Invariant Learning Dynamics of Transformers in Inductive Reasoning Tasks

Tiberiu Musat* ETH Zurich
Tiago Pimentel ETH Zurich
Nicholas Zucchet Stanford
Thomas Hofmann ETH Zurich

Abstract

We present a theoretical framework to explain the emergence of inductive reasoning abilities in Transformer language models. While previous works on Transformer learning dynamics have so far been mostly tied to specific tasks, we study a generalized class of inductive tasks that unifies several synthetic tasks known in the literature, including in-context n-grams and multi-hop reasoning. In this class, we theoretically prove that the training dynamics of attention models can be confined to a highly interpretable, low-dimensional invariant manifold. On this manifold, the learning dynamics are captured by a handful of interpretable coordinates rather than millions of parameters, making both theoretical and empirical analysis more tractable. Using this framework, we characterize how data statistics govern the competition between in-context and in-weights learning, we study how random initializations determine the 'winning' circuit when multiple solutions are possible, and we demonstrate that the coordinate frame associated with the manifold can be used to automatically detect which circuits have been learned in trained models. By casting circuit formation as a low-dimensional dynamical phenomenon, we take a step toward a predictive theory of how Transformers learn.

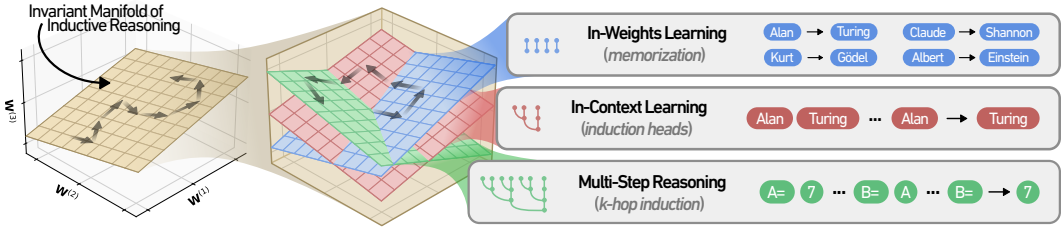

Figure 1: The *Invariant Manifold of Inductive Reasoning* (IMIR) is an interpretable low-dimensional subspace of the parameter space which training trajectories never leave. Each induction circuit resides and evolves within a subspace of the IMIR, depicted schematically as a colored plane.

1 Introduction

While great progress has been made toward understanding the circuits (i.e. subnetworks, or weight structures) present in large language models [1–4], we are still far from understanding the learning dynamics that underlie their formation. As a result, skill acquisition in language models remains mostly unpredictable [5], which complicates model development and poses AI safety issues. A better understanding of circuit emergence could reduce these risks, provide practical insights into how to interpret the internal mechanics of trained models, as well as help design better training pipelines [6]. In this work, we focus on the emergence of circuits responsible for **inductive abilities**: general-purpose

*Correspondence at tiberiu@musat.ai, tiago.pimentel@inf.ethz.ch

Preprint.

capabilities that allow a language model to recognize patterns, infer rules, and perform abstract reasoning on the fly based on the provided context, rather than relying on static, memorized patterns.

Inductive abilities of transformers have been studied extensively by prior work, both empirically and theoretically. However, inductive abilities are typically studied in isolation, limited to specific tasks such as in-context n-grams [7–9] or k-hop induction [10–13]. Without a unifying theoretical framework, our understanding progresses slowly, and three important phenomena remain particularly puzzling. ① **Circuit competition** is a fundamental aspect of language model training [14–16]. However, most previous theoretical works assume staged learning algorithms where only one component is trained at a time, while others are kept frozen [17, 18]. A few works study the dynamical interaction of different components during training, but are limited to small transformers with only one [19] or two [20] attention heads. ② **Data distributional properties** strongly influence the emergence of in-context learning [21–23], but a mechanistic understanding is currently limited to phenomenological models [24]. ③ **Randomly initialized** networks contain high-performing subnetworks that can be trained in isolation, according to the *lottery ticket hypothesis* [25], but we currently lack a mechanistic understanding of why specific initializations yield 'winning' tickets.

Our paper's central contribution is the **Invariant Manifold of Inductive Reasoning (IMIR)**: a low-dimensional subspace of the parameter space which exhibits self-contained learning dynamics. A model whose parameters belong to the IMIR will remain on the IMIR when updated via gradient descent, a behavior known in dynamical systems theory as an *invariant manifold*. We theoretically prove the existence of the IMIR and provide its mathematical description (section 3) for a generalized class of induction tasks (section 2). Importantly, our framework considers all parameter interactions in a model during training, making it more tractable to study phenomena like circuit competition. Moreover, each dimension of the IMIR is highly interpretable, corresponding to a specific elementary action within a specific transformer component. This allows us to find reduced and interpretable descriptions of the transformer circuits responsible for inductive reasoning abilities.

In section 4, we leverage our framework to conduct a circuit competition study (addressing ①). Concretely, we first investigate the circuit competition between in-context learning and in-weights learning (section 4.2), where we theoretically prove the data dependency of in-context learning (addressing ②). Second, we investigate the impact of random initialization on the competition between several in-context learning circuits within a single transformer model; we find that the winning circuit is determined by the model's initialization, with sharp phase transitions between the regions of initialization space favoring each circuit (partially addressing ③ in this simplified setup). In section 5, we show how the highly interpretable structure of the IMIR can be used for automatic detection of inductive circuits. Finally, in section 6, we discuss other possible applications of our framework and future research directions.

Overall, our work provides a theoretical framework for understanding the emergence of inductive circuits and opens the door for future work on the learning dynamics of large language models.

2 A Unified Class of Inductive Tasks

A great number of inductive abilities have been studied in the existing literature [3, 7, 10, 11, 26, 27]. However, models' inductive abilities are usually investigated separately across these tasks, which makes it difficult to arrive at general results. We show that many inductive tasks, while being relatively different at the surface level, all share some intrinsic structure requiring in-context manipulation of token tuples. This will enable us to derive general results that apply to a large variety of tasks. We describe our task class in section 2.1 and we give some examples of induction tasks. Then, we describe how the task depends on token relations and block positions in section 2.2.

2.1 The Block-List Task Class

In natural language next-token prediction tasks, most words depend only on a tiny fraction of their context: the immediately preceding block of words and a few relevant blocks spread throughout its (potentially much larger) context. These relevant blocks may correspond to sentences, clauses, phrases, or even short spans such as consecutive word pairs. Inspired by this observation about the dependency structure in natural language, we now define a general class of **block-list tasks** where a

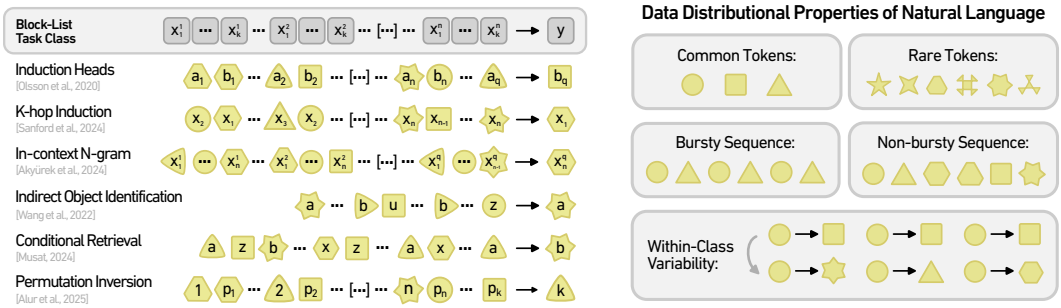

Figure 2: We define a block-list class of language tasks where the next token must be predicted based on a list of token tuples. We then study the training dynamics in this class of tasks.

next token must be predicted based on a list of n blocks of at most k tokens each:

$$[\mathbf{x}_1^1 \dots \mathbf{x}_{k_1}^1]\,[\mathbf{x}_1^2 \dots \mathbf{x}_{k_2}^2]\;\dots\;[\mathbf{x}_1^n \dots \mathbf{x}_{k_n}^n] \;\longrightarrow\; \mathbf{x}_{k_n+1}^n \tag{1}$$

where $[\mathbf{x}_1^n \dots \mathbf{x}_{k_n}^n]$ denotes the block of tokens immediately preceding the token to be predicted. The other blocks $[\mathbf{x}_1^i \dots \mathbf{x}_{k_i}^i]$, with $i < n$, denote potentially (but not necessarily) relevant blocks of tokens. Under our task's definition, each k_i satisfies $k_i \leq k$, and we denote the language vocabulary as $\mathcal{V}$, so that $[\mathbf{x}_1^i \dots \mathbf{x}_{k_i}^i] \in \mathcal{V}^{k_i}$. Many important next-token prediction tasks fit this description, and we now give three examples.

In-context associative recall [17, 18, 26]. In this task, we have sequences with the pairwise structure $[\mathbf{x}_1^1\mathbf{x}_2^1]\,[\mathbf{x}_1^2\mathbf{x}_2^2]\;\dots\;[\mathbf{x}_1^n] \to\; \mathbf{x}_2^n$ and must predict token $\mathbf{x}_2^n$ by finding a previous occurrence[2] $\mathbf{x}_1^i$ of token $\mathbf{x}_1^n$ with $i < n$ in the sequence and copying the token following it (i.e., $\mathbf{x}_2^i$). In our general formulation, we have blocks of two tokens in the context ($k_i = 2$ for $i < n$), while the query token forms a single-token block. Notably, this is one of the most studied inductive tasks.

k-hop induction [10–13]. This task generalizes the in-context associative recall task by considering chained pairs of associations. It has a similar structure to the previous task; however, once $\mathbf{x}_1^i$ is identified in the context, we may use $\mathbf{x}_2^i$ for a second-lookup operation. Formally, we must find a chain of k bigrams such that: $(\mathbf{x}_1^n = \mathbf{x}_1^{i_1}) \to (\mathbf{x}_2^{i_1} = \mathbf{x}_1^{i_2}) \to (\mathbf{x}_2^{i_2} = \mathbf{x}_1^{i_3}) \cdots \to \mathbf{x}_2^{i_k}$.

In-context n-grams [7–9]. A different way to generalize the in-context associative recall task is by considering relations between n-grams of arbitrary length ℓ. In this task, we thus have sequences with the structure $[\mathbf{x}_1^1\mathbf{x}_2^1 \cdots \mathbf{x}_\ell^1]\,[\mathbf{x}_1^2\mathbf{x}_2^2 \cdots \mathbf{x}_\ell^2]\;\dots\;[\mathbf{x}_1^n \cdots \mathbf{x}_{\ell-1}^n] \to\; \mathbf{x}_\ell^n$ and must predict token $\mathbf{x}_\ell^n$ by finding a previous occurrence of its entire prefix $\mathbf{x}_1^n \cdots \mathbf{x}_{\ell-1}^n$ in the sequence.

Several other tasks also fit this general block-list structure, among them: indirect object identification [3], conditional retrieval [11], inverting permutations [27], the in-context recall tasks used to study the emergence of sparse attention [19], in-context language learning [7], and reasoning with fragments of natural language [28]. We present these tasks in fig. 2.

2.2 Data Symmetries

In inductive reasoning tasks, predictions typically depend on high-level structure, rather than on arbitrary choices of tokens or absolute positions. We now describe a few symmetries present in our data, which may impose such structure.

Token Symmetries. What a model must learn in an inductive task is the relation between tokens, not the identity of any particular token. For instance, in the associative recall task of section 2.1, the rule "return the token that previously followed the query" is unaffected if we consistently relabel the vocabulary: the instance $\mathrm{a\,b} \dots \mathrm{a} \to \mathrm{b}$ and its relabeling $\mathrm{u\,v} \dots \mathrm{u} \to \mathrm{v}$ are solved by the same mechanism. A model that depends on this relational structure, rather than on specific token identities, is precisely the kind of model that generalizes to tokens unseen during training, which

[2] If multiple occurrences are allowed, a common approach is to predict the same distribution over the next token.

is a defining feature of in-context learning. We formalize this property as an invariance of the data distribution under a group of token relabelings. Consider a valid sequence of tokens $\mathbf{x} \in \mathcal{V}^N$, where $N = \sum_{i=1}^{n} k_i$. Then, applying an admissible token permutation $\pi \in S_{\mathcal{V}}$—where $S_{\mathcal{V}}$ denotes the symmetric group on $\mathcal{V}$, i.e., the set of all bijections $\pi : \mathcal{V} \to \mathcal{V}$—should result in another valid sequence $\pi(\mathbf{x}) = \pi(\mathbf{x}_1) \dots \pi(\mathbf{x}_N)$. However, what counts as an admissible permutation?

We define admissible token permutations based on the relational structure that a task imposes on the underlying vocabulary. Not every relabeling preserves the task: when the prediction relies on fixed relations between tokens, such as memorized associations, a permutation that does not respect those relations yields a different task. The admissible permutations are therefore exactly those that preserve this structure. Let $\mathcal{R} \subseteq \mathcal{V}^\rho$ be an ρ-ary relation—i.e., a set of ρ-tuples of tokens—characterizing basic skills required to solve the task. For instance, in the associative recall task, $\mathcal{R}$ is a set of associated token pairs (a binary relation, $\rho = 2$, formalized below); in a task whose context encodes memorized transitions of a Markov chain, $\mathcal{R}$ is the set of allowed transitions $(\mathbf{x}, \mathbf{x}')$. The relation $\mathcal{R}$ induces a subgroup of the symmetric group $S_{\mathcal{V}}$ called the automorphism group of $\mathcal{R}$, which is defined as

$$\mathrm{Aut}(\mathcal{V}, \mathcal{R}) := \{\pi \in S_{\mathcal{V}} \ : \ \mathbf{w} \in \mathcal{R} \Leftrightarrow \pi(\mathbf{w}) \in \mathcal{R}\}. \tag{2}$$

Its elements are precisely the permutations preserving the relational structure of $\mathcal{R}$.

To simplify the exposition throughout the remainder of this work, we specialize $\mathcal{R}$ to be an irreflexive symmetric binary relation of distinct, *unordered* token pairs:

$$\mathcal{R} = \{\{\mathbf{a}_1, \mathbf{b}_1\}, \dots, \{\mathbf{a}_k, \mathbf{b}_k\}\}, \qquad \mathbf{a}_i, \mathbf{b}_i \in \mathcal{V}, \tag{3}$$

where, for simplicity of presentation, we assume for now that every token belongs to exactly one association, i.e., $\bigcup_{i=1}^{k} \{\mathbf{a}_i, \mathbf{b}_i\} = \mathcal{V}$ and $2k = |\mathcal{V}|$.[3] We show how our theory can be extended to arbitrarily complex task structures in appendix B.

Positional Symmetries. In natural language, the relevant blocks $[\mathbf{x}_1^i \cdots \mathbf{x}_{k_i}^i]$ may appear at any position in a document, separated by an arbitrary number of irrelevant words. We can simulate this effect by giving each block a random positional offset $\phi_i \in \mathbb{N}$, as if blocks were spread out in a very large hypothetical context. We can ensure that blocks don't overlap by enforcing $\phi_{i+1} \geq \phi_i + k_i$. Then, the positional embedding of the token $\mathbf{x}_j^i$ is determined by its shifted position $\phi_i + j$. This generalizes the encoding of positions and includes the special case with no offset: $\phi_{i+1} = \phi_i + k_i$.

3 Invariant Manifolds of Learning Dynamics

In this section, we introduce our key theoretical contribution, the identification and characterization of the **Invariant Manifold of Inductive Reasoning (IMIR)**, a low-dimensional subspace to which parameters remain confined when trained via gradient descent on any block-list task. In section 3.1, we describe the transformer architecture and training setup considered throughout the paper. In section 3.3, we present our main theorem along with a brief proof sketch. In section 3.4, we provide an interpretation of the IMIR's structure.[4]

3.1 Transformer Architecture

Our models are based on an attention-only transformer, which we describe here. Let $N = \sum_{i=1}^{n} k_i$ denote the length of the input sequence. First, we map every token $\mathbf{x} \in \mathcal{V}$ to an embedding vector $\mathbf{e}_{\mathbf{x}} \in \mathbb{R}^d$. Applying this embedding to our input sequence and target output, we obtain the model input $\mathbf{T} \in \mathbb{R}^{d \times N}$ and output $\mathbf{y} \in \mathbb{R}^d$. Second, we use sinusoidal positional embeddings $\mathbf{P} \in \mathbb{R}^{d \times N}$, following Vaswani et al. [29], with random block-level offsets as described in section 2.2. As is standard, we then combine token and position embeddings to obtain the model inputs: $\mathbf{H}^{(0)} = \mathbf{T} + \mathbf{P}$.

We pass these inputs through a multi-head attention-only transformer with L attention layers and H attention heads per layer. The activations at each layer $\ell \in [L]$ are given by:

$$\mathbf{H}^{(\ell)} \ = \ \mathbf{H}^{(\ell-1)} \ + \ \sum_{h=1}^{H} \mathbf{W}_P^{(\ell,h)} \mathbf{W}_V^{(\ell,h)} \, \mathbf{H}^{(\ell-1)} \sigma\left(\mathbf{H}^{(\ell-1)\mathsf{T}} \mathbf{W}_Q^{(\ell,h)\mathsf{T}} \, \mathbf{W}_K^{(\ell,h)} \, \mathbf{H}^{(\ell-1)} \right), \tag{4}$$

[3] When we later distinguish common from rare tokens (section 4.1), only common tokens carry associations and we instead have that $\bigcup_{i=1}^{k} \{\mathbf{a}_i, \mathbf{b}_i\} = \mathcal{V}^c$, with rare tokens remaining unpaired.

[4] We note that Corollary 1 establishes that the IMIR is *invariant*: a trajectory initialized on the IMIR remains on it throughout training. We do not prove the stronger property that the IMIR is *attractive*; we only show this empirically in our experiments.

where: (i) $\mathbf{W}_K^{(\ell,h)}, \mathbf{W}_Q^{(\ell,h)}, \mathbf{W}_V^{(\ell,h)} \in \mathbb{R}^{d_h \times d}$ and $\mathbf{W}_P^{(\ell,h)} \in \mathbb{R}^{d \times d_h}$ denote the key, query, value, and projection matrices of attention head $h \in [H]$ in layer $\ell \in [L]$; (ii) $d, d_h \in \mathbb{N}$ denote the embedding and attention head dimension; (iii) $\mathbf{H}^{(\ell)} \in \mathbb{R}^{d \times N}$ denotes the residual stream at layer $\ell \in [L]$, and (iv) $\sigma : \mathbb{R}^{N \times N} \to \mathbb{R}^{N \times N}$ denotes the column-wise softmax function with auto-regressive masking. Finally, we obtain our predicted next-token by passing the hidden activations of the last token through a final linear layer $\tilde{\mathbf{y}} = \bar{\mathbf{W}}\, \mathbf{H}_{:,N}^{(L)}$, where $\bar{\mathbf{W}} \in \mathbb{R}^{d \times d}$, followed by a weight-tied unembedding layer. We train the model using the cross-entropy loss. We give a more detailed description of the architecture in appendix A.2.

Beyond Attention-Only Transformers. Standard Transformers additionally include feed-forward networks (FFN) and layer normalizations (LN). We omit these components from the core analysis in order to focus on attention layers, which are primarily responsible for inductive capabilities. Nevertheless, our theoretical results extend naturally to standard Transformers with FFNs and LNs, as we discuss in appendix A.5.

3.2 Selection, Writing, and Action Bases

Our main result (Theorem 1 and Corollary 1 below) states that, when a transformer is trained on a block-list task, gradient descent confines its key–query and output weights to the span of a small set of basis matrices. In this section, we construct these basis matrices, each of which encodes one elementary action that is possibly useful for solving a block-list task. Our construction depends on three intermediate bases: **selection matrices** are learnable mappings of tokens or positions (e.g., "map each position to the previous"), **writing bases** track how each head writes information into the residual stream (and thus how later layers can read it back), and **action bases** track how chained attention heads transform the original embeddings (extending writing bases to multiple layers).

When selecting contextual information, an attention head must compare transformed query and key representations. In our block-list task setup, this can be done in two ways: tokens can be compared directly to themselves via their identity, or they can be compared to their symmetry-related counterparts. These operations can be represented via the two matrices: $\mathbf{I}^{(t)} \in \mathbb{R}^{d \times d}$, the identity transformation acting only on the subspace of token embeddings, and $\mathbf{C} \in \mathbb{R}^{d \times d}$, a projection matrix that maps the embeddings of symmetrically associated tokens to one another (associated via $\mathcal{R}$), mapping the embedding of each token $\mathbf{a}_i$ to the embedding of its associated partner $\mathbf{b}_i$.[5] Similarly, in our setup, positions can be either compared directly to themselves, or mapped to one of their (at most $k-1$) previous positions, which can be represented by matrices: $\mathbf{I}^{(p)} \in \mathbb{R}^{d \times d}$, which denotes the identity transformation acting only on the subspace of positions; and $\mathbf{M}$, which maps position embeddings to their immediate predecessor, i.e. $\mathbf{M}\mathbf{P}_{:,i+1} = \mathbf{P}_{:,i}$, and whose matrix powers $\mathbf{M}^j$ thus shift positions back by j time steps. Based on these observations, we define the following sets of **selection matrices** for tokens and positions, respectively:

$$\mathcal{T} \;=\; \{\, \mathbf{I}^{(t)}, \mathbf{C} \,\}, \qquad\qquad \mathcal{P} \;=\; \{\, \mathbf{I}^{(p)}, \mathbf{M}, \ldots, \mathbf{M}^{k-1} \,\}, \tag{5}$$

Now, we note that each attention head can only write information to the residual stream via its output–value matrix transformation. Together with the identity defined by the transformer's residual connections, we say a layer $\ell \in [L]$ spans the following **writing basis** on the residual stream:

$$\mathcal{V}_\ell \;=\; \Big\{\, \mathbf{I} \,\Big\} \cup \Big\{\, \mathbf{W}_P^{(\ell,h)}\, \mathbf{W}_V^{(\ell,h)} \;:\; h \in [H] \,\Big\}. \tag{6}$$

Notably, after several self-attention layers, a representation may accumulate information written by the output-value projections of any set of heads in previous layers. We thus define the **action basis** $\mathcal{V}_{1:\ell}$ of a collection of layers $1 : \ell \in [L]$, as well as its **inverse action basis** $\mathcal{V}_{\ell:1}$:

$$\mathcal{V}_{1:0} = \{\, \mathbf{I} \,\} \qquad \mathcal{V}_{1:\ell} = \mathcal{V}_\ell\; \mathcal{V}_{\ell-1}\; \ldots\; \mathcal{V}_1 \qquad \mathcal{V}_{\ell:1} = \Big\{\, \mathbf{V}^+ \;:\; \mathbf{V} \in \mathcal{V}_{1:\ell} \,\Big\} \tag{7}$$

where the product of two matrix sets $\mathcal{A}$ and $\mathcal{B}$ is defined as $\mathcal{A}\mathcal{B} = \{\, \mathbf{A}\mathbf{B} : \mathbf{A} \in \mathcal{A},\, \mathbf{B} \in \mathcal{B} \,\}$ and $\mathbf{V}^+$ denotes the Moore-Penrose pseudo-inverse of $\mathbf{V}$. Note that $\mathcal{V}_{1:\ell}$ contains exactly $(H+1)^\ell$ matrices, one for each subset of heads in layers 1 to ℓ with at most one head per layer.

[5] Formally, $\mathbf{C}\,\mathbf{e}_{\mathbf{a}_i} = \mathbf{e}_{\mathbf{b}_i}$ and $\mathbf{C}\,\mathbf{e}_{\mathbf{b}_i} = \mathbf{e}_{\mathbf{a}_i}$ for all $\{\mathbf{a}_i, \mathbf{b}_i\} \in \mathcal{R}$, while collapsing all other directions to zero. See appendix B for an extension to non-symmetric token relations.

Finally, each attention head may construct keys and queries using any of the representations that have accumulated up to that layer. This is achieved by rotating a selection matrix on the left side for the query and on the right side for the key, aligning the desired subspaces. Likewise, the final output layer may select any token representation by reading from the appropriate subspace. Therefore, we define the **basis weights** $\mathcal{W}_\ell$ at layer $\ell \in [L]$ and the **basis output weights** $\bar{\mathcal{W}}$ as

$$\mathcal{W}_\ell = \mathcal{V}_{1:\ell-1} \left(\mathcal{T} \cup \mathcal{P}\right) \mathcal{V}_{\ell-1:1} \qquad\qquad \bar{\mathcal{W}} = \mathcal{T}\, \mathcal{V}_{L:1}. \tag{8}$$

Intuitively, each basis weight in $\mathcal{W}_\ell$ composes three operations: its right factor (in $\mathcal{V}_{\ell-1:1}$) reads from one position's residual stream a representation written there by some subset of earlier heads, its middle factor (in $\mathcal{T} \cup \mathcal{P}$) applies an elementary token or position action to it, and its left factor (in $\mathcal{V}_{1:\ell-1}$) matches the result against a representation written into another position's residual stream by a (possibly different) subset of earlier heads.[6] Similarly, each basis output weight in $\bar{\mathcal{W}}$ reads a representation written by a subset of heads and maps it to an output token, either directly or through its association.

