Interleaved Head Attention

Sai Surya Duvvuri[*] UT Austin
Chanakya Ekbote[*] MIT
Rachit Bansal Harvard University
Rishabh Tiwari UC Berkeley
Devvrit Khatri UT Austin
David Brandfonbrener Meta
Paul Pu Liang[‡] MIT
Inderjit Dhillon[‡] UT Austin
Manzil Zaheer[‡] Meta

Abstract

Multi-Head Attention (MHA) is the core computational primitive underlying modern Large Language Models (LLMs). However, within a single attention layer, MHA has an intrinsic *linear scaling* limitation: H attention heads produce exactly H independent attention matrices, with no communication between heads during attention computation. While deep Transformers can compose information across layers, this places much of the burden of building and combining intermediate relations on depth, potentially requiring additional layers to expose interactions useful for downstream computation. Our focus is complementary: increasing the compositional bandwidth available within each attention layer. To this end, we propose *Interleaved Head Attention* (IHA), which enables within-layer cross-head mixing by constructing P *pseudo-heads* per head, typically with $P = H$, where each pseudo query, key, and value is a learned linear combination of all H original queries, keys, and values, respectively. Interactions between pseudo-query and pseudo-key heads induce up to P^2 attention patterns per head with modest parameter overhead $\mathcal{O}(H^2P)$. We provide theory showing improved parameter efficiency on Polynomial Filters, where IHA uses $\Theta(\sqrt{k}n^2)$ parameters versus $\Theta(kn^2)$ for MHA, and on the order-sensitive CPM-3 task, where IHA uses $\lceil\sqrt{N_{\max}}\rceil$ heads versus $N_{\max}$ for MHA. On real-world benchmarks, IHA improves Multi-Key retrieval on RULER by 10–20% at 4k–16k context lengths and, after OpenThoughts reasoning fine-tuning, improves GSM8K by 5.8% and MATH-500 by 2.8% under majority voting over full attention, with modest throughput overhead.

1 Introduction

Multi-head attention (MHA) is the core computational primitive underlying modern Large Language Models [27]. In a standard MHA layer, each of the H heads computes attention independently: a query head attends only to its corresponding key–value projections, and heads do not communicate during attention computation. Thus, a single MHA layer produces exactly H independently parameterized attention matrices. Because each head produces only one attention map, a single MHA layer has limited ability to represent multiple token-to-token relations at once. This matters in multi-step reasoning, where the answer often depends not on one direct association, but on a chain of intermediate associations. For example, to answer "*Where was the author of The Hobbit born?*", the model must first associate *The Hobbit* with *J.R.R. Tolkien*, and then associate *J.R.R. Tolkien* with *born in South Africa*. Stacked Transformer layers can compose such information over depth, but this

[*]Equal contribution. Correspondence to saisurya@cs.utexas.edu and cekbote@mit.edu.
[‡]Equal mentorship.

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

places much of the burden of building and combining intermediate relations on successive layers. Our goal is complementary: we aim to increase the number of token-to-token relations that can interact within a single attention layer, potentially reducing the need to defer such interactions to later layers. To formalize this within-layer limitation, we study the Polynomial Filter problem [8, 4, 17, 10] as a controlled proxy for multi-step information propagation. In this setting, an r-step dependency means that information reaches a token through r successive relation applications: $r = 1$ is direct, while $r = 2$ composes two relation steps. In the one-layer setting, we show that an individual head of MHA can realize only one such dependency length at a time. Therefore, representing k distinct dependency lengths in parallel within one layer requires $\Theta(k)$ heads in our construction, leading to parameter requirements that scale linearly with the number of dependency patterns.

This motivates a mechanism that increases the number of effective attention patterns available within a layer, without simply adding more heads. In this work, we propose Interleaved Head Attention (IHA), a drop-in replacement for MHA that addresses this within-layer bottleneck. While standard MHA computes each head in isolation within a layer, IHA enables cross-head mixing before attention is computed. For each head, IHA constructs P pseudo-queries, pseudo-keys, and pseudo-values (typically $P = H$) as learned linear combinations of the original heads' query, key, and value projections. Interacting the P pseudo-queries with the P pseudo-keys induces up to P^2 attention patterns per head, yielding quadratic scaling in P within a layer (and typically in H when $P = H$). Unlike Talking-Heads and Knocking [25, 31], which mix heads at the level of attention logits/weights, IHA performs this mixing before attention while preserving the standard attention operator, making it compatible with efficient kernels such as FlashAttention [7]. The extra parameters come only from pseudo-head mixing and scale as $\mathcal{O}(H^2P)$, which is modest since $H, P \ll d_{\text{model}}$, where d_{model} denotes the model dimension. On Polynomial Filters of order k, we show that one-layer MHA needs $\Theta(kn^2)$ parameters in our construction, whereas one-layer IHA matches the same expressivity with $\Theta(\sqrt{k}n^2)$ parameters using $\mathcal{O}(\sqrt{k})$ heads.

We run extensive experiments on long-context modeling and supervised fine-tuning for reasoning. On the RULER long-context benchmark [15], IHA achieves 10–20% relative improvements over full attention on Multi-Key Retrieval across 4k to 16k context lengths. After fine-tuning on OpenThoughts [12] for reasoning, IHA outperforms full attention baselines by 5.8% on GSM8K [6] and 2.8% on MATH-500 [14] under majority voting. Our contributions are:

Main Contributions.

1. **Interleaved Head Attention (IHA).** We introduce IHA, a drop-in alternative to MHA that increases within-layer compositional bandwidth by enabling cross-head mixing before attention is computed. IHA constructs P pseudo-queries, pseudo-keys, and pseudo-values per head (typically $P = H$); interactions between pseudo-queries and pseudo-keys induce up to P^2 attention patterns per head. (Sec. 4)
2. **Theory.** We prove that one-layer IHA strictly generalizes one-layer MHA while adding only $4H^2P$ parameters. We also show improved asymptotic scaling on two synthetic benchmarks: $\Theta(\sqrt{k}\,n^2)$ **vs.** $\Theta(kn^2)$ parameters on Polynomial Filters, and $\Theta(N_{\max}^{2.5})$ **vs.** $\Theta(N_{\max}^3)$ attention cost on CPM-3 compared with the best-known one-layer MHA construction. (Subsec. 4.1, Subsec. 4.3)
3. **Empirical results.** Under FLOP-matched training, IHA improves Multi-Key retrieval on RULER by **10–20%** (4k–16k) and, after OpenThoughts fine-tuning, improves GSM8K by **5.8%** and MATH-500 by **2.8%** (Maj@16) over full attention, with modest throughput overhead, IHA is within **1–2%** of MHA. (Sec. 5, App. E)

2 Related Work

Recent work has begun to characterize when standard multi-head attention (MHA) is inefficient for compositional, multi-step reasoning. A recurring theme is that many reasoning primitives require both aggregating evidence spread across positions and composing intermediate transformations before producing an answer, as in relation/function composition, Match-3 [16, 22], and multi-hop QA benchmarks such as bAbI [28]. We study this behavior through two controlled proxies: polynomial filters [8, 4], which model k-step aggregation, and CPM-3, which isolates order-sensitive composition and counting. In these settings, IHA realizes the relevant primitives more efficiently than MHA,

yielding quadratic improvements in the heads/parameters required by the corresponding one-layer constructions. In parallel, prior work also extends attention by enriching token interactions or improving efficiency. Higher-order mechanisms model interactions among three or more tokens, including simplicial/trilinear attention [5, 21], Strassen-style constructions [16], and multi-token attention [11], often using specialized operators. Efficiency-oriented methods such as MQA/GQA [24, 1] reduce KV-cache cost by sharing key/value projections, while Talking-Heads and Knocking-Heads [25, 31] mix information across heads at the logit/weight level, and Differential Attention [29] modifies attention maps. IHA takes a different point in this design space: it mixes query, key and value representations across heads before standard attention, inducing up to P^2 pseudo-query/pseudo-key interaction patterns per head with only $\mathcal{O}(H^2P)$ additional mixing parameters. Because the attention operation itself is unchanged, IHA remains compatible with FlashAttention [7] and keeps the KV cache in the original per-head representation. Further discussion appears in App. B.

3 Background

Notation. We denote matrices with bold uppercase letters (e.g., $\boldsymbol{X}$) and vectors with bold lowercase (e.g., $\boldsymbol{x}$). For an input sequence of N tokens with embedding dimension D, the input matrix is $\boldsymbol{X} \in \mathbb{R}^{N\times D}$. We use h to index attention heads, $d = D/H$ for per-head dimension where H is the total number of heads, and $\mathrm{softmax}(\cdot)$ for row-wise softmax. We use $[\boldsymbol{A}, \boldsymbol{B}]$ for column-wise (horizontal) concatenation and $[\boldsymbol{A}; \boldsymbol{B}]$ for row-wise (vertical) stacking. In Algorithm 1, we use $\texttt{reshape}(\boldsymbol{T}, [d_1, \ldots, d_k])$ to reshape tensor $\boldsymbol{T}$ to dimensions $d_1 \times \cdots \times d_k$, `einsum` for Einstein summation following NumPy conventions (e.g., `'mhp,nmd->hpnd'` contracts index m and permutes the result), and `merge_pseudo` to interleave the pseudo-head dimension P into the sequence dimension, transforming shape $(H, P, N, d) \to (H, NP, d)$. Throughout, $\mathbb{1}[\cdot]$ denotes the indicator function; e.g., $\mathbb{1}[x = y] = 1$ if $x = y$ and 0 otherwise. We also write this equivalently as $\mathbf{1}_{(x=y)}$.

3.1 Multi-head Attention and Polynomial Filters

MHA. We use standard causal scaled dot-product self-attention [27], with the causal mask implicit.

Definition 1. (MHA) Given an input $\boldsymbol{X} \in \mathbb{R}^{N\times D}$ and H heads with per-head dimension $d := D/H$, head $h \in [H]$ forms: $\boldsymbol{Q}^{(h)} := \boldsymbol{X}\boldsymbol{W}_Q^{(h)}$, $\boldsymbol{K}^{(h)} := \boldsymbol{X}\boldsymbol{W}_K^{(h)}$, $\boldsymbol{V}^{(h)} := \boldsymbol{X}\boldsymbol{W}_V^{(h)}$, where $\boldsymbol{W}_Q^{(h)}, \boldsymbol{W}_K^{(h)}, \boldsymbol{W}_V^{(h)} \in \mathbb{R}^{D\times d}$ are learned projections. The per-head output is:

$$\widetilde{\boldsymbol{X}}^{(h)} \;=\; \mathrm{softmax}\!\left(\frac{1}{\sqrt{d}}\,\boldsymbol{Q}^{(h)}{\boldsymbol{K}^{(h)}}^{\top}\right)\boldsymbol{V}^{(h)} \quad \forall h \in [H]$$

where the softmax is applied row-wise (and respects the causal mask). Finally, the head outputs are concatenated and projected as: $\widetilde{\boldsymbol{X}} := \big[\widetilde{\boldsymbol{X}}^{(1)}, \ldots, \widetilde{\boldsymbol{X}}^{(H)}\big]\boldsymbol{W}_O \in \mathbb{R}^{N\times D}$, (with $\boldsymbol{W}_O \in \mathbb{R}^{D\times D}$ being the output projection).

Polynomial filters. Polynomial graph filters are a core primitive in graph signal processing.

Definition 2. (Polynomial Graph Filters) Given a graph adjacency matrix $\boldsymbol{A} \in \mathbb{R}^{N\times N}$ and input features $\boldsymbol{X} \in \mathbb{R}^{N\times d}$, one computes the polynomial filter bank, $\widetilde{\boldsymbol{X}} = \big[\boldsymbol{X},\ \boldsymbol{A}\boldsymbol{X},\ \ldots,\ \boldsymbol{A}^{k-1}\boldsymbol{X}\big]$

The above defines a node representation which aggregates information from nodes up to k hops away. This is a controlled proxy for *multi-step reasoning* in language tasks such as [28]: consider the story *(i)* "Mary picked up the football." *(ii)* "Mary went to the kitchen." and the question "Where is the football?" A model must connect *football* to *Mary* (fact *i*) and then *Mary* to *kitchen* (fact *ii*), i.e., a two-step composition. If we build a fact graph with one node per sentence and connect two nodes when they share an entity (here, facts *i* and *ii* are connected via *Mary*), then one-hop aggregation $\boldsymbol{A}\boldsymbol{X}$ retrieves the directly linked fact, while two-hop aggregation $\boldsymbol{A}^2\boldsymbol{X}$ captures precisely the required two-step chain. In a language model, this graph is implicitly induced by the particular input (and thus varies across examples); however, as a controlled proxy for analyzing multi-step information

propagation, we treat $\boldsymbol{A}$ as fixed for a given instance and study how well attention can realize the corresponding k-hop operators. A natural question is therefore: how many attention heads does MHA need to represent (or approximate) these k-hop aggregations, and to produce the full filter bank $\left[\boldsymbol{X}, \boldsymbol{A}\boldsymbol{X}, \ldots, \boldsymbol{A}^{k-1}\boldsymbol{X}\right]$ in parallel? We aim to answer this question via Thm. 1.

Theorem 1. (MHA Polynomial Filter Representation): Given a graph adjacency matrix $\boldsymbol{A} \in \mathbb{R}^{N\times N}$ and input features $\boldsymbol{X} \in \mathbb{R}^{N\times d}$, we concatenate the input with the identity matrix: $\widehat{\boldsymbol{X}} = [\boldsymbol{X}, \boldsymbol{I}]$. Then, representing the polynomial filter bank $\widetilde{\boldsymbol{X}} = [\boldsymbol{X}, \boldsymbol{A}\boldsymbol{X}, \ldots, \boldsymbol{A}^{k-1}\boldsymbol{X}]$ using one-layer linear MHA (without softmax) requires at least k attention heads, with parameter complexity $2N(N+d)k + d(N+d)k$. Note that we assume $d \ll N$ and $k \ll N$.

Proof sketch. **Why does MHA need k heads?** We augment $\boldsymbol{X}$ with positional encodings $\widehat{\boldsymbol{X}} = [\boldsymbol{X}, \boldsymbol{I}_N]$. In linear (no-softmax) attention, head h outputs: $\boldsymbol{O}_h = \widehat{\boldsymbol{X}}\boldsymbol{W}_Q^{(h)}{\boldsymbol{W}_K^{(h)}}^\top \widehat{\boldsymbol{X}}^\top \widehat{\boldsymbol{X}}\boldsymbol{W}_V^{(h)} \in \mathbb{R}^{N\times d}$. To realize $\boldsymbol{A}^i\boldsymbol{X}$, choose $\boldsymbol{W}_Q^{(h)} = \left[\boldsymbol{0}_{d\times N}; \boldsymbol{A}^i\right]$, $\boldsymbol{W}_K^{(h)} = [\boldsymbol{0}_{d\times N}; \boldsymbol{I}_N]$, $\boldsymbol{W}_V^{(h)} = [\boldsymbol{I}_d; \boldsymbol{0}_{N\times d}]$, which gives $\widehat{\boldsymbol{X}}\boldsymbol{W}_Q^{(h)}{\boldsymbol{W}_K^{(h)}}^\top \widehat{\boldsymbol{X}}^\top = \boldsymbol{A}^i$ and $\widehat{\boldsymbol{X}}\boldsymbol{W}_V^{(h)} = \boldsymbol{X}$, hence $\boldsymbol{O}_h = \boldsymbol{A}^i\boldsymbol{X}$. Since each head yields a single power $\boldsymbol{A}^i$, producing $\boldsymbol{A}^0, \ldots, \boldsymbol{A}^{k-1}$ requires at least k heads. See Subsec. D.2 for more details. Note that the assumption that $d \ll N$ and $k \ll N$ is consistent with prior works on polynomial graph filters such as [8, 4, 17, 10].

Intuition. The fundamental bottleneck in MHA is head isolation: each head uses only its own query/key/value projections and therefore contributes at most one attention pattern. Thus, targets requiring k distinct relational patterns typically require $\Omega(k)$ heads in a single MHA layer. IHA relaxes this head-per-pattern constraint via cross-head mixing, as described in Sec. 4.

4 Interleaved-Head Attention (IHA)

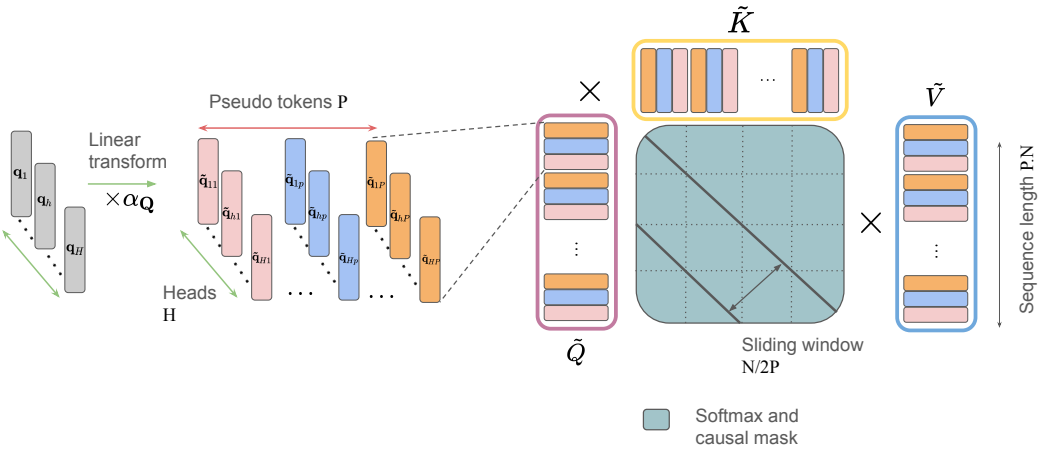

Figure 1: **Overview of Interleaved Head Attention (IHA).** First, the model generates P pseudo-tokens for each of the H original heads via a learned linear transformation ($\times\alpha_{\mathbf{Q}}$) operating on the heads axis (green). These tokens are then interleaved to create an expanded sequence of length $P \cdot N$. Finally, standard causal self-attention is computed on this expanded sequence, utilizing a sliding window (e.g., $N/2P$) to manage computational complexity while enabling cross-head interaction. Different linear transforms are used in query, key and values.

IHA overcomes the one-to-one coupling of MHA by constructing, for each head, P pseudo-queries, pseudo-keys, and pseudo-values as learned linear combinations of the H queries, keys and values respectively (typically $P = H$). This enlarges the set of query/key/value projections and allows attention to mix information across heads, rather than restricting each query head to its paired key–value head. Within each head, the P pseudo-queries attending to the P pseudo-keys can induce up to P^2 distinct attention patterns, and this mechanism is applied independently across heads. The added

expressivity incurs only modest overhead: pseudo-mixing weights scale as $\mathcal{O}(H^2P)$, which is small relative to the overall parameter budget since H and P are typically much smaller than the model dimension. The IHA algorithm is given in Alg. 1 and the architecture figure can be found in Fig. 1.

Algorithm 1. Interleaved-Head Attention (Figure 1)

Input: $\boldsymbol{X} \in \mathbb{R}^{N\times D}$, heads H, pseudo-heads P, head dim $d = D/H$
Params: $\boldsymbol{W}_Q, \boldsymbol{W}_K, \boldsymbol{W}_V, \boldsymbol{W}_O \in \mathbb{R}^{D\times D}$; $\boldsymbol{\alpha}^{Q,K,V} \in \mathbb{R}^{H\times H\times P}$; $\boldsymbol{R} \in \mathbb{R}^{H\times P}$
Output: $\widetilde{\boldsymbol{X}} \in \mathbb{R}^{N\times D}$

// Step 1: Project and reshape to per-head representations
$\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V} \leftarrow \texttt{reshape}(\boldsymbol{X}\boldsymbol{W}_Q, \boldsymbol{X}\boldsymbol{W}_K, \boldsymbol{X}\boldsymbol{W}_V)$ ▷ $\to [N, H, d]$
// Step 2: Generate pseudo-heads via learned mixing across heads
$\widetilde{\boldsymbol{Q}} \leftarrow \texttt{einsum}(\texttt{'mhp,nmd} \to \texttt{hpnd'}, \boldsymbol{\alpha}^Q, \boldsymbol{Q})$ ▷ (H, P, N, d)
$\widetilde{\boldsymbol{K}} \leftarrow \texttt{einsum}(\texttt{'mhp,nmd} \to \texttt{hpnd'}, \boldsymbol{\alpha}^K, \boldsymbol{K})$ ▷ (H, P, N, d)
$\widetilde{\boldsymbol{V}} \leftarrow \texttt{einsum}(\texttt{'mhp,nmd} \to \texttt{hpnd'}, \boldsymbol{\alpha}^V, \boldsymbol{V})$ ▷ (H, P, N, d)
// Step 3: Merge pseudo dimension into sequence dimension (interleaved)
$\overline{\boldsymbol{Q}}, \overline{\boldsymbol{K}}, \overline{\boldsymbol{V}} \leftarrow \texttt{merge_pseudo}(\widetilde{\boldsymbol{Q}}, \widetilde{\boldsymbol{K}}, \widetilde{\boldsymbol{V}})$ ▷ $\to [H, NP, d]$
// Step 4: Standard scaled dot-product attention per head
$\overline{\boldsymbol{O}}_h \leftarrow \mathrm{softmax}\left(\frac{1}{\sqrt{d}}\overline{\boldsymbol{Q}}_h\overline{\boldsymbol{K}}_h^\top\right)\overline{\boldsymbol{V}}_h, \quad \forall h$ ▷ $\overline{\boldsymbol{O}} \in \mathbb{R}^{H\times NP\times d}$
// Step 5: Unmerge and collapse pseudo-heads
$\boldsymbol{P} \leftarrow \texttt{reshape}(\overline{\boldsymbol{O}}, \texttt{[H,N,P,d]})$ ▷ Reshape to (H, N, P, d)
$\boldsymbol{O} \leftarrow \texttt{einsum}(\texttt{'hp,hnpd} \to \texttt{hnd'}, \boldsymbol{R}, \boldsymbol{P})$ ▷ Collapse: (H, N, d)
// Step 6: Concatenate heads and project output
$\widetilde{\boldsymbol{X}} \leftarrow \texttt{reshape}(\boldsymbol{O}, \texttt{[N,D]}) \cdot \boldsymbol{W}_O$
return $\widetilde{\boldsymbol{X}}$

In Alg. 1, Step 2 constructs, for each head h and token index $n \in [N]$, P pseudo-head tokens by taking learned linear combinations of the H original heads' query/key/value projections. Step 3 interleaves them by replacing each original token with P consecutive virtual tokens, so the sequence becomes $(1,1), (1,2), \ldots, (1,P), (2,1), \ldots, (N,P)$ where (n,p) denotes the p-th pseudo-head token at position n. Step 4 then runs standard scaled dot-product attention once on this NP length sequence. This lets different pseudo-head tokens attend differently, yielding up to P^2 attention patterns per head (and HP^2 overall) without custom kernels. Interleaving is useful with RoPE because RoPE depends on the position index: giving each (n,p) its own virtual position assigns each pseudo-head token a distinct RoPE phase, and variable-length inference is handled by generating RoPE for length NP. The procedure is also compatible with FlashAttention [7], since Step 4 is standard attention. For the theoretical analysis, App. C gives an equivalent, more algebraic formulation of IHA that omits interleaving and the output projection. Since our proofs do not use RoPE positional encodings, interleaving is unnecessary, and dropping the projection cleanly isolates the core pseudo-head attention computation.

