \quad \text{for all } t \geq 0,\ \pi.$$

Thus $A(t)$ always lies in the fixed-point subspace

$$\mathcal{S} := \{A : T_\pi A = A \text{ for all } \pi\}.$$

To describe $\mathcal{S}$, consider triples $(v, u, w) \in V^3$ with $u \neq w$. Two such triples lie in the same orbit under the action of permutations if and only if they have the same equality pattern among $\{v, u, w\}$. There are exactly three possibilities:

1. $v = u$, $v \neq w$;
2. $v = w$, $v \neq u$;
3. v, u, w are all distinct.

Given any two triples with the same pattern, we can construct a permutation π mapping the first triple to the second (e.g., map $u \mapsto u'$, $w \mapsto w'$, and extend arbitrarily), so these are indeed the three orbits. If $A \in \mathcal{S}$, then for any two entries $A_{v,(u,w)}$ and $A_{v',(u',w')}$ from the same orbit we must have

$$A_{v,(u,w)} = A_{v',(u',w')},$$

because some π sends one triple to the other and $A = T_\pi A$. Therefore there exist scalars α, β, γ such that

$$A_{v,(u,w)} = \begin{cases} \alpha & v = u, \\ \beta & v = w, \\ \gamma & v \notin \{u, w\}, \end{cases}$$

which is exactly the form $A(\alpha, \beta, \gamma)$. This gives the representation of $A(t)$ in terms of $\alpha(t), \beta(t), \gamma(t)$.

For the dynamics, write $A(t) = A(\alpha(t), \beta(t), \gamma(t))$ and define $\tilde{L}$ as above. By the chain rule,

$$\partial_\alpha \tilde{L}(\alpha, \beta, \gamma) = \sum_{v=u,\, u\neq w} \frac{\partial}{\partial A_{v,(u,w)}} L(D_m^Q, A(\alpha, \beta, \gamma)),$$

and similarly for β, γ. Gradient flow gives

$$\frac{d}{dt} A_{v,(u,w)}(t) = -\frac{\partial}{\partial A_{v,(u,w)}} L(D_m^Q, A(t)).$$

By symmetry, all entries with $v = u$ share the same value $\alpha(t)$ and the same time derivative $\dot{\alpha}(t)$. Summing the previous identity over all such entries and comparing with the expression for $\partial_\alpha \tilde{L}$ yields

$$N_\alpha \, \dot{\alpha}(t) = -\partial_\alpha \tilde{L}\big(\alpha(t), \beta(t), \gamma(t)\big),$$

where N_α is the number of pairs $(v, (u, w))$ with $v = u$ and $u \neq w$. For each ordered pair (u, w) with $u \neq w$, there is exactly one v with $v = u$, so

$$N_\alpha = |E_{\text{dir}}| = |V|(|V| - 1),$$

where $E_{\text{dir}} = \{(u, w) \in V^2 : u \neq w\}$. The same reasoning gives $N_\beta = |V|(|V| - 1)$, and for γ, for each (u, w) there are $|V| - 2$ vertices $v \notin \{u, w\}$, so

$$N_\gamma = (|V| - 2)|V|(|V| - 1).$$

Dividing by these constants gives the claimed ODEs for α, β, γ. □

Lemma 62. *Let $\mathcal{Y}$ be a finite set. For each $\theta \in \mathbb{R}$, let $o_\theta : \mathcal{Y} \to \mathbb{R}$ be differentiable in θ, and define the softmax distribution P_θ on $\mathcal{Y}$ by*

$$P_\theta(y) := \frac{\exp(o_\theta(y))}{\sum_{z\in\mathcal{Y}} \exp(o_\theta(z))}.$$

Write $d_\theta(y) := \partial_\theta o_\theta(y)$. Then for every $y \in \mathcal{Y}$,

$$\partial_\theta P_\theta(y) := P_\theta(y)\Big(d_\theta(y) - \sum_{z\in\mathcal{Y}} P_\theta(z)\, d_\theta(z)\Big).$$

Equivalently, for any function $w : \mathcal{Y} \to \mathbb{R}$,

$$\sum_{y\in\mathcal{Y}} \big(\partial_\theta P_\theta(y)\big)\, w(y) := \sum_{y\in\mathcal{Y}} P_\theta(y)\, d_\theta(y)\, w(y) - \Big(\sum_{y\in\mathcal{Y}} P_\theta(y)\, d_\theta(y)\Big)\Big(\sum_{y\in\mathcal{Y}} P_\theta(y)\, w(y)\Big),$$

i.e.

$$\sum_{y\in\mathcal{Y}} \big(\partial_\theta P_\theta(y)\big)\, w(y) := \mathrm{Cov}_{Y\sim P_\theta}\big(d_\theta(Y), w(Y)\big).$$

Proof. Fix θ and abbreviate $o(y) := o_\theta(y)$, $P(y) := P_\theta(y)$, and $d(y) := \partial_\theta o_\theta(y)$. Let

$$Z := \sum_{z \in \mathcal{Y}} e^{o(z)}, \qquad \text{so} \qquad P(y) = \frac{e^{o(y)}}{Z}.$$

Differentiate using the quotient rule:

$$\partial_\theta P(y) := \frac{\partial_\theta e^{o(y)}}{Z} - \frac{e^{o(y)}}{Z^2}\,\partial_\theta Z.$$

Since $\partial_\theta e^{o(y)} = e^{o(y)} d(y)$ and

$$\partial_\theta Z = \sum_{z \in \mathcal{Y}} \partial_\theta e^{o(z)} := \sum_{z \in \mathcal{Y}} e^{o(z)} d(z),$$

we obtain

$$\partial_\theta P(y) := \frac{e^{o(y)}}{Z} d(y) - \frac{e^{o(y)}}{Z} \cdot \frac{1}{Z} \sum_{z \in \mathcal{Y}} e^{o(z)} d(z).$$

Recognizing $e^{o(y)}/Z = P(y)$ and $e^{o(z)}/Z = P(z)$ gives

$$\partial_\theta P(y) := P(y)\Big(d(y) - \sum_{z \in \mathcal{Y}} P(z) d(z)\Big),$$

which proves the first identity.

Multiplying this by $w(y)$ and summing over $y \in \mathcal{Y}$ yields

$$\sum_{y \in \mathcal{Y}} (\partial_\theta P(y)) w(y) := \sum_{y \in \mathcal{Y}} P(y) d(y) w(y) - \Big(\sum_{z \in \mathcal{Y}} P(z) d(z)\Big)\Big(\sum_{y \in \mathcal{Y}} P(y) w(y)\Big),$$

which is the stated centered form. The covariance reformulation follows from $\mathrm{Cov}(A, B) = \mathcal{E}[AB] - \mathcal{E}[A]\mathcal{E}[B]$ with $Y \sim P_\theta$. □

Lemma 63. *Let $(D_t)_{t \geq 0}$ be a Markov chain on $\{0, 1, \ldots, K\}$ with absorbing boundaries $\{0, K\}$. For each $i \in \{1, \ldots, K-1\}$ assume*

$$\Pr(D_{t+1} = i - 1 \mid D_t = i) = p_f(i), \qquad \Pr(D_{t+1} = i + 1 \mid D_t = i) = p_b(i),$$

and $\Pr(D_{t+1} = i \mid D_t = i) = 1 - p_f(i) - p_b(i)$ (holds allowed). Assume $p_f(i), p_b(i) > 0$ and that there exists $\rho \in (0, 1)$ such that for all $i \in \{1, \ldots, K-1\}$,

$$\frac{p_b(i)}{p_f(i)} \;\leq\; \rho.$$

Let $\tau_0 := \inf\{t \geq 0 : D_t = 0\}$ and $\tau_K := \inf\{t \geq 0 : D_t = K\}$. Then for every $k \in \{0, 1, \ldots, K\}$,

$$\Pr_k(\tau_K < \tau_0) \;\leq\; \rho^{\,K-k}.$$

Proof. Let $h(i) := \Pr_i(\tau_K < \tau_0)$. Then $h(0) = 0$, $h(K) = 1$, and for $1 \leq i \leq K-1$, conditioning on one step gives

$$h(i) = p_f(i) h(i-1) + p_b(i) h(i+1) + \big(1 - p_f(i) - p_b(i)\big) h(i),$$

hence

$$p_f(i)\big(h(i) - h(i-1)\big) = p_b(i)\big(h(i+1) - h(i)\big). \tag{45}$$

With $\Delta_i := h(i) - h(i-1)$, (45) implies $\Delta_{i+1} = \frac{p_f(i)}{p_b(i)} \Delta_i$, hence

$$\Delta_i = \Delta_1 \prod_{m=1}^{i-1} \frac{p_f(m)}{p_b(m)}.$$

Since $h(k) = \sum_{i=1}^{k} \Delta_i$ and $h(K) = \sum_{i=1}^{K} \Delta_i = 1$, we get

$$h(k) = \frac{\sum_{i=1}^{k} \prod_{m=1}^{i-1} \frac{p_f(m)}{p_b(m)}}{\sum_{i=1}^{K} \prod_{m=1}^{i-1} \frac{p_f(m)}{p_b(m)}}.$$