3.3 Invariant Dynamics Theory

We now move forward to our main theorem. Before that, we highlight that our result relies on a few assumptions, which we state informally here and formally in appendix A.1. First, we put forward the following *data-related assumptions*: (i) token associations in $\mathcal{R}$ are fully interchangeable, so the set of valid inputs $\mathcal{X}$, where $\mathbf{x} \in \mathcal{X} \subseteq \mathcal{V}^N$, is closed under association-preserving token permutations; (ii) block positions carry random offsets; and (iii) the embeddings of tokens which do not belong to any association in $\mathcal{R}$, termed here rare tokens, are lexinvariant [30] (i.e., they are re-sampled for every sequence). We believe these assumptions to be straightforward and motivated by the traditional structure of block-list tasks. Second, we put forward the following *architecture-related assumptions*: (iv) token and position embeddings are orthonormal; (v) we analyze the merged key–query product $\mathbf{W}_Q^{(\ell)\intercal}\mathbf{W}_K^{(\ell)}$ as a single learnable matrix; and (vi) the output–value maps are fixed projections onto mutually orthogonal subspaces (following the associative-memory [17] and disentangled-residual-stream view [18, 31]). The extension of our results to models with learnable value projection matrices is sketched in appendices A.3.6 and A.4.

Theorem 1 (Gradient Confinement)**.** *Consider the population-wise training dynamics of a model satisfying Assumptions 4 to 6 on a block-list task satisfying Assumptions 1 to 3. Further, let $\mathcal{S}$ be the space of transformers with $\mathbf{W}_Q^{(\ell,h)\intercal}\,\mathbf{W}_K^{(\ell,h)} \in \operatorname{span}(\mathcal{W}_\ell)$ and $\bar{\mathbf{W}} \in \operatorname{span}(\bar{\mathcal{W}})$ and let $\mathbf{W}$ be the tuple of all weights in this transformer. If a transformer has weights in $\mathcal{S}$, then its expected gradient is confined to $\mathcal{S}$, i.e.:*

$$\mathbf{W} \in \mathcal{S} \quad \Longrightarrow \quad \mathbb{E}\big[\nabla_{\mathbf{W}} \mathcal{L}(\mathbf{W})\big] \in \mathcal{S}. \tag{9}$$

Proof Sketch. Our proof relies on two key symmetries of the data, regarding token embeddings and position embeddings, respectively. First, the only correlations between tokens are those between associated tokens. This means that the data distribution is preserved under any permutation of tokens that preserves common associations. Second, blocks have random offsets to simulate an arbitrary number of irrelevant tokens between each pair of consecutive blocks. This means that the data distribution is preserved when changing the positional offsets of blocks. We apply these two transformations (token permutation and position offset shifts) to the model input and compute how this transforms the hidden representations and model output. A key result is that all attention scores are perfectly invariant to our transformations, hence the hidden and output transformations are essentially identical to our input transformations (applied to different subspaces). Importantly, passing permuted tokens through the model is equivalent to permuting its outputs, which implies that the loss is also invariant to our transformations for any input sequence. Finally, since the data distribution is invariant to our transformations, the expected gradients must also be invariant to the same transformations. We compute exactly how the gradient is transformed under our input transformations and we show that any gradients invariant under those transformations must fit the structure described above. We provide a comprehensive proof in appendix A.3. □

[6] The inverse is needed on the right side to read information from an earlier head which encoded it in its write space. E.g., if a first-layer head stored some information $\mathbf{e}_\mathbf{x}$ using the projection $\mathbf{V}^{(1)} = \mathbf{W}_P^{(1,1)}\mathbf{W}_V^{(1,1)}$, one must decode it by using the inverse, to recover value $\mathbf{e}_\mathbf{x}$ via $\mathbf{V}^{(1)^+}\mathbf{V}^{(1)}\mathbf{e}_\mathbf{x}$. The left side does not need an inverse since it is used in the attention head through a transpose, $\left(\mathbf{V}^{(1)}\,\mathbf{e}_\mathbf{x}\right)^\intercal = \mathbf{e}_\mathbf{x}{}^\intercal\,\mathbf{V}^{(1)\intercal}$, which, by our assumptions (the value maps are orthogonal projections), already equals the required pseudo-inverse: $\mathbf{V}^{(1)\intercal} = \mathbf{V}^{(1)^+}$.

Corollary 1 (Invariant Manifold)**.** *Under the same assumptions, $\mathcal{S}$ is an **invariant manifold**[7] under the population-wise training dynamics: if $\mathbf{W} \in \mathcal{S}$ at some training step, then $\mathbf{W} \in \mathcal{S}$ for the rest of training. In other words, the weight structure described above is preserved during training with gradient descent.*

Proof. A gradient-descent step updates $\mathbf{W} \mapsto \mathbf{W} - \eta\, \mathbb{E}[\nabla_{\mathbf{W}} \mathcal{L}(\mathbf{W})]$. By Theorem 1 the update direction lies in $\mathcal{S}$, and the constraints defining $\mathcal{S}$ are linear in the key–query and output weights, so $\mathbf{W} \in \mathcal{S}$ implies the updated weights remain in $\mathcal{S}$. □

3.4 The Geometry of Reasoning Circuits

Each of the basis weights in $\mathcal{W}_\ell$ and $\bar{\mathcal{W}}$ corresponds to a simple and interpretable action over specific subspaces of embeddings. For attention layers, such interpretable actions might be, for example, "attending to the previous position" or "attending to the token retrieved by head 1", while an output action might be something like "read the token retrieved by head 3". When the model is on the IMIR, by decomposing it into its components corresponding to each weight matrix in $\mathcal{W}_\ell$ and $\bar{\mathcal{W}}$, we can describe the model as a sum of highly interpretable actions. When the model is not on the IMIR, projecting it onto the IMIR provides a partial description. We give a more comprehensive and intuitive description of the weight spaces in appendix C.1. We also give a theoretical motivation for the weight spaces from representational symmetries in appendix C.2, discussing why they capture all actions relevant for inductive tasks.

4 Understanding Circuit Competition

4.1 Data Distributional Properties of Natural Language

One important aspect not discussed above is how to sample the tokens $\mathbf{x}$ used in the tasks. Notably, prior work [19, 21, 22, 24] has shown that this choice can have a strong impact on the training dynamics of language models, and, in particular, that standard properties of token distributions in natural languages (e.g., Zipfian distributions or burstiness) can accelerate learning. We incorporate a number of natural language distributional properties into our setup as follows:

1. **Tokens differ in frequency.** Token frequencies in natural language typically follow a long-tailed distribution [32, 33]. To model this and bring our setup closer to natural language (while keeping it analytically tractable), we consider a subset of the tokens in our vocabulary to form a set of **common tokens** $\mathcal{V}^c$ with total probability of occurrence $\mathbb{P}(v \in \mathcal{V}^c) = 1 - f_r$ where $f_r \in [0; 1]$. The remaining tokens are termed **rare tokens**: $\mathcal{V}^r = \mathcal{V} \setminus \mathcal{V}^c$, with $\mathbb{P}(v \in \mathcal{V}^r) = f_r$. We assume that rare tokens form a virtually infinite set, where no rare token is ever observed twice.[8]
2. **Tokens can be bursty.** Documents in natural language tend to contain a concept or association multiple times, or not at all. In other words, in natural language, concepts do not show up uniformly, but rather in 'bursts'. Consider, for example, the word 'burstiness' itself, which appears several times in this document, while most works do not contain it at all. We incorporate this behavior in our task class by assuming that block pairs present in the context have an average multiplicity of $b \geq 1$. For example, an input sequence with $b = 2$ might look like "$\mathrm{ab}, \mathrm{gh}, \mathrm{gh}, \mathrm{ab}, \mathrm{g} \to \mathrm{h}$". We write $\mathcal{D}(b)$ for the resulting data distribution at burstiness b, making the dependence explicit when we compare gradients across burstiness levels.
3. **Tokens are polysemous.** The same token can present different meanings and associations in different documents. For example, in computer science books, the name 'Alan' might be commonly followed by the surname 'Turing', but at times it might also be followed by a different surname such as 'Baker'.[9] In our setup, we model this aspect by assuming that the next token to be predicted after a common token differs from its common association with frequency $f_v \in [0, 1]$.

[7] Not to be confused with *invariant subspaces*, referring to vector subspaces preserved by linear maps.

[8] In practice, we model such a virtually "infinite" vocabulary using a finite vocabulary and a lexinvariant assumption [30], where a rare token's embeddings are repeatedly re-sampled at every batch, inhibiting the model from memorizing it.

[9] Alan Baker (1939–2018) was an English mathematician, known for his work on effective methods in number theory, in particular those arising from transcendental number theory.

Past work has shown that in-context learning abilities emerge optimally in transformers trained with many rare tokens, high burstiness, and high within-class variability [21, 22]. In the absence of these properties, in-context learning emerges after significantly more training, or not at all. The same properties were shown to enable learning in a 3-parameter phenomenological model of induction-head formation [24]. Finally, burstiness was shown to accelerate the emergence of sparse attention in single-layer transformers [19]. We exemplify these properties in fig. 2.

4.2 The Role of Data: In-Context Learning *vs.* In-Weights Learning

We now use our Corollary 1 to study a very simple setting, but which already exhibits circuit competition: a 2-layer, single-head attention-only transformer trained to predict the second token of an in-context bigram. The IMIR for this architecture is 28-dimensional,[10] with 4 dims for the first layer actions, 16 dims for the second layer, and 8 dims for output layer:

$$\mathbf{W}_Q^{(1,1)\mathsf{T}}\mathbf{W}_K^{(1,1)} \in \operatorname{span}(\underbrace{\{\mathbf{I}^{(t)}, \mathbf{C}, \mathbf{I}^{(p)}, \mathbf{M}\}}_{\mathcal{T}\cup\mathcal{P}}) \tag{10a}$$

$$\mathbf{W}_Q^{(2,1)\mathsf{T}}\mathbf{W}_K^{(2,1)} \in \operatorname{span}(\underbrace{\{\mathbf{I}, \mathbf{W}_P^{(1,1)}\mathbf{W}_V^{(1,1)}\}}_{\mathcal{V}_{1:\ell-1}}\underbrace{\{\mathbf{I}^{(t)}, \mathbf{C}, \mathbf{I}^{(p)}, \mathbf{M}\}}_{\mathcal{T}\cup\mathcal{P}}\underbrace{\{\mathbf{I}, \left(\mathbf{W}_P^{(1,1)}\mathbf{W}_V^{(1,1)}\right)^+\}}_{\mathcal{V}_{\ell-1:1}}) \tag{10b}$$

$$\bar{\mathbf{W}} \in \operatorname{span}(\underbrace{\{\mathbf{I}^{(t)}, \mathbf{C}\}}_{\mathcal{T}}\underbrace{\{\mathbf{I}, \left(\mathbf{W}_P^{(2,1)}\mathbf{W}_V^{(2,1)}\right)^+\}\{\mathbf{I}, \left(\mathbf{W}_P^{(1,1)}\mathbf{W}_V^{(1,1)}\right)^+\}}_{\mathcal{V}_{L:1}}) \tag{10c}$$

As mentioned above, this IMIR allows us to additively decompose a model's learned weights into these interpretable directions, which we do and plot in fig. 3 (center-bottom) (we also plot each parameter individually in appendix D.3). Notably, from this figure, we see that learning appears to concentrate on four directions, three of which are useful for *in-context learning* (ICL) and one for *in-weights learning* (IWL):

$$\alpha:\ \mathbf{W}_Q^{(1,1)\mathsf{T}}\mathbf{W}_K^{(1,1)} \propto \mathbf{M}, \qquad \gamma:\ \bar{\mathbf{W}} \propto \mathbf{I}^{(t)}\left(\mathbf{W}_P^{(2,1)}\mathbf{W}_V^{(2,1)}\right)^+, \tag{11a}$$

$$\beta:\ \mathbf{W}_Q^{(2,1)\mathsf{T}}\mathbf{W}_K^{(2,1)} \propto \mathbf{I}^{(t)}\left(\mathbf{W}_P^{(1,1)}\mathbf{W}_V^{(1,1)}\right)^+, \qquad \delta:\ \bar{\mathbf{W}} \propto \mathbf{C}. \tag{11b}$$

Note that (α, β, γ) assemble the canonical induction head: the α subspace allows the first layer's head to pay attention to a previous token, the β enables the second layer's head to match the lookup token $\mathbf{x}_1^n$ to a previous one, and then γ reads the information out and extracts the retrieved label. Together they thus implement in-context learning, the only solution that works on the rare part of the data. The fourth direction δ accounts for in-weights learning, routing the input token to its canonical partner through a memorized association map $\mathbf{C}$. Causal ablations confirming that (α, β, γ) alone yields ICL and δ alone yields IWL are reported in appendix D.2.

IWL emerges first and starves ICL of gradient. The direction δ is a one-dimensional fix that already solves every query answerable by a memorized canonical association—i.e., common-token queries without within-class variation, which make up a $(1-f_r)(1-f_v)$ fraction of the data. It is the easiest direction for SGD to discover, empirically saturating well before (α, β, γ) have moved appreciably (fig. 3, center). Once δ is in place, then Theorem 2 (presented below) starts taking effect, with ICL gradients—i.e., gradients with respect to the (α, β, γ) coordinates—being suppressed exactly on the data handled by IWL,

$$\nabla_{\mathrm{ICL}}\mathcal{L}(\mathbf{W} + \mathbf{W}_{\mathrm{IWL}}) \approx \left(f_r + (1-f_r)\, f_v\right) \nabla_{\mathrm{ICL}}\mathcal{L}(\mathbf{W}). \tag{12}$$

The induction head therefore emerges later than it would in a rare-only ($f_r = 1$) world, with a delay that grows as f_r decreases or as $f_v \to 0$. Per-circuit dynamics across data regimes are in appendix D.3.

Theorem 2. *In the presence of an IWL circuit with logit scale δ, the ICL circuit receives a population gradient suppressed by a data-dependent factor:*

$$\mathbb{E}\left[\nabla_{\mathrm{ICL}}\mathcal{L}(\mathbf{W} + \mathbf{W}_{\mathrm{IWL}})\right] = \Big(f_r + (1-f_r)f_v\Big)\, \mathbb{E}\left[\nabla_{\mathrm{ICL}}\mathcal{L}(\mathbf{W})\right] + O\big(|\mathcal{V}|\, e^{-\delta} + |\mathcal{V}|^{-1}\big), \tag{13}$$

where $\mathbf{W}$ denotes the ICL-only configuration (no in-weights learning). The error term vanishes in the saturated-IWL ($\delta \to \infty$) and large-vocabulary ($|\mathcal{V}| \to \infty$) limit.

[10] The IMIR has 28-dimensions for the space of key-query products and final projections, not the full parameter space.

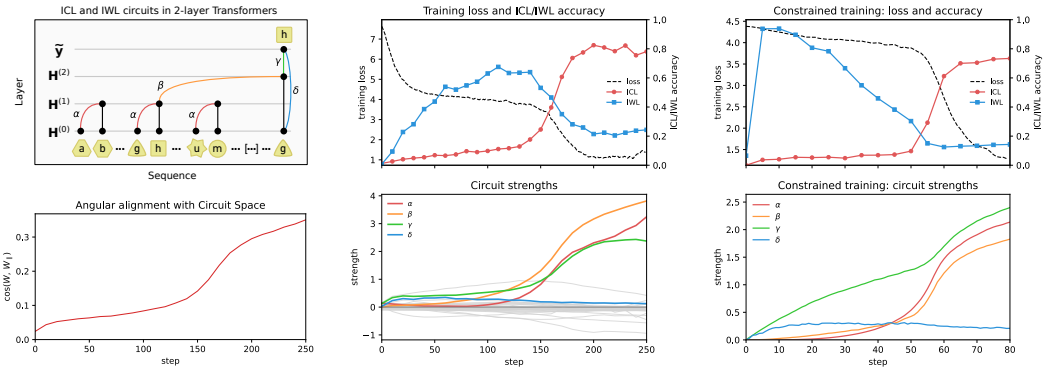

Figure 3: In two-layer transformers, ICL and IWL circuits compete during training. Top left: ICL has three circuit units (α, β, and γ, forming an induction head) and IWL has one circuit unit (δ, predicting the next token directly from the current token). Bottom left: the parameters become aligned with the IMIR during training. Center: the emergence of ICL/IWL abilities and circuits. Right: emergence of ICL/IWL during isolated training (only α, β, γ, δ) exhibits similar dynamics. Training details in appendix D.1.

Proof Sketch. The IWL solution solves common queries whose in-context partner equals their canonical partner, a fraction $f_c(1 - f_v)$ of the data; there the loss is near-zero and, up to a residual of order $|\mathcal{V}|\, e^{-\delta}$, the ICL direction receives no gradient. On the remaining $f_r + (1 - f_r)f_v$ of the data the IWL solution is either inactive (rare query) or points at the wrong token (scrambled partner), so the ICL gradient is unchanged up to a finite-vocabulary cross-talk of order $|\mathcal{V}|^{-1}$. We give a detailed proof in appendix D.4. □

Burstiness amplifies the ICL gradient. Theorem 2 describes how the rare-token frequency f_r and the within-class variability f_v reduce the gradient received by the ICL circuit. Burstiness, which we introduced in section 4.1, has the opposite effect. In a bursty sequence, the block relevant to the query appears b times in the context, and each occurrence provides the same in-context evidence to the induction circuit, whereas the distractor blocks provide no consistent signal. The ICL gradient therefore grows with b, which accelerates the emergence of the induction head.

Theorem 3. *In the distractor-dominated regime, where the number of context blocks grows with the burstiness held fixed ($n \to \infty$ with b fixed, so $n \gg b$), the ICL circuit receives a population gradient proportional to burstiness:*

$$\mathbb{E}_{\mathcal{D}(b)}\big[\nabla_{\mathrm{ICL}}\mathcal{L}(\mathbf{W})\big] \;=\; b\,\mathbb{E}_{\mathcal{D}(1)}\big[\nabla_{\mathrm{ICL}}\mathcal{L}(\mathbf{W})\big]\,\big(1 \;+\; O\big(b/n\big)\big), \tag{14}$$

where $\mathbb{E}_{\mathcal{D}(b)}$ denotes expectation over the data distribution at burstiness b.

Proof Sketch. Burstiness acts as a multiplier on the relevant blocks: the ICL gradient decomposes into per-block signals that are identical for the b relevant blocks and zero in expectation for distractors, so the relevant signal is multiplied by b. The only cross-block coupling is the shared layer-2 softmax denominator Z_q at the query position, which depends on b only through a relative $O(b/n)$ shift; since every per-block signal scales as $1/Z_q$, this yields a *multiplicative* $\big(1+O(b/n)\big)$ correction rather than an additive one. This is crucial, because the leading term is itself of order b/n. We give a detailed proof in appendix D.5. □

The induction head is a winning ticket. The Lottery Ticket Hypothesis (LTH) [25] posits that a randomly initialized network contains a sparse subnetwork which, trained in isolation, matches the dense network's performance. Our framework identifies one such subnetwork explicitly: the four-parameter $(\alpha, \beta, \gamma, \delta)$ IMIR sub-manifold. We test the prediction directly via *constrained training*: at every SGD step we project the model back onto the subspace spanned by $\alpha, \beta, \gamma, \delta$. The resulting dynamics closely match those of the unconstrained network (fig. 3, right).

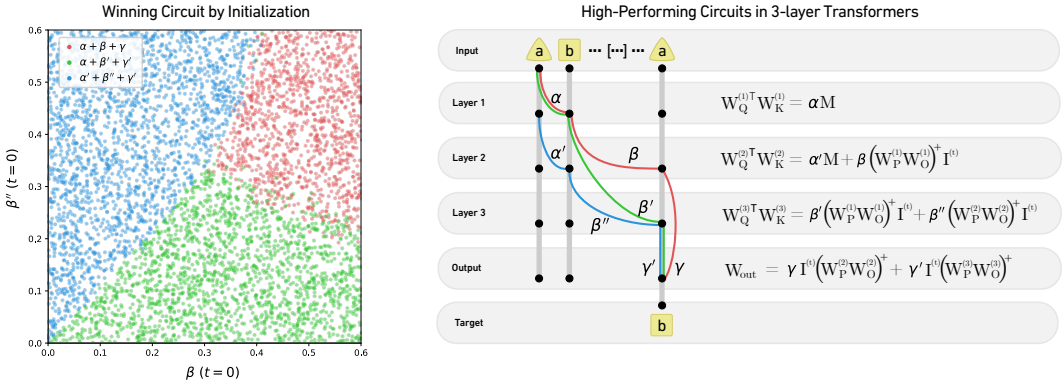

Figure 4: Training a 3-layer Transformer to solve in-context bi-grams yields three different solutions. On the left, we show the winning solution at various initializations. On the right, we show the precise mechanism of each possible solution with formulas for model weights.

4.3 The Role of Initialization: Competing Induction Heads

An induction head can use any pair of attention heads in different layers. Thus, in a transformer with three layers and one head per layer, trained on the same task, three distinct induction-head circuits are possible, as depicted in fig. 4 (right). A randomly initialized model must converge to one of them (or a composition of them). To better understand how the initialization influences the learned circuit, we vary two parameters (denoted as β and β'') and we plot the 'winning' circuit in fig. 4 (left). We give the full training details in appendix E.1 and we plot the full training dynamics for 12 seeds in appendix E.2.

Winning mechanics are complex to predict. In the LTH literature, winning subnetworks are typically identified by weight magnitude after training, and whether they can instead be predicted from properties of the initialization—such as initial weight magnitude—remains actively debated [34–36]. The simplest initialization-based predictor in our setup would be that each circuit wins when its own units start large. We can analyze this predictor by looking at fig. 4 (left). In fact, this predictor aligns with the fact that the $(\alpha', \beta'', \gamma')$-circuit (layers 2 and 3, blue) tends to emerge when β'' is initialized with a large magnitude. However, the $(\alpha, \beta', \gamma')$-circuit (layers 1 and 3, green) seems to benefit from a larger β, which is not part of this circuit. Moreover, the (α, β, γ)-circuit (layers 1 and 2, red) seems to require not just a large β (part of the circuit), but also a large β'' (not part of the circuit). Together, these results suggest the history is not that simple.

The story of a circuit battle. A closer look at the phase transitions in fig. 4 (left) also sheds light on our transformer's learning dynamics. First, note that these phase transitions are very sharp between the blue and green solutions and between blue and red, but not between green and red. This is because the red and green circuits are cooperative: both share α, while β and β' are in different layers. Meanwhile, blue is competing with both red and green, trying to use the same layers in different ways. This might also explain why red requires a large β'': it stimulates the blue circuit, which in turn inhibits the green circuit. Notably, the learning dynamics plots in appendix E.2 appear to support this analysis.

5 Automated Circuit Detection for Induction Tasks

Our theoretical framework offers two benefits that may improve circuit discovery in trained models. First, the IMIR has a low dimensionality compared to the full parameter space, which greatly reduces the circuit search space. Second, while most interpretability works identify circuits at the level of attention heads [3, 4, 37, 38], the IMIR allows us to identify circuits at the more granular level of precise weight structures, enabling us to understand the exact mechanism of each head.[11]

To illustrate these benefits, we apply our framework to the question of Automated Circuit Discovery [4] of induction circuits. We train deep attention-only single-head transformers on the two-hop induction task [10] (training details in appendix F.1). We discover the learned circuit using a simple algorithm: we project the learned parameters to the IMIR and we ablate the dimensions that don't

[11]The bases present in our IMIR are also similar to the variables produced by the D-RASP decompilation procedure in Friedman et al. [31], Weiss et al. [39].

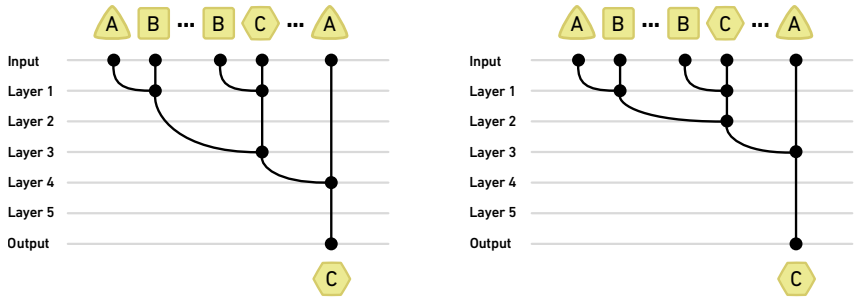

Figure 5: Two auto-discovered circuits in deep Transformers for two-hop induction.

impact performance. We depict the recovered circuits schematically in fig. 5. We describe the complete algorithm in appendix F.2 and the complete circuits in appendix F.3.

One interesting finding is that the attention paths depicted in fig. 5 are not implemented using only a minimal set of necessary directions, but also leverage a few additional 'aiding' directions. The additional directions marginally increase the accuracy by reducing the attention to irrelevant tokens. This kind of structure would be invisible to existing automated circuit discovery algorithms with granularity at the attention head level.

