How Does *m*HC Use Its Residual Streams? Selective Routing and Near-Identity Mixing

Anonymous authors
Paper under double-blind review

Abstract

Hyper-Connections and their manifold-constrained variant (*m*HC) widen a residual pathway from one stream to n, yet how trained models use this capacity remains unclear: how broadly blocks read and write, how strongly the residual pathway mixes streams, and whether the streams carry distinct representations. We examine these properties in the four-stream residual pathway of DeepSeek-V4-Flash using effective stream counts, cross-stream residual weights, and inter-stream cosine similarity. Read/write routing is concentrated but varies across depth: a typical attention or FFN site effectively uses about two streams, while the dominant stream changes across layers and the representations remain directionally distinct. Residual mixing is modest and occurs primarily in early layers; in layers 22–42, the pathway mostly carries each stream forward separately. Targeted interventions establish the functional significance of these patterns. Replacing the late mixers by identity increases C4 perplexity by only 1.9% and preserves the six-task average score, whereas replacing the early mixers increases perplexity by 41%. Fixing each early mixer to its C4 diagnostic mean increases perplexity by only 0.2% and reduces the average score by 0.25 percentage points, showing that its site-specific structure matters more than its token-wise variation on the evaluated metrics. Likewise, retaining the three largest routing weights per token at every site increases perplexity by at most 2.7% and changes the average score by at most 0.4 points. Thus, the studied model realizes only part of the flexibility afforded by four-stream *m*HC: individual blocks rarely require all four streams, and late residual mixing provides little measured benefit.

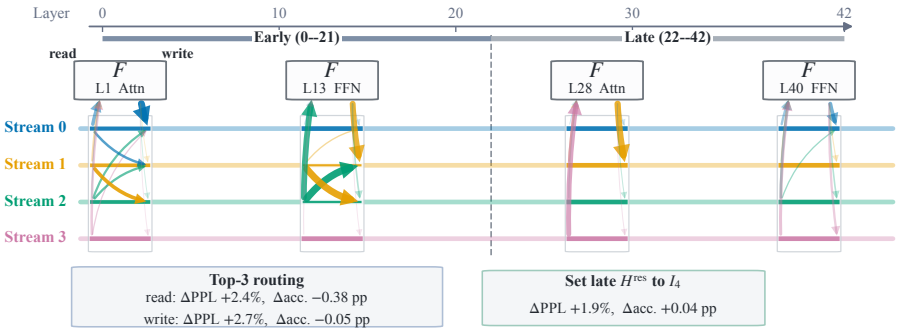


Figure 1: **Realized multi-stream computation in DeepSeek-V4-Flash.** At four sites spanning depth and both sublayer types, color identifies the source stream, arrows show the direction of information flow, and line width encodes the measured token-averaged routing or residual weight. Routing concentrates on changing stream subsets, whereas residual exchange largely disappears after mid-depth; the callouts give the exact intervention effects on C4 perplexity and the six-task average score.

1 Introduction

Hyper-Connections (HC) (Zhu et al., 2024) generalize the standard single-stream residual pathway (He et al., 2016) to n coupled streams. At each sublayer, a read map aggregates the streams into a single branch input, a write map distributes the branch output back across the streams, and a

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

residual mixer linearly recombines the incoming stream states along the skip pathway. Manifold-constrained Hyper-Connections (*m*HC) (Xie et al., 2025) are designed to improve training stability by constraining each residual mixer to be doubly stochastic, thereby limiting signal amplification as these mixers compose across depth. This design widens the state propagated across depth while retaining a single computation branch at each sublayer.

However, *m*HC makes broad routing and cross-stream mixing possible, while training determines how and where the resulting model uses this flexibility. Because the read, write, and residual maps are token-conditioned and learned at each layer, their realized behavior can vary across both tokens and depth. Accordingly, the read and write maps may distribute branch computation broadly or concentrate it on a subset of streams; the residual pathway may mix the stream states or propagate them separately; and the streams may encode distinct or redundant representations.

Recent work on 120M- and 360M-parameter HC models built with nanoGPT (Karpathy, 2022) finds a dominant-stream regime: read/write signals and interpretable features concentrate in one stream, while residual mixing remains close to identity (Alimaskina et al., 2026). Because model scale and training regime may shape how the learned maps allocate computation across streams, it remains unclear whether this regime is characteristic of trained *m*HC models or only one of several possible organizations of multi-stream computation. Moreover, these structural patterns do not by themselves establish post-training functional redundancy. Determining this requires testing whether weak routing weights can be pruned and whether near-identity residual mixers can be replaced by exact identity maps without degrading model quality.

We address *both gaps* with a true-forward analysis of DeepSeek-V4-Flash, a 284B-parameter *m*HC model pretrained on more than 32T tokens (DeepSeek-AI, 2026). Our framework measures effective read/write width, cross-stream residual weights, and inter-stream representation similarity across tokens, sublayers, and depth. These depth-resolved diagnostics distinguish a single globally dominant stream from shifting stream use across layers and determine whether the streams retain distinct representations. We pair them with targeted inference-time interventions to connect the observed routing and mixing patterns to model quality.

Applying this framework reveals concentrated but depth-varying routing: a typical attention or FFN site effectively uses about two of the four streams, while the dominant stream changes across layers and the stream representations remain directionally distinct. Routing ablations show that this concentration creates local redundancy but does not reduce the pathway to two streams. Retaining the three largest realized read or write weights for every token at every site increases perplexity on C4 (Raffel et al., 2020) by at most 2.7% and changes the average downstream score by at most 0.4 percentage points; retaining only two weights per token instead increases perplexity by 12–14% and reduces the average downstream score by 2.9–6.6 percentage points.

Residual mixing also varies with depth. Cross-stream residual weights are modest overall and most pronounced in layers 0–21 of the 43-layer model, whereas the residual mixers in layers 22–42 remain close to the identity. Residual-mixer ablations confirm that this depth profile has functional significance. Replacing the mixers in layers 22–42 by exact identity maps increases C4 perplexity by only 1.9% and preserves the average score across six downstream benchmarks, whereas replacing those in layers 0–21 increases perplexity by 41%. Yet fixing each early mixer to its C4 diagnostic mean raises perplexity by only 0.2% and reduces the six-task average score by 0.25 points, indicating that its learned site-specific structure accounts for most of its measured benefit, while token-wise variation contributes little on the evaluated metrics. Figure 1 provides a data-driven overview of this analysis, tracing realized routing and residual exchange across representative depths and linking both to their measured intervention effects.

In summary, our contributions are threefold:

- **True-forward analysis of multi-stream computation.** We introduce a framework that resolves read/write routing, residual mixing, and stream representations across tokens, sublayers, and depth, and connects these measurements to model quality through inference-time interventions.
- **A depth-resolved characterization of DeepSeek-V4-Flash.** We show that its four streams exhibit locally concentrated but depth-varying routing, retain distinct representations, and exchange information primarily in the first half of the network.