Using $p_f(m)/p_b(m) \geq \rho^{-1}$ in both numerator and denominator yields

$$h(k) \leq \frac{\sum_{i=1}^{k} \rho^{-(i-1)}}{\sum_{i=1}^{K} \rho^{-(i-1)}} := \rho^{K-k} \cdot \frac{\sum_{j=0}^{k-1} \rho^{j}}{\sum_{j=0}^{K-1} \rho^{j}} \leq \rho^{K-k}.$$

□

Lemma 64. *Let $(X_t)_{t\geq 0}$ be a Markov chain on $\{0, 1, \ldots, N\}$ such that N is absorbing, from each $i \in \{1, \ldots, N-1\}$ the chain moves to $i+1$ with probability p and to $i-1$ with probability q, where $p, q \geq 0$, $p+q=1$ and $p > q$, and from 0 the chain moves to 1 with probability 1. Let $T_N := \inf\{t \geq 0 : X_t = N\}$. Then for every $k \in \{0, 1, \ldots, N\}$,*

$$\mathbb{E}_k[T_N] := \frac{N-k}{p-q} + \frac{2pq}{(p-q)^2}\left[\left(\frac{q}{p}\right)^N - \left(\frac{q}{p}\right)^k\right],$$

and in particular,

$$\mathbb{E}_k[T_N] \leq \frac{N-k}{p-q}.$$

Proof. For $i \in \{0, 1, \ldots, N\}$ define $m_i := \mathbb{E}_i[T_N]$. If $1 \leq i \leq N-1$, conditioning on the first step and using the Markov property gives

$$m_i = 1 + p\, m_{i+1} + q\, m_{i-1}. \tag{46}$$

At N we have $m_N = 0$ since N is absorbing, and at 0 we have

$$m_0 = 1 + m_1 \tag{47}$$

since the chain moves deterministically from 0 to 1 in one step.

Set $\delta := p - q > 0$ and rewrite (46) as

$$p\, m_{i+1} - m_i + q\, m_{i-1} = -1, \qquad 1 \leq i \leq N-1. \tag{48}$$

A particular solution is given by $m_i^{\text{part}} = -i/\delta$, because

$$p\left(-\frac{i+1}{\delta}\right) - \left(-\frac{i}{\delta}\right) + q\left(-\frac{i-1}{\delta}\right) := \frac{-p+q}{\delta} = -1$$

using $p + q = 1$. For the associated homogeneous equation

$$p\, u_{i+1} - u_i + q\, u_{i-1} = 0,$$

a trial solution $u_i = r^i$ leads to the characteristic equation $pr^2 - r + q = 0$, whose two roots are $r = 1$ and $r = q/p$. Writing $\rho := q/p \in (0, 1)$, the general homogeneous solution is $u_i = A + B\rho^i$, hence every solution to (48) has the form

$$m_i = A + B\rho^i - \frac{i}{\delta}. \tag{49}$$

Imposing $m_N = 0$ in (49) yields

$$0 = A + B\rho^N - \frac{N}{\delta} \quad \Longrightarrow \quad A = \frac{N}{\delta} - B\rho^N.$$

Next, (47) gives $m_0 = 1 + m_1$. From (49),

$$m_0 = A + B, \qquad m_1 = A + B\rho - \frac{1}{\delta},$$

so

$$A + B = 1 + \left(A + B\rho - \frac{1}{\delta}\right) \quad \Longrightarrow \quad B(1-\rho) = 1 - \frac{1}{\delta}.$$

Now $1 - \rho = 1 - q/p = (p-q)/p = \delta/p$ and, since $p + q = 1$, we have $1 - \frac{1}{\delta} = \frac{\delta - 1}{\delta} = \frac{(p-q)-(p+q)}{\delta} = -\frac{2q}{\delta}$. Therefore

$$B \cdot \frac{\delta}{p} = -\frac{2q}{\delta} \quad \Longrightarrow \quad B = -\frac{2pq}{\delta^2}.$$

Substituting back gives

$$A = \frac{N}{\delta} + \frac{2pq}{\delta^2}\rho^N,$$

and hence for $k \in \{0, 1, \dots, N\}$,

$$m_k := \frac{N-k}{\delta} + \frac{2pq}{\delta^2}\big(\rho^N - \rho^k\big).$$

This is the stated closed form after recalling $\delta = p - q$ and $\rho = q/p$.

To see that this solution is the unique solution to (46)–(47), suppose $\tilde{m}$ is another solution and let $d_i := m_i - \tilde{m}_i$. Then $d_N = 0$, $d_0 = d_1$, and d solves the homogeneous recurrence $pd_{i+1} - d_i + qd_{i-1} = 0$, so $d_i = A' + B'\rho^i$. The condition $d_0 = d_1$ implies $A' + B' = A' + B'\rho$, hence $B'(1-\rho) = 0$ and therefore $B' = 0$, and then $d_N = 0$ forces $A' = 0$, so $d \equiv 0$.

Finally, since $\rho \in (0,1)$ and $k \le N$, we have $\rho^N \le \rho^k$ and thus $\rho^N - \rho^k \le 0$. Because $2pq/\delta^2 \ge 0$, the correction term $\frac{2pq}{\delta^2}(\rho^N - \rho^k)$ is nonpositive, and therefore

$$\mathbb{E}_k[T_N] = m_k \le \frac{N-k}{\delta} = \frac{N-k}{p-q}.$$

□

Lemma 65. *Let $(X_t)_{t\ge 0}$ be a nearest–neighbor random walk with*

$$\Pr(X_{t+1} = X_t + 1) = p, \qquad \Pr(X_{t+1} = X_t - 1) = q = 1 - p, \qquad p - q \ge c > 0,$$

run on a chain of length L with the right endpoint absorbing. *Assume the walk starts at distance k from the absorbing endpoint (i.e. $X_0 = L - k$), where k is a fixed constant independent of L. Let*

$$\tau := \inf\{t \ge 0 : \ X_t = L\}$$

be the absorption time at the right edge. Then for any $t \ge 1$,

$$\Pr(\tau > t) \ \le \ \Pr\left(\sum_{i=1}^{t} \xi_i \le k - 1\right), \qquad \xi_i \in \{+1, -1\}, \ \ \mathcal{E}[\xi_i] = p - q =: \mu \ (\ge c),$$

and by Hoeffding's inequality,

$$\Pr(\tau > t) \ \le \ \exp\left(-\frac{(\mu t - (k-1))^2}{2t}\right) \ \le \ \exp\left(-\frac{(ct - (k-1))^2}{2t}\right).$$

In particular, taking $t = \lfloor L^{1/4} \rfloor$ and using that k is constant, for all L large enough (e.g. $L^{1/4} \ge 2(k-1)/c$),

$$\Pr\big(\tau > \lfloor L^{1/4} \rfloor\big) \ \le \ \exp\left(-\frac{c^2}{8}\, L^{1/4}\right).$$