6 Conclusion and Future Work

In this paper, we identified an invariant manifold for a class of transformers trained on block-list tasks. This manifold allows us to both study transformers' training dynamics, and to decompose trained models' weights into interpretable subspaces. We now discuss three directions for future work in which our theoretical framework might prove useful in addition to the use cases demonstrated in this paper.

Improving Reasoning Abilities. Mechanistic interpretability has already proven useful for the design of improved neural network architectures [12, 40–42]. For example, the selection mechanism in *Mamba* architectures [43] is inspired by induction heads [26]. A better understanding of inductive reasoning abilities (and their emergence) may thus enable the development of improved learning algorithms. In particular, our theoretical framework provides a geometric description of a large class of induction circuits. Useless circuits can be identified and pruned more effectively, while remaining circuits can be described using a reduced number of parameters, reducing the memory footprint and inference latency. Moreover, it is known that learning induction heads is more difficult in long sequences [20], while deep circuits require either an implicit curriculum or an exponential amount of data [11, 44]. Can we design neural architectures without these limitations?

Accelerating Experiments at Scale. In deep learning, many important phenomena, such as data dependency of ICL [21] or generalization beyond overfitting [45], began as empirical observations in synthetic tasks, before they were later shown to hold in general settings [22, 46]. There is a simple reason for this: training large models on full datasets is slow and costly, whereas uncovering novel phenomena may require running a great number of experiments quickly. Our framework can further accelerate experimentation in two ways. First, by constraining learning to the IMIR, or even a reduced subset of interesting dimensions, training can be greatly optimized, reducing experiment time and computational demands. Second, by providing the geometric description of meaningful directions, our framework can aid hypothesis generation and understanding of training dynamics.

Invariant Manifolds in Other Tasks. In this work, we have shown how invariant manifolds can be used to understand training dynamics and model internals. While invariant manifolds of learning dynamics have been studied before [20, 47, 48], to the best of our knowledge, this work is the first to identify the geometry of the invariant manifolds responsible for learning a large class of tasks. If inductive tasks exhibit a low-dimensional invariant manifold, then maybe other tasks also do. Uncovering them could lead to a better understanding of learning dynamics in deep neural networks.

References

[1] Zifan Zheng, Yezhaohui Wang, Yuxin Huang, Shichao Song, Mingchuan Yang, Bo Tang, Feiyu Xiong, and Zhiyu Li. Attention heads of large language models: A survey. *Patterns*, 6(2), 2025.

[2] Chris Olah, Nick Cammarata, Ludwig Schubert, Gabriel Goh, Michael Petrov, and Shan Carter. Zoom in: An introduction to circuits. *Distill*, 5(3):e00024–001, 2020.

[3] Kevin Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. Interpretability in the wild: a circuit for indirect object identification in GPT-2 small. In *The Eleventh International Conference on Learning Representations*, 2023.

[4] 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.

[5] 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. *Transactions on Machine Learning Research*, 2022.

[6] Jamie Simon, Daniel Kunin, Alexander Atanasov, Enric Boix-Adserà, Blake Bordelon, Jeremy Cohen, Nikhil Ghosh, Florentin Guth, Arthur Jacot, Mason Kamb, et al. There will be a scientific theory of deep learning. *arXiv preprint arXiv:2604.21691*, 2026.

[7] Ekin Akyürek, Bailin Wang, Yoon Kim, and Jacob Andreas. In-context language learning: Architectures and algorithms. In *International Conference on Machine Learning*. PMLR, 2024.

[8] Benjamin L Edelman, Ezra Edelman, Surbhi Goel, Eran Malach, and Nikolaos Tsilivis. The evolution of statistical induction heads: In-context learning Markov chains. *Advances in Neural Information Processing Systems*, 37, 2024.

[9] Aditya Varre, Gizem Yüce, and Nicolas Flammarion. Learning in-context n-grams with transformers: Sub-n-grams are near-stationary points. In *International Conference on Machine Learning*. PMLR, 2025.

[10] Clayton Sanford, Daniel Hsu, and Matus Telgarsky. Transformers, parallel computation, and logarithmic depth. In *International Conference on Machine Learning*. PMLR, 2024.

[11] Tiberiu Musat. Mechanism and emergence of stacked attention heads in multi-layer transformers. In *The Thirteenth International Conference on Learning Representations*, 2025.

[12] Zhiwei Wang, Yunji Wang, Zhongwang Zhang, Zhangchen Zhou, Hui Jin, Tianyang Hu, Jiacheng Sun, Zhenguo Li, Yaoyu Zhang, and Zhi-Qin John Xu. Understanding the language model to solve the symbolic multi-step reasoning problem from the perspective of buffer mechanism. *arXiv preprint arXiv:2405.15302*, 2024.

[13] Zeyuan Allen-Zhu. Physics of language models: Part 4.1, architecture design and the magic of canon layers. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL `https://openreview.net/forum?id=kxv0M6I7Ud`.

[14] Aaditya K Singh, Ted Moskovitz, Sara Dragutinovic, Felix Hill, Stephanie CY Chan, and Andrew M Saxe. Strategy coopetition explains the emergence and transience of in-context learning. In *International Conference on Machine Learning*. PMLR, 2025.

[15] William Merrill, Nikolaos Tsilivis, and Aman Shukla. A tale of two circuits: Grokking as competition of sparse and dense subnetworks. *arXiv preprint arXiv:2303.11873*, 2023.

[16] Ziqian Zhong, Ziming Liu, Max Tegmark, and Jacob Andreas. The clock and the pizza: Two stories in mechanistic explanation of neural networks. *Advances in neural information processing systems*, 36:27223–27250, 2023.

[17] 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, 2023.

[18] Eshaan Nichani, Alex Damian, and Jason D Lee. How transformers learn causal structure with gradient descent. In *International Conference on Machine Learning*. PMLR, 2024.

[19] Nicolas Zucchet, Francesco d'Angelo, Andrew K Lampinen, and Stephanie CY Chan. The emergence of sparse attention: impact of data distribution and benefits of repetition. *Advances in Neural Information Processing Systems*, 38, 2025.

[20] Tiberiu Musat, Tiago Pimentel, Lorenzo Noci, Alessandro Stolfo, Mrinmaya Sachan, and Thomas Hofmann. On the emergence of induction heads for in-context learning, 2025. URL `https://arxiv.org/abs/2511.01033`.

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

[22] Minsung Kim, Dong-Kyum Kim, Jea Kwon, Nakyeong Yang, Kyomin Jung, and Meeyoung Cha. How training data shapes the use of parametric and in-context knowledge in language models. *arXiv preprint arXiv:2510.02370*, 2025.

[23] Aaditya Singh, Stephanie Chan, Ted Moskovitz, Erin Grant, Andrew Saxe, and Felix Hill. The transient nature of emergent in-context learning in transformers. *Advances in neural information processing systems*, 36:27801–27819, 2023.

[24] Gautam Reddy. The mechanistic basis of data dependence and abrupt learning in an in-context classification task. In *The Twelfth International Conference on Learning Representations*, 2024.

[25] Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *International Conference on Learning Representations*, 2019.

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

[27] Rohan Alur, Chris Hays, Manish Raghavan, and Devavrat Shah. The impossibility of inverse permutation learning in transformer models. *arXiv preprint arXiv:2509.24125*, 2025.

[28] Viktor Schlegel, Kamen Pavlov, and Ian Pratt-Hartmann. Can transformers reason in fragments of natural language? In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11184–11199, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.768. URL `https://aclanthology.org/2022.emnlp-main.768/`.

[29] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[30] Qian Huang, Eric Zelikman, Sarah Chen, Yuhuai Wu, Gregory Valiant, and Percy S Liang. Lexinvariant language models. *Advances in Neural Information Processing Systems*, 36:23990–24012, 2023.

[31] Dan Friedman, Alexander Wettig, and Danqi Chen. Learning transformer programs. *Advances in Neural Information Processing Systems*, 36:49044–49067, 2023.

[32] Steven T Piantadosi. Zipf's word frequency law in natural language: A critical review and future directions. *Psychonomic bulletin & review*, 21(5):1112–1130, 2014.

[33] George Kingsley Zipf. *Human Behavior and the Principle of Least Effort: An Introduction to Human Ecology*. Addison-Wesley Press, Cambridge, MA, 1949.

[34] Jonathan Frankle, Gintare Karolina Dziugaite, Daniel M Roy, and Michael Carbin. Pruning neural networks at initialization: Why are we missing the mark? In *International Conference on Learning Representations*, 2021.

[35] Mansheej Paul, Feng Chen, Brett W Larsen, Jonathan Frankle, Surya Ganguli, and Gintare Karolina Dziugaite. Unmasking the lottery ticket hypothesis: What's encoded in a winning ticket's mask? In *The Eleventh International Conference on Learning Representations*, 2023.

[36] Bohan Liu, Zijie Zhang, Peixiong He, Zhensen Wang, Yang Xiao, Ruimeng Ye, Yang Zhou, Wei-Shinn Ku, and Bo Hui. A survey of lottery ticket hypothesis. *arXiv preprint arXiv:2403.04861*, 2024.

[37] 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*, 2021. https://transformer-circuits.pub/2021/framework/index.html.

[38] Arnab Sen Sharma, Giordano Rogers, Natalie Shapira, and David Bau. Llms process lists with general filter heads. *arXiv preprint arXiv:2510.26784*, 2025.

[39] Gail Weiss, Yoav Goldberg, and Eran Yahav. Thinking like transformers. In *International Conference on Machine Learning*, pages 11080–11090. PMLR, 2021.

[40] Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. In *The Twelfth International Conference on Learning Representations*, 2024.

[41] Jeffrey Willette, Heejun Lee, and Sung Ju Hwang. Delta attention: Fast and accurate sparse attention inference by delta correction. *Advances in Neural Information Processing Systems*, 38, 2025.

[42] Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision transformers need registers. In *The Twelfth International Conference on Learning Representations*, 2024.

[43] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. In *First Conference on Language Modeling*, 2024.

[44] Zixuan Wang, Eshaan Nichani, Alberto Bietti, Alex Damian, Daniel Hsu, Jason D Lee, and Denny Wu. Learning compositional functions with transformers from easy-to-hard data. In *Proceedings of the Thirty Eighth Conference on Learning Theory*. PMLR, 2025.

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

[46] Ziming Liu, Eric J Michaud, and Max Tegmark. Omnigrok: Grokking beyond algorithmic data. In *The Eleventh International Conference on Learning Representations*, 2023.

[47] Amir Joudaki, Giulia Lanzillotta, Mohammad Samragh Razlighi, Iman Mirzadeh, Keivan Alizadeh, Thomas Hofmann, Mehrdad Farajtabar, and Fartash Faghri. Barriers for learning in an evolving world: Mathematical understanding of loss of plasticity. *arXiv preprint arXiv:2510.00304*, 2025.

[48] Zeru Zhang, Jiayin Jin, Zijie Zhang, Yang Zhou, Xin Zhao, Jiaxiang Ren, Ji Liu, Lingfei Wu, Ruoming Jin, and Dejing Dou. Validating the lottery ticket hypothesis with inertial manifold theory. *Advances in neural information processing systems*, 34:30196–30210, 2021.

[49] Xinting Huang, Aleksandra Bakalova, Satwik Bhattamishra, William Merrill, and Michael Hahn. Discovering interpretable algorithms by decompiling transformers to rasp, 2026. URL `https://arxiv.org/abs/2602.08857`.

A Details of the theoretical results

A.1 Overview of the assumptions

Theorem 1 (and its Corollary 1) holds under the six assumptions stated formally below, each followed by its motivation. The first three constrain the *data* distribution (Assumptions 1, 3 and 4); the last three fix the *architecture* and parameterization we analyze (Assumptions 5 and 6). The same assumptions are summarized informally in words just above Theorem 1.

Assumption 1 ($\mathcal{R}$-invariance of common tokens)**.** *Partition the vocabulary into a set of common and rare tokens: $\mathcal{V}^c \cup \mathcal{V}^r = \mathcal{V}$. The correlations among common tokens are limited to a set of symmetric, balanced, non-overlapping associations $\mathcal{R} \subset [\mathcal{V}^c]^2$, so that the data distribution is unchanged under a π permutation.*

In other words, $\mathbf{w}$ and $\pi(\mathbf{w})$ are equally likely for any string $\mathbf{w} \in \mathcal{V}^*$ and any permutation $\pi \in S_{\mathcal{V}}$ that fixes rare tokens and preserves associations, i.e. $\pi(\mathbf{x}) \in \mathcal{V}^r$ for all $\mathbf{x} \in \mathcal{V}^r$ and $\{\pi(\mathbf{a}), \pi(\mathbf{b})\} \in \mathcal{R}$ for all $\{\mathbf{a}, \mathbf{b}\} \in \mathcal{R}$. We do this to emphasize inductive abilities and limit the room for memorization, considering languages in which common associations are fully interchangeable.

Assumption 2 (Independent positional offsets)**.** *Each block carries an independent positional offset $\phi_{j,b}$, drawn i.i.d. and marginally uniform, so that the data distribution is unchanged under a global per-frequency offset shift $\phi_{j,b} \mapsto \phi_{j,b} + \Delta\phi_j$.*

Our data uses these random offsets to simulate the presence of an arbitrary number of irrelevant tokens that may separate consecutive blocks in a real document.

Assumption 3 (Lexinvariance of rare tokens)**.** *Rare-token embeddings are resampled i.i.d. before each sequence from a distribution whose mean and second moment match those of the common tokens, $\mu = \frac{1}{|\mathcal{V}^c|}\sum_{\mathbf{x}\in\mathcal{V}^c} \mathbf{e}_\mathbf{x}$ and $S = \frac{1}{|\mathcal{V}^c|}\sum_{\mathbf{x}\in\mathcal{V}^c} \mathbf{e}_\mathbf{x}\mathbf{e}_\mathbf{x}^\intercal$.*

We target the limit $|\mathcal{V}^r| \gg |\mathcal{V}^c|$ in which individual rare-token associations cannot be memorized. Following Huang et al. [30], resampling the embeddings every sequence inhibits memorization and lets us handle rare tokens distributionally.

Assumption 4 (Orthonormal centered embeddings)**.** *The common-token embeddings $\{\mathbf{e}_\mathbf{x}\}_{\mathbf{x}\in\mathcal{V}^c}$ and the sinusoidal basis vectors $\{\mathbf{q}_j, \mathbf{r}_j\}_{j\in[m]}$ are mutually orthogonal, with $\mathbf{e_a}^\intercal\mathbf{e_b} = \mathbf{e_a}^\intercal\mathbf{q}_j = \mathbf{e_a}^\intercal\mathbf{r}_j = 0$ for distinct $\mathbf{a}, \mathbf{b} \in \mathcal{V}^c$ and all $j \in [m]$; all common tokens share a common norm, $\|\mathbf{e_a}\|_2 = \|\mathbf{e_b}\|_2$; and the common embeddings are* centered*, $\sum_{\mathbf{x}\in\mathcal{V}^c} \mathbf{e}_\mathbf{x} = \mathbf{0}$. In particular $\mathbb{U}^{(\mathrm{t})} \perp \mathbb{U}^{(\mathrm{p})}$.*

We adopt the *associative-memory* view of Bietti et al. [17], in which representations are near-orthogonal; this is realized by large embedding dimension $d \gg 1$ (where random vectors are near-orthogonal), by data whitening, or by one-hot encodings [18]. Centering is the standard mean-subtraction preprocessing, and it is what makes the token commutant in section A.3.6 exactly $\mathrm{span}\{\mathbf{I}^{(t)}, \mathbf{C}\}$: without it a residual rank-one term $\mathbf{1}\mathbf{1}^\intercal$ survives and the manifold is only approximately invariant. With centering, the matched mean of Assumption 3 is $\mu = \mathbf{0}$.

Assumption 5 (Merged key–query parameterization)**.** *Queries and keys enter the loss only through the merged product $\mathbf{W}^{(\ell,h)} = \mathbf{W}_Q^{(\ell,h)^\intercal}\mathbf{W}_K^{(\ell,h)}$, which we treat as the learnable attention parameter of head (ℓ, h).*

The attention scores $\mathbf{H}^{(\ell-1)^\intercal}\mathbf{W}_Q^{(\ell,h)^\intercal}\mathbf{W}_K^{(\ell,h)}\mathbf{H}^{(\ell-1)}$ depend on queries and keys only through this product, the "QK circuit" of mechanistic analyses [37]. Treating it as a single matrix removes the gauge freedom $(\mathbf{W}_Q^{()}, \mathbf{W}_K^{()}) \mapsto (\mathbf{R}\mathbf{W}_Q^{()}, \mathbf{R}\mathbf{W}_K^{()})$ and is exactly what the basis weights $\mathcal{W}_\ell$ describe. Relaxing it to independently learnable factors is discussed in appendix A.4.

Assumption 6 (Fixed orthogonal output–value maps)**.** *Each output–value map $\mathbf{V}^{(\ell,h)} = \mathbf{W}_P^{(\ell,h)}\mathbf{W}_V^{(\ell,h)}$ is a fixed isometry onto a write subspace that is orthogonal to the residual stream entering layer ℓ and to the write subspaces of the other heads of layer ℓ, so that $\mathbf{V}^{(\ell,h)^\intercal}\mathbf{V}^{(\ell,h')} = \delta_{hh'}\mathbf{I}$ on that input space.*

This is the multi-head form of the *disentangled residual stream* [18, 31], again justified by the associative-memory view: large random projections write into near-orthogonal subspaces. It is the

main simplification of our analysis. The detailed proof in appendix A.3 establishes the theorem under it, and appendix A.4 sketch (without a complete proof) how learnable value and projection factorizations might be accommodated.

A.2 Transformer Architecture

In this section, we recap our transformer architecture in a bit more detail.

First, we denote our model's vocabulary as $\mathcal{V}$, and we map every token $\mathbf{x} \in \mathcal{V}$ to a non-learnable embedding vector $\mathbf{e}_{\mathbf{x}} \in \mathbb{R}^d$. We use $\mathrm{Embed} : \mathcal{V} \to \mathbb{R}^d$ to denote the mapping $\mathrm{Embed}[\mathbf{x}] = \mathbf{e}_{\mathbf{x}}$. Applying this embedding to our input sequence and target output, we obtain the inputs $\mathbf{T} \in \mathbb{R}^{d \times N}$ and output $\mathbf{y} \in \mathbb{R}^d$.

Second, we add *sinusoidal* position embeddings $\mathbf{P} \in \mathbb{R}^{d \times N}$ to our model. As discussed above, we assume each block can be separated by an arbitrary number of irrelevant words, which we model by adding random offsets for each block. Further, we use m frequencies, which gives us:

$$\mathbf{P}_{:,bk+i} = \sum_{j=1}^{m} \Big(\sin\left(\omega_j i + \phi_{j,b}\right) \mathbf{q}_j + \cos\left(\omega_j i + \phi_{j,b}\right) \mathbf{r}_j \Big) \tag{15}$$

where $\mathbf{q}_j, \mathbf{r}_j \in \mathbb{R}^d$ are fixed orthogonal basis vectors, $\omega \in \mathbb{R}^m$ are fixed frequencies of order k (i.e. $\omega_j = 2\pi N_j / k$), and $\phi \in \mathbb{R}^{m \times n}$ are block offsets sampled i.i.d. for every sequence during training.

Given these token position embeddings, we add them together to get our model's inputs $\mathbf{H}^{(0)} = \mathbf{T} + \mathbf{P}$. We then use an attention-only transformer with L attention layers and H attention heads per layer. The representations at each layer $\ell \in [L]$ are given by:

$$\mathbf{H}^{(\ell)} \;=\; \mathbf{H}^{(\ell-1)} \;+\; \sum_{h=1}^{H} \mathbf{W}_P^{(\ell,h)} \mathbf{W}_V^{(\ell,h)} \, \mathbf{H}^{(\ell-1)} \sigma\Big(\mathbf{H}^{(\ell-1)\intercal} \mathbf{W}_Q^{(\ell,h)\intercal} \, \mathbf{W}_K^{(\ell,h)} \, \mathbf{H}^{(\ell-1)} \Big), \tag{16}$$

where σ denotes the column-wise softmax function with autoregressive masking, $\mathbf{W}^{(\ell,h)} \in \mathbb{R}^{d \times d}$ denote the merged key-query attention weights for head $h \in [H]$ in layer $\ell \in [L]$, $\mathbf{V}^{(\ell,h)} \in \mathbb{R}^{d \times d}$ denote the merged output-value projection weights, and $\mathbf{H}^{(\ell)} \in \mathbb{R}^{d \times N}$ denote the residual stream at layer $\ell \in [L]$.

Finally, we get predicted outputs by passing the hidden activations corresponding to the last input token through a final linear layer:

$$\mathbf{z} \;=\; \bar{\mathbf{W}} \, \mathbf{H}^{(L)}{}_{:,N} \tag{17}$$

where we denote the output weights as $\bar{\mathbf{W}} \in \mathbb{R}^{d \times d}$ and the final output as $\mathbf{z} \in \mathbb{R}^d$. We train the model to predict the correct token using the cross-entropy loss with a tied unembedding matrix:

$$\mathcal{L} \;=\; -\mathbf{z}^\intercal \mathbf{y} \;+\; \log\left(\sum_{\mathbf{x} \in \mathcal{V}} e^{\,\mathbf{z}^\intercal \mathbf{e}_{\mathbf{x}}} \right). \tag{18}$$

A.3 Proof of Theorem 1

Theorem 1 (Gradient Confinement)**.** *Consider the population-wise training dynamics of a model satisfying Assumptions 4 to 6 on a block-list task satisfying Assumptions 1 to 3. Further, let $\mathcal{S}$ be the space of transformers with $\mathbf{W}_Q^{(\ell,h)\intercal} \, \mathbf{W}_K^{(\ell,h)} \in \mathrm{span}(\mathcal{W}_\ell)$ and $\bar{\mathbf{W}} \in \mathrm{span}(\bar{\mathcal{W}})$ and let $\mathbf{W}$ be the tuple of all weights in this transformer. If a transformer has weights in $\mathcal{S}$, then its expected gradient is confined to $\mathcal{S}$, i.e.:*

$$\mathbf{W} \in \mathcal{S} \quad \Longrightarrow \quad \mathbb{E}\big[\nabla_{\mathbf{W}} \mathcal{L}(\mathbf{W})\big] \in \mathcal{S}. \tag{9}$$

Proof. The proof proceeds in six steps. *First* (section A.3.1), we set up the objects the argument acts on: the orthogonal token and position embedding subspaces $\mathbb{U}^{(\mathrm{t})}, \mathbb{U}^{(\mathrm{p})}$, the selection matrices $\mathcal{T}, \mathcal{P}$ that generate the basis weights, and the representation spaces reachable at each layer. *Second* (section A.3.2), we isolate a simplified single-head subspace with identity queries and fixed orthogonal value matrices, which already exposes the mechanism; the learnable, multi-head case is recovered at

the end. *Third* (section A.3.3), we encode the two data symmetries of a block-list task (permuting associated common tokens and shifting block positional offsets) as a pair of orthogonal rotations acting on $\mathbb{U}^{(\mathrm{t})}, \mathbb{U}^{(\mathrm{p})}$. *Fourth* (section A.3.4), we show that these rotations leave all attention scores invariant, so the hidden representations and the model output are simply rotated by the same transformation; consequently the loss is unchanged. *Fifth* (section A.3.5), we propagate this equivariance backward to find how each weight's gradient transforms, and observe that, because the data distribution is invariant, the *population* gradient must itself be invariant under the transformation. *Sixth* (section A.3.6), we analyze which matrices can be invariant: a commutant argument forces every gradient block to vanish across concepts and, within a concept, to be a polynomial in the selection matrices, exactly the basis-weight structure $\mathrm{span}(\mathcal{W}_\ell), \mathrm{span}(\bar{\mathcal{W}})$ defining $\mathcal{S}$. This establishes $\mathbb{E}[\nabla_{\mathbf{W}}\mathcal{L}] \in \mathcal{S}$ and hence Theorem 1. □

We note that in appendix A.4 we then lift our argument to independently learnable factorizations via circuit folding.

A.3.1 Core Concepts

We begin by introducing several key concepts necessary for our proof.

Embedding Spaces. We use $\mathbb{U}^{(\mathrm{t})}, \mathbb{U}^{(\mathrm{p})} \subset \mathbb{R}^d$ to denote the vector spaces containing the *token* and *position* embeddings:

$$\mathbb{U}^{(\mathrm{t})} \;=\; \mathrm{span}\big(\{\, \mathbf{e}_{\mathbf{x}} : \mathbf{x} \in \mathcal{V}^c \,\}\big), \qquad\qquad \mathbb{U}^{(\mathrm{p})} \;=\; \mathrm{span}\big(\{\, \mathbf{q}_j, \mathbf{r}_j : j \in [m] \,\}\big) \tag{19}$$

Two remarks are in order. First, $\mathbb{U}^{(\mathrm{t})}$ is spanned by *common*-token embeddings only: rare tokens have no fixed embedding that could span a subspace, since their embeddings are re-sampled for every sequence under our lexinvariance assumption (appendix A.1); they are instead handled distributionally in the arguments below. Second, $\mathbb{U}^{(\mathrm{p})}$ is spanned by the $2m$ sinusoidal basis vectors $\mathbf{q}_j, \mathbf{r}_j$ because every position embedding is a linear combination of them (appendix A.2): a position corresponds to a rotation angle within each frequency plane, rather than to a separate direction, which is what allows offset shifts to act as rotations of $\mathbb{U}^{(\mathrm{p})}$.