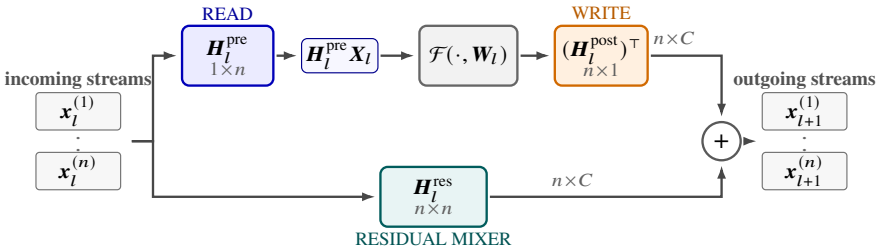


Figure 2: Data flow through HC residual sublayer l. The read map $\boldsymbol{H}_l^{\text{pre}}$ forms the computation-branch input, the write map $\boldsymbol{H}_l^{\text{post}}$ distributes the branch output across streams, and the residual mixer $\boldsymbol{H}_l^{\text{res}}$ carries forward and linearly recombines the incoming stream states.

- **Functional localization of routing and mixing capacity.** Controlled interventions identify where additional routing weights and learned residual mixing affect model quality, revealing local redundancy in read/write maps and limited functional dependence on late-layer residual mixing.

2 Background and Related Work

2.1 Residual Streams and Hyper-Connections

Consider a residual sublayer with hidden dimension C. A standard residual architecture propagates a single state, whereas Hyper-Connections (HC) maintain n coupled states $\boldsymbol{X}_l \in \mathbb{R}^{n \times C}$ at sublayer l (Zhu et al., 2024). Here and below, l indexes residual sublayers, including the attention and FFN sublayers within each Transformer layer; we suppress the token index for clarity. Each sublayer applies three maps to this state: a read map $\boldsymbol{H}_l^{\text{pre}} \in \mathbb{R}^{1 \times n}$, a write map $\boldsymbol{H}_l^{\text{post}} \in \mathbb{R}^{1 \times n}$, and a residual mixer $\boldsymbol{H}_l^{\text{res}} \in \mathbb{R}^{n \times n}$. In HC, each map combines learned static parameters with a token-dependent component. We refer to the resulting value for a given token and sublayer as the *realized* map. Figure 2 summarizes the resulting data flow.

Let $\mathcal{F}(\cdot, \boldsymbol{W}_l)$ denote the computation performed by the sublayer. Its multi-stream update is

$$\boldsymbol{X}_{l+1} = \boldsymbol{H}_l^{\text{res}} \boldsymbol{X}_l + (\boldsymbol{H}_l^{\text{post}})^\top \mathcal{F}(\boldsymbol{H}_l^{\text{pre}} \boldsymbol{X}_l, \boldsymbol{W}_l), \tag{1}$$

where $\boldsymbol{H}_l^{\text{pre}} \boldsymbol{X}_l$ forms the input to the computation branch, $(\boldsymbol{H}_l^{\text{post}})^\top \mathcal{F}(\boldsymbol{H}_l^{\text{pre}} \boldsymbol{X}_l, \boldsymbol{W}_l)$ distributes its output across the streams, and $\boldsymbol{H}_l^{\text{res}} \boldsymbol{X}_l$ carries forward and linearly recombines the incoming stream states.

2.2 Manifold-Constrained Hyper-Connections

Across successive residual sublayers, the residual mixers compose multiplicatively along the skip pathway. For sublayers $l, \ldots, L-1$, the resulting cumulative mixer is

$$\Pi_{L:l}^{\text{res}} := \boldsymbol{H}_{L-1}^{\text{res}} \cdots \boldsymbol{H}_l^{\text{res}}. \tag{2}$$

Without constraints on the individual mixers, the norm of this product can grow rapidly with depth, amplifying signals propagated along the residual pathway. To control this cumulative transformation, mHC constrains every realized residual mixer to the Birkhoff polytope (Xie et al., 2025):

$$\mathcal{B}_n := \left\{ \boldsymbol{H} \in \mathbb{R}^{n \times n} \;\middle|\; \boldsymbol{H}\mathbf{1}_n = \mathbf{1}_n,\ \mathbf{1}_n^\top \boldsymbol{H} = \mathbf{1}_n^\top,\ \boldsymbol{H} \geq 0 \right\}, \tag{3}$$

where $\mathbf{1}_n$ is the all-ones vector. In practice, mHC implements this projection using Sinkhorn–Knopp normalization (Sinkhorn & Knopp, 1967; Xie et al., 2025). Under the exact constraint, the Birkhoff polytope is closed under multiplication, and every $\boldsymbol{H} \in \mathcal{B}_n$ has spectral norm one. Consequently, the cumulative mixer $\Pi_{L:l}^{\text{res}}$ remains non-expansive, preventing signal amplification along the skip pathway. Although $\boldsymbol{I}_n$ is a valid doubly stochastic mixer, mHC learns $\boldsymbol{H}_l^{\text{res}}$ rather than fixing it to $\boldsymbol{I}_n$ to enable cross-stream information exchange (Xie et al., 2025).

2.3 Analysis of Learned Stream Utilization

The original HC and *m*HC studies establish the optimization and performance properties of multi-stream residual pathways (Zhu et al., 2024; Xie et al., 2025). Recent analysis of 120M- and 360M-parameter HC models trained with nanoGPT further identifies a dominant-stream regime in which read/write signals and interpretable features concentrate in one stream while the residual mixers remain close to identity (Karpathy, 2022; Alimaskina et al., 2026). Breaking initialization symmetry can mitigate this concentration (Alimaskina et al., 2026). These findings establish stream collapse as one possible organization of learned HC computation. Our study examines how routing, residual mixing, and stream representations are organized in a separately trained large-scale *m*HC model and tests the functional significance of these patterns through inference-time interventions.

3 Measuring Realized Multi-Stream Computation

3.1 Read/Write Routing Breadth

We characterize read/write routing at each sublayer from two complementary perspectives: whether tokens share the same dominant stream and how broadly the routing weights are distributed across streams. For diagnostic sequence m, let $h^q_{l,m,t,j}$ denote the realized routing weight assigned to stream j for its t-th token at sublayer l, where $q \in \{pre, post\}$ and $j \in \{1, \dots, n\}$. The realized read and write weights are nonnegative under the *m*HC parameterization, so we normalize them to obtain the token-level load distribution

$$p^q_{l,m,t,j} \coloneqq \frac{h^q_{l,m,t,j}}{\sum_{k=1}^{n} h^q_{l,m,t,k}}. \tag{4}$$

We summarize the resulting token-level distributions with two sublayer-level statistics.

Winner consistency. For each token, we identify the stream receiving the largest routing weight. For an input sequence m containing T_m tokens, its winner consistency is the largest fraction of tokens sharing the same winner:

$$c^q_{l,m} \coloneqq \max_{j \in \{1,\dots,n\}} \frac{1}{T_m} \sum_{t=1}^{T_m} \mathbb{I}\left[j = \arg\max_k p^q_{l,m,t,k} \right], \tag{5}$$

where $c^q_{l,m} \in [1/n, 1]$. We report $c^q_l = \frac{1}{M} \sum_{m=1}^{M} c^q_{l,m}$ over the M diagnostic sequences. A value of 1 means that all tokens within every sequence share the same winner, while lower values indicate that the dominant stream varies across tokens.