Proof. Write $X_t = X_0 + \sum_{i=1}^{t} \xi_i$ with i.i.d. increments $\xi_i \in \{+1, -1\}$ of mean $\mu = p - q \ge c$. If $\tau > t$ then $X_t \le L - 1$, hence $\sum_{i=1}^{t} \xi_i = X_t - X_0 \le (L-1) - (L-k) = k - 1$, proving the first inequality. Hoeffding's inequality for bounded i.i.d. variables yields

$$\Pr\left(\sum_{i=1}^{t} \xi_i - \mu t \le -(\mu t - (k-1))\right) \le \exp\left(-\frac{(\mu t - (k-1))^2}{2t}\right),$$

and $\mu \ge c$ gives the stated simplification. For $t = \lfloor L^{1/4} \rfloor$ and $L^{1/4} \ge 2(k-1)/c$, we have $\mu t - (k-1) \ge ct/2$, giving the final display. □

Lemma 66. *Let $x, y : [0, T] \to \mathbb{R}$ be differentiable functions with initial values $x(0) = a$ and $y(0) = b$. Assume that for every $t \in [0, T]$,*

$$x'(t) \ge 0, \qquad y'(t) \le 0,$$

and there exists a constant $c > 0$ such that

$$x'(t) \ \ge \ c\big(-y'(t)\big).$$

Then for every $t \in [0, T]$,

$$x(t) - a \ \ge \ c\big(b - y(t)\big),$$

equivalently,

$$x(t) + c\, y(t) \ \ge \ a + c\, b.$$

Proof. Define $F(t) := x(t) + c\,y(t)$. Then F is differentiable and

$$F'(t) = x'(t) + c\,y'(t) \;\geq\; 0 \qquad \text{for all } t \in [0,T],$$

by the assumed inequality $x'(t) \geq c(-y'(t))$. Hence F is nondecreasing on $[0,T]$, so

$$x(t) + c\,y(t) = F(t) \;\geq\; F(0) = x(0) + c\,y(0) = a + c\,b.$$

Rearranging gives $x(t) - a \geq c(b - y(t))$, as claimed. □

Lemma 67. *Fix $C > 0$, $K > 0$, $\varepsilon \in (0,1)$, $d > 0$, and $D \in \mathbb{R}$. Let $\alpha \in C^1([0,\infty))$ satisfy*

$$\dot\alpha(t) > K \qquad \textit{for all } t \geq 0,$$

and define

$$g(t) := \frac{e^{\alpha(t)-D}}{e^{\alpha(t)-D} + C} \in (0,1).$$

Let $f : [0,\infty) \to (0,1)$ be continuous and assume

$$f(0) = g(0) \qquad \textit{and} \qquad f(t) > g(t) \;\textit{ for all } t > 0.$$

If $f(0) < 1 - \varepsilon$, then there exists $T_\varepsilon \in (0, \bar T_\varepsilon)$ such that

$$f(T_\varepsilon) = 1 - \varepsilon,$$

where

$$\bar T_\varepsilon := \frac{1}{K} \log\!\left(\frac{1-\varepsilon}{\varepsilon} \cdot \frac{1 - f(0)}{f(0)} \right).$$

Moreover,

$$\alpha(T_\varepsilon) - \alpha(0) \;<\; \log\!\left(\frac{1-\varepsilon}{\varepsilon} \cdot \frac{1 - f(0)}{f(0)} \right).$$

Proof. Set

$$\Delta_\varepsilon := \log\!\left(\frac{1-\varepsilon}{\varepsilon} \cdot \frac{1 - f(0)}{f(0)} \right), \qquad \bar T_\varepsilon := \frac{\Delta_\varepsilon}{dK}.$$

From $f(0) = g(0) = \frac{e^{\alpha(0)-D}}{e^{\alpha(0)-D}+C}$ we obtain

$$e^{\alpha(0)-D} = C \frac{f(0)}{1 - f(0)}, \qquad \text{hence} \qquad \alpha(0) - D = \log\!\left(C \frac{f(0)}{1 - f(0)} \right).$$

Consequently,

$$(\alpha(0) - D) + \Delta_\varepsilon = \log\!\left(C \frac{f(0)}{1 - f(0)} \right) + \log\!\left(\frac{1-\varepsilon}{\varepsilon} \cdot \frac{1 - f(0)}{f(0)} \right) = \log\!\left(C \frac{1-\varepsilon}{\varepsilon} \right).$$

Since $\dot\alpha(t) > K$ for all $t \geq 0$, integration yields

$$\alpha(t) - \alpha(0) = \int_0^t \dot\alpha(s)\, ds > Kt \qquad (t > 0).$$

Subtracting D gives

$$\alpha(t) - D > (\alpha(0) - D) + Kt \qquad (t > 0).$$

In particular,

$$\alpha(\bar T_\varepsilon) - D > (\alpha(0) - D) + K \bar T_\varepsilon = (\alpha(0) - D) + \Delta_\varepsilon = \log\!\left(C \frac{1-\varepsilon}{\varepsilon} \right).$$

The map $x \mapsto \frac{e^x}{e^x + C}$ is strictly increasing, hence

$$g(\bar T_\varepsilon) > \frac{e^{\log(C(1-\varepsilon)/\varepsilon)}}{e^{\log(C(1-\varepsilon)/\varepsilon)} + C} = 1 - \varepsilon.$$

By assumption $f(t) > g(t)$ for $t > 0$, thus $f(\bar{T}_\varepsilon) > g(\bar{T}_\varepsilon) > 1 - \varepsilon$, while $f(0) \leq 1 - \varepsilon$. By continuity of f, the intermediate value theorem implies the existence of $T_\varepsilon \in (0, \bar{T}_\varepsilon)$ such that $f(T_\varepsilon) = 1 - \varepsilon$.

Finally, since $T_\varepsilon > 0$ and $f(T_\varepsilon) > g(T_\varepsilon)$, we have $g(T_\varepsilon) < 1 - \varepsilon$. By monotonicity of $x \mapsto \frac{e^x}{e^x + C}$ it follows that

$$\alpha(T_\varepsilon) - D < \log\left(C\frac{1-\varepsilon}{\varepsilon}\right).$$

Subtracting $\alpha(0) - D = \log\left(C\frac{f(0)}{1-f(0)}\right)$ yields

$$\alpha(T_\varepsilon) - \alpha(0) < \log\left(C\frac{1-\varepsilon}{\varepsilon}\right) - \log\left(C\frac{f(0)}{1-f(0)}\right) = \log\left(\frac{1-\varepsilon}{\varepsilon} \cdot \frac{1-f(0)}{f(0)}\right).$$

□

Lemma 68. *Fix $n \geq 3$. Consider an input sequence $s_{1:L}$ sampled autoregressively as described in Section 3, such that the prefix $s_{1:2n-2}$ contains the multiset of edges G of two disjoint directed chains $\mathcal{C}_a = (a_1, \ldots, a_n)$ and $\mathcal{C}_b = (b_1, \ldots, b_n)$, and all subsequent tokens $s_{2n-1}, \ldots, s_L$ are vertices in $\mathcal{C}_a \cup \mathcal{C}_b$ (in particular, the last token $v = s_L$ is a vertex).*

Assume the transformer's parameters satisfy:

1. *The attention matrix has the symmetric form $A = A(\alpha, \beta, \gamma)$ on the vertex–edge block as in Lemma 61 and γ on all vertex–vertex blocks.*
2. *The value matrix has the following form: there exist $\alpha', \beta', \gamma' \in \mathbb{R}$ such that for every edge token $(u, w) \in \mathcal{E}$ and any output token $k \in \Sigma$,*

$$V_{(u,w),k} = \begin{cases} \alpha' & \text{if } k = u, \\ \beta' & \text{if } k = w, \\ \gamma' & \text{otherwise}, \end{cases} \qquad \text{and} \qquad V_{v,:} = 0 \text{ for all vertex tokens } v \in \mathcal{V}.$$

For Softmax Attention, let $Z_v = \sum_{j=1}^{L} \exp(A_{s_L, s_j})$ be the partition function. We define the weight function $w(\mu)$ as:

$$w(\mu) = \begin{cases} \mu & \text{for Linear Attention}, \\ e^\mu / Z_v & \text{for Softmax Attention}. \end{cases}$$

Note that for any token x, the attention weight from v to x is given by $w(A_{v,x})$.