Identity Matrices. We use $\mathbf{I}^{(t)}, \mathbf{I}^{(p)} \in \mathbb{R}^{d\times d}$ to denote the identity transformations acting only on the subspaces $\mathbb{U}^{(\mathrm{t})}$ and $\mathbb{U}^{(\mathrm{p})}$, respectively, while collapsing all other directions to zero. Formally, they are the orthogonal projections onto these subspaces; under our orthonormality assumptions (appendix A.1), they can be written as

$$\mathbf{I}^{(t)} \;=\; \sum_{\mathbf{x}\in\mathcal{V}^c} \mathbf{e}_{\mathbf{x}}\, \mathbf{e}_{\mathbf{x}}{}^{\intercal}, \qquad\qquad \mathbf{I}^{(p)} \;=\; \sum_{j=1}^{m} \Big(\mathbf{q}_j\, \mathbf{q}_j{}^{\intercal} + \mathbf{r}_j\, \mathbf{r}_j{}^{\intercal}\Big). \tag{20}$$

Note that these definitions are basis-independent: the subspaces $\mathbb{U}^{(\mathrm{t})}$ and $\mathbb{U}^{(\mathrm{p})}$ need not be aligned with the neuron axes.

Representation Spaces. We use $\mathbb{U}^{(\ell)} \subset \mathbb{R}^d$ to denote the vector space containing the representations at layer $\ell \in [L]$ such that $\mathbf{H}^{(\ell)} \in \mathbb{U}^{(\ell)}$ always holds. Therefore, we have that

$$\mathbb{U}^{(0)} = \mathbb{U}^{(\mathrm{t})} + \mathbb{U}^{(\mathrm{p})}, \qquad \mathbb{U}^{(\ell)} = \mathbb{U}^{(\ell-1)} + \sum\nolimits_{h=1}^{H} \mathbf{W}_P^{(\ell,h)} \mathbf{W}_V^{(\ell,h)} \mathbb{U}^{(\ell-1)}, \;\; \forall \ell \in [L] \tag{21}$$

with the sum of vector spaces defined as $\mathcal{A} + \mathcal{B} = \{\, \mathbf{a} + \mathbf{b} \;:\; \mathbf{a} \in \mathcal{A},\, \mathbf{b} \in \mathcal{B} \,\}$ and products of matrices and spaces defined as $\mathbf{A}\mathcal{B} = \{\, \mathbf{A}\mathbf{b} \;:\; \mathbf{b} \in \mathcal{B} \,\}$. This recursion follows directly from the architecture in appendix A.2. The first term, $\mathbb{U}^{(\ell-1)}$, accounts for the residual connection, which lets each layer's representations span the same directions as the previous layer's. Each remaining term, $\mathbf{W}_P^{(\ell,h)} \mathbf{W}_V^{(\ell,h)} \mathbb{U}^{(\ell-1)}$, is the "writing space" of attention head h: the head's update is a convex combination of the previous layer's representations mapped through $\mathbf{W}_P^{(\ell,h)} \mathbf{W}_V^{(\ell,h)}$, so its output–value matrices constrain the set of directions in which it can write information.

A.3.2 Proving the Invariance of a Simplified Space

We will prove the invariance of a simplified transformer where: (i) only the key matrices $\mathbf{W}_K^{(\ell,h)}$ and the output matrix $\bar{\mathbf{W}}$ are learnable and the others are considered fixed orthogonal matrices; (ii) with a single attention head per layer ($H = 1$); and (iii) full-dimension head ($d_h = d$). Extending the proof to multi-head attention is straightforward, and we later informally sketch a 'circuit folding' technique that generalizes our proof to a larger class of invariant spaces with arbitrary d_h and where all matrices are learnable.

For this simplified calculation thus, we consider full-dimension single-head attention layers ($H = 1$ and $d_h = d$) and we take the query matrix to be an identity matrix $\mathbf{W}_Q^{(\ell,1)} = \mathbf{I}$. (Note that this is mathematically equivalent to considering a merged key–query matrix.) Regarding projection and value matrices, we assume that they always project their input representations into orthogonal subspace, i.e. $\mathbb{U}^{(\ell-1)} \perp \mathbf{W}_P^{(\ell,1)}\mathbf{W}_V^{(\ell,1)}\mathbb{U}^{(\ell-1)}$. We further simplify notation by denoting $\mathbf{W}^{(\ell)} = \mathbf{W}_K^{(\ell,1)}$ and $\mathbf{V}^{(\ell)} = \mathbf{W}_P^{(\ell,1)}\mathbf{W}_V^{(\ell,1)}$. This leads to the following simplified self-attention formulation:

$$\mathbf{H}^{(\ell)} \;=\; \mathbf{H}^{(\ell-1)} \;+\; \mathbf{V}^{(\ell)}\,\mathbf{H}^{(\ell-1)}\sigma\left(\mathbf{H}^{(\ell-1)\intercal}\,\mathbf{W}^{(\ell)}\,\mathbf{H}^{(\ell-1)}\right), \tag{22}$$

These simplifications are in line with the *associative memory* view of Bietti et al. [17], where the large, randomly initialized matrices project their inputs into nearly-orthogonal representations. A related approach was also used by Nichani et al. [18], who analyzed a transformer with a "disentangled" residual stream [31] where the outputs of different layers are concatenated rather than added together. Note that this requires an exponentially growing residual dimensionality $d = \Theta(H^L)$.

A.3.3 Input Transformations

We now define two likelihood-preserving rotations to the token and position embeddings. For token embeddings, we will apply an association-preserving permutation $\pi \in S_{\mathcal{V}^c}$ of the common tokens. For position embeddings, we apply a frequency-wise change in position encoding offsets $\phi'_{j,b} = \phi_{j,b} + \Delta\phi_j$, with $\Delta\phi \in \mathbb{R}^m$, which also preserves associations between positions.

We define the transformations of common embeddings $\mathbf{E}^{(t)} \in \mathcal{O}(\mathbb{U}^{(\mathrm{c})})$ and position embeddings $\mathbf{E}^{(p)} \in \mathcal{O}(\mathbb{U}^{(\mathrm{p})})$ as:

$$\mathbf{E}^{(t)} \;=\; \sum_{\mathbf{x}\,\in\mathcal{V}^c}\Big(\|\mathbf{e}_{\mathbf{x}}\|_2\,\|\mathbf{e}_{\pi(\mathbf{x})}\|_2\Big)^{-1}\,\mathbf{e}_{\mathbf{x}}\,\mathbf{e}_{\pi(\mathbf{x})}{}^{\intercal} \tag{23}$$

$$\mathbf{E}^{(p)} \;=\; \sum_{j=1}^{m}\big(\cos\Delta\phi_j\big)\big(\mathbf{q}_j\,\mathbf{q}_j{}^{\intercal} + \mathbf{r}_j\,\mathbf{r}_j{}^{\intercal}\big) + \big(\sin\Delta\phi_j\big)\big(\mathbf{r}_j\,\mathbf{q}_j{}^{\intercal} - \mathbf{q}_j\,\mathbf{r}_j{}^{\intercal}\big) \tag{24}$$

Together, they define the transformed embeddings

$$\tilde{\mathbf{T}} = \mathbf{E}^{(t)}\mathbf{T}, \qquad\qquad \tilde{\mathbf{P}} = \mathbf{E}^{(p)}\mathbf{P}. \tag{25}$$

Since our permutation of common tokens preserves the associations and our offset shift maintains position correlations, we have the following commutations:

$$\mathbf{E}^{(c)}\mathbf{C} \;=\; \mathbf{C}\mathbf{E}^{(c)}, \qquad\qquad \mathbf{E}^{(p)}\mathbf{M} \;=\; \mathbf{M}\mathbf{E}^{(p)}. \tag{26}$$

Note that the token and position subspaces are orthogonal by assumption, $\mathbb{U}^{(\mathrm{t})} \perp \mathbb{U}^{(\mathrm{p})}$ (appendix A.1), so our transformations do not act outside their respective subspaces, i.e. $\mathbf{E}^{(p)}\mathbf{T} = \mathbf{E}^{(t)}\mathbf{P} = \mathbf{0}$. This allows us to define a global transformation $\Lambda^{(0)} \in \mathcal{O}(\mathbb{U}^{(0)})$ of the inputs:

$$\tilde{\mathbf{H}}^{(0)} = \Lambda^{(0)}\mathbf{H}^{(0)}, \qquad\qquad \Lambda^{(0)} = \mathbf{E}^{(t)} + \mathbf{E}^{(p)}. \tag{27}$$

A.3.4 Forward Pass

In this section, we aim to show that applying a transformation $\Lambda^{(0)} = \mathbf{E}^{(t)} + \mathbf{E}^{(p)}$ to our inputs $\mathbf{H}^{(0)}$: (i) leaves attention scores unchanged; (ii) rotates the output $\mathbf{z}$ by the same rotation $\mathbf{E}^{(t)}$; (iii) leaves the loos unchanged.

Before that, however, recall the definition of the invariant subspace, adapted to the simplified setup:

$$\mathbf{W}^{(\ell,*)} \in \mathbb{S}^{(\ell)} = \operatorname{span}\Big(\mathcal{V}_{1:\ell-1}\left(\mathcal{T}\cup\mathcal{P}\right)\mathcal{V}_{\ell-1:1}\Big), \qquad \bar{\mathbf{W}} \in \bar{\mathbb{S}} = \operatorname{span}\Big(\mathcal{T}\,\mathcal{V}_{L:1}\Big), \tag{28}$$

where we remind the reader that we defined the selection matrices and action spaces as:

$$\mathcal{T} = \{\, \mathbf{I}^{(t)}, \mathbf{C} \,\}, \qquad \mathcal{V}_{1:\ell} = \Big\{\, \mathbf{I},\, \mathbf{V}^{(\ell,h)} \;:\; h\in[H] \,\Big\}\; \mathcal{V}_{1:\ell-1}, \tag{29}$$

$$\mathcal{P} = \{\, \mathbf{I}^{(p)}, \mathbf{M}, \ldots, \mathbf{M}^{k-1} \,\}, \qquad \mathcal{V}_{\ell:1} = \{\, \mathbf{V}^\intercal \;:\; \mathbf{V}\in\mathcal{V}_{1:\ell} \,\}\,. \tag{30}$$

and that we have the following token and position association matrices

$$\mathbf{C} = \sum_{\{\mathbf{a},\mathbf{b}\}\in\mathcal{R}} \Big(\|\mathbf{e_a}\|_2\,\|\mathbf{e_b}\|_2\Big)^{-1}\,\mathbf{e_a}\,\mathbf{e_b}^\intercal \tag{31}$$

$$\mathbf{M} = \sum_{j=1}^{m} (\cos\omega_j)\left(\mathbf{q}_j\,\mathbf{q}_j^\intercal + \mathbf{r}_j\,\mathbf{r}_j^\intercal\right) + (\sin\omega_j)\left(\mathbf{q}_j\,\mathbf{r}_j^\intercal - \mathbf{r}_j\,\mathbf{q}_j^\intercal\right) \tag{32}$$

Note that $\mathcal{V}_{\ell:1}$ is defined via *transposes*, whereas the main-text definition in section 3.3 uses *pseudo-inverses*. The two coincide in this simplified setup where the fixed $\mathbf{V}^{(\ell)}$ are isometries onto orthogonal subspaces, i.e. $\mathbf{V}^+ = \mathbf{V}^\intercal$.

Attention scores are unchanged. We now aim to establish that attention scores remain unchanged under these transformations. Let $\Lambda^{(\ell)} \in \mathcal{O}(\mathbb{U}^{(\ell)})$ be a layer-wise transformation matrix, for which it holds that:

$$\tilde{\mathbf{H}}^{(0)} = \Lambda^{(0)}\mathbf{H}^{(0)} \implies \tilde{\mathbf{H}}^{(\ell)} = \Lambda^{(\ell)}\mathbf{H}^{(\ell)} \tag{33}$$

Note that for any layer, $\Lambda^{(\ell-1)}$ commutes with any $\mathbf{W}^{(\ell)} \in \mathbb{S}^{(\ell)}$. This implies that attention scores are invariant to our input transformations:

$$\tilde{\mathbf{H}}^{(\ell-1)\intercal}\mathbf{W}^{(\ell)}\tilde{\mathbf{H}}^{(\ell-1)} = \mathbf{H}^{(\ell-1)\intercal}\Lambda^{(\ell-1)\intercal}\mathbf{W}^{(\ell)}\Lambda^{(\ell-1)}\mathbf{H}^{(\ell-1)} = \mathbf{H}^{(\ell-1)\intercal}\mathbf{W}^{(\ell)}\mathbf{H}^{(\ell-1)} \tag{34}$$

Outputs are rotated. We can now show the layer-wise transformation matrices can be defined recursively as $\Lambda^{(\ell)} = \Lambda^{(\ell-1)} + \mathbf{V}^{(\ell)}\Lambda^{(\ell-1)}\mathbf{V}^{(\ell)\intercal}$. Expanding the definition of $\tilde{\mathbf{H}}^{(\ell)}$ we get:

$$\tilde{\mathbf{H}}^{(\ell)} = \tilde{\mathbf{H}}^{(\ell-1)} + \mathbf{V}^{(\ell)}\tilde{\mathbf{H}}^{(\ell-1)}\sigma\left(\tilde{\mathbf{H}}^{(\ell-1)\intercal}\mathbf{W}^{(\ell)}\tilde{\mathbf{H}}^{(\ell-1)}\right) \qquad \text{definition of } \tilde{\mathbf{H}}^{(\ell)} \tag{35a}$$

$$= \Lambda^{(\ell-1)}\mathbf{H}^{(\ell-1)} + \mathbf{V}^{(\ell)}\Lambda^{(\ell-1)}\mathbf{H}^{(\ell-1)}\sigma\left(\mathbf{H}^{(\ell-1)\intercal}\mathbf{W}^{(\ell)}\mathbf{H}^{(\ell-1)}\right) \tag{35b}$$

$$= \left(\Lambda^{(\ell-1)} + \mathbf{V}^{(\ell)}\Lambda^{(\ell-1)}\mathbf{V}^{(\ell)\intercal}\right)\mathbf{H}^{(\ell)} \qquad \text{by } \mathbb{U}^{(\ell-1)} \perp \mathbf{V}^{(\ell)}\mathbb{U}^{(\ell-1)} \tag{35c}$$

$$= \Lambda^{(\ell)}\mathbf{H}^{(\ell)} \tag{35d}$$

which shows that $\Lambda^{(\ell)} = \Lambda^{(\ell-1)} + \mathbf{V}^{(\ell)}\Lambda^{(\ell-1)}\mathbf{V}^{(\ell)\intercal}$. The final layer $\bar{\mathbf{W}} \in \bar{\mathbb{S}}$ projects $\mathbf{H}^{(L)}$ back into $\mathbb{U}^{(t)}$, which gives $\tilde{\bar{\mathbf{H}}} = \mathbf{E}^{(t)}\bar{\mathbf{H}}$ and $\tilde{\mathbf{z}} = \mathbf{E}^{(t)}\mathbf{z}$.

Loss is unchanged. Finally, we note that, under our transformations, the target output $\mathbf{y}$ are rotated by the same transformation as the inputs: $\tilde{\mathbf{y}} = \mathbf{E}^{(t)}\mathbf{y}$. This fact, combined with the previous result imply the loss is unchanged $\tilde{\mathcal{L}} = \mathcal{L}$.

A.3.5 Backward Pass

We now backward propagate through our model, and show how the previous transformation affects our gradients. We also show here that the population gradients are preserved under such a transformation.

Transformed Gradients. We start by the final softmax and unembedding layers. Propagating the loss gradient through them, we get

$$\frac{\partial\tilde{\mathcal{L}}}{\partial\tilde{\bar{\mathbf{H}}}} = \mathbf{E}^{(t)}\frac{\partial\mathcal{L}}{\partial\bar{\mathbf{H}}} \tag{36}$$

Second, the gradient of the final linear layer weights will be:

$$\frac{\partial \tilde{\mathcal{L}}}{\partial \bar{\mathbf{W}}} \;=\; \tilde{\mathbf{H}}^{(L)} \left(\frac{\partial \tilde{\mathcal{L}}}{\partial \tilde{\bar{\mathbf{H}}}} \right)^{\intercal} \;=\; \Lambda^{(L)}\, \mathbf{H}^{(L)}\, \frac{\partial \mathcal{L}}{\partial \bar{\mathbf{H}}}\, \mathbf{E}^{(t)\intercal} \;=\; \Lambda^{(L)}\, \frac{\partial \mathcal{L}}{\partial \bar{\mathbf{W}}}\, \mathbf{E}^{(t)\intercal} \tag{37}$$

A key ingredient of the proof is that attention scores are invariant to our input transformations: association-preserving permutations don't affect the attention between a token and itself or its associated token, which are the only allowed token actions; likewise, the attention between different positions is not affected by a global offset change. Since the attention scores are invariant to our transformations, their gradients are also invariant. For layers $\ell \in [L]$, we get:

$$\frac{\partial \tilde{\mathcal{L}}}{\partial \mathbf{W}^{(\ell)}} \;=\; \tilde{\mathbf{H}}^{(\ell-1)}\, \frac{\partial \tilde{\mathcal{L}}}{\partial\, \sigma\left(\tilde{\mathbf{H}}^{(\ell-1)\intercal} \mathbf{W}^{(\ell)} \tilde{\mathbf{H}}^{(\ell-1)} \right)}\, \tilde{\mathbf{H}}^{(\ell-1)\intercal} \tag{38a}$$

$$\;=\; \Lambda^{(\ell-1)}\, \mathbf{H}^{(\ell-1)}\, \frac{\partial \tilde{\mathcal{L}}}{\partial\, \sigma\left(\mathbf{H}^{(\ell-1)\intercal} \mathbf{W}^{(\ell)} \mathbf{H}^{(\ell-1)} \right)}\, \mathbf{H}^{(\ell-1)\intercal}\, \Lambda^{(\ell-1)\intercal} \tag{38b}$$

$$\;=\; \Lambda^{(\ell-1)}\, \frac{\partial \mathcal{L}}{\partial \mathbf{W}^{(\ell)}}\, \Lambda^{(\ell-1)\intercal} \tag{38c}$$

Population Gradients. We now note that, since our transformations (by construction) preserve the likelihood of the input–output pairs, they result in an identical data distribution. As the loss is also preserved under transformation, this implies that the population gradient is invariant to our transformations:

$$\mathbb{E}\left[\frac{\partial \tilde{\mathcal{L}}}{\partial \mathbf{W}^{(\ell)}} \right] \;=\; \frac{\partial \mathbb{E}\big[\tilde{\mathcal{L}}\big]}{\partial \mathbf{W}^{(\ell)}} \;=\; \frac{\partial \mathbb{E}\big[\mathcal{L}\big]}{\partial \mathbf{W}^{(\ell)}} \;=\; \mathbb{E}\left[\frac{\partial \mathcal{L}}{\partial \mathbf{W}^{(\ell)}} \right]. \tag{39}$$

A.3.6 Gradient Blocks

Finally, in this section we conclude our proof that our model's gradient will live inside the constructed manifold. To that end, we first analyze which matrices are invariant with respect to the studied transformations. Rather than analyzing the gradient directly, we decompose it into blocks indexed by pairs of concepts (tokens or positions) and pairs of layer subsets, i.e., by the possible interaction paths through the network. This decomposition is convenient because each block is easier to treat in isolation, and all blocks follow the same symmetry argument. Moreover, it loses nothing: the residual-stream activations are supported entirely on the concept subspaces and their images under the composite value maps, so any gradient block not considered below is zero, since the corresponding activations are zero.

For a subset of layers $S \subseteq [L]$, we define the product of the corresponding value matrices, taken in descending layer order (later layers on the left), matching the composites in $\mathcal{V}_{1:L}$:

$$\mathbf{V}^{(S)} \;=\; \prod_{\ell \,\in\, S} \mathbf{V}^{(\ell)} \tag{40}$$

For any layer $\mathbf{W}^{(\ell)}$ or $\bar{\mathbf{W}}$, any two layer subsets $S_1, S_2 \subseteq [L]$, and any two embedding 'concepts' $\mathbf{x_1}, \mathbf{x_2} \in \{\mathrm{t}, \mathrm{p}\}$ (i.e., for either tokens or positions), the corresponding gradient block satisfies:

$$\mathbf{I}^{(\mathbf{x_1})}\, \mathbf{V}^{(S_1)\intercal}\, \frac{\partial \mathbb{E}\big[\mathcal{L}\big]}{\partial \mathbf{W}^{(\ell)}}\, \mathbf{V}^{(S_2)}\, \mathbf{I}^{(\mathbf{x_2})} \;=\; \mathbf{E}^{(\mathbf{x_1})\intercal} \left(\mathbf{I}^{(\mathbf{x_1})}\, \mathbf{V}^{(S_1)\intercal}\, \frac{\partial \mathbb{E}\big[\mathcal{L}\big]}{\partial \mathbf{W}^{(\ell)}}\, \mathbf{V}^{(S_2)}\, \mathbf{I}^{(\mathbf{x_2})} \right) \mathbf{E}^{(\mathbf{x_2})} \tag{41}$$

Recall that our embedding transformations, $\mathbf{E}^{(t)}$, and $\mathbf{E}^{(p)}$, can be chosen independently. Moreover, by shifting every position offset by $\Delta\phi' = \Delta\phi + \pi$, we obtain $\mathbf{E}^{(p)'} = -\mathbf{E}^{(p)}$. This implies that any gradient block mapping different concepts must be zero:

$$\mathbf{I}^{(t)}\, \mathbf{V}^{(S_1)\intercal}\, \frac{\partial \mathbb{E}\big[\mathcal{L}\big]}{\partial \mathbf{W}^{(\ell)}}\, \mathbf{V}^{(S_2)}\, \mathbf{I}^{(p)} \;=\; \mathbf{0} \tag{42}$$

We are left with the gradient blocks mapping two rotations of the same embedding space. Because the expected loss $\mathbb{E}[\mathcal{L}]$ is computed over a data distribution that is invariant to our embedding transformations, the expected gradients must commute with those transformations.

Let $\mathbf{G}^{(p)} = \mathbf{I}^{(p)}\mathbf{V}^{(S_1)\intercal}\frac{\partial \mathbb{E}\left[\mathcal{L}\right]}{\partial \mathbf{W}^{(\ell)}}\mathbf{V}^{(S_2)}\mathbf{I}^{(p)}$ and $\mathbf{G}^{(t)} = \mathbf{I}^{(t)}\mathbf{V}^{(S_1)\intercal}\frac{\partial \mathbb{E}\left[\mathcal{L}\right]}{\partial \mathbf{W}^{(\ell)}}\mathbf{V}^{(S_2)}\mathbf{I}^{(t)}$. By the chain rule and the invariance of the loss, the gradient blocks must commute with any valid transformation matrix:

$$\mathbf{G}^{(p)}\mathbf{E}^{(p)} = \mathbf{E}^{(p)}\mathbf{G}^{(p)} \quad \forall \mathbf{E}^{(p)} \in \mathcal{O}(\mathbb{U}^{(\mathrm{p})}), \tag{43}$$

$$\mathbf{G}^{(t)}\mathbf{E}^{(t)} = \mathbf{E}^{(t)}\mathbf{G}^{(t)} \quad \forall \mathbf{E}^{(t)} \in \mathcal{O}(\mathbb{U}^{(\mathrm{c})}). \tag{44}$$

We can analyze these two constraints using standard linear algebra and symmetry properties.

Token Subspace. The transformation $\mathbf{E}^{(t)}$ represents permutations of common tokens that strictly preserve associations (e.g., swapping one pair of associated tokens with another, or swapping tokens within a pair).

Let us view $\mathbf{G}^{(t)}$ as a weighted adjacency matrix connecting tokens in $\mathbb{U}^{(\mathrm{c})}$. For $\mathbf{G}^{(t)}$ to commute with all valid permutations $\mathbf{E}^{(t)}$, the weights must be symmetric and uniform across all interchangeable token relationships. Because any token pair can be swapped with any other pair, the diagonal entries of $\mathbf{G}^{(t)}$ (self-connections) must all be identical. Similarly, because tokens within a pair can be swapped, the entries connecting a token to its associated pair must all be identical. Any connection between tokens belonging to *different* pairs would break the permutation symmetry unless all such connections were identical.

Therefore, $\mathbf{G}^{(t)}$ must have all entries equal except for the diagonal and the off-diagonal entries corresponding to valid associations, i.e. $\mathbf{G}^{(t)} = c_1\mathbf{1}^{(t)} + c_2\mathbf{I}^{(t)} + c_3\mathbf{C}$. The bias term $\mathbf{1}^{(t)}$ of the gradient vanishes ($c_1 \approx 0$) when all tokens have the same L1 norm in the basis of common tokens. In our setup, all common tokens have the same L2 norm (i.e. $\|\mathbf{e}_\mathbf{x}\|_2 = \text{const.}$), which ensures equal L1 norm in the token basis. Furthermore, lex-invariant rare token embeddings are generated with the same mean and variance as common tokens, which ensures nearly-equal L1 norm when the dimensionality is large.