Load-effective stream count. We first aggregate the normalized routing weights within each sequence:

$$\bar{p}^q_{l,m,j} \coloneqq \frac{1}{T_m} \sum_{t=1}^{T_m} p^q_{l,m,t,j}. \tag{6}$$

The effective number of streams carrying this load is

$$n_{\text{eff}}(\bar{\boldsymbol{p}}^q_{l,m}) \coloneqq \frac{1}{\sum_{j=1}^{n} (\bar{p}^q_{l,m,j})^2} \in [1, n]. \tag{7}$$

The effective stream count equals 1 when all routing load falls on one stream and n when the load is uniform. As with winner consistency, we report this quantity averaged over the M diagnostic sequences.

3.2 Cross-Stream Residual Mixing

We quantify how strongly each residual sublayer mixes its incoming streams by measuring the deviation of its realized residual mixer from the identity. Let $\boldsymbol{H}^{\text{res}}_{l,t} \in \mathbb{R}^{n \times n}$ denote the realized residual mixer for token t at sublayer l. We define its identity deviation as

$$\delta(\boldsymbol{H}^{\text{res}}_{l,t}) \coloneqq \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} \left| (\boldsymbol{H}^{\text{res}}_{l,t})_{ij} - \mathbb{I}[i = j] \right|. \tag{8}$$

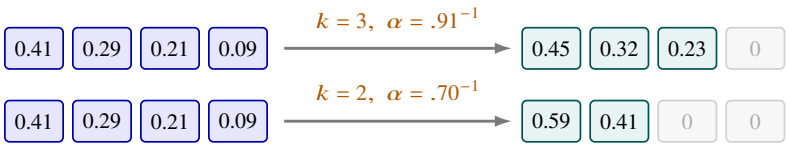


Figure 3: Scale-preserving top-k routing for $k \in \{2, 3\}$. Retained weights are rescaled by α to preserve their original sum.

The metric is zero when each stream is carried forward independently through the identity mixer. Under the exact doubly stochastic constraint, it is proportional to the total off-diagonal weight and therefore increases with cross-stream residual mixing. We average identity deviation across tokens within each sublayer and examine its profile across depth.

3.3 Inter-Stream Representation Similarity

We assess whether the residual streams carry distinct representations by measuring their pairwise alignment. Let $\boldsymbol{x}_{l,t}^{(s)} \in \mathbb{R}^C$ denote the hidden state carried by stream s for token t at sublayer l. We define the mean pairwise cosine similarity as

$$s_{l,t} \coloneqq \frac{2}{n(n-1)} \sum_{1 \le i < j \le n} \frac{\langle \boldsymbol{x}_{l,t}^{(i)}, \boldsymbol{x}_{l,t}^{(j)} \rangle}{\|\boldsymbol{x}_{l,t}^{(i)}\|_2 \|\boldsymbol{x}_{l,t}^{(j)}\|_2}. \tag{9}$$

Higher values indicate more closely aligned stream representations, whereas lower values indicate greater directional separation. We average this similarity across tokens within each sublayer and examine how it evolves across depth.

3.4 Inference-Time Interventions

The diagnostics above describe how the trained model uses its residual streams, but establishing which components are functionally necessary requires a counterfactual test: would model quality change if selected routing paths or cross-stream residual interactions were suppressed? We perform this test through two targeted inference-time interventions without retraining: routing sparsification and residual-mixer replacement.

For routing sparsification, we suppress the sequence index m and let $\mathcal{T}_{l,t}^{q}(k)$ index the k largest realized weights of the read or write map for token t at sublayer l. We define the scale-preserving top-k map as

$$\alpha_{l,t}^{q}(k) \coloneqq \frac{\sum_{r=1}^{n} h_{l,t,r}^{q}}{\sum_{r \in \mathcal{T}_{l,t}^{q}(k)} h_{l,t,r}^{q}}, \qquad \widetilde{h}_{l,t,j}^{q} \coloneqq \alpha_{l,t}^{q}(k)\, h_{l,t,j}^{q}\, \mathbb{I}\left[j \in \mathcal{T}_{l,t}^{q}(k) \right]. \tag{10}$$

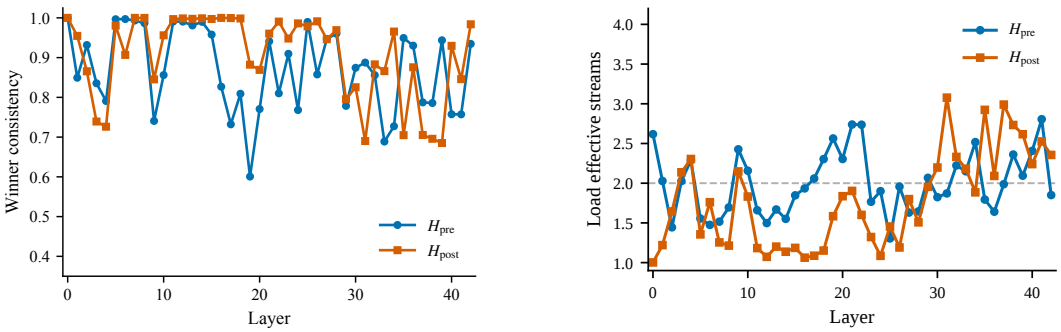
This intervention changes the routing support while preserving the total routing weight. Figure 3 illustrates the operation for $k \in \{2, 3\}$. We apply it to the read and write maps separately.

For residual-mixer replacement, we substitute $\boldsymbol{I}_n$ for $\boldsymbol{H}_{l,t}^{\text{res}}$ at every token within a selected depth range. This removes learned cross-stream exchange within that range while leaving the per-stream skip paths unchanged. We evaluate both interventions using perplexity and downstream-task scores.

4 Experiments

4.1 Experimental Setup

Model. We study the publicly released 0731 checkpoint of DeepSeek-V4-Flash (DeepSeek-AI, 2026), a 43-layer Mixture-of-Experts (MoE) language model with 284B total and 13B activated parameters, pretrained on more than 32T tokens. Its 4096-dimensional backbone combines hybrid attention and MoE feed-forward networks with four-stream *m*HC residual pathways. We instrument the attention and MoE-FFN residual sublayers in every layer, yielding 86 observation sites.



(a) Winner consistency.

(b) Load-effective stream count.

Figure 4: Depth profiles of routing concentration. Each point averages the attention and FFN sites within a layer. (a) Winner consistency measures the fraction of tokens sharing the most frequent dominant stream; its read/write averages are 0.871 and 0.905. (b) The load-effective stream count averages 1.998 and 1.775, respectively, indicating that routing typically spans only about two of the four streams. The dashed line in (b) marks two effective streams.