Then the logits $H^{(1)}_{L,:}$ depend on v, G, and in the softmax case, the partition function Z_v (we denote them by $o(v, G, Z_v)$). In the case of Linear Attention, the dependence on Z_v is absent, and we simply write $o(v, G)$. Furthermore, $o(v, G, Z_v)_k = \Psi_k + C$, where C is a constant independent of k, we define $\Delta_\alpha \coloneqq \alpha' - \gamma'$, $\Delta_\beta \coloneqq \beta' - \gamma'$, and Ψ_k is given by the following cases (and symmetrically when swapping $\mathcal{C}_a$ and $\mathcal{C}_b$).

Case 1: v is a middle vertex. *If $v = a_i$ for some $i \in \{3, \ldots, n-2\}$ (and symmetrically if $v = b_i$), then:*

$$\Psi_k = \begin{cases} \Delta_\alpha w(\gamma) + \Delta_\beta w(\alpha), & k = a_{i+1}, \\ \Delta_\alpha w(\beta) + \Delta_\beta w(\gamma), & k = a_{i-1}, \\ \Delta_\alpha w(\alpha) + \Delta_\beta w(\beta), & k = a_i, \\ \Delta_\alpha w(\gamma), & k \in \{a_1, b_1\}, \\ \Delta_\beta w(\gamma), & k \in \{a_n, b_n\}, \\ (\Delta_\alpha + \Delta_\beta) w(\gamma), & k \in (\mathcal{C}_a \cup \mathcal{C}_b) \setminus \{a_{i-1}, a_i, a_{i+1}, a_1, a_n, b_1, b_n\}, \\ 0, & k \in \mathcal{E}. \end{cases}$$

Case 2: v is the penultimate vertex. *If $v = a_{n-1}$ (and symmetrically if $v = b_{n-1}$), then:*

$$\Psi_k = \begin{cases} \Delta_\beta w(\alpha), & k = a_n, \\ \Delta_\alpha w(\beta) + \Delta_\beta w(\gamma), & k = a_{n-2}, \\ \Delta_\alpha w(\alpha) + \Delta_\beta w(\beta), & k = a_{n-1}, \\ \Delta_\alpha w(\gamma), & k \in \{a_1, b_1\}, \\ \Delta_\beta w(\gamma), & k \in \{b_n\}, \\ (\Delta_\alpha + \Delta_\beta) w(\gamma), & k \in (\mathcal{C}_a \cup \mathcal{C}_b) \setminus \{a_{n-2}, a_{n-1}, a_n, a_1, b_1, b_n\}, \\ 0, & k \in \mathcal{E}. \end{cases}$$

Case 3: v is the second vertex (and not penultimate). *If $m \geq 4$ and $v = a_2$ (and symmetrically if $v = b_2$), then:*

$$\Psi_k = \begin{cases} \Delta_\alpha w(\gamma) + \Delta_\beta w(\alpha), & k = a_3, \\ \Delta_\alpha w(\beta), & k = a_1, \\ \Delta_\alpha w(\alpha) + \Delta_\beta w(\beta), & k = a_2, \\ \Delta_\beta w(\gamma), & k \in \{a_n, b_n\}, \\ \Delta_\alpha w(\gamma), & k \in \{b_1\}, \\ (\Delta_\alpha + \Delta_\beta) w(\gamma), & k \in (\mathcal{C}_a \cup \mathcal{C}_b) \setminus \{a_1, a_2, a_3, a_n, b_1, b_n\}, \\ 0, & k \in \mathcal{E}. \end{cases}$$

Case 4: v is the first vertex (and not penultimate). *If $m \geq 3$ and $v = a_1$ (and symmetrically if $v = b_1$), then:*

$$\Psi_k = \begin{cases} \Delta_\alpha w(\gamma) + \Delta_\beta w(\alpha), & k = a_2, \\ \Delta_\alpha w(\alpha), & k = a_1, \\ \Delta_\beta w(\gamma), & k \in \{a_n, b_n\}, \\ \Delta_\alpha w(\gamma), & k \in \{b_1\}, \\ (\Delta_\alpha + \Delta_\beta) w(\gamma), & k \in (\mathcal{C}_a \cup \mathcal{C}_b) \setminus \{a_1, a_2, a_n, b_1, b_n\}, \\ 0, & k \in \mathcal{E}. \end{cases}$$

Case 5: v is a terminal vertex. *If $v = a_n$ (and symmetrically if $v = b_n$), then:*

$$\Psi_k = \begin{cases} \Delta_\beta w(\beta), & k = a_n, \\ \Delta_\alpha w(\beta) + \Delta_\beta w(\gamma), & k = a_{n-1}, \\ \Delta_\alpha w(\gamma), & k \in \{a_1, b_1\}, \\ \Delta_\beta w(\gamma), & k \in \{b_n\}, \\ (\Delta_\alpha + \Delta_\beta) w(\gamma), & k \in (\mathcal{C}_a \cup \mathcal{C}_b) \setminus \{a_{n-1}, a_n, a_1, b_1, b_n\}, \\ 0, & k \in \mathcal{E}. \end{cases}$$

Proof. Let $s_{1:L}$ denote the input sequence with $s_L = v$. From the transformer architecture (Section 2.2), the output logits at position L are given by the attention-weighted sum of value vectors. Specifically:

$$H^{(1)}_{L,:} = \sum_{j=1}^{L} \text{weight}(s_L, s_j) V_{s_j,:},$$

where $\text{weight}(s_L, s_j)$ is the attention weight of token s_j when attending from s_L. Using our definition of $w(\mu)$ and the attention matrix entries A_{s_L,s_j}, this weight is exactly $w(A_{s_L,s_j})$.

By our assumption on the value matrix, $V_{s_j,:} = 0$ whenever s_j is a vertex token. Since the sequence $s_{1:L}$ consists of edge tokens (the set G) and vertex tokens, the sum over $j = 1, \ldots, L$ restricts to the indices corresponding to edges in G:

$$H^{(1)}_{L,:} = \sum_{e \in G} w(A_{v,e}) V_{e,:}.$$

This expression confirms that the logits depend on v, G, and implicitly on the context via Z_v (in the Softmax case), justifying the notation $o(v, G, Z_v)$.

For a specific output token $k \in \Sigma$, we have:

$$o(v, G, Z_v)_k = \sum_{e \in G} w(A_{v,e}) V_{e,k}.$$

Using the symmetric structure of V where $V_{e,k} = \alpha'$ if k is the source of e, β' if k is the target of e, and γ' otherwise (including when k is an edge token), we substitute:

$$\begin{aligned} o(v, G, Z_v)_k &= \sum_{e=(u,w) \in G} w(A_{v,e}) \left(\alpha' \mathbb{1}\{k = u\} + \beta' \mathbb{1}\{k = w\} + \gamma' \mathbb{1}\{k \notin \{u, w\}\}\right) \\ &= \alpha' \sum_{e \in G, \text{src}(e) = k} w(A_{v,e}) + \beta' \sum_{e \in G, \text{tgt}(e) = k} w(A_{v,e}) + \gamma' \sum_{e \in G, k \notin e} w(A_{v,e}). \end{aligned}$$

Let $W_{out}(v, k) = \sum_{e \in G, \text{src}(e) = k} w(A_{v,e})$ be the total attention weight on edges outgoing from k, and $W_{in}(v, k) = \sum_{e \in G, \text{tgt}(e) = k} w(A_{v,e})$ be the total weight on edges incoming to k. The total weight is $W_{total}(v) = \sum_{e \in G} w(A_{v,e})$. We can rewrite the third term using the identity $\mathbb{1}\{k \notin e\} = 1 - \mathbb{1}\{k = \text{src}(e)\} - \mathbb{1}\{k = \text{tgt}(e)\}$:

$$\gamma' \sum_{e \in G, k \notin e} w(A_{v,e}) = \gamma' \left(W_{total}(v) - W_{out}(v, k) - W_{in}(v, k)\right).$$