Positional Subspace. The transformation $\mathbf{E}^{(p)}$ represents a continuous shift in the positional offsets. By definition of our sinusoidal embeddings, $\mathbb{U}^{(\mathrm{p})}$ is spanned by orthogonal 2D subspaces corresponding to the fundamental frequencies. In this basis, any rotation $\mathbf{E}^{(p)}(\Delta\phi)$ is a block-diagonal matrix where each 2×2 block is a standard 2D rotation matrix.

For $\mathbf{G}^{(p)}$ to commute with $\mathbf{E}^{(p)}(\Delta\phi)$ for *all* continuous values of $\Delta\phi$, standard matrix algebra dictates that $\mathbf{G}^{(p)}$ must also be block-diagonal in the exact same basis, and its 2×2 blocks must be scaled 2D rotation matrices themselves. Recall that we considered that all frequencies have order k, i.e. $\omega_j = 2\pi N_j/k$, but there are fundamentally only k such distinct frequencies. Permuting the offsets of equivalent frequencies preserves the output and loss of the model, hence the gradient sub-block rotations corresponding to equivalent frequencies must be equal. Therefore, the gradient contains at most k distinct sub-blocks, each of which is a 2×2 scaled rotation.

A polynomial of order k in $\mathbf{M}$ (i.e., $\mathbf{G}^{(p)} = c_0\mathbf{I}^{(p)} + c_1\mathbf{M} + \ldots + c_{k-1}\mathbf{M}^{k-1}$) requires one additional constraint: sub-blocks corresponding to opposite frequencies (i.e., ω and $\omega' = 2\pi - \omega$) must be the transpose of each other (i.e., scaled rotations with the same scaling factors but opposite rotations). To establish this, we must examine the structure of the gradient in more detail. Note that $\mathbf{G}^{(p)}$ is the gradient block of an attention matrix $\mathbf{W}^{(\ell)}$, which is obtained as:

$$\frac{\partial \mathcal{L}}{\partial \mathbf{W}^{(\ell)}} \;=\; \mathbf{H}^{(\ell-1)}\,\frac{\partial \mathcal{L}}{\partial A}\,\mathbf{H}^{(\ell-1)\intercal}, \qquad \text{where } A = \sigma\left(\mathbf{H}^{(\ell-1)\intercal}\,\mathbf{W}^{(\ell)}\,\mathbf{H}^{(\ell-1)}\right). \tag{45}$$

In particular, our gradient block will be $\mathbf{G}^{(p)} = \mathbb{E}[\sum_{ij}(\partial\mathcal{L}/\partial A_{ij})\,p_i p_j{}^\intercal]$, where $p_i, p_j \in \mathbb{U}^{(\mathrm{p})}$ are the positional components at two locations $i, j \in [nk]$ in the sequence. Since attention score gradients $\partial\mathcal{L}/\partial A_{ij}$ are shared across frequencies, it suffices to focus on a single attention path. Isolating the 2×2 gradient sub-block corresponding to a frequency ω, we get $\mathbf{G}_{ij\omega} = \mathbb{E}[p_{\omega i}p_{\omega j}{}^\intercal]$. Further using the fact that $p_{\omega i} = \mathbf{M}_\omega^{i-j}p_{\omega j} = \mathbf{M}_{-\omega}^{i-j\intercal}p_{\omega j}$ and taking the expectation over global offsets, which preserve attention scores and their gradients, we obtain that opposite-frequency sub-blocks are transpositions of each other.

By combining the structural constraints of both the token and positional subspaces, we recover the desired geometry for all gradient blocks.

Multi-Head Attention. The above proof can be easily extended to multi-head attention:

$$\mathbf{H}^{(\ell)} \;=\; \mathbf{H}^{(\ell-1)} \;+\; \sum_{h=1}^{H} \mathbf{V}^{(\ell,h)}\, \mathbf{H}^{(\ell-1)} \sigma\left(\mathbf{H}^{(\ell-1)\intercal}\, \mathbf{W}^{(\ell,h)}\, \mathbf{H}^{(\ell-1)}\right). \tag{46}$$

By applying the same input transformations as before, we obtain slightly more complex global transformations $\Lambda^{(\ell)} = \Lambda^{(\ell-1)} + \sum_{h=1}^{H} \mathbf{V}^{(\ell,h)} \Lambda^{(\ell-1)} \mathbf{V}^{(\ell,h)\intercal}$, where the orthogonality of the per-head write subspaces (Assumption 6) keeps the heads' contributions independent. Looking at the gradient blocks indexed by concepts and sets of heads, the desired structure can be established using identical arguments.

A.4 Circuit Folding: Invariant Dynamics with Interacting Subspaces

Theorem 1 assumes fixed orthogonal output–value maps (Assumption 6); relaxing this is more delicate, and we record the extension here as a sketch rather than a proven claim. Allowing the value maps $\mathbf{V}^{(\ell,h)}$ to be learned lets concept subspaces rotate or mix, and allowing the key–query and output–value factorizations to be learned independently (rather than only through their merged products, Assumption 5) requires tracking the intermediate representations rather than just those products. We expect the same block-level arguments to extend to factorizations in which subspaces interact only within a single concept (tokens or positions) and are transformed only by their fundamental actions; this appendix sets up the circuit-folding formalism needed to make this precise. We emphasize that this is a sketch, and leave a complete treatment to future work.

To illustrate the idea, consider any learnable matrix $\mathbf{W} \in \mathbb{R}^{d_2 \times d_1}$ of the network. For any orthogonal input and output concept spaces, we can establish the following invariant manifold:

$$\mathbf{W} \;\in\; \mathrm{span}(\mathcal{B}_t^{o\intercal} \mathcal{T} \mathcal{B}_t^i + \mathcal{B}_p^{o\intercal} \mathcal{P} \mathcal{B}_p^i) \tag{47}$$

where $\mathcal{B}_t^i \subset \mathbb{R}^{d_t \times d_1}, \mathcal{B}_t^o \subset \mathbb{R}^{d_t \times d_2}, \mathcal{B}_p^i \subset \mathbb{R}^{d_p \times d_1}, \mathcal{B}_p^o \subset \mathbb{R}^{d_p \times d_2}$ denote the sets of orthogonal basis vectors for the orthogonal input/output token/position spaces.

Consider a sequence of representations $\mathbf{H} \in \mathbb{R}^{d_i \times n}$ that we transform by applying the same input transformation as before. Each token space is transformed with $\mathbf{E}^{(t)} \in \mathcal{O}(\mathbb{U}^{(\mathrm{c})})$ and each position space is transformed with $\mathbf{E}^{(p)} \in \mathcal{O}(\mathbb{U}^{(\mathrm{p})})$. We define the global input transformation:

$$\Lambda_i = \sum_{\mathbf{B} \in \mathcal{B}_t^i} \mathbf{B}^\intercal \mathbf{E}^{(t)} \mathbf{B} \;+\; \sum_{\mathbf{B} \in \mathcal{B}_p^i} \mathbf{B}^\intercal \mathbf{E}^{(p)} \mathbf{B}. \tag{48}$$

Since output subspaces are connected only to input subspaces of the same concept via transformations that commute with the input transformations, we obtain the following output transformation:

$$\mathbf{W}\Lambda_i \mathbf{H} \;=\; \Lambda_o \mathbf{W} \mathbf{H}, \qquad\qquad \Lambda_o = \sum_{\mathbf{B} \in \mathcal{B}_t^o} \mathbf{B}^\intercal \mathbf{E}^{(t)} \mathbf{B} \;+\; \sum_{\mathbf{B} \in \mathcal{B}_p^o} \mathbf{B}^\intercal \mathbf{E}^{(p)} \mathbf{B}. \tag{49}$$

Similarly, if the output gradient is transformed via Λ_o, the propagated input gradient will be transformed via Λ_i. Computing the transformed gradient blocks of $\mathbf{W}$ and applying the same reasoning as in the original proof, we can show that the gradient of $\mathbf{W}$ fits the desired structure.

A.5 Beyond Attention Only: Feed-Forward Networks & Layer Norm

We extend Corollary 1 to standard Transformer layers, which interleave multi-head attention with two further operations: position-wise feed-forward networks (FFN) and layer normalization (LN). Both operations commute with the embedding-space symmetries used in appendix A.3, provided their parameters fit the natural extension of the IMIR structure described below. The argument is the same in spirit as the multi-head extension of section A.3.6: one writes down a global residual-stream rotation $\Lambda^{(\ell)}$, checks that the new operation commutes with it, and propagates the block-level analysis through the new gradient blocks.

We adopt the notation of appendix A.3: $\mathbf{E}^{(t)} \in \mathcal{O}(\mathbb{U}^{(\mathrm{c})})$ is the orthogonal map induced by an association-preserving permutation of common tokens, $\mathbf{E}^{(p)} \in \mathcal{O}(\mathbb{U}^{(\mathrm{p})})$ is the rotation on the position subspace induced by a global offset shift, and $\Lambda^{(\ell)}$ is the composite rotation on the residual stream at layer ℓ.

Layer Normalization. A LayerNorm at layer ℓ rescales each column of the residual stream:

$$\mathrm{LN}(\mathbf{H}^{(\ell)})_{:,i} \;=\; \mathbf{g} \odot \frac{\mathbf{H}^{(\ell)}{}_{:,i} - \mu_i \mathbf{1}}{\sigma_i} \;+\; \mathbf{b}, \tag{50}$$

where $\mu_i, \sigma_i \in \mathbb{R}$ are the column mean and standard deviation and $\mathbf{g}, \mathbf{b} \in \mathbb{R}^d$ are learnable gain and bias vectors. Two facts make LN compatible with $\Lambda^{(\ell-1)}$.

1. The column statistics μ_i, σ_i depend on the residual column only through inner products with the all-ones direction $\mathbf{1}$ and with itself. Both $\mathbf{E}^{(t)}$ and $\mathbf{E}^{(p)}$ act on $\mathbb{U}^{(\mathrm{c})}$ and $\mathbb{U}^{(\mathrm{p})}$ respectively, and these subspaces are orthogonal to $\mathbf{1}$ in our setup (common-token and sinusoidal position embeddings are zero-mean by construction). Hence μ_i and σ_i are unchanged by $\Lambda^{(\ell-1)}$.
2. Provided the gain and bias decompose into the same fundamental actions that span the IMIR — i.e. $\mathbf{g}, \mathbf{b} \in \mathrm{span}(\mathcal{T} \cup \mathcal{P})$ applied to $\mathbf{1}$, with the special case $\mathbf{g} = \mathbf{1}$, $\mathbf{b} = \mathbf{0}$ recovering RMSNorm — the elementwise multiplication and addition commute with $\Lambda^{(\ell-1)}$ in the same block-diagonal sense as $\mathbf{C}$ and $\mathbf{M}$ in the proof of Corollary 1.

Together,

$$\mathrm{LN}(\Lambda^{(\ell-1)}\mathbf{H}^{(\ell-1)}) \;=\; \Lambda^{(\ell-1)}\,\mathrm{LN}(\mathbf{H}^{(\ell-1)}), \tag{51}$$

and the gradient of LN with respect to its input and to $\mathbf{g}, \mathbf{b}$ inherits the block structure of appendix A.3 verbatim.

Feed-Forward Networks. Each FFN is a position-wise two-layer MLP,

$$\mathrm{FFN}(\mathbf{H}^{(\ell)})_{:,i} \;=\; W^{(\mathrm{out})}\,\sigma\big(W^{(\mathrm{in})}\mathbf{H}^{(\ell)}{}_{:,i}\big), \qquad W^{(\mathrm{in})} \in \mathbb{R}^{d_h \times d},\; W^{(\mathrm{out})} \in \mathbb{R}^{d \times d_h}, \tag{52}$$

with σ an elementwise non-linearity (e.g. GELU). Biases are omitted for brevity; they are handled like the LN bias above.

Define the FFN counterpart of the IMIR by applying the circuit-folding construction of appendix A.4 to each FFN matrix. $W^{(\mathrm{in})}$ maps the residual stream into the FFN-hidden space: let $\mathcal{B}^o_t, \mathcal{B}^o_p$ denote the orthogonal basis matrices of the token-only and position-only blocks of the FFN-hidden axes (with the FFN-hidden dimension partitioned as $d_h = d_h^{(t)} + d_h^{(p)}$). $W^{(\mathrm{out})}$ maps back, with the same FFN-hidden bases playing the role of input. We require

$$W^{(\mathrm{in})} \;\in\; \mathrm{span}\big(\mathcal{B}_t^{o\intercal}\,\mathcal{T}\,\mathcal{V}_{\ell:1} \;+\; \mathcal{B}_p^{o\intercal}\,\mathcal{P}\,\mathcal{V}_{\ell:1}\big), \tag{53}$$

$$W^{(\mathrm{out})} \;\in\; \mathrm{span}\big(\mathcal{V}_{1:\ell}\,\mathcal{T}\,\mathcal{B}_t^{o} \;+\; \mathcal{V}_{1:\ell}\,\mathcal{P}\,\mathcal{B}_p^{o}\big), \tag{54}$$

which is exactly the circuit-folding form $\mathbf{W} \in \mathrm{span}(\mathcal{B}_t^{o\intercal}\mathcal{T}\mathcal{B}_t^{i} + \mathcal{B}_p^{o\intercal}\mathcal{P}\mathcal{B}_p^{i})$ of appendix A.4, instantiated with $\mathcal{B}_t^i = \mathcal{V}_{\ell:1}$ and $\mathcal{B}_p^i = \mathcal{V}_{\ell:1}$ for $W^{(\mathrm{in})}$, and with the roles of input and output swapped for $W^{(\mathrm{out})}$.

Forward invariance. Define the FFN-hidden rotation $\Lambda^{(\ell)^{(\mathrm{hid})}} = \mathcal{B}_t^{o\intercal}\mathbf{E}^{(t)}\mathcal{B}_t^{o} + \mathcal{B}_p^{o\intercal}\mathbf{E}^{(p)}\mathcal{B}_p^{o}$, matching $\Lambda^{(\ell-1)}$ on the residual side. By the circuit-folding lemma (appendix A.4), $W^{(\mathrm{in})}\Lambda^{(\ell-1)} = \Lambda^{(\ell)^{(\mathrm{hid})}}W^{(\mathrm{in})}$ and, by the symmetric statement for $W^{(\mathrm{out})}$, $W^{(\mathrm{out})}\Lambda^{(\ell)^{(\mathrm{hid})}} = \Lambda^{(\ell-1)}W^{(\mathrm{out})}$. The non-linearity σ is applied coordinate-wise within each FFN-hidden sub-block. Because $\Lambda^{(\ell)^{(\mathrm{hid})}}$ permutes coordinates only within the token sub-block (via $\mathbf{E}^{(t)}$, which is a permutation of the orthonormal token basis vectors) and only within the position sub-block (via the block-diagonal $\mathbf{E}^{(p)}$), and because σ is shared across coordinates of the same sub-block, σ commutes with $\Lambda^{(\ell)^{(\mathrm{hid})}}$. Composing,

$$\mathrm{FFN}(\Lambda^{(\ell-1)}\mathbf{H}^{(\ell-1)}) \;=\; \Lambda^{(\ell-1)}\,\mathrm{FFN}(\mathbf{H}^{(\ell-1)}), \tag{55}$$

so that the residual update $\mathbf{H}^{(\ell)} = \mathbf{H}^{(\ell-1)} + \mathrm{FFN}(\mathbf{H}^{(\ell-1)})$ also commutes with $\Lambda^{(\ell-1)}$.

Backward invariance. The gradients

$$\frac{\partial \mathcal{L}}{\partial W^{(\mathrm{out})}} \;=\; \frac{\partial \mathcal{L}}{\partial \mathbf{H}^{(\ell)}}\,\big(\sigma(W^{(\mathrm{in})}\mathbf{H}^{(\ell-1)})\big)^{\intercal}, \tag{56}$$

$$\frac{\partial \mathcal{L}}{\partial W^{(\mathrm{in})}} \;=\; \Big(W^{(\mathrm{out})\intercal}\,\frac{\partial \mathcal{L}}{\partial \mathbf{H}^{(\ell)}} \;\odot\; \sigma'(W^{(\mathrm{in})}\mathbf{H}^{(\ell-1)})\Big)\,\mathbf{H}^{(\ell-1)\intercal} \tag{57}$$

are outer products in which one factor lives in the residual stream (rotated by $\Lambda^{(\ell-1)}$) and the other in the FFN-hidden space (rotated by $\Lambda^{(\ell)^{(\mathrm{hid})}}$). The same gradient-block analysis as in appendix A.3 (cross-concept blocks vanish under $\mathbf{E}^{(p)} \mapsto -\mathbf{E}^{(p)}$; same-concept blocks commute with $\mathbf{E}^{(t)}$ resp. $\mathbf{E}^{(p)}$) implies that the expected gradient blocks are polynomials in the fundamental actions and therefore fit the prescribed FFN-IMIR structure.

Putting it together. Inserting any number of FFN and LN layers, in any order, between the attention layers does not break Corollary 1. The extended manifold consists of the original attention-weight constraints of appendix A.3, plus the FFN-weight constraints above, plus the LN gain/bias constraints. The forward pass commutes with $\Lambda^{(\ell)}$ at every layer (attention, FFN, or LN); the loss is therefore invariant under our embedding rotations; the expected gradient is invariant too; and the same block-by-block argument forces each gradient block — attention, FFN, or LN — into the prescribed polynomial form. The full Transformer thus retains the same low-dimensional invariant manifold as the attention-only model.

B Arbitrary Relational Structure

In the main text we specialized the discussion to irreflexive symmetric binary relations, which already captures the bulk of the inductive tasks studied in the literature. The framework, however, applies to arbitrary relational structures on the vocabulary. In this appendix we describe the general construction and then work out the directed pairwise case as a concrete extension. The two ingredients are always the same: a relation $\mathcal{R}$ induces a token symmetry group, the symmetry group induces orthogonal transformations on representations, and the corresponding commutant pins down the admissible operators that learning dynamics can produce.

B.1 Automorphism Groups of General Relations

Let

$$\mathcal{R} \subseteq \mathcal{V}^{\rho} \tag{58}$$

be an ρ-ary relation on the vocabulary $\mathcal{V} = \{1, \dots, m\}$, characterizing the basic relational skills required by the task. Examples include binary associations, equivalence relations, transition structures of a Markov chain, or higher-order dependencies. The relation $\mathcal{R}$ induces a subgroup of the symmetric group, the *automorphism group* of $\mathcal{R}$:

$$\begin{aligned} \mathrm{Aut}(\mathcal{V}, \mathcal{R}) &:= \big\{\pi \in S_m \ : \ \mathbf{w} \in \mathcal{R} \iff \pi \circ \mathbf{w} \in \mathcal{R}\big\}, \\ \pi \circ (w_1, \dots, w_\rho) &:= (\pi(w_1), \dots, \pi(w_\rho)). \end{aligned} \tag{59}$$

Its elements are precisely the permutations of $\mathcal{V}$ that preserve the relational structure of $\mathcal{R}$.

B.2 Orthogonal Action on Representations

To translate vocabulary symmetries into representation-space symmetries we exploit the orthogonality of the embedding matrix. As in the main text we assume $\mathbf{E}^\intercal \mathbf{E} = \mathbf{I}$ and, for simplicity of exposition, $d = m$, so that no embedding dimensions are redundant. Every permutation $\pi \in \mathrm{Aut}(\mathcal{V}, \mathcal{R})$ induces a permutation matrix $\mathbf{P}_\pi \in \mathbb{R}^{m \times m}$ acting on basis vectors by $\mathbf{P}_\pi \mathbf{e}_i = \mathbf{e}_{\pi(i)}$, and the corresponding orthogonal action on representations is

$$\mathbf{U}_\pi \ = \ \mathbf{E}\mathbf{P}_\pi\mathbf{E}^\intercal, \qquad \mathbf{U}_\pi \mathbf{e}_i \ = \ \mathbf{e}_{\pi(i)}. \tag{60}$$

The induced symmetry group on representation space is

$$\mathcal{U} := \big\{\mathbf{U}_\pi \ : \ \pi \in \mathrm{Aut}(\mathcal{V}, \mathcal{R})\big\} \ \subseteq \ O(d), \tag{61}$$

and the admissible operators are exactly those that commute with every element of $\mathcal{U}$:

$$\mathcal{C}(\mathcal{U}) := \big\{\mathbf{W} \in \mathbb{R}^{d \times d} \ : \ \mathbf{W}\mathbf{U} = \mathbf{U}\mathbf{W} \ \ \forall\, \mathbf{U} \in \mathcal{U}\big\}. \tag{62}$$

By the gradient-invariance argument of section 3.3, the population gradient flow preserves $\mathcal{C}(\mathcal{U})$, so $\mathcal{C}(\mathcal{U})$ provides the elementary admissible building blocks of any reasoning circuit, regardless of the arity or structure of $\mathcal{R}$.

B.3 Directed Pairwise Relations

To illustrate the construction beyond the symmetric pairwise case treated in the main text, consider an *ordered* (directed) binary relation. Write

$$\mathcal{R} = \{(\mathbf{a}_1, \mathbf{b}_1), \dots, (\mathbf{a}_k, \mathbf{b}_k)\} \ \subseteq \ \mathcal{V} \times \mathcal{V}, \tag{63}$$

where every vocabulary item appears at most once, and let

$$A = \{\mathbf{a}_1, \dots, \mathbf{a}_k\}, \qquad B = \{\mathbf{b}_1, \dots, \mathbf{b}_k\}, \qquad D = \mathcal{V} \setminus (A \cup B) \tag{64}$$

denote the sources, the targets, and the unrelated tokens. The automorphisms of $\mathcal{R}$ jointly permute the directed pairs while preserving orientation, and independently permute the unrelated tokens:

$$\mathbf{a}_i \mapsto \mathbf{a}_{\sigma(i)}, \qquad \mathbf{b}_i \mapsto \mathbf{b}_{\sigma(i)}, \qquad d_j \mapsto d_{\tau(j)}, \qquad \sigma \in S_k, \ \tau \in S_{m-2k}. \tag{65}$$

Because sources and targets cannot be exchanged, the admissible operators now distinguish four pair-local roles. Define the role-preserving projectors and role-switching couplings

$$\mathbf{I}_A := \sum_{i=1}^{k} \mathbf{e}_{\mathbf{a}_i}\mathbf{e}_{\mathbf{a}_i}{}^{\mathsf{T}}, \qquad \mathbf{I}_B := \sum_{i=1}^{k} \mathbf{e}_{\mathbf{b}_i}\mathbf{e}_{\mathbf{b}_i}{}^{\mathsf{T}}, \qquad \mathbf{I}_D := \sum_{d\in D} \mathbf{e}_d\mathbf{e}_d{}^{\mathsf{T}}, \tag{66}$$

$$\mathbf{C}_{A\to B} := \sum_{i=1}^{k} \mathbf{e}_{\mathbf{b}_i}\mathbf{e}_{\mathbf{a}_i}{}^{\mathsf{T}}, \qquad \mathbf{C}_{B\to A} := \sum_{i=1}^{k} \mathbf{e}_{\mathbf{a}_i}\mathbf{e}_{\mathbf{b}_i}{}^{\mathsf{T}}. \tag{67}$$

A direct orbit count shows that, after centering the embeddings within each symmetry class so that the global rank-one couplings drop out, the commutant reduces to the pair-local algebra

$$\mathcal{C}(\mathcal{U}) \;=\; \mathrm{span}\big\{\mathbf{I}_A,\ \mathbf{I}_B,\ \mathbf{I}_D,\ \mathbf{C}_{A\to B},\ \mathbf{C}_{B\to A}\big\}. \tag{68}$$

This generalizes the unordered case in two ways. First, the identity $\mathbf{I} = \mathbf{I}_A + \mathbf{I}_B + \mathbf{I}_D$ splits into separate role projectors, so attention heads can now condition on whether a token plays the role of a source, of a target, or of an unrelated filler. Second, the symmetric association operator $\mathbf{C} = \mathbf{C}_{A\to B} + \mathbf{C}_{B\to A}$ of the main text splits into two oriented operators, allowing the circuit algebra to encode directional relations such as $\mathbf{a} \to \mathbf{b}$ without collapsing onto its inverse.

Before centering, the full commutant additionally contains uniform coupling operators of the form $\mathbf{J}_{RS} := \sum_{r\in R}\sum_{s\in S} \mathbf{e}_r\mathbf{e}_s{}^{\mathsf{T}}$, for role types $R, S \in \{A, B, D\}$. These operators correspond to nonzero mean directions of the symmetry classes; centering the embeddings within each class removes them, and the local algebra above is recovered as the essential symmetry-compatible structure. The construction of the basis weights $\mathcal{W}_\ell$ in section 3.3 then proceeds verbatim, with the token selection set $\mathcal{T} = \{\mathbf{I}^{(t)}, \mathbf{C}\}$ replaced by the directed selection set $\{\mathbf{I}_A, \mathbf{I}_B, \mathbf{I}_D, \mathbf{C}_{A\to B}, \mathbf{C}_{B\to A}\}$, and the invariant manifold theorem (Corollary 1) carries over without further change.

