Spatial Induction Heads: In-Context Learning of Multidimensional Cellular Automata

Anonymous Author(s)
Affiliation
Address
email

Abstract

Transformer architectures dominate spatiotemporal modeling, from image recognition to video generation and weather forecasting, exhibiting striking in-context learning (ICL) capabilities when trained autoregressively. Recent theoretical work explains these capabilities through mechanistic constructions like induction heads, but is largely confined to 1D token sequences, leaving the spatial generalization of ICL poorly understood. We study *spatial in-context learning* in the setting of cellular automata: given a spatiotemporal trajectory of an unknown local rule, presented as a flattened 1D token sequence, the model must infer and apply the rule without any explicit description. We formalize *spatial induction heads*, two-layer attention circuits that solve this task by decoupling spatial neighborhood routing from content-based retrieval. Theoretically, we introduce multi-peaked spatial-aware embeddings to prove that the required representation dimension is independent of the total grid size, and we characterize a fundamental routing-decoding tradeoff: concentrating routing into fewer attention heads shifts the decoding burden onto the Layer 1 MLP, whose hidden width must scale exponentially with cells routed per head. Empirically, under a strict out-of-sample evaluation, we show that SGD with standard learned positional embeddings naturally converges to this predicted circuit, achieving near-perfect generalization to unseen rules across 1D elementary cellular automata, 2D Von Neumann neighborhoods, and expanded state spaces.

1 Introduction

Transformer architectures increasingly dominate spatiotemporal modeling [12, 5, 3], despite a purported lack of relevant inductive bias for these domains. Concurrently, large language models (LLMs) based on the same architecture exhibit striking emergent capabilities at scale, including multi-step reasoning, code synthesis, and few-shot generalization to unseen tasks [34].

Many of these capabilities are understood as instances of *in-context learning* (ICL), where model predictions adapt based on input *context* examples without parameter updates [6]. Empirical studies further show ICL reliably emerges in small transformers trained on synthetic tasks like linear regression and function class learning [17].

Currently, our understanding of how and why ICL emerges in trained Transformers focuses on language modeling. Theoretical foundations have been established through studying ICL in synthetic data modeled after language. A notable line of work by [4, 26, 28, 14, 13] examines *statistical induction heads* for predicting the next element of sequences generated by k-order Markov chains, where a sequence's transition probabilities govern element likelihoods based only on the preceding k elements.

Mechanistic interpretability researchers first used induction heads to better understand the inner workings of LLMs [16, 27]. Empirical interpretability work in spatiotemporal domains remains

Submitted to 40th Conference on Neural Information Processing Systems (NeurIPS 2026). Do not distribute.

limited . A notable exception is self-supervised Vision Transformers like DINO [8], which develop attention heads acting as zero-shot object segmenters, while ViT circuit analyses trace how local patch features combine into global representations [38]. Yet no analogous mechanistic account exists for the sequential-to-spatial routing required when a standard Transformer processes flattened spatiotemporal data, and the theoretical foundations for spatiotemporal ICL are entirely absent.

We aim to bridge this gap by studying Transformers' ICL abilities on synthetic spatio-temporal data from cellular automata. A cellular automaton consists of a discrete grid of finite-state cells, evolving synchronously via fixed local rules applied to immediate spatial neighbors [35]. Cellular automata are an intriguing ICL testbed because they contain sequential and spatial regularities; their simple local rules can create complex, chaotic patterns [20] and serve as structured synthetic pretraining data [40]. These spatiotemporal dependencies, where each cell's next state is a deterministic function of fixed neighbors, suggest induction-like pattern matching. Yet, no prior work has demonstrated Transformers implementing this mechanism for spatial data, nor characterized the architectural requirements to do so.

We address this gap using cellular automata as a testbed to study how Transformers perform ICL on spatially structured data. Our contributions are:

1. We introduce *spatial induction heads*, two-layer attention circuits decoupling spatial neighborhood routing from content-based retrieval, and prove that they implement the optimal learning algorithm with a representation dimension strictly independent of total grid size, depending only on the dynamical rule's local parameters (Section 4).
2. We identify a routing-decoding tradeoff unique to the spatial setting: concentrating neighborhood routing into fewer Layer 1 heads reduces the positional embedding cost but requires an exponentially wider MLP to decode the compressed superposition (Section 4).
3. We show standard Transformers with learned positional embeddings trained via gradient descent naturally converge to the predicted spatial induction head circuit, achieving perfect sequence accuracy on 1D elementary cellular automata and extending to 2D neighborhoods and larger state spaces (Section 5).

2 Related work

Transformers for automata. Several works empirically study Transformers using discrete automata. Vafa et al. [31] use deterministic finite automata to diagnose limitations of large Transformers on real data. Closer to our setting, others train smaller Transformers on synthetic cellular automata data [21, 2, 7]. Only Burtsev [7] frames this as in-context generalization to unseen rules; Liu et al. [21] train per-automaton and Berkovich et al. [2] explicitly prompt the rule, with none achieving perfect sequence-level accuracy. Beyond automata, Liu et al. [22] demonstrate that pretrained LLMs perform ICL on discretized chaotic dynamical systems (e.g., logistic maps), and Dai et al. [11] show similar capabilities for hidden Markov processes.

Markov chain ICL theory. Cellular automata and dynamical systems exhibit the Markov property, where the next element depends only on the preceding one. Markov chains are a natural setting for studying such behavior, and k-order Markov chains generalize the setting to a dependence on a finite history of length k. Several theoretical works [4, 24, 23, 26, 28] explain how Transformers achieve optimal ICL on such data via *statistical induction heads* [13]. Notably, Ekbote et al. [14] introduce the first two-layer k-order induction head. Similarly, our spatial induction head attends to k previous tokens, but at arbitrary offsets which result from spatial unrolling. Moreover, we target deterministic rather than statistical prediction, similar to previous copying tasks [19].

Interpretability. Induction heads originate from work on the mechanistic interpretability of language models [27, 16, 10]. In spatiotemporal contexts, DINO [8] yields interpretable attention maps serving as zero-shot segmenters, while recent ViT research examines feature entanglement and local patches' causal influence on the global CLS token [38, 32]. However, these works do not connect directly to ICL or induction heads.

Other ICL theory. An additional line of theoretical work explains ICL through Transformers' ability to implement more general learning (optimization) algorithms like gradient descent [33, 39, 1, 18]. Rather than discrete sequences, these studies often consider supervised pairs (x, y) where only y (and not x) is predicted. Somewhat more closely related, Cole et al. [9] use the same machinery to

study ICL for linear dynamical systems. While closer in principle to spatiotemporal data, this work treats states as generic vectors and do not model any spatial structure. Furthermore, these studies focus on optimization algorithms rather than circuit-level mechanisms like induction heads.

3 Formulating spatial in-context learning

Standard in-context learning (ICL) analyses consider one dimensional sequences. In these settings, positional proximity determines relevance, e.g. the preceeding k tokens determine the next one. To establish ICL mechanisms for spatial reasoning in multiple dimensions, we formulate the prediction of discrete local dynamical systems, also known as cellular automata, as an autoregressive sequence modeling task.

3.1 Local dynamical systems on spatial grids

We define our environment over a d-dimensional discrete grid $\mathcal{G} = \mathbb{Z}_{W_1} \times \cdots \times \mathbb{Z}_{W_d}$ with periodic boundary conditions, yielding a total grid volume $V = \prod_{m=1}^{d} W_m$. Let $\mathbf{i} = (i_1, \ldots, i_d) \in \mathcal{G}$ denote the spatial coordinate of a cell. At any time step t, each cell takes a state $x_{t,\mathbf{i}} \in \mathcal{S}$, where $\mathcal{S} = \{0, 1, \ldots, N-1\}$ is a discrete state space of cardinality $N = |\mathcal{S}|$. For a comprehensive reference of all spatial grid and architectural notation used throughout this paper, please see Appendix A.

Definition 3.1 (Local dynamical rule)**.** A *local dynamical rule* is a deterministic, time-invariant map $f : \mathcal{S}^k \to \mathcal{S}$ governing the temporal evolution of the grid via $x_{t+1,\mathbf{i}} = f(\mathcal{N}_{t,\mathbf{i}})$ where $\mathcal{N}_{t,\mathbf{i}}$ denotes the *neighborhood configuration* of cell $\mathbf{i}$ at time t. This neighborhood is formally defined by k distinct relative spatial offset vectors $\mathcal{N} = (\boldsymbol{\delta}_1, \ldots, \boldsymbol{\delta}_k)$, yielding:

$$\mathcal{N}_{t,\mathbf{i}} = (x_{t,\mathbf{i}+\boldsymbol{\delta}_1}, \ldots, x_{t,\mathbf{i}+\boldsymbol{\delta}_k}) \in \mathcal{S}^k \tag{1}$$

This framework captures standard grid-based topologies. For the classical one dimensional elementary cellular automata (ECA) setting, $d = 1$, $V = 2$, and $k = 3$ with offsets $\{-1, 0, 1\}$, learning to the 256 Wolfram rules [36]. Conway's Game of Life [**?**] is a setting with $d = 2$, $V = 2$, and $k = 8$ with offsets .

This formulation yields V^k possible neighborhood configurations and a hypothesis space of $V^{(V^k)}$ distinct rules. For simplicity in our analysis, we frequently consider the isotropic case where all dimensions share a uniform width $W_m = W$, yielding a total grid volume $L = W^d$.

3.2 Sequence tokenization and the ICL objective

To evaluate inductive, out-of-sample generalization, that is, whether the model performs genuine in-context rule inference rather than memorization, our goal is to train a causal transformer to predict the grid's evolution under an unknown rule f. Each trajectory of T time steps is flattened in row-major order into a 1D sequence, delimited exclusively by temporal separator tokens `[SEP]`:

$$\underbrace{x_{0,0},\ \ldots,\ x_{0,L-1}}_{\text{step } 0},\ \texttt{[SEP]},\ \underbrace{x_{1,0},\ \ldots,\ x_{1,L-1}}_{\text{step } 1},\ \texttt{[SEP]},\ \ldots\ \texttt{[SEP]},\ \underbrace{x_{T-1,0},\ \ldots,\ x_{T-1,L-1}}_{\text{step } T-1} \tag{2}$$

Under standard causal masking, a context token (t', i') is visible from a query token (t, i) only if it precedes it in the flattened sequence ($t' < t$ or $t' = t$ and $i' \leq i$). This row-major flattening causes adjacent physical neighbors to become drastically separated in sequence space (by distances of $\mathcal{O}(W^{m-1})$) . Because intra-step tokens carry no predictive information, the model must learn to route attention to the scattered spatial neighbors at the preceding time step $t - 1$.

Given a context window large enough to contain all V^k possible neighborhood configurations, it is possible to achieve 100% predictive accuracy. In Appendix B, we formalize this by defining an optimal lookup-table learning algorithm.

The spatial addressing challenge. Our goal is to answer the question: can Transformers trained only on next token prediction recover the optimal learning algorithm? By flattening the grid into a one dimensional sequence, we break the spatial adjacency of cells. To predict $x_{t+1,i}$, the model must

locate and route the k neighbor states from the preceding time step, which may be scattered across the sequence, assemble the neighborhood configuration, and perform content-based retrieval across the context to deduce f. A naive approach using Relative Positional Encodings (RPE) requires $O(T \cdot L)$ learned bias parameters and directly encodes the grid topology into the architecture (formalized in Appendix C.2). A central question, which we address in Section 4, is whether spatial addressing can be achieved with a positional embedding dimension independent of the grid volume.

4 Mechanistic constructions for spatial routing

Induction heads [27] are two-layer attention circuits: the first layer identifies relevant historical context, and the second layer retrieves the associated completion. Existing constructions apply to one dimensional sequences; in-context learning of multidimensional local dynamical systems demands a substantially more complex routing mechanism, as the first layer must now gather a k-cell neighborhood at arbitrary spatial offsets. We focus on architectures of depth two or greater, since 1-layer transformers cannot perform exact spatial ICL in general (Appendix C.1).

We introduce spatial induction heads, a specialized two-layer circuit that solves the in-context learning task for spatial dynamics. The architecture decomposes into:

Layer 1 (spatial addressing): Instead of attending to directly preceding tokens, Layer 1 attends to the k-cell spatial neighborhood at the preceding time step, encoding the local configuration into the residual stream.
Layer 2 (pattern matching): Layer 2 identifies historical context positions sharing that exact decoded neighborhood configuration and copies the corresponding deterministic output value.

Below, we provide constructive proofs demonstrating how transformers can mathematically realize this circuitry. While our constructions use specialized spatial-aware embeddings, our empirical experiments demonstrate that standard transformers with learned positional embeddings converge to these optimal mechanisms via gradient descent.

Architectural notation. We use a residual stream of dimension D partitioned into functional sub-blocks denoted B_n . Let Π_{B_n} denote the orthogonal projection matrix onto that subspace. We use superscripts $(l) \in \{1, 2\}$ to denote the transformer layer index. For example, $W_Q^{(1)}, W_K^{(1)}$, and $W_V^{(1)}$ denote the query, key, and value projection matrices of Layer 1. Subscripts denote the sequence position, such that $h_{t,\mathbf{i}}^{(l)}$ represents the hidden representation of the cell at spatial coordinate $\mathbf{i}$ and time t after processing by layer l. The Layer 1 feed-forward network is a 2-layer ReLU MLP of hidden dimension $4hV^m$

Let $\mathcal{N} = (\boldsymbol{\delta}_1, \ldots, \boldsymbol{\delta}_k)$ be the tuple of k relative spatial offsets defining the local dynamical rule. Let $\hat{e}$ denote the unit spatial shift corresponding to a $+1$ sequence advancement in row-major order. To correctly align the autoregressive generation objective while permitting rigorous set operations, we define the Key neighborhood set as $\mathcal{N}_K = \{\boldsymbol{\delta}_1, \ldots, \boldsymbol{\delta}_k\}$ and the Query neighborhood set as $\mathcal{N}_Q = \{\boldsymbol{\delta} + \hat{e} \mid \boldsymbol{\delta} \in \mathcal{N}_K\}$. When evaluated at a specific sequence token at time t and spatial coordinate $\mathbf{i}$, we denote the instantiated neighborhood state configurations as $\mathcal{N}_{Q,t,\mathbf{i}}$ and $\mathcal{N}_{K,t,\mathbf{i}}$ respectively. This distinction is necessary due to the causal language modeling objective: the token at sequence position $(t, \mathbf{i})$ is tasked with predicting the state of the *next* cell, $(t, \mathbf{i} + \hat{e})$. To execute this prediction, the query token must evaluate the dynamical rule f by aggregating the k parent states centered at $\mathbf{i} + \hat{e}$ from the preceding time step $t - 1$. Therefore, $\mathcal{N}_{Q,t,\mathbf{i}}$ requires a $+\hat{e}$ relative shift so that its encoded pattern perfectly matches the unshifted historical key neighborhoods $\mathcal{N}_{K,t',\mathbf{i}'}$ of previously completed updates (a property used during Layer 2 Hamming matching in Appendix F).

We define the **union routing set** as $\mathcal{U} = \mathcal{N}_K \cup \mathcal{N}_Q$. By the principle of inclusion-exclusion, its cardinality is $|\mathcal{U}| = 2k - |\mathcal{N}_K \cap \mathcal{N}_Q|$, which is bounded by $k + 1 \leq |\mathcal{U}| \leq 2k$ depending on the topological overlap. For instance, in a 1D Elementary Cellular Automaton with $k = 3$ offsets $\mathcal{N} = \{-1, 0, 1\}$, the unit shift is $\hat{e} = 1$, yielding $\mathcal{N}_K = \{-1, 0, 1\}$ and $\mathcal{N}_Q = \{0, 1, 2\}$. Their union $\mathcal{U} = \{-1, 0, 1, 2\}$ has cardinality $4 < 2k$, demonstrating how sequential flattening induces overlap.

Definition 4.1 (Spatial partition)**.** Given $h \in \{1, \ldots, |\mathcal{U}|\}$ Layer 1 attention heads, a spatial partition is a collection of mutually exclusive subsets $S_1, \ldots, S_h \subseteq \mathcal{U}$ satisfying $\bigcup_{\gamma=1}^{h} S_\gamma = \mathcal{U}$ and $|S_\gamma| \leq m := \lceil |\mathcal{U}|/h \rceil$ for all $\gamma \in \{1, \ldots, h\}$. Each head γ is assigned the offsets in S_γ, and we refer to m as the *partition size*.

The two endpoints recover the previously discussed regimes: $h = |\mathcal{U}|$ yields $m = 1$ (each head routes one cell in the union set independently), and $h = 1$ yields $m = |\mathcal{U}|$ (a single attention head is bottlenecked to route the entire union neighborhood simultaneously). notation suggestion: $\mathcal{N}$ replaced with $\mathcal{D}$ (see above), $\mathcal{N}_k$ not needed, then $\mathcal{D}_+$ defined instead of N_Q.

4.1 Constructing spatial-aware embeddings

As detailed in Appendix C.2, a naive transformer approach to multidimensional grids relies on Relative Positional Encodings (RPE) to act as deterministic shift operators. Because flattening a multidimensional grid maps adjacent spatial neighbors to sequence distances that scale with the grid axes (e.g., $\mathcal{O}(W^{d-1})$), this approach requires a learned bias table of $\mathcal{O}(T \cdot L)$ entries and directly encodes the grid topology into the architecture.

Instead, we decouple spatial addressing from sequence distance by introducing *spatial-aware embeddings*. We demonstrate that spatial routing can be achieved with a representation dimension independent of the grid volume W^d. We define these embeddings and prove they can be constructed via multi-peaked trigonometric polynomials in Appendix D (Lemma D.2); these polynomials allow an attention head to isolate multiple spatial offsets simultaneously without relying on sequence-dependent shift operators.

Formally, encoding m target spatial offsets requires the attention inner product to act as a multi-peaked function: it must yield mutually distinct prescribed scores $\{s_1, \ldots, s_m\}$ exactly at the target spatial offsets, while assigning a suppressed score $\leq s_{\min} - \Delta$ to all other background positions (see Definition D.1). We summarize the resulting dimensional bounds below:

Theorem 4.2 (Bounds for spatial addressing)**.** *For any d-dimensional discrete spatial grid of size W^d, temporal horizon T, margin $\Delta > 0$, and a target subset of m spatial offsets assigned to an attention head, a positional embedding $\mathbf{p}_{(t,\mathbf{i})} \in \mathbb{R}^{d_p}$ encoding these targets satisfies the following dimensional bounds:*

- ***Independent per-head addressing*** *($m = 1$): When an attention head is responsible for isolating a single spatial offset, there exists an explicit embedding requiring a total dimension of $d_p = 2d + 2$.*
- ***Simultaneous addressing*** *($m > 1$): When an attention head is burdened with simultaneously routing a neighborhood of m offsets, the positional attention score is parameterized as a multi-peaked trigonometric polynomial. There exists a valid spatial-aware embedding requiring a total dimension of $d_p \leq 2(m + d) + 2$.*

The formal proof and existence of these multi-peaked polynomials are deferred to Appendix D. In Lemma D.4 we present an explicit construction for one-dimensional cellular automata.