Data and evaluation. We use C4 (Raffel et al., 2020) following the sampling and evaluation protocols of GPTQ (Frantar et al., 2023). For the true-forward diagnostics, we randomly sample 512 sequences of 1,024 tokens from the first English C4 training shard. For perplexity, we follow the GPTQ evaluation protocol (Frantar et al., 2023), evaluating 256 contiguous sequences of 2,048 tokens constructed from the first English C4 validation shard. We additionally evaluate six downstream benchmarks in the zero-shot setting: ARC-Easy and ARC-Challenge (Clark et al., 2018), PIQA (Bisk et al., 2020), HellaSwag (Zellers et al., 2019), MMLU (Hendrycks et al., 2021), and GSM8K (Cobbe et al., 2021). We report normalized accuracy for ARC-Easy, ARC-Challenge, PIQA, and HellaSwag, accuracy for MMLU, and exact-match accuracy for GSM8K, together with the unweighted average of the six task scores.

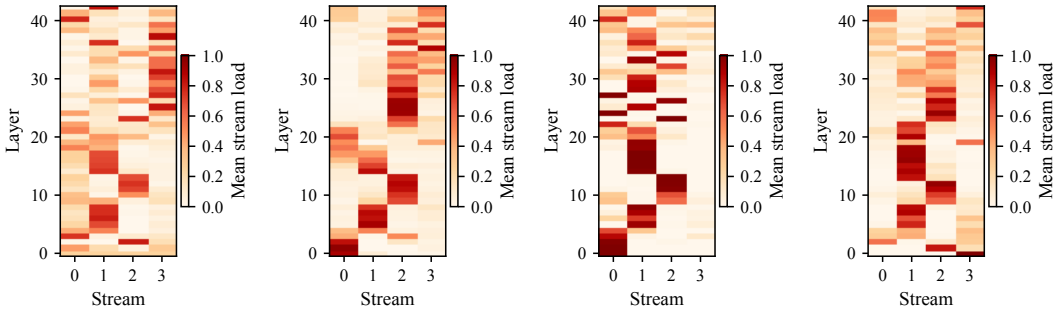
Implementation. We build our forward-pass instrumentation and inference-time interventions on PyTorch (Paszke et al., 2019) and Hugging Face Transformers (Wolf et al., 2020). Downstream evaluation uses the LM Evaluation Harness (Gao et al., 2023).

4.2 Organization of Read/Write Routing

We first characterize how the trained model routes branch computation across its four residual streams. Figure 4 shows that routing is both narrow and largely stable across tokens at a given sublayer. Winner consistency averages 0.871 for the read maps and 0.905 for the write maps, with medians of 0.931 and 0.988 across sublayers; thus, most maps assign a large majority of tokens within an input sequence to the same dominant stream. The corresponding load-effective stream counts average 1.998 and 1.775, with medians of 1.876 and 1.542. A typical sublayer therefore concentrates most of its routing load on roughly two streams, with stronger concentration in the write maps.

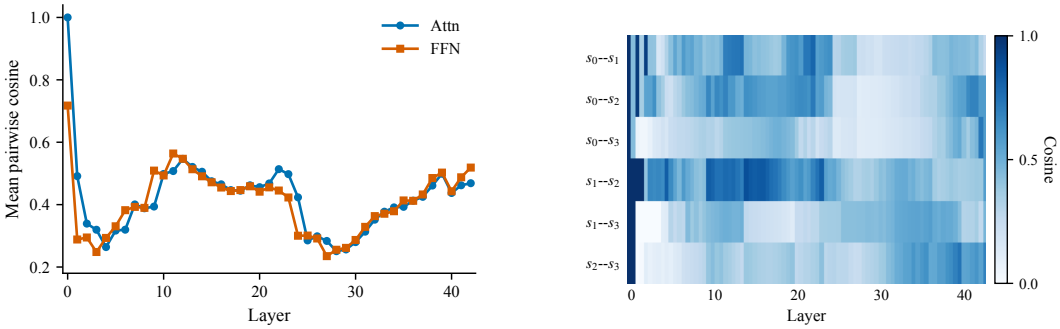
This local concentration does not reduce to a single stream dominating throughout the network. Figure 5 instead reveals depth-structured assignments: for a fixed map and sublayer type, 62.5% of adjacent layer pairs retain the same winner, producing contiguous intervals whose dominant stream changes at several depth boundaries. The reorganization is especially clear in the read maps. Streams 0 and 1 dominate 32 of the 44 sites in layers 0–21, whereas streams 2 and 3 dominate 36 of the 42 sites in layers 22–42. Every stream therefore becomes dominant in read or write routing somewhere in the network, but over different depth ranges.

Routing breadth also changes with depth, primarily through the write maps. Their mean effective stream count increases from 1.467 in layers 0–21 to 2.098 in layers 22–42, while winner consistency decreases from 0.940 to 0.870; read width remains nearly unchanged at 1.971 and 2.025, respectively. The read and write schedules are coupled but not identical: their winners differ at 38.4% of the 86 sites, so a sublayer need not write primarily to the stream from which it reads. Taken



(a) $\boldsymbol{H}^{\text{pre}}$ at attention sites. (b) $\boldsymbol{H}^{\text{pre}}$ at FFN sites. (c) $\boldsymbol{H}^{\text{post}}$ at attention sites. (d) $\boldsymbol{H}^{\text{post}}$ at FFN sites.

Figure 5: Mean normalized routing load by layer and stream, shown separately for the read and write maps at attention and FFN sites. High-load regions form contiguous depth intervals, but their stream assignments change across layers, sublayer types, and read/write maps.



(a) Mean over the six stream pairs. (b) Pair-resolved similarity.

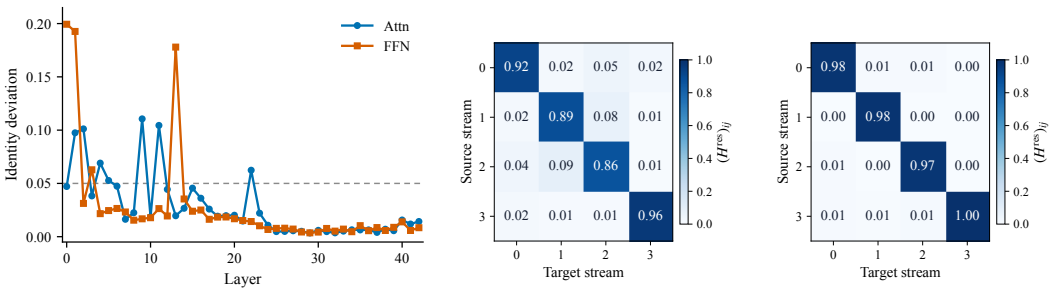
Figure 6: Inter-stream representation similarity across depth. (a) Pairwise cosine similarity, averaged over tokens and the six stream pairs at each attention and FFN site. (b) Token-averaged cosine similarity for each stream pair, with attention and FFN sites shown in forward-pass order within each layer. The streams begin from identical states but rapidly diverge, and no stream pair remains uniformly aligned throughout the network.

together, routing is locally narrow and largely token-stable, yet its allocation is reorganized across depth rather than collapsing onto one globally dominant stream. Appendix B.1 reports the routing statistics by depth range and sublayer type, while Appendix B.2 identifies the lowest-load stream at each site.