Substituting this back:

$$\begin{aligned} o(v, G, Z_v)_k &= \alpha' W_{out}(v, k) + \beta' W_{in}(v, k) + \gamma' \left(W_{total}(v) - W_{out}(v, k) - W_{in}(v, k)\right) \\ &= (\alpha' - \gamma') W_{out}(v, k) + (\beta' - \gamma') W_{in}(v, k) + \gamma' W_{total}(v). \end{aligned}$$

Defining $\Delta_\alpha = \alpha' - \gamma'$, $\Delta_\beta = \beta' - \gamma'$, and $C = \gamma' W_{total}(v)$ (which is independent of k), we get:

$$o(v, G, Z_v)_k = \Delta_\alpha W_{out}(v, k) + \Delta_\beta W_{in}(v, k) + C.$$

We define $\Psi_k = \Delta_\alpha W_{out}(v, k) + \Delta_\beta W_{in}(v, k)$. It remains to compute $W_{out}(v, k)$ and $W_{in}(v, k)$ in each of the configurations listed in the lemma statement. Since G is the disjoint union of two directed chains, each $k \in (\mathcal{C}_a \cup \mathcal{C}_b)$ has at most one outgoing edge $(k, \text{succ}(k))$ (absent iff $k \in \{a_n, b_n\}$) and at most one incoming edge $(\text{pred}(k), k)$ (absent iff $k \in \{a_1, b_1\}$). For such edges, by the assumed form of $A(\alpha, \beta, \gamma)$ on the vertex–edge block,

$$A_{v,(k,\text{succ}(k))} = \begin{cases} \alpha, & v = k, \\ \beta, & v = \text{succ}(k), \\ \gamma, & \text{otherwise}, \end{cases} \qquad A_{v,(\text{pred}(k),k)} = \begin{cases} \alpha, & v = \text{pred}(k), \\ \beta, & v = k, \\ \gamma, & \text{otherwise}. \end{cases}$$

Applying $w(\cdot)$ and substituting into $\Psi_k = \Delta_\alpha W_{out}(v, k) + \Delta_\beta W_{in}(v, k)$ gives exactly the case-by-case expressions stated in the lemma (with the $\mathcal{C}_b$ cases obtained by symmetry). □

I Additional Experiments and Experimental Details

I.1 Theoretically Inspired Setting

We corroborate our theoretical findings from Sections 4 and 5 in the chain identification task of Section 3. Each example is specified by a dependency graph consisting of two directed chains, together with a starting vertex and a queried terminal vertex. The model must output a chain of intermediate variables that identifies the correct dependency path. We train single-layer linear Transformers, as defined in Section 2, without the simplifying assumptions from Section 5.2: the attention and value matrices are trained jointly with policy gradient, implemented using REINFORCE [71], from a small Gaussian initialization. Full experimental details and hyperparameters are provided in Appendix I.1.4.

Table 2: In line with our theory (Section 5), on the chain identification task (Section 3), a Transformer trained with simple examples achieves high accuracy through the efficient chain traversal (reasoning) algorithm, thereby demonstrating an implicit bias towards efficiency (Section 4). In this table, "Test Acc." and "Chain Trav." refer to the fraction of test examples that are solved accurately and using the chain traversal algorithm, respectively. For further details on this experiment see Appendices I.1.1 and I.1.4.

Chain size n	Test Acc.	Chain Trav.
4	100%	100%
8	100%	99.3%
12	99.3%	94.2%

I.1.1 Emergence of Efficient Reasoning

Table 2 reports the results of an experiment where: *(i)* the chain length n takes a value in $\{4, 8, 12\}$; *(ii)* the training data distribution is drawn from a uniform distribution over $[n-1]$ (Section 3.1), meaning in particular that it has simplicity mass greater than zero (Definition 1); and *(iii)* the test data distribution is identical to the training data distribution. As can be seen in the table, in line with our theory (Section 5.3), the learned Transformers achieve extremely high test accuracy through the efficient chain traversal algorithm (Section 3.3), thereby demonstrating an implicit bias towards efficiency (recall that, as proven in Section 4.2, algorithms far less efficient than chain traversal also solve the chain identification task).

I.1.2 Out-of-Distribution Generalization

To demonstrate our theoretical prediction of out-of-distribution generalization (Section 5.3), we conduct an experiment where: *(i)* the chain length n equals 12; *(ii)* the training data distribution is drawn from a uniform distribution over $\{n-4, n-3, n-2, n-1\}$ (Section 3.1), meaning in particular that it has 4-simplicity mass of one (Definition 1), i.e., all of its examples require at most four reasoning steps; and *(iii)* the test data distribution is drawn from a uniform distribution over $\{n-s, n-s+1, \ldots, n-1\}$, where $s \in \{4, 8, 11\}$. In this experiment, when $s = 4$ the test data distribution is identical to the training data distribution, whereas when s is larger the test data distribution entails examples more complex than those seen in training. The results of the experiment were perfect: under all test data distributions, the learned Transformer solved all test examples using the efficient chain traversal algorithm (Section 3.3). [21] This confirms that the Transformer indeed generalizes out of distribution, including to complex examples unseen during training. For further details on the experiment see Appendix I.1.4.

I.1.3 Solving Complex Tasks Requires Training on Simple Tasks

Figure 2 reports the results of an experiment where: *(i)* the chain length n equals 5; *(ii)* the training data distribution is drawn from a uniform distribution over $[k]$ (Section 3.1), where $k \in [n-1]$; and *(iii)* the test data distribution is drawn from a distribution that places all probability mass on 1. In this experiment the test data distribution entails only the most complex examples (those that require the maximal number of reasoning steps), whereas the training data distribution entails only complex examples when k is small, and a mix of complex and simple examples when k is large. As can be seen in the figure, in line with our theory (Sections 5.3 and 5.4), the accuracy of the Transformer on complex test examples is higher when the training data includes more simple examples, despite such simple examples inducing a distribution shift.

I.1.4 Experimental Details

We use the following implementation throughout the theoretically inspired experiments. We use a single-layer transformer with attention matrix $A \in \mathbb{R}^{|\Sigma|\times|\Sigma|}$ and value matrix $V \in \mathbb{R}^{|\Sigma|\times|\Sigma|}$, as defined in Section 2.2. We initialize both matrices with small random values sampled from a normal distribution with standard deviation 10^{-8}. We optimize $\mathcal{L}(\mathcal{D}, \theta)$ using the REINFORCE

[21] Interestingly, these results surpass those in Appendix I.1.1. This is another example (on top of others in Section 6 and Appendix I.1) where replacing in-distribution complex examples with simple examples may be more effective than training on the target distribution.

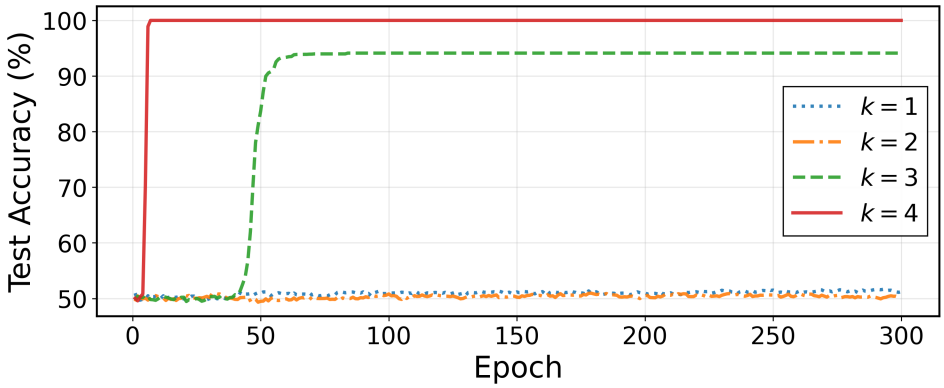


Figure 2: In line with our theory (Section 5), on the chain identification task (Section 3), the accuracy of a Transformer on complex test examples is higher when its training data includes simple examples, despite such simple examples inducing a distribution shift. This figure shows the accuracy on complex test examples throughout training, for different training data distributions: ones entailing only complex examples (small k), and ones entailing a mix of complex and simple examples (large k). For further details on this experiment see Appendices I.1.3 and I.1.4.

algorithm [71] with the advantage function computed as the difference between the reward and the average reward across the batch. We use the Adam optimizer with learning rate 0.01. For each chain size, we train on $1{,}000{,}000$ samples over 300 epochs with batch size $50{,}000$, and validate on $1{,}000$ samples. During evaluation, we measure test accuracy and chain traversal correctness using greedy decoding (temperature 0). For experiments with a single chain size (see Appendices I.1.1 and I.1.3), we set max new tokens to $n + 1$ where n is the chain size; for generalization experiments across multiple chain sizes (see Appendix I.1.2), we use $\max(\text{test chain sizes}) + 1$. All experiments were implemented using a single NVIDIA RTX A6000 GPU.