In the following sections, we (i) establish a strict expressivity separation by showing that the class of functions representable by MHA is contained in (and generally a strict subset of) those representable by IHA, (ii) analyze IHA on two synthetic benchmarks (the polynomial filter and CPM3 tasks; defined later), and (iii) experimentally show that IHA outperforms other attention variants.

4.1 IHA Strictly Generalizes MHA

We formalize the sense in which IHA strictly generalizes standard multi-head attention (MHA) while making the parameter overhead explicit. Fix a sequence length N and number of heads H. Let $\mathcal{M}$ denote the set of all single-layer H-head MHA modules with query, key, value matrices as defined in Def. 1, requiring Q parameters in total. Let $\mathcal{P}_P$ denote the corresponding set of H-head IHA modules as in Alg. 1 (and App. C) with P pseudo-heads per head, whose query, key, and value weight matrices have the same dimensions as those in MHA (hence also contributing Q parameters), but which additionally introduce mixing tensors $\alpha^Q, \alpha^K, \alpha^V \in \mathbb{R}^{H\times H\times P}$ and a collapse map $\boldsymbol{R} \in \mathbb{R}^{H\times HP}$.

Theorem 2. (IHA Superset Property). For any $P \geq 1$, every module in $\mathcal{P}_P$ has $Q + 4H^2P$ parameters (namely Q from the query, key, value projections plus $3H^2P$ from $\alpha^Q, \alpha^K, \alpha^V$

and H^2P from $\boldsymbol{R}$). Moreover, for every $P \geq 1$, $\mathcal{M} \subseteq \mathcal{P}_P$ and for every $P \geq 2$ the inclusion is strict: $\mathcal{M} \subsetneq \mathcal{P}_P$.

Proof sketch. *Inclusion.* Fix any MHA instance with weights $\{\boldsymbol{W}_Q^{(m)}, \boldsymbol{W}_K^{(m)}, \boldsymbol{W}_V^{(m)}\}_{m=1}^H$. We construct an IHA instance (with any chosen $P \geq 1$) that computes the same function by selecting parameters that ignore all but one pseudo-channel. Specifically, for all $m, i \in [H]$ and all $j \in [P]$, set $\alpha^Q_{m,i,j} = \alpha^K_{m,i,j} = \alpha^V_{m,i,j} = \mathbb{1}[m = i]$ and choose $\boldsymbol{R}$ to select only the $(i, 1)$ pseudo-block: $\boldsymbol{R}_{i,\,(i'-1)P+j} = \mathbb{1}[i' = i \text{ and } j = 1]$. Then $\widetilde{\boldsymbol{Q}}_{i,j} = \boldsymbol{X}\boldsymbol{W}_Q^{(i)}$, $\widetilde{\boldsymbol{K}}_{i,j} = \boldsymbol{X}\boldsymbol{W}_K^{(i)}$, and $\widetilde{\boldsymbol{V}}_{i,j} = \boldsymbol{X}\boldsymbol{W}_V^{(i)}$ for all j, so the stacked attention produces copies of the original MHA head outputs and the collapse map returns exactly the MHA outputs. Hence $\mathcal{M} \subseteq \mathcal{P}_P$.

Strictness. Consider the repeated-token subspace $\mathcal{S} = \{\boldsymbol{X} = \mathbf{1}_N \boldsymbol{x}^\top : \boldsymbol{x} \in \mathbb{R}^d\}$, $N \geq 2$. On $\mathcal{S}$, every MHA head has identical queries/keys/values at all positions, so each score matrix has identical rows and the row-wise softmax is uniform; consequently each head output reduces to $\mathbf{1}_N \boldsymbol{x}^\top \boldsymbol{W}_V^{(m)}$, which is linear in $\boldsymbol{x}$. Therefore every H-head MHA module is linear on $\mathcal{S}$. In contrast, IHA with $P \geq 2$ can be parameterized (using only the additional $4H^2P$ mixing/collapse parameters on top of the same projections) so that the stacked attention produces a nonlinear function of $\boldsymbol{x}$ on $\mathcal{S}$: for example, by creating two pseudo-query/key variants with opposite signs and choosing the pseudo-values/collapse so that the output involves a difference of softmax-normalized terms depending on the cosine score $\langle \boldsymbol{x}^\top \boldsymbol{W}_Q, \boldsymbol{x}^\top \boldsymbol{W}_K \rangle$, yielding a nonlinearity (e.g., a $\tanh$-like dependence). Since no MHA can be nonlinear on $\mathcal{S}$, this IHA mapping cannot be represented by any MHA, proving $\mathcal{M} \subsetneq \mathcal{P}_P$ for all $P \geq 2$. For a detailed proof please refer to Subsec. D.1.

4.2 Representing Polynomial Filters using IHA

As established in Subsec. 3.1, polynomial graph filters provide a clean proxy for multi-hop information propagation (and thus multi-step reasoning): the i-th term $\boldsymbol{A}^i\boldsymbol{X}$ aggregates information from i-hop neighborhoods. Computing $\boldsymbol{X}, \boldsymbol{A}\boldsymbol{X}, \ldots, \boldsymbol{A}^{k-1}\boldsymbol{X}$ in parallel captures k-step composition in a controlled setting. In Thm. 3, we show how IHA represents this polynomial filter.

Theorem 3. (Representing Polynomial Filters). Given a graph adjacency matrix $\boldsymbol{A} \in \mathbb{R}^{N\times N}$ and input features $\boldsymbol{X} \in \mathbb{R}^{N\times d}$ with $d \ll N$, we concatenate the input with the identity matrix $\widehat{\boldsymbol{X}} = [\boldsymbol{X}, \boldsymbol{I}]$. For one-layer MHA architectures *without softmax* that are capable of representing all polynomial filter constructions with k heads (with $k \ll N$), there exists an equivalent one-layer attention-based IHA architecture without softmax that requires only $\lceil\sqrt{k}\rceil$ heads. In terms of parameter complexity, an MHA construction with k heads requires $2N(N+d)k + d(N+d)k$ parameters, whereas the equivalent IHA construction with $\lceil\sqrt{k}\rceil$ heads requires $2N(N+d)\lceil\sqrt{k}\rceil + d(N+d)\lceil\sqrt{k}\rceil^2 + 4\lceil\sqrt{k}\rceil^3$ parameters. Here d denotes the embedding dimension, k the polynomial order, and N the number of nodes.

Proof sketch. We represent the polynomial filter bank, $\widetilde{\boldsymbol{X}} = \left[\boldsymbol{X},\ \boldsymbol{A}\boldsymbol{X},\ \ldots,\ \boldsymbol{A}^{k-1}\boldsymbol{X}\right]$ using a single *linear-attention* layer (i.e., without softmax). Since $\boldsymbol{X} \in \mathbb{R}^{N\times d}$ is low-rank when $d < N$, we augment the input with positional encodings: $\widehat{\boldsymbol{X}} = [\boldsymbol{X}, \boldsymbol{I}]$.

Why MHA needs k heads. As established in Section 3.1, in linear MHA, each head produces exactly one attention operator (matrix) $\boldsymbol{S}_h \in \mathbb{R}^{N\times N}$. To represent all k distinct powers $\boldsymbol{A}^0, \ldots, \boldsymbol{A}^{k-1}$ in parallel, one therefore needs k independently parameterized heads. The parameter-minimal exact MHA construction thus uses k heads.

Why IHA only needs $\lceil\sqrt{k}\rceil$ heads. Let $H = \lceil\sqrt{k}\rceil$ (and in the construction we set the number of pseudo-heads to $P = H$). IHA exploits the factorization: $\boldsymbol{A}^i = \boldsymbol{A}^{(h-1)H+(j-1)}$, $h, j \in \{1, \ldots, H\}$ so that $H^2 \geq k$ distinct powers can be generated via pairwise query–key interactions. Instead of assigning one head per power, IHA assigns heads to *blocks* of powers. Concretely, we choose H query and key matrices such that, $\boldsymbol{W}^{(h)}_{Q,\texttt{IHA}} = \left[\mathbf{0}_{d\times N};\ \boldsymbol{A}^{(h-1)H}\right]$ and $\boldsymbol{W}^{(j)}_{K,\texttt{IHA}} =$

$\left[\mathbf{0}_{d\times N};\ \left(\boldsymbol{A}^{j-1}\right)^\top\right]$, (where $\forall\, h, j \in \{1,\cdots,H\}$). When query head h interacts with key head j, the resulting (linear) attention matrix is $\boldsymbol{S}_{h,j} = \widehat{\boldsymbol{X}}\boldsymbol{W}^{(h)}_{Q,\texttt{IHA}}\left(\boldsymbol{W}^{(j)}_{K,\texttt{IHA}}\right)^\top\widehat{\boldsymbol{X}}^\top = \boldsymbol{A}^{(h-1)H+(j-1)}$. Thus, H query matrices and H key matrices generate $H^2 \geq k$ distinct polynomial powers through pairwise interaction. Pseudo-head mixing ensures that, within each head h, a *single* query attends to *all* H key/-value branches, producing the entire block $\left[\boldsymbol{A}^{(h-1)H}\boldsymbol{X},\ \boldsymbol{A}^{(h-1)H+1}\boldsymbol{X},\ \ldots,\ \boldsymbol{A}^{(h-1)H+(H-1)}\boldsymbol{X}\right]$ in one shot. Value matrices are chosen to route each $\boldsymbol{A}^{(h-1)H+(j-1)}\boldsymbol{X}$ into a distinct d-dimensional output block, and concatenating the H heads recovers $\left[\boldsymbol{X},\ \boldsymbol{A}\boldsymbol{X},\ \ldots,\ \boldsymbol{A}^{H^2-1}\boldsymbol{X}\right]$ with any extra (H^2-k) blocks treated as padding. Consequently, IHA represents the same polynomial filter bank using only $O(\sqrt{k})$ heads, reducing the dominant parameter cost from $\Theta(kN^2)$ (MHA) to $\Theta(\lceil\sqrt{k}\rceil N^2)$, up to lower-order routing terms (including the $4H^3$ pseudo-mixing/collapse parameters). For more details and the full proof, see Subsec. D.2.

4.3 Representing CPM-3 using IHA

Polynomial filters provide a controlled proxy for *multi-step retrieval*: they aggregate information over k-hop neighborhoods, but are insensitive to the order in which intermediate evidence is composed. To study order-sensitive composition, we introduce *Count Permutation Match-3* (CPM-3). Given scalar tokens $(x_1,\ldots,x_N)$, CPM-3 asks, for each position i, how many ordered pairs (j_1,j_2) satisfy a fixed three-token predicate involving (x_i,x_{j_1},x_{j_2}). This captures, in a controlled arithmetic setting, the key challenge in bAbI-style multi-fact QA [28]: combining supporting facts in the correct order rather than retrieving them independently. The predicate is chosen so that (j_1,j_2) and (j_2,j_1) are generally not equivalent. Thus, CPM-3 separates two requirements: making many candidate tokens jointly available for comparison, and evaluating an order-sensitive composition before summing valid pairs. We now define the predicate and target output. The input is a length-N sequence of natural numbers $(x_1,\ldots,x_N)$, with $N \leq N_{\max}$. For each position i, the desired output is $\mathrm{CPM}_i(3) = \left|\{(j_1,j_2)\in[N]^2 :\ \phi(x_i,x_{j_1},x_{j_2})=0\}\right|$, where the predicate is the (order-sensitive) modular expression $\phi(x_i,x_{j_1},x_{j_2}) := x_i + Gx_{j_1} + x_{j_2} \bmod M$, with modulus $M\in\mathbb{N}$ and coefficient $G > 2M$. The condition $G > 2M$ ensures the predicate is not permutation invariant: typically $\phi(x_i,x_{j_1},x_{j_2}) \neq \phi(x_i,x_{j_2},x_{j_1})$ when $x_{j_1}\neq x_{j_2}$. We next ask how many attention heads different mechanisms require to realize CPM-3 in a single layer; the result is stated in Thm. 4.

Theorem 4. (Count Permutation Match-3) Let $N_{\max}$ denote the maximum number of tokens that can be processed by the model in the worst case. There exists a one-layer IHA transformer that can represent CPM-3 using $\lceil\sqrt{N_{\max}}\rceil$ attention heads. The number of parameters required by this IHA construction is upper bounded by $37N^2_{\max}\sqrt{N_{\max}} + N^2_{\max}(N_{\max}-1) + N^2_{\max}$. In contrast, the best currently known construction based on (MHA) requires $N_{\max}$ attention heads, and its parameter count is lower bounded by $3N^3_{\max} + N^2_{\max}(N_{\max}-1) + N^2_{\max}$. Throughout, we assume the vocabulary size is at most on the order of the maximum sequence length, i.e., $|\mathcal{V}| = O(N_{\max})$.

Proof sketch. We sketch why CPM-3 can be implemented with $\lceil\sqrt{N_{\max}}\rceil$ IHA heads but (in known constructions) requires $N_{\max}$ MHA heads. The task is to output, for each position i, the count of ordered pairs (j_1,j_2) such that $x_i + Gx_{j_1} + x_{j_2} \equiv 0 \pmod M$, with $G > 2M$ ensuring order sensitivity. As in the polynomial-filter construction, we use positional encodings to make positions addressable: $\widehat{\boldsymbol{X}} = [\boldsymbol{X},\boldsymbol{I}] \in \mathbb{R}^{N_{\max}\times(N_{\max}+1)}$, $\boldsymbol{X}\in\mathbb{R}^{N_{\max}\times 1}$. Note that the CPM-3 output at position i depends on *all* ordered pairs (j_1,j_2), so a convenient one-layer strategy is to first use attention to build, at every i, a local "workspace" that contains all token values $\{x_j\}_{j=1}^{N_{\max}}$ in a fixed, known order. A downstream MLP can then (i) select any ordered pair of coordinates (j_1,j_2), (ii) form $x_i + Gx_{j_1} + x_{j_2}$, (iii) test the modulo constraint, and sum indicators. We now sketch why MHA needs $N_{\max}$ heads, while IHA needs only $\lceil\sqrt{N_{\max}}\rceil$.

Why MHA needs $N_{\max}$ heads. With positional encodings $\hat{\boldsymbol{X}} = [\boldsymbol{X},\boldsymbol{I}]$ and hard attention (softmax temperature 0), each MHA head can implement one cyclic shift of the sequence. Let $\boldsymbol{P}\in\mathbb{R}^{N_{\max}\times N_{\max}}$ be the cyclic permutation matrix. For

$h \in \{1, \ldots, N_{\max}\}$ set $\boldsymbol{W}_{V,\texttt{MHA}}^{(h)} = \begin{bmatrix} 1 & \boldsymbol{0}_{N_{\max}\times 1}^\top \end{bmatrix}^\top$, $\boldsymbol{W}_{Q,\texttt{MHA}}^{(h)} = \begin{bmatrix} \boldsymbol{0}_{1\times N_{\max}}; \boldsymbol{P}^{h-1} \end{bmatrix}$, and $\boldsymbol{W}_{K,\texttt{MHA}}^{(h)} = [\boldsymbol{0}_{1\times N_{\max}}; \boldsymbol{I}_{N_{\max}\times N_{\max}}]$ and so the head produces $\boldsymbol{S}_h = \boldsymbol{P}^{h-1}$ and outputs $\boldsymbol{P}^{h-1}\boldsymbol{X}$. Concatenating all heads yields $\begin{bmatrix} \boldsymbol{X}, \boldsymbol{P}\boldsymbol{X}, \ldots, \boldsymbol{P}^{N_{\max}-1}\boldsymbol{X} \end{bmatrix}$, which places all $N_{\max}$ symbols into each position's workspace. Since each head contributes only one shift $\boldsymbol{P}^t$, producing all $N_{\max}$ shifts in one layer requires $N_{\max}$ heads, giving attention-parameter scaling $\Omega(N_{\max}^3)$.

Why IHA only needs $\lceil\sqrt{N_{\max}}\rceil$ heads. Let $H = \lceil\sqrt{N_{\max}}\rceil$ and set $P = H$. IHA factors each shift index $t \in \{0, \ldots, N_{\max} - 1\}$ as $t = (h-1)H + (j-1)$, $h, j \in \{1, \ldots, H\}$, so $\boldsymbol{P}^t = \boldsymbol{P}^{(h-1)H}\boldsymbol{P}^{j-1}$. Define query and key matrices as: $\boldsymbol{W}_{Q,\texttt{IHA}}^{(h)} = \begin{bmatrix} \boldsymbol{0}_{1\times N_{\max}}; \boldsymbol{P}^{(h-1)H} \end{bmatrix}$, $\boldsymbol{W}_{K,\texttt{IHA}}^{(j)} = \begin{bmatrix} \boldsymbol{0}_{1\times N_{\max}}; \left(\boldsymbol{P}^{j-1}\right)^\top \end{bmatrix}$, so the (h, j) (attention) realizes $\boldsymbol{S}_{h,j} = \boldsymbol{P}^{(h-1)H+(j-1)}$. The crucial difference from MHA is that pseudo-head mixing lets a *single* query head h attend to *all* $j \in \{1, \ldots, H\}$ key/value heads, producing in one head the block $\begin{bmatrix} \boldsymbol{P}^{(h-1)H}\boldsymbol{X}, \boldsymbol{P}^{(h-1)H+1}\boldsymbol{X}, \ldots, \boldsymbol{P}^{(h-1)H+(H-1)}\boldsymbol{X} \end{bmatrix}$, with value projections routing each shift into a distinct coordinate block. Concatenating over $h \in \{1, \ldots, H\}$ yields all $\{\boldsymbol{P}^t\boldsymbol{X}\}_{t=0}^{N_{\max}-1}$ at each position. i.e., the same output as above, but using only $H = \lceil\sqrt{N_{\max}}\rceil$ heads. The downstream MLP is then identical to that of the MHA construction. The key difference is that IHA can realize many distinct attention patterns per head: with P pseudo-queries and pseudo-keys per head, each head can implement up to P^2 different attention maps. Taking $P = H = \lceil\sqrt{N_{\max}}\rceil$ provides enough distinct patterns to cover the $N_{\max}$ required shifts while reducing the head count from $N_{\max}$ to $\lceil\sqrt{N_{\max}}\rceil$. Consequently, the best-known one-layer MHA construction incurs $\Theta(N_{\max}^3)$ attention cost, whereas the IHA construction achieves $\Theta(N_{\max}^2\sqrt{N_{\max}})$. For full details, see Subsec. D.3.

5 Experiments

We evaluate Interleaved Head Attention (IHA) in large-scale language model training, asking whether it improves (i) long-context retrieval and length generalization beyond the pretraining window, and (ii) math and code reasoning before and after supervised fine-tuning. To isolate architectural effects, we fix the backbone, optimizer, data, and training budget across variants, comparing IHA against strong attention baselines. This tests whether the expressivity results and controlled-proxy separations from the preceding sections translate into practical gains in downstream evaluation.

5.1 Experimental Setup

Model architecture and training. All experiments use a 2.4B-parameter decoder-only Transformer with hidden size 2560, 26 layers, and $H = 20$ attention heads (head dimension 128). We use a $4\times$ FFN expansion (FFN size 10,240), vocabulary size 128,256 (Llama 3 tokenizer; [9]), and RoPE positional encoding [26] with $\theta = 500{,}000$. Pretraining context length is 8,192 tokens. All models are trained for 240,000 steps (240B tokens) with identical hyperparameters: peak learning rate 8×10^{-4} with 1,000-step warmup and cosine decay [18] to 8×10^{-6}, AdamW [19] (β_1=0.9, β_2=0.95, weight decay 0.1), gradient clipping 1.0, and BF16 mixed precision [20]. Training uses FSDP [30] over 128 H200 GPUs.

Baselines. We compare five attention mechanisms. (1) **Global Attention** [27] is standard multi-head self-attention, where every layer attends to the full sequence. (2) **Global+Local** [27] is a hybrid schedule that alternates local sliding-window attention (window size 512) with periodic global-attention layers in a 4:1 ratio. (3) **Talking Heads** [25] augments multi-head attention by learning to mix information across heads both before and after the softmax, enabling richer head-to-head interactions. (4) **Diff Transformer** [29] defines attention as the difference of two softmax attention maps, which can sharpen or suppress patterns via contrastive weighting. and (5) **IHA (Ours)**, interleaved-head attention with pseudo-heads. Let N be the sequence length, d the per-head dimension, H the number of heads, and P the number of pseudo-heads per head. Since interleaving expands the effective sequence length from N to NP, global IHA has per-head complexity $\mathcal{O}(P^2N^2d)$, i.e., a factor-P^2 over global MHA; we therefore FLOP-match all comparisons. We use a hybrid local–global schedule (four sliding-window IHA layers with $W := N/(2P^2)$ followed by one global layer) so the average compute cost matches the global-attention baseline; see App. E for details. In our experiments, $P = H = 20$.