B.4 General Recipe

The two examples treated so far – symmetric pairwise relations in the main text and directed pairwise relations above – follow the same recipe, and the recipe extends to arbitrary $\mathcal{R}$:

1. Compute the automorphism group $\mathrm{Aut}(\mathcal{V}, \mathcal{R})$ of the relation, and partition $\mathcal{V}$ into orbits of role-equivalent tokens.
2. Form the induced orthogonal symmetry group $\mathcal{U} \subseteq O(d)$ via $\mathbf{U}_\pi = \mathbf{E}\mathbf{P}_\pi\mathbf{E}^{\mathsf{T}}$.
3. Compute the commutant $\mathcal{C}(\mathcal{U})$. Its dimension equals the number of orbits of $\mathrm{Aut}(\mathcal{V}, \mathcal{R})$ acting on $\mathcal{V} \times \mathcal{V}$, and a basis is given by the orbit-indicator operators $\sum_{(\mathbf{x},\mathbf{x}')\in\mathcal{O}} \mathbf{e}_{\mathbf{x}}\mathbf{e}_{\mathbf{x}'}{}^{\mathsf{T}}$, indexed by orbits $\mathcal{O}$.
4. Use $\mathcal{C}(\mathcal{U})$ as the token selection set in place of $\mathcal{T}$ and form the basis weights $\mathcal{W}_\ell$ as in section 3.3.

For binary $\mathcal{R}$ defining a directed graph, this recipe identifies the isomorphic connected components of the graph; the symmetric pair and the directed pair are the simplest such components. For higher-arity relations the orbit count grows accordingly, and the commutant captures progressively richer relational structure (such as equivalence classes, n-cycles, or transition kernels). In every case the same gradient-invariance argument applies and the corresponding basis weights remain invariant under population gradient flow.

C The Geometry of Reasoning Circuits

In this appendix, we discuss the geometry of the Invariant Manifold of Inductive Reasoning (IMIR) in more detail. We first give an intuitive description of its coordinate frame, and then we give a theoretical motivation for its origin based on representational symmetries.

C.1 What is an Attention Head?

Now that our key theoretical result has been established, we give an intuitive description of its geometry. Using a toy attention example, we introduce the mechanisms of the circuits that exist in the Invariant Manifold of Inductive Reasoning. Consider the input embeddings $\mathbf{h}_i^0 = \mathbf{e_a} + \mathbf{p}_i$ (where $\mathbf{h}_i^0 \in \mathbb{R}^d$) of token $\mathbf{a} \in \mathcal{V}$ at position $i \in [N]$. What can an attention head do with this input?

Token Actions. In our setup, we consider that the only correlations present in the data are between non-overlapping pairs of associated tokens in $\mathcal{R}$, as formalized by the $\mathcal{R}$-invariance assumption of appendix A.1. Thus, for each token, the model can learn only two possible mappings: to the same token or to the associated token. Applying this to attention heads, we get two possible token actions. First, an attention head may look for the exact token it has, constructing key and query vectors that align when the input tokens match, i.e. $\mathbf{W}_K^{(1,1)}\,\mathbf{e_a} \approx \mathbf{W}_Q^{(1,1)}\,\mathbf{e_b}$ when $\mathbf{a} = \mathbf{b}$. This is achieved when the key-query circuit is approximately a multiple of the identity transformation over the subspace of token embeddings, i.e. $\mathbf{W}_K^{(1,1)\intercal}\mathbf{W}_Q^{(1,1)} \approx \alpha\mathbf{I}^{(t)}$, where $\mathbf{I}^{(t)} = \sum_{\mathbf{x}\in\mathcal{V}} \mathbf{e_x}\mathbf{e_x}^\intercal$. The second possible action for the attention head is to look for the associated token, i.e. $\mathbf{W}_Q^{(1,1)}\,\mathbf{e_a} \approx \mathbf{W}_K^{(1,1)}\,\mathbf{e_b}$ when $\{\mathbf{a},\mathbf{b}\} \in \mathcal{R}$. In this case, the key-query circuit must act as a mapping between associated tokens, i.e. $\mathbf{W}_K^{(1,1)\intercal}\mathbf{W}_Q^{(1,1)} \approx \alpha\mathbf{C}$, where $\mathbf{C} = \sum_{\{\mathbf{a},\mathbf{b}\}\in\mathcal{R}}(\mathbf{e_a}\mathbf{e_b}^\intercal + \mathbf{e_b}\mathbf{e_a}^\intercal)$.

Position Actions. In our setup, each block has a random positional offset, simulating an arbitrary (unknown) number of imaginary tokens separating consecutive blocks. This breaks positional correlations between separate blocks, which means that positional embeddings may only be used to attend to a previous position in the same block (up to $k-1$ positions back). Indeed, since block offsets are sampled independently, the relative position between tokens of different blocks is uniformly random and carries no usable information. This is achieved by constructing key and query vectors that align for a certain position distance, i.e. $\mathbf{W}_K^{(1,1)}\,\mathbf{p}_i \approx \mathbf{W}_Q^{(1,1)}\,\mathbf{p}_{i-j}$ for a distance $j \in [k-1]$. Using translation-invariant sinusoidal embeddings, this is achieved when the key-query circuit encodes a specific rotation $\mathbf{W}_K^{(1,1)\intercal}\mathbf{W}_Q^{(1,1)} \approx \alpha\mathbf{M}^j$, where $\mathbf{M} = \mathbb{E}[\mathbf{p}_{i-1}\mathbf{p}_i{}^\intercal]$ and $\mathbf{M}^j$ denotes the j-th matrix power of $\mathbf{M}$. The power arises because $\mathbf{M}$ rotates every frequency plane by its unit-position angle, mapping each position embedding to that of the preceding position ($\mathbf{M}\,\mathbf{p}_i = \mathbf{p}_{i-1}$); applying it j times thus shifts positions back by j, i.e. $\mathbf{M}^j\,\mathbf{p}_i = \mathbf{p}_{i-j}$. Deep circuits might also benefit from attending to the exact same position using $\mathbf{W}_K^{(1,1)\intercal}\mathbf{W}_Q^{(1,1)} \approx \alpha\mathbf{I}^{(p)}$, where $\mathbf{I}^{(p)} = \mathbb{E}[\mathbf{p}_i\mathbf{p}_i{}^\intercal]$.

To see how this extends to stacked attention heads, assume that the first attention head has attended entirely to a token $\mathbf{b} \in \mathcal{V}$ at position $j \in [N]$. The residual stream at position i has become $\mathbf{h}_i^1 = \mathbf{e_a} + \mathbf{p}_i + \mathbf{W}_P^{(1,1)}\mathbf{W}_V^{(1,1)}(\mathbf{e_b} + \mathbf{p}_j)$. How could a second attention head leverage these representations?

Output-Value Projections. While the previously described actions remain perfectly valid, a whole new range of possibilities is unlocked. A second head can now attend using the information retrieved by the first head. For example, the second head can attend to positions containing the token retrieved by the first head if $\mathbf{W}_K^{(2,1)}(\mathbf{W}_P^{(1,1)}\mathbf{W}_V^{(1,1)}\mathbf{e_b}) \approx \mathbf{W}_Q^{(2,1)}\mathbf{e_b}$ or $\mathbf{W}_K^{(2,1)\intercal}\mathbf{W}_Q^{(2,1)} \approx \mathbf{W}_P^{(1,1)}\mathbf{W}_V^{(1,1)}\mathbf{I}^{(t)}$. The information retrieved by the first head can also be used by the second head when constructing queries. There are twice as many ways to construct keys and queries, so the number of total possible behaviors has increased four-fold.

Combined Projections. To get an intuition for even deeper circuits, assume that the second attention head at position i has attended entirely to a token $\mathbf{a}' \in \mathcal{V}$ at position $i' \in [N]$. Moreover, assume that the first head at position i' has attended entirely to a token $\mathbf{b}' \in \mathcal{V}$ at position $j' \in [N]$, so

that $\mathbf{h}^1_{i'} = \mathbf{e}_{\mathbf{a}'} + \mathbf{p}_{i'} + \mathbf{W}_P^{(1,1)}\mathbf{W}_V^{(1,1)}(\mathbf{e}_{\mathbf{b}'} + \mathbf{p}_{j'})$. Therefore, after the second head, the residual stream at position i will be $\mathbf{h}^2_i = \mathbf{e}_{\mathbf{a}} + \mathbf{p}_i + \mathbf{W}_P^{(1,1)}\mathbf{W}_V^{(1,1)}(\mathbf{e}_{\mathbf{b}} + \mathbf{p}_j) + \mathbf{W}_P^{(2,1)}\mathbf{W}_V^{(2,1)}(\mathbf{e}_{\mathbf{a}'} + \mathbf{p}_{i'}) + \mathbf{W}_P^{(2,1)}\mathbf{W}_V^{(2,1)}\mathbf{W}_P^{(1,1)}\mathbf{W}_V^{(1,1)}(\mathbf{e}_{\mathbf{b}'} + \mathbf{p}_{j'})$. Note that the residual stream is now a sum of eight embedding terms — the four token embeddings $\mathbf{e}_{\mathbf{a}}, \mathbf{e}_{\mathbf{b}}, \mathbf{e}_{\mathbf{a}'}, \mathbf{e}_{\mathbf{b}'}$ and the four position embeddings $\mathbf{p}_i, \mathbf{p}_j, \mathbf{p}_{i'}, \mathbf{p}_{j'}$ — each rotated into its own subspace by the output–value projections of the heads that wrote it. Each of them could be used by a third head to construct queries and keys by accessing the corresponding subspaces.[12]

C.2 Representational Symmetry of Inductive Tasks

Token Embeddings. We use a data representation with fixed, non-learnable embeddings, mapping every $\mathbf{x} \in \mathcal{V}$ to a vector $\mathbf{e}_{\mathbf{x}} \in \mathbb{R}^d$, resulting in the embedding matrix $\mathbf{E} \in \mathbb{R}^{m \times d}$. However, in the context of induction head learning, the model should not memorize specific embeddings in its parameters. Rather, it should learn to associate tokens with certain roles, as is the case in the class of block-list tasks described above. Hence, we want the model weights to be invariant under transformation symmetries S of the data representation $\mathbf{E}$: What is learned with $\mathbf{E}$ should generalize to all input representations $\mathbf{E}' = \mathbf{U}\mathbf{E}$, with $\mathbf{U} \in S$.

Specifically, natural transformation symmetries to consider are subgroups of orthogonal transformations $S \subseteq O(d)$. As is well established, such orthogonal invariances can be tied to the commutant of the symmetry group

$$\mathcal{C}(S) := \{\mathbf{W} : \mathbf{W}\mathbf{U} = \mathbf{U}\mathbf{W},\ \forall\, \mathbf{U} \in S\}\,. \tag{69}$$

It can be shown easily that, for instance, equivariance of linear operators and invariance of bi-linear operators constrains them to be in $\mathcal{C}(S)$. The larger S, the more constrained are operators, for instance, $\mathcal{C}(O(d)) = \mathrm{span}(\mathbf{I})$ and invariant operators can only re-scale data.

What is then a suitable choice for S? Our focus is on retaining associative pairwise relations between vocabulary items. Hence, we assume that the task implies a symmetric, irreflexive binary relation $\sim$ on $\mathcal{V}$ such that each element is related to *exactly* one other element (a perfect matching). This induces a subgroup $S \subseteq O(d)$ consisting of permutation matrices $\mathbf{U}_\pi$ corresponding to permutations π of $\mathcal{V}$ that preserve the relation, i.e. $\mathbf{x} \sim \mathbf{y} \iff \pi(\mathbf{x}) \sim \pi(\mathbf{y})$. The commutation relation implies that operators in $\mathcal{C}(S)$ are constant on orbits of index pairs under S. There are three such orbits

$$\mathcal{O}_1 = \{(\mathbf{x}, \mathbf{x}) : \mathbf{x} \in \mathcal{V}\}, \quad \mathcal{O}_2 = \{(\mathbf{x}, \mathbf{y}) : \mathbf{x} \sim \mathbf{y},\ \mathbf{x}, \mathbf{y} \in \mathcal{V}\}, \quad \mathcal{O}_3 = \overline{\mathcal{O}}_1 \cap \overline{\mathcal{O}}_2\,. \tag{70}$$

and the commutant is given by

$$\mathcal{C}(S) = \mathrm{span}\{\mathbf{I}, \mathbf{C}, \mathbf{1}\mathbf{1}^\top\}, \quad \mathbf{C} := \mathbf{1}\{x \sim y\}\,. \tag{71}$$

Note that by centering embeddings such that $\mathbf{1}^\top\mathbf{E} = \mathbf{0}$ one can eliminate $\mathbf{1}\mathbf{1}^\top$. The interpretation is very clear: For each token, a (bi-)linear map can learn only one of two possible things (and superpositions thereof): auto-association of a token with itself (i.e. $\mathbf{I}$) or hetero-association of a token with its partner (i.e. $\mathbf{C}$).

Positional Embeddings. Token positions are encoded via sinusoidal embeddings as in [29]. To model the fact that relevant blocks may appear at arbitrary positions, we add random offsets for each block. Let $\omega_i \in \mathbb{R}_+$, $i \in [k]$, denote the embedding frequencies. For a token at relative position t inside block b, with offset ϕ_b, we define

$$p_{b,t} = \big(\sin(\omega_1(t + \phi_b)), \cos(\omega_1(t + \phi_b)), \ldots, \sin(\omega_k(t + \phi_b)), \cos(\omega_k(t + \phi_b))\big)^\top \in \mathbb{R}^{2k}.$$

In the spirit of the previous paragraph, we require that proper induction does not learn or memorize absolute token positions, but only relative ones. We thus require invariance under relative position shifts. It is a key property of sinusoidal embeddings that shifts in position correspond to rotations within each subspace V_j. More abstractly, the above positional embeddings admit a decomposition into orthogonal two-dimensional subspaces corresponding to different frequencies, i.e. $\mathbb{R}^{2k} = \bigoplus_j V_j$. The corresponding subgroup $P \subseteq O(d)$ consists of block-diagonal orthogonal matrices of the form

$$\mathbf{U} = \mathrm{diag}(\mathbf{U}_1, \ldots, \mathbf{U}_k), \quad \mathbf{U}_j \in SO(2). \tag{72}$$

[12]The number of combined projections across layers is analogous to the number of possible code calls when converting a transformer into D-RASP, a programming language proposed to abstract transformers [49].

As before, symmetry of the data distribution implies that admissible operators must commute with all $\mathbf{U} \in P$, which already constrains them substantially. Commutation alone, however, does not fully determine the structure used in the main text: additional properties of the sinusoidal embeddings, unrelated to this symmetry, further reduce the admissible operators to the desired polynomial structure in the positional shift operator (see appendix A.3).

D In-Context Learning vs. In-Weights Learning

D.1 Training details

We train a 2-layer, 1-head-per-layer attention-only transformer of width 128 on an in-context bigram prediction task with a vocabulary of 16 common tokens and 64 rare tokens, plus sinusoidal positional embeddings with 4 frequencies (8 features). Each sequence consists of 4 bigram demonstrations followed by a query, giving a context length of 9. Tokens within each sequence are sampled as rare with probability 0.9, with single-shot demonstrations within context and zero per-sequence frequency variation. To facilitate interpretability, we train the simplified architecture described in section A.3.2, with merged key-query matrix, fixed output-value projections, and orthogonal representations. The width of 128 is chosen so that the model exactly admits the geometric construction of the bigram circuit on the manifold (base dimension 32, branching $(1 + H)^L = 4$, hence $4 \cdot 32 = 128$). We optimize with vanilla SGD at learning rate 0.1 and batch size 256 for 250 steps; the random seed is fixed at 0. Every 10 steps we estimate top-1 accuracy on two probe distributions, each averaged over 8 fresh batches: an *ICL probe* in which every token is rare, so that the query bigram can only be solved by attending to its in-context demonstration, and an *IWL probe* in which every token is common and the context is shuffled so the target pair is absent, forcing reliance on the in-weights bigram statistics. For the constrained training, we use the same optimization hyperparameters, while ablating all model parameters except for α, β, γ, and δ after every step of gradient descent.

Compute resources. The model has approximately 0.05 M learnable parameters; one full training run (250 SGD steps at batch size 256, plus the periodic ICL and IWL probes) takes well under a minute on a single consumer-grade GPU and uses less than 100 MB of GPU memory. Constrained training adds a single per-step IMIR projection, which is dominated by a constant-size matrix multiply and does not change the wall-clock cost in any noticeable way.

D.2 Circuit Ablations

In this appendix, we causally test the circuit interpretation given in section 4.2. We take the model trained as described in appendix D.1 and ablate all of its parameters except for a chosen subset of the four highlighted IMIR directions; we then re-evaluate the training loss and the ICL and IWL probe accuracies (defined in appendix D.1). Each row of table 1 corresponds to one such subset. The results confirm our interpretation: keeping only (α, β, γ) preserves ICL accuracy (0.75, vs. 0.77 for the unablated model) while destroying IWL; keeping only δ yields near-perfect IWL accuracy and no ICL; and removing any single direction from (α, β, γ) destroys ICL entirely, confirming that these three directions jointly implement the induction head.

Table 1: Ablation study confirming that the $\alpha\beta\gamma$-circuit implements In-Context Learning (ICL), while the δ-circuit accounts for In-Weights Learning (IWL).

Ablated except	Train Loss	ICL Accuracy	IWL Accuracy
all (original)	1.41	**0.77**	0.22
$\alpha, \beta, \gamma, \delta$	**1.06**	**0.76**	0.14
α, β, γ	**1.03**	**0.75**	0.10
δ	4.34	0.01	**0.99**
α, β	4.38	0.00	0.06
α, γ	3.99	0.05	0.00
β, γ	7.47	0.03	0.00

D.3 All circuits during training

As described in section 4.2, the IMIR of our two-layer, single-head model is 28-dimensional. We additively decompose the model's key–query and output weights into these 28 basis directions, and we track the resulting coefficients throughout training. fig. 6 plots every coefficient, across the data regimes considered in section 4.2. Only the four highlighted directions (α, β, γ, and δ) grow appreciably during training; this is the evidence behind our claim in section 4.2 that learning concentrates on these four directions.

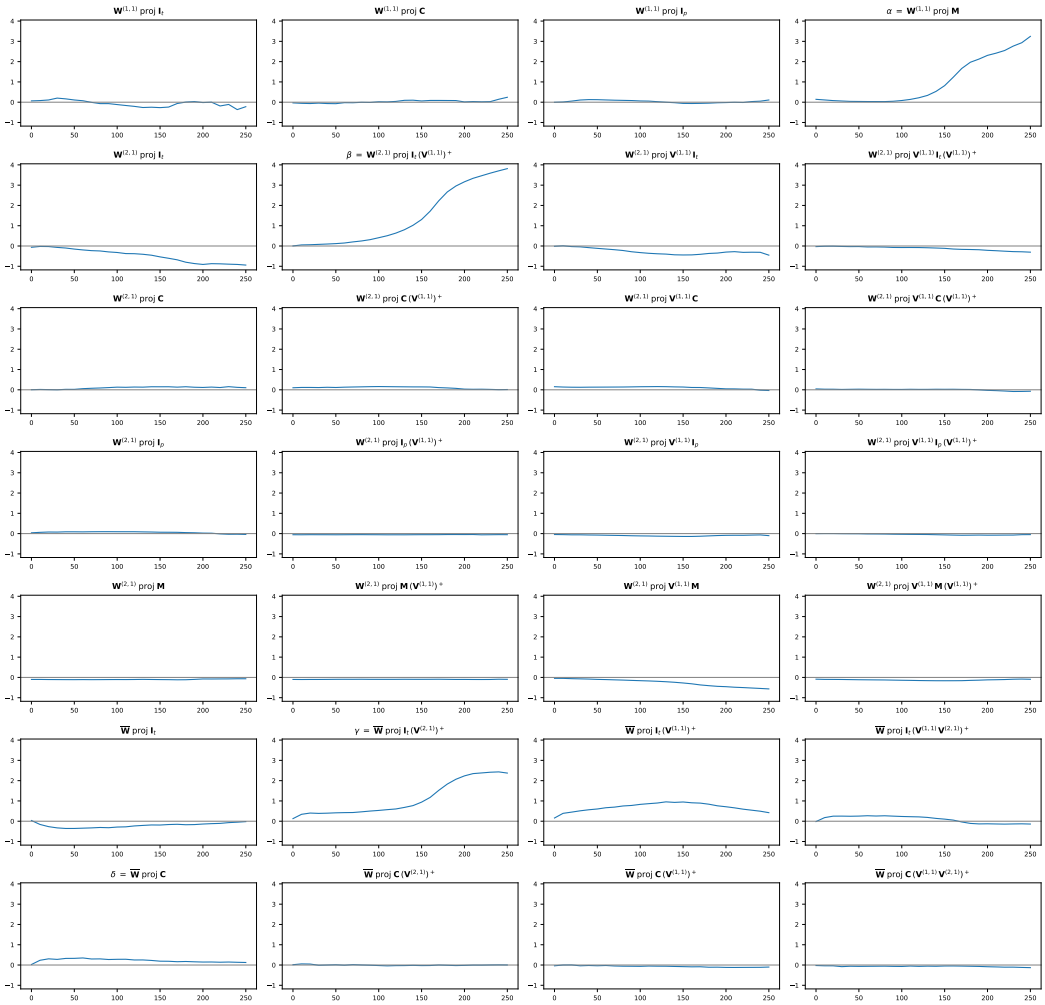

Figure 6: All 28 IMIR coefficients of a 2-layer, single-head Transformer trained on in-context bigrams, tracked over 250 training steps: the 4 first-layer directions (top row), the 16 second-layer directions (middle rows), and the 8 output directions (bottom rows). Only the four highlighted directions (α, β, γ, δ) grow appreciably during training.

D.4 Proof of Theorem 2

Theorem 2. *In the presence of an IWL circuit with logit scale δ, the ICL circuit receives a population gradient suppressed by a data-dependent factor:*

$$\mathbb{E}\big[\nabla_{\mathrm{ICL}}\mathcal{L}(\mathbf{W}+\mathbf{W}_{\mathrm{IWL}})\big] \;=\; \Big(f_r+(1-f_r)f_v\Big)\,\mathbb{E}\big[\nabla_{\mathrm{ICL}}\mathcal{L}(\mathbf{W})\big] \;+\; O\big(|\mathcal{V}|\,e^{-\delta}+|\mathcal{V}|^{-1}\big), \quad (13)$$

where $\mathbf{W}$ denotes the ICL-only configuration (no in-weights learning). The error term vanishes in the saturated-IWL ($\delta\to\infty$) and large-vocabulary ($|\mathcal{V}|\to\infty$) limit.

We give the detailed proof of Theorem 2, following the proof sketch: we partition the data distribution into the part on which the IWL solution succeeds and the remainder, and we show that the IWL solution kills the ICL gradient on the former while leaving it unchanged on the latter.

We work in the simplified architecture of section A.3.2 with the in-context bigram task ($k=2$); the argument extends to arbitrary k without modification. The ICL induction circuit is specified by three pseudo-parameters α,β,γ via

$$\mathbf{W}^{(1)} \;=\; \alpha\,\mathbf{M}, \qquad\qquad \mathbf{W}^{(2)} \;=\; \beta\,\mathbf{I}^{(t)}\,\mathbf{V}^{(1)\intercal}, \qquad\qquad \bar{\mathbf{W}} \;=\; \gamma\,\mathbf{I}^{(t)}\,\mathbf{V}^{(2)\intercal}, \quad (73)$$

and the IWL solution by a single pseudo-parameter δ via $\bar{\mathbf{W}}=\delta\,\mathbf{C}$. We write $\mathbf{W}=(\alpha,\beta,\gamma,0)$ for an ICL-only configuration and $\mathbf{W}+\mathbf{W}_{\mathrm{IWL}}=(\alpha,\beta,\gamma,\delta)$ for the configuration with the IWL solution superimposed on top, so that $\bar{\mathbf{W}}=\gamma\,\mathbf{I}^{(t)}\,\mathbf{V}^{(2)\intercal}+\delta\,\mathbf{C}$.

Setup: the ICL gradient, the regime, and the disjointness assumption. We pin down the three objects the comment flags as under-specified.