I.2 Real-World Experiments

All real-world pre-training experiments were implemented using eight NVIDIA RTX A6000 GPUs.

I.2.1 Synthetic Pre-Training Setting

This section provides full experimental context for the *synthetic pre-training* setting of Section 6 (Table 1). This setting follows the controlled reasoning framework of Zhang et al. [73], and differs from the real-world pre-training setting in Appendix I.2.2 in one central respect: the base model has already been pre-trained on chain-of-thought traces from the same synthetic reasoning family. It therefore tests our prediction in a setting where the model has task-specific reasoning primitives before RL, but must still generalize to examples requiring more reasoning steps than those encountered during pre-training.

Task and Complexity.

We use the synthetic arithmetic reasoning task introduced by Zhang et al. [73]. Each problem is generated from a directed dependency graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, whose nodes represent intermediate quantities and whose directed edges represent arithmetic dependencies. The graph is rendered into a natural-language math problem, and the model must produce a parseable chain-of-thought solution that computes the designated answer node. Following Zhang et al. [73], we measure the complexity of an instance by the number of arithmetic operations,

$$\mathrm{op}(\mathcal{G}) = |\mathcal{E}|,$$

which controls the amount of compositional reasoning required to solve the problem.

Base Model and Pre-Training.

The base model is the same 100M-parameter decoder-only Qwen2.5-style Transformer used by Zhang et al. [73]. It is pre-trained for 10B tokens on synthetic chain-of-thought data from the task family above, using their pre-training recipe: 20% examples with two to four arithmetic operations, 30% examples with five to seven arithmetic operations, and 50% examples with eight to ten arithmetic operations. Thus, the model is exposed during pre-training to reasoning traces up to roughly ten operations.

Post-Training Distributions.

We keep the architecture, optimizer, GRPO setup, prompt format, reward, and other hyperparameters identical to Zhang et al. [73]; the only aspect varied for our comparison is the post-training data distribution. For a target complexity n, *Hard-*n denotes post-training only on examples with $\mathrm{op}(\mathcal{G}) = n$. By contrast, *Mixed-*$n$ denotes a distribution that samples uniformly from operation counts between the edge of the pre-training range and the target complexity, i.e. $\mathrm{op}(\mathcal{G}) \in \{10, 11, \ldots, n\}$.

All runs use 200 GRPO optimization steps. Training-time validation is performed every ten steps on the same *Hard*-20 subset used for final evaluation.

Evaluation.

We use the final-answer checker implemented in the code of Zhang et al. [73]. An output is counted as correct if its extracted final answer matches the ground truth. We report `pass@128` accuracy on the corresponding *Hard-*n test distribution; in Table 1, this is *Hard*-20. For each test example, we sample 128 completions at temperature 1 from the final checkpoint and count the example as solved if at least one final answer is correct. The model retains its $2{,}048$-token context, with prompt and response limits of $1{,}024$ tokens each. The prompt-length filter retains 197 of the 200 *Hard*-20 test examples; all synthetic pre-training results use this same subset.

I.2.2 Real-World Pre-Training Setting

This section provides full experimental details for the *real-world pre-training* setting of Section 6. Unlike the synthetic pre-training setting (Appendix I.2.1), in which the base model was pre-trained on chain-of-thought traces of the specific reasoning task, here RL is applied directly to standard open-weight language models with broad language pre-training but *no prior task-specific exposure*.

We fine-tune two such base models, Qwen2.5-3B-Instruct [56] (Qwen/Qwen2.5-3B-Instruct) and Llama-3.2-3B-Instruct (meta-llama/Llama-3.2-3B-Instruct), using GRPO [62]. The code builds on philschmid/deep-learning-pytorch-huggingface and is adapted to TRL 1.2 / Transformers 5.5.

This appendix provides comprehensive details for reproducing the real-world experiments. Section I.2.2 describes the mathematical reasoning task, including the equation system construction, graph generation procedure, complexity measures, and the mixed/hard data distributions used for training. Section I.2.2 details the training procedure, including the architecture and reasoning framework, hyperparameter settings (Table 3), and the prompt template used for both fine-tuning and evaluation. Finally, Section I.2.2 presents the results for both base models (Qwen and Llama), showing learning curves across different data distributions and providing concrete examples of model completions from the evaluation set.

Training Data.

Task description. We construct a mathematical reasoning task in which the input is presented as natural-language text containing a *shuffled* list of affine equations over the binary field $\mathrm{GF}(2)$, of the form

$$x_v \;=\; \Big(\sum_{u \in \mathrm{Pa}(v)} x_u \;+\; c_v\Big) \bmod 2, \qquad c_v \in \{-1, 0, +1\},$$

where $\mathrm{Pa}(v)$ denotes the set of parent variables whose values are needed to compute x_v. Each input also contains one or more *constant assignments* $x_r = c_r$ ($c_r \in \{0, 1\}$) for the root variables, and a query for a specified *target variable* x_t (see Figures 5 and 6 for examples of inputs and their corresponding labels). The dependency structure of the equation system forms a directed acyclic graph (DAG) with up to two parents per non-root node ($\text{in-deg} \leq 2$). The model is trained to compute the value of x_t, which is located at the sink of the DAG.

Complexity measure. For a given example, let the ancestor graph of the target be the subgraph containing all variables that can influence x_t. We define the example's complexity as the depth of this ancestor graph, i.e., the length of the longest directed path from a constant-assigned root variable to the target.

Graph generation. We sample DAGs conditioned on this depth. For target depth $D-1$, we partition the variables into D ordered buckets and force one "spine" path to pass through one variable in each bucket, ending at the target. This guarantees an ancestor path of the desired length. We then add additional edges only from earlier buckets to later buckets, preserving acyclicity and the in-degree cap of two, so the graph may contain branching dependencies while retaining controlled depth. After sampling the graph, we randomize variable labels, sample root values uniformly from $\{0, 1\}$, assign offsets $c_v \in \{-1, 0, +1\}$, and shuffle the resulting equations before concatenating them into the prompt.

Data distributions (mixed/hard). For a maximum dependency depth n, we denote by n-Mixed the mixed data distribution in which the ancestor-graph depth is sampled uniformly between 1 and $n-1$, and by n-Hard the hard data distribution that contains only examples whose ancestor-graph depth is exactly $n-1$. Throughout we use 15 variables.

Training Setting.

Architecture and training. For each base model, we fine-tune with a method inspired by DeepSeek-R1 [19]. The LLM is prompted to output its reasoning process within `<think>` and `</think>` tags, followed by the final answer within `<answer>` and `</answer>` tags. Crucially, unlike training in DeepSeek-R1, we provide *no explicit instructions on how to reason* (e.g., no prompts to "think step-by-step"); the LLM must discover effective reasoning strategies on its own. We employ GRPO [62], where the reward depends solely on the correctness of the final answer and adherence to the specified output format, with no supervision provided for the intermediate reasoning steps.

Hyperparameters. We fine-tune Qwen2.5-3B-Instruct and Llama-3.2-3B-Instruct using GRPO with the following configuration. We use a KL coefficient $\beta = 10^{-4}$, generate 8 completions per prompt with sampling temperature 0.9 during training, and set the max completion length to 1024 tokens. We optimize with AdamW using a learning rate of 5×10^{-7} with a cosine learning-rate scheduler and a warmup ratio of 0.03. Training is performed in bfloat16 with FlashAttention 2 across 8 GPUs (DeepSpeed ZeRO-3, vLLM colocated on every training rank). We use a per-device batch size of 8 generations and 8 gradient accumulation steps, training for 800 optimization steps in total (Table 3 summarizes the hyperparameters). Evaluation is performed over 1,000 samples drawn from the 15-Hard distribution (i.e. ancestor-graph depth fixed at 14). All test evaluations use greedy decoding (temperature 0) and a max completion length of 2,048 tokens.