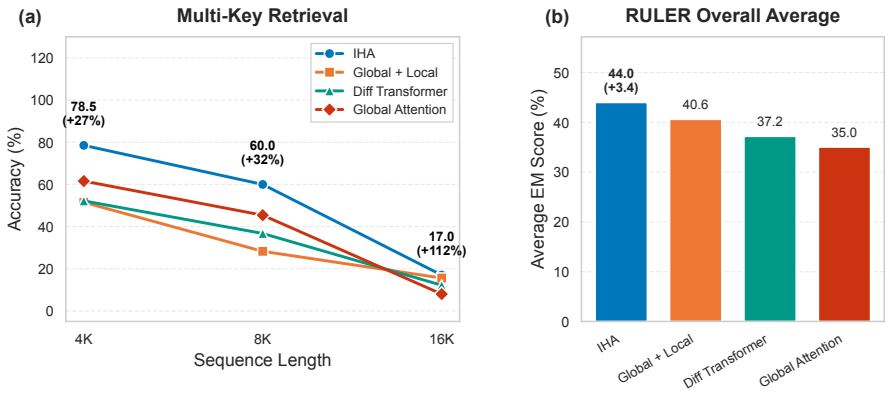

Figure 2: **RULER long-context results after 64k fine-tuning.** (a) Multi-Key Retrieval accuracy at 4k/8k/16k context lengths. (b) Average EM across the full RULER suite. IHA performs best overall.

Table 1: Pre-trained model evaluation (5-shot). IHA achieves the best overall reasoning performance. Δ is relative to **Global Attention** (green/red = gain/drop), and **Avg. Rank**↓ is the mean rank across metrics (lower is better).

Model	GSM8K P@1	Δ	GSM8K Maj@5	Δ	MATH-500 P@1	Δ	MBPP P@1	Δ	HumanEval P@1	Δ	Avg. Rank↓
IHA (Ours)	**8.34%**	+2.73	**8.42%**	+2.81	**3.54%**	+0.66	24.5%	+1.1	17.1%	–0.1	**1.4**
Global Attention	5.61%	–	5.61%	–	2.88%	–	23.4%	–	**17.2%**	–	2.9
Global+Local	6.82%	+1.21	6.90%	+1.29	2.26%	–0.62	23.6%	+0.2	16.0%	–1.2	2.9
Talking Heads	5.46%	–0.15	5.38%	–0.23	–	–	23.8%	+0.4	16.0%	–1.2	4.0
Diff Transformer	5.46%	–0.15	5.61%	–	–	–	**25.0%**	+1.6	15.4%	–1.8	3.5

Benchmarks. For long-context evaluation we use RULER [15]. For reasoning and coding we evaluate GSM8K [6], MATH-500 [14], MBPP [2], and HumanEval [3].

5.2 Long Context Evaluation

For long-context evaluation, we fine-tuned all models at 64k (beyond the pretraining window) and evaluated on RULER (Fig. 2). IHA is consistently stronger on retrieval: on Multi-Key Retrieval it improves over Global Attention by **+27%** (4k), **+32%** (8k), and **+112%** (16k), in relative performance. Across the full RULER suite, IHA achieves the best average EM (exact match) (**44.0%**), outperforming Global+Local (**40.6%**), Diff Transformer (**37.2%**), and Global Attention (**35.0%**).

5.3 Reasoning Evaluation

Pretraining. We evaluate *pre-trained* models in a 5-shot setting to probe reasoning ability prior to supervised instruction tuning (Tab. 1). **IHA (Ours)** consistently improves over **Global Attention** on the core reasoning benchmarks: on GSM8K it achieves the best scores (**8.34%** EM and **8.42%** Maj@5; **+2.73/+2.81**), and it also leads on MATH-500 EM with **3.54%** (**+0.66**). Coding results are mixed, with a modest gain on the MBPP benchmark to 24.5% (second best) while HumanEval is near parity, but IHA is the most consistent method overall, achieving the best mean rank across reported metrics (**Avg. Rank**↓= 1.4). Overall, these results indicate that IHA's added expressivity translates into stronger reasoning performance even before downstream fine-tuning.

Supervised fine-tuning. We fine-tuned all variants on OpenThoughts [12] (8B tokens) and evaluated with temperature 0.6 using 16 generations (Tab. 2). **IHA (Ours)** achieves the best overall performance, leading on all reasoning metrics (e.g., **54.2%** GSM8K Maj@16, **+5.8** over **Global Attention**; **18.4%** MATH-500 Maj@16, **+2.8**) . On coding (MBPP), **Talking Heads** is best (15.9% P@1, 43.1% P@10), while IHA is *second best* on both MBPP metrics (15.5% P@1, 41.6% P@10) suggesting that while IHA's expressivity excels at logical state tracking for math, head mixing is well-suited for function-level code generation. Overall, IHA remains consistently strong across tasks after SFT, attaining the best **Avg. Rank**↓ (**1.5**).

Table 2: SFT evaluation after fine-tuning on OpenThoughts. IHA achieves the best overall performance, with larger gains than at pre-training. Δ is relative to **Global Attention** (green/red = gain/drop), and **Avg. Rank**$\downarrow$ is the mean rank across metrics (lower is better).

Model	GSM8K P@1	Δ	GSM8K Maj@16	Δ	MATH-500 P@1	Δ	MATH-500 Maj@16	Δ	MBPP P@1	Δ	MBPP P@10	Δ	Avg. Rank$\downarrow$
IHA (Ours)	**34.3%**	+4.8	**54.2%**	+5.8	**10.0%**	+1.2	**18.4%**	+2.8	15.5%	+0.8	41.6%	+0.4	**1.5**
Global Attention	29.5%	–	48.4%	–	8.8%	–	15.6%	–	14.7%	–	41.2%	–	3.8
Global+Local	26.5%	–3.0	46.9%	–1.5	7.6%	–1.2	15.0%	–0.6	15.0%	+0.3	41.9%	+0.7	4.3
Talking Heads	29.3%	–0.2	49.4%	+1.0	7.8%	–1.0	18.2%	+2.6	**15.9%**	+1.2	**43.1%**	+1.9	2.5
Diff Transformer	31.6%	+2.1	53.5%	+5.1	9.0%	+0.2	18.0%	+2.4	15.3%	+0.6	39.2%	–2.0	2.8

Efficiency and additional ablations. IHA adds only modest overhead under FLOP-matched settings, with smaller overhead at longer sequence lengths. On the 2.4B model, IHA is 1.07–$1.10\times$ slower than MHA at $N = 8192$ and achieves 0.91–$0.94\times$ its throughput. At $N = 16384$, IHA is only 1.01–$1.02\times$ slower, with throughput within 1–2% of MHA. During decoding, IHA does not increase KV-cache size because it caches the original per-head keys and values, not pseudo-head mixtures. See App. E for more details. Since sweeping P and depth is expensive at the 2.4B scale, we study them on controlled synthetic tasks, where IHA shows the clearest gains on aggregation and relation-composition problems in App. F. We also provide additional experiments on model scale and pseudo-head count in Subsec. G.2, wider-MHA controls in Subsec. G.3, layer-placement ablations in App. H, and analyses of learned head-mixing tensors in App. I.

6 Conclusion and Limitations

We introduced *Interleaved Head Attention (IHA)*, which overcomes MHA's linear scaling by learning P pseudo-queries, pseudo-keys, and pseudo-values per head as linear combinations of the original heads. Interactions between pseudo-queries and pseudo-keys induce up to P^2 attention patterns per head. Our theory shows improved parameter efficiency on Polynomial Filters and on the order-sensitive CPM-3 task. Under FLOP-matched training, IHA improves Multi-Key retrieval on RULER by 10–20% (4k–16k) and, after OpenThoughts fine-tuning, improves reasoning on GSM8K by 5.8% and MATH-500 by 2.8% (majority vote) over full attention. **Limitations.** Global IHA can increase attention cost (scaling as $\mathcal{O}(P^2N^2)$), which we mitigate with a sliding-window schedule; future work includes adaptive pseudo-head allocation and extensions to encoder–decoder and vision architectures.

References

[1] J. Ainslie, J. Lee-Thorp, M. de Jong, Y. Zemlyanskiy, F. Lebrón, and S. Sanghai. GQA: Training generalized multi-query transformer models from multi-head checkpoints. *arXiv preprint arXiv:2305.13245*, 2023.

[2] J. Austin, A. Odena, M. Nye, M. Bosma, H. Michalewski, D. Dohan, E. Jiang, C. Cai, M. Terry, Q. Le, and C. Sutton. Program synthesis with large language models, 2021.

[3] M. Chen, J. Tworek, H. Jun, Q. Yuan, H. P. de Oliveira Pinto, J. Kaplan, H. Edwards, Y. Burda, N. Joseph, G. Brockman, A. Ray, R. Puri, G. Krueger, M. Petrov, H. Khlaaf, G. Sastry, P. Mishkin, B. Chan, S. Gray, N. Ryder, M. Pavlov, A. Power, L. Kaiser, M. Bavarian, C. Winter, P. Tillet, F. P. Such, D. Cummings, M. Plappert, F. Chantzis, E. Barnes, A. Herbert-Voss, W. H. Guss, A. Nichol, A. Paino, N. Tezak, J. Tang, I. Babuschkin, S. Balaji, S. Jain, W. Saunders, C. Hesse, A. N. Carr, J. Leike, J. Achiam, V. Misra, E. Morikawa, A. Radford, M. Knight, M. Brundage, M. Murati, K. Mayer, P. Welinder, B. McGrew, D. Amodei, S. McCandlish, I. Sutskever, and W. Zaremba. Evaluating large language models trained on code, 2021.

[4] E. Chien, J. Peng, P. Li, and O. Milenkovic. GPR-GNN: Generalized pagerank graph neural network. In *International Conference on Learning Representations (ICLR)*, 2021.

[5] J. Clift, D. Doryn, D. Murfet, and J. Wallbridge. Logic and the 2-simplicial transformer. *arXiv preprint arXiv:1909.00668*, 2019.

[6] K. Cobbe, V. Kosaraju, M. Bavarian, M. Chen, H. Jun, L. Kaiser, M. Plappert, J. Tworek, J. Hilton, R. Nakano, C. Hesse, and J. Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

[7] T. Dao, D. Y. Fu, S. Ermon, A. Rudra, and C. Ré. FlashAttention: Fast and memory-efficient exact attention with IO-awareness. In *Advances in Neural Information Processing Systems*, 2022.

[8] M. Defferrard, X. Bresson, and P. Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in Neural Information Processing Systems*, volume 29, 2016.

[9] A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Yang, A. Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[10] C. Ekbote, A. Deshpande, A. Iyer, S. SELLAMANICKAM, and R. Bairi. Figure: Simple and efficient unsupervised node representations with filter augmentations. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 35403–35425. Curran Associates, Inc., 2023.

[11] O. Golovneva, T. Wang, J. Weston, and S. Sukhbaatar. Multi-token attention. *arXiv preprint arXiv:2504.00927*, 2025.

[12] E. Guha, R. Marten, S. Keh, N. Raoof, G. Smyrnis, H. Bansal, M. Nezhurina, J. Mercat, T. Vu, Z. Sprague, et al. OpenThoughts: Data recipes for reasoning models, 2025.

[13] A. Gupta, H. Mehta, and J. Berant. Simplifying and understanding state space models with diagonal linear rnns. *arXiv preprint arXiv:2212.00768*, 2022.

[14] D. Hendrycks, C. Burns, S. Kadavath, A. Arora, S. Basart, E. Tang, D. Song, and J. Steinhardt. Measuring mathematical problem solving with the MATH dataset. *NeurIPS*, 2021.

[15] C.-P. Hsieh, S. Sun, S. Kriman, S. Acharya, D. Rekesh, F. Jia, Y. Zhang, and B. Ginsburg. RULER: What's the real context size of your long-context language models? *arXiv preprint arXiv:2404.06654*, 2024.

[16] A. Kozachinskiy, F. Urrutia, H. Jimenez, T. Steifer, G. Pizarro, M. Fuentes, F. Meza, C. B. Calderon, and C. Rojas. Strassen attention, split vc dimension and compositionality in transformers. *arXiv preprint arXiv:2501.19215*, 2025.

[17] V. Lingam, C. Ekbote, M. Sharma, R. Ragesh, A. Iyer, and S. Sellamanickam. A piece-wise polynomial filtering approach for graph neural networks, 2021.

[18] I. Loshchilov and F. Hutter. SGDR: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations*, 2017.

[19] I. Loshchilov and F. Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

[20] P. Micikevicius, S. Narang, J. Alben, G. Diamos, E. Elsen, D. Garcia, B. Ginsburg, M. Houston, O. Kuchaiev, G. Venkatesh, and H. Wu. Mixed precision training. In *International Conference on Learning Representations*, 2018.

[21] A. Roy et al. Fast and simplex: 2-simplicial attention in triton. *arXiv preprint arXiv:2507.02754*, 2025.

[22] C. Sanford, D. Hsu, and M. Telgarsky. Representational strengths and limitations of transformers. *Advances in Neural Information Processing Systems*, 36, 2023.

[23] N. Saunshi, N. Dikkala, Z. Li, S. Kumar, and S. J. Reddi. Reasoning with latent thoughts: On the power of looped transformers. In *International Conference on Learning Representations (ICLR)*, 2025.

[24] N. Shazeer. Fast transformer decoding: One write-head is all you need. *arXiv preprint arXiv:1911.02150*, 2019.

[25] N. Shazeer, Z. Lan, Y. Cheng, N. Ding, and L. Hou. Talking-heads attention. *arXiv preprint arXiv:2003.02436*, 2020.

[26] J. Su, Y. Lu, S. Pan, A. Murtadha, B. Wen, and Y. Liu. RoFormer: Enhanced transformer with rotary position embedding, 2021.

[27] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

[28] J. Weston, A. Bordes, S. Chopra, A. M. Rush, B. van Merriënboer, A. Joulin, and T. Mikolov. Towards AI-complete question answering: A set of prerequisite toy tasks. In *International Conference on Learning Representations (ICLR)*, 2016.

[29] T. Ye, L. Li, G. Huang, et al. Differential transformer. *arXiv preprint arXiv:2410.05258*, 2024.

[30] Y. Zhao, A. Gu, R. Varma, L. Luo, C.-C. Huang, M. Xu, L. Wright, H. Shojanazeri, M. Ott, S. Shleifer, A. Desmaison, C. Balioglu, P. Damania, B. Nguyen, G. Chauhan, Y. Hao, A. Mathews, and S. Li. PyTorch FSDP: Experiences on scaling fully sharded data parallel. In *Proceedings of the VLDB Endowment*, volume 16, pages 3848–3860, 2023.

[31] Z. Zhou et al. Knocking-heads attention. *arXiv preprint arXiv:2510.23052*, 2025.

Appendix

Contents

1 Introduction **1**

2 Related Work **2**

3 Background **3**

3.1 Multi-head Attention and Polynomial Filters 3

4 Interleaved-Head Attention (IHA) **4**

4.1 IHA Strictly Generalizes MHA 5

4.2 Representing Polynomial Filters using IHA 6

4.3 Representing CPM-3 using IHA 7

5 Experiments **8**

5.1 Experimental Setup 8

5.2 Long Context Evaluation 9

5.3 Reasoning Evaluation 9

6 Conclusion and Limitations **10**

A Appendix Organization **15**

B Extended Related Work **15**

C An Equivalent Definition of IHA for Analysis **16**

D Theoretical Properties of IHA **16**

D.1 IHA Superset Property 17

D.2 Representing Polynomial Filters 18

D.3 Representing Count Permutation Match 3 (CPM-3) 23

E Computational Efficiency of IHA **29**

E.1 Global IHA complexity 29

E.2 FLOP-matched local-global schedule 30

E.3 Training throughput 30

E.4 KV-cache behavior during decoding 30

E.5 IHA-CUDA: Hopper Kernels for Interleaved Head Attention 31

F Effect of the Number of Pseudo-Heads **32**
F.1 Boolean relation composition 32
F.2 Atomic sequence-manipulation tasks 33

G Additional controlled pretraining experiments **34**
G.1 Experimental setup and evaluation 34
G.2 Model scale and pseudo-head count 35
G.3 Parameter overhead and wider-MHA controls 35

H Layer-placement ablation **36**

I Analysis of learned head-mixing tensors **36**

J Broader Impact **38**

A Appendix Organization

Appendix organization. The appendix is organized as follows. Appendix **??** provides an architecture figure illustrating the interleaved-head construction. Appendix B gives additional discussion of related work. Appendix C presents an equivalent formulation of IHA used for the theoretical analysis. Appendix D contains the main proofs, including the IHA superset property, the polynomial-filter construction, and the CPM-3 construction. Appendix E details the computational cost, FLOP-matched local–global schedule, training throughput, KV-cache behavior and implementation details. Finally, Appendix F studies the effect of varying the number of pseudo-heads on controlled synthetic tasks.

B Extended Related Work

Hardness results for compositional reasoning. Recent theory has begun to formalize when standard multi-head attention (MHA) is an inefficient mechanism for compositional multi-step reasoning. [16] prove hardness results for binary relation composition and function composition, and [22] show that Match-3 requires $\Theta(N^3)$ parameters under standard attention-based constructions. These settings share two structural requirements: global aggregation of evidence across many positions, and composition of intermediate relational signals prior to producing an output. Related behavior also appears in multi-hop QA benchmarks such as bAbI [28], which require combining information across multiple hops. We study these challenges through two controlled proxies that separate these requirements. Polynomial filters provide a standard k-hop aggregation primitive from spectral GNNs [8, 4, 17, 10], and CPM-3 isolates order-sensitive composition and counting. In both settings, we show that IHA realizes the relevant primitives with quadratic improvements in head and parameter efficiency relative to comparable MHA constructions.

Higher-order and multi-token attention. Several works enrich token interactions by going beyond pairwise query-key attention. The 2-Simplicial Transformer [5] generalizes attention to trilinear interactions, and [21] provide an efficient Triton implementation. Strassen-style attention constructions use fast matrix multiplication ideas to accelerate particular compositional patterns [16]. Multi-Token Attention [11] introduces mechanisms that mix information across small token groups, for example via local mixing over attention weights. These approaches typically modify the attention operator or introduce specialized higher-order structure. IHA is complementary. It preserves the standard attention operator and induces effective higher-order behavior by learning cross-head mixing of Q, K, and V into pseudo-heads, yielding a broad family of quadratic interaction patterns while remaining compatible with optimized attention kernels.

Iterative computation via depth or recurrence. Another approach to multi-step reasoning is to increase the number of sequential transformations, either via depth or recurrence. Looped transformers [23] show that iterating a k-layer block for L loops can match a kL-layer model, and can support chain-of-thought-like behavior through repeated refinement. These methods increase computation across iterations. Our approach is complementary. IHA increases within-layer interaction capacity by constructing P pseudo-heads per head, typically with $P = H$, using learned linear combinations of the original queries, keys, and values. Interactions between pseudo queries and pseudo keys induce up to P^2 attention patterns per head within a single attention computation, without adding sequential depth.

Efficient and cross-head attention variants. A large body of work improves attention efficiency and head utilization. Multi-Query Attention (MQA) [24] and Grouped Query Attention (GQA) [1] reduce inference cost by sharing key and value projections across heads. Other methods explicitly couple heads by mixing attention logits or weights, including Talking-Heads [25] and Knocking-Heads [31], or by shaping attention maps, as in Differential Attention [29]. In contrast, IHA enables cross-head interaction within attention by mixing Q, K, and V representations into pseudo-heads via learned linear combinations. This induces quadratic interaction structure while preserving the standard attention computation, and it remains compatible with efficient kernels such as FlashAttention [7].

C An Equivalent Definition of IHA for Analysis

As discussed Sec. 4, **IHA** constructs P *pseudo-heads* as learned linear combinations of the H original heads. For ease of analysis, we present an equivalent *non-interleaved* formulation in which attention is computed over an expanded sequence of length PN (obtained by stacking the P pseudo-head variants per token), and we omit the output projection/collapse used in the implementation. This simplification is used only for proofs: since our theoretical results do not rely on positional encodings (e.g., RoPE), interleaving is unnecessary in the analysis, and removing the projection yields a cleaner algebraic description without affecting the core attention mechanism.

Definition 3.
$\boldsymbol{X} \in \mathbb{R}^{N\times D}$, heads H, pseudos P, per-head dim d
$\boldsymbol{W}_Q^{(m)}, \boldsymbol{W}_K^{(m)}, \boldsymbol{W}_V^{(m)} \in \mathbb{R}^{D\times d}$ for $m = 1, \ldots, H$
$\alpha^Q, \alpha^K, \alpha^V \in \mathbb{R}^{H\times H\times P}$ and $\boldsymbol{R} \in \mathbb{R}^{H\times HP}$
Pseudo-head mixing across heads
for $h = 1$ to H **do**
  for $j = 1$ to P **do**
    $\widetilde{\boldsymbol{Q}}_{h,j} \coloneqq \sum_{m=1}^{H} \alpha^Q_{m,h,j}\, \boldsymbol{X}\boldsymbol{W}_Q^{(m)} \in \mathbb{R}^{N\times d}$
    $\widetilde{\boldsymbol{K}}_{h,j} \coloneqq \sum_{m=1}^{H} \alpha^K_{m,h,j}\, \boldsymbol{X}\boldsymbol{W}_K^{(m)} \in \mathbb{R}^{N\times d}$
    $\widetilde{\boldsymbol{V}}_{h,j} \coloneqq \sum_{m=1}^{H} \alpha^V_{m,h,j}\, \boldsymbol{X}\boldsymbol{W}_V^{(m)} \in \mathbb{R}^{N\times d}$
  end for
end for
Pseudo-major stacking (row-wise concatenation to length PN)
for $h = 1$ to H **do**
  $\overline{\boldsymbol{Q}}_h \coloneqq \left[\widetilde{\boldsymbol{Q}}_{h,1}^\top; \ldots; \widetilde{\boldsymbol{Q}}_{h,P}^\top\right]^\top \in \mathbb{R}^{PN\times d}$
  $\overline{\boldsymbol{K}}_h \coloneqq \left[\widetilde{\boldsymbol{K}}_{h,1}^\top; \ldots; \widetilde{\boldsymbol{K}}_{h,P}^\top\right]^\top \in \mathbb{R}^{PN\times d}$
  $\overline{\boldsymbol{V}}_h \coloneqq \left[\widetilde{\boldsymbol{V}}_{h,1}^\top; \ldots; \widetilde{\boldsymbol{V}}_{h,P}^\top\right]^\top \in \mathbb{R}^{PN\times d}$
end for
Attention (per head)
for $h = 1$ to H **do**
  $\boldsymbol{S}_h \coloneqq \frac{1}{\sqrt{d}}\, \overline{\boldsymbol{Q}}_h\, \overline{\boldsymbol{K}}_h^\top \in \mathbb{R}^{PN\times PN}$
  $\overline{\boldsymbol{P}}_h \coloneqq \mathrm{softmax}(\boldsymbol{S}_h)\, \overline{\boldsymbol{V}}_h \in \mathbb{R}^{PN\times d}$
end for
Unstack and collapse $HP \to H$
for $h = 1$ to H **do**
  for $t = 1$ to N **do**
    $\boldsymbol{O}_h[t,:] \coloneqq \sum_{h'=1}^{H} \sum_{j=1}^{P} \boldsymbol{R}_{h,(h'-1)P+j}\, \boldsymbol{P}_{h',j}[t,:] \in \mathbb{R}^d$
  end for
end for
Concatenate heads
$\widetilde{\boldsymbol{X}} \coloneqq [\boldsymbol{O}_1, \boldsymbol{O}_2, \ldots, \boldsymbol{O}_H] \in \mathbb{R}^{N\times D}$

D Theoretical Properties of IHA

In this section, we establish theoretical properties of IHA that highlight its advantages over MHA. We present and prove several key representational results below.