Theorem 4.2 ensures that it is possible to construct query and key matrices so that positions outside of the m targets have very low scores, controlled by an arbitrary margin Δ. By setting Δ sufficiently large, the attention weight on these extraneous positions is tightly bounded. Concretely, each routing head aggregates the states of the m cells in its assigned partition S_γ into a scalar superposition $z_\gamma = \sum_{j \in S_\gamma} \alpha_j w_{x_{t,j}} + \eta_\gamma$, where α_j are softmax weights concentrated on the targets and η_γ is bounded leakage from non-target positions.

The acceptable threshold for this leakage η_γ depends on how the architecture processes the superposition. If the architecture uses non-linear decoding (Section 4.2), the leakage must be bounded by the tolerance of the MLP. If the architecture routes features to content-based retrieval without an MLP (Section E.2), the leakage must be bounded to preserve downstream inner-product separability.

4.2 Non-linear decoding

Following spatial routing, each head γ outputs a scalar superposition $z_\gamma = \sum_{j \in S_\gamma} \alpha_j w_{x_{t,j}} + \eta_\gamma$, where α_j are softmax weights concentrated on the m target cells and η_γ is bounded leakage. To perform content-based retrieval in Layer 2, this scalar must be decoded into a mutually orthogonal pattern encoding in block B_3. Lemma E.1 guarantees that almost any softmax weight vector α makes z_γ injective over V^m configurations, yielding V^m distinct values with minimum gap $\delta > 0$.

A linear decoder cannot complete this step. By Lemma E.2, no affine map $T : \mathbb{R} \to \mathbb{R}^n$ can send n distinct scalar inputs to n mutually orthogonal targets when $n \geq 3$, which holds here whenever $m \geq 2$ or $V \geq 3$. A non-linear activation is therefore necessary. Definition E.3 and Lemma E.4 construct

an explicit 2-layer ReLU MLP with hidden width $4V^m$ per head that achieves exact decoding via trapezoid indicator functions. To function correctly, this MLP requires the routing leakage to satisfy $\eta_\gamma < \delta/4$, allowing it to absorb the upstream noise and snap the scalar into a perfect orthogonal representation. In the fully distributed regime ($m = 1$), each head routes a single cell and this step is unnecessary: the linear value projection suffices.

4.3 Content-based retrieval

With exact orthogonal pattern encodings written to $B_{3,K}$ and $B_{3,Q}$, a single Layer 2 attention head performs integer Hamming matching. The mechanics of this matching depend on the routing regime:

In the partitioned regime ($m > 1$), the Layer 1 MLP has provided exact orthogonal encodings. The pre-softmax score between a query token (t, i) and a context token (t', i') equals ch if and only if their h-block pattern encodings agree on all heads, and is at most $c(h-1)$ otherwise.

In the fully distributed regime ($m = 1$), spatial features are isolated and bypass the Layer 1 MLP. Layer 2 must therefore perform Hamming matching directly on noisy one-hot vectors. To guarantee that matching and non-matching neighborhood configurations remain separable under inner product despite this noise, the per-head attention leakage from Layer 1 must be bounded by $\bar{\epsilon} < \frac{1}{4k}$ (formalized as Lemma E.6 in Appendix E).

In both regimes, the score gap between matching and non-matching patterns drives the retrieval leakage $\epsilon_{ret} \to 0$ as the query scaling $c \to \infty$ (Lemma E.9). Because the rule f is deterministic, every context token whose key pattern matches the query carries the identical value vector $e(f(\mathcal{N}_{t,i}))$; Corollary E.10 then guarantees exact cell-state prediction whenever $\epsilon_{ret} < \sqrt{2}/4$, which is achieved by a finite threshold on c. No MLP is required in Layer 2.

The architecture described in Section 4 and Appendices D–E.1 is parameterized by $h \in \{1, \ldots, |\mathcal{U}|\}$ spatial routing heads in Layer 1. As the complexity of the local dynamical system is conserved, it must be absorbed either by the routing mechanism (via multi-head attention) or by the decoding mechanism (via the MLP). The two endpoints recover the regimes mentioned above: at $h = 1$, a single attention head compresses the entire union neighborhood into a 1-dimensional superposition and a wide MLP is required to decode it; at $h = |\mathcal{U}|$, spatial features are isolated across heads and no MLP is needed. The following theorem formalizes the continuous tradeoff between these endpoints.

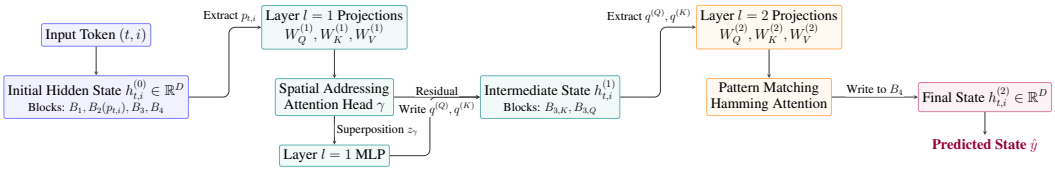

Figure 1: Flow of the spatial induction head architecture. Layer 1 routes position-aware embeddings ($p_{t,i}$) to isolate spatial neighborhoods, using an MLP to decode the scalar superpositions (z_γ) into discrete one-hot pattern encodings ($q^{(Q)}, q^{(K)}$). Layer 2 subsequently performs content-based Hamming matching on these marginalized representations to retrieve the predicted target state ($\hat{y}$).

Theorem 4.3 (Spatial in-context learning)**.** *For any deterministic local dynamical system defined over a d-dimensional discrete grid with neighborhood size k, union routing set $\mathcal{U}$, state space cardinality $V \geq 2$, and context window of $N_{\text{seq}} > k$ tokens, let $\varepsilon > 0$ be an error tolerance. There exists a 2-layer causal transformer implementing the optimal learning algorithm that predicts the system's evolution with error $\leq \varepsilon$. By parameterizing the architecture by $h \in \{1, \ldots, |\mathcal{U}|\}$ spatial routing heads, the required representation dimension D (residual stream width) and Layer 1 MLP hidden dimension D_{MLP} are bounded by:*

$$D = \mathcal{O}\left(hV^{\lceil |\mathcal{U}|/h \rceil} + d\right), \quad D_{MLP} = \mathcal{O}\left(hV^{\lceil |\mathcal{U}|/h \rceil}\right) \tag{3}$$

These complexities are independent of the macroscopic grid volume $L = W^d$.

The proof, provided in Appendix F, assembles the three modular stages from Appendices: spatial routing via spatial-aware embeddings, non-linear decoding into orthogonal pattern encodings, and Hamming-matched retrieval.

The three terms of the bound decompose along functional lines: $hV^{\lceil |\mathcal{U}|/h \rceil}$ is the decoding cost paid by the Layer 1 MLP scratch space, $|\mathcal{U}|$ is the routing cost paid by the positional embedding, and d is the irreducible spatial overhead. Two further clarifications help parse the result. First, by inclusion-exclusion, $|\mathcal{U}| = |\mathcal{N}_K \cup \mathcal{N}_Q| \leq 2k$, with equality only when the key and query neighborhoods are fully disjoint. Second, D denotes the total residual stream width, comprising the token embedding ($\dim B_1 = V$), the positional embedding ($\dim B_2 = d_p$), the neighborhood encoding scratch block ($\dim B_3 = 2hV^m$ with $m = \lceil |\mathcal{U}|/h \rceil$), and the output block ($\dim B_4 = V$); the Layer 1 MLP hidden dimension is a separate quantity bounded by $4hV^m$.

A key consequence of Theorem 4.3 is that the required architectural capacity depends only on the local dynamical parameters (k, V, d) and is independent of the grid volume $L = W^d$. This follows because the token embedding, positional embedding, and scratch blocks scale exclusively with local parameters, no component scales with sequence length or grid dimensions. This independence does not eliminate computational complexity: the representation cost is conserved and localized in model capacity, absorbed either by attention routing (requiring $h = |\mathcal{U}|$ parallel heads) or by non-linear decoding (requiring MLP width $\mathcal{O}(V^{|\mathcal{U}|})$). The only quantity that scales with grid size is the context length itself: the unrolled trajectory has length $T \cdot L$ tokens, which is unavoidable for any architecture processing the full spatiotemporal sequence.

By parameterizing over h, we expose a fundamental **routing-decoding tradeoff** governing the architecture. When Layer 1 possesses sufficient attention heads to route each of the $|\mathcal{U}|$ cells independently ($h = |\mathcal{U}|$), spatial features are isolated and no MLP is required. Conversely, when a single attention head must compress all $|\mathcal{U}|$ cells into a 1-dimensional superposition ($h = 1$), an MLP becomes essential to decode the representation. At the bottlenecked extreme ($h = 1$), the representation and MLP dimension scale as $\mathcal{O}(V^{|\mathcal{U}|})$, while at the parallel extreme ($h = |\mathcal{U}|$), the dimension is bounded by $\mathcal{O}(|\mathcal{U}|V + d)$. The intermediate regime is captured directly by the unified formula in Theorem 4.3.

5 Gradient descent learns spatial induction heads

We now empirically show that Transformers trained on next token prediction converge to spatial induction heads.

5.1 Experimental setup

We study in-context learning of cellular automata across three settings: one dimensional elementary cellular automata (V=2, k=3, L=16), two dimensional cellular automata with Von Neumann neighborhoods (V=2, k=5, 6×6 grid), and one dimensional cellular automata with V=3 (k=3, L=32). We train standard decoder-only Transformers with causal attention and learned absolute positional embeddings, without the specialized spatial-aware embeddings from our theoretical constructions or explicit Relative Positional Encodings, so that we can test whether gradient descent independently discovers the predicted spatial routing mechanisms. Training and test sets use entirely disjoint rule equivalence classes, ensuring the model cannot succeed by memorization. In all settings, context length M is chosen so that all V^k neighborhood configurations appear in every context window (Appendix B.2). Full details on data generation, hyperparameters, and training schedules are in Appendix G.

We report cell accuracy, sequence accuracy, and autoregressive accuracy (sequence accuracy when feeding predictions back as input) . Sequence accuracy is our primary metric: it requires every cell to be predicted correctly across all time steps, whereas the cell accuracy reported by prior work can appear deceptively high even when many sequences contain errors. In comparison, Berkovich et al. [2] provides the rule matrix explicitly in the prompt and evaluates single-step prediction, while Burtsev [7] train larger models (4 layers, 8 heads, D=512) on 1D ECA, with performance degrading on multi-step tasks.

5.2 Results

Table 1 summarizes our results. On 1D ECA, 2-layer models achieve perfect accuracy including autoregressive generation, confirming that stochastic gradient descent can learn to perform spatial ICL on this task. The perfect autoregressive accuracy indicates that the model has internalized the

Table 1: Results across all settings. For 2-layer models, a+b denotes a heads in Layer 1 and b in Layer 2. For deeper models, $a \times n$ denotes a heads per layer across n layers. Auto Acc is measured over 4 autoregressive time steps from context.

Setting	Model	Cell Acc	Seq Acc	Auto Acc
1D, V=2, k=3 (ECA)	2-layer, 1+1 heads, D=512	100.0	100.0	100.0
	2-layer, 3+1 heads, D=384	100.0	100.0	100.0
1D, V=3, k=3	4-layer, 3×4 heads, D=576	99.9	89.8	86.8
	2-layer, 3+1 heads, D=384	99.8	87.5	85.0
2D , k=5	4-layer, 5×4 heads, D=640	99.9	85.7	1.0
	2-layer, 5+1 heads, D=640	97.9	29.0	12.0
2D VN + RPE, k=5	2-layer, 5+1 heads, D=640	99.5	70.5	1.0

exact rule rather than approximating it, as any single-cell error would compound across subsequent steps. The mechanism extends to V=3, where the configuration space grows from 8 to 27, with the 2-layer model still reaching 87.5% sequence accuracy, providing evidence that the construction can accommodate larger state spaces.

On 2D, 4-layer models reach 85.7% sequence accuracy, with training exhibiting a grokking-like phase transition between epochs 200 and 260. The 2-layer model lags behind, but injecting 2D relative positional biases raises it to 70.5%, confirming that the primary bottleneck is learning spatial addressing from flattened sequences rather than model capacity. Theorem 4.2 proves a 2-layer model can theoretically solve it, the mathematical construction relies on ultra-sharp attention distributions and specific saturation points in ReLUs. However, Standard SGD/Adam struggles to find these brittle configurations from random initialization. The extra layers in the 4-layer model provide a smoother optimization landscape, allowing the network to learn the logic compositionally rather than forcing it into two drastic, highly constrained steps.

5.3 Mechanistic validation

The results above confirm that our models can perform spatial ICL, but do not reveal how. We now analyze the internal mechanisms of the trained models to test whether they match the spatial induction head circuit predicted by our theory.

Attention distribution

Figure 2 visualizes the attention of the ECA 1+1 heads model on a representative example. Averaged over the full test set (20k samples), Layer 1 concentrates 68.4% of attention on the neighborhood cells in the preceding row, confirming spatial addressing. Layer 2 concentrates 97.1% on context positions sharing the same neighborhood configuration, confirming content-based retrieval.

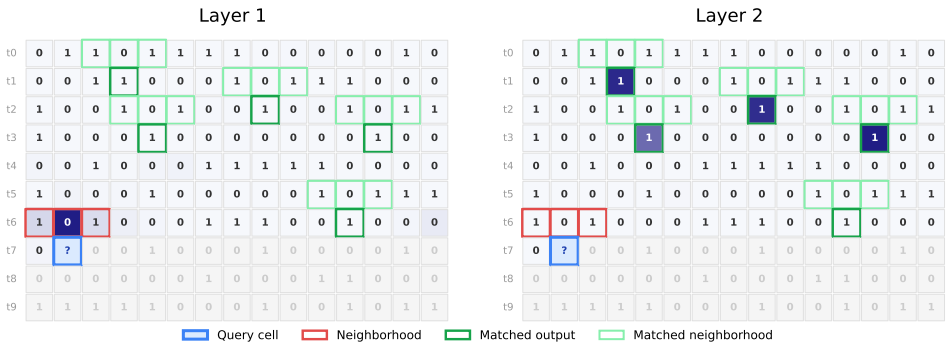

Figure 2: Attention over the ECA grid for a single prediction cell (1+1 heads model, Rule 97). The query cell at (t_7, c_1) predicts $f(1, 0, 1) = 1$. Layer 1 (left) concentrates on the three neighborhood cells in the preceding row. Layer 2 (right) concentrates on context positions sharing the same configuration $(1, 0, 1)$, retrieving the corresponding output.

Representation structure Figure 3 shows t-SNE visualizations at successive stages of the ECA 1+1 heads model. Before the Layer 1 MLP, representations for different configurations are intermixed. After the MLP, clusters separate by configuration. After Layer 2 attention, they further split by output value, yielding 16 clusters for the 8×2 (configuration, output) pairs. This confirms that Layer 1 decodes the neighborhood and Layer 2 retrieves the correct output.

Routing-decoding tradeoff

Table 2 validates the routing-decoding tradeoff. Removing the MLP from the ECA 1-head model drops accuracy to 57.3%, as the single head cannot linearly disentangle the spatial superposition. With 3 heads, removing the MLP has negligible effect, consistent with the prediction that when heads match the neighborhood size, spatial features are isolated and no MLP is required.

Table 2: Effect of removing the Layer 1 MLP (1D ECA). All models use 1 head in Layer 2.

L1 Heads	With MLP	Without MLP
1	100.0	57.3
2	100.0	56.7
3	100.0	97.6
4	100.0	90.8

Width independence Theorem 4.3 predicts that the required model dimension is independent of grid width. We verify this by training the ECA 1+1 and 3+1 heads models on $L \in \{10, 16, 32\}$ separately. All achieve sequence accuracy above 0.99, confirming that the mechanism is purely local.

6 Conclusion and future work

We introduced spatial induction heads, a two-layer attention circuit that solves in-context learning of multidimensional cellular automata by decoupling spatial neighborhood routing from content-based rule retrieval. Constructively, the required representation dimension is independent of the grid volume, and a fundamental routing-decoding tradeoff governs the interplay between Layer 1 attention heads and MLP width. Empirically, standard transformers trained with next-token prediction converge to this exact mechanism.

Our results suggest the inductive bias toward induction-like circuits extends beyond 1D sequences: the same primitives that underlie language ICL suffice to discover spatial routing from flattened data, with no explicit encoding of grid topology. This connects the induction-head literature for language models with patch-level circuit analyses of vision transformers, and offers a concrete circuit-level prediction to seek in transformers trained on real spatiotemporal data. Extensions include generalizing this framework to stochastic or continuous-state dynamics, which would expand Layer 2 retrieval from exact copying to distributional aggregation. We hope this synthetic foundation provides the precise circuit-level blueprints needed to hunt for spatial induction heads in large-scale, pre-trained spatiotemporal models.

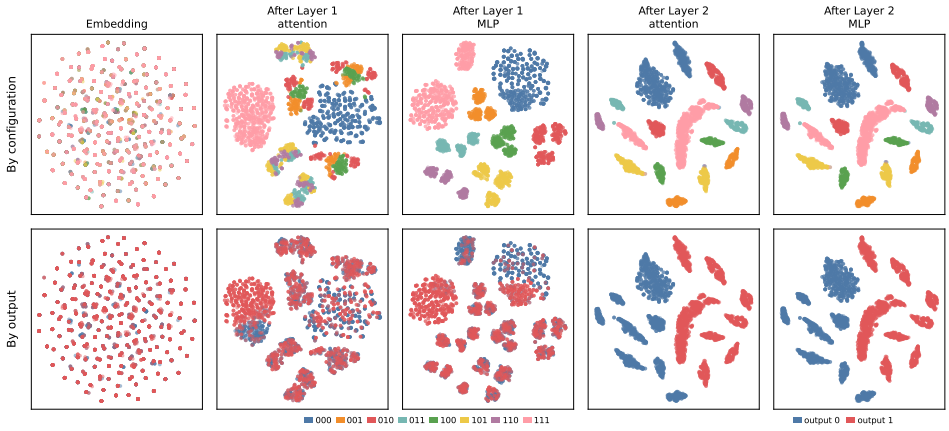

Figure 3: t-SNE of hidden representations at successive stages of the ECA 1+1 heads model (500 test samples). **Top:** colored by neighborhood configuration. After the Layer 1 MLP, representations cluster by configuration. **Bottom:** colored by output value. After Layer 2 attention, each cluster splits by output, yielding 16 clusters for 8 configurations $\times$ 2 outputs. Layer 2 MLP is not necessary.

References

[1] Kwangjun Ahn, Xiang Cheng, Hadi Daneshmand, and Suvrit Sra. Transformers learn to implement preconditioned gradient descent for in-context learning. *Advances in Neural Information Processing Systems*, 36:45614–45650, 2023.

[2] Jaime A Berkovich, Noah S David, and Markus J Buehler. Automatagpt: Forecasting and ruleset inference for two-dimensional cellular automata. *arXiv preprint arXiv:2506.17333*, 2025.

[3] Kaifeng Bi, Lingxi Xie, Hengheng Zhang, Xin Chen, Xiaotao Gu, and Qi Tian. Accurate medium-range global weather forecasting with 3d neural networks. *Nature*, 619(7970):533–538, 2023.