Prompt template. The exact prompt template used for fine-tuning and evaluation is shown in Figure 3. The `{equation_system}` placeholder is replaced at runtime with a serialized, shuffled equation system as described in Appendix I.2.2.

Results.

Training accuracy. Figure 4 reports the training curves for Qwen2.5-3B-Instruct and Llama-3.2-3B-Instruct. For both base models, post-training on 15-Mixed improves accuracy on the 15-Hard evaluation distribution relative to post-training only on 15-Hard, matching the qualitative behavior predicted by our theory. [22]

Table 4 additionally reports the final Llama test accuracies in the same format as the real-world pre-training rows of Table 1.

Example completions. Figures 5 and 6 show randomly drawn 15-Hard evaluation examples together with actual greedy-decoded completions from Qwen2.5-3B-Instruct and Llama-3.2-3B-Instruct respectively. Both examples illustrate the learned strategy: identify the relevant dependencies and then compute the target value forward.

[22] The plots are produced by sliding a window of the most recent 300 training prompts over the chronological log of completions, computing the average correctness inside each window, and smoothing the resulting curve with a 300-sample moving average. Each x-axis tick corresponds to a single optimization step (512 generations / 64 unique prompts per step). The semi-transparent line is the unsmoothed window accuracy; the solid line is the smoothed trend.

Table 3: Hyperparameters used for fine-tuning Qwen2.5-3B-Instruct and Llama-3.2-3B-Instruct on the equations-reasoning task.

Hyperparameter	**Value**
RL algorithm	GRPO
KL coefficient β	10^{-4}
Generations per prompt	8
Sampling temperature (training)	0.9
Max completion length (training)	1,024 tokens
Max prompt length (training)	512 tokens
Optimizer	AdamW
Learning rate	5×10^{-7}
LR schedule	cosine, warmup ratio 0.03
Per-device batch size	8 generations
Gradient accumulation steps	8
Optimization steps	800
Number of GPUs	8
Precision	bfloat16 + FlashAttention 2
Distributed strategy	DeepSpeed ZeRO-3, vLLM colocated
Variables per example n	15
Max in-degree	2
Edge probability	0.2
Modulus	2
Constant range c_v	$\{-1, 0, +1\}$, $\Pr(c_v = 0) = 1 - 0.6\overline{6}$
Eval samples	1,000
Eval distribution	15-Hard (depth $= 14$)
Decoding (eval)	greedy (temperature 0)
Max completion length (eval)	2,048 tokens

Table 4: Same setting as the real-world pre-training experiment in Table 1, but using Llama-3.2-3B-Instruct as the real-world pre-trained base model. "Test Acc." is the fraction of 15-Hard test examples solved correctly.

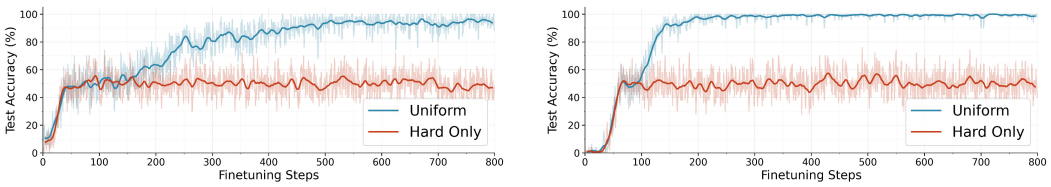
Training Distribution	Test Acc. (*Hard*-15)
Hard-15	50.2%
Mixed-15	99.3%
Mixed-10	98.6%

LLM Annotation for Efficient Reasoning.

To verify that the LLM learns the efficient step-by-step reasoning algorithm described in Section 6.2, we employ automated LLM annotation. We use the real-world pre-training setting for this analysis, since in the synthetic pre-training setting the model is already exposed during pre-training to chain-of-thought traces that demonstrate the efficient algorithm. We use GPT-5.1 with reasoning effort set to "low" to classify whether each solution follows the target algorithm. We sample 100 correct solutions from the 15-Uniform evaluation set and apply binary classification to each. The judge LLM is provided with the system prompt shown in Figure 7.

```
[
{
     "role": "system",
     "content": "You are a helpful assistant that can solve systems of equations.",
},
{
     "role": "user",
     "content": (
     f"Given this equation system: {dag_input}\n\nFind the value of the "
     "target variable. Show your reasoning in <think> </think> tags, and "
     "provide your final answer in <answer> </answer> tags, for example "
     "<answer>12</answer>."
     ),
},
{
     "role": "assistant",
     "content": "I'll think about the equation system and solve it.\n<think>",
},
]
```

Figure 3: Prompt template used for fine-tuning and evaluation in the real-world experiments. The `{equation_system}` placeholder is replaced with the (shuffled) equation system at runtime; the assistant turn is forced to begin with the marker `"I'll think about the equation system and solve it.\n<think>"` so that completions deterministically start inside the `<think>` block.



(a) Qwen2.5-3B-Instruct **(b) Llama-3.2-3B-Instruct**

Figure 4: Training curves in the real-world pre-training setting. In both panels, the model is evaluated on 15-Hard while being post-trained either on 15-Mixed (blue) or 15-Hard (red).

Input.

```
x_12 = (x_2 + x_15 + 1) mod 2.   x_11 = (x_10) mod 2.
x_6  = (x_1 + x_15 - 1) mod 2.   x_3  = (x_4 + 1) mod 2.
x_8  = (x_12 + x_14 - 1) mod 2.  x_13 = (x_9 + x_10 + 1) mod 2.
x_2  = (x_3 + x_10) mod 2.       x_1  = (x_8 + x_10 + 1) mod 2.
x_9  = (x_6 + x_15) mod 2.       x_4  = (x_5 + x_7 - 1) mod 2.
x_5  = (x_7 + x_14) mod 2.       x_15 = (x_11 + 1) mod 2.
@x_10 = 1.@                      x_14 = (x_7 + x_11) mod 2.
x_7  = (x_15 + 1) mod 2.         !Find x_13 (mod 2).!
```

Solution.

```
<think>Starting from the known values:
x_10 = 1.
x_15 = (x_11 + 1) mod 2.
Since x_11 = (x_10) mod 2 and x_10 = 1, then x_11 = 1.
So, x_15 = (1 + 1) mod 2 = 0.
Now we know:
x_7  = (x_15 + 1) mod 2 = (0 + 1) mod 2 = 1.
x_14 = (x_7 + x_11) mod 2 = (1 + 1) mod 2 = 0.
x_5  = (x_7 + x_14) mod 2 = (1 + 0) mod 2 = 1.
x_4  = (x_5 + x_7 - 1) mod 2 = (1 + 1 - 1) mod 2 = 1.
x_3  = (x_4 + 1) mod 2 = (1 + 1) mod 2 = 0.
x_2  = (x_3 + x_10) mod 2 = (0 + 1) mod 2 = 1.
x_8  = (x_12 + x_14 - 1) mod 2.
To find x_12, we use: x_12 = (x_2 + x_15 + 1) mod 2
                          = (1 + 0 + 1) mod 2 = 0.
So, x_8 = (0 + 0 - 1) mod 2 = 1.
Finally, x_1 = (x_8 + x_10 + 1) mod 2
             = (1 + 1 + 1) mod 2 = 3 mod 2 = 1.
Now we can find x_9: x_9 = (x_6 + x_15) mod 2.
We need to find x_6 first: x_6 = (x_1 + x_15 - 1) mod 2.
Using x_1 = 1 and x_15 = 0, x_6 = (1 + 0 - 1) mod 2 = 0.
So, x_9 = (0 + 0) mod 2 = 0.
And finally, x_13 = (x_9 + x_10 + 1) mod 2
                  = (0 + 1 + 1) mod 2 = 2 mod 2 = 0.</think>
<answer>0</answer>
```

Figure 5: A 15-Hard evaluation example and the actual greedy-decoded completion of Qwen2.5-3B-Instruct fine-tuned on 15-Mixed.