D.1 IHA Superset Property

Theorem 5. (IHA Superset Property; Parameter-Aware): Fix a sequence length n and number of heads h. Let $\mathcal{M}$ denote the set of all single-layer h-head multi-head attention (MHA) modules with query, key, value matrices and inputs bounded in the Frobenius norm, requiring Q parameters in total.
Let $\mathcal{P}_p$ denote the corresponding set of h-head IHA modules as in Alg. 1 with p pseudos per head, whose base query, key, and value weight matrices have the same dimensions as those in MHA (hence also contributing Q parameters), but which additionally introduce

$$\alpha^Q, \alpha^K, \alpha^V \in \mathbb{R}^{h\times h\times p} \quad \text{and} \quad R^\ell \in \mathbb{R}^{h\times hp},$$

i.e., an additional $3h^2p + h^2p = 4h^2p$ parameters, for a total of $Q + 4h^2p$ parameters.
Then for every $p \geq 1$,

$$\mathcal{M} \subseteq \mathcal{P}_p,$$

and for every $p \geq 2$ the inclusion is strict:

$$\mathcal{M} \subsetneq \mathcal{P}_p.$$

Proof. **(Inclusion $\mathcal{M} \subseteq \mathcal{P}_p$).** Fix any **MHA** instance with weights $\{W_Q^{(m)}, W_K^{(m)}, W_V^{(m)}\}_{m=1}^h$. We construct a IHA instance (with any chosen $p \geq 1$) that realizes the same function by selecting parameters that ignore all but one pseudo-channel.

Specifically, set for all $m, i \in \{1, \dots, h\}$ and $\forall\, j \in \{1, \cdots, p\}$,

$$\alpha^Q_{m,i,j} = \mathbf{1}_{(m=i)}, \qquad \alpha^K_{m,i,j} = \mathbf{1}_{(m=i)}, \qquad \alpha^V_{m,i,j} = \mathbf{1}_{(m=i)},$$

Further, choosing $R^\ell \in \mathbb{R}^{h\times hp}$ so that it selects only the $(i, 1)$ pseudo-block:

$$R^\ell_{i,\,(i'-1)p+j} = \begin{cases} 1 & \text{if } i' = i \text{ and } j = 1, \\ 0 & \text{otherwise.} \end{cases}$$

Then for each head i we have $\widetilde{\boldsymbol{Q}}_{i,j} = \boldsymbol{X}\boldsymbol{W}_Q^{(i)}$, $\widetilde{\boldsymbol{K}}_{i,j} = \boldsymbol{X}\boldsymbol{W}_K^{(i)}$, $\widetilde{\boldsymbol{V}}_{i,j} = \boldsymbol{X}\boldsymbol{W}_V^{(i)}$ $\forall j \in \{1, \cdots, p\}$. Consequently, the stacked attention produces an output whose only nonzero contribution is exactly the usual MHA head output, and the collapse via R^ℓ returns O_i equal to that MHA head output. Concatenating heads yields exactly the original MHA module. Hence $\mathcal{M} \subseteq \mathcal{P}_p$.

Strictness for $p \geq 2$ (works for any $n \geq 2$ and any d and any $h \geq 1$). It suffices to exhibit one h-head IHA/PseudoIHA configuration (with $p \geq 2$) that cannot be represented by any h-head MHA configuration.

MHA is linear on repeated-token inputs. Fix $n \geq 2$ and consider inputs with repeated tokens

$$\boldsymbol{X} = \mathbf{1}_n \boldsymbol{x}^\top \in \mathbb{R}^{n\times d},$$

where $\mathbf{1}_n \in \mathbb{R}^n$ is the all-ones vector and $\boldsymbol{x} \in \mathbb{R}^d$. For any MHA head m, define

$$\boldsymbol{q} := \boldsymbol{x}^\top \boldsymbol{W}_Q^{(m)}, \qquad \boldsymbol{k} := \boldsymbol{x}^\top \boldsymbol{W}_K^{(m)}, \qquad \boldsymbol{v} := \boldsymbol{x}^\top \boldsymbol{W}_V^{(m)}.$$

Then every token position has identical query/key/value, so the attention score matrix is constant:

$$\boldsymbol{S}^{(m)} = (\boldsymbol{X}\boldsymbol{W}_Q^m)(\boldsymbol{X}\boldsymbol{W}_K^m)^\top = (\mathbf{1}_n \boldsymbol{q})(\mathbf{1}_n \boldsymbol{k})^\top = (\langle \boldsymbol{q}, \boldsymbol{k}\rangle)\, \mathbf{1}_n \mathbf{1}_n^\top.$$

Since each row of $S^{(m)}$ is the same, hence the row-wise softmax is uniform:

$$\sigma\big(S^{(m)}\big) = \frac{1}{n} \mathbf{1}_n \mathbf{1}_n^\top.$$

Therefore the head output equals

$$\mathrm{Att}(\boldsymbol{X}\boldsymbol{W}_Q^{(m)}, \boldsymbol{X}\boldsymbol{W}_K^{(m)}, \boldsymbol{X}\boldsymbol{W}_V^{(m)}) = \frac{1}{n} \mathbf{1}_n \mathbf{1}_n^\top (\mathbf{1}_n v) = \mathbf{1}_n \boldsymbol{v} = \mathbf{1}_n \boldsymbol{x}^\top \boldsymbol{W}_V^{(m)},$$

which is linear in $\boldsymbol{x}$. Concatenating heads and applying any fixed output projection preserves linearity in each head. Hence *every* h-head **MHA** module is linear on the repeated-token subspace $\{\mathbf{1}_n \boldsymbol{x}^\top : \boldsymbol{x} \in \mathbb{R}^d\}$.

IHA yields a non-linear mapping of the input when $p \geq 2$. It is enough to consider the case $p = 2$, since for any $p > 2$ we can deactivate the additional pseudo channels by setting their mixing coefficients to zero. Concretely, we could impose

$$\alpha^Q_{m,i,j} = \alpha^K_{m,i,j} = \alpha^V_{m,i,j} = 0 \qquad \forall\, m, i \in \{1, \ldots, h\},\ \forall\, j > 2,$$

so that only the first two pseudo heads contribute (with the right reduction matrix R). We now construct a **IHA** layer whose output is *nonlinear* even on repeated-token inputs of the form $\boldsymbol{X} = \mathbf{1}_n \boldsymbol{x}^\top$.

Towards this, we focus on only one head h with two psudo tokens / heads. We let $\alpha^Q_{m,i,1} = \alpha^K_{m,i,1} = \alpha^V_{m,i,1} = \alpha^V_{m,i,2} = \mathbf{1}_{(m=i)}$ and $\alpha^Q_{m,i,2} = \alpha^K_{m,i,2} = -\mathbf{1}_{(m=i)}$. Hence, for any arbitrary head h, we obtain,

$$\overline{\boldsymbol{Q}}_h = \begin{bmatrix} \widetilde{\boldsymbol{Q}}_{h,1} \\ \widetilde{\boldsymbol{Q}}_{h,2} \end{bmatrix} \qquad \overline{\boldsymbol{K}}_h = \begin{bmatrix} \widetilde{\boldsymbol{K}}_{h,1} \\ \widetilde{\boldsymbol{K}}_{h,2} \end{bmatrix} \qquad \overline{\boldsymbol{V}}_h = \begin{bmatrix} \widetilde{\boldsymbol{V}}_{h,1} \\ \widetilde{\boldsymbol{V}}_{h,2} \end{bmatrix}$$

Hence, on computing, we obtain:

$$\overline{\boldsymbol{Q}}_h = \begin{bmatrix} \boldsymbol{Q}_h \\ -\boldsymbol{Q}_h \end{bmatrix} \qquad \overline{\boldsymbol{K}}_h = \begin{bmatrix} \boldsymbol{K}_h \\ -\boldsymbol{K}_h \end{bmatrix} \qquad \overline{\boldsymbol{V}}_h = \begin{bmatrix} \boldsymbol{V}_h \\ \boldsymbol{V}_h \end{bmatrix}$$

Note, that here, $\boldsymbol{Q}_h, \boldsymbol{K}_h, \boldsymbol{V}_h$ refer to the query, key and value matrices of **MHA** respectively. Hence, on computing attention, we obtain:

$$\overline{\boldsymbol{P}}_h = \text{softmax}\left(\begin{bmatrix} \boldsymbol{Q}_h \boldsymbol{K}_h^\top & -\boldsymbol{Q}_h \boldsymbol{K}_h^\top \\ -\boldsymbol{Q}_h \boldsymbol{K}_h^\top & \boldsymbol{Q}_h \boldsymbol{K}_h^\top \end{bmatrix} \right) \begin{bmatrix} \boldsymbol{V}_h \\ \boldsymbol{V}_h \end{bmatrix}$$

Further, choosing $R^\ell \in \mathbb{R}^{h \times hp}$ so that it selects only the $(i, 1)$ pseudo-block:

$$R^\ell_{i,\,(i'-1)p+j} = \begin{cases} 1 & \text{if } i' = i \text{ and } j = 1, \\ 0 & \text{otherwise.} \end{cases}$$

The output of each head is:

$$\boldsymbol{O}_h = \text{softmax}\left(\begin{bmatrix} \boldsymbol{Q}_h \boldsymbol{K}_h^\top & -\boldsymbol{Q}_h \boldsymbol{K}_h^\top \end{bmatrix} \right) \begin{bmatrix} \boldsymbol{V}_h \\ \boldsymbol{V}_h \end{bmatrix}$$

Thus the overall **IHA** layer computes a nonlinear function of the input data.

Conclusion. On the repeated-token subspace $\{\mathbf{1}_n \boldsymbol{x}^\top : \boldsymbol{x} \in \mathbb{R}^d\}$, every h-head **MHA** layer reduces to a linear map in $\boldsymbol{x}$, whereas the h-head **IHA** construction above is nonlinear on the same set. Consequently, no h-head **MHA** configuration can represent this **IHA** mapping, and thus $\mathcal{M} \subsetneq \mathcal{P}_p$ for all $p \geq 2$. □

D.2 Representing Polynomial Filters

Theorem 6. (Representing Polynomial Filters): Given a graph adjacency matrix $\boldsymbol{A} \in \mathbb{R}^{N \times N}$ and input features $\boldsymbol{X} \in \mathbb{R}^{N \times d}$ with $d < N$, we concatenate the input with the identity matrix: $\widehat{\boldsymbol{X}} = [\boldsymbol{X}, \boldsymbol{I}]$. For one-layer attention-based multi-head architectures without softmax that are capable of representing all polynomial filter constructions with k heads, there exists an equivalent one-layer attention-based **IHA** architecture without softmax that requires only $\lceil \sqrt{k} \rceil$ heads. In terms of parameter complexity, an **MHA** construction with k heads requires $2n(N+d)k + d(N+d)k$ parameters, whereas the equivalent **IHA** construction with $\lceil \sqrt{k} \rceil$ heads requires $2n(N+d)\lceil \sqrt{k} \rceil + d(d+N)\lceil \sqrt{k} \rceil^2 + 4\lceil \sqrt{k} \rceil^3$ parameters. Here, d denotes

the embedding dimension, k the number of hops in the polynomial filter, and N the number of nodes, and we assume that $k << N$ and $d << N$.

Background. Given fixed input data $\boldsymbol{X} \in \mathbb{R}^{N \times d}$ and a full-rank graph adjacency matrix $\boldsymbol{A} \in \mathbb{R}^{N \times N}$, the goal of this task is to obtain representations that depend on the graph in a polynomial manner. Specifically, given a polynomial order k, our objective is to compute:

$$\widetilde{\boldsymbol{X}} \coloneqq [\boldsymbol{X}, \boldsymbol{A}\boldsymbol{X}, \ldots, \boldsymbol{A}^{k-1}\boldsymbol{X}].$$

Note that, in most cases, it is not possible to recover $\boldsymbol{A}$ or its powers $\boldsymbol{A}^i$ from any linear combination of the input features $\boldsymbol{X}$, primarily because $\boldsymbol{X}$ is typically sparse or low-rank.

Proof. In this proof, our goal is to determine whether we can represent $\widetilde{\boldsymbol{X}}$ using MHA and IHA and if yes, the goal is to understand the number of parameters needed to do so. Since $\boldsymbol{A}$ is full-rank, we cannot directly use $\boldsymbol{X}$. Instead, we define

$$\widehat{\boldsymbol{X}} = [\boldsymbol{X} \quad \boldsymbol{I}],$$

which effectively augments $\boldsymbol{X}$ with positional encodings. Henceforth, we work with $\widehat{\boldsymbol{X}}$. We now examine the constructions and parameter requirements for MHA and IHA, omitting explicit layer dependence ℓ for brevity.

MHA. From the definition, for linear attention, each head in MHA computes

$$\widetilde{\boldsymbol{X}}^{(h)}_{\texttt{MHA}} = \left(\widehat{\boldsymbol{X}}\boldsymbol{W}^h_{Q,\texttt{MHA}} \left(\boldsymbol{W}^h_{K,\texttt{MHA}}\right)^\top \widehat{\boldsymbol{X}}^\top\right) \widehat{\boldsymbol{X}}\boldsymbol{W}^h_{V,\texttt{MHA}}.$$

Since $d < N$, we have $\widehat{\boldsymbol{X}}\boldsymbol{W}^h_{Q,\texttt{MHA}} \left(\boldsymbol{W}^h_{K,\texttt{MHA}}\right)^\top \widehat{\boldsymbol{X}}^\top \in \mathbb{R}^{N \times N}$. To recover a polynomial filter, we must be able to solve

$$[\boldsymbol{X}, \boldsymbol{A}\boldsymbol{X}, \ldots, \boldsymbol{A}^{k-1}\boldsymbol{X}] = \left[\widetilde{\boldsymbol{X}}^{(1)}_{\texttt{MHA}}, \ldots, \widetilde{\boldsymbol{X}}^{(H)}_{\texttt{MHA}}\right].$$

There are multiple design choices that can satisfy this equation. However, we are constrained by the fact that $\boldsymbol{W}^h_{V,\texttt{MHA}}$ cannot depend on the input data $\widehat{\boldsymbol{X}}$, and that the downstream embedding dimension must be kd. Under these constraints, we can construct a minimal solution such that, for each head h,

$$\widehat{\boldsymbol{X}}\boldsymbol{W}^h_{V,\texttt{MHA}} = \boldsymbol{X},$$
$$\left(\widehat{\boldsymbol{X}}\boldsymbol{W}^h_{Q,\texttt{MHA}} \left(\boldsymbol{W}^h_{K,\texttt{MHA}}\right)^\top\right) \widehat{\boldsymbol{X}}^\top = \boldsymbol{A}^{h-1}.$$

This construction requires exactly $2n(N+d)k + d(N+d)k$ parameters. To show that it is indeed minimal, we proceed as follows. We first argue that atleast k heads are needed to represent $\widetilde{\boldsymbol{X}}$, and then we make arguments about the parameters. Towards this we argue about the rank of $\widetilde{\boldsymbol{X}}$. By definition, we know that:

$$\widetilde{\boldsymbol{X}} = \begin{bmatrix}\boldsymbol{X} & \boldsymbol{A}\boldsymbol{X} & \cdots & \boldsymbol{A}^{k-1}\boldsymbol{X}\end{bmatrix}$$

Hence,

$$\begin{aligned}\text{rank}(\widetilde{\boldsymbol{X}}) &= \min\left(N, \sum_{i=0}^{k} \text{rank}(\boldsymbol{A}^i\boldsymbol{X})\right) \\ &\leq \min(N, kd) \\ &\leq kd\end{aligned} \tag{1}$$

Where in the above, we have used that $\text{rank}(\boldsymbol{A}^i\boldsymbol{X}) \leq d, \forall\, i \in \{0, \cdots, k-1\}$, that $kd < N$ by assumption. Moreover, we would also like to note that for a generic $\boldsymbol{X}$, the rank can be tight. That is $\exists X$ such that $\text{rank}(\widetilde{\boldsymbol{X}}) = kd$. It is tight. Hence we will make the argument that ecven IHA needs to have ranks greater or equal to this to be able to represnt the output. Hence, towards this lets first assume that we have the number of heads H to be less than k. Moreover, to match the dimensions

of $\widetilde{\boldsymbol{X}}$, we let the value weights for some heads to be arbitrary such that the final dimensions match. We weould like to note that for any particular head h $\boldsymbol{W}^h_{K,\texttt{MHA}}$ must depend only on $\boldsymbol{A}$ and not on $\boldsymbol{X}$; hence, we can write

$$\boldsymbol{W}^h_{K,\texttt{MHA}} = \begin{bmatrix} B_h(\boldsymbol{A})_{d\times d_h} \\ C_h(\boldsymbol{A})_{N\times d_h} \end{bmatrix}.$$

Note that for the dimensions to match, we would need $\sum_{h=1}^{H} d_h = kd$. Moreover, per head, we define for shorthand, $\text{Att}_h(\widehat{\boldsymbol{X}}) := \left(\widehat{\boldsymbol{X}}\boldsymbol{W}^h_{Q,\texttt{MHA}}\left(\boldsymbol{W}^h_{K,\texttt{MHA}}\right)^\top\right)\widehat{\boldsymbol{X}}^\top$.

We let the representation of MHA for H number of heads less than k be denoted by $\widetilde{\boldsymbol{X}}'$, where:

$$\widetilde{\boldsymbol{X}}' = \left[\text{Att}_1(\widehat{\boldsymbol{X}})\boldsymbol{X}B_1(\boldsymbol{A}) + \text{Att}_1(\widehat{\boldsymbol{X}})C_1(\boldsymbol{A}),\ \cdots,\ \text{Att}_h(\widehat{\boldsymbol{X}})\boldsymbol{X}B_H(\boldsymbol{A}) + \text{Att}_H(\widehat{\boldsymbol{X}})C_H(\boldsymbol{A})\right]$$

Now since we want $\widetilde{\boldsymbol{X}}' = \widetilde{\boldsymbol{X}}, \forall\ \boldsymbol{X}$, if we substitute $\boldsymbol{X} = \boldsymbol{0}$, then we can easily see that would imply that $\forall h \in \{1, \cdots, H\}$ $\text{Att}_h C_h(\boldsymbol{A}) = \boldsymbol{0}$. However, $\text{Att}_h \neq 0$ as if it were $\boldsymbol{0}$, then $\widetilde{\boldsymbol{X}}' = \boldsymbol{0}$ which would then imply that $\widetilde{\boldsymbol{X}}' \neq \widetilde{\boldsymbol{X}}$ for any arbitrary non-zero $\boldsymbol{X}$. Hence, the only solution is that $C_h(\boldsymbol{A}) = \boldsymbol{0}$. Hence, we can cleanly write that:

$$\widetilde{\boldsymbol{X}}' = \left[\text{Att}_1(\widehat{\boldsymbol{X}})\boldsymbol{X}B_1(\boldsymbol{A}),\ \cdots,\ \text{Att}_H(\widehat{\boldsymbol{X}})\boldsymbol{X}B_H(\boldsymbol{A})\right]$$

Now we try to compute the rank of $\widetilde{\boldsymbol{X}}'$. Note that again, we have assumed that $H < k$. Hence,

$$\begin{aligned}
\text{rank}(\widetilde{\boldsymbol{X}}') &= \min\left(N, \sum_{i=1}^{H} \text{rank}(\text{Att}_1(\widehat{\boldsymbol{X}})\boldsymbol{X}B_i(\boldsymbol{A}))\right) \\
&\leq \min\left(N, \sum_{i=1}^{H} \min\left(\text{rank}(\text{Att}_1(\widehat{\boldsymbol{X}})),\ \text{rank}(\boldsymbol{X}),\ \text{rank}(B_i(\boldsymbol{A}))\right)\right) \\
&\leq \min(N, Hd) \\
&\leq Hd
\end{aligned} \tag{2}$$

Now from Eq. 1, we can see that the rank of the concatenation of the embeddings of polynomial filter are kd when tight, however the rank in the case of $\widetilde{\boldsymbol{X}}'$ is at most Hd. If $H < k$, clearly the polynomial filter is more expressive and has a higher rank than that of $\widetilde{\boldsymbol{X}}'$ which implies that if $H < k$, then the polynomial filter cannot be represented Multi-Head attention. Now we argue that for heads $H > k$, there are constructions that can represnt the polynomial filter, however, it is not tight in terms of the parameters. Let us assume that there are $H > k$ heads. Moreover, we only make the arguement for the first head of MHA, and then this argument will essentially hold for all of the heads. We split this situation into multiple cases. Note that we are not really specifying what the dimension of each head in this case and assume it to be arbitrary. Hence, for the first case, we assume that the output dimension of the head is less than that of the first output of the polynomial filter. Hence, for equality, we need:

$$\text{Att}_1(\widehat{\boldsymbol{X}})XB_1(\boldsymbol{A}) = XQ_1$$

Note that $B_1(\boldsymbol{A}) \in \mathbb{R}^{N\times d_1}$, where $d_1 \leq d$ Note that $Q_1 \in \mathbb{R}^{d\times d_1}$, and clearly Q_1 is a d_1 rank matrix with one hot vectors across each column to isolate the rows that correspond to $\text{Att}_1(\widehat{\boldsymbol{X}})XB_1(\boldsymbol{A})$. Therefore, on taking the $\text{vec}(\cdot)$ operator on both sides, we obtain:

$$\left(B(\boldsymbol{A})^T \otimes \text{Att}_1(\widehat{\boldsymbol{X}})\right)\text{vec}(\boldsymbol{X}) = \left(Q_1^T \otimes \boldsymbol{I}\right)\text{vec}(X)$$