4.3 Inter-Stream Representation Geometry

Having established that routing is locally concentrated, we ask whether this concentration is accompanied by redundancy in the stream representations. Figure 6 tracks their pairwise cosine similarity across depth. The streams are identical at the first attention site, where their mean pairwise cosine is 1.0, but separate rapidly as the first layers transform and redistribute their states. From layer 1 onward, mean similarity ranges from 0.235 to 0.564 across attention and FFN sites, with an average of 0.404. Their directional separation is not monotonic: the streams partially realign around layers 10–13, separate more strongly around layers 27–29, and become progressively more aligned toward the output, without returning to their initial agreement.

The pair-resolved view further shows that this geometry is heterogeneous rather than symmetric. Beyond layer 0, pair-specific averages range from 0.288 for streams 0 and 3 to 0.560 for streams 1 and 2, and no pair remains uniformly aligned across depth. Thus, local routing concentration does not coincide with representation collapse: even weakly routed streams can carry directionally distinct states. However, representational distinctness does not establish functional importance; Section 4.5 tests whether these weak routing paths materially affect model quality.



(a) Identity deviation across depth. (b) Layers 0–21; $\delta = 0.046$. (c) Layers 22–42; $\delta = 0.009$.

Figure 7: Depth structure of realized residual mixing. (a) Identity deviation at each attention and FFN site, averaged over diagnostic tokens. (b–c) Residual mixers averaged over tokens and sites in the indicated depth ranges. Off-diagonal transfer is localized primarily to early layers, while the later mean mixer is nearly diagonal and performs little direct cross-stream exchange.

4.4 Depth Structure of Residual Mixing

We next examine how the strength of learned residual mixing varies across depth. Figure 7a reveals pronounced variation across the network: identity deviation averages 0.028 across all 86 sites, but the largest deviations occur at only a small number of early sites. After a final attention-side spike at layer 22, the mixers settle close to identity: 32 of the 40 sites in layers 23–42 have identity deviation below 0.01.

Figures 7b and 7c show how this depthwise reduction appears in the residual mixers themselves. Over layers 0–21, the mean mixer retains visible off-diagonal weights, with an identity deviation of 0.046. Over layers 22–42, these weights largely vanish and the deviation falls to 0.009, leaving a near-diagonal map that carries each stream forward with little direct residual exchange. The complete layerwise matrices for attention and FFN sites are provided in Appendix C (Figures 9 and 10).

The late mixers therefore realize little of their available cross-stream flexibility despite being learned and token-dependent by design. Their near-static identity structure motivates replacing them with $\boldsymbol{I}_4$ at inference, thereby avoiding the corresponding Sinkhorn–Knopp iterations; Section 4.5 evaluates the resulting effect on model quality.

4.5 Functional Effects of Routing and Mixing Interventions

The preceding diagnostics characterize realized stream use, but they do not determine which routing and mixing effects are necessary for model quality. We therefore apply the inference-time interventions from Section 3.4, without retraining, and measure their effects on C4 perplexity and downstream-task scores in Tables 1 and 2.

Table 1: C4 perplexity under the inference-time interventions. "Token mean" replaces each selected mixer by its average over the C4 diagnostic sequences.

Type	Action	Scope	PPL ↓	Δ PPL	Relative Δ
Baseline			11.1587	—	—
Routing	top-3	$\boldsymbol{H}^{\text{pre}}$	11.4220	+0.2633	+2.4%
		$\boldsymbol{H}^{\text{post}}$	11.4587	+0.3000	+2.7%
	top-2	$\boldsymbol{H}^{\text{pre}}$	12.5052	+1.3465	+12.1%
		$\boldsymbol{H}^{\text{post}}$	12.7206	+1.5619	+14.0%
Mixer	$\boldsymbol{H}^{\text{res}} \to \boldsymbol{I}_4$	22–42	11.3728	+0.2141	+1.9%
		0–21	15.7835	+4.6248	+41.4%
		All layers	15.8842	+4.7255	+42.3%
	Token mean	0–21	11.1832	+0.0245	+0.2%

Table 2: Zero-shot downstream performance under the inference-time interventions. All values are percentages; Δ Avg. is measured in percentage points.

Type	Action	Scope	ARC-E	ARC-C	PIQA	HellaSwag	MMLU	GSM8K	Avg.	Δ Avg.
Baseline			87.92	66.38	84.22	85.63	85.44	95.75	84.22	—
Routing	top-3	$\boldsymbol{H}^{\text{pre}}$	87.96	66.30	83.90	85.39	85.07	94.39	83.84	-0.38
		$\boldsymbol{H}^{\text{post}}$	87.84	66.98	84.22	85.85	85.37	94.77	84.17	-0.05
	top-2	$\boldsymbol{H}^{\text{pre}}$	85.02	62.54	82.81	83.96	81.66	91.81	81.30	-2.92
		$\boldsymbol{H}^{\text{post}}$	87.42	66.81	82.97	84.99	85.41	58.15	77.63	-6.59
Mixer	$\boldsymbol{H}^{\text{res}} \to \boldsymbol{I}_4$	22–42	88.05	67.32	84.28	85.81	85.41	94.69	84.26	+0.04
		All layers	85.56	62.71	83.30	82.98	79.90	91.21	80.94	-3.28
	Token mean	0–21	87.71	67.49	83.51	85.61	85.14	94.39	83.98	-0.25

Routing sparsification. Routing exhibits a clear threshold between removing the lowest-weight route for each token and retaining only two routes. Keeping the three largest realized read or write weights for every token at every site raises perplexity by only 2.4–2.7% and changes the average downstream score by at most 0.38 percentage points. Keeping only the two largest weights per token, however, raises perplexity by 12.1–14.0% and reduces the average downstream score by 2.92–6.59 points; for $\boldsymbol{H}^{\text{post}}$, the larger average drop is driven primarily by GSM8K. The lowest-weight routing path for each token is therefore largely dispensable under the evaluated interventions. Consistent with this redundancy being local rather than stream-wide, Appendix B.2 (Figure 8) shows that the stream with the lowest mean routing load changes across layers, maps, and sublayer types.

Residual-mixer replacement. The functional importance of learned residual mixing closely follows its depth profile. Replacing the near-identity mixers in layers 22–42 by exact identity maps raises perplexity by only 1.9% and changes the six-task average score by +0.04 percentage points. In contrast, applying the same replacement to layers 0–21 raises perplexity by 41.4%, nearly matching the 42.3% increase from replacing the mixers throughout the network. Fixing each early mixer to its site-specific mean over the C4 diagnostic sequences yields a perplexity of 11.1832, only 0.2% above the 11.1587 baseline, and changes the six-task average score from 84.22% to 83.98% (−0.25 percentage points). For this checkpoint and the evaluated metrics, the learned site-specific structure therefore accounts for most of the measured benefit of the early mixers, while their token-wise variation contributes little; the late mixers can instead be replaced by identity with little measured loss, eliminating their Sinkhorn–Knopp iterations at inference time.

Together, these interventions reveal underused flexibility in four-stream *m*HC, both within routing maps and across depth in residual mixers. This is not a wholesale collapse of multi-stream computation: the remaining routing support and the site-specific structure of the early mixers materially affect model quality. Appendix A discusses possible mechanisms and design implications.