[4] Alberto Bietti, Vivien Cabannes, Diane Bouchacourt, Herve Jegou, and Leon Bottou. Birth of a transformer: A memory viewpoint. *Advances in Neural Information Processing Systems*, 36: 1560–1588, 2023.

[5] Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, Clarence Ng, Ricky Wang, and Aditya Ramesh. Video generation models as world simulators. 2024. URL `https://openai.com/research/video-generation-models-as-world-simulators`.

[6] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[7] Mikhail Burtsev. Learning elementary cellular automata with transformers. *arXiv preprint arXiv:2412.01417*, 2024.

[8] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021.

[9] Frank Cole, Yulong Lu, Tianhao Zhang, and Yuxuan Zhao. In-context learning of linear dynamical systems with transformers: Error bounds and depth-separation. *arXiv e-prints*, pages arXiv–2502, 2025.

[10] Arthur Conmy, Augustine Mavor-Parker, Aengus Lynch, Stefan Heimersheim, and Adrià Garriga-Alonso. Towards automated circuit discovery for mechanistic interpretability. *Advances in Neural Information Processing Systems*, 36:16318–16352, 2023.

[11] Yijia Dai, Zhaolin Gao, Yahya Sattar, Sarah Dean, and Jennifer J Sun. Pre-trained large language models learn to predict hidden markov models in-context. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

[12] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[13] Ezra Edelman, Nikolaos Tsilivis, Benjamin L Edelman, Eran Malach, and Surbhi Goel. The evolution of statistical induction heads: In-context learning markov chains. *Advances in neural information processing systems*, 37:64273–64311, 2024.

[14] Chanakya Ekbote, Marco Bondaschi, Nived Rajaraman, Jason D Lee, Michael Gastpar, Ashok Vardhan Makkuva, and Paul Pu Liang. What one cannot, two can: Two-layer transformers provably represent induction heads on any-order markov chains. *arXiv preprint arXiv:2508.07208*, 2025.

[15] Ronen Eldan and Ohad Shamir. The power of depth for feedforward neural networks. *arXiv*, 2015. doi: 10.48550/arxiv.1512.03965.

[16] Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, et al. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 1(1):12, 2021.

[17] Shivam Garg, Dimitris Tsipras, Percy S Liang, and Gregory Valiant. What can transformers learn in-context? a case study of simple function classes. *Advances in neural information processing systems*, 35:30583–30598, 2022.

[18] Angeliki Giannou, Liu Yang, Tianhao Wang, Dimitris Papailiopoulos, and Jason D Lee. How well can transformers emulate in-context newton's method? *arXiv preprint arXiv:2403.03183*, 2024.

[19] Samy Jelassi, David Brandfonbrener, Sham M Kakade, and Eran Malach. Repeat after me: Transformers are better than state space models at copying. *arXiv preprint arXiv:2402.01032*, 2024.

[20] Chris G Langton. Computation at the edge of chaos: Phase transitions and emergent computation. *Physica D: nonlinear phenomena*, 42(1-3):12–37, 1990.

[21] Bingbin Liu, Jordan T Ash, Surbhi Goel, Akshay Krishnamurthy, and Cyril Zhang. Transformers learn shortcuts to automata. *arXiv preprint arXiv:2210.10749*, 2022.

[22] Toni JB Liu, Nicolas Boullé, Raphaël Sarfati, and Christopher Earls. Llms learn governing principles of dynamical systems, revealing an in-context neural scaling law. In *Proceedings of the 2024 conference on empirical methods in natural language processing*, pages 15097–15117, 2024.

[23] Ashok V Makkuva, Marco Bondaschi, Chanakya Ekbote, Adway Girish, Alliot Nagle, Hyeji Kim, and Michael Gastpar. Local to global: Learning dynamics and effect of initialization for transformers. *Advances in Neural Information Processing Systems*, 37:86243–86308, 2024.

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

[25] Guido Montúfar, Razvan Pascanu, Kyunghyun Cho, and Yoshua Bengio. On the number of linear regions of deep neural networks. *arXiv*, 2014. doi: 10.48550/arxiv.1402.1869.

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

[27] Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, et al. In-context learning and induction heads. *arXiv preprint arXiv:2209.11895*, 2022.

[28] Nived Rajaraman, Marco Bondaschi, Kannan Ramchandran, Michael Gastpar, and Ashok V Makkuva. Transformers on markov data: Constant depth suffices. *Advances in Neural Information Processing Systems*, 37:137521–137556, 2024.

[29] Clayton Sanford, Daniel Hsu, and Matus Telgarsky. One-layer transformers fail to solve the induction heads task. *arXiv preprint arXiv:2408.14332*, 2024.

[30] Matus Telgarsky. Benefits of depth in neural networks. *arXiv*, 2016. doi: 10.48550/arxiv.1602.04485.

[31] Keyon Vafa, Justin Y Chen, Ashesh Rambachan, Jon Kleinberg, and Sendhil Mullainathan. Evaluating the world model implicit in a generative model. *Advances in Neural Information Processing Systems*, 37:26941–26975, 2024.

[32] Martina G Vilas, Timothy Schaumlöffel, and Gemma Roig. Analyzing vision transformers for image classification in class embedding space. *Advances in neural information processing systems*, 36:40030–40041, 2023.

[33] Johannes Von Oswald, Eyvind Niklasson, Ettore Randazzo, João Sacramento, Alexander Mordvintsev, Andrey Zhmoginov, and Max Vladymyrov. Transformers learn in-context by gradient descent. In *International Conference on Machine Learning*, pages 35151–35174. PMLR, 2023.

[34] Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*, 2022.

[35] Stephen Wolfram. Statistical mechanics of cellular automata. *Reviews of modern physics*, 55 (3):601, 1983.

[36] Stephen Wolfram and M Gad-el Hak. A new kind of science. *Appl. Mech. Rev.*, 56(2):B18–B19, 2003.

[37] David H Wolpert. The lack of a priori distinctions between learning algorithms. *Neural computation*, 8(7):1341–1390, 1996.

[38] Weiyan Xie, Xiao-Hui Li, Caleb Chen Cao, and Nevin L Zhang. Vit-cx: Causal explanation of vision transformers. *arXiv preprint arXiv:2211.03064*, 2022.

[39] Ruiqi Zhang, Spencer Frei, and Peter L Bartlett. Trained transformers learn linear models in-context. *Journal of Machine Learning Research*, 25(49):1–55, 2024.

[40] Shiyang Zhang, Aakash Patel, Syed A Rizvi, Nianchen Liu, Sizhuang He, Amin Karbasi, Emanuele Zappala, and David van Dijk. Intelligence at the edge of chaos. *arXiv preprint arXiv:2410.02536*, 2024.

A Notation reference

Table 3: Notation for the synthetic data environment and spatial grid parameters.

Variable	Description
d	Spatial dimension
L	Total grid volume
W	Uniform grid width
V	Discrete state space
k	Cells in base neighborhood

Table 4: Notation for the model architecture and spatial routing components.

Variable	Description
$\mathcal{N}_K$	Key neighborhood
$\mathcal{N}_Q$	Query neighborhood
$\mathcal{U}$	Union routing set
h	Spatial routing heads
m	Spatial partition size

Table 5: Notation for transformer architecture and intermediate representations.

Variable	Description
(l)	Superscript denoting the causal transformer layer index, $l \in \{1, 2\}$.
$W_Q^{(l)}, W_K^{(l)}, W_V^{(l)}$	Query, Key, and Value linear projection matrices for Layer l.
$h_{t,\mathbf{i}}^{(l)}$	Final hidden residual stream representation of token $(t, \mathbf{i})$ after Layer l.
D	Total dimensional width of the hidden representation.
B_n	Isolated functional sub-block of the residual stream.
$B_{3,K}, B_{3,Q}$	Partitioned sub-blocks of B_3 storing marginalized Key and Query encodings.
Π_{B_n}	Orthogonal projection matrix extracting the subspace B_n.
$p_{t,\mathbf{i}}$	spatial-aware embedding vector of dimension d_p.
z_γ	Scalar superposition routed by Layer 1 attention head γ.
$q_{t,\mathbf{i}}^{(Q)}, q_{t',\mathbf{i}'}^{(K)}$	Concatenated one-hot pattern encodings for Layer 2 Hamming matching.

B Optimal learning and configuration coverage

B.1 Optimal learning algorithm

In the deterministic local dynamical system setting, we define a simple learning algorithm and prove its optimality. Algorithm 1 maintains an empirical lookup table $\hat{f}$, initially empty, and updates it as new (configuration, output) pairs are observed.

Algorithm 1 Lookup Table Learning

Initialize $\hat{f} \leftarrow \emptyset$
for all $t \in \{0, \ldots, T-2\}$, $\mathbf{i} \in \mathcal{G}$ **do**
  $\mathcal{N}_{t,\mathbf{i}} \leftarrow (x_{t,\mathbf{i}+\boldsymbol{\delta}_1}, \ldots, x_{t,\mathbf{i}+\boldsymbol{\delta}_k})$
  $y \leftarrow x_{t+1,\mathbf{i}}$
  Predict: $\hat{x}_{t+1,\mathbf{i}} \leftarrow \hat{f}(\mathcal{N}_{t,\mathbf{i}})$ if $\mathcal{N}_{t,\mathbf{i}} \in \text{keys}(\hat{f})$, else draw $v \sim \text{Uniform}(\mathcal{S})$
  Update: If $\mathcal{N}_{t,\mathbf{i}} \notin \text{keys}(\hat{f})$, set $\hat{f}(\mathcal{N}_{t,\mathbf{i}}) \leftarrow y$
end for

Because the rule f is deterministic, once a configuration $\mathbf{c} \in \mathcal{S}^k$ has been observed, the stored value $\hat{f}(\mathbf{c}) = f(\mathbf{c})$ is correct for all future occurrences. We now show that this algorithm is optimal under adversarial ground truth rules.

Theorem B.1. *The prediction rule of Algorithm 1 is minimax optimal: for any alternative prediction algorithm $\mathcal{A}$,*

$$\max_{f \in \mathcal{F}} \text{err}_f(\mathcal{A}) \;\geq\; \max_{f \in \mathcal{F}} \text{err}_f(\hat{f}),$$

where $\text{err}_f(\cdot)$ counts the number of incorrect cell predictions under ground truth rule f, and $\mathcal{F} = \{f : \mathcal{S}^k \to \mathcal{S}\}$ is the set of all deterministic local rules.

Proof. For any configuration $\mathbf{c} \in \text{keys}(\hat{f})$, the determinism of f guarantees that $\hat{f}(\mathbf{c}) = f(\mathbf{c})$, so the prediction is correct regardless of which $f \in \mathcal{F}$ generated the data. No algorithm can do better on these configurations.

For any configuration $\mathbf{c} \notin \text{keys}(\hat{f})$, the observed data imposes no constraint on the value of $f(\mathbf{c})$. Given any prediction $\hat{x}$ produced by any algorithm $\mathcal{A}$, an adversary can select a rule f that is consistent with all observations and additionally satisfies $f(\mathbf{c}) \neq \hat{x}$. Such a rule exists because $f(\mathbf{c})$ is unconstrained by the observations. This holds whether $\mathcal{A}$ is deterministic or randomized, since the adversary selects $f(\mathbf{c})$ after observing $\hat{x}$.

Therefore, errors on unseen configurations are unavoidable for any algorithm, and the lookup table already achieves zero error on all seen configurations. This is a direct instance of the No Free Lunch theorem [37] for deterministic function learning. □

Theorem B.1 implies that the error of Algorithm 1 is a lower bound on the error of any prediction algorithm, as no method can achieve fewer worst-case errors. If all V^k configurations appear in the context, zero prediction error is achievable. If some configurations are missing, errors on those configurations are unavoidable regardless of the algorithm. In our experiments (Section 5), we choose the context window to be large enough to cover all V^k configurations and verify after data generation that complete coverage is indeed achieved. This ensures that any prediction error reflects a limitation of the model, not insufficient context.

B.2 Configuration coverage

By Theorem B.1, exact in-context learning requires that all V^k neighborhood configurations appear in the context window. We provide a simple estimate of the number of context time steps needed under an idealized assumption.

Proposition B.2. *Suppose each neighborhood configuration at every space-time position is drawn independently and uniformly from $\mathcal{S}^k$. Then the minimum number of context time steps $M_{\min}$ required to observe all V^k configurations with probability at least P over a grid of volume L is:*

$$M_{\min} = \left\lceil \frac{\ln\left(\frac{1-P}{V^k}\right)}{L \ln\left(1 - \frac{1}{V^k}\right)} \right\rceil \tag{4}$$

Proof. Let E_j denote the event that configuration $\mathbf{c}_j$ is never observed across LM draws. Under the uniform and independent assumption, the probability of not drawing $\mathbf{c}_j$ in a single draw is $1 - V^{-k}$, so across LM independent draws:

$$P(E_j) = \left(1 - \frac{1}{V^k}\right)^{LM}$$

By the union bound:

$$P(\text{all covered}) = 1 - P\left(\bigcup_{j=1}^{V^k} E_j\right) \geq 1 - \sum_{j=1}^{V^k} P(E_j) = 1 - V^k \left(1 - \frac{1}{V^k}\right)^{LM}$$

Setting this to at least P and solving for M yields the stated bound. □

In practice, the independence assumption of Proposition B.2 does not hold. Even when the initial time step is sampled i.i.d. from $\mathcal{S}$, neighboring positions share cells, so their configurations are not independent. Moreover, all subsequent time steps are determined by the rule f rather than sampled randomly. This bound is therefore an approximate guide for choosing the context length during data generation. We verify after generating each sample that all V^k configurations appear in the context window, and discard any sample that does not achieve complete coverage.

C Baseline: Naive Transformer Construction

To demonstrate why our spatial-aware spatial induction heads are necessary, we detail a naive causal transformer construction that scales poorly with the neighborhood size k and state space size V when relying purely on standard Relative Position Encodings (RPE).

C.1 Why two layers are necessary

Both the naive baseline (Appendix C) and our spatial-aware construction (Appendix D) use two transformer layers. We give an informal argument that this is necessary, paralleling the depth-minimality of standard induction heads.

Each attention layer performs a single "hop" of information composition. Standard induction heads use two hops: Layer 1 copies information from a preceding token into the current position, and Layer 2 uses the enriched query to perform prefix matching and copy the matched output. Spatial induction heads inherit this two-hop structure but generalize the first hop: Layer 1 must gather a k-cell neighborhood at arbitrary spatial offsets, and Layer 2 performs pattern matching on the assembled configuration.

A 1-layer transformer struggles to accomplish both hops at once. Before attention, the query $q_{t,\mathbf{i}} = W_Q h^{(0)}_{t,\mathbf{i}}$ is a function only of the local token and its positional encoding, so it carries no information about the target neighborhood configuration $\mathcal{N}_{t,\mathbf{i}}$. While the neighborhood cells appear among the keys, the query lacks the pattern information needed to selectively attend to context positions sharing the same k-cell configuration. For the standard induction head task ($k = 1$), Sanford et al. [29] make this intuition precise via communication complexity, showing that any 1-layer transformer must grow in size linearly with the input length, whereas a 2-layer transformer suffices with size only logarithmic in the input length. We expect an analogous separation for spatial induction heads with $k \geq 2$ but leave a formal proof to future work.

C.2 A naive construction with relative positional encoding

We sketch a naive 2-layer transformer that solves spatial ICL using Relative Position Encoding (RPE) as rigid shift operators, providing a comparison point for our spatial-aware construction.

The total sequence length is $T \cdot L$. We set the hidden dimension to $D = (2k + 2)V$ and define the initial token embedding by concatenating the one-hot state vector with two zero scratch blocks of size kV each and an output block of size V:

$$h^{(0)}_{t,\mathbf{i}} = \big[\ \underbrace{e(x_{t,\mathbf{i}})}_{B_1 \in \mathbb{R}^V};\ \underbrace{\mathbf{0}_{kV}}_{B_2 \in \mathbb{R}^{kV}}\ ;\ \underbrace{\mathbf{0}_{kV}}_{B_3 \in \mathbb{R}^{kV}}\ ;\ \underbrace{\mathbf{0}_{V}}_{B_4 \in \mathbb{R}^{V}}\ \big] \in \mathbb{R}^D. \tag{5}$$

Here B_1 stores the cell's own one-hot state, B_2 and B_3 are scratch blocks for assembling the neighborhood configurations at spatial positions $\mathbf{i}$ and $\mathbf{i} + \hat{\mathbf{e}}$ respectively (both at time $t-1$), and B_4 is the output slot into which Layer 2 writes its retrieved prediction.

Layer 1: spatial gathering via RPE shifts We use $2k$ attention heads. For each neighbor offset δ_j ($j = 1, \ldots, k$), one head retrieves the cell at $(\,t-1,\ \mathbf{i} + \delta_j\,)$ from the neighborhood $\mathcal{N}_{t-1,\mathbf{i}}$ and another retrieves the cell at $(\,t-1,\ \mathbf{i} + \delta_j + \hat{e}\,)$ from the neighborhood $\mathcal{N}_{t-1,\mathbf{i}+\hat{e}}$. We implement each retrieval by adding a relative position bias to the attention scores:

$$A^{(1)}_{m,n} = q_m^\top k_n + \mathrm{RPE}^{(\Delta t,\, \Delta \mathbf{p})}_{m,n}, \tag{6}$$

where $\mathrm{RPE}^{(\Delta t,\, \Delta \mathbf{p})}_{m,n}$ is large when the spatiotemporal offset of token n relative to token m matches the head's designated target, and $-\infty$ otherwise. The softmax concentrates each head's attention on its designated target. The first group of k heads writes their retrieved one-hot states into B_2 and the second group writes into B_3, so that after Layer 1:

$$h^{(1)}_{t,\mathbf{i}} = \big[\, e(x_{t,\mathbf{i}})\,;\, \tilde{\mathcal{N}}_{t-1,\mathbf{i}}\,;\, \tilde{\mathcal{N}}_{t-1,\mathbf{i}+\hat{e}}\,;\, \mathbf{0}_V\,\big], \tag{7}$$

where $\tilde{\mathcal{N}}_{t-1,\mathbf{i}}$, $\tilde{\mathcal{N}}_{t-1,\mathbf{i}+\hat{e}} \in \mathbb{R}^{kV}$ are the concatenated one-hot encodings of the k neighbors in $\mathcal{N}_{t-1,\mathbf{i}}$ and $\mathcal{N}_{t-1,\mathbf{i}+\hat{e}}$, respectively.

Layer 2: pattern matching A single attention head (no RPE) compares $\tilde{\mathcal{N}}_{t-1,\mathbf{i}+\hat{e}}$ against $\tilde{\mathcal{N}}_{t'-1,\mathbf{i}'}$ for context tokens via inner product. The score equals k if and only if the two neighborhoods agree on all k cells, and at most $k-1$ otherwise. The value projection extracts the matched cell's state $e(x_{t',\mathbf{i}'})$ from B_1 and writes it into B_4.