(A1) The ICL gradient is a scalar. Throughout we use the single definition of appendix D.5: $\nabla_{\mathrm{ICL}}\mathcal{L}$ is the directional derivative of the loss along the unit ICL tuple direction $\hat{e}_{\mathrm{ICL}}\propto(\mathbf{W}^{(1)}_{\mathrm{ICL}},\mathbf{W}^{(2)}_{\mathrm{ICL}},\bar{\mathbf{W}}_{\mathrm{ICL}})$ under the Frobenius inner product,

$$\begin{aligned}\nabla_{\mathrm{ICL}}\mathcal{L} \;&=\; \big\langle \mathbf{W}^{(1)}_{\mathrm{ICL}},\frac{\partial\mathcal{L}}{\partial\mathbf{W}^{(1)}}\big\rangle+\big\langle \mathbf{W}^{(2)}_{\mathrm{ICL}},\frac{\partial\mathcal{L}}{\partial\mathbf{W}^{(2)}}\big\rangle+\big\langle \bar{\mathbf{W}}_{\mathrm{ICL}},\frac{\partial\mathcal{L}}{\partial\bar{\mathbf{W}}}\big\rangle \;=\; \big(\mathbf{p}(\mathbf{z})-\mathbf{y}\big)^{\top}J,\\ J\;&:=\;\frac{\mathrm{d}\mathbf{z}_{\mathrm{ICL}}}{\mathrm{d}t}\Big|_{t=0}\in\mathbb{R}^{|\mathcal{V}|},\end{aligned} \quad (74)$$

where J is the logit-space directional Jacobian along $\hat{e}_{\mathrm{ICL}}$ and $J_x:=\langle\mathbf{e}_x,J\rangle$ is its coordinate in the (orthonormal) embedding basis; the chain rule gives the second equality because the loss sees the ICL direction only through the logits $\mathbf{z}$. This collapses the three pseudo-parameter partials $\partial\mathbf{z}_{\mathrm{ICL}}/\partial\theta$ of the sketch into one logit-space vector J, reconciling the notation with appendix D.5.

(A2) Geometry. Token embeddings are orthonormal (the associative-memory idealization; lex-invariant rare embeddings share the common tokens' norm and are pairwise orthogonal up to $O(|\mathcal{V}|^{-1})$ fluctuations), under the standing assumptions of section A.3.2, and all gradients are population (data-averaged) gradients, consistent with Corollary 1.

(A3) Asymptotic regime. We work in the *saturated-IWL, large-vocabulary* regime: the sequence length N, the number of blocks n, the common-vocabulary size $|\mathcal{V}^c|$, and the ICL readout scale $\gamma=O(1)$ are all held *fixed*, while $|\mathcal{V}|\to\infty$ (the rare vocabulary grows) and the IWL scale $\delta\to\infty$ jointly, fast enough that $|\mathcal{V}|\,e^{-\delta}\to 0$ (equivalently $\delta-\log|\mathcal{V}|\to\infty$). Because $\gamma=O(1)$, the ICL logits are $O(1)$ and the softmax partition function is $\Theta(|\mathcal{V}|)$, so every off-target probability is $O(|\mathcal{V}|^{-1})$ — the fact that drives both error terms.

(A4) Distractor disjointness. For a common query with canonical partner $\mathbf{x}'_q$, no distractor block carries $\mathbf{x}'_q$ as an item or a label; i.e. the query's canonical association is token-disjoint from the distractor context. Under (A4) the induction circuit never reads $\mathbf{x}'_q$, so its logit-space contribution — and hence $J_{\mathbf{x}'_q}$ — is pure orthonormality fluctuation, $J_{\mathbf{x}'_q}=O(|\mathcal{V}|^{-1})$. This is the clean fix for the cross-talk flagged above; Remark 1 quantifies the deviation of the actual finite-vocabulary sampler from (A4).

We summarize the residual error of the IWL solution by

$$\epsilon_{\mathrm{IWL}} \;:=\; \mathbb{E}_{\mathcal{D}_{\mathrm{c,can}}}\big[\,1-\mathbf{p}(\mathbf{z})_{\mathbf{x}'_q}\,\big] \;=\; O\big(|\mathcal{V}|\,e^{-\delta}\big), \quad (75)$$

the probability mass that the saturated model fails to place on the canonical token on the part of the data the IWL solution covers; under (A3), $\epsilon_{\mathrm{IWL}}\to 0$, and the rate is derived in Lemma 2.

Output decomposition. Let q denote the query position. The simplified architecture yields the orthogonal decomposition

$$\mathbf{H}^{(2)}_{:,N} \;=\; \underbrace{\mathbf{e}_{\mathbf{x}_q}}_{\in\,\mathbb{U}^{(\mathrm{t})}} \;+\; \underbrace{\mathbf{p}_q}_{\in\,\mathbb{U}^{(\mathrm{p})}} \;+\; \underbrace{\mathbf{V}^{(1)}\mathbf{u}^{(1)}}_{\in\,\mathbf{V}^{(1)}\mathbb{U}^{(0)}} \;+\; \underbrace{\mathbf{V}^{(2)}\mathbf{u}^{(2)}}_{\in\,\mathbf{V}^{(2)}\mathbb{U}^{(1)}}, \tag{76}$$

with the four addends in pairwise orthogonal subspaces (cf. section A.3.2). The map $\mathbf{C}$ acts only on the common-token subspace and vanishes on the other three subspaces, so

$$\mathbf{C}\,\mathbf{H}^{(2)}_{:,N} \;=\; \mathbf{C}\,\mathbf{e}_{\mathbf{x}_q} \;=\; \begin{cases} \mathbf{e}_{\mathbf{x}'_q} & \text{if } \mathbf{x}_q \in \mathcal{V}^c \text{ with canonical partner } \mathbf{x}'_q, \\ \mathbf{0} & \text{if } \mathbf{x}_q \in \mathcal{V}^r. \end{cases} \tag{77}$$

The model logits at q therefore split into an ICL part and an IWL part,

$$\mathbf{z} \;=\; \bar{\mathbf{W}}\,\mathbf{H}^{(2)}_{:,N} \;=\; \mathbf{z}_{\mathrm{ICL}}(\alpha,\beta,\gamma) \;+\; \delta\,\mathbf{C}\,\mathbf{e}_{\mathbf{x}_q}, \qquad \mathbf{z}_{\mathrm{ICL}} \;=\; \gamma\,\mathbf{I}^{(t)}\,\mathbf{V}^{(2)\intercal}\,\mathbf{H}^{(2)}_{:,N}, \tag{78}$$

where the IWL part depends on δ but not on α,β,γ.

Lemma 1 (The ICL Jacobian is blind to the IWL solution)**.** *The directional Jacobian J of* (74) *is independent of the IWL scale δ, so the only δ-dependence of $\nabla_{\mathrm{ICL}}\mathcal{L}$ is funneled through the probabilities:*

$$\nabla_{\mathrm{ICL}}\mathcal{L} \;=\; \big(\mathbf{p}(\mathbf{z}) - \mathbf{y}\big)^{\top} J, \qquad \mathbf{p}(\mathbf{z})_x \;=\; \frac{e^{\mathbf{e}_x{}^{\intercal}\mathbf{z}}}{\sum_{x'} e^{\mathbf{e}_{x'}{}^{\intercal}\mathbf{z}}}. \tag{79}$$

Moreover J obeys two structural bounds: (J1) *an ℓ_1 bound $\|J\|_1 = \sum_x |J_x| = O(1)$; and* (J2) *under* (A4)*, $|J_{\mathbf{x}'_q}| = O(|\mathcal{V}|^{-1})$.*

Proof. For each $\theta \in \{\alpha,\beta,\gamma\}$, $\partial\mathbf{H}^{(2)}_{:,N}/\partial\theta$ lives entirely in $\mathbf{V}^{(1)}\mathbb{U}^{(0)} + \mathbf{V}^{(2)}\mathbb{U}^{(0)} + \mathbf{V}^{(2)}\mathbb{U}^{(1)}$ (it is built from differentiating the layer-1/layer-2 attention scores and attended content, all produced by the value rotations of those layers). All three subspaces are orthogonal to $\mathbb{U}^{(\mathrm{t})}$, so $\mathbf{C}$ annihilates them, $\mathbf{C}\,\partial\mathbf{H}^{(2)}_{:,N}/\partial\theta = \mathbf{0}$, and hence $\partial\mathbf{z}/\partial\theta = \partial\mathbf{z}_{\mathrm{ICL}}/\partial\theta$ is independent of δ; assembling the three partials into the directional derivative (74) makes J δ-independent. *(J1):* $J = \mathrm{d}\mathbf{z}_{\mathrm{ICL}}/\mathrm{d}t$ is, by the induction-circuit construction, a combination of in-context token embeddings weighted by layer-2 attention weights (which sum to 1) and the bounded value rotations, scaled by $\gamma = O(1)$; in the orthonormal embedding basis its coordinates therefore have ℓ_1 mass $O(1)$. *(J2):* the induction circuit reads only in-context tokens; under (A4) the canonical partner $\mathbf{x}'_q$ appears in no block, so $J_{\mathbf{x}'_q}$ collects only the $O(|\mathcal{V}|^{-1})$ orthonormality fluctuations. □

Splitting the data into "IWL-solvable" and "IWL-unhelpful". The data distribution $\mathcal{D}(f_r, f_v, b)$ partitions into three sub-distributions, $\mathcal{D}_r$ (rare query, frequency f_r), $\mathcal{D}_{c,\mathrm{can}}$ (common query whose in-context partner equals the canonical partner $\mathbf{x}'_q$, frequency $(1-f_r)(1-f_v)$), and $\mathcal{D}_{c,\mathrm{scr}}$ (common query whose in-context partner differs from $\mathbf{x}'_q$, frequency $(1-f_r)f_v$). We bundle the latter two according to whether the IWL solution succeeds at solving them.

Lemma 2 (IWL-solvable part)**.** *On $\mathcal{D}_{c,\mathrm{can}}$ (mass $(1-f_r)(1-f_v)$),*

$$\nabla_{\mathrm{ICL}}\mathcal{L}_{c,\mathrm{can}}\big(\mathbf{W} + \mathbf{W}_{\mathrm{IWL}}\big) \;=\; O(\epsilon_{\mathrm{IWL}}) \;=\; O\big(|\mathcal{V}|\,e^{-\delta}\big) \;\xrightarrow[\delta\to\infty]{}\; 0. \tag{80}$$

Proof. Here the target equals the canonical partner, $\mathbf{y} = \mathbf{e}_{\mathbf{x}'_q}$, and both the IWL term $\delta\,\mathbf{e}_{\mathbf{x}'_q}$ and the ICL readout $\gamma\,\mathbf{e}_{\mathbf{x}'_q}$ add to the *correct* logit, while every competing logit stays $O(1)$ by orthonormality. Hence the correct-token probability is

$$\mathbf{p}(\mathbf{z})_{\mathbf{x}'_q} \;=\; \frac{e^{\gamma+\delta}}{e^{\gamma+\delta} + \sum_{x\neq\mathbf{x}'_q} e^{\mathbf{e}_x{}^{\intercal}\mathbf{z}}} \;=\; 1 - O\big(|\mathcal{V}|\,e^{-(\gamma+\delta)}\big) \;=\; 1 - \epsilon_{\mathrm{IWL}}, \tag{81}$$

the second equality because the denominator's $|\mathcal{V}|-1$ off-target logits are each $O(1)$ by orthonormality. So the residual is $\|\mathbf{p}(\mathbf{z}) - \mathbf{y}\| = O(\epsilon_{\mathrm{IWL}})$. By Lemma 1, $J = O(1)$ is independent of δ (J1), so (79) gives a *suppressed* (not exactly cancelled) ICL gradient of order $\epsilon_{\mathrm{IWL}} = O(|\mathcal{V}|\,e^{-\delta})$, decaying exponentially in the IWL margin δ. □

Lemma 3 (IWL-unhelpful part)**.** *On $\mathcal{D}_r \cup \mathcal{D}_{c,\mathrm{scr}}$ (mass $f_r + (1-f_r)f_v$), under* (A4)*,*

$$\nabla_{\mathrm{ICL}}\mathcal{L}_r\big(\mathbf{W}+\mathbf{W}_{\mathrm{IWL}}\big) = \nabla_{\mathrm{ICL}}\mathcal{L}_r\big(\mathbf{W}\big), \quad \nabla_{\mathrm{ICL}}\mathcal{L}_{c,\mathrm{scr}}\big(\mathbf{W}+\mathbf{W}_{\mathrm{IWL}}\big) = \nabla_{\mathrm{ICL}}\mathcal{L}_{c,\mathrm{scr}}\big(\mathbf{W}\big)+O\big(|\mathcal{V}|^{-1}\big). \tag{82}$$

Proof. Rare queries. On $\mathcal{D}_r$ the IWL term is identically zero ($\mathbf{C}\,\mathbf{e}_{\mathbf{x}_q} = \mathbf{0}$), so $\mathbf{z}$ and hence $\mathbf{p}(\mathbf{z})$ are unchanged by adding $\mathbf{W}_{\mathrm{IWL}}$; with J also δ-independent (Lemma 1) the gradient is *exactly* unchanged.

Scrambled common queries. The target is the in-context partner $\mathbf{x}^* \neq \mathbf{x}'_q$, and the IWL contribution $\delta\,\mathbf{e}_{\mathbf{x}'_q}$ adds logit mass to a *wrong* token. By Lemma 1 only $\mathbf{p}(\mathbf{z})$ carries the δ-dependence, so with $\Delta\mathbf{p} := \mathbf{p}(\mathbf{z} + \delta\,\mathbf{e}_{\mathbf{x}'_q}) - \mathbf{p}(\mathbf{z})$,

$$\nabla_{\mathrm{ICL}}\mathcal{L}_{c,\mathrm{scr}}\big(\mathbf{W} + \mathbf{W}_{\mathrm{IWL}}\big) - \nabla_{\mathrm{ICL}}\mathcal{L}_{c,\mathrm{scr}}\big(\mathbf{W}\big) \;=\; \Delta\mathbf{p}^\top J \;=\; \sum_x \Delta p_x\, J_x. \tag{83}$$

Adding $\mathbf{W}_{\mathrm{IWL}}$ only rescales the softmax normalizer, so for every $x \neq \mathbf{x}'_q$,

$$\mathbf{p}(\mathbf{z} + \delta\,\mathbf{e}_{\mathbf{x}'_q})_x \;=\; \frac{\mathbf{p}(\mathbf{z})_x}{1 + (e^\delta - 1)\,\mathbf{p}(\mathbf{z})_{\mathbf{x}'_q}} \;\leq\; \mathbf{p}(\mathbf{z})_x \qquad (x \neq \mathbf{x}'_q). \tag{84}$$

Hence $\max_{x\neq\mathbf{x}'_q} |\Delta p_x| \leq \max_{x\neq\mathbf{x}'_q} \mathbf{p}(\mathbf{z})_x = O(|\mathcal{V}|^{-1})$ by (A3): under orthonormal embeddings and $O(1)$ logits the partition function is $\Theta(|\mathcal{V}|)$, so each off-target mass is $O(|\mathcal{V}|^{-1})$. The single coordinate $x = \mathbf{x}'_q$ instead absorbs $\Theta(1)$ of mass, $\Delta p_{\mathbf{x}'_q} = \Theta(1)$. We therefore split it off and bound the remainder by Hölder against the Jacobian ℓ_1 norm:

$$\Big|\sum_x \Delta p_x\, J_x\Big| \;\leq\; \underbrace{\big|\Delta p_{\mathbf{x}'_q}\big|\,\big|J_{\mathbf{x}'_q}\big|}_{\Theta(1)\,\cdot\,O(|\mathcal{V}|^{-1})\text{ by (J2)}} \;+\; \underbrace{\Big(\max_{x\neq\mathbf{x}'_q}|\Delta p_x|\Big)\sum_{x\neq\mathbf{x}'_q}|J_x|}_{O(|\mathcal{V}|^{-1})\,\cdot\,O(1)\text{ by (J1)}} \;=\; O\big(|\mathcal{V}|^{-1}\big). \tag{85}$$

This is where *(A4)* is load-bearing: it gives $|J_{\mathbf{x}'_q}| = O(|\mathcal{V}|^{-1})$ (J2), without which the first term would be $\Theta(1)$ on the event that $\mathbf{x}'_q$ appears as a distractor token (Remark 1). Note that the per-coordinate bound $|\Delta p_x| = O(|\mathcal{V}|^{-1})$ does *not* suffice on its own — summed over $|\mathcal{V}|$ coordinates it gives $O(1)$ — which is why the bound is carried as $\ell_\infty(\Delta\mathbf{p}) \cdot \ell_1(J)$. □

Lemma 4 (IWL-free gradients agree across sub-distributions)**.** *Under* (A4)*, the IWL-free per-sub-distribution ICL gradients agree up to $O(|\mathcal{V}|^{-1})$:*

$$\nabla_{\mathrm{ICL}}\mathcal{L}_r(\mathbf{W}) \;=\; \nabla_{\mathrm{ICL}}\mathcal{L}_{c,\mathrm{can}}(\mathbf{W}) \;=\; \nabla_{\mathrm{ICL}}\mathcal{L}_{c,\mathrm{scr}}(\mathbf{W}) \;=\; \nabla_{\mathrm{ICL}}\mathcal{L}(\mathbf{W} \mid f_r = 1) \;+\; O\big(|\mathcal{V}|^{-1}\big). \tag{86}$$

Proof. At $\mathbf{W}$ (no IWL) the logits are $\mathbf{z}_{\mathrm{ICL}}$ alone, which by construction of the induction circuit equals (a multiple of) the in-context partner embedding $\mathbf{e}_{\mathbf{x}^*}$ in all three sub-distributions — the canonical association is never read, only the in-context demonstration is, and (A4) ensures the canonical partner does not re-enter as a distractor token. Together with the lex-invariance of rare embeddings and the orthonormality of common-token embeddings, this gives a per-sequence loss depending on the target only through γ and $|\mathcal{V}|$, not on whether the target is rare or common. The three gradients therefore agree, exactly in the orthonormal idealization and up to $O(|\mathcal{V}|^{-1})$ finite-size corrections otherwise; by linearity of expectation the data-averaged $\nabla_{\mathrm{ICL}}\mathcal{L}(\mathbf{W})$ equals this common value up to the same order. □

Remark 1 (Finite-vocabulary cross-talk: deviation from (A4))**.** *The sampler of section 4.1 draws the $\Theta(n)$ distractor associations independently and does* not *enforce (A4): a distractor common block coincides with the query's canonical pair with probability $\Theta(1/|\mathcal{V}^c|)$ per block, so the canonical partner $\mathbf{x}'_q$ surfaces in the context with probability $\Theta(n\,f_c/|\mathcal{V}^c|)$ — governed by the* common-*vocabulary size $|\mathcal{V}^c|$ (e.g.* 16 *in our experiments), not the total $|\mathcal{V}|$. On that event the induction circuit* does *read $\mathbf{x}'_q$, so $|J_{\mathbf{x}'_q}| = \Theta(1)$ and the first term of* (85) *is $\Theta(1)$ rather than $O(|\mathcal{V}|^{-1})$. Carrying it explicitly replaces the $O(|\mathcal{V}|^{-1})$ error of Lemmas 3 and 4 (and of Theorem 2) by $O\big(|\mathcal{V}|^{-1}+n\,f_c/|\mathcal{V}^c|\big)$. This term does not vanish in the orthonormal idealization; (A4) removes it by fiat, and it is the only place where the finite common vocabulary enters the bound.*

Combining the two parts. Summing the per-sub-distribution contributions weighted by their frequencies and substituting Lemmas 2 and 3 ($O(\epsilon_{\mathrm{IWL}})$ on $\mathcal{D}_{c,\mathrm{can}}$, unchanged-up-to-$O(|\mathcal{V}|^{-1})$ on $\mathcal{D}_{c,\mathrm{scr}}$, exactly unchanged on $\mathcal{D}_r$),

$$\begin{aligned}\nabla_{\mathrm{ICL}}\mathcal{L}\big(\mathbf{W}+\mathbf{W}_{\mathrm{IWL}}\big) \;&=\; f_r\,\nabla_{\mathrm{ICL}}\mathcal{L}_r\big(\mathbf{W}\big) && \text{(87a)}\\ &\quad+\;(1-f_r)(1-f_v)\,O(\epsilon_{\mathrm{IWL}})\\ &\quad+\;(1-f_r)\,f_v\,\Big(\nabla_{\mathrm{ICL}}\mathcal{L}_{c,\mathrm{scr}}\big(\mathbf{W}\big)+O\big(|\mathcal{V}|^{-1}\big)\Big)\\ &=\;\big(f_r+(1-f_r)\,f_v\big)\,\nabla_{\mathrm{ICL}}\mathcal{L}(\mathbf{W})\;+\;O(\epsilon_{\mathrm{IWL}})\;+\;O\big(|\mathcal{V}|^{-1}\big), && \text{(87b)}\end{aligned}$$

where the final step used Lemma 4 (equality of the IWL-free per-sub-distribution gradients). Taking expectations gives the statement of Theorem 2: the ICL gradient is suppressed by the data-coverage factor $f_r+(1-f_r)f_v$, exactly in the saturated-IWL, large-vocabulary limit and up to the two stated error terms otherwise. Without the disjointness assumption (A4), the $O(|\mathcal{V}|^{-1})$ term is replaced by $O\big(|\mathcal{V}|^{-1}+n\,f_c/|\mathcal{V}^c|\big)$, per Remark 1. □

D.5 Proof of Theorem 3

Theorem 3. *In the distractor-dominated regime, where the number of context blocks grows with the burstiness held fixed ($n\to\infty$ with b fixed, so $n\gg b$), the ICL circuit receives a population gradient proportional to burstiness:*

$$\mathbb{E}_{\mathcal{D}(b)}\big[\nabla_{\mathrm{ICL}}\mathcal{L}(\mathbf{W})\big]\;=\;b\,\mathbb{E}_{\mathcal{D}(1)}\big[\nabla_{\mathrm{ICL}}\mathcal{L}(\mathbf{W})\big]\,\big(1\;+\;O\big(b/n\big)\big), \tag{14}$$

where $\mathbb{E}_{\mathcal{D}(b)}$ denotes expectation over the data distribution at burstiness b.

The proof rests on a single observation: the ICL gradient decomposes into a sum of *per-block* contributions in which every relevant block carries the same expected signal and every distractor block carries zero. With b identical relevant copies in context, the resulting sum is b times the corresponding $b=1$ quantity.

Setup. We work in the simplified two-attention-layer architecture of section A.3.2 (single head per layer, fixed orthogonal output–value matrices $\mathbf{V}^{(\ell)}$, merged key–query matrices $\mathbf{W}^{(\ell)}$). The all-rare regime $f_r=1$ is assumed; the $(f_r+(1-f_r)f_v)$ pre-factor of Theorem 2 is independent of b and multiplies the conclusion below without altering its derivation. The extension to deeper or multi-head models follows from section A.3.6 and appendix A.4.

A single training sequence has n blocks of size $k=2$ followed by the query item at the last position $2n+1=N$. Block $r\in[n]$ has its item at position $i_r=2r-1$ and its label at position $j_r=2r$. The block indices split into the *relevant* set $\mathcal{R}=\{r:\mathbf{x}_{i_r}=\mathbf{a}_q\}$ of size $|\mathcal{R}|=b$ and the *distractor* set $\mathcal{D}=[n]\setminus\mathcal{R}$, whose items and labels are drawn from the rare vocabulary independently of $(\mathbf{a}_q,\mathbf{b}_q)$. We invoke the never-repeat (lexinvariance) assumption of section 4.1: distractor tokens are drawn from the effectively infinite rare vocabulary, so no distractor item coincides with $\mathbf{a}_q$. (With a finite rare vocabulary $\mathcal{V}^r$ this would fail with probability $\sim 1/|\mathcal{V}^r|$ per block, contributing an extra $O(n/|\mathcal{V}^r|)$ term; lexinvariance removes it.)

Layer-2 attention at the query. Write $s_{ik}^{(\ell)}$ for the pre-softmax attention logit from position i to position k at layer ℓ, and $A_{ik}^{(\ell)}=e^{s_{ik}^{(\ell)}}/Z_i^{(\ell)}$ for the corresponding attention weight, with $Z_i^{(\ell)}=\sum_k e^{s_{ik}^{(\ell)}}$. The query at position N reads out through the layer-2 denominator

$$Z_q\;\coloneqq\;Z_N^{(2)}\;=\;\sum_k e^{s_{Nk}^{(2)}}\;=\;\underbrace{\textstyle\sum_{r\in\mathcal{R}}e^{s_{Nj_r}^{(2)}}}_{b\text{ relevant keys}}\;+\;\underbrace{\textstyle\sum_{r\in\mathcal{D}}e^{s_{Nj_r}^{(2)}}}_{\Theta(n)\text{ distractor keys}}, \tag{88}$$

which couples all blocks. At a fixed ICL configuration the relevant logits take a common value s_{rel} and the distractor logits average to $\bar{s}_{\mathrm{dis}}$, so $Z_q=b\,e^{s_{\mathrm{rel}}}+(n-b)\,e^{\bar{s}_{\mathrm{dis}}}=\Theta(n)$ when $n\to\infty$ with b fixed. Consequently each relevant attention weight $A_{Nj_r}^{(2)}=e^{s_{\mathrm{rel}}}/Z_q=\Theta(1/n)$, and, since every per-block signal below carries exactly one such factor, the gradient is of order b/n. This is precisely why the error must be controlled *relative* to the leading term.