5 Conclusion

We studied how a trained *m*HC model realizes the flexibility of its multi-stream residual pathway during inference. True-forward measurements of DeepSeek-V4-Flash reveal locally concentrated but depth-varying routing, directionally distinct stream representations, and residual exchange that occurs primarily in early layers. Inference-time interventions further show that the weakest routing weight per token at every site and the late residual mixers can be suppressed with little measured loss, whereas the remaining routing support and the learned structure of the early mixers are functionally important. Fixing the early mixers to their C4 diagnostic means also nearly preserves model quality, indicating that their site-specific structure contributes more than their token-wise variation on the evaluated metrics.

The four-stream pathway thus exhibits structured under-utilization rather than uniform participation or complete stream collapse. More broadly, architectural width does not determine effective multi-stream capacity; its realization depends on the component- and depth-specific organization learned during training. Our analysis identifies candidate mechanisms whose causal roles require training-time routing and mixer trajectories.

AI USE STATEMENT

In this work, we used generative AI tools to assist with programming and polish manuscript text. We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Ekaterina Alimaskina, Gleb Molodtsov, and Aleksandr Beznosikov. Analyzing stream collapse in Hyper-Connections: From diagnosis to mitigation, 2026. URL https://arxiv.org/abs/2606.03483.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. PIQA: Reasoning about physical commonsense in natural language. In *AAAI*, 2020. URL https://arxiv.org/abs/1911.11641.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Øyvind Tafjord. Think you have solved question answering? Try ARC, the AI2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018. URL https://arxiv.org/abs/1803.05457.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021. URL https://arxiv.org/abs/2110.14168.

DeepSeek-AI. DeepSeek-V4: Towards highly efficient Million-Token context intelligence, 2026. URL https://arxiv.org/abs/2606.19348.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. GPTQ: Accurate post-training quantization for generative pre-trained transformers. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2210.17323.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. A framework for few-shot language model evaluation, 2023. URL https://zenodo.org/records/10256836.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *ICLR*, 2021. URL https://arxiv.org/abs/2009.03300.

Andrej Karpathy. nanoGPT: The simplest, fastest repository for training and fine-tuning medium-sized GPTs. https://github.com/karpathy/nanoGPT, 2022. GitHub repository.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Qwen Team. On the design of Qwen3.8-Next architecture: Evaluation, efficiency, and training stability. Technical report, Alibaba Group, August 2026. URL https://arxiv.org/abs/2608.30320.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *JMLR*, 21(140):1–67, 2020.

Richard Sinkhorn and Paul Knopp. Concerning nonnegative matrices and doubly stochastic matrices. *Pacific Journal of Mathematics*, 21(2):343–348, 1967.

Tencent Hy Team. Hy4-preview model card. https://huggingface.co/tencent/Hy4-preview, 2026. Accessed September 2026.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Perric Cistac, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 38–45, 2020.

Zhenda Xie, Yixuan Wei, Huanqi Cao, Chenggang Zhao, Chengqi Deng, Jiashi Li, Damai Dai, Huazuo Gao, Jiang Chang, Kuai Yu, Liang Zhao, Shangyan Zhou, Zhean Xu, Zhengyan Zhang, Wangding Zeng, Shengding Hu, Yuqing Wang, Jingyang Yuan, Lean Wang, and Wenfeng Liang. mHC: Manifold-Constrained Hyper-Connections. *arXiv preprint arXiv:2512.24880*, 2025. URL https://arxiv.org/abs/2512.24880.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4791–4800, 2019. doi: 10.18653/v1/P19-1472. URL https://aclanthology.org/P19-1472/.

Defa Zhu, Hongzhi Huang, Zihao Huang, Yutao Zeng, Yunyao Mao, Banggu Wu, Qiyang Min, and Xun Zhou. Hyper-Connections. *arXiv preprint arXiv:2409.19606*, 2024. URL https://arxiv.org/abs/2409.19606.

A Discussion and Implications

Our findings identify two properties of the trained pathway that call for further analysis: routing is concentrated within individual sublayers but reorganizes across depth, and learned residual mixing becomes nearly identity after mid-depth. We identify candidate optimization mechanisms associated with these patterns and consider implications for future multi-stream designs.

A.1 Routing Concentration

Read/write routing and stream representations are learned jointly, creating a possible feedback loop: greater read weight propagates a larger branch-mediated gradient to the parameters producing that stream, while the write map determines which persistent streams receive the branch output. These interactions may reinforce existing routing preferences, although the realized coefficients remain coupled through a shared token-dependent router. The following local analysis formalizes this channel; validating it requires training-time trajectories of routing margins, stream-wise gradients, and representations.

For a single token and sublayer, write the realized read and write maps as $\boldsymbol{r} = (r_1, \ldots, r_n)$ and $\boldsymbol{w} = (w_1, \ldots, w_n)$, such that $\boldsymbol{z} = \sum_j r_j \boldsymbol{x}^{(j)}$, $\boldsymbol{y} = F(\boldsymbol{z})$, and the branch adds $w_i \boldsymbol{y}$ to output stream i. Let $\boldsymbol{g}_z = \partial \mathcal{L}/\partial \boldsymbol{z}$ and $\boldsymbol{g}_i^+ = \partial \mathcal{L}/\partial \boldsymbol{x}^{+(i)}$. Conditioned on the realized routing coefficients, the direct branch-mediated gradients are

$$\left.\frac{\partial \mathcal{L}}{\partial \boldsymbol{x}^{(j)}}\right|_{\text{branch}} = r_j \boldsymbol{g}_z, \qquad \frac{\partial \mathcal{L}}{\partial r_j} = \left\langle \boldsymbol{g}_z, \boldsymbol{x}^{(j)} \right\rangle, \qquad \frac{\partial \mathcal{L}}{\partial w_i} = \left\langle \boldsymbol{g}_i^+, \boldsymbol{y} \right\rangle. \tag{11}$$

At the level of the realized read coefficients, define the negative-gradient direction

$$d_{r_j} \coloneqq -\frac{\partial \mathcal{L}}{\partial r_j} = -\left\langle \boldsymbol{g}_z, \boldsymbol{x}^{(j)} \right\rangle. \tag{12}$$

Stream j is favored over stream k by this local direction when $d_{r_j} > d_{r_k}$. A larger r_j also scales the branch-mediated gradient propagated through $\boldsymbol{x}^{(j)}$, and hence the gradient received by the upstream parameters that produce this state. If the resulting parameter update makes d_{r_j} more favorable on subsequent examples, the existing read preference can be reinforced. The write map participates in the same feedback by controlling which persistent states receive the branch output and influence later computation. Because the realized routing coefficients are coupled through a shared token-dependent router, these descent scores identify a possible feedback channel rather than the router parameters' actual update.

A.2 Residual Mixing

Movement away from identity depends on both the optimization pressure for cross-stream exchange and the sensitivity of Sinkhorn–Knopp normalization to that pressure. The final checkpoint cannot distinguish weak functional demand from weak gradient transmission to off-diagonal entries. The following local analysis formalizes this distinction. Neither local analysis reconstructs the training dynamics of the checkpoint, whose causal mechanisms require training-time routing and mixer trajectories.