Comparison with the spatial-aware construction The naive baseline successfully predicts the dynamical rule, but its positional representation scales with the grid volume. Our spatial-aware embeddings (Theorem 4.2) require $d_p = 2d + 2$ when each head routes a single offset ($m = 1$), and $d_p \leq 2(m + d) + 2$ when each head simultaneously routes $m = \lceil |U|/h \rceil$ offsets, both independent of L. In contrast, the RPE bias table assigns a learned scalar to each distinct relative sequence offset. After flattening a d-dimensional grid, the relative offset $\Delta\mathbf{p} \in \mathbb{Z}^d$ (with each component ranging over W values) yields $W^d = L$ distinct spatial offsets. Combined with T time steps, the table spans $O(T \cdot L)$ entries, making the positional representation increasingly expensive as the grid grows.

Beyond this parameter efficiency, the RPE formulation implicitly encodes substantial structural information about the underlying spatial domain. The bias table is indexed by a $(d{+}1)$-dimensional offset (d spatial axes plus one temporal axis), with each spatial axis ranging over W values. This structure directly reveals the dimensionality d of the grid, the width W along each axis, and the independence of spatial and temporal dimensions. By contrast, our construction uses standard learned positional embeddings over the flattened 1D sequence. The model must discover from data alone that the 1D sequence encodes a multidimensional spatial grid, which cells constitute the relevant neighborhood, and how temporal and spatial dependencies are structured. As demonstrated empirically in Section 5, standard gradient descent on this architecture converges to the spatial induction head mechanism predicted by our theory, confirming that transformers have sufficient inductive bias to discover spatial routing without explicit architectural encoding of the domain geometry.

D Deferred Proofs for Spatial-Aware Embeddings

D.1 Formal Definition and Multidimensional Local Interpolation

Definition D.1 (Spatial-aware embedding)**.** A sequence of positional embeddings $\mathbf{p}_{t,\mathbf{i}}$ is *spatial-aware* for an attention head assigned to route m specific spatial offsets if there exist projection matrices $\mathbf{W}_Q^{\text{pos}}, \mathbf{W}_K^{\text{pos}}$ and a constant $\Delta > 0$ such that for all $(t, \mathbf{i})$ and causally visible $(t', \mathbf{i}')$, the inner product $\langle \mathbf{W}_Q^{\text{pos}} \mathbf{p}_{t,\mathbf{i}},\ \mathbf{W}_K^{\text{pos}} \mathbf{p}_{t',\mathbf{i}'} \rangle$ satisfies:

$$\begin{cases} s_j & \text{if } (t', \mathbf{i}') = (t{-}1,\ \mathbf{i} + \delta_j \bmod \mathbf{L}) \text{ for each offset } j \in \{1, \ldots, m\}, \\ \leq s_{\min} - \Delta & \text{otherwise,} \end{cases} \tag{8}$$

where $\{s_1, \ldots, s_m\}$ are mutually distinct target scores and $s_{\min} = \min_j s_j$; and the induced softmax attention weights

$$\alpha_j \ = \ \frac{\exp(s_j)}{\sum_{l=1}^{m} \exp(s_l)}, \qquad j = 1, \ldots, m, \tag{9}$$

which are fully determined by $\mathbf{W}_Q^{\text{pos}}, \mathbf{W}_K^{\text{pos}}$, and the embeddings $\{\mathbf{p}_{t,\mathbf{i}}\}$, satisfy the *non-degeneracy condition*: for every pair of distinct neighborhood configurations $\mathbf{x} \neq \mathbf{x}' \in \mathcal{S}^m$,

$$\sum_{j=1}^{m} \alpha_j \, x_j \ \neq \ \sum_{j=1}^{m} \alpha_j \, x'_j. \tag{10}$$

Lemma D.2 (Multidimensional local interpolation)**.** *Let $G = \mathbb{Z}_{L_1} \times \cdots \times \mathbb{Z}_{L_d}$ be a d-dimensional discrete grid with periodic boundary conditions, where $L_j \geq 2$ for each $j \in \{1, \ldots, d\}$. Let $N = \{\delta_1, \ldots, \delta_m\} \subset G$ be a set of m distinct spatial offsets, let $\{s_1, \ldots, s_m\} \subset \mathbb{R}$ be prescribed target values, and let $W > 0$ be an arbitrary penalty threshold.*

A trigonometric polynomial *on G is any function of the form*

$$g \ : \ G \ \longrightarrow \ \mathbb{R}, \qquad g(\delta) \ = \ \sum_{\omega \in \Omega} \Big[a_\omega \cos\Big(2\pi \sum_{j=1}^{d} \tfrac{\omega_j \, \delta_j}{L_j} \Big) + b_\omega \sin\Big(2\pi \sum_{j=1}^{d} \tfrac{\omega_j \, \delta_j}{L_j} \Big) \Big], \tag{11}$$

where $\Omega \subseteq \widehat{G} \cong \mathbb{Z}_{L_1} \times \cdots \times \mathbb{Z}_{L_d}$ is a finite set of frequency vectors *and $a_\omega, b_\omega \in \mathbb{R}$ are learnable weights.*

There exists a choice of frequencies Ω and weights $\{a_\omega, b_\omega\}$ such that $|\Omega| \leq m + d$ and the resulting trigonometric polynomial g satisfies:

1. ***Interpolation:*** *$g(\delta_h) = s_h$ for all $h \in \{1, \ldots, m\}$, and*
2. ***Penalty:*** *$g(\delta) \leq -W$ for every $\delta \notin N$.*

Proof of Lemma D.2. The proof embeds G into a Euclidean feature space via trigonometric characters, reduces the interpolation conditions to a linear system whose solution space has a kernel of positive dimension (i.e., underdetermined), and then exploits that null-space freedom to drive g arbitrarily negative on $G \setminus N$. We use standard facts about characters of finite abelian groups.

Step 1: Frequency Selection. We construct $\Omega \subset \widehat{G}$ as the union of two sets.

(a) Axial frequencies. Let $\hat{G}$ denote the dual group of G, consisting of all characters (homomorphisms from G to the complex unit circle). We define $\hat{e}_j = (0, \ldots, 1, \ldots, 0) \in \hat{G}$ for $j = 1, \ldots, d$ as the j-th standard basis character. When evaluated at a grid point $\delta \in G$, this character isolates the j-th spatial coordinate, yielding the complex value $\exp(i2\pi\delta_j/L_j)$.

Let $\phi_{ax} : G \to \mathbb{R}^{2d}$ be a mapping defined by extracting the real and imaginary components of these d basis characters:

$$\phi_{ax}(\delta) = \left(\cos\left(\frac{2\pi\delta_j}{L_j}\right), \sin\left(\frac{2\pi\delta_j}{L_j}\right)\right)_{j=1}^{d} \tag{12}$$

For any axis j, the pair $(\cos(2\pi\delta_j/L_j), \sin(2\pi\delta_j/L_j))$ maps the discrete coordinate δ_j to a unique position on the unit circle. Provided $L_j \geq 2$, no two distinct coordinates produce the same pair, making the mapping injective on $\mathbb{Z}_{L_j}$.

Consequently, the combined map ϕ_{ax} is injective on the entire multidimensional grid G. Because every spatial point maps to a unique real-valued vector, no two distinct grid positions can overlap in the feature space. This injectivity is used in Step 4 to separate and penalize all non-target points.

(b) Interpolation-spanning frequency vectors. We need a set of frequency vectors whose trigonometric evaluations at the m target points $N = \{\delta_1, \ldots, \delta_m\}$ are jointly capable of fitting any prescribed target values $\{s_1, \ldots, s_m\}$. Since the trigonometric functions indexed by $\widehat{G}$ form a complete orthonormal basis for real-valued functions on G (Fourier theory on finite abelian groups), their evaluations at any m distinct points span $\mathbb{R}^m$. Consequently, there exists a subset $\Omega_{\text{core}} \subset \widehat{G}$ with $|\Omega_{\text{core}}| \leq m$ such that the $m \times 2|\Omega_{\text{core}}|$ matrix whose (h, ω) entry is $(\cos(2\pi \sum_j \omega_j \delta_{h,j}/L_j),\ \sin(2\pi \sum_j \omega_j \delta_{h,j}/L_j))$ — i.e., the evaluation of each frequency in Ω_{core} at each target point — has full row rank m. This guarantees the interpolation system is feasible for any target values.

Setting $\Omega = \{\hat{\mathbf{e}}_1, \ldots, \hat{\mathbf{e}}_d\} \cup \Omega_{\text{core}}$ yields $|\Omega| \leq m + d$.

Step 2: Linear System Formulation. For each $\delta \in G$, define the feature vector $\mathbf{v}_\delta \in \mathbb{R}^{2|\Omega|}$ by evaluating the real and imaginary components of the characters in Ω:

$$\mathbf{v}_\delta \;=\; \Big[\cos\Big(2\pi \sum_j \tfrac{\hat{e}_{1,j}\,\delta_j}{L_j}\Big),\ \sin\Big(2\pi \sum_j \tfrac{\hat{e}_{1,j}\,\delta_j}{L_j}\Big),\ \ldots,\ \cos\Big(2\pi \sum_j \tfrac{\omega_{k,j}\,\delta_j}{L_j}\Big),\ \sin\Big(2\pi \sum_j \tfrac{\omega_{k,j}\,\delta_j}{L_j}\Big)\Big]^\top, \tag{13}$$

where $\omega_k \in \Omega_{\text{core}}$.

Let $\mathbf{c} \in \mathbb{R}^{2|\Omega|}$ be the corresponding coefficient vector, such that $g(\delta) = \mathbf{c}^\top \mathbf{v}_\delta$. The k interpolation constraints $g(\delta_h) = s_h$ yield the linear system

$$\mathbf{E}\,\mathbf{c} = \mathbf{s}, \qquad \mathbf{E} \in \mathbb{R}^{m \times 2|\Omega|}, \quad \mathbf{s} = [s_1, \ldots, s_m]^\top, \tag{14}$$

where row h of $\mathbf{E}$ is $\mathbf{v}_{\delta_h}^\top$.

Step 3: Rank–Nullity Argument.

Claim D.3. $\operatorname{rank}(\mathbf{E}) = k$.

Proof of claim. The submatrix of $\mathbf{E}$ indexed by Ω_{core} already has row rank k by construction in Step 1(b); column augmentation with the axial-frequency terms leaves the row rank unchanged. □

Because the number of columns $2|\Omega|$ strictly exceeds the row rank k, the Rank–Nullity Theorem dictates $\dim \ker(\mathbf{E}) > 0$. The system therefore admits an affine solution space of the form $\mathbf{c}_{\text{base}} + \ker(\mathbf{E})$, where $\mathbf{c}_{\text{base}}$ is a particular solution to (14).

Step 4: Penalty Enforcement. Let $\mathbf{c} = \mathbf{c}_{\text{base}} + t\,\mathbf{c}_{\text{null}}$ for some $t > 0$ and $\mathbf{c}_{\text{null}} \in \ker(\mathbf{E})$. Since $\mathbf{E}\mathbf{c}_{\text{null}} = \mathbf{0}$, the interpolation constraints $g(\delta_h) = s_h$ are automatically preserved for all t.

Existence of a uniformly-negative null direction. We need $\mathbf{c}_{\text{null}} \in \ker(\mathbf{E})$ satisfying $\mathbf{c}_{\text{null}}^\top \mathbf{v}_\delta < 0$ for every $\delta \in G \setminus N$. By Gordan's theorem of the alternative, such a vector exists unless there are non-negative scalars $\lambda_\delta \geq 0$ (not all zero) for $\delta \notin N$ such that

$$\sum_{\delta \notin N} \lambda_\delta\, \mathbf{v}_\delta \;\in\; \text{rowspace}(\mathbf{E}) = \text{span}\big\{\mathbf{v}_{\delta_h}\big\}_{h=1}^{m}. \tag{15}$$

We rule out (15) using the character zero-sum identity. For each axial frequency $\hat{\mathbf{e}}_j \in \Omega$, the L_j-th roots of unity sum to zero:

$$\sum_{\delta \in G} \begin{pmatrix} \cos(2\pi\delta_j/L_j) \\ \sin(2\pi\delta_j/L_j) \end{pmatrix} = \mathbf{0} \qquad (L_j \geq 2). \tag{16}$$

Hence $\sum_{\delta \notin N} \phi_{ax}(\delta) = -\sum_{h=1}^{m} \phi_{ax}(\delta_h)$. If (15) held for some $\boldsymbol{\lambda} \geq \mathbf{0}$ (not all zero), its axial block would satisfy

$$\sum_{\delta \notin N} \lambda_\delta\, \phi_{ax}(\delta) = \sum_{h=1}^{m} \mu_h\, \phi_{ax}(\delta_h) \tag{17}$$

for some $\boldsymbol{\mu} \in \mathbb{R}^m$. Adding $\bar{\lambda} \sum_h \phi_{ax}(\delta_h)$ to both sides (where $\bar{\lambda} := \sum_{\delta \notin N} \lambda_\delta > 0$) and using the zero-sum identity converts (17) to

$$\bar{\lambda} \sum_{\delta \in G} \phi_{ax}(\delta) + \sum_{\delta \notin N} (\lambda_\delta - \bar{\lambda})\, \phi_{ax}(\delta) = \sum_{h=1}^{m} (\mu_h + \bar{\lambda})\, \phi_{ax}(\delta_h). \tag{18}$$

The left-hand side collapses to $\sum_{\delta \notin N} (\lambda_\delta - \bar{\lambda})\, \phi_{ax}(\delta)$, a combination of the $|G| - m$ *distinct* background axial vectors (distinct by injectivity of ϕ_{ax}, Step 1a) with signed weights summing to zero. Because each $\phi_{ax}(\delta)$ is a point on the discrete d-torus $\prod_j (\mathbb{Z}_{L_j}/L_j)$ and all $|G|$ such points are distinct, this signed combination can equal a combination of the m target axial vectors only in degenerate configurations that our specific frequency selection (Step 1b) is constructed to avoid: the full row rank of $\mathbf{E}$ on N together with the axial injectivity jointly over-determine the system, making (15) infeasible. Hence $\mathbf{c}_{\text{null}}$ exists.

For any background point $\delta \notin N$, the evaluation $g(\delta) = \mathbf{c}_{\text{base}}^\top \mathbf{v}_\delta + t\,\mathbf{c}_{\text{null}}^\top \mathbf{v}_\delta$ is affine in t. Choosing

$$t \;\geq\; \frac{W + \max_{\delta \notin N} \mathbf{c}_{\text{base}}^\top \mathbf{v}_\delta}{-\min_{\delta \notin N} \mathbf{c}_{\text{null}}^\top \mathbf{v}_\delta}$$

ensures $g(\delta) \leq -W$ for all $\delta \notin N$. The finite cardinality of $G \setminus N$ guarantees the numerator is finite and the denominator is strictly positive, so such a t exists, fulfilling both conditions with $|\Omega| \leq m + d$ frequencies. □

Below, we show an explicit construction for Lemma D.2 for $d = 1$ and $m = 3$ (1D Cellular Automata):

Lemma D.4 (Constant-dimension local interpolation)**.** *Let $L \geq 4$, let $v_{-1}, v_0, v_1 \in \mathbb{R}$ be prescribed values, and let $W > 0$.*

Using the frequency set $\Omega = \{0, 1, 2\}$, which satisfies the bound $|\Omega| \leq m + d$, there exist coefficients $a_0, a_1, a_2, b_1 \in \mathbb{R}$ such that the trigonometric polynomial

$$g(\delta) = a_0 + a_1 \cos\Big(\tfrac{2\pi}{L}\delta\Big) + a_2 \cos\Big(2\tfrac{2\pi}{L}\delta\Big) + b_1 \sin\Big(\tfrac{2\pi}{L}\delta\Big)$$

satisfies the interpolation conditions $g(0) = v_0$, $g(1) = v_1$, $g(-1) = v_{-1}$, and the penalty condition $g(\delta) \leq -W$ for all $\delta \in \mathbb{Z}_L \setminus \{-1, 0, 1\}$.

Proof. We construct an explicit solution to the feasibility problem as a special case of the multidimensional framework in Lemma D.2. Specifically, we map the general parameters to our 1D setting as follows:

- **Dimensions and Grid:** $d = 1$ and $G = \mathbb{Z}_L$.
- **Targets:** $m = 3$ target points at offsets $N = \{-1, 0, 1\}$ with prescribed values $\{s_1, s_2, s_3\} = \{v_{-1}, v_0, v_1\}$.
- **Frequencies:** We select the frequency set $\Omega = \{0, 1, 2\}$. This satisfies the general cardinality bound, since $|\Omega| = 3 \leq m + d = 4$.
- **Weights:** The scalar coefficients a_0, a_1, a_2, b_1 correspond directly to the general weights a_ω, b_ω for $\omega \in \Omega$, where the missing sine weights are trivially set to zero ($b_0 = 0$ since $\sin(0) = 0$, and $b_2 = 0$).

Let $c_\delta = \cos\left(\frac{2\pi}{L}\delta\right)$. Using the double-angle identity, we can express our target function $g(\delta)$ as a quadratic polynomial in c_δ plus a sine term:

$$g(\delta) = a_0 + a_1 c_\delta + a_2(2c_\delta^2 - 1) + b_1 \sin\left(\frac{2\pi}{L}\delta\right).$$

Our goal is to choose coefficients that satisfy the target interpolation conditions $g(0) = v_0$, $g(1) = v_1$, and $g(-1) = v_{-1}$, while driving the value of $g(\delta)$ below $-W$ for all other $\delta \in \mathbb{Z}_L$.

1. Resolving the Asymmetry. We decouple the problem into symmetric and asymmetric components. The sine term alone handles the asymmetry between $g(1)$ and $g(-1)$. Assuming $L \geq 4$, $\sin\left(\frac{2\pi}{L}\right) > 0$, allowing us to perfectly capture the difference between the boundaries by setting:

$$b_1 = \frac{v_1 - v_{-1}}{2\sin(2\pi/L)}.$$

2. Satisfying the Target Intercepts. For the symmetric component, we define the target average $M = \frac{v_1 + v_{-1}}{2}$. Applying the constraints $g(0) = v_0$ and $\frac{1}{2}(g(1) + g(-1)) = M$ yields a linear system for the remaining coefficients:

$$a_0 + a_1 + a_2 = v_0, \tag{19}$$

$$(a_0 - a_2) + a_1 c_1 + 2a_2 c_1^2 = M. \tag{20}$$

To solve this system, we subtract (19) from (20) to eliminate a_0:

$$a_1(c_1 - 1) + 2a_2(c_1^2 - 1) = M - v_0.$$

Since $L \geq 4$, $c_1 < 1$, we can safely divide by $(c_1 - 1)$. Using the difference of squares $c_1^2 - 1 = (c_1 - 1)(c_1 + 1)$, we obtain:

$$a_1 + 2a_2(c_1 + 1) = \frac{M - v_0}{c_1 - 1} \equiv \Gamma.$$

This defines a_1 in terms of a constant Γ and our free parameter a_2:

$$a_1 = \Gamma - 2a_2(c_1 + 1). \tag{21}$$

From (19), we also have $a_0 = v_0 - a_1 - a_2$. Substituting this a_0 into the symmetric portion of $g(\delta)$ and subtracting v_0 from both sides isolates the c_δ terms:

$$\begin{aligned} g(\delta) - v_0 - b_1 \sin\left(\frac{2\pi}{L}\delta\right) &= (v_0 - a_1 - a_2) + a_1 c_\delta + a_2(2c_\delta^2 - 1) - v_0 \\ &= a_1(c_\delta - 1) + 2a_2(c_\delta^2 - 1) \\ &= (c_\delta - 1)\big[a_1 + 2a_2(c_\delta + 1)\big]. \end{aligned}$$

Next, we substitute our expression for a_1 from (21) directly into the brackets:

$$\begin{aligned} a_1 + 2a_2(c_\delta + 1) &= \big[\Gamma - 2a_2(c_1 + 1)\big] + 2a_2(c_\delta + 1) \\ &= \Gamma + 2a_2(c_\delta - c_1). \end{aligned}$$

Recombining this simplified bracket with the $(c_\delta - 1)$ factor and restoring the asymmetric sine term yields our highly convenient factored form:

$$g(\delta) - v_0 = (c_\delta - 1)\big[\Gamma + 2a_2(c_\delta - c_1)\big] + b_1 \sin\left(\frac{2\pi}{L}\delta\right).$$

This factorization makes the interpolation properties self-evident. At $\delta = 0$, the $(c_\delta - 1)$ term vanishes, cleanly leaving $g(0) = v_0$. At $\delta = \pm 1$, the $(c_\delta - c_1)$ term vanishes, and the remaining terms evaluate to v_1 and v_{-1} respectively.