Since, we want this to be true for all X, clearly,

$$\begin{aligned}
&B(\boldsymbol{A})^T \otimes \text{Att}_1(\widehat{\boldsymbol{X}}) = Q_1^T \otimes \boldsymbol{I} \\
\implies &\text{rank}(\text{Att}_1(\widehat{\boldsymbol{X}})) = \text{rank}(Q_1^T)\cdot\text{rank}(\boldsymbol{I})/\text{rank}(B(\boldsymbol{A}))
\end{aligned}$$

Note that we have used $\text{rank}(\boldsymbol{A}\otimes\boldsymbol{B}) = \text{rank}(\boldsymbol{A})\cdot\text{rank}(\boldsymbol{B})$ We know that $\text{rank}(Q_1^T) = d_1$ and hence, since $\text{rank}(B(\boldsymbol{A})) \leq d_1$, then clearly,

$$\text{rank}(\text{Att}_1(\widehat{\boldsymbol{X}})) \geq N$$

For this to be satisfied, we can see that the query and the key matrices both have to be of size at-least $(N+d)N$. Moreover, the value matrix weights are then going to be of size $(N+d)d_1$. Hence, the total dimensions required for this is $2(N+d)N + (N+d)d_1$. We then argue about the second case which is what happens if $d_1 > d$. For the argument, we assume that this spans two actual polynimial filters that is $d < d_1 < 2d$, but we will then show that the argument will hold even if d_1 was arbitarty. Due to this, we obtain that, for equality between the representations when the number of heads are greater than that of the degree of the polynomial filter ($H > k$), the following:

$$\Big[\text{Att}_1(\widehat{\boldsymbol{X}})\boldsymbol{X}B_1(\boldsymbol{A})\boldsymbol{Q}_1 \quad \text{Att}_1(\widehat{\boldsymbol{X}})\boldsymbol{X}B_1(\boldsymbol{A})\boldsymbol{Q}_2\Big] = \big[\boldsymbol{X} \quad \boldsymbol{A}\boldsymbol{X}\widehat{\boldsymbol{Q}}_{\boldsymbol{2}}\big]$$

Note that $\boldsymbol{Q_1} \in \mathbb{R}^{d_1 \times d}, \boldsymbol{Q_2} \in \mathbb{R}^{d_1 \times (d_1 - d)}, \widehat{\boldsymbol{Q}}_{\boldsymbol{2}} \in \mathbb{R}^{d \times d_1}$, with ranks $d, d_1, d_1 - d$ respectively. Now we just equate them,

$$\Big((B_1(\boldsymbol{A})\boldsymbol{Q}_1)^T \otimes \text{Att}_1(\widehat{\boldsymbol{X}})\Big) \cdot \text{vec}\boldsymbol{X} = (\boldsymbol{I}_{d\times d} \otimes \boldsymbol{I}_{N\times N}) \cdot \text{vec}\boldsymbol{X}$$
$$\Big((B_1(\boldsymbol{A})\boldsymbol{Q}_2)^T \otimes \text{Att}_1(\widehat{\boldsymbol{X}})\Big) \cdot \text{vec}\boldsymbol{X} = \Big(\hat{\boldsymbol{Q}}_2 \otimes \boldsymbol{A}\Big) \cdot \text{vec}\boldsymbol{X}$$

Now we use the same rank argument as before, to conclude that $\text{rank}(\text{Att}_1(\widehat{\boldsymbol{X}})) \geq N$, which again implies that the key and query weight matrices have weights $(N+d)N$ respectively.

We make two generalizations now. We would first like to note that this argument also holds when $d_1 \geq (k-1)d$ and hence, the above proof works for any dimensions. We would also like to note that this while this proof is done for the first head of MHA, it also generlizes to other heads. The proof holds similarly, and constructively that is for the second head of MHA, one can make the similar argument as that of the first with the only difference being that one needs to udnerstand the dimensions of the output polynomial filter that is which dimensions of the second head of MHA corresponds to which dimensiond of the polynomial filter. Then by repeating the argument of the ranks, one can obtain that again, key and query weight matrices have weights $(N+d)N$ respectively. This argument can be repeated for the third head and so on to finally conclude that the number of dimensions needed to represent this is $Hn(N+d) + (d+N)(kd)$, when $H \geq k$. Clearly, this is minimal when $H = k$.

IHA. We present a construction that achieves comparable parametric complexity while requiring only $\lceil\sqrt{k}\rceil$ heads. The proof proceeds by explicit construction. We note that $\widetilde{\boldsymbol{X}}^{(\mathbf{1})}$ denotes the representation after the first layer of attention.

$$\boldsymbol{W}_{K,\texttt{IHA}}^{(1,h)} = \begin{bmatrix}\mathbf{0}_{d\times N} \\ (A^{h-1})^\top\end{bmatrix} \;\forall\, h \in \{1,2,\cdots,\lceil\sqrt{k}\rceil\}$$
$$\boldsymbol{W}_{Q,\texttt{IHA}}^{(1,h)} = \begin{bmatrix}\mathbf{0}_{d\times N} \\ A^{(h-1)\cdot\lceil\sqrt{k}\rceil}\end{bmatrix} \;\forall\, h \in \{1,2,\cdots,\lceil\sqrt{k}\rceil\}$$
$$\boldsymbol{W}_{V,\texttt{IHA}}^{(1,h)} = \begin{bmatrix}\boldsymbol{L}_{d\times d\lceil\sqrt{k}\rceil}^{h} \\ \mathbf{0}_{N\times d\lceil\sqrt{k}\rceil}\end{bmatrix} \in \mathbb{R}^{(N+d)\times\lceil d\sqrt{k}\rceil}$$
$$\text{where, } \boldsymbol{L}_{d\times d\lceil\sqrt{k}\rceil}^{(1,h)} = \big[\mathbf{0}_{d\times(h-1)d} \quad I_{d\times d} \quad \mathbf{0}_{d\times\lceil\sqrt{k}\rceil - hd}\big]\,\forall\, h \;\in \{1,\;2,\;\cdots,\;\lceil\sqrt{k}\rceil\}$$
$$\text{with the convention that } \mathbf{0}_{d\times 0} \text{ represents an empty vector.}$$
$$\text{moreover, } m_{k,h}^{(1)} = 1\;\forall k,h \in \{1,2,\cdots,\lceil\sqrt{k}\rceil\}$$

Hence, after one layer of attention we obtain the following. Using bracket notation $[\cdot,\cdot]$ for concatenation:

$$\widetilde{\boldsymbol{X}}^{(\mathbf{1})} = \Big[\widetilde{\boldsymbol{X}}_{\texttt{IHA}}^{(1,1)},\; \widetilde{\boldsymbol{X}}_{\texttt{IHA}}^{(1,2)},\; \ldots,\; \widetilde{\boldsymbol{X}}_{\texttt{IHA}}^{(1,\lceil\sqrt{k}\rceil)}\Big]$$

where each $\widetilde{\boldsymbol{X}}_{\texttt{IHA}}^{(1,h)}$ aggregates over all key heads.

$$\widetilde{\boldsymbol{X}}^{(\mathbf{1})} = \Big[\boldsymbol{X} \quad \boldsymbol{A}\boldsymbol{X} \quad \cdots \quad \boldsymbol{A}^{\lceil\sqrt{k}\rceil^2-1}\boldsymbol{X}\Big]$$

Hence, under this construction, the number of parameters is $2n(N+d)\lceil\sqrt{k}\rceil + d(d+N)\lceil\sqrt{k}\rceil^2) + \lceil\sqrt{k}\rceil^2$

Example. We present an example here to solidify the intuition of why such a construction helps. We assume that $k = 4$. Hence, using the above constructions, the query, key, value matrices are defined as follows.

$$\begin{aligned}
\boldsymbol{W}_{Q,\texttt{IHA}}^{(1,1)} &= \begin{bmatrix} \mathbf{0}_{d\times N} \\ \boldsymbol{I}_{N\times N} \end{bmatrix} \quad \boldsymbol{W}_{Q,\texttt{IHA}}^{(1,2)} = \begin{bmatrix} \mathbf{0}_{d\times N} \\ \left(A^2\right)^\top \end{bmatrix} \\
\boldsymbol{W}_{K,\texttt{IHA}}^{(1,1)} &= \begin{bmatrix} \mathbf{0}_{d\times N} \\ \boldsymbol{I}_{N\times N} \end{bmatrix} \quad \boldsymbol{W}_{K,\texttt{IHA}}^{(1,2)} = \begin{bmatrix} \mathbf{0}_{d\times N} \\ \left(A\right)^\top \end{bmatrix} \\
\boldsymbol{W}_{V,\texttt{IHA}}^{(1,1)} &= \begin{bmatrix} \boldsymbol{I}_{d\times d} & \mathbf{0}_{d\times d} \\ \mathbf{0}_{N\times d} & \mathbf{0}_{N\times d} \end{bmatrix} \quad \boldsymbol{W}_{V,\texttt{IHA}}^{(1,2)} = \begin{bmatrix} \mathbf{0}_{d\times d} & \boldsymbol{I}_{d\times d} \\ \mathbf{0}_{N\times d} & \mathbf{0}_{N\times d} \end{bmatrix} \\
m_{k,h}^{(1)} &= 1 \; \forall \, k, h \in \{1, 2\}
\end{aligned}$$

We can see that first head of attention computes to

$$\begin{aligned}
Z_1 &= \begin{bmatrix} \left(\widehat{\boldsymbol{X}}\boldsymbol{W}_{Q,\texttt{IHA}}^{(1,1)}(\boldsymbol{W}_{K,\texttt{IHA}}^{(1,1)})^\top\widehat{\boldsymbol{X}}^\top\right) & \left(\widehat{\boldsymbol{X}}\boldsymbol{W}_{Q,\texttt{IHA}}^{(1,1)}(\boldsymbol{W}_{K,\texttt{IHA}}^{(1,2)})^\top\widehat{\boldsymbol{X}}^\top\right) \end{bmatrix} \begin{bmatrix} \widehat{X}\boldsymbol{W}_{V,\texttt{IHA}}^{(1,1)} \\ \widehat{X}\boldsymbol{W}_{V,\texttt{IHA}}^{(1,2)} \end{bmatrix} \\
&= \begin{bmatrix} \boldsymbol{I} & \boldsymbol{A} \end{bmatrix} \begin{bmatrix} \boldsymbol{X} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{X} \end{bmatrix} \\
&= \begin{bmatrix} \boldsymbol{X} & \boldsymbol{A}\boldsymbol{X} \end{bmatrix}
\end{aligned}$$

Similarly, for the second layer of attention, we obtain

$$\begin{aligned}
Z_2 &= \begin{bmatrix} \left(\widehat{\boldsymbol{X}}\boldsymbol{W}_{Q,\texttt{IHA}}^{(1,2)}(\boldsymbol{W}_{K,\texttt{IHA}}^{(1,1)})^\top\widehat{\boldsymbol{X}}^\top\right) & \left(\widehat{\boldsymbol{X}}\boldsymbol{W}_{Q,\texttt{IHA}}^{(1,2)}(\boldsymbol{W}_{K,\texttt{IHA}}^{(1,2)})^\top\widehat{\boldsymbol{X}}^\top\right) \end{bmatrix} \begin{bmatrix} \widehat{X}\boldsymbol{W}_{V,\texttt{IHA}}^{(1,1)} \\ \widehat{X}\boldsymbol{W}_{V,\texttt{IHA}}^{(1,2)} \end{bmatrix} \\
&= \begin{bmatrix} \boldsymbol{A}^2 & \boldsymbol{A}^3 \end{bmatrix} \begin{bmatrix} \boldsymbol{X} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{X} \end{bmatrix} \\
&= \begin{bmatrix} \boldsymbol{A}^2\boldsymbol{X} & \boldsymbol{A}^3\boldsymbol{X} \end{bmatrix}
\end{aligned}$$

Hence, on concatenating the the embeddings from both the heads, we obtain

$$\widetilde{\boldsymbol{X}}^{(\mathbf{1})} = \begin{bmatrix} \boldsymbol{X} & \boldsymbol{A}\boldsymbol{X} & \boldsymbol{A}^2\boldsymbol{X} & \boldsymbol{A}^3\boldsymbol{X} \end{bmatrix}$$

□

IHA. We present a construction that requires only $H \coloneqq \lceil\sqrt{k}\rceil$ heads. We set the number of pseudo heads to be $P \coloneqq H$. Note that the input to **IHA** is the same as **MHA** as defined below.

$$\widehat{\boldsymbol{X}} = \begin{bmatrix} \boldsymbol{X} & \boldsymbol{I} \end{bmatrix},$$

The proof proceeds by explicit construction. For every base head index $m \in \{1, 2, \ldots, H\}$ define

$$\begin{aligned}
\boldsymbol{W}_{K,\texttt{IHA}}^{(1,m)} &= \begin{bmatrix} \mathbf{0}_{d\times N} \\ \left(A^{m-1}\right)^\top \end{bmatrix}, \\
\boldsymbol{W}_{Q,\texttt{IHA}}^{(1,m)} &= \begin{bmatrix} \mathbf{0}_{d\times N} \\ A^{(m-1)\cdot H} \end{bmatrix}, \\
\boldsymbol{W}_{V,\texttt{IHA}}^{(1,m)} &= \begin{bmatrix} \boldsymbol{L}_{d\times dH}^{(1,m)} \\ \mathbf{0}_{N\times dH} \end{bmatrix} \in \mathbb{R}^{(N+d)\times dH},
\end{aligned}$$

where $\boldsymbol{L}_{d\times dH}^{(1,m)}$ routes into the m-th d-block (same selector trick as before):

$$\boldsymbol{L}_{d\times dH}^{(1,m)} \;\coloneqq\; \begin{bmatrix} \mathbf{0}_{d\times(m-1)d} & I_{d\times d} & \mathbf{0}_{d\times(H-m)d} \end{bmatrix}, \qquad \forall\, m \in \{1, \ldots, H\},$$

with the convention that $\mathbf{0}_{d\times 0}$ is empty.

We choose pseudo-head coefficients to be one-hot routers so that, *inside each head* h, the $P = H$ pseudo-heads instantiate the same "key heads":

$$\begin{aligned}
&\alpha_{m,h,j}^{Q} \coloneqq \mathbb{1}[m = h] \cdot \mathbb{1}[j = 1], \\
&\alpha_{m,h,j}^{K} \coloneqq \mathbb{1}[m = j], \\
&\alpha_{m,h,j}^{V} \coloneqq \mathbb{1}[m = j], \qquad \forall\, m, h \in \{1, \ldots, H\},\ \forall j \in \{1, \ldots, P\}.
\end{aligned}$$

Thus, for each head h and pseudo-head j,

$$\widetilde{\boldsymbol{Q}}_{h,j} = \sum_{m=1}^{H} \alpha^{Q}_{m,h,j}\, \widehat{\boldsymbol{X}}\boldsymbol{W}^{(1,m)}_{Q,\texttt{IHA}} = \mathbb{1}[j=1]\, \widehat{\boldsymbol{X}}\boldsymbol{W}^{(1,h)}_{Q,\texttt{IHA}},$$

$$\widetilde{\boldsymbol{K}}_{h,j} = \sum_{m=1}^{H} \alpha^{K}_{m,h,j}\, \widehat{\boldsymbol{X}}\boldsymbol{W}^{(1,m)}_{K,\texttt{IHA}} = \widehat{\boldsymbol{X}}\boldsymbol{W}^{(1,j)}_{K,\texttt{IHA}},$$

$$\widetilde{\boldsymbol{V}}_{h,j} = \sum_{m=1}^{H} \alpha^{V}_{m,h,j}\, \widehat{\boldsymbol{X}}\boldsymbol{W}^{(1,m)}_{V,\texttt{IHA}} = \widehat{\boldsymbol{X}}\boldsymbol{W}^{(1,j)}_{V,\texttt{IHA}}.$$

For each head h, **IHA** stacks pseudo-queries and keys row-wise:

$$\overline{\boldsymbol{Q}}_h \coloneqq \left[\widetilde{\boldsymbol{Q}}^{\top}_{h,1}; \ldots; \widetilde{\boldsymbol{Q}}^{\top}_{h,H}\right]^{\top}, \quad \overline{\boldsymbol{K}}_h \coloneqq \left[\widetilde{\boldsymbol{K}}^{\top}_{h,1}; \ldots; \widetilde{\boldsymbol{K}}^{\top}_{h,H}\right]^{\top}, \quad \overline{\boldsymbol{V}}_h \coloneqq \left[\widetilde{\boldsymbol{V}}^{\top}_{h,1}; \ldots; \widetilde{\boldsymbol{V}}^{\top}_{h,H}\right]^{\top}.$$

IHA computes $\overline{\boldsymbol{P}}_h = \mathrm{softmax}(\overline{\boldsymbol{Q}}_h \overline{\boldsymbol{K}}_h^{\top})\overline{\boldsymbol{V}}_h$. Consider the output corresponding to the *first pseudo query* (i.e., the first N rows of $\overline{\boldsymbol{P}}_h$), which we denote by $\boldsymbol{P}_{h,1} \in \mathbb{R}^{N \times dH}$. By construction, $\widetilde{\boldsymbol{Q}}_{h,1} = \widehat{\boldsymbol{X}}\boldsymbol{W}^{(1,h)}_{Q,\texttt{IHA}}$ and $\widetilde{\boldsymbol{K}}_{h,j} = \widehat{\boldsymbol{X}}\boldsymbol{W}^{(1,j)}_{K,\texttt{IHA}}$ for all j, so the same "aggregate over all key heads" block product:

$$\boldsymbol{P}_{h,1} = \left[\left(\widehat{\boldsymbol{X}}\boldsymbol{W}^{(1,h)}_{Q,\texttt{IHA}}(\boldsymbol{W}^{(1,1)}_{K,\texttt{IHA}})^{\top}\widehat{\boldsymbol{X}}^{\top}\right) \quad \cdots \quad \left(\widehat{\boldsymbol{X}}\boldsymbol{W}^{(1,h)}_{Q,\texttt{IHA}}(\boldsymbol{W}^{(1,H)}_{K,\texttt{IHA}})^{\top}\widehat{\boldsymbol{X}}^{\top}\right)\right] \begin{bmatrix} \widehat{\boldsymbol{X}}\boldsymbol{W}^{(1,1)}_{V,\texttt{IHA}} \\ \vdots \\ \widehat{\boldsymbol{X}}\boldsymbol{W}^{(1,H)}_{V,\texttt{IHA}} \end{bmatrix}.$$

Using the definitions of $\boldsymbol{W}^{(1,h)}_{Q}$ and $\boldsymbol{W}^{(1,j)}_{K}$,

$$\widehat{\boldsymbol{X}}\boldsymbol{W}^{(1,h)}_{Q,\texttt{IHA}}(\boldsymbol{W}^{(1,j)}_{K,\texttt{IHA}})^{\top}\widehat{\boldsymbol{X}}^{\top} \;=\; A^{(h-1)H}\, A^{j-1} \;=\; A^{(h-1)H+(j-1)}.$$

Moreover, $\widehat{\boldsymbol{X}}\boldsymbol{W}^{(1,j)}_{V,\texttt{IHA}}$ routes $\boldsymbol{X}$ into the j-th d-block inside the dH-dimensional head space. Therefore,

$$\boldsymbol{P}_{h,1} = \left[A^{(h-1)H}\boldsymbol{X} \quad A^{(h-1)H+1}\boldsymbol{X} \quad \cdots \quad A^{(h-1)H+(H-1)}\boldsymbol{X}\right].$$

IHA then collapses the HP pseudo-outputs down to H heads via $\boldsymbol{R} \in \mathbb{R}^{H \times HP}$. We choose $\boldsymbol{R}$ to select *only* pseudo $j=1$ from each head h:

$$\boldsymbol{R}_{h,(h-1)H+1} = 1, \qquad \boldsymbol{R}_{h,(h-1)H+j} = 0 \;\; \forall j \in \{2, \ldots, H\}, \qquad \boldsymbol{R}_{h,(h'-1)H+j} = 0 \;\; \forall h' \neq h.$$

Hence the collapsed head output is

$$\boldsymbol{O}_h \;=\; \boldsymbol{P}_{h,1} \;=\; \left[A^{(h-1)H}\boldsymbol{X} \quad A^{(h-1)H+1}\boldsymbol{X} \quad \cdots \quad A^{(h-1)H+(H-1)}\boldsymbol{X}\right].$$

Concatenate heads. Finally on concatenating representations from different heads, we obtain,

$$\widetilde{\boldsymbol{X}}^{(\mathbf{1})} = \left[\boldsymbol{O}_1, \boldsymbol{O}_2, \ldots, \boldsymbol{O}_H\right] = \left[\boldsymbol{X} \quad \boldsymbol{A}\boldsymbol{X} \quad \cdots \quad \boldsymbol{A}^{H^2-1}\boldsymbol{X}\right].$$

Thus the construction realizes all powers up to A^{H^2-1} in one layer. If $H^2 > k$, the extra $(H^2 - k)$ blocks may be treated as padding (or zeroed by an output mask).