The ICL direction parameterizes the canonical induction circuit:

$$\mathbf{W}_{\mathrm{ICL}}^{(1)} \propto \mathbf{M}, \qquad \mathbf{W}_{\mathrm{ICL}}^{(2)} \propto \mathbf{V}^{(1)\intercal}\,\mathbf{I}^{(t)}, \qquad \bar{\mathbf{W}}_{\mathrm{ICL}} \propto \mathbf{I}^{(t)}\,\mathbf{V}^{(2)\intercal}. \tag{89}$$

The ICL gradient is the directional derivative of the loss along this tuple under the Frobenius inner product $\langle \mathbf{A}, \mathbf{B}\rangle = \mathrm{tr}(\mathbf{A}^\intercal \mathbf{B})$:

$$\nabla_{\mathrm{ICL}}\mathcal{L} \;=\; \big\langle \mathbf{W}_{\mathrm{ICL}}^{(1)}, \frac{\partial \mathcal{L}}{\partial \mathbf{W}^{(1)}} \big\rangle \;+\; \big\langle \mathbf{W}_{\mathrm{ICL}}^{(2)}, \frac{\partial \mathcal{L}}{\partial \mathbf{W}^{(2)}} \big\rangle \;+\; \big\langle \bar{\mathbf{W}}_{\mathrm{ICL}}, \frac{\partial \mathcal{L}}{\partial \bar{\mathbf{W}}} \big\rangle. \tag{90}$$

Per-block decomposition. Each of the three components decomposes as a sum over context blocks,

$$\nabla_{\mathrm{ICL}}\mathcal{L} \;=\; \sum_{r=1}^{n} \Big(g_r^{(1)} \;+\; g_r^{(2)} \;+\; g_r^{(\mathrm{o})} \Big), \tag{91}$$

with per-block signals

$$g_r^{(2)} \;=\; \frac{\partial \mathcal{L}}{\partial s_{N j_r}^{(2)}} \big\langle \mathbf{H}_{:,N}^{(1)},\; \mathbf{W}_{\mathrm{ICL}}^{(2)}\,\mathbf{H}_{:,j_r}^{(1)} \big\rangle, \tag{92}$$

$$g_r^{(1)} \;=\; \frac{\partial \mathcal{L}}{\partial s_{j_r i_r}^{(1)}} \big\langle \mathbf{H}_{:,j_r}^{(0)},\; \mathbf{W}_{\mathrm{ICL}}^{(1)}\,\mathbf{H}_{:,i_r}^{(0)} \big\rangle, \tag{93}$$

$$g_r^{(\mathrm{o})} \;=\; A_{N\,j_r}^{(2)} \big\langle \mathbf{z} - \mathbf{y},\; \bar{\mathbf{W}}_{\mathrm{ICL}}\,\mathbf{V}^{(2)}\,\mathbf{H}_{:,j_r}^{(1)} \big\rangle. \tag{94}$$

Three observations justify the decomposition. *(i)* Only the residual stream at the last position feeds the readout, so $\partial\mathcal{L}/\partial s_{ij}^{(2)}$ is nonzero only on row $i = N$. *(ii)* $\mathbf{W}_{\mathrm{ICL}}^{(1)} \propto \mathbf{M}$ shifts positions by one, so its inner product with $\frac{\partial \mathcal{L}}{\partial \mathbf{W}^{(1)}}$ is supported on adjacent within-block pairs (j_r, i_r). *(iii)* The residual stream at N decomposes additively over context positions through the layer-2 attention sum, and within each block the only term that contributes through $\bar{\mathbf{W}}_{\mathrm{ICL}}$ is the one at the label position j_r.

Two facts about the per-block signals. The per-block signals satisfy a clean dichotomy.

Distractor blocks contribute zero in expectation. For every $r \in \mathcal{D}$ and every $\bullet \in \{1, 2, \mathrm{o}\}$, the inner product inside $g_r^{(\bullet)}$ contracts the query embedding $\mathbf{e}_{\mathbf{a}_q}$ against an embedding of $\mathbf{x}_{i_r}$ or $\mathbf{x}_{j_r}$, both of which are rare tokens drawn independently of $\mathbf{a}_q$. By the same orthogonality argument used to exclude cross-concept gradient blocks in appendix A.3, distinct rare-token inner products vanish in expectation, so

$$\mathbb{E}\big[g_r^{(\bullet)} \,\big|\, r \in \mathcal{D} \big] \;=\; 0. \tag{95}$$

Relevant blocks contribute identical signals. For every $r \in \mathcal{R}$ the item is $\mathbf{a}_q$ and the label is $\mathbf{b}_q$ by definition; the only quantity that varies with r is the i.i.d. block positional offset, which the analysis of appendix A.3 shows to integrate out from the expected gradient. Consequently

$$\mathbb{E}_{\mathcal{D}(b)}\big[g_r^{(\bullet)} \,\big|\, r \in \mathcal{R} \big] \;=\; \tilde{g}^{(\bullet)}(b) \tag{96}$$

takes a single value $\tilde{g}^{(\bullet)}(b)$ that does not depend on the choice of $r \in \mathcal{R}$. Each signal carries exactly one factor of the relevant layer-2 attention weight $A_{N j_r}^{(2)} = e^{s_{\mathrm{rel}}}/Z_q$: explicitly in $g_r^{(\mathrm{o})}$, and through the chain rule in $g_r^{(2)}$ (the softmax derivative $\partial\mathcal{L}/\partial s_{N j_r}^{(2)} = A_{N j_r}^{(2)}\big(\partial_A \mathcal{L}|_{j_r} - \langle A_{N\cdot}^{(2)}, \partial_A \mathcal{L}\rangle\big)$) and in $g_r^{(1)}$ (the layer-1 output $\mathbf{H}_{:,j_r}^{(1)}$ reaches the readout only via the query's layer-2 attention to j_r). We therefore write

$$\tilde{g}^{(\bullet)}(b) \;=\; \frac{\kappa^{(\bullet)}(b)}{Z_q(b)}, \tag{97}$$

isolating the dominant b-channel, the shared denominator Z_q of eq. (88). The residual numerator $\kappa^{(\bullet)}(b)$ is built from the query residual $\mathbf{z} - \mathbf{y}$ and the softmax cross-term; it depends on b only through the *aggregate* relevant attention mass $b\,e^{s_{\mathrm{rel}}}/Z_q = \Theta(b/n)$, which the query absorbs into $\mathbf{y}$. Hence $\kappa^{(\bullet)}(b) = \kappa^{(\bullet)}(1)\,\big(1 + O(b/n)\big)$: the numerator is b-independent to leading order, with a correction of the same relative order as the denominator ratio computed below.

Conclusion. Taking the population expectation of eq. (91) under $\mathcal{D}(b)$ and substituting the two facts, the $n-b$ distractor blocks drop out and each of the b relevant blocks contributes $\sum_\bullet \tilde{g}^{(\bullet)}(b)$; using the factorization eq. (97),

$$\mathbb{E}_{\mathcal{D}(b)}\big[\nabla_{\mathrm{ICL}}\mathcal{L}(\mathbf{W})\big] \;=\; b\sum_{\bullet}\tilde{g}^{(\bullet)}(b) \;=\; \frac{b}{Z_q(b)}\sum_{\bullet}\kappa^{(\bullet)}(b). \tag{98}$$

This decomposition is exact; the additive remainder of the original bound is gone. The residual b-dependence beyond the explicit prefactor sits in the two factors $Z_q(b)$ and $\kappa^{(\bullet)}(b)$, both of which we now control *relative* to their $b=1$ values.

For the denominator, write $\bar{e}_{\mathrm{dis}} = e^{\bar{s}_{\mathrm{dis}}}$ for the mean distractor exponential; from eq. (88),

$$\frac{Z_q(1)}{Z_q(b)} \;=\; \frac{e^{s_{\mathrm{rel}}} + (n-1)\,\bar{e}_{\mathrm{dis}}}{b\,e^{s_{\mathrm{rel}}} + (n-b)\,\bar{e}_{\mathrm{dis}}} \;=\; 1 \;+\; \frac{(b-1)\,(\bar{e}_{\mathrm{dis}} - e^{s_{\mathrm{rel}}})}{Z_q(b)} \;=\; 1 + O\big(b/n\big), \tag{99}$$

since $Z_q(b) = \Theta(n)$ (distractor-dominated) and the numerator of the correction is $O(b)$. The numerator factor obeys the same relative bound, $\kappa^{(\bullet)}(b) = \kappa^{(\bullet)}(1)\,(1+O(b/n))$, by Fact 2. Specializing eq. (98) to $b=1$ gives $\mathbb{E}_{\mathcal{D}(1)}[\nabla_{\mathrm{ICL}}\mathcal{L}(\mathbf{W})] = \big(\sum_\bullet \kappa^{(\bullet)}(1)\big)/Z_q(1)$, so dividing the two instances of eq. (98) and inserting both relative bounds yields

$$\begin{aligned}\mathbb{E}_{\mathcal{D}(b)}\big[\nabla_{\mathrm{ICL}}\mathcal{L}(\mathbf{W})\big] \;&=\; b\,\frac{Z_q(1)}{Z_q(b)}\,\frac{\sum_\bullet \kappa^{(\bullet)}(b)}{\sum_\bullet \kappa^{(\bullet)}(1)}\,\mathbb{E}_{\mathcal{D}(1)}\big[\nabla_{\mathrm{ICL}}\mathcal{L}(\mathbf{W})\big] \\ &=\; b\,\mathbb{E}_{\mathcal{D}(1)}\big[\nabla_{\mathrm{ICL}}\mathcal{L}(\mathbf{W})\big]\,\big(1+O(b/n)\big).\end{aligned} \tag{100}$$

The remainder is now *multiplicative*: it scales the leading term rather than adding to it, so it is genuinely lower-order and vanishes as $n\to\infty$ with b fixed, the distractor-dominated regime assumed by the theorem. The ICL gradient is therefore proportional to b in this limit. □

E Competing Induction Heads

This appendix supports section 4.3, which studies the dependence of the winning ICL circuit on initialization in a 3-layer transformer.

E.1 Training details

We train a 3-layer, 1-head-per-layer attention-only transformer of width 256 on the in-context bigram task of appendix D.1 restricted to the all-rare regime ($f_r = 1.0$), so the only solution to the task is an in-context induction circuit. The vocabulary is 16 common and 16 rare tokens, sinusoidal positional embeddings with 4 frequencies (8 features), and each sequence consists of 4 bigram demonstrations followed by a query (context length 9); both burstiness and within-class variability are set to their neutral values ($b = 1$, $f_v = 0$). We train the simplified architecture of section A.3.2, with merged key-query matrices, fixed output-value projections, and orthogonal representations. The width $256 = 32 \times (1 + H)^L$ with $H = 1$ and $L = 3$ is chosen so that the model exactly admits the geometric construction of all three competing induction circuits on the manifold (base dimension 32, branching $(1 + H)^L = 8$). Optimization uses vanilla SGD at learning rate 0.1, micro-batch size 64, gradient-accumulation factor 2 (effective batch 128), for at most $12{,}000$ steps, with early-stop once the training cross-entropy drops below 0.5. We constrain the dynamics to the IMIR by projecting $\mathbf{W}$ onto the seven-dimensional sub-manifold spanned by the named circuit units $\alpha, \alpha', \beta, \beta', \beta'', \gamma, \gamma'$ after every SGD step. For the phase-transition scan of fig. 4, each seed samples $(s_\beta, s_{\beta''}) \sim \mathrm{Uniform}(0,\, 0.6)^2$ i.i.d., zeroes every other IMIR direction, and sets the initial strengths of β and β'' to s_β and $s_{\beta''}$ respectively; the seed is then plotted at $(s_\beta, s_{\beta''})$ and colored by the winning circuit. The winning circuit is identified post-training by causal ablation on a fixed evaluation batch: we rebuild three reduced transformers from the trained IMIR coordinates, retaining only one of $\{\alpha, \beta, \gamma\}$, $\{\alpha, \beta', \gamma'\}$, or $\{\alpha', \beta'', \gamma'\}$ at a time, and pick the one with the lowest cross-entropy at the query position. The same fixed batch is used across all seeds so the three ablation losses are directly comparable.

Compute resources. Each individual seed trains a 3-layer transformer of width 256 ($\sim$0.2 M learnable parameters) for at most $12{,}000$ SGD steps with effective batch size 128, taking a few minutes on a single GPU and using under 0.5 GB of GPU memory. The phase-transition scan parallelizes trivially across seeds: we run it on an 8-GPU node via per-seed sharding, with one persistent worker process per GPU pulling seeds from a shared queue. The scan reported in fig. 4 aggregates several thousand seeds and completes in a few hours of wall-clock on the 8-GPU node.

E.2 Training Dynamics

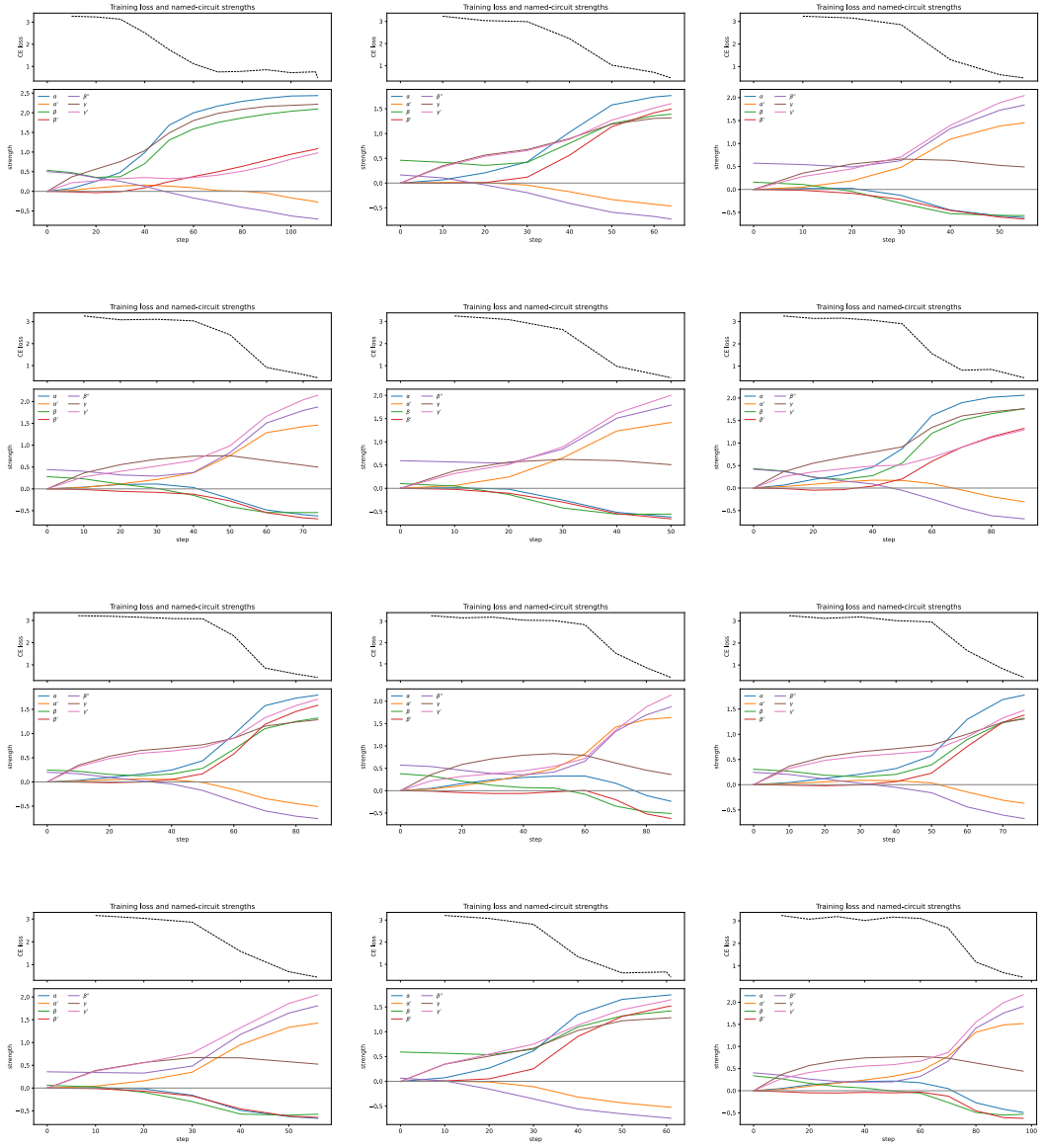

Figure 7: Training dynamics of 12 random seeds of a 3-layer, single-head Transformer trained on in-context bigrams (cf. section 4.3). For each seed, the top panel shows the training cross-entropy and the bottom panel the strengths of the seven named circuit units (α, α', β, β', β'', γ, γ'); the units that grow together indicate which of the three competing induction circuits wins.

F Towards Automated Interpretability

F.1 Training Details

We train a 5-layer, 1-head-per-layer attention-only transformer of width 1024 on the k-hop induction task with $k = 2$ hops, vocabulary of 16 common and 32 rare tokens, sinusoidal positional embeddings with 4 frequencies (8 features), and 9 context blocks per sequence. The all-rare regime $f_r = 1.0$ is used; an implicit-curriculum loss samples the target uniformly from the k hop endpoints $\{t_1, \ldots, t_k\}$ along the query chain, and within-chain blocks are emitted in reverse-hop order so that the target-bearing hop sits furthest from the query position. The width $1024 = 32 \times (1 + H)^L$ with $H = 1$ and $L = 5$ admits the orthogonal-subspace construction of all IMIR directions exactly. Optimization uses AdamW at learning rate 10^{-3}, micro-batch size 1, gradient-accumulation factor 32 (effective batch 32), for $50{,}000$ optimizer steps; the random seed is fixed at 6.

Compute resources. The model has approximately 32 M learnable parameters; the bulk of the cost comes from the layer-wise attention matrices over the relatively long context. We launch training via `torchrun` on a single 8-GPU node, with the model wrapped in DistributedDataParallel and gradients all-reduced on the final micro-batch of each accumulation window. One full training run (50,000 optimizer steps, effective batch $32 \cdot 8 = 256$ across ranks) takes a few hours of wall-clock on this hardware. The circuit-detection routine of appendix F.2 runs post-training on a single GPU and completes in minutes: each greedy round evaluates one forward pass per remaining IMIR direction on a fixed eval batch, so the total cost is quadratic in the number of non-zero IMIR directions of the trained model.

F.2 Circuit Detection Algorithm

Our circuit-detection routine performs greedy backward elimination on the manifold coordinates of the trained model. Starting from the full set of IMIR directions, it repeatedly drops the single direction whose ablation increases the loss the least, and stops when even the cheapest available ablation would push the loss on a fixed evaluation batch above a user-supplied budget τ. The surviving directions, sorted by their per-circuit ablation cost, are the algorithm's identification of the model's essential circuits.

Algorithm 1 Greedy backward circuit elimination on the IMIR.

Require: trained transformer $f_{\mathbf{W}}$, fixed eval batch $\mathcal{B}$, loss budget τ, free-ablation slack $\rho \geq 1$
Ensure: essential circuit indices $\mathcal{A}$ and strengths $\{s_i\}_{i \in \mathcal{A}}$

$\{(\mathbf{e}_i, s_i)\}_i \leftarrow \textsc{ReduceToManifold}(\mathbf{W})$ ▷ basis directions on $\mathcal{S}$ and their strengths
$\mathbf{W} \leftarrow \sum_i s_i \, \mathbf{e}_i$ ▷ project the trained weights onto $\mathcal{S}$
$\mathcal{A} \leftarrow \{i : s_i \neq 0\}$; $\mathcal{L}_{\text{cur}} \leftarrow \mathcal{L}(f_{\mathbf{W}}; \mathcal{B})$

while $\mathcal{L}_{\text{cur}} < \tau$ **and** $\mathcal{A} \neq \emptyset$ **do**
  $i_\star \leftarrow \textsc{None}$; $\mathcal{L}_\star \leftarrow +\infty$
  for all $i \in \mathcal{A}$ in random order **do**
    $\mathcal{L}_i \leftarrow \mathcal{L}\big(f_{\mathbf{W} - s_i \mathbf{e}_i}; \mathcal{B}\big)$ ▷ tentative ablation of direction i
    if $\mathcal{L}_i < \mathcal{L}_\star$ **then**
      $\mathcal{L}_\star \leftarrow \mathcal{L}_i$; $i_\star \leftarrow i$
    end if
    if $\mathcal{L}_\star \leq \rho \cdot \mathcal{L}_{\text{cur}}$ **then**
      break ▷ free ablation found; skip rest of scan
    end if
  end for
  if $\mathcal{L}_\star \geq \tau$ **then**
    break ▷ even the cheapest ablation crosses the budget
  end if
  $\mathbf{W} \leftarrow \mathbf{W} - s_{i_\star} \, \mathbf{e}_{i_\star}$ ▷ commit the ablation
  $\mathcal{A} \leftarrow \mathcal{A} \setminus \{i_\star\}$; $\mathcal{L}_{\text{cur}} \leftarrow \mathcal{L}_\star$
end while

return $\mathcal{A}$, sorted by per-circuit ablation cost $\mathcal{L}(f_{\mathbf{W} - s_i \mathbf{e}_i}; \mathcal{B}) - \mathcal{L}_{\text{cur}}$ in descending order.

The free-ablation slack ρ is a small optimization: once the inner scan finds an ablation whose cost is within a factor ρ of the current loss, we accept it without finishing the round. This keeps each round at $O(|\mathcal{A}|)$ in the worst case but typically much cheaper when several directions are obviously prunable. The randomized scan order avoids the bias of always probing the same prefix first. The fixed evaluation batch $\mathcal{B}$ and the deterministic RNG seed used by the embedding layer ensure that two evaluations of the same model produce identical losses, which is required for the greedy comparisons to be meaningful. We use the values $\tau = 100$ and $\rho = 1.01$ in our experiments.

F.3 Discovered Circuits

We run the greedy backward-elimination procedure of Algorithm 1 on two trained 5-layer transformers solving the $k = 2$ hop induction task, and report the discovered circuits in tables 2 and 3. In addition to the four units one would expect from a canonical induction-head construction — a previous-token head, two item-match heads chained through value rotations, and a readout that inverts the chain — the algorithm consistently recovers a small number of *aiding* units that further lift accuracy without being part of any standard induction template. We separate the two groups in the tables for clarity.

The aiding units are interesting on their own: removing them drops the isolated subcircuit's accuracy from ~ 0.92 down to ~ 0.6 (table 2, $0.922 \to 0.594$), which is much more than a slight refinement. They appear to compose with the canonical induction units to form a slightly redundant ensemble that exploits multiple paths through the residual stream simultaneously. This kind of structure would be invisible to existing automated circuit discovery algorithms with granularity at the attention head level. We list the four canonical units under *Essential* and the recovered extras under *Aiding* in each circuit's table.

Table 2: Discovered subcircuit C_1 in a trained 5-layer transformer ($k = 2$ hops). The accuracy table (left) reports top-1 accuracy on a fixed evaluation batch at four levels of restriction; the circuit-unit table (right) lists the seven IMIR directions kept by the greedy elimination, separated into the four canonical induction units (*Essential*) and three non-canonical helpers (*Aiding*).

Configuration	**Acc.**
Trained model	0.938
Full IMIR (1428 dirs.)	0.953
Recovered C_1 (7 units)	0.922
Essentials (4 units)	0.594

Layer	**Direction**	**Type**	**Strength**
1	$\mathbf{M}$	Essential	$+2.94$
3	$\mathbf{I}^{(t)}\,(\mathbf{V}^{(1,1)})^{+}$	Essential	$+9.24$
4	$\mathbf{V}^{(3,1)}\,\mathbf{M}\,(\mathbf{V}^{(1,1)})^{+}$	Essential	$+3.68$
out	$\mathbf{I}^{(t)}\,(\mathbf{V}^{(3,1)}\,\mathbf{V}^{(4,1)})^{+}$	Essential	$+1.42$
2	$\mathbf{I}^{(t)}\,(\mathbf{V}^{(1,1)})^{+}$	Aiding	$+3.81$
3	$\mathbf{V}^{(1,1)}\,\mathbf{V}^{(2,1)}\,\mathbf{I}^{(p)}$	Aiding	-1.33
4	$\mathbf{V}^{(3,1)}\,\mathbf{I}^{(t)}$	Aiding	$+1.14$

Table 3: Discovered subcircuit C_2 in a second trained 5-layer transformer ($k = 2$ hops). Compared with C_1, C_2 lives entirely on layers $1, 2, 3$, uses canonical token-identity directions on every essential unit, and recovers a single aiding unit.

Configuration	**Acc.**
Trained model	0.906
Full IMIR (1428 dirs.)	0.906
Recovered C_2 (5 units)	0.844
Essentials (4 units)	0.719

Layer	**Direction**	**Type**	**Strength**
1	$\mathbf{M}$	Essential	$+2.35$
2	$\mathbf{I}^{(t)}\,(\mathbf{V}^{(1,1)})^{+}$	Essential	$+2.68$
3	$\mathbf{I}^{(t)}\,(\mathbf{V}^{(1,1)})^{+}$	Essential	$+5.79$
out	$\mathbf{I}^{(t)}\,(\mathbf{V}^{(2,1)}\,\mathbf{V}^{(3,1)})^{+}$	Essential	$+2.94$
2	$\mathbf{I}^{(t)}$	Aiding	-0.47