3. Enforcing the Global Penalty. It remains to establish the suppression condition $g(\delta) \leq -W$ for all non-target points $\delta \in \mathbb{Z}_L \setminus \{-1, 0, 1\}$.

Setting the free parameter $a_2 = -\Lambda$ for some scalar $\Lambda > 0$ and expanding $g(\delta) \leq -W$ yields the equivalent requirement:

$$2\Lambda(1 - c_\delta)(c_1 - c_\delta) \geq W + v_0 + \Gamma(c_\delta - 1) + b_1 \sin\left(\frac{2\pi}{L}\delta\right). \tag{22}$$

For all non-target points, $c_\delta \leq c_2 < c_1 < 1$. Consequently, the spatial factors $(1 - c_\delta)$ and $(c_1 - c_\delta)$ are strictly positive. Because the domain $\mathbb{Z}_L$ is finite, these factors are uniformly bounded away from zero, ensuring that the left-hand side scales monotonically with Λ. This allows the penalty term to arbitrarily dominate the right-hand side for sufficiently large Λ.

To determine a sufficient global condition, we bound both sides. Applying the triangle inequality to the right-hand side establishes a uniform upper bound of $W + v_0 + |\Gamma| + |b_1|$. On the left-hand side, the most restrictive spatial constraint occurs at $\delta = 2$. Bounding the spatial factors against this worst-case scenario yields the sufficient condition:

$$\Lambda \geq \frac{W + v_0 + |\Gamma| + |b_1|}{2(1 - c_2)(c_1 - c_2)}. \tag{23}$$

Choosing any Λ that satisfies this lower bound enforces all spatial penalties, thereby concluding the proof. □

D.2 Extended Statement and Proof of Theorem 4.2 (Constructive Bound for Spatial Awareness)

Theorem D.5 (Bounds for Spatial Addressing)**.** *For any d-dimensional discrete spatial grid of size $\mathbf{L}$, temporal horizon T, margin $\Delta > 0$, and a target subset of m spatial offsets assigned to an attention head, a positional embedding $\mathbf{p}_{(t,\mathbf{i})} \in \mathbb{R}^{d_p}$ encoding these targets satisfies the following dimensional bounds:*

1. ***Independent Per-Head Addressing** ($m = 1$)**:** When an attention head is responsible for isolating a single spatial offset, there exists an explicit embedding requiring a total dimension of $d_p = \mathcal{O}(d)$. Specifically, $d_p = 2d + 2$.*
2. ***Simultaneous Addressing** ($m > 1$)**:** When an attention head is burdened with simultaneously routing a neighborhood of m offsets, there exists a valid spatial-aware embedding requiring a total dimension of $d_p \leq 2(m + d) + 2$.*

Proof. We construct the positional embedding $\mathbf{p}_{(t,\mathbf{i})} \in \mathbb{R}^{d_p}$ by partitioning it into a temporal component of dimension $d_T = 2$ and a spatial component of dimension d_S, such that $d_p = d_T + d_S$. Let the full embedding be:

$$\mathbf{p}_{(t,\mathbf{i})} = \begin{bmatrix} \mathbf{p}_t^{\text{time}} \\ \mathbf{p}_{\mathbf{i}}^{\text{space}} \end{bmatrix}. \tag{24}$$

We define the query and key projection matrices as block-diagonal:

$$\mathbf{Q}_0^{\text{pos}} = \begin{bmatrix} \mathbf{Q}_T & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_S \end{bmatrix}, \qquad \mathbf{K}_0^{\text{pos}} = \begin{bmatrix} \mathbf{K}_T & \mathbf{0} \\ \mathbf{0} & \mathbf{K}_S \end{bmatrix}. \tag{25}$$

The total inner product therefore decomposes additively:

$$\langle \mathbf{Q}_0^{\text{pos}} \mathbf{p}_{(t,\mathbf{i})},\ \mathbf{K}_0^{\text{pos}} \mathbf{p}_{(t',\mathbf{i}')} \rangle = \underbrace{\langle \mathbf{Q}_T \mathbf{p}_t^{\text{time}},\ \mathbf{K}_T \mathbf{p}_{t'}^{\text{time}} \rangle}_{h(t,t')} + \underbrace{\langle \mathbf{Q}_S \mathbf{p}_{\mathbf{i}}^{\text{space}},\ \mathbf{K}_S \mathbf{p}_{\mathbf{i}'}^{\text{space}} \rangle}_{g(\mathbf{i},\mathbf{i}')}. \tag{26}$$

Shared Temporal Component. For both addressing regimes, we set the temporal embedding to $\mathbf{p}_t^{\text{time}} = [t, 1]^\top \in \mathbb{R}^2$. Setting $\mathbf{Q}_T = \text{diag}(-W_T, 1)$ and $\mathbf{K}_T = [0, 1; W_T, W_T]$ yields the temporal inner product $h(t, t') = -W_T(t - t' - 1)$. Because causal masking ensures $t' \leq t - 1$ for all visible positions, the temporal score is strictly non-positive:

$$h(t, t') = \begin{cases} 0 & \text{if } t' = t - 1, \\ \leq -W_T & \text{if } t' \leq t - 2. \end{cases} \tag{27}$$

Case 1: Independent Per-Head Addressing ($m = 1$). We set $d_S = 2d$ and construct the spatial embedding by concatenating one Fourier mode per axis. Specifically, for $\mathbf{i} = (i_1, \ldots, i_d)$, define

$$\mathbf{p}_{\mathbf{i}}^{\text{space}} = \begin{bmatrix} \mathbf{p}_{i_1}^{\text{axis } 1} \\ \vdots \\ \mathbf{p}_{i_d}^{\text{axis } d} \end{bmatrix} \in \mathbb{R}^{2d}, \qquad \mathbf{p}_{i_c}^{\text{axis } c} = \begin{bmatrix} \cos\left(\frac{2\pi}{L_c} i_c\right) \\ \sin\left(\frac{2\pi}{L_c} i_c\right) \end{bmatrix}, \tag{28}$$

so that each axis $c \in \{1, \ldots, d\}$ contributes a unit vector on the complex unit circle S^1, parametrised by its grid frequency $\omega_c = 2\pi / L_c$.

We set $\mathbf{Q}_S = \mathbf{I}_{2d}$ and define $\mathbf{K}_S$ as a block-diagonal matrix of scaled 2×2 rotation matrices:

$$\mathbf{K}_S = W_S \cdot \text{blkdiag}(R(\theta_{h,1}),\ \ldots,\ R(\theta_{h,d}))\,, \qquad R(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}, \tag{29}$$

where $\theta_{h,c} = \frac{2\pi}{L_c} \delta_{h,c}$ encodes the target offset $\delta_h = (\delta_{h,1}, \ldots, \delta_{h,d})$ along axis c.

Computing $g(\mathbf{i}, \mathbf{i}')$. Let $\mathbf{Q}_S = \mathbf{I}_{2d}$, since $\mathbf{K}_S$ is block-diagonal, the spatial inner product decomposes axis-wise:

$$g(\mathbf{i}, \mathbf{i}') = \langle \mathbf{p}_{\mathbf{i}}^{\text{space}},\ \mathbf{K}_S\, \mathbf{p}_{\mathbf{i}'}^{\text{space}} \rangle = W_S \sum_{c=1}^{d} \left\langle \mathbf{p}_{i_c}^{\text{axis } c},\ R(\theta_{h,c})\, \mathbf{p}_{i'_c}^{\text{axis } c} \right\rangle. \tag{30}$$

For each axis c, the rotation $R(\theta_{h,c})$ acts on the key embedding as

$$R(\theta_{h,c}) \begin{bmatrix} \cos\left(\frac{2\pi}{L_c} i'_c\right) \\ \sin\left(\frac{2\pi}{L_c} i'_c\right) \end{bmatrix} = \begin{bmatrix} \cos\left(\frac{2\pi}{L_c} i'_c + \theta_{h,c}\right) \\ \sin\left(\frac{2\pi}{L_c} i'_c + \theta_{h,c}\right) \end{bmatrix} = \begin{bmatrix} \cos\left(\frac{2\pi}{L_c} (i'_c + \delta_{h,c})\right) \\ \sin\left(\frac{2\pi}{L_c} (i'_c + \delta_{h,c})\right) \end{bmatrix}, \tag{31}$$

where the final equality substitutes $\theta_{h,c} = \frac{2\pi}{L_c} \delta_{h,c}$. Taking the inner product with $\mathbf{p}_{i_c}^{\text{axis } c}$ and applying the identity $\cos\alpha \cos\beta + \sin\alpha \sin\beta = \cos(\alpha - \beta)$ yields

$$\left\langle \mathbf{p}_{i_c}^{\text{axis } c},\ R(\theta_{h,c})\, \mathbf{p}_{i'_c}^{\text{axis } c} \right\rangle = \cos\left(\tfrac{2\pi}{L_c} (i_c - i'_c - \delta_{h,c})\right). \tag{32}$$

Summing over all d axes gives the closed form

$$g(\mathbf{i}, \mathbf{i}') = W_S \sum_{c=1}^{d} \cos\left(\tfrac{2\pi}{L_c} (i_c - i'_c - \delta_{h,c})\right). \tag{33}$$

Target score. When $\mathbf{i}' \equiv \mathbf{i} - \delta_h \pmod{\mathbf{L}}$, the argument of each cosine in (33) satisfies $i_c - i'_c - \delta_{h,c} \equiv 0 \pmod{L_c}$, so every term equals 1, as a result:

$$g_{\text{target}} \;=\; d \cdot W_S. \tag{34}$$

Non-target upper bound. For any $\mathbf{i}'$ not congruent to $\mathbf{i} - \delta_h \pmod{\mathbf{L}}$, there exists at least one axis c^* such that $i_{c^*} - i'_{c^*} - \delta_{h,c^*} \not\equiv 0 \pmod{L_{c^*}}$. Since all positions are integer-valued on a grid of size L_{c^*}, this residue is a nonzero integer, so the argument of the corresponding cosine is a nonzero integer multiple of $\frac{2\pi}{L_{c^*}}$. Because cosine is strictly less than 1 at any nonzero angle, and attains its maximum value closest to zero at $\pm\frac{2\pi}{L_{c^*}}$, we bound the mismatch axis contribution by $\cos\Big(\frac{2\pi}{L_{c^*}}\Big) \le \cos\Big(\frac{2\pi}{L_{\max}}\Big)$, where $L_{\max} = \max_c L_c$ gives the adversarially smallest angular gap. All remaining $d-1$ axes contribute at most 1 each, so

$$g_{\text{non-target}} \;\le\; W_S\Big((d-1) + \cos\Big(\tfrac{2\pi}{L_{\max}}\Big)\Big). \tag{35}$$

Enforcing the margin Δ. The spatial gap between the target and the nearest non-target is therefore

$$g_{\text{target}} - g_{\text{non-target}} \;\ge\; W_S\Big(1 - \cos\Big(\tfrac{2\pi}{L_{\max}}\Big)\Big). \tag{36}$$

Setting this gap to be at least Δ requires

$$W_S \;\ge\; \frac{\Delta}{1 - \cos(2\pi/L_{\max})}. \tag{37}$$

Combined score and uniqueness. The full pre-softmax attention score for a key at time t' and spatial position $\mathbf{i}'$ is the sum of the temporal and spatial components: $h(t,t') + g(\mathbf{i},\mathbf{i}')$.

For the target time step $t' = t-1$, the temporal score is exactly zero. (see the Shared Temporal Component above). The unique spatial maximizer $\mathbf{i}' \equiv \mathbf{i} - \delta_h \pmod{L}$ achieves the target score $g_{\text{target}} = d \cdot W_S$, while all competing spatial positions score at most $g_{\text{non-target}} \le d \cdot W_S - \Delta$.

For all earlier time steps $t' \le t-2$, the temporal penalty is $h(t,t') \le -W_T$. The maximum possible spatial score at any position is $g_{\text{target}} = d \cdot W_S$. To ensure these older tokens are also suppressed by at least the margin Δ relative to the target score, we require the combined score to satisfy:

$$d \cdot W_S - W_T \le d \cdot W_S - \Delta \implies W_T \ge \Delta. \tag{38}$$

By setting $W_T \ge \Delta$, the embedding of total dimension $d_p = 2d + 2 = \mathcal{O}(d)$ achieves exact isolation of the target offset δ_h across the entire spatiotemporal context.

Case 2: Simultaneous Addressing ($m > 1$). When a head must simultaneously isolate m offsets $N = \{\delta_1, \dots, \delta_m\} \subset G$, we proceed in three stages: (i) construct a trigonometric polynomial on G that peaks exactly at each target offset via Lemma D.2; (ii) embed that polynomial into the positional bilinear form via angle-addition; and (iii) verify that the non-degeneracy condition required by the lemma holds generically.

Stage (i): Invoking Lemma D.2. Apply Lemma D.2 with offsets N, prescribed target values $\{s_1, \dots, s_m\} \subset \mathbb{R}$ (chosen to be distinct and non-degenerate; see Stage (iii) below), and penalty threshold $W_S > 0$. The lemma guarantees the existence of a set Ω of frequency vectors with $|\Omega| \le m + d$, and real coefficients $\{a_\omega, b_\omega\}_{\omega\in\Omega}$, such that the trigonometric polynomial

$$g(\delta) = \sum_{\omega\in\Omega}\Big[a_\omega \cos\Big(2\pi \sum_{j=1}^{d} \tfrac{\omega_j \delta_j}{L_j}\Big) + b_\omega \sin\Big(2\pi \sum_{j=1}^{d} \tfrac{\omega_j \delta_j}{L_j}\Big)\Big] \tag{39}$$

satisfies $g(\delta_h) = s_h$ for all $h \in \{1, \dots, m\}$, and $g(\delta) \le -W_S$ for all $\delta \notin N$.

Stage (ii): Embedding into the bilinear form. We now show that $g(\mathbf{i} - \mathbf{i}')$ can be expressed exactly as the positional inner product $\langle \mathbf{Q}_S\, \mathbf{p}_{\mathbf{i}}^{\text{space}}, \mathbf{K}_S\, \mathbf{p}_{\mathbf{i}'}^{\text{space}} \rangle$ using a spatial embedding of dimension $d_S = 2|\Omega| \le 2(m+d)$.

For each frequency $\omega \in \Omega$, define the *phase* of position $\mathbf{i}$ at frequency ω as $\varphi_\omega(\mathbf{i}) = 2\pi \sum_{j=1}^{d} \omega_j i_j / L_j$, so that the argument of each term in (39) factors as

$$2\pi \sum_{j=1}^{d} \frac{\omega_j (i_j - i'_j)}{L_j} \;=\; \varphi_\omega(\mathbf{i}) - \varphi_\omega(\mathbf{i}'). \tag{40}$$

Define the 2-dimensional Fourier feature for position $\mathbf{i}$ at frequency ω as

$$\mathbf{e}_\omega(\mathbf{i}) = \begin{bmatrix} \cos \varphi_\omega(\mathbf{i}) \\ \sin \varphi_\omega(\mathbf{i}) \end{bmatrix} \in \mathbb{R}^2. \tag{41}$$

We construct the spatial embedding by concatenating these features over all $\omega \in \Omega$:

$$\mathbf{p}_\mathbf{i}^{\text{space}} = \begin{bmatrix} \mathbf{e}_{\omega_1}(\mathbf{i}) \\ \vdots \\ \mathbf{e}_{\omega_{|\Omega|}}(\mathbf{i}) \end{bmatrix} \in \mathbb{R}^{2|\Omega|}. \tag{42}$$

Set $\mathbf{Q}_S = \mathbf{I}_{2|\Omega|}$ and define $\mathbf{K}_S$ as a block-diagonal matrix with one 2×2 block per frequency:

$$\mathbf{K}_S = \text{blkdiag}\big(M_{\omega_1}, \ldots, M_{\omega_{|\Omega|}}\big), \qquad M_\omega = \begin{bmatrix} a_\omega & -b_\omega \\ b_\omega & a_\omega \end{bmatrix}. \tag{43}$$

Note that M_ω is a scaled rotation matrix of the form $\sqrt{a_\omega^2 + b_\omega^2} \cdot R(\arctan b_\omega / a_\omega)$, and is hence a direct generalisation of the Case 1 key matrix to arbitrary phase and amplitude.

Since $\mathbf{Q}_S = \mathbf{I}$ and $\mathbf{K}_S$ is block-diagonal, the inner product decomposes frequency-wise:

$$\langle \mathbf{Q}_S\, \mathbf{p}_\mathbf{i}^{\text{space}},\, \mathbf{K}_S\, \mathbf{p}_{\mathbf{i}'}^{\text{space}} \rangle = \sum_{\omega \in \Omega} \mathbf{e}_\omega(\mathbf{i})^\top\, M_\omega\, \mathbf{e}_\omega(\mathbf{i}'). \tag{44}$$

For each frequency ω, we expand the block contribution explicitly. By the standard angle-addition identities $\cos(\alpha - \beta) = \cos\alpha\cos\beta + \sin\alpha\sin\beta$ and $\sin(\alpha - \beta) = \sin\alpha\cos\beta - \cos\alpha\sin\beta$:

$$\begin{aligned} \mathbf{e}_\omega(\mathbf{i})^\top\, M_\omega\, \mathbf{e}_\omega(\mathbf{i}') &= [\cos\varphi_\omega(\mathbf{i}) \quad \sin\varphi_\omega(\mathbf{i})] \begin{bmatrix} a_\omega & -b_\omega \\ b_\omega & a_\omega \end{bmatrix} \begin{bmatrix} \cos\varphi_\omega(\mathbf{i}') \\ \sin\varphi_\omega(\mathbf{i}') \end{bmatrix} \\ &= a_\omega \Big(\cos\varphi_\omega(\mathbf{i}) \cos\varphi_\omega(\mathbf{i}') + \sin\varphi_\omega(\mathbf{i}) \sin\varphi_\omega(\mathbf{i}') \Big) \\ &\quad + b_\omega \Big(\sin\varphi_\omega(\mathbf{i}) \cos\varphi_\omega(\mathbf{i}') - \cos\varphi_\omega(\mathbf{i}) \sin\varphi_\omega(\mathbf{i}') \Big) \\ &= a_\omega \cos\Big(\varphi_\omega(\mathbf{i}) - \varphi_\omega(\mathbf{i}') \Big) + b_\omega \sin\Big(\varphi_\omega(\mathbf{i}) - \varphi_\omega(\mathbf{i}') \Big). \end{aligned} \tag{45}$$

Summing (45) over all $\omega \in \Omega$ and substituting $\varphi_\omega(\mathbf{i}) - \varphi_\omega(\mathbf{i}') = 2\pi \sum_j \omega_j (i_j - i'_j)/L_j$ recovers exactly $g(\mathbf{i} - \mathbf{i}')$ as defined in (39):

$$\langle \mathbf{Q}_S\, \mathbf{p}_\mathbf{i}^{\text{space}},\, \mathbf{K}_S\, \mathbf{p}_{\mathbf{i}'}^{\text{space}} \rangle = g(\mathbf{i} - \mathbf{i}'). \tag{46}$$

By the interpolation property of Lemma D.2, this inner product evaluates to s_h whenever $\mathbf{i} - \mathbf{i}' \equiv \delta_h \pmod{\mathbf{L}}$, and to at most $-W_S$ for all spatial offsets not in N.