Input.

```
x_3  = (x_1 + x_9 + 1) mod 2.    x_6  = (x_5) mod 2.
x_4  = (x_10 + x_12 + 1) mod 2.  x_2  = (x_1 + x_13 - 1) mod 2.
x_8  = (x_1 + 1) mod 2.          x_5  = (x_12 + 1) mod 2.
x_1  = (x_6) mod 2.              x_14 = (x_7) mod 2.
@x_12 = 1.@                      x_13 = (x_12 + x_15 + 1) mod 2.
x_15 = (x_8 + 1) mod 2.          x_10 = (x_8 + x_14) mod 2.
x_9  = (x_11 + x_15 + 1) mod 2.  x_7  = (x_2 - 1) mod 2.
x_11 = (x_4 - 1) mod 2.          !Find x_3 (mod 2).!
```

Solution.

```
<think>
To solve for x_3, we need to start from the bottom and work our way up.
We are given x_12 = 1.
x_15 = (x_8 + 1) mod 2.
x_8 = (x_1 + 1) mod 2.
x_1 = (x_6) mod 2.
x_6 = (x_5) mod 2.
x_5 = (x_12 + 1) mod 2 = (1 + 1) mod 2 = 0.
x_6 = (x_5) mod 2 = (0) mod 2 = 0.
x_1 = (x_6) mod 2 = (0) mod 2 = 0.
x_8 = (x_1 + 1) mod 2 = (0 + 1) mod 2 = 1.
x_15 = (x_8 + 1) mod 2 = (1 + 1) mod 2 = 0.
x_13 = (x_12 + x_15 + 1) mod 2 = (1 + 0 + 1) mod 2 = 0.
x_11 = (x_4 - 1) mod 2.
x_4 = (x_10 + x_12 + 1) mod 2.
x_10 = (x_8 + x_14) mod 2.
x_14 = (x_7) mod 2.
x_7 = (x_2 - 1) mod 2.
x_2 = (x_1 + x_13 - 1) mod 2.
x_1 = 0.
x_13 = 0.
x_2 = (x_1 + x_13 - 1) mod 2 = (0 + 0 - 1) mod 2 = 1.
x_7 = (x_2 - 1) mod 2 = (1 - 1) mod 2 = 0.
x_14 = (x_7) mod 2 = (0) mod 2 = 0.
x_10 = (x_8 + x_14) mod 2 = (1 + 0) mod 2 = 1.
x_4 = (x_10 + x_12 + 1) mod 2 = (1 + 1 + 1) mod 2 = 1.
x_11 = (x_4 - 1) mod 2 = (1 - 1) mod 2 = 0.
x_9 = (x_11 + x_15 + 1) mod 2 = (0 + 0 + 1) mod 2 = 1.
x_3 = (x_1 + x_9 + 1) mod 2 = (0 + 1 + 1) mod 2 = 0.
</think>
<answer>0</answer>
```

Figure 6: A 15-Hard evaluation example and the actual greedy-decoded completion of Llama-3.2-3B-Instruct fine-tuned on 15-Mixed.

```
You are a strict binary classifier.

Given:

(A) a system of equations and a request to find a target variable, and

(B) a solution writeup produced by a model,

Output True iff the writeup's solving strategy is:

1) backtrace from the target variable through dependencies until
reaching an explicit constant assignment,

2) then compute forward along that dependency chain to obtain the target.

Otherwise output False.  Judge the described procedure (not correctness).

Output exactly one word:  True or False.
```

Figure 7: System prompt provided to GPT-5.1 for classifying whether solutions follow the efficient step-by-step reasoning algorithm.

I.2.3 Simplicity and Data Diversity

To isolate the effect of simple examples from increased data diversity, we ran an ablation study in both pre-training settings of Section 6. Within each setting, we held the number of distinct complexity levels, the mass on the hardest level, and the training budget fixed. Including simple examples substantially improves accuracy on hard instances, indicating that this benefit is not explained by the number of distinct complexity levels.

We use the same models, training procedures, and evaluation as in Section 6. As in that section, an example's *complexity* measures the extent of reasoning required to compute its answer. Within each setting, we post-train on two distributions with the same number of distinct complexity levels: one includes simple examples, whereas the other contains only harder examples. We sample complexities uniformly from $\{10, 13, 17, 20\}$ and $\{17, 18, 19, 20\}$ in the synthetic pre-training setting, and from $\{5, 10, 15\}$ and $\{13, 14, 15\}$ in the real-world pre-training setting. Table 5 reports accuracy on *Hard*-n, consisting of examples of complexity n, with $n = 15$ in the real-world pre-training setting and $n = 20$ in the synthetic pre-training setting.

Table 5: Test accuracy on *Hard*-n under post-training distributions with the same number of distinct complexity levels. Within each setting, the first row includes simple examples and the second contains only harder examples.

Real-World Pre-Training		**Synthetic Pre-Training**	
Training Dist. Complexity Levels	Test Acc. (*Hard*-15)	Training Dist. Complexity Levels	Test Acc. (*Hard*-20)
$\{5, 10, 15\}$	96.80%	$\{10, 13, 17, 20\}$	80.20%
$\{13, 14, 15\}$	52.90%	$\{17, 18, 19, 20\}$	48.22%

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: The abstract and introduction state the paper's theoretical claims about outcome-based RL learning efficient reasoning from sufficiently simple examples, and the empirical claims are limited to the synthetic and real-world mathematical reasoning experiments reported in the paper.

   Guidelines:

   - The answer [N/A] 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 [N/A] 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: We discuss limitations in a dedicated Limitations section, including the stylized theoretical setting, assumptions of the analysis, and the restricted scope of the real-world experiments.

   Guidelines:

   - The answer [N/A] 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: The main theoretical results state their assumptions in the main text, and the complete proofs are provided in the appendix.

   Guidelines:

   - The answer [N/A] 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: The main text and appendix describe the experimental tasks, data distributions, training algorithms, hyperparameters, prompts, evaluation protocols, and compute setup used for the reported experiments.

   Guidelines:

   - The answer [N/A] 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: The paper provides detailed reproducibility information and states that code for reproducing the experiments will be made available for the camera-ready version, but the current manuscript does not yet provide an open code release.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.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://neurips.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) necessary to understand the results?

   Answer: [Yes]

   Justification: The experimental section and appendix specify the task construction, train/test distributions, optimizer, RL setup, decoding and evaluation procedures, prompt template, and hyperparameters.

   Guidelines:

   - The answer [N/A] 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: [No]

   Justification: The reported experiments are long-running single-run demonstrations intended to test the qualitative behavior predicted by the theory; the paper does not report error bars or statistical significance tests.

Guidelines:

- The answer [N/A] 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 appendix reports the number of GPUs and distributed training setup used for the real-world experiments, along with the relevant training hyperparameters.

   Guidelines:

   - The answer [N/A] 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: The research uses synthetic mathematical reasoning tasks and open-weight models in a standard experimental setting, and does not involve human subjects, private data, or deployed systems.

   Guidelines:

   - The answer [N/A] 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: [N/A]

Justification: This is foundational research on the training dynamics of reasoning in Transformers and does not introduce a deployed system, dataset, or model with direct societal impact.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] 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., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: We do not release any datasets, generators, or models.

Guidelines:

- The answer [N/A] 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: The paper cites the prior synthetic reasoning framework, the open-weight base models used in the real-world experiments, and the codebase on which the implementation builds, with links where applicable.

Guidelines:

- The answer [N/A] 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: [N/A]

    Justification: The paper does not release a new dataset, benchmark, or model as a standalone asset.

    Guidelines:

    - The answer [N/A] 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: [N/A]

    Justification: Our research does not involve crowdsourcing or human subjects.

    Guidelines:

    - The answer [N/A] 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: [N/A]

Justification: Our research does not involve human subjects.

Guidelines:

- The answer [N/A] 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.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: We use GPT-5.1 to annotate whether generated solutions follow the efficient reasoning algorithm. The annotation procedure and judge prompt are described in Appendix I.2.2.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.