Hence, under this construction, the number of parameters is $2n(N+d)\lceil\sqrt{k}\rceil + d(d+N)\lceil\sqrt{k}\rceil^2) + 4\lceil\sqrt{k}\rceil^3$

Example ($k = 4$). Let $k = 4$, so $H = P = 2$. Then

$$\boldsymbol{W}^{(1,1)}_{Q} = \begin{bmatrix}0\\ I\end{bmatrix}, \quad \boldsymbol{W}^{(1,2)}_{Q} = \begin{bmatrix}0\\ A^2\end{bmatrix}, \quad \boldsymbol{W}^{(1,1)}_{K} = \begin{bmatrix}0\\ I\end{bmatrix}, \quad \boldsymbol{W}^{(1,2)}_{K} = \begin{bmatrix}0\\ A^{\top}\end{bmatrix},$$

and $\boldsymbol{W}^{(1,1)}_{V}, \boldsymbol{W}^{(1,2)}_{V}$ route into the first/second d-block, exactly as in the old proof. Choose α one-hot as above, and choose $\boldsymbol{R}$ to pick pseudo $j=1$ from each head. Then head $h=1$ outputs $[\boldsymbol{X}, \boldsymbol{A}\boldsymbol{X}]$, head $h=2$ outputs $[\boldsymbol{A}^2\boldsymbol{X}, \boldsymbol{A}^3\boldsymbol{X}]$, and concatenation yields

$$\widetilde{\boldsymbol{X}}^{(\mathbf{1})} = \left[\boldsymbol{X} \quad \boldsymbol{A}\boldsymbol{X} \quad \boldsymbol{A}^2\boldsymbol{X} \quad \boldsymbol{A}^3\boldsymbol{X}\right].$$

D.3 Representing Count Permutation Match 3 (CPM-3)

Theorem 7. (Count Permutation Match-3): Let $N_{\max}$ denote the maximum number of tokens that can be processed by the model in the worst case. There exists a one-layer transformer with *interleaved-head attention* (**IHA**) that can represent the permutation match-3 task using $\lceil\sqrt{N_{\max}}\rceil$ attention heads. The number of parameters required by this IHA construction is upper bounded by $37N_{\max}^2\sqrt{N_{\max}} + N_{\max}^2(N_{\max}-1) + N_{\max}^2$. In contrast, the best currently known construction based on *multi-head attention* (**MHA**) requires $N_{\max}$ attention heads, and its parameter count is lower bounded by $3N_{\max}^3 + N_{\max}^2(N_{\max}-1) + N_{\max}^2$. Throughout, we assume the vocabulary size is at most on the order of the maximum sequence length, i.e., $|\mathcal{V}| = O(N_{\max})$.

Count Permutation Match-3 (CPM-3): We define a task denoted as Count Permutation Match-3 (CPM-3) where the goal is given a sequence of natural numbers denoted as $\{x_i\}_{i\in\mathbb{N}}$, the goal is to be able to count the number of occurrences of a specific function, defined as follows. Let us define the number of triples (i, j_1, j_2) where i denotes the token in consideration, that satisfy:

$$\mathrm{CPM}_i(3) = \mathrm{Count}\Big(\forall j_1, j_2 \ : \phi(x_i, x_{j_1}, x_{j_2}) = 0\Big),$$

where,

$$\phi(x_i, x_{j_1}, x_{j_2}) := x_i + Gx_{j_1} + x_{j_2} \bmod M$$

where M is an arbitrary number and G is another number such that$G > 2M$.

The above condition is required to make sure that the function is not permutation invariant. That is, if $x_{j_1} \neq x_{j_2}$ then, one can clearly see that:

$$\phi(x_i, x_{j_1}, x_{j_2}) \neq \phi(x_i, x_{j_2}, x_{j_1})$$

Example. Let the sequence be $(1, 2, 3)$ and let $G = 10$. Then we can compute $x_{j_1} + x_{j_2}$ as follows:

$$11,\ 12,\ 13,\ 21,\ 22,\ 23,\ 31,\ 32,\ 33$$

Then we can finally compute $x_i + x_{j_1} + x_{j_2}$ which is computed as follows:

$$1 + x_{j_1} + x_{j_2} := 12,\ 13,\ 14,\ 22,\ 23,\ 24,\ 32,\ 33,\ 34$$
$$2 + x_{j_1} + x_{j_2} := 13,\ 14,\ 15,\ 23,\ 24,\ 25,\ 33,\ 34,\ 35$$
$$3 + x_{j_1} + x_{j_2} := 14,\ 15,\ 16,\ 24,\ 25,\ 26,\ 34,\ 35,\ 36$$

$$\mathrm{CPM}_1(3) := 3$$
$$\mathrm{CPM}_2(3) := 3$$
$$\mathrm{CPM}_3(3) := 3$$

Proof. The proof will follow by construction as before. That is, we show that a representative solution exists, which is constructed via different components such as attention, MLP etc. We first describe the Encoder and Positional Encodings below

IHA. We first show the **IHA** construction and then proceed to the **MHA** construction.

We prove realizability by explicit construction. The construction has three components: (i) an encoder/positional encoding map, (ii) one **IHA** attention layer (with pseudo-heads) that brings all symbols into a single vector space at each position via structured cyclic shifts, and (iii) an MLP that computes $\mathrm{CPM}_i(3)$ by enumerating ordered pairs (j_1, j_2) and aggregating indicators of the constraint $\phi(x_i, x_{j_1}, x_{j_2}) \equiv 0 \pmod M$.

Encoder and Positional Encoding. The input is a length-$N_{\max}$ sequence of natural numbers $\{x_i\}_{i=1}^{N_{\max}}$. Assume the encoder and positional encoding produce

$$\hat{\boldsymbol{X}} = [\,\boldsymbol{X},\ \boldsymbol{I}\,],$$

where $\boldsymbol{X} \in \mathbb{R}^{N_{\max}\times 1}$ stores the scalar token values, and $\boldsymbol{I} \in \mathbb{R}^{N_{\max}\times N_{\max}}$ is the identity positional encoding. Hence

$$\hat{\boldsymbol{X}} \in \mathbb{R}^{N_{\max}\times(N_{\max}+1)}.$$

IHA Attention (Layer 1). Let $\boldsymbol{P} \in \mathbb{R}^{N_{\max} \times N_{\max}}$ denote the cyclic permutation matrix and $\boldsymbol{e}_1 = [1, \mathbf{0}_{1 \times N_{\max}}]^\top$. Set

$$H \coloneqq \lceil \sqrt{N_{\max}} \rceil, \qquad P \coloneqq H.$$

We construct a single new-IHA layer with H heads and P pseudos per head. For each index $m \in \{1, \dots, H\}$ define

$$\boldsymbol{W}_Q^{(m)} \coloneqq \begin{bmatrix} \mathbf{0}_{1 \times N_{\max}} \\ \boldsymbol{P}^{(m-1)H} \end{bmatrix}, \qquad \boldsymbol{W}_K^{(m)} \coloneqq \begin{bmatrix} \mathbf{0}_{1 \times N_{\max}} \\ \left(\boldsymbol{P}^{m-1}\right)^\top \end{bmatrix},$$

and define value projections that route the scalar symbol coordinate into the m-th coordinate of an H-dimensional value space:

$$\boldsymbol{W}_V^{(1)} = \left[\boldsymbol{e_1} \cdot \lceil \sqrt{N_{\max}} \rceil \quad \mathbf{0}_{(N_{\max}+1) \times (\lceil \sqrt{N_{\max}} \rceil - 1)}\right],$$

$$\boldsymbol{W}_{V,\texttt{IHA}}^{(1,\lceil \sqrt{N_{\max}} \rceil)} = \left[\mathbf{0}_{(N_{\max}+1) \times (\lceil \sqrt{N_{\max}} \rceil - 1)} \quad \boldsymbol{e_1} \cdot \lceil \sqrt{N_{\max}} \rceil\right],$$

$$\boldsymbol{W}_{V,\texttt{IHA}}^{(1,h)} = \left[\mathbf{0}_{(N_{\max}+1) \times (h-1)} \quad \boldsymbol{e_1} \cdot \lceil \sqrt{N_{\max}} \rceil \quad \mathbf{0}_{(N_{\max}+1) \times (\lceil \sqrt{N_{\max}} \rceil - h)}\right],$$

Pseudo-head mixing. Let $\alpha^Q, \alpha^K, \alpha^V \in \mathbb{R}^{H \times H \times P}$ be the new-IHA mixing coefficients. We set them to one-hot routers:

$$\begin{aligned} \alpha^Q_{m,h,j} &\coloneqq \mathbb{1}[m = h] \cdot \mathbb{1}[j = 1], \\ \alpha^K_{m,h,j} &\coloneqq \mathbb{1}[m = j], \\ \alpha^V_{m,h,j} &\coloneqq \mathbb{1}[m = j], \qquad \forall m, h \in \{1, \dots, H\},\ \forall j \in \{1, \dots, P\}. \end{aligned}$$

Consequently, for each head h and pseudo j,

$$\begin{aligned} \widetilde{\boldsymbol{Q}}_{h,j} &= \sum_{m=1}^{H} \alpha^Q_{m,h,j}\, \hat{\boldsymbol{X}} \boldsymbol{W}_Q^{(m)} = \mathbb{1}[j = 1]\, \hat{\boldsymbol{X}} \boldsymbol{W}_Q^{(h)}, \\ \widetilde{\boldsymbol{K}}_{h,j} &= \sum_{m=1}^{H} \alpha^K_{m,h,j}\, \hat{\boldsymbol{X}} \boldsymbol{W}_K^{(m)} = \hat{\boldsymbol{X}} \boldsymbol{W}_K^{(j)}, \\ \widetilde{\boldsymbol{V}}_{h,j} &= \sum_{m=1}^{H} \alpha^V_{m,h,j}\, \hat{\boldsymbol{X}} \boldsymbol{W}_V^{(m)} = \hat{\boldsymbol{X}} \boldsymbol{W}_V^{(j)}. \end{aligned}$$

Pseudo-major stacking and attention. For each $h \in \{1, \dots, H\}$ define the stacked pseudo matrices

$$\overline{\boldsymbol{Q}}_h \coloneqq \left[\widetilde{\boldsymbol{Q}}_{h,1}^\top; \dots; \widetilde{\boldsymbol{Q}}_{h,P}^\top\right]^\top, \quad \overline{\boldsymbol{K}}_h \coloneqq \left[\widetilde{\boldsymbol{K}}_{h,1}^\top; \dots; \widetilde{\boldsymbol{K}}_{h,P}^\top\right]^\top, \quad \overline{\boldsymbol{V}}_h \coloneqq \left[\widetilde{\boldsymbol{V}}_{h,1}^\top; \dots; \widetilde{\boldsymbol{V}}_{h,P}^\top\right]^\top.$$

Let

$$\boldsymbol{S}_h \coloneqq \frac{1}{\sqrt{H}}\, \overline{\boldsymbol{Q}}_h \overline{\boldsymbol{K}}_h^\top, \qquad \overline{\boldsymbol{P}}_h \coloneqq \mathrm{softmax}(\boldsymbol{S}_h)\, \overline{\boldsymbol{V}}_h.$$

We set the softmax temperature to 0 (hard attention), so that the attention map implements the unique routing induced by the permutation structure. Define $\boldsymbol{P}_{h,1} \in \mathbb{R}^{N_{\max} \times H}$ to be the output block of $\overline{\boldsymbol{P}}_h$ corresponding to the first pseudo (i.e., the rows aligned with $\widetilde{\boldsymbol{Q}}_{h,1}$). Under hard attention, the construction ensures that $\widetilde{\boldsymbol{Q}}_{h,1}$ routes to each pseudo-key $\widetilde{\boldsymbol{K}}_{h,j}$ according to the cyclic shifts, yielding

$$\boldsymbol{P}_{h,1} = \sum_{j=1}^{H} \boldsymbol{P}^{(h-1)H+(j-1)}\, \hat{\boldsymbol{X}} \boldsymbol{W}_V^{(j)}.$$

Since $\hat{\boldsymbol{X}} \boldsymbol{W}_V^{(j)}$ places the scalar symbol value into the j-th coordinate of $\mathbb{R}^H$, it follows that $\boldsymbol{P}_{h,1}$ contains, in an H-dimensional value space, the H cyclic shifts

$$\left[\boldsymbol{P}^{(h-1)H} \boldsymbol{X}, \quad \boldsymbol{P}^{(h-1)H+1} \boldsymbol{X}, \quad \dots, \quad \boldsymbol{P}^{(h-1)H+(H-1)} \boldsymbol{X}\right]$$

Let $\boldsymbol{R} \in \mathbb{R}^{H \times HP}$ be the collapse matrix in the new-IHA definition. Choose $\boldsymbol{R}$ to select only pseudo $j = 1$ from each head:

$$\boldsymbol{R}_{h,(h-1)H+1} = 1, \qquad \boldsymbol{R}_{h,(h-1)H+j} = 0 \; \forall j \in \{2, \ldots, H\}, \qquad \boldsymbol{R}_{h,(h'-1)H+j} = 0 \; \forall h' \neq h.$$

Then the per-head output is $\boldsymbol{O}_h = \boldsymbol{P}_{h,1}$.

Representation after Layer 1. Let $\hat{\boldsymbol{X}}^{(1)}$ denote the representation after the new-IHA layer, obtained by concatenating heads:

$$\hat{\boldsymbol{X}}^{(1)} \coloneqq [\, \boldsymbol{O}_1, \boldsymbol{O}_2, \ldots, \boldsymbol{O}_H \,] \in \mathbb{R}^{N_{\max} \times H^2}.$$

By the expression for $\boldsymbol{O}_h = \boldsymbol{P}_{h,1}$, $\hat{\boldsymbol{X}}^{(1)}$ contains *all* cyclic shifts $\{\boldsymbol{P}^t \boldsymbol{X}\}_{t=0}^{H^2-1}$ arranged in a fixed, known indexing across its H^2 coordinates. In particular, for every token position i, the vector $\hat{\boldsymbol{X}}^{(1)}[i, :]$ provides access to the entire multiset of symbols $\{x_1, \ldots, x_{N_{\max}}\}$ (with their cyclic order), within a single vector space.

MLP. Note that $\hat{\boldsymbol{X}}^{(1)}$ allows each token to access all $N_{\max}$ symbols within a single (known) coordinate system. Hence, the **MLP** can be constructed as follows. The first linear layer is chosen to enumerate all ordered pairs (j_1, j_2) (i.e., all ${}^{N_{\max}}P_2 = N_{\max}(N_{\max} - 1)$ permutations, or $N_{\max}^2$ pairs if allowing $j_1 = j_2$), and for each such pair it forms the quantity

$$x_i + G x_{j_1} + x_{j_2}.$$

This can be implemented by a linear map whose hidden width scales with the number of pairs, with weights that (i) select x_{j_1} and x_{j_2} from $\hat{\boldsymbol{X}}^{(1)}[i, :]$ and (ii) multiply the selected x_{j_1} by G while leaving x_{j_2} unscaled. Furthermore, if the activation after this first layer is defined as $f(\cdot) = \mathrm{ReLU}\big(1 - \phi(\cdot)\big)$ where $\phi(\cdot)$ is the modulo-M operation, then the activation produces a nonzero output if and only if

$$x_i + G x_{j_1} + x_{j_2} \equiv 0 \pmod{M}.$$

Finally, the second linear layer is responsible solely for aggregating (summing) these indicators across all ordered pairs (j_1, j_2), thereby producing $\mathrm{CPM}_i(3)$ at each token position i.

Total Parameter Count. Considering all parameters described above, the total parameter count (for the new-IHA construction) denoted by $T_{\textbf{IHA}}$ can be upper bounded by the sum of: (i) base query/key/value projections, (ii) pseudo mixing coefficients, (iii) the collapse matrix $\boldsymbol{R}$, and (iv) the MLP parameters. Concretely, with $H = \lceil \sqrt{N_{\max}} \rceil$ and $P = H$, the attention-layer parameters satisfy

$$\begin{aligned}
(\text{Query} + \text{Key}) &= 2 \cdot H \cdot (N_{\max} + 1) N_{\max}, \\
\text{Value} &= H \cdot (N_{\max} + 1) \cdot H = (N_{\max} + 1) H^2, \\
\text{Pseudo Mix Coeffs} &= 3 \cdot H \cdot H \cdot P = 3H^3, \\
\text{Collapse } (\boldsymbol{R}) &= H \cdot (HP) = H^3.
\end{aligned}$$

For the MLP, the first layer requires width proportional to the number of ordered pairs, i.e. ${}^{N_{\max}}P_2$, and thus contributes on the order of $H^2 \cdot N_{\max}(N_{\max} - 1)$ parameters (up to constant factors depending on the exact hidden width choice), while the second layer aggregates these counts and contributes at most $N_{\max}^2$ parameters. Putting these together,

$$\begin{aligned}
T_{\textbf{IHA}} \;&=\; (\text{Query} + \text{Key}) + \text{Value} + \text{MLP} + \text{Pseudo Mix Coeffs} + \text{Collapse} \\
&\leq\; 2(N_{\max}^2 + N_{\max})H \;+\; (N_{\max} + 1)H^2 \;+\; H^2 N_{\max}(N_{\max} - 1) \;+\; N_{\max}^2 \;+\; 4H^3. \\
&\leq\; 2(N_{\max}^2 + N_{\max})\lceil \sqrt{N_{\max}} \rceil \;+\; (N_{\max} + 1)\lceil \sqrt{N_{\max}} \rceil^2 \\
&\quad +\; \lceil \sqrt{N_{\max}} \rceil^2 N_{\max}(N_{\max} - 1) \;+\; N_{\max}^2 \;+\; 4\lceil \sqrt{N_{\max}} \rceil^3 \\
&\leq\; 37 N_{\max}^{2.5} + N_{\max}^2(N_{\max} - 1) + N_{\max}^2
\end{aligned}$$

Substituting $H = \lceil \sqrt{N_{\max}} \rceil$ yields a polynomial bound in $N_{\max}$ (with the same dominant scaling coming from the MLP term), completing the construction.

Example $N_{\max} = 4$). We solidify the intuition behind the construction with a concrete instance. Let the input tokens be $1, 2, 3, 4$ and $N_{\max} = 4$. Then the encoder with positional encoding produce

$$\hat{\boldsymbol{X}} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 2 & 0 & 1 & 0 & 0 \\ 3 & 0 & 0 & 1 & 0 \\ 4 & 0 & 0 & 0 & 1 \end{bmatrix} \in \mathbb{R}^{4\times 5}.$$

Set $H \coloneqq \lceil \sqrt{N_{\max}} \rceil = 2$ and $P \coloneqq H = 2$. Let $\boldsymbol{P} \in \mathbb{R}^{4\times 4}$ be the cyclic permutation matrix (shifting down by one):

$$\boldsymbol{P} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \qquad \boldsymbol{e}_1 = [1, 0, 0, 0, 0]^\top \in \mathbb{R}^5.$$

Let $\boldsymbol{u}_1 = [1, 0]^\top$ and $\boldsymbol{u}_2 = [0, 1]^\top$ denote the standard basis of $\mathbb{R}^2$.

Query, Keys and Values. We define the query and key matrices for $m \in \{1, 2\}$:

$$\boldsymbol{W}_{Q,\texttt{IHA}}^{(1,m)} = \begin{bmatrix} \boldsymbol{0}_{1\times N_{\max}} \\ \boldsymbol{P}^{(m-1)H} \end{bmatrix}, \qquad \boldsymbol{W}_K^{(1,m)} = \begin{bmatrix} \boldsymbol{0}_{1\times N_{\max}} \\ \left(\boldsymbol{P}^{m-1}\right)^\top \end{bmatrix},$$

Hence, explicitly,

$$\boldsymbol{W}_Q^{(1,1)} = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{I} \end{bmatrix}, \qquad \boldsymbol{W}_Q^{(1,2)} = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{P}^2 \end{bmatrix},$$

$$\boldsymbol{W}_K^{(1,1)} = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{I} \end{bmatrix}, \qquad \boldsymbol{W}_K^{(1,2)} = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{P}^\top \end{bmatrix},$$

and

$$\boldsymbol{W}_V^{(1,1)} = 2\,\boldsymbol{e}_1\,\boldsymbol{u}_1^\top = \begin{bmatrix} 2 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, \qquad \boldsymbol{W}_{V,\texttt{IHA}}^{(1,2)} = 2\,\boldsymbol{e}_1\,\boldsymbol{u}_2^\top = \begin{bmatrix} 0 & 2 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}.$$

Pseudo-head mixing. In **IHA**, each head $h \in \{1, 2\}$ forms pseudos $j \in \{1, 2\}$ via $\alpha^Q, \alpha^K, \alpha^V \in \mathbb{R}^{H\times H\times P}$. We choose one-hot mixing:

$$\alpha^Q_{m,h,j} \coloneqq \mathbb{1}[m = h]\mathbb{1}[j = 1], \qquad \alpha^K_{m,h,j} \coloneqq \mathbb{1}[m = j], \qquad \alpha^V_{m,h,j} \coloneqq \mathbb{1}[m = j].$$

Therefore, for each head h,

$$\widetilde{\boldsymbol{Q}}_{h,1} = \hat{\boldsymbol{X}}\boldsymbol{W}_Q^{(1,h)}, \quad \widetilde{\boldsymbol{Q}}_{h,2} = \boldsymbol{0}, \qquad \widetilde{\boldsymbol{K}}_{h,1} = \hat{\boldsymbol{X}}\boldsymbol{W}_K^{(1,1)}, \quad \widetilde{\boldsymbol{K}}_{h,2} = \hat{\boldsymbol{X}}\boldsymbol{W}_K^{(1,2)},$$
$$\widetilde{\boldsymbol{V}}_{h,1} = \hat{\boldsymbol{X}}\boldsymbol{W}_V^{(1,1)}, \quad \widetilde{\boldsymbol{V}}_{h,2} = \hat{\boldsymbol{X}}\boldsymbol{W}_V^{(1,2)}.$$

Stacking and attention (per head). New-IHA stacks pseudos row-wise:

$$\overline{\boldsymbol{Q}}_h = \begin{bmatrix} \widetilde{\boldsymbol{Q}}_{h,1} \\ \widetilde{\boldsymbol{Q}}_{h,2} \end{bmatrix}, \quad \overline{\boldsymbol{K}}_h = \begin{bmatrix} \widetilde{\boldsymbol{K}}_{h,1} \\ \widetilde{\boldsymbol{K}}_{h,2} \end{bmatrix}, \quad \overline{\boldsymbol{V}}_h = \begin{bmatrix} \widetilde{\boldsymbol{V}}_{h,1} \\ \widetilde{\boldsymbol{V}}_{h,2} \end{bmatrix}.$$

Let $\overline{\boldsymbol{P}}_h = \mathrm{softmax}(\overline{\boldsymbol{Q}}_h \overline{\boldsymbol{K}}_h^\top)\overline{\boldsymbol{V}}_h$. If the softmax temperature is set to 0, the attention reduces to hard attention induced by the permutation structure.