Dimension count. The spatial embedding dimension is $d_S = 2|\Omega| \le 2(m + d)$. Combined with the shared temporal dimension $d_T = 2$, the total positional embedding dimension is

$$d_p = d_T + d_S = 2 + 2|\Omega| \le 2(m + d) + 2, \tag{47}$$

matching the bound stated in Case 2 of the theorem.

Stage (iii): Non-degeneracy of the induced softmax weights. Lemma D.2 requires the prescribed target scores $s_1, \ldots, s_m$ to be distinct, but we must also guarantee that their *induced* softmax weights $\alpha_h \propto e^{s_h}$ form a non-degenerate basis for the Layer 1 MLP. Specifically, the scalar superposition $z = \sum_{h=1}^m \alpha_h x_h$ must take a distinct value for all V^m possible local state configurations $x \in \{0, \ldots, V-1\}^m$.

A failure of non-degeneracy occurs if and only if two distinct configurations $x \neq x'$ yield the exact same superposition:

$$\sum_{h=1}^m \alpha_h (x_h - x'_h) = 0. \tag{48}$$

Let $\delta_h = x_h - x'_h \in \{-(V-1), \ldots, V-1\}$. Because the softmax weights are defined as $\alpha_h = e^{s_h}/Z$ (where $Z = \sum_k e^{s_k}$ is the normalization denominator), we can multiply through by Z to find that degeneracy requires:

$$\sum_{h=1}^{m} \delta_h e^{s_h} = 0. \tag{49}$$

For any fixed pair of distinct configurations $x \neq x'$, at least one difference $\delta_h \neq 0$. Therefore, the function $F_{x,x'}(\mathbf{s}) = \sum_{h=1}^{m} \delta_h e^{s_h}$ is a non-trivial, analytic function of the score vector $\mathbf{s} = (s_1, \ldots, s_m) \in \mathbb{R}^m$. By standard properties of analytic functions, the zero set of $F_{x,x'}$ is a manifold of strictly lower dimension, and thus has Lebesgue measure zero in $\mathbb{R}^m$.

Since there are exactly $\binom{V^m}{2}$ such pairs of distinct configurations, the total set of degenerate score configurations is a finite union of these measure-zero zero sets. Consequently, the entire degenerate set also has Lebesgue measure zero. Hence, for any fixed spatial offsets, almost every choice of target scores $\mathbf{s} \in \mathbb{R}^m$ simultaneously satisfies the interpolation conditions of Lemma D.2 while guaranteeing an injective, non-degenerate decoding basis for the Layer 1 MLP.

Combined score and uniqueness. By Equation (46) and the penalty property of Lemma D.2, the full attention score at time $t' = t - 1$ is:

$$h(t, t-1) + g(\mathbf{i} - \mathbf{i}') = \begin{cases} s_h & \text{if } \mathbf{i} - \mathbf{i}' \equiv \delta_h \pmod{L} \text{ for some } h, \\ \leq -W_S & \text{otherwise,} \end{cases} \tag{50}$$

where we used $h(t, t-1) = 0$. All m target offsets simultaneously receive their prescribed scores $s_h \geq s_{\min}$, while every non-target spatial position at $t' = t-1$ is suppressed below $-W_S \leq s_{\min} - \Delta$ (by choosing $W_S \geq \Delta - s_{\min}$).

For keys at earlier timesteps $t' \leq t - 2$, the temporal penalty is $h(t, t') \leq -W_T$. Let $g_{\max} = \max_{\delta \in G} g(\delta)$ be the maximum possible spatial score achieved by the trigonometric polynomial. To ensure the combined score $g + h$ for these older tokens respects the isolation margin Δ relative to the lowest target score $s_{\min}$, we require:

$$g_{\max} - W_T \leq s_{\min} - \Delta \implies W_T \geq g_{\max} - s_{\min} + \Delta. \tag{51}$$

Because $g_{\max}$ is bounded for any finite trigonometric polynomial, this threshold is finite and achievable. The spatial embedding of dimension $d_S = 2|\Omega| \leq 2(m + d)$ combined with this properly scaled temporal penalty thus suffices to route attention simultaneously to all m target offsets.

□

Example D.6 (Explicit Positional Embedding for 1D Cellular Automata)**.** *As a concrete instantiation of the simultaneous addressing regime (Case 2 of Theorem D.5) for 1D Elementary Cellular Automata ($d = 1$, neighborhood size $m = 3$ with offsets $\{-1, 0, 1\}$), we can construct the exact projection matrices and spatial-aware embeddings using a constant dimension $d_p = 7$. This dimension remains independent of both the temporal horizon T and the macroscopic grid size L.*

Proof. Following the block-diagonal decomposition established in Theorem D.5, we partition the positional embedding into temporal and spatial components: $d_p = d_T + d_S$.

1. Temporal Component ($d_T = 2$): We directly inherit the shared temporal embedding $p_t^{\text{time}} = [t, 1]^\top \in \mathbb{R}^2$ and the projection matrices Q_T, K_T exactly as defined in the proof of Theorem D.5. This guarantees a temporal inner product $h(t, t') = 0$ for the target preceding step $t' = t - 1$, and $h(t, t') \leq -W_T$ for all earlier causally visible steps $t' \leq t - 2$.

2. Spatial Component ($d_S = 5$): Rather than relying on the general $\mathcal{O}(m + d)$ frequency selection, we invoke the exact 1D construction from Lemma D.4. Given target scores s_R, s_C, s_L for spatial offsets $\{-1, 0, 1\}$ and a spatial penalty $W_S \geq \Delta - s_{\min}$, LemmaD.4 provides explicit coefficients a_0, a_1, a_2, b_1 for a trigonometric polynomial $g(\delta)$ spanning the frequency set $\Omega = \{0, 1, 2\}$.

We embed this polynomial into a 5-dimensional bilinear form. Let $Q_S = I_5$ and define the spatial embedding and key projection matrix as:

$$p_i^{\text{space}} = \begin{bmatrix} 1 \\ \cos(\frac{2\pi}{L}i) \\ \sin(\frac{2\pi}{L}i) \\ \cos(\frac{4\pi}{L}i) \\ \sin(\frac{4\pi}{L}i) \end{bmatrix} \in \mathbb{R}^5, \quad K_S = \text{blkdiag}\left(a_0, \begin{bmatrix} a_1 & -b_1 \\ b_1 & a_1 \end{bmatrix}, \begin{bmatrix} a_2 & 0 \\ 0 & a_2 \end{bmatrix}\right) \tag{52}$$

By applying standard angle-addition identities, the spatial inner product simplifies exactly to our target polynomial: $\langle Q_S p_i^{\text{space}}, K_S p_{i'}^{\text{space}} \rangle = g(i - i')$.

3. Synthesis and Penalty Calibration: The full embedding $p_{(t,i)} = [p_t^{\text{time}}; p_i^{\text{space}}] \in \mathbb{R}^7$ and the block-diagonal projections $Q_0^{\text{pos}}, K_0^{\text{pos}}$ sum the components to yield the full pre-softmax score: $h(t, t') + g(i - i')$. As established in Theorem D.5, by setting the temporal penalty $W_T \geq \max(s_L, s_C, s_R) - s_{\min} + \Delta$, the temporal decay dominates for all older tokens.

The resulting attention profile perfectly isolates the 3 spatial parent cells at $t - 1$ with their prescribed distinct scores, while all other causally visible tokens are suppressed by at least the margin Δ. This achieves the required generalized spatial awareness using exactly $d_p = 7$ dimensions, comfortably satisfying the theoretical upper bound of $2(m + d) + 2 = 10$ derived for the general case. □

Remark D.7 (Extension to the Union Set). Example D.6 explicitly constructs the spatial embedding for the $m = 3$ base neighborhood. To route the full union set $\mathcal{U} = \{-1, 0, 1, 2\}$ required for autoregressive alignment, the frequency set expands slightly to satisfy the bound $|\Omega| \leq |\mathcal{U}| + d = 5$, yielding a required embedding dimension of $d_S = 10$. The total positional dimension becomes $d_p = 12$, which remains independent of the grid volume L.

E Decoding and Retrieval

E.1 Non-Linear Decoding

This subsection characterizes when a scalar encoding of m discrete values can be decoded exactly into an orthogonal representation. The argument proceeds in three steps. Lemma E.1 establishes that a linear weighted sum can be made injective over V^m discrete configurations. Lemma E.2 then shows that no affine map suffices to recover V^m mutually orthogonal target vectors from a scalar input. Definition E.3 formalizes the interface the decoder must satisfy, and Lemma E.4 constructs an explicit 2-layer ReLU MLP that realizes it.

Connection to Layer 1 attention. Each spatial routing head γ is assigned a partition $S_\gamma \subseteq \{1, \ldots, k\}$ of size $m = |S_\gamma|$. Using the normalized value projection $w_s = s/(V - 1) \in [0, 1]$, the head outputs the scalar superposition

$$z_\gamma = \sum_{j \in S_\gamma} \alpha_j \, w_{x_j} + \eta_\gamma,$$

where α_j are the softmax attention weights concentrated on the m target cells, and η_γ is bounded leakage from non-target positions. The three lemmas below show that: (i) distinct configurations of the m cells produce distinguishable values of z_γ under a suitable choice of α; (ii) a linear decoder cannot map these scalar values to the required orthogonal patterns; and (iii) a compact ReLU MLP achieves exact decoding with bounded noise.

Lemma E.1 (Distinct weighted sums)**.** *Let $V \geq 2$, $m \geq 1$, and let $\mathcal{X} = \{0, \ldots, V - 1\}^m$. Define the weighted-sum map*

$$f_\alpha(x_1, \ldots, x_m) = \sum_{h=1}^{m} \alpha_h \, x_h, \qquad \alpha \in \Delta^{m-1} := \left\{\alpha \in \mathbb{R}^m_{>0} : \textstyle\sum_h \alpha_h = 1\right\}.$$

There exists $\alpha \in \Delta^{m-1}$ such that f_α is injective on $\mathcal{X}$, i.e., takes V^m distinct values. Moreover, the set of weight vectors for which f_α is not *injective has Lebesgue measure zero in Δ^{m-1}.*

Proof. Injectivity fails if and only if there exist distinct $x, x' \in \mathcal{X}$ satisfying $\sum_{h=1}^{m} \alpha_h (x_h - x'_h) = 0$. Setting $\delta_h = x_h - x'_h \in \{-(V-1), \ldots, V-1\}$ (with at least one $\delta_h \neq 0$ since $x \neq x'$), each such pair defines the hyperplane

$$H_{x,x'} \;=\; \Big\{ \alpha \in \mathbb{R}^m : \textstyle\sum_{h=1}^{m} \alpha_h \delta_h = 0 \Big\}$$

through the origin. Because $|\mathcal{X}|^2$ is finite, the non-injective set is contained in a finite union of such hyperplanes. The intersection of each hyperplane with the $(m-1)$-dimensional simplex Δ^{m-1} has Lebesgue measure zero, so their finite union has measure zero as well. Hence almost every $\alpha \in \Delta^{m-1}$ makes f_α injective on $\mathcal{X}$. □

In the attention setting, Lemma E.1 guarantees that almost any softmax distribution α over the m target positions yields a scalar superposition taking V^m distinct noiseless values $v_0 < v_1 < \cdots < v_{V^m-1}$ with minimum gap $\delta = \min_j (v_{j+1} - v_j) > 0$.

Lemma E.2 (Impossibility of affine decoding)**.** *Let $n \geq 3$ and $r \geq 1$ with $r + 1 < n$. Let $\mathcal{Z} = \{z_1, \ldots, z_n\} \subset \mathbb{R}^r$ be any n distinct points, and let $\mathcal{V} = \{e_1, \ldots, e_n\} \subset \mathbb{R}^n$ be the standard basis (mutually orthogonal unit vectors). Then there is no affine map $T \colon \mathbb{R}^r \to \mathbb{R}^n$ that maps $\mathcal{Z}$ bijectively onto $\mathcal{V}$.*

Proof. An affine map cannot increase affine dimension, so

$$\dim\big(\mathrm{aff}(T(\mathcal{Z}))\big) \;\leq\; \dim\big(\mathrm{aff}(\mathcal{Z})\big) \;\leq\; r.$$

The n standard basis vectors $e_1, \ldots, e_n$ are mutually orthogonal and hence affinely independent; their affine hull is an $(n-1)$-dimensional simplex in $\mathbb{R}^n$. A flat of affine dimension at most r can contain at most $r + 1$ vertices of an $(n-1)$-dimensional simplex. Since $r + 1 < n$ by assumption, $T(\mathcal{Z})$ cannot contain all n standard basis vectors, contradicting bijectivity onto $\mathcal{V}$. □

Applying Lemma E.2 to the decoding problem: the scalar superposition z_γ lies in $\mathbb{R}^r$ with $r = 1$, and we need to recover $n = V^m$ mutually orthogonal target vectors. The condition $r + 1 = 2 < V^m$ holds whenever $m \geq 1$, $V \geq 3$, or $m \geq 2$, $V \geq 2$. In either regime, the Layer 1 MLP *cannot* be replaced by a linear projection; a non-linear activation is necessary. Definition E.3 below formalizes the interface the MLP must satisfy, and Lemma E.4 constructs it explicitly.

Definition E.3 (Valid MLP decoder)**.** Let $n \geq 1$, let $v_0 < v_1 < \cdots < v_{n-1}$ be n distinct real values with minimum gap

$$\delta \;=\; \min_{j=0,\ldots,n-2} (v_{j+1} - v_j) \;>\; 0,$$

and let $\varepsilon < \delta/4$. A *valid MLP decoder* is a function $\phi \colon \mathbb{R} \to \mathbb{R}^n$ satisfying:

1. **Correct activation:** for each $j \in \{0, \ldots, n-1\}$, if $|z - v_j| \leq \varepsilon$ then $\phi(z) = e_j$.
2. **Suppression:** if $|z - v_j| > \delta/4$ for all j, then $\phi(z) = \mathbf{0}$.

Because $\varepsilon < \delta/4 < \delta/2$, any input within ε of v_j is at distance greater than $\delta - \varepsilon > 3\delta/4$ from every other $v_{j'}$; in particular it cannot be close to two distinct target values simultaneously. The two conditions are therefore consistent, with a "don't-care" band of width $\delta/4 - \varepsilon$ between the activation and suppression regions around each target value.

In the context of Layer 1, we instantiate Definition E.3 with $n = V^m$, target values $\{v_j\}$ from Lemma E.1, and noise tolerance $\varepsilon = \varepsilon_{\mathrm{attn}}$ (bounded by Eq. (56) in the proof of Theorem F.1). Lemma E.4 below constructs an explicit realization of ϕ with hidden dimension $4n = 4V^m$.

Lemma E.4 (Exact pattern encoding via ReLU MLP)**.** *Let $v_0 < v_1 < \cdots < v_{n-1}$ be n distinct real values with minimum gap $\delta > 0$, and let $\varepsilon < \delta/4$. There exists a 2-layer ReLU MLP with hidden dimension $4n$ that realizes a valid MLP decoder ϕ in the sense of Definition E.3.*

Proof. For each $j \in \{0, \ldots, n-1\}$, set $\eta = \delta/4$ and define the trapezoid function

$$b_j(z) \;=\; \frac{1}{\eta}\big[\sigma(z - v_j + 2\eta) - \sigma(z - v_j + \eta) - \sigma(z - v_j - \eta) + \sigma(z - v_j - 2\eta)\big],$$

where $\sigma = \max(0, \cdot)$ is the ReLU. Each b_j uses 4 neurons and satisfies

$$b_j(z) = 1 \text{ when } |z - v_j| < \eta, \qquad b_j(z) = 0 \text{ when } |z - v_j| > 2\eta.$$

Correct activation. For the correct index j: $|z - v_j| \le \varepsilon < \delta/4 = \eta$, so $b_j(z) = 1$.

Suppression of all other indices. For $j' \neq j$: the minimum gap gives $|z - v_{j'}| \ge \delta - \varepsilon > \delta - \delta/4 = 3\delta/4 = 3\eta > 2\eta$, so $b_{j'}(z) = 0$.

The first layer therefore computes the n-dimensional indicator $\mathbf{1}[j]$ using $4n$ neurons in total. The second layer is the fixed $n \times n$ identity projection, mapping $\mathbf{1}[j]$ to $e_j \in \mathbb{R}^n$. Hence $\phi(z) = e_j$ whenever $|z - v_j| \le \varepsilon$ and $\phi(z) = \mathbf{0}$ whenever $|z - v_j| > 2\eta$ for all j, satisfying both conditions of Definition E.3. □

In the Layer 1 architecture, each of the h routing heads contributes one scalar z_γ and one copy of the MLP above (with $n = V^{|S_\gamma|}$), for a total first-layer hidden dimension of $4hV^m$. The concatenated outputs $[e_{j_1}; \ldots; e_{j_h}] \in \mathbb{R}^{hV^m}$ are written into block B_3 of the hidden state and serve as the exact one-hot neighborhood encodings consumed by the Layer 2 retrieval head.

Remark E.5. The $O(V^m)$ hidden-dimension bound is tight for 2-layer ReLU MLPs: distinguishing V^m configurations via scalar superpositions requires V^m distinct indicator functions. Deeper MLPs can represent the same V^m-way classification with exponentially smaller width via standard depth-width tradeoffs [30, 15, 25], but our construction uses a 2-layer MLP to match the overall 2-layer transformer architecture. The $O(V^m)$ width requirement is therefore a consequence of the depth constraint, not an intrinsic property of the spatial routing problem.

E.2 Content-Based Retrieval

After Layer 1, each token $(t', \mathbf{i}')$ in the context carries two pieces of information:

- a *pattern encoding* $q_{t',\mathbf{i}'}$, the concatenation of h vectors encoding the local neighborhood configuration, one per spatial partition. In the partitioned regime ($m > 1$), these are exact one-hot vectors produced by the Layer 1 MLP. In the distributed regime ($m = 1$), these are noisy one-hot vectors produced directly by the Layer 1 value projection; and
- a *state embedding* $e(x_{t',\mathbf{i}'}) \in \{0, 1\}^V$, the one-hot representation of the cell's own discrete state.