Write the realized mixer at a token and sublayer as $\boldsymbol{H}^{\text{res}} = \mathcal{S}(\boldsymbol{A})$, where $\boldsymbol{A}$ contains the pre-Sinkhorn logits and $\mathcal{S}$ denotes Sinkhorn normalization, and let

$$\boldsymbol{h} \coloneqq \operatorname{vec}(\boldsymbol{H}^{\text{res}}), \qquad \boldsymbol{a} \coloneqq \operatorname{vec}(\boldsymbol{A}), \qquad \boldsymbol{J}_{\mathcal{S}} \coloneqq \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{a}}, \qquad \boldsymbol{g}_h \coloneqq \frac{\partial \mathcal{L}}{\partial \boldsymbol{h}}. \tag{13}$$

For an idealized local gradient step on the logits, first-order expansion gives

$$\Delta \boldsymbol{h} \approx -\eta \boldsymbol{J}_{\mathcal{S}} \boldsymbol{J}_{\mathcal{S}}^{\top} \boldsymbol{g}_h. \tag{14}$$

If $\boldsymbol{P}_{\text{off}}$ projects a vectorized mixer onto its off-diagonal entries, then

$$\|\boldsymbol{P}_{\text{off}} \Delta \boldsymbol{h}\|_2 \le \eta \, \|\boldsymbol{P}_{\text{off}} \boldsymbol{J}_{\mathcal{S}}\|_2 \, \|\boldsymbol{J}_{\mathcal{S}}\|_2 \, \|\boldsymbol{g}_h\|_2. \tag{15}$$

Movement away from identity can therefore remain small either because the loss gradient supplies little pressure for cross-stream residual exchange or because the local Sinkhorn Jacobian weakly transmits that pressure to off-diagonal entries. The final checkpoint does not reveal which factor governed the late-layer mixers or how they reached the observed regime.

Table 3: Mean diagnostics by depth range and sublayer type. Winner consistency (WC) and effective stream count (n_{eff}) are averaged over the indicated routing sites; cosine similarity is additionally averaged over the six stream pairs.

		Winner consistency		n_{eff}		Cosine	$\delta(\boldsymbol{H}^{\text{res}})$
Layers	**Site**	$\boldsymbol{H}^{\text{pre}}$	$\boldsymbol{H}^{\text{post}}$	$\boldsymbol{H}^{\text{pre}}$	$\boldsymbol{H}^{\text{post}}$		
0–21	Attention	0.874	0.970	2.174	1.317	0.456	0.045
	FFN	0.905	0.909	1.768	1.618	0.438	0.047
22–42	Attention	0.833	0.917	2.104	1.836	0.385	0.010
	FFN	0.872	0.823	1.947	2.360	0.378	0.008

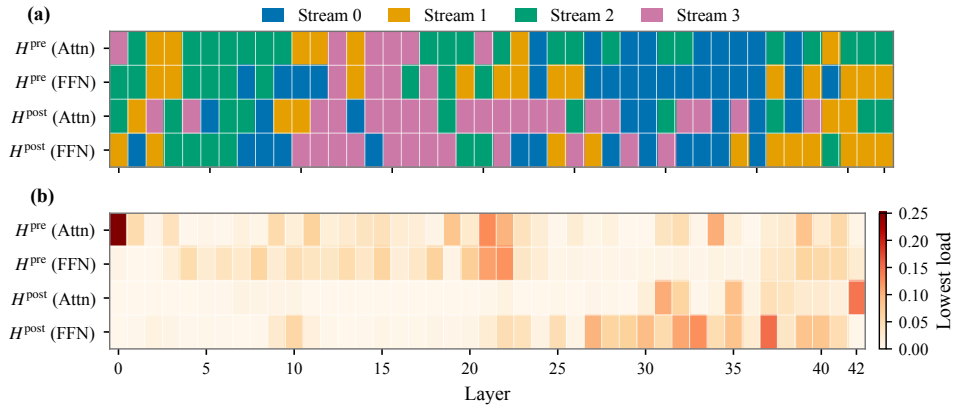


Figure 8: Lowest-load routing stream across depth. (a) Identity of the stream with the smallest mean normalized routing load at each layer, map, and sublayer type. (b) Its corresponding mean load. The weakest stream changes across sites rather than remaining globally fixed.

A.3 Directions for Improving *m*HC

Recent designs suggest that multi-stream residual parameterization remains unsettled: Hy4-preview adopts four-stream identity Hyper-Connections (Tencent Hy Team, 2026), while Qwen3.8-Flash-Next uses data-dependent Gated Residual across four branches (Qwen Team, 2026). These choices accord with our finding that not every degree of freedom in full *m*HC is functionally required. In *m*HC, all three maps are predicted from incoming states: this is necessary for $\boldsymbol{H}^{\text{pre}}$, but $\boldsymbol{H}^{\text{post}}$ selects where to write the branch output and $\boldsymbol{H}^{\text{res}}$ mixes incoming states without conditioning on the computation produced by the branch. A promising alternative is to retain input-conditioned $\boldsymbol{H}^{\text{pre}}$ while allowing $\boldsymbol{H}^{\text{post}}$, and potentially $\boldsymbol{H}^{\text{res}}$, to condition additionally on the branch output; evaluating it requires matched training comparisons with full and simplified alternatives.

B Additional Diagnostic Results

B.1 Diagnostics by Depth and Sublayer Type

Table 3 places the routing, representation, and residual-mixing diagnostics under a common depth split. The reduction in residual mixing is consistent across attention and FFN sites, while the change in routing breadth is concentrated in the write maps, particularly at FFN sites.

B.2 Identity of the Lowest-Load Route

Figure 8 ranks streams by their mean normalized routing load at each site. All four streams appear as the lowest-load stream somewhere in each map/site combination, so the locally weak route does not correspond to a single stream that can be removed throughout the network. Its mean load is