Head $h = 1$. Since $\widetilde{\boldsymbol{Q}}_{1,1} = \hat{\boldsymbol{X}}\boldsymbol{W}_Q^{(1,1)}$ and the two pseudo-keys correspond to $\boldsymbol{I}$ and $\boldsymbol{P}$, the first-pseudo output of head 1 (denoted $\boldsymbol{P}_{1,1}$) takes the form

$$\boldsymbol{P}_{1,1} = 2\,\boldsymbol{I} \cdot \left(\hat{\boldsymbol{X}}\boldsymbol{W}_V^{(1,1)}\right) \;+\; 2\,\boldsymbol{P} \cdot \left(\hat{\boldsymbol{X}}\boldsymbol{W}_V^{(1,2)}\right).$$

Compute the value projections:

$$\hat{\boldsymbol{X}}\boldsymbol{W}_V^{(1,1)} = \begin{bmatrix} 2 & 0 \\ 4 & 0 \\ 6 & 0 \\ 8 & 0 \end{bmatrix} \qquad \hat{\boldsymbol{X}}\boldsymbol{W}_V^{(1,2)} = \begin{bmatrix} 0 & 2 \\ 0 & 4 \\ 0 & 6 \\ 0 & 8 \end{bmatrix}.$$

Thus,

$$\boldsymbol{P}_{1,1} = \frac{1}{2}\boldsymbol{I} \cdot \begin{bmatrix} 2 & 0 \\ 4 & 0 \\ 6 & 0 \\ 8\text{‘} & 0 \end{bmatrix} + \frac{1}{2}\boldsymbol{P} \cdot \begin{bmatrix} 0 & \frac{1}{2} \\ 0 & 1 \\ 0 & \frac{3}{2} \\ 0 & 2 \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 2 & 3 \\ 3 & 4 \\ 4 & 1 \end{bmatrix}.$$

Head $h = 2$. Here $\widetilde{\boldsymbol{Q}}_{2,1} = \hat{\boldsymbol{X}}\boldsymbol{W}_Q^{(1,2)}$, so the induced shifts are $\boldsymbol{P}^2$ and $\boldsymbol{P}^3$. Hence

$$\boldsymbol{P}_{2,1} = \frac{1}{2}\,\boldsymbol{P}^2 \cdot \left(\hat{\boldsymbol{X}}\boldsymbol{W}_V^{(1,1)}\right) \;+\; \frac{1}{2}\,\boldsymbol{P}^3 \cdot \left(\hat{\boldsymbol{X}}\boldsymbol{W}_V^{(1,2)}\right) = \begin{bmatrix} 3 & 4 \\ 4 & 1 \\ 1 & 2 \\ 2 & 3 \end{bmatrix}.$$

Collapse $HP \to H$ (select pseudo $j = 1$). New-IHA uses $\boldsymbol{R} \in \mathbb{R}^{H \times HP}$. Choose $\boldsymbol{R}$ to pick only pseudo $j = 1$ from each head:

$$\boldsymbol{R}_{1,1} = 1, \quad \boldsymbol{R}_{2,3} = 1, \quad \text{and all other entries are } 0,$$

so that $\boldsymbol{O}_1 = \boldsymbol{P}_{1,1}$ and $\boldsymbol{O}_2 = \boldsymbol{P}_{2,1}$.

Concatenate heads. The representation after the attention layer is

$$\hat{\boldsymbol{X}}^{(1)} = [\boldsymbol{O}_1, \boldsymbol{O}_2] = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 2 & 3 & 4 & 1 \\ 3 & 4 & 1 & 2 \\ 4 & 1 & 2 & 3 \end{bmatrix}.$$

Thus each token position now contains all symbols in a fixed, known order (a cyclic listing), which allows the subsequent MLP to enumerate ordered pairs (j_1, j_2), form $x_i + Gx_{j_1} + x_{j_2}$, apply the mod-M test, and aggregate counts to compute $\mathrm{CPM}_i(3)$.

MHA. We now proceed with the **MHA** construction.

Encoder and Positional Encoding: The input to the network is a sequence of tokens where each token is a natural number. Concretely its defined as: $\{x_i\}_{i \in \mathbb{N}} \in \mathbb{N}$. We assume the encoder and the positional encoding to be defined such that they output the following:

$$\hat{\boldsymbol{X}} = [\,\boldsymbol{X},\ \boldsymbol{I}\,],$$

where I is the identity matrix used for positional encoding, and X integer value of of the input symbol.

$$\boldsymbol{X} \in \mathbb{R}^{N_{\max} \times 1}, \qquad \boldsymbol{I} \in \mathbb{R}^{N_{\max} \times N_{\max}}, \qquad \hat{\boldsymbol{X}} \in \mathbb{R}^{N_{\max} \times (N_{\max}+1)}.$$

Layer 1. (Attention) We provide the sketch of what the intended outcome of the first layer is via an example and then proceed with the construction.

Main Idea. Similar to the prior construction, the goal of the first layer of attention is to able to permute all input symbols to obtain all the symbols in a sequence within a single vector space. For example, given an input sequence 1 2 3, we obtain the following:

Token 1: 1 2 3
Token 2: 2 3 2
Token 3: 3 1 2.

This allows us to obtain all tokens that can then be used by the MLP to obtain all the permutations needed to solve the **CPM-3** task. We first define the construction (the query, key and value matrices) in generality below and then describe an example below to make things concrete. Note that in the equation below, $\boldsymbol{P} \in \mathbb{R}^{N_{\max} \times N_{\max}}$ denotes a cyclic permutation matrix and $\boldsymbol{e}_1 = [1, \mathbf{0}_{1 \times N_{\max}}]^T$. We first define the query matrices below.

$$\boldsymbol{W}_{Q,\texttt{MHA}}^{(1,h)} = \begin{bmatrix} \mathbf{0}_{1 \times N_{\max}} \\ \boldsymbol{P}^{(h-1)} \end{bmatrix}$$

where, $h \in \{1, 2, \cdots, N_{\max}\}$. We now define the key matrices below.

$$\boldsymbol{W}_{K,\texttt{MHA}}^{(1,h)} = \begin{bmatrix} \mathbf{0}_{1 \times N_{\max}} \\ \boldsymbol{I}_{N_{\max} \times N_{\max}} \end{bmatrix},$$

where, $h \in \{1, 2, \cdots, N_{\max}\}$. We now define the value matrices below.

$$\boldsymbol{W}_{V,\texttt{IHA}}^{(1,h)} = [\boldsymbol{e_1}],$$

where, $h \in \{1, 2, 3, \cdots, N_{\max}\}$.

Hence, basis this construction, if the temperature of softmax is set to 0 (leading to hard attention), one can obtain the following (Note that $\cdot||\cdot$ denotes the concatenation operation), and $\hat{X}^{(1)}$ denote the representation after the first layer of **IHA**,

$$\hat{X}^{(1)} = \hat{\boldsymbol{X}}\boldsymbol{W}_{V,\texttt{IHA}}^{(1,1)} \,||\, \boldsymbol{P}\hat{\boldsymbol{X}}\boldsymbol{W}_{V,\texttt{IHA}}^{(1,2)} || \,\cdots\, ||\, \boldsymbol{P}^{(N_{\max}-1)}\hat{\boldsymbol{X}}\boldsymbol{W}_{V,\texttt{IHA}}^{(1,N_{\max})}$$

MLP. Note that $\hat{X}^{(1)}$ allows for having all the $N_{\max}$ tokens in the same vector space. Then the **MLP** can be constructed as follows. The first layer can contain $\lceil N_{\max} \rceil^2 \times (N_{\max})(N_{\max} - 1)$ parameters where the first layer via the MLP is responsible for all ${}^{N_{\max}}P_2$ permutations along with multiplications with the right token with G. Furthermore, if the activation function after the first layer is defined as $f(\cdot) = \text{ReLU}(1 - \phi(\cdot))$ where $\phi(\cdot)$ the modulo-M operation, then the second linear layer is responsible solely for aggregating all the resulting counts. This aggregation can be implemented by a linear layer whose parameter matrix has dimension $\max^2$.

Total Parameter Count. Considering all the parameters described above, the total parameter count denoted by $T_{\textbf{MHA}}$ is as follows:

$$\begin{aligned} T_{\textbf{MHA}} &= (\text{Query Parameters} + \text{Key Parameters}) + \text{Value Parameters} + \text{MLP Parameters} \\ &= 2N_{\max}^3 + N_{\max}^3 + (N_{\max}+1)(N_{\max})(N_{\max}-1) + N_{\max}^2 \\ &> 3N_{\max}^3 + (N_{\max})^2(N_{\max}-1) + N_{\max}^2 \end{aligned}$$

□

E Computational Efficiency of IHA

This section provides additional details on the compute and memory requirements of IHA. We first describe the cost of global IHA, then explain the FLOP-matched local-global schedule used in our experiments. We then report practical training-throughput measurements and discuss KV-cache behavior during autoregressive decoding.

E.1 Global IHA complexity

Let N denote the sequence length, H the number of attention heads, $d = D/H$ the per-head dimension, and P the number of pseudo-heads per head. In global IHA, each token is expanded into P pseudo-tokens per head. Thus, interleaving increases the effective sequence length from N to NP. A global IHA layer therefore has per-head attention cost

$$\mathcal{O}\big((NP)^2 d\big) = \mathcal{O}\big(P^2 N^2 d\big),$$

which is a factor-P^2 increase over global MHA. Therefore, in our main comparisons, we use FLOP-matched IHA so that improvements cannot be attributed simply to additional attention compute.

E.2 FLOP-matched local-global schedule

To preserve the benefits of cross-head mixing while controlling compute, we use a hybrid local-global schedule. In local IHA layers, we use sliding-window attention with window size

$$W := \frac{N}{2P^2}.$$

A sliding-window IHA layer attends with NP query pseudo-tokens and WP key pseudo-tokens. Its cost is

$$\begin{aligned}\mathcal{O}\big(H \cdot (NP) \cdot (WP) \cdot d\big) &= \mathcal{O}\big(HNWP^2 d\big) \\ &= \mathcal{O}\left(HN \cdot \frac{N}{2P^2} \cdot P^2 d\right) \\ &= \mathcal{O}\big(HN^2 d/2\big).\end{aligned}$$

Thus, each local IHA layer costs approximately same as that of a global MHA layer, upto constant terms.

In our FLOP-matched setting, we use a 4:1 local-global schedule: four local IHA layers followed by one global attention layer. Averaging over this five-layer block gives

$$\frac{4 \cdot \mathcal{O}(HN^2 d/2) + \mathcal{O}(HN^2 d)}{5} = \frac{3}{5}\mathcal{O}(HN^2 d),$$

which matches the global-attention baseline up to constant factors. This schedule lets us study the effect of cross-head mixing without giving IHA substantially more attention compute.

E.3 Training throughput

We profile the practical overhead of IHA on our 2.4B model. We compare standard MHA with FLOP-matched IHA at two sequence lengths and two IHA layer fractions. Here, Frac denotes the fraction of attention layers replaced by IHA, N is the sequence length, and WSL is the sliding-window size used in local IHA layers. We report per-step training latency and training throughput, together with the IHA/MHA ratio for each metric.

Table 3: Training efficiency of FLOP-matched IHA on the 2.4B model. We report per-step training latency and throughput. Ratios compare IHA to standard MHA under the same setting.

Frac	N	WSL	Train MHA	Train IHA	Ratio	Tput MHA	Tput IHA	Ratio
50%	8192	1023	537 ms	572 ms	1.07×	15,268	14,311	0.94×
50%	16384	2049	1,248 ms	1,261 ms	1.01×	13,125	12,996	0.99×
80%	8192	1023	537 ms	592 ms	1.10×	15,250	13,832	0.91×
80%	16384	2049	1,240 ms	1,267 ms	1.02×	13,211	12,928	0.98×

To support IHA efficiently, we use a FlashAttention-compatible implementation (see Subsec. E.4) in which the P pseudo-head variants are packed into the sequence dimension and processed as pseudo-tokens. Under the FLOP-matched setting, local IHA layers use a reduced sliding-window size so that the expanded pseudo-token attention has comparable compute to the MHA baseline.

Table 3 shows that the practical overhead is modest and decreases with sequence length. At $N = 8192$, IHA incurs a 1.07–$1.10\times$ training-latency overhead and achieves 0.91–$0.94\times$ the throughput of MHA. At $N = 16384$, the overhead narrows to 1.01–$1.02\times$ latency, with throughput within 1–2% of MHA. This suggests that most of the overhead comes from pseudo-head mixing and pseudo-token construction rather than the attention computation itself, and that this overhead is increasingly amortized at longer sequence lengths. Even when 80% of attention layers are replaced by IHA, the overhead remains small under matched settings.

E.4 KV-cache behavior during decoding

IHA changes the attention computation, but it does not require storing pseudo-head keys and values in the KV cache. During autoregressive decoding, IHA caches the original per-head key and value

projections, just as MHA does. The pseudo-head mixtures are computed on the fly from the cached representations using the learned mixing coefficients and are not stored.

Let t denote the number of decoded tokens so far. Standard MHA stores

$$2 \times H \times t \times d$$

scalars in the KV cache, where the factor of 2 accounts for keys and values. Global IHA stores the same number of scalars, since it caches only the original per-head key and value projections.

In the FLOP-matched local IHA layers used in our experiments, the sliding-window size is $W = N/(2P^2)$. Thus, the local-layer KV cache stores

$$2 \times H \times W \times d$$

scalars. This can be smaller than the global MHA cache when local attention is used.

The decoding procedure differs only in the attention computation. MHA attends directly over the cached keys and values. IHA first forms pseudo-head mixtures from the cached keys and values via the learned mixing weights, and then attends over these mixed pseudo-head representations. Since the cache itself contains the same original per-head key and value projections as MHA, standard KV-cache optimizations such as quantization, token eviction, and compression can be applied to IHA without modification at the cache level.

Overall, global IHA has a factor-P^2 attention-cost increase over global MHA, but our FLOP-matched local-global schedule controls this cost by using sliding-window IHA in most IHA layers. In practice, the measured training overhead is modest on the 2.4B model and becomes very small at longer sequence lengths. During decoding, IHA does not increase the KV-cache size relative to MHA because it caches original per-head keys and values, not pseudo-head mixtures.

E.5 IHA-CUDA: Hopper Kernels for Interleaved Head Attention

We implement GPU kernels for Interleaved Head Attention (IHA) on NVIDIA Hopper GPUs, targeting the SM90 architecture. The goal is to make IHA practical at long context lengths by adapting the FlashAttention-3 execution pattern rather than implementing attention with generic PyTorch operations. IHA differs from standard multi-head attention in two steps. First, before attention, it mixes information across heads to form P pseudo-heads for each original head. Second, attention is computed over these pseudo-heads, with a shared softmax across the P pseudo-head group.

Our attention kernel builds on FlashAttention-3 using the CuTe domain-specific language (DSL). In standard FlashAttention-3, the sequence is divided into tiles, and each cooperative thread array (CTA), i.e., each GPU thread block, is responsible for a block of query tokens while iterating over the corresponding key-value tiles. A naive IHA implementation would treat each pseudo-query block separately, causing many CTAs to repeat similar key-value tile traversals. This is inefficient because IHA introduces an additional pseudo-head dimension that should be handled inside the kernel schedule rather than by launching many independent thread blocks. Our IHA kernel instead uses a nested traversal over query pseudo-tokens and key-value tiles. This reduces redundant CTA launches while preserving the key FlashAttention-3 optimizations: tiled matrix multiplication, on-chip reuse of key-value blocks, and numerically stable online softmax.

Before the attention kernel is called, IHA constructs pseudo-token representations using a head-mixing operation. For each projection $X \in \{Q, K, V\}$, where Q, K, and V denote queries, keys, and values, respectively, the premix operation is

$$Y_{h,p} = \sum_s W_{h,p,s} X_s, \qquad W \in \mathbb{R}^{H \times P \times H}.$$

Here, H is the number of original attention heads, P is the number of pseudo-heads per head, s indexes the source head, and (h, p) indexes the output pseudo-head. This operation is applied independently at every batch element, sequence position, and head-feature coordinate. In other words, the same small mixing matrix is used across all tokens to produce pseudo-query, pseudo-key, and pseudo-value tensors.

Although this premix operation is algebraically just a small linear map over heads, it is performance-sensitive. The mixing weights W are tiny compared with the activation tensors, while the input and

output tensors scale with batch size, sequence length, number of heads, and head dimension. As a result, the operation is memory-bound: performance is dominated by reading and writing activations rather than by arithmetic. We therefore implement the premix step using custom Triton kernels instead of `torch.einsum`. The forward pass launches one fused premix kernel each for Q, K, and V. Each kernel tiles over the sequence dimension and unrolls the small head loops at compile time. When rotary positional embeddings (RoPE) are used, the rotation can also be fused into the premix kernel to avoid an additional memory pass. For training, we also implement custom backward kernels for the premix step. A naive implementation would launch separate backward kernels for Q, K, and V, and separately compute gradients with respect to both the input activations and the mixing weights. Instead, we fuse the backward computation across all three projections into two Triton kernels:

$$dX = W^\top dY,$$
$$dW = \sum_{b,n,d} dY X^\top.$$

The first kernel computes gradients with respect to the input activations by applying the transpose of the premix map. The second computes gradients with respect to the mixing weights by reducing over batch index b, sequence position n, and head dimension d. This reduces the premix backward pass from six kernel launches to two, lowering launch overhead and improving training efficiency.

The premix stage is the main source of additional cost in IHA relative to standard multi-head attention. Its cost scales as

$$\text{cost}_{\text{premix}} = O(H^2 PN),$$
$$\text{cost}_{\text{attn}} = O(N^2 H),$$
$$\frac{\text{premix}}{\text{attention}} \propto \frac{HP}{N}.$$

Thus, the premix cost grows linearly with sequence length N, while the attention computation grows quadratically with N. Consequently, the relative overhead of constructing pseudo-heads decreases at longer context lengths. This is why IHA-CUDA is especially effective in long-context regimes: the additional head-mixing cost is increasingly amortized by the dominant attention computation.

Overall, IHA-CUDA integrates Interleaved Head Attention into Hopper-optimized FlashAttention-3 kernels with modest training overhead. The implementation combines CuTe DSL attention kernels for the main attention computation with fused Triton kernels for pseudo-head construction. This design preserves much of the throughput of optimized multi-head attention baselines while enabling richer within-layer attention interactions.

F Effect of the Number of Pseudo-Heads

We study how the number of pseudo-heads P affects IHA in controlled synthetic settings. In particular, we probe the tradeoff between within-layer cross-head interaction, controlled by P, and compositional capacity from depth. Standard MHA can build richer head interactions through stacking layers, whereas IHA introduces learned cross-head mixing within each layer. Varying both P and depth therefore tests whether pseudo-heads improve fixed-depth expressivity and reduce the depth required for compositional tasks. Note that large-scale sweeps over P and depth are expensive at the 2.4B parameter scale used in our main experiments. We therefore complement the large-scale results with targeted synthetic tasks that isolate compositional structure. These experiments are intended as diagnostics of the inductive bias introduced by cross-head mixing, rather than as a full efficiency study.

F.1 Boolean relation composition

Following [16], we consider boolean relation-composition tasks. These tasks test whether a model can infer composed relations from an input adjacency matrix. The input is a random boolean matrix

$$R \in \{0,1\}^{m \times m},$$

where $R_{ij} = 1$ indicates a direct relation from entity i to entity j. We flatten R into a sequence of length m^2. The model predicts, for each pair (i, j), whether a composed relation holds. This gives a position-wise binary classification problem trained with binary cross-entropy over all valid positions.

Binary (2-hop) Task: For the 2-hop task, the target is $R \circ R$:

$$(R \circ R)_{ij} = 1 \iff \exists k \text{ such that } R_{ik} = 1 \wedge R_{kj} = 1.$$

Thus, the model must decide whether there exists an intermediate entity k witnessing a two-step path from i to j. We sample m uniformly from $\{6, \ldots, 10\}$, so the sequence length ranges from 36 to 100, and sample entries of R i.i.d. from $\mathrm{Bernoulli}(0.325)$ to keep the target labels approximately balanced.

Ternary (3-hop) Task: For the 3-hop task, the target is $R \circ R \circ R$:

$$(R \circ R \circ R)_{ij} = 1 \iff \exists k, \ell \text{ such that } R_{ik} = 1 \wedge R_{k\ell} = 1 \wedge R_{\ell j} = 1.$$

This task is harder because the model must reason through two latent intermediate indices rather than one. We sample m uniformly from $\{5, \ldots, 8\}$, so the sequence length ranges from 25 to 64, and sample entries of R i.i.d. from $\mathrm{Bernoulli}(0.264)$ to avoid saturation of the 3-hop target.

These tasks are useful diagnostics because success requires composing multiple relational interactions, rather than relying on a single local dependency. They therefore provide a simple setting for testing whether cross-head mixing helps models combine intermediate relational structure.

All models use model dimension 128, learning rate 10^{-3}, four attention heads, and matched depth. We compare standard MHA against pseudo-IHA while varying $P \in \{2, 4, 8\}$. The case $P = 1$ corresponds to standard MHA.

Table 4: Effect of the number of pseudo-heads P on boolean relation-composition tasks. We report mean accuracy and standard deviation over three seeds. Δ denotes the improvement over **Standard MHA** with the same depth.