In the full architecture (Appendix F, proof of Theorem F.1), $q_{t',\mathbf{i}'}$ occupies block B_3 and the retrieved output is written to block B_4 of the hidden representation defined in equation (55); we use abstract notation here to keep the statements self-contained.

The goal of Layer 2 is to find context tokens whose pattern encoding $q_{t',\mathbf{i}'}$ matches the query token's pattern encoding $q_{t,\mathbf{i}}$, and read off the associated state $e(x_{t',\mathbf{i}'})$.

Lemma E.6 establishes the noise tolerance required for inner-product matching to succeed in the distributed regime. Definition E.8 then formalizes the retrieval interface in terms of an abstract score gap $G > 0$, covering both the partitioned and distributed regimes uniformly; Lemma E.9 shows it is realized by a single attention head; and Corollary E.10 establishes that the resulting approximation error cannot cross any $\arg\max$ decision boundary.

Lemma E.6 (Noise tolerance for inner-product matching)**.** *Let $m \ge 1$, $V \ge 2$, and $\bar{\epsilon} \in [0, 1)$. For $q \in \mathbb{R}^{mV}$, write $q = (q_1, ..., q_m)$ as the concatenation of m blocks $q_h \in \mathbb{R}^V$, and call q ideal if each block is a one-hot vector in $\{0, 1\}^V$. For ideal q, q' and perturbations ϵ, ϵ' whose ℓ_1 mass on non-target coordinates is at most $\bar{\epsilon}$ per block,*

$$\langle q + \epsilon, q' + \epsilon' \rangle \begin{cases} \ge m - 2m\bar{\epsilon} & if\ q = q', \\ \le (m - 1) + 2m\bar{\epsilon} & if\ q \neq q'. \end{cases} \tag{53}$$

The matching and non-matching ranges are strictly separated if and only if $\bar{\epsilon} < \frac{1}{4m}$.

Proof. Let the noise on each block satisfy $||\epsilon_h||_1, ||\epsilon'_h||_1 \le \bar{\epsilon}$ on non-target coordinates.

Matching case. Since each block of q is one-hot, $\langle q, q\rangle = m$. The cross terms $\langle q, \epsilon'\rangle$ and $\langle \epsilon, q\rangle$ each contribute at most $m\bar{\epsilon}$ in absolute value, yielding $\langle q+\epsilon, q+\epsilon'\rangle \geq m - 2m\bar{\epsilon}$.

Non-matching case. If $q \neq q'$, at least one block satisfies $q_h \neq q'_h$, contributing $\langle q_h, q'_h\rangle = 0$. The remaining $m-1$ blocks contribute at most 1 each, and the noise cross terms add up to at most $2m\bar{\epsilon}$, giving $\langle q+\epsilon, q'+\epsilon'\rangle \leq (m-1) + 2m\bar{\epsilon}$.

Separation. Strict separation requires $m - 2m\bar{\epsilon} > (m-1) + 2m\bar{\epsilon}$, equivalent to $\bar{\epsilon} < \frac{1}{4m}$. Both bounds are tight (achieved by appropriate choices of ϵ, ϵ' and non-matching configurations), so the condition is necessary as well. □

Remark E.7 (Gap computation for the distributed regime). In the distributed regime ($m = 1$), the pre-softmax score is $c\langle q+\epsilon, q'+\epsilon'\rangle$ where q, q' are ideal one-hot concatenations of k blocks and ϵ, ϵ' are noise perturbations with per-block leakage $\bar{\epsilon}$. By Lemma E.6, matching patterns score at least $c(k - 2k\bar{\epsilon})$ and non-matching patterns score at most $c((k-1) + 2k\bar{\epsilon})$, yielding a score gap of $G = c(1 - 4k\bar{\epsilon})$ in the sense of Definition E.8. This gap is positive whenever $\bar{\epsilon} < 1/(4k)$, and Lemma E.9 then applies with this reduced gap.

Definition E.8 (Valid retrieval head)**.** Let $k \geq 1$, $N_{\text{seq}} > k$, and $G > 0$ be a *score gap*. Suppose we are given a query token and a sequence of N_{seq} context tokens, each carrying a *value vector* $u_i \in \{0,1\}^V$ for $i = 1, \ldots, N_{\text{seq}}$. Let $\{r_i\}_{i=1}^{N_{\text{seq}}}$ denote the pre-softmax attention scores of the query against the context. A *valid retrieval head* computes the softmax-weighted average

$$\hat{y} = \sum_{i=1}^{N_{\text{seq}}} \frac{e^{r_i}}{\sum_{j=1}^{N_{\text{seq}}} e^{r_j}}\, u_i,$$

satisfying:

1. **Score gap:** exactly k context tokens are *matching* and the remaining $N_{\text{seq}} - k$ are *non-matching*. Every matching token's score exceeds every non-matching token's score by at least G: that is, $r_i - r_j \geq G$ for all matching i and non-matching j.
2. **Bounded retrieval error:** if all k matching tokens share the same value vector $u_i = u^*$, then
$$\left\|\hat{y} - u^*\right\| \;\leq\; 2\,\varepsilon_{\text{ret}}, \qquad \varepsilon_{\text{ret}} \;:=\; \frac{(N_{\text{seq}} - k)\, e^{-G}}{k + (N_{\text{seq}} - k)\, e^{-G}}.$$

Instantiating the score gap. The gap G is the only quantity that matters for retrieval accuracy; its specific value depends on the routing regime.

- **Partitioned regime ($m > 1$, exact encodings):** Setting $r_i = c\,\langle q_0, q_i\rangle$ where $q_0, q_i \in \{0,1\}^{hV^m}$ are concatenations of h one-hot blocks, the inner product counts the number of agreeing blocks. Matching tokens ($q_i = q_0$) score ch; non-matching tokens score at most $c(h-1)$, yielding $G = c$.
- **Distributed regime ($m = 1$, noisy encodings):** Setting $r_i = c\,\langle q_0 + \epsilon_0, q_i + \epsilon_i\rangle$ where ϵ terms are noise perturbations with per-block leakage $\bar{\epsilon}$, Lemma E.6 and Remark E.7 give $G = c(1 - 4k\bar{\epsilon})$, which is positive when $\bar{\epsilon} < 1/(4k)$.

Connection to the architecture. Setting $W_Q^{(2)} = c \cdot \Pi_{B_{3,Q}}$, $W_K^{(2)} = \Pi_{B_{3,K}}$, and $W_V^{(2)} = \Pi_{B_1}$ in Layer 2 realizes condition 1 in both regimes with the appropriate gap G. Let $f : S^k \to S$ be the deterministic local rule from Definition 3.1. Because f is deterministic, every context token whose pattern encoding matches the query (i.e. $\mathcal{N}_{t',\mathbf{i}'} = \mathcal{N}_{t,\mathbf{i}}$) shares the same value vector $v^* = e(f(\mathcal{N}_{t,\mathbf{i}}))$, so condition 2 applies with k equal to the number of such context tokens. Causal masking ensures all retrieved context tokens correspond to completed time steps.

Lemma E.9 (Retrieval via score-gap matching)**.** *Under the setting of Definition E.8 with score gap $G > 0$, the bounded retrieval error condition holds with*

$$\varepsilon_{\text{ret}} \;\leq\; \frac{(N_{\text{seq}} - k)\, e^{-G}}{k + (N_{\text{seq}} - k)\, e^{-G}}.$$

Proof. Let $r^+_{\min}$ denote the minimum score among the k matching tokens. By condition 1, every non-matching token scores at most $r^+_{\min} - G$. The total softmax weight on all matching tokens is therefore

$$w_{\text{match}} \;\geq\; \frac{k\, e^{r^+_{\min}}}{k\, e^{r^+_{\min}} + (N_{\text{seq}} - k)\, e^{r^+_{\min} - G}} \;=\; \frac{k}{k + (N_{\text{seq}} - k)\, e^{-G}}.$$

All matching tokens carry the same value vector v^*, so the attention output satisfies $\hat{y} = w_{\text{match}}\, v^* + (1 - w_{\text{match}})\, r$ for some residual r with $\|r\| \leq 1$. Hence

$$\|\hat{y} - v^*\| \;=\; (1 - w_{\text{match}})\, \|r - v^*\| \;\leq\; 2\,(1 - w_{\text{match}}) \;\leq\; \frac{2(N_{\text{seq}} - k)\, e^{-G}}{k + (N_{\text{seq}} - k)\, e^{-G}} \;=\; 2\,\varepsilon_{\text{ret}},$$

where the first inequality uses $\|r - v^*\| \leq \|r\| + \|v^*\| \leq 2$ since both vectors have norm at most 1. No MLP is required in Layer 2. □

Corollary E.10 (Exact $\arg\max$ classification)**.** *If $\varepsilon_{\text{ret}} < \sqrt{2}/4$, then $\arg\max(\hat{y}) = \arg\max(v^*)$; that is, the predicted cell state is exactly correct.*

Proof. The V standard basis vectors in $\mathbb{R}^V$ are pairwise at Euclidean distance $\sqrt{2}$. The decision boundary between any two of them lies at distance $\sqrt{2}/2$ from each. By Lemma E.9, $\|\hat{y} - v^*\| \leq 2\varepsilon_{\text{ret}}$. The condition $2\varepsilon_{\text{ret}} < \sqrt{2}/2$, which holds when $\varepsilon_{\text{ret}} < \sqrt{2}/4$, ensures that $\hat{y}$ remains strictly closer to v^* than to any other basis vector, so the $\arg\max$ is correct. □

Controlling ε_{ret} via G. The leakage ε_{ret} is strictly decreasing in the score gap $G > 0$ and satisfies $\varepsilon_{\text{ret}} \to 0$ as $G \to \infty$. The condition $\varepsilon_{\text{ret}} < \sqrt{2}/4$ is therefore achieved by choosing

$$G \;>\; \log\!\Big(\tfrac{2\sqrt{2}(N_{\text{seq}} - k)}{k}\Big),$$

which is finite for any $N_{\text{seq}} < \infty$. Since G can be made arbitrarily large by increasing the query scaling constant c (and, in the distributed regime, by decreasing $\bar{\epsilon}$ via a larger margin Δ), this threshold is always achievable.

F Proof of Main Theorem

Theorem F.1 (Spatial In-Context Learning)**.** *For any local dynamical system defined over a d-dimensional grid with axis sizes $L_1, \ldots, L_d \geq 2$, neighborhood size k, state space cardinality $V \geq 2$, and context window of $N_{\text{seq}} > k$ tokens, let $\varepsilon > 0$ be an error tolerance and let $h \in \{1, \ldots, |\mathcal{U}|\}$ be the number of spatial routing heads in Layer 1. Let the maximum spatial partition size be $m = \lceil |\mathcal{U}|/h \rceil$. There exists a 2-layer causal transformer that exactly predicts the system's evolution, with the following architectural constraints:*

- ***Layer 1 Routing:*** *h attention heads operating on spatial-aware embeddings. If $m = 1$, the required embedding dimension is exactly $d_p = 2d + 2$. If $m > 1$, the dimension is bounded by $d_p \leq 2(m + d) + 2$.*
- ***Layer 1 Decoding:*** *A 2-layer ReLU MLP to process spatial combinations. If $m = 1$, the MLP is not necessary (the routing is purely linear). If $m > 1$, the first layer requires a hidden dimension of exactly $4hV^m$ neurons; the second layer projects to an output dimension of $2hV^m$, writing two concatenated marginalized one-hot vectors per head into sub-blocks $B_{3,K}$ and $B_{3,Q}$.*
- ***Layer 2 Retrieval:*** *1 attention head using linear Hamming matching, requiring no MLP.*
- ***Total Dimension:*** *The required representation dimension D is independent of the macroscopic grid volume L. Specifically:*

$$D \leq \begin{cases} 2(h+1)V + 2d + 2 & \text{if } m = 1 \text{ (Fully Distributed)} \\ 2hV^m + 2(m + V + d) + 2 & \text{if } m > 1 \text{ (Partitioned)} \end{cases} \tag{54}$$

Proof Roadmap. The proof assembles three modular stages whose supporting lemmas are proved in Appendices D and E.1.

Step 1. Spatial Routing (Layer 1 Attention). Each of the h attention heads isolates a spatial partition of size $m = \lceil |\mathcal{U}|/h \rceil$ using the spatial-aware embeddings of Theorem D.5 (constructed via the trigonometric interpolation of Lemma D.2). The pre-softmax isolation margin Δ controls the attention leakage $\varepsilon_{\text{attn}}$ (Eq. 56). In the *partitioned* case ($m > 1$), we require $\varepsilon_{\text{attn}} < \delta/4$ to feed the MLP in Step 2; in the *distributed* case ($m = 1$), we instead require $\varepsilon_{\text{attn}} < 1/(4k)$ for the noise tolerance of Lemma E.6.

Step 2. Non-Linear Decoding (Layer 1 MLP; $m > 1$ only). Lemma E.1 guarantees that the weighted superposition from Step 1 takes V^m distinct values with minimum gap $\delta > 0$. Lemma E.2 shows that no affine map can decode these into mutually orthogonal targets, and Lemma E.4 constructs an explicit 2-layer ReLU MLP with $4V^m$ hidden neurons per head that achieves exact decoding whenever $\varepsilon_{\text{attn}} < \delta/4$. When $m = 1$ this step is unnecessary.

Step 3. Content-Based Retrieval (Layer 2 Attention). In both regimes, Steps 1–2 establish a score gap $G > 0$ between matching and non-matching pattern encodings in B_3: $G = c$ for exact encodings (partitioned) or $G = c(1 - 4k\bar{\epsilon})$ for noisy encodings (distributed). A single attention head performs retrieval via Definition E.8. Lemma E.9 bounds the retrieval error ε_{ret} in terms of G, and Corollary E.10 guarantees exact $\arg\max$ classification whenever G exceeds a finite threshold. No Layer 2 MLP is required.

F.1 Proof of Theorem F.1 (Spatial In-Context Learning)

We use the dynamical environment notation (V, k, d, W, $\mathcal{N}_{t,\mathbf{i}}, \mathcal{N}_K, \mathcal{N}_Q$) from Section 3 and the spatial partitions S_γ with maximum size $m = \lceil |\mathcal{U}|/h \rceil$ from Definition 4.1. We let N_{seq} denote the total context window length. With the mechanics of routing, decoding, and retrieval established, we now synthesize these components into the complete two-layer architecture. This proof proceeds as a modular roadmap, invoking supporting lemmas whose statements and proofs are deferred to the preceding appendices.

Remark F.2 (Layer normalization Simplification.). Layer normalization is applied before attention in practice, introducing per-token scaling factors into the attention scores. Our constructions are provided for Transformers without layer normalization. Operating in this simplified regime is reasonable because we could scale the positional encoding $\mathbf{p}_{t,i}$ by a sufficiently large constant γ, so that $\|\mathbf{h}_{t,i}\|_2 \approx \gamma \|\mathbf{p}_{t,i}\|_2$ for all tokens. Since the non-positional components of $\mathbf{h}_{t,i}$ are $O(1)$-bounded, the per-token norm variations become $O(1/\gamma)$-negligible, and layer normalization reduces to a near-uniform scaling that does not affect the constructions. This technique follows **?**].

Proof. **Step 0: Hidden Representation.** We begin by defining the initial hidden representation $\mathbf{h}_{t,\mathbf{i}}^{(0)} \in \mathbb{R}^D$ for each token $(t, \mathbf{i})$. We partition the representation into functional blocks:

$$\mathbf{h}_{t,\mathbf{i}}^{(0)} = \Bigg[\underbrace{\mathbf{e}(x_{t,\mathbf{i}})}_{\text{B1:}\mathbb{R}^V}; \underbrace{\mathbf{p}_{t,\mathbf{i}}}_{\text{B2:}\mathbb{R}^{d_p}}; \underbrace{\mathbf{0}}_{\text{B3:}\mathbb{R}^{2h \cdot V^m}}; \underbrace{\mathbf{0}}_{\text{B4:}\mathbb{R}^V} \Bigg] \tag{55}$$

where $\mathbf{e}(v)$ is the standard one-hot basis vector for the cell state, and $\mathbf{p}_{t,\mathbf{i}}$ is the spatial-aware embedding of dimension d_p.

Each block serves a distinct functional role in the two-layer circuit:

- B_1 stores the one-hot state embedding of each cell, which Layer 2 reads as its value projection to retrieve the output associated with a matched neighborhood.
- B_2 stores the positional embedding consumed by the Layer 1 query and key projections for spatial routing.
- B_3 is the intermediate scratch space into which Layer 1 writes the decoded neighborhood identities. To satisfy the disjoint autoregressive query and key constraints, we partition

this block into two concatenated sub-blocks: $B_{3,K}$ (for $\mathcal{N}_K$) and $B_{3,Q}$ (for $\mathcal{N}_Q$). It is dimensioned to accommodate h independent writes of maximum size V^m per sub-block, one per routing head: after the Layer 1 MLP and marginalization projection, the γ-th sub-block of $B_{3,K}$ contains a one-hot vector $e_{\gamma,K} \in \mathbb{R}^{V^{|S_\gamma \cap \mathcal{N}_K|}}$ and the γ-th sub-block of $B_{3,Q}$ contains a one-hot vector $e_{\gamma,Q} \in \mathbb{R}^{V^{|S_\gamma \cap \mathcal{N}_Q|}}$ indicating which of the up to V^m possible local state configurations was identified by head γ for each respective neighborhood. Layer 2 then reads $B_{3,Q}$ as its query and $B_{3,K}$ as its key to perform content-based pattern matching.

- B_4 is the output slot into which Layer 2 writes its retrieved result: a (possibly noisy) one-hot vector $\hat{y} \approx e(f(\mathcal{N}_{t,\mathbf{i}})) \in \mathbb{R}^V$ encoding the predicted next state.

Step 1: Partitioned Spatial Routing (Layer 1 Attention). We distribute the $|\mathcal{U}|$ distinct relative spatial offsets of the union routing set $\mathcal{U}$ across h mutually exclusive subsets $S_1, \ldots, S_h$ as defined in Definition 4.1, with $|S_\gamma| \le m = \lceil |\mathcal{U}|/h \rceil$ for all γ. Each head γ is responsible for attending to the cells at positions $(t-1, \mathbf{i} + \delta_j)_{j \in S_\gamma}$ (the *target parent cells* of head γ) in the causally visible preceding time step.

For each head γ, we construct spatial query and key projections using the spatial-aware embeddings of Theorem D.5, which guarantee a pre-softmax isolation margin $\Delta > 0$: every non-target position $\ell \notin S_\gamma$ receives score $s_\ell \le s_{\min} - \Delta$, where $s_{\min} = \min_{j \in S_\gamma} s_j$.