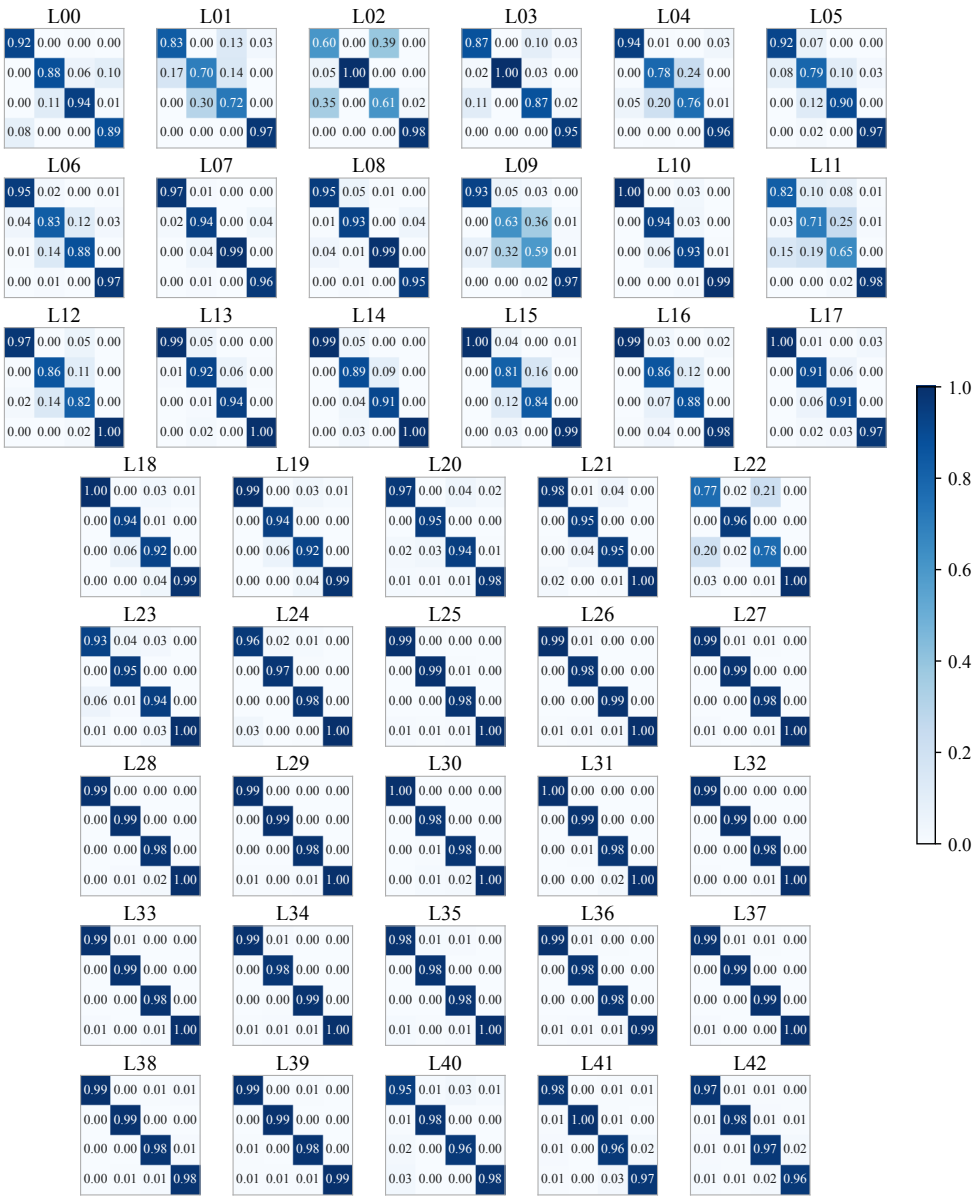


Figure 9: Token-averaged realized residual mixers at the attention site of every layer.

generally small but varies across depth, favoring site-adaptive sparsification over the removal of a globally fixed stream.

C Layerwise Residual Mixers

Figures 9 and 10 show the token-averaged realized residual mixer at every layer, separately for attention and FFN sites. They complement the depth profile and range averages in Figure 7 by exposing which stream pairs account for the residual exchange at each site.

Attention sites. Figure 9 shows that the off-diagonal structure is both depth- and pair-specific. The most visible exchange occurs at several early and middle sites, including layers 1–2, 9, and 11, followed by a final pronounced deviation at layer 22. Beyond this point, the matrices are consistently near diagonal, with only small site-level fluctuations through layer 42.

FFN sites. Figure 10 exhibits the same broad reduction in residual exchange but a different set of high-mixing sites. The strongest off-diagonal weights occur at layers 0–1 and 13, whereas the mixers from layer 23 onward remain close to identity. The locations and stream pairs responsible for early mixing therefore differ between attention and FFN, while the late near-identity regime is shared by both sublayer types.

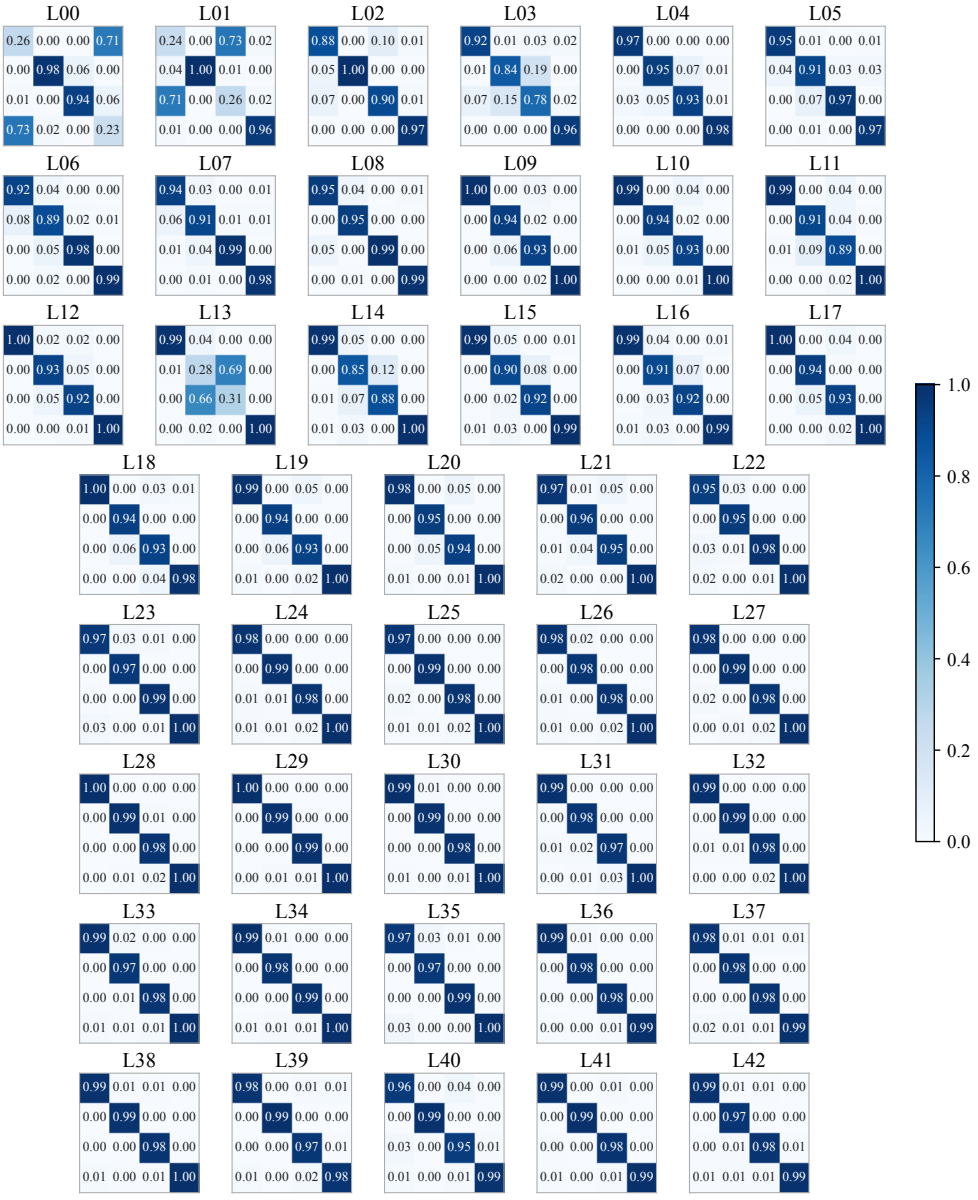


Figure 10: Token-averaged realized residual mixers at the FFN site of every layer.