Layers	Task	Standard MHA	Δ	IHA ($P = 2$)	Δ	IHA ($P = 4$)	Δ	IHA ($P = 8$)	Δ
1	**Binary (2-hop)**	70.97 ± 0.07	–	74.61 ± 0.21	+3.64	75.50 ± 0.22	+4.53	76.00 ± 0.45	+5.04
1	**Ternary (3-hop)**	76.96 ± 0.03	–	80.63 ± 0.16	+3.67	80.89 ± 0.09	+3.93	81.20 ± 0.16	+4.24
2	**Binary (2-hop)**	79.81 ± 1.23	–	97.09 ± 1.93	+17.28	**99.94** ± 0.04	+20.13	99.79 ± 0.09	+19.98
2	**Ternary (3-hop)**	82.57 ± 0.40	–	86.33 ± 0.49	+3.76	**89.08** ± 1.62	+6.51	88.64 ± 0.45	+6.06
3	**Binary (2-hop)**	99.62 ± 0.27	–	99.96 ± 0.07	+0.34	**100.00** ± 0.00	+0.38	**100.00** ± 0.00	+0.38
3	**Ternary (3-hop)**	88.27 ± 0.19	–	98.96 ± 0.37	+10.69	98.52 ± 1.94	+10.25	**99.13** ± 1.31	+10.86

Table 4 shows that pseudo-IHA improves over standard MHA for every tested value of P. At one layer, performance improves as P increases on both tasks. At two layers, all $P > 1$ variants substantially outperform standard MHA, with $P = 4$ achieving the strongest mean performance. At three layers, the 2-hop task is nearly saturated for all pseudo-IHA variants, while IHA continues to give large gains on the harder 3-hop task.

These results suggest that cross-head mixing is especially useful when the model must compose multiple latent relations before producing an output. Increasing P beyond 1 is consistently beneficial in this setting, although the best value depends on task difficulty and depth. In our experiments, setting $P = H$, where H is the number of attention heads, is a robust default.

F.2 Atomic sequence-manipulation tasks

We also evaluate IHA on continuous-valued atomic sequence-manipulation tasks following the setup of Gupta et al. [13]. These tasks probe basic sequence-modeling capabilities such as aggregation, reversal, and context-dependent selection. Since these are regression tasks, we report test R^2, where higher is better and a score near zero corresponds to performance comparable to predicting the mean target.

Each example is generated synthetically. Unless otherwise noted, we sample $\tilde{x} \in \mathbb{R}^L$ with entries $\tilde{x}_i \sim \mathcal{N}(0, 1)$ and normalize by the ℓ_∞ norm, $x = \tilde{x}/\|\tilde{x}\|_\infty$. We define the different tasks below

`cumsum`**:** Following [13], the target is the normalized prefix sum $y_i = (i + 1)^{-1/2} \sum_{j \le i} x_j$.

`reverse`**:** The target is the reversed sequence. Following [13], we pad the input and target to length $2L$, with $x_{\text{in}} = [x, 0^L]$ and $y_{\text{target}} = [0^L, \text{reverse}(x)]$. We evaluate only the final L positions.

Table 5: Atomic sequence-manipulation tasks. We report test R^2. Δ denotes the improvement over **Standard MHA**. IHA substantially improves `cumsum`, while giving little improvement on `reverse` and `select`.

Task	Standard MHA	Δ	IHA ($P = 2$)	Δ	IHA ($P = 4$)	Δ
`cumsum`	0.30 ± 0.01	–	0.99 ± 0.01	+0.69	**1.00** ± 0.00	+0.70
`reverse`	**0.01** ± 0.01	–	0.00 ± 0.00	–0.01	**0.01** ± 0.01	–
`select`	0.01 ± 0.02	–	0.01 ± 0.02	–	**0.02** ± 0.00	+0.01

`select`: Following [13], we sample a real-valued sequence $x \in \mathbb{R}^{L+M}$ and independently sample a subset of $M = 32$ source indices $i_1 < \cdots < i_M$ uniformly without replacement from $\{1, \ldots, L+M\}$. The input has two channels and length $L + 2M$. The first channel contains the values $x_1, \ldots, x_{L+M}$ followed by M zero-valued output slots. The second channel is a binary mask that marks the selected source positions, i.e., it is one at positions $i_1, \ldots, i_M$ and zero elsewhere. The target is zero on the first $L + M$ positions and contains the selected values in order on the final M positions: $y_{\text{target}} = [0^{L+M}, x_{i_1}, \ldots, x_{i_M}]$. We evaluate only the final M output positions.

Each model has approximately 207K parameters, one attention layer, model dimension 128, four attention heads, and learning rate 10^{-3}. Models are trained on 200K examples and evaluated on 5K validation and 5K test examples. Results are averaged over two random seeds.

Tab. 5 provide suggestive evidence for when cross-head mixing is most useful. IHA substantially improves `cumsum`, while producing much smaller changes on `reverse` and `select`. This is consistent with the view that `cumsum` requires progressive aggregation across positions, whereas `reverse` and `select` are closer to positional routing or retrieval. Thus, IHA appears most helpful in this setting for aggregation-style computations, although a complete mechanistic explanation is left for future work.

Overall, these controlled experiments suggest that IHA provides the clearest gains on tasks requiring aggregation or composition of intermediate structure. The boolean relation-composition tasks probe latent relational reasoning through $R \circ R$ and $R \circ R \circ R$, while the atomic tasks provide a simpler diagnostic of sequence-level aggregation. Together, they support the view that the pseudo-head parameter P controls a useful form of cross-head interaction, with $P = H$ serving as a practical default in our experiments.

G Additional controlled pretraining experiments

We evaluate IHA at two additional model scales to study its dependence on model size, pseudo-head count, and layer placement. All architectures use global attention throughout, allowing us to evaluate IHA without differences in the local–global attention schedule. Wider MHA controls further test whether additional feed-forward capacity reproduces its gains.

G.1 Experimental setup and evaluation

The models contain approximately 239M and 365M parameters, with $H = 6$ and $H = 8$ attention heads, respectively. We train them on 2.82B and 4.68B tokens, corresponding to approximately 20 tokens per non-embedding parameter of the MHA baseline. At each scale, all variants share the training data, random seeds, tokenizer, optimizer configuration, learning-rate schedule, and token budget. We compare MHA, Differential Transformer, Talking-Heads Attention, IHA, and MHA-wide, which increases the feed-forward hidden dimension. We sweep $P \in \{2, 3, 4, 6, 8, 12\}$ at 239M and $P \in \{2, 4, 8, 10\}$ at 365M. These comparisons match training-token budgets and closely match parameter counts, rather than training FLOPs or wall-clock time. All variants use standard RoPE with base $\theta = 500{,}000$. IHA applies RoPE to the PN interleaved token–pseudo-head positions, assigning distinct rotary phases to pseudo-heads associated with the same original token.

Evaluation. We measure corpus-level token perplexity on eight domains as outlined in the main paper, while also evaluating the gold solutions from the full GSM8k test set and MATH test set. Note that these perplexities measure mathematical-solution modeling independently of generation sampling and complement the post-SFT answer-accuracy evaluations. Every architecture uses the same evaluation procedure. Lower perplexity is better in every domain. We summarize performance

Table 6: **Global-attention results at 239M parameters.** Models use $H = 6$ heads and 2.82B training tokens. Lower is better in every column; bold indicates the best result.

	Text and code perplexity ↓						Solution perplexity ↓		
Model	WikiText 103	C4	DCLM	RedPajama	Dolma v1.5	Dolma Code	GSM8K	MATH	Avg. rank ↓
MHA	45.17	55.28	44.88	23.22	35.34	9.00	15.63	9.33	8.88
MHA-wide	44.43	55.37	44.89	23.12	35.39	9.02	15.83	9.41	9.50
Differential	44.34	54.99	44.71	23.16	35.26	9.13	15.53	9.36	8.38
Talking-Heads	43.07	53.32	43.36	22.22	33.91	8.60	**14.63**	9.07	5.00
IHA, $P = 2$	42.28	53.34	43.34	22.12	33.94	8.60	15.25	9.15	5.75
IHA, $P = 3$	42.21	53.21	43.22	22.15	33.89	8.65	15.76	9.22	6.00
IHA, $P = 4$	41.91	53.00	43.05	21.98	33.66	8.50	15.24	9.08	2.75
IHA, $P = 6 = H$	**41.55**	**52.77**	**42.92**	**21.93**	**33.57**	**8.48**	14.72	**9.05**	**1.125**
IHA, $P = 8$	41.93	52.97	43.08	22.08	33.75	8.62	15.42	9.18	4.38
IHA, $P = 12$	42.24	52.90	43.05	22.05	33.75	8.58	14.90	9.07	3.25

Table 7: **Global-attention results at 365M parameters.** Models use $H = 8$ heads and 4.68B training tokens. Lower is better in every column; bold indicates the best result.

	Text and code perplexity ↓						Solution perplexity ↓		
Model	WikiText 103	C4	DCLM	RedPajama	Dolma v1.5	Dolma Code	GSM8K	MATH	Avg. rank ↓
MHA	31.03	41.27	33.72	17.34	26.43	6.76	11.90	7.32	7.62
MHA-wide	31.01	41.26	33.73	17.37	26.39	6.74	12.03	7.22	7.38
Differential	29.93	40.19	32.92	16.76	25.57	6.47	11.26	7.07	5.88
Talking-Heads	29.23	39.54	32.33	16.50	25.19	6.45	10.99	7.02	3.75
IHA, $P = 2$	29.03	39.62	32.42	16.64	25.32	6.54	10.66	7.03	4.75
IHA, $P = 4$	**28.79**	39.34	32.23	16.56	25.16	6.47	10.21	6.96	2.75
IHA, $P = 8 = H$	28.92	**39.20**	**32.09**	**16.47**	**25.05**	**6.44**	**10.04**	**6.95**	**1.125**
IHA, $P = 10$	29.03	39.23	32.12	16.52	25.09	6.46	10.48	7.00	2.75

using average rank (denoted as AvgRank), where rank 1 denotes the lowest perplexity among the models in each comparison. AvgRank is computed separately within each table.

G.2 Model scale and pseudo-head count

Tables 6 and 7 show that $P = H$ achieves the best AvgRank, 1.125, at both scales and improves over MHA and MHA-wide on all eight domains. It attains the lowest perplexity in seven domains at each scale: Talking-Heads performs best on GSM8K at 239M, while IHA with $P = 4$ performs best on WikiText-103 at 365M. Thus, IHA's improvements persist at both additional scales with global attention in every layer.

Smaller pseudo-head counts are also effective: $P = 4 < H$ outperforms every non-IHA baseline in AvgRank at both scales. Increasing P beyond H provides no consistent further benefit. These sweeps support $P = H$ as a strong default in the tested settings, with smaller P reducing the pseudo-head expansion.

G.3 Parameter overhead and wider-MHA controls

IHA retains the query, key, value, and output projections and adds $4H^2P$ mixing and collapse parameters per replaced layer. With $P = H$, the additions are 5,184 parameters at 239M and 16,384 at 365M. MHA-wide instead adds 442,368 and 786,432 parameters through feed-forward width,

Table 8: **Parameter budgets.** Total parameters and additions relative to MHA at each scale. IHA adds at most 0.0056% parameters, while MHA-wide increases feed-forward width.

239M ($H = 6$)

Model	Total (M)	Added params.	Added (%)
MHA	239.469	0	0.0000
MHA-wide	239.911	442,368	0.1847
Differential	239.473	—	0.0019
Talking-Heads	239.469	—	0.0002
IHA, $P = 2$	239.470	1,728	0.0007
IHA, $P = 3$	239.471	2,592	0.0011
IHA, $P = 4$	239.472	3,456	0.0014
IHA, $P = 6 = H$	239.474	5,184	0.0022
IHA, $P = 8$	239.475	6,912	0.0029
IHA, $P = 12$	239.479	10,368	0.0043

365M ($H = 8$)

Model	Total (M)	Added params.	Added (%)
MHA	365.429	0	0.0000
MHA-wide	366.215	786,432	0.2152
Differential	365.435	—	0.0017
Talking-Heads	365.430	—	0.0003
IHA, $P = 2$	365.433	4,096	0.0011
IHA, $P = 4$	365.437	8,192	0.0022
IHA, $P = 8 = H$	365.445	16,384	0.0045
IHA, $P = 10$	365.449	20,480	0.0056

approximately $85.3\times$ and $48\times$ the corresponding IHA additions. Across the full sweeps in Table 8, IHA increases the total parameter count by at most 0.0056%.

Despite its larger parameter budget, MHA-wide does not reproduce IHA's gains. Increasing P beyond H likewise adds parameters without improving overall performance. These controls suggest that the gains are not explained by the small increase in parameter count alone.

For the original 2.4B configuration, $H = P = 20$ gives 32,000 additional parameters per replaced layer. Replacing all 26 layers would add 832,000 parameters, less than 0.035% of the model; the actual addition under the 4:1 local-IHA/global-attention schedule is smaller.

H Layer-placement ablation

We replace different subsets of attention layers in the 239M model with IHA at $P = 4$, retaining MHA elsewhere. All variants share the training data, random seed, optimizer configuration, learning-rate schedule, and 2.82B-token budget, with global attention in every layer. The all-layer variant is the $P = 4$ model in Table 6.

Table 9 shows that all-layer IHA achieves the best AvgRank and the lowest perplexity on seven of eight domains. Among partial replacements, alternating layers performs best overall and achieves the lowest GSM8K perplexity. The first-and-last configuration ranks next, outperforming both adjacent two-layer configurations despite using the same number of IHA layers.

Replacing only the first layer improves over MHA on all eight domains, whereas replacing only the last layer worsens all eight. However, replacing the first two layers is less effective overall than replacing the first alone. At this scale, the results favor coverage across depth over concentrating IHA in adjacent layers, with an isolated early layer more useful than an isolated late layer.

I Analysis of learned head-mixing tensors

We analyze the learned query, key, and value mixing tensors across both pseudo-head sweeps. For a model with L layers, $x \in \{q, k, v\}$, and layer ℓ, let $\alpha_\ell^x \in \mathbb{R}^{H\times H\times P}$ and $M_{\ell,p}^x = \alpha_\ell^x[:, :, p]$. Entry $M_{\ell,p}^x[m, h]$ gives the contribution of source head m to mixed head slot h. Every slice is initialized to I_H.

Metrics. We measure the relative strength of cross-head mixing using

$$C_x = \frac{1}{L}\sum_{\ell=1}^{L}\frac{1}{P}\sum_{p=1}^{P}\frac{\|\operatorname{offdiag}(M_{\ell,p}^x)\|_F}{\|\operatorname{diag}(M_{\ell,p}^x)\|_F},$$

Table 9: **Layer placement at 239M parameters ($P = 4$).** Unreplaced layers use MHA; all layers retain global attention. Bold indicates the lowest perplexity and average rank. Parameter counts appear below.

	Text and code perplexity ↓						Solution perplexity ↓		
IHA layers	WikiText 103	C4	DCLM	RedPajama	Dolma v1.5	Dolma Code	GSM8K	MATH	Avg. rank ↓
None (MHA)	45.17	55.28	44.88	23.22	35.34	9.00	15.63	9.33	6.62
First layer	43.54	54.30	44.11	22.53	34.52	8.69	15.26	9.16	3.62
Last layer	46.33	55.62	45.12	23.45	35.81	9.22	16.15	9.55	8.00
First two	44.23	54.17	44.02	22.71	34.66	8.93	15.02	9.25	4.25
Last two	44.37	55.10	44.66	23.05	35.19	9.02	15.96	9.42	6.38
First and last	42.74	53.92	43.81	22.48	34.42	8.79	15.47	9.27	3.12
Every other	43.12	53.91	43.80	22.55	34.43	8.80	**14.92**	9.23	2.75
All layers	**41.91**	**53.00**	**43.05**	**21.98**	**33.66**	**8.50**	15.24	**9.08**	**1.25**

IHA layers	Total (M)	Added (%)	IHA layers	Total (M)	Added (%)
None (MHA)	239.469	0.0000	Last two	239.470	0.0005
First layer	239.469	0.0002	First and last	239.470	0.0005
Last layer	239.469	0.0002	Every other	239.470	0.0007
First two	239.470	0.0005	All layers	239.472	0.0014

Table 10: **Learned head-mixing structure.** Cross/self ratios and pseudo-head spreads are defined in Eqs. equation **??**–equation **??** and averaged across layers.

	Cross/self ratio C_x			Pseudo-head spread S_x		
P	Query	Key	Value	Query	Key	Value
239M ($H = 6$)						
2	0.214	0.220	0.101	0.21	0.21	0.33
3	0.241	0.237	0.099	0.20	0.21	0.30
4	0.279	0.226	0.104	0.19	0.20	0.28
$6 = H$	0.246	0.167	0.098	0.16	0.14	0.24
8	0.256	0.157	0.098	0.17	0.13	0.22
12	0.251	0.140	0.092	0.15	0.12	0.20
365M ($H = 8$)						
2	0.273	0.241	0.183	0.27	0.22	0.46
4	0.332	0.271	0.166	0.25	0.25	0.36
$8 = H$	0.339	0.215	0.172	0.23	0.19	0.30
10	0.332	0.216	0.181	0.22	0.19	0.28

where $\mathrm{diag}(M)$ retains the diagonal entries and $\mathrm{offdiag}(M) = M - \mathrm{diag}(M)$. To measure variation among pseudo-head slices, we define

$$\overline{M}^x_\ell = \frac{1}{P}\sum_{p=1}^{P} M^x_{\ell,p}, \qquad S_x = \frac{1}{L}\sum_{\ell=1}^{L} \frac{\frac{1}{P}\sum_{p=1}^{P} \|M^x_{\ell,p} - \overline{M}^x_\ell\|_F}{\|\overline{M}^x_\ell\|_F}.$$

Both statistics are averaged over layers; zero spread corresponds to identical pseudo-head slices within each layer.

Table 10 shows that all three tensors develop nonzero off-diagonal structure. At $P = H$, queries have the strongest cross-head mixing, followed by keys and values, at both scales. Values nevertheless have the greatest pseudo-head spread: their mixing is more same-head dominated, but their slices differ more from the mean. Across the sweeps, query cross/self ratios remain substantial, while key

ratios peak at small or intermediate P and value ratios vary less within each scale. Pseudo-head spread remains nonzero and generally decreases as P grows. The learned slices therefore retain distinct weight structures throughout the sweep. These statistics are consistent with stronger mixing in the query/key components that determine attention routing; identifying the functional roles of individual slices requires targeted probing and causal ablations.

J Broader Impact

IHA increases the within-layer compositional capacity of Transformer models and may improve performance on long-context retrieval, mathematical reasoning, and multi-hop reasoning tasks. These improvements could support beneficial applications such as scientific and legal document analysis, code assistance, educational tutoring, and workflows that require synthesizing information across long or complex inputs. At the same time, stronger reasoning and long-context capabilities can amplify risks already associated with language models. Models using IHA may be more effective at generating persuasive misinformation, assisting harmful automated workflows, or extracting sensitive information from long documents if deployed without appropriate safeguards. These risks are not specific to IHA, but capability-improving architectural changes can contribute to them. IHA also raises compute and access considerations. Although our FLOP-matched experiments show modest overhead, training large language models remains resource-intensive. Future work should study whether IHA can reduce the number of layers, heads, or parameters needed for a target capability, which could improve efficiency and broaden access. Before deployment in high-stakes settings, IHA-equipped models should be evaluated for safety, privacy, robustness, hallucination, memorization, and bias, especially in long-context use cases involving sensitive or proprietary data.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The claims in the abstract and introduction accurately reflect our contributions and scope. We support the architectural claims in Section 4 and Appendix E, the empirical claims in Section 5 and Appendix F, and the limitations in the conclusion and appendix.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: We discuss limitations in the conclusion, including the increased cost of global IHA and the use of a sliding-window schedule to mitigate this cost. We provide additional computational efficiency and scaling details in Appendix F.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

Answer: [Yes]

Justification: We state the assumptions for the main theoretical results in Section 4, including the one-layer setting, linear-attention variants for polynomial filters, and assumptions on sequence length, head count, and dimensions. We include proof sketches in the main text and complete proofs in Appendix E.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

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

Answer: [Yes]

Justification: We specify the architecture, attention variants, training setup, optimizer, data scale, context lengths, evaluation benchmarks, and FLOP-matching procedure in Section 5 and Appendix F. These details provide a reasonable path to reproduce the main experimental comparisons.

Guidelines:

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

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

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

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

   Answer: [No].

   Justification: We cite the fine-tuning data and evaluation benchmarks, but do not currently release code or full reproduction scripts.

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: Section 5 gives model, optimizer, training scale, hardware, fine-tuning data, benchmarks, and decoding settings. Appendix F gives additional efficiency details.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: For the main 2.4B-scale experiments, repeated runs are computationally expensive, so we report matched single-run benchmark results. For the synthetic experiments, we report standard deviations across runs.

   Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: We specify the main training hardware, including FSDP over 128 H200 GPUs, along with the model scale, token budget, and training schedule in Section 5. We provide additional throughput and efficiency measurements in Appendix F.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: We have reviewed the NeurIPS Code of Ethics and the research conforms to it. Our work proposes and evaluates a model architecture using standard public benchmarks and does not involve human-subject experiments, deceptive studies, or release of high-risk data.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: We discuss potential positive impacts, including improved long-context reasoning, document analysis, code assistance, education, and scientific workflows. We also discuss potential negative impacts, including misuse, misinformation, privacy risks, compute costs, and the need for safety evaluation before deployment.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [N/A]

    Justification: We do not release a high-risk pretrained model, image generator, or scraped dataset as part of this work. The paper introduces an architecture and reports experiments, and the broader-impact discussion recommends safety, privacy, robustness, memorization, and bias evaluations before future deployment.

    Guidelines:

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

12. **Licenses for existing assets**

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

Answer: [Yes]

Justification: We cite the datasets, benchmarks, tokenizer, and prior methods used in our experiments, including RULER, OpenThoughts, GSM8K, MATH-500, MBPP, HumanEval, and Llama 3. We use these assets consistently with their intended research use and credit the original creators.

Guidelines:

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

13. **New assets**

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

Answer: [N/A]

Justification: We do not introduce or release a new dataset, benchmark, pretrained model, or other standalone asset. The main contribution is a model architecture together with theoretical and empirical evaluation.

Guidelines:

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

14. **Crowdsourcing and research with human subjects**

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

Answer: [N/A]

Justification: The work does not involve crowdsourcing experiments or research with human subjects. All reported evaluations use existing benchmarks and model-training experiments.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

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

    Answer: [N/A]

    Justification: The work does not involve human-subject research or crowdsourcing, so IRB or equivalent approval is not applicable.

    Guidelines:

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

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: ChatGPT and Claude Code were used for writing, coding assistance, debugging, and auxiliary proof sanity checks. All generated or suggested code was reviewed, tested, and validated by the authors, and all methodological decisions, proofs, experimental design, and reported results were determined and verified by the authors.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.