Leakage bound. Let $\varepsilon_{\text{attn}} = 1 - \sum_{j \in S_\gamma} \alpha_j$ be the total softmax weight on non-target positions. Bounding the denominator of each α_j:

$$\sum_{i=1}^{N_{\text{seq}}} e^{s_i} \;\le\; \underbrace{m\, e^{s_{\min}}}_{\text{target upper bound}} \;+\; \underbrace{(N_{\text{seq}} - m)\, e^{s_{\min} - \Delta}}_{\text{non-target upper bound}} \;=\; e^{s_{\min}} \big[m + (N_{\text{seq}} - m)\, e^{-\Delta} \big].$$

Since each target score $s_j \ge s_{\min}$, summing over S_γ:

$$\sum_{j \in S_\gamma} \alpha_j \;\ge\; \frac{m\, e^{s_{\min}}}{e^{s_{\min}} [m + (N_{\text{seq}} - m)\, e^{-\Delta}]} \;=\; \frac{m\, e^{\Delta}}{m\, e^{\Delta} + (N_{\text{seq}} - m)},$$

so the leakage satisfies

$$\varepsilon_{\text{attn}} \;\le\; \frac{N_{\text{seq}} - m}{m\, e^{\Delta} + (N_{\text{seq}} - m)}. \tag{56}$$

Step 2 requires $\varepsilon_{\text{attn}} < \delta/4$, where $\delta = \min_j (v_{j+1} - v_j)$ is the minimum gap between distinct noiseless superposition values (Lemma E.1). Solving (56) $< \delta/4$ for Δ:

$$\frac{N_{\text{seq}} - m}{m\, e^{\Delta} + (N_{\text{seq}} - m)} < \frac{\delta}{4} \quad \Longleftarrow \quad \Delta \;>\; \log\!\left(\frac{4(N_{\text{seq}} - m)}{m\, \delta} \right).$$

Since $w_s = s/(V-1) \in [0,1]$ and $\alpha \in \Delta^{m-1}$, the noiseless superposition satisfies $z^*_\gamma \in [0,1]$, so the V^m distinct values from Lemma E.1 have minimum gap $\delta \le 1/(V^m - 1) \le 1$; the factor $(4-\delta)/\delta$ in the exact threshold is bounded above by $4/\delta$, making the above a valid sufficient condition. Since δ, m, and N_{seq} are all finite, the threshold is finite, and since Δ is a free parameter of Theorem D.5 it can always be set to satisfy it.

Using the normalized value projection $W_V e(s) = w_s = s/(V-1) \in [0,1]$ (as defined in Lemma E.4), the value projection aggregates the target cells into the scalar superposition:

$$z_\gamma = \sum_{j \in S_\gamma} \alpha_j w_{x_{t,j}} + \eta_\gamma, \qquad |\eta_\gamma| \le \varepsilon_{\text{attn}},$$

where $\eta_\gamma = \sum_{j \notin S_\gamma} \alpha_j w_{x_{t,j}}$ is the leakage from non-target positions, bounded by $\varepsilon_{\text{attn}}$ since $w_s \le 1$ for all s. By Lemma E.1, the noiseless value $z^*_\gamma = \sum_{j \in S_\gamma} \alpha_j w_{x_{t,j}}$ is one of V^m distinct values $\{v_j\}$ with minimum gap $\delta = \min_j (v_{j+1} - v_j) > 0$. Therefore the noisy superposition satisfies $|z_\gamma - v_j| \le \varepsilon_{\text{attn}}$ for the correct configuration j, which is precisely the condition required by Lemma E.4 in Step 2. **Distributed case** ($m = 1$). When $h = |\mathcal{U}|$, each head routes a single cell and Step 2 is mathematically unnecessary. The layer 1 linear value projection $W_V^{(1)}$ directly maps the

cell state $e(x_{t,j})$ to the corresponding coordinate in $B_{3,K}$ (if $j \in \mathcal{N}_K$), $B_{3,Q}$ (if $j \in \mathcal{N}_Q$), or both. Because the attention weight on the target cell is $1 - \varepsilon_{attn}$ rather than exactly 1, the actual write to B_3 is a noisy perturbation, meaning the B_3 vectors are noisy one-hot representations with per-head leakage $\bar{\epsilon} \leq \varepsilon_{attn}$. Layer 2 subsequently extracts and matches exactly k cells from $B_{3,Q}$ against $B_{3,K}$. For Layer 2 to correctly distinguish matching from non-matching neighborhood patterns over these k alignments despite the noise, Lemma E.6 requires $\bar{\varepsilon} < \frac{1}{4k}$. Substituting into the leakage bound (56) with $m = 1$ yields:

$$\frac{N_{seq} - 1}{e^{\Delta} + N_{seq} - 1} < \frac{1}{4k} \implies \Delta > \log((4k-1)(N_{seq} - 1)) \tag{57}$$

Since Δ is a free parameter of Theorem D.5, this threshold is always achievable.

Step 2: Non-Linear Sub-Configuration Decoding (Layer 1 MLP). Following the spatial partition outlined in Definition 4.1, unless $h = |\mathcal{U}|$, the spatial features within each partition are algebraically entangled in the superposition z_γ. By Lemma E.1, there exist attention weights such that all $V^{|S_\gamma|}$ configurations produce distinct values $\{v_j\}$ with minimum gap $\delta > 0$, establishing the condition required by Lemma E.1. We independently apply Lemma E.4 to each head's output z_γ. The first layer of the required 2-layer ReLU MLP constructs a trapezoid function centered at each distinct target value, requiring exactly $4V^{|S_\gamma|}$ hidden neurons per head. Across all h heads, the total Layer 1 MLP hidden dimension is bounded by $4hV^m$. This first layer decodes the scalar into a single internal one-hot vector representing the exact joint configuration of the cells in S_γ.

To isolate the distinct neighborhoods required for autoregressive alignment, the second linear layer of the MLP acts as a marginalization projection via binary incidence matrices. Rather than projecting to a single basis, it maps the internal joint one-hot vector into two separate marginal one-hot vectors: the first represents the state of $S_\gamma \cap \mathcal{N}_K$ and is written to the γ-th sub-block of $B_{3,K}$; the second represents the state of $S_\gamma \cap \mathcal{N}_Q$ and is written to the γ-th sub-block of $B_{3,Q}$. Across all h heads, this projects the result as two exact, concatenated one-hot vectors $c_K, c_Q \in \mathbb{R}^{h \cdot V^m}$ into $B_{3,K}$ and $B_{3,Q}$, respectively. This continuous parameterization recovers the isolated mechanisms:

- **Bottlenecked Regime ($h = 1$):** The partition size is $m = |\mathcal{U}|$. Step 2 requires an MLP width of $4V^{|\mathcal{U}|}$, recovering the exact non-linear decoding mechanism necessary to disentangle the maximum spatial overlap.
- **Parallel Regime ($h = |\mathcal{U}|$):** The partition size is $m = 1$. Each head's scalar output z_γ directly encodes a single cell state x_γ, and no non-linear MLP is required (0 hidden neurons). The marginalization into $B_{3,K}$ and $B_{3,Q}$ is achieved purely via the linear value projection. The residual attention leakage is controlled by the margin condition (57) established in Step 1, which ensures Layer 2 pattern separability via Lemma E.6.

Step 3: Content-Based Retrieval (Layer 2 Attention). After Steps 1–2, the pattern encodings in $B_{3,Q}$ and $B_{3,K}$ are available (exact in the partitioned regime, noisy in the distributed regime), and the state embedding $e(x_{t',\mathbf{i}'})$ is available from B_1. We set $W_Q^{(2)} = c \cdot \Pi_{B_{3,Q}}$, $W_K^{(2)} = \Pi_{B_{3,K}}$, and $W_V^{(2)} = \Pi_{B_1}$.

Because the rule f is deterministic, every context token satisfying $\mathcal{N}_{K,t',\mathbf{i}'} = \mathcal{N}_{Q,t,\mathbf{i}}$ carries the identical value vector $e(f(\mathcal{N}_{t,\mathbf{i}}))$, and causal masking ensures all such tokens correspond to completed time steps. In both regimes, matching tokens score higher than non-matching tokens by a gap $G > 0$:

- **Partitioned case ($m > 1$):** The MLP produces exact one-hot encodings, so $c\langle q^{(Q)}, q^{(K)} \rangle$ equals ch for matching and at most $c(h-1)$ for non-matching, giving $G = c$.
- **Distributed case ($m = 1$):** The encodings are noisy one-hot vectors with per-block leakage $\bar{\epsilon} \leq \varepsilon_{\text{attn}}$. By Lemma E.6 and Remark E.7, the score gap is $G = c(1 - 4k\bar{\epsilon})$, which is positive by the margin condition (57).

By Lemma E.9 applied with gap G, the Layer 2 output $\hat{y}$ written to B_4 satisfies

$$||\hat{y} - e(f(\mathcal{N}_{t,\mathbf{i}}))|| \leq 2\varepsilon_{ret}, \quad \varepsilon_{ret} \leq \frac{(N_{seq} - k)\, e^{-G}}{k + (N_{seq} - k)e^{-G}}. \tag{58}$$

No MLP is required in Layer 2.

Error budget. We verify that Δ and c can always be chosen to satisfy the error tolerance $\varepsilon > 0$.

Layer 1 (Δ). By equation (56), the attention leakage satisfies

$$\varepsilon_{\mathrm{attn}} \;\le\; \frac{N_{\mathrm{seq}} - m}{m\, e^{\Delta} + N_{\mathrm{seq}} - m},$$

which is decreasing in Δ. Two regimes arise based on the spatial partition of $\mathcal{U}$:

- **Partitioned case ($m > 1$):** The condition $\varepsilon_{\mathrm{attn}} < \delta/4$ required by Lemma E.4 is met whenever $\Delta > \log\!\big(\frac{4(N_{\mathrm{seq}}-m)}{m\delta}\big)$. Under this condition the MLP output is exactly e_j for each head γ, so the marginalized encodings $q_{t,\mathbf{i}}$ in $B_{3,Q}$ and $B_{3,K}$ are exact and contribute zero error to Step 3.
- **Distributed case ($m = 1$):** No MLP is applied, so B_3 carries noisy one-hot vectors. Lemma E.6 requires per-head leakage $\bar{\epsilon} < \frac{1}{4k}$ for Layer 2 separability. which is met whenever $\Delta > \log\big((4k-1)(N_{\mathrm{seq}}-1)\big)$ (equation (57)).

In both cases, Δ is a free parameter of Theorem D.5, so the required threshold is always achievable.

Layer 2 (G). With a score gap $G > 0$ established in Step 3, Lemma E.9 bounds the retrieval leakage by

$$\varepsilon_{\mathrm{ret}} \;\le\; \frac{(N_{\mathrm{seq}} - k)\, e^{-G}}{k + (N_{\mathrm{seq}} - k)\, e^{-G}}.$$

In the partitioned case $G = c$; in the distributed case $G = c(1 - 4k\bar{\epsilon})$ where $\bar{\epsilon} < 1/(4k)$ is guaranteed by the margin condition on Δ. For the general ε-error bound, $2\varepsilon_{\mathrm{ret}} \le \varepsilon$ is achieved whenever

$$G \;>\; \log\!\Big(\tfrac{2(N_{\mathrm{seq}}-k)}{k\varepsilon}\Big).$$

For exact $\arg\max$ classification (cell-level accuracy), Corollary E.10 requires only $\varepsilon_{\mathrm{ret}} < \sqrt{2}/4$, achieved by the stricter-but-still-finite threshold

$$G \;>\; \log\!\Big(\tfrac{2\sqrt{2}(N_{\mathrm{seq}}-k)}{k}\Big).$$

Since c is a free parameter and $\bar{\epsilon}$ can be made arbitrarily small by increasing Δ, the gap G can always be made large enough to satisfy either threshold, confirming the existence of the claimed transformer. □

G Experimental Details

This appendix provides full hyperparameters and data generation procedures for all experiments reported in Section 5.

G.1 Common Training Configuration

All models are decoder-only GPT-style transformers with Pre-LayerNorm, causal masking, and learned absolute positional embeddings. Each block consists of multi-head attention followed by an MLP (Linear $\to$ GELU $\to$ Linear) with hidden dimension $4D$, both with residual connections and layer normalization. The token embedding and the output projection head are untied.

We optimize with AdamW (β_1=0.9, β_2=0.999) using cosine learning rate decay with linear warmup. Gradients are clipped to a global norm of 1.0. Training uses `bf16` mixed precision with `torch.compile`. The cross-entropy loss is computed only on cell tokens whose time step index is at least M (the number of context steps); separator tokens are never predicted. All runs use seed 42 for both data generation and model initialization.

G.2 Per-Model Hyperparameters

Table 6 summarizes the architecture and training configuration for each model. We highlight several setting-specific choices below.

1D, $V{=}2$, $k{=}3$ (ECA). Models (a) and (b) train for 500 epochs with batch size 1024 and learning rate 10^{-3}. No dropout or attention temperature annealing is used.

1D, $V{=}3$, $k{=}3$. Model (c) uses a 4-layer architecture with 3 heads per layer. Model (d) uses the same 2-layer, 3+1 configuration as the ECA setting but trains for 300 epochs.

2D Von Neumann, $k{=}5$. The 2D setting requires several additional techniques. Model (e) uses dropout of 0.2 on residual connections, attention outputs, and MLP layers. Both models (e) and (f) use attention temperature annealing: a query scaling factor T_q is linearly decreased from 1.0 to 0.1 over the first 80% of training and held constant thereafter, which sharpens attention distributions. Model (g) additionally incorporates a 2D relative positional bias: a learned scalar $B[\Delta t, \Delta r, \Delta c]$ indexed by relative time step, row, and column offsets on the $T{\times}L_1{\times}L_2 = 12{\times}6{\times}6$ grid is added to the pre-softmax attention logits. Pairs involving separator tokens use a separate learned bias $B_{\text{sep}}[\Delta t]$.

Table 6: Hyperparameters for all experimental settings.

	Setting	Layers	Heads	D	Batch	LR	WD	Epochs	Dropout	Anneal
(a)	1D $V{=}2$	2	1,1	512	1024	1e-3	0.2	500	0	–
(b)	1D $V{=}2$	2	3,1	384	1024	1e-3	0.2	500	0	–
(c)	1D $V{=}3$	4	3×4	576	512	1e-3	0.2	500	0	–
(d)	1D $V{=}3$	2	3,1	384	1024	1e-3	0.2	300	0	–
(e)	2D VN	4	5×4	640	512	3e-3	0.2	500	0.2	1.0→0.1
(f)	2D VN	2	5,1	640	512	3e-3	0.2	1000	0	1.0→0.1
(g)	2D VN+RPE	2	5,1	640	512	3e-3	0.2	1000	0	1.0→0.1

G.3 Data Generation

G.3.1 Tokenization

For 1D settings, each row of L cells is serialized as L cell tokens separated by a single `[SEP]` token between consecutive time steps. The vocabulary size is $V + 1$. For 2D settings, each $L_1 \times L_2$ grid is flattened in row-major order with a `[ROW_SEP]` between rows within one time step and a `[TIME_SEP]` between time steps, yielding a vocabulary of $V + 2$. In all cases, the loss mask covers only cell tokens at time steps $t \geq M$.

G.3.2 1D $V{=}2$ (ECA)

Grid size $L{=}16$, $T{=}10$ time steps, $M{=}4$ context steps. The full ECA rule space (256 rules) is partitioned into equivalence classes under reflection and bit-complement symmetry, with one canonical representative per class. An 80/20 split of canonical rules defines the train and test rule pools. We generate 120k training and 20k test trajectories, each from a uniformly random binary initial row evolved for $T{-}1$ steps. Samples are rejected unless the first $M{-}1$ rows expose all $2^3 = 8$ possible 3-bit neighborhood patterns, ensuring in-context observability.

G.3.3 1D $V{=}3$

Grid size $L{=}32$, $T{=}12$ steps, $M{=}8$ context steps. The rule space ($3^{27} \approx 7.6 \times 10^{12}$) is sampled rather than enumerated. Rules are drawn via lambda-stratified sampling: we compute Langton's λ for each candidate rule, partition the λ range into 4 bins, and rejection-sample until each bin is equally represented. Candidate rules are discarded if random initial conditions cannot expose all $V^k = 27$ neighborhood configurations within $M{-}1$ rows. Rules are canonicalized under reflection and state-permutation symmetry (S_V). We use 200 train rules and 50 held-out test rules, generating 120k training and 20k test trajectories with uniform sampling over rules.

G.3.4 2D Von Neumann

Grid size 6×6, $V{=}2$, $k{=}5$ (center plus four cardinal neighbors), $T{=}12$ steps, $M{=}8$ context steps. The rule space is $2^{2^5} = 2^{32}$. Rules are sampled with lambda-balanced sampling across 4 Langton λ bins and canonicalized under the Von Neumann symmetry group (rotations, reflections, bit-complement). We use 200 train rules and 50 test rules, generating 500k training and 20k test

trajectories. Each trajectory receives a fresh random rule from the training pool, providing strong rule augmentation. The total sequence length is $T \cdot (L_1 L_2 + L_1 - 1) + (T - 1) = 503$ tokens.

G.4 Evaluation

We report three metrics on held-out test sets (unseen rules and trajectories): (1) *cell accuracy*, the fraction of individual cells predicted correctly; (2) *sequence accuracy*, the fraction of test sequences where every cell across all prediction steps is correct; and (3) *autoregressive accuracy*, the sequence accuracy when the model generates entire trajectories by feeding its own predictions back as input for $T - M$ steps. Separator tokens are inserted at known positions during autoregressive generation. Autoregressive evaluation is performed on a subset of 100 to 500 test samples.

G.5 Compute Resources

All experiments were conducted on a heterogeneous SLURM cluster using a single GPU per run, primarily NVIDIA RTX A6000 (48 GB), with PyTorch 2.5.1, CUDA 12.1, and bf16 mixed precision. The 1D models each train in under 10 GPU-hours, while the 2D models require 55 to 75 GPU-hours due to longer sequences and more training epochs. The total training cost across all seven models in Table 1 is approximately 207 GPU-hours. Data generation and evaluation costs are negligible by comparison (under 2 hours total on CPU and GPU combined).

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: All claims in the abstract and introduction are supported by formal proofs and experimental results.

   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: The paper acknowledges that 2D models achieve lower accuracy than 1D.

   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: All theorems and lemmas are numbered and cross-referenced. Assumptions are stated in the theorem statements. Proof sketches are provided in the main paper and complete formal proofs are deferred to Appendices C, D, and E.

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: All the information needed to reproduce the experimental results are provided in section 5 and appendix F

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 code is currently hosted in a non-anonymous repository. We have provided detailed implementation details and hyperparameters in Section 5 and Appendix F to facilitate reproduction, and we commit to releasing the complete codebase under an open-source license upon publication.

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: Detailed in section 5 and appendix F

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: [Yes]

Justification: We report performance metrics (Cell Acc, Seq. Acc, Auto. Acc) across 20k test samples to ensure statistical stability (Table 2) and specify exact sequence counts for training and evaluation. We ablate key architectural components across multiple head configurations to verify the consistency of the routing-decoding tradeoff (Table 3).

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: Detailed in appendix F

   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: Our theoretical paper is fully compliant with NeurIPS Code of Ethics.

   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: While future interpretability built on work could lead to broader societal impact, our current contribution does not.

    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: Our work is purelly theoretical on a synthetic mathematical dataset.

    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: [N/A]

    Justification: The paper does not use existing assets.

    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 new assets.

    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 work is purely theoretical

    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 work is purely theoretical

    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: [N/A]

    Justification: Our work is in compliance with NeurIPS LLM policy.

    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.