DECOUPLED SEMANTIC ROUTING: ROLE-SPECIFIC RESIDUAL STREAM READOUT FOR ATTENTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Queries, keys, and values serve different purposes in attention, yet multi-stream Transformers typically construct all three from the same mixture of residual streams. We study whether they benefit from separate mixtures. A gradient probe on a trained Manifold-Constrained Hyper-Connections (mHC) model reveals aligned query and key preferences but opposing value preferences in deeper layers. Motivated by this observation, we introduce Decoupled Semantic Routing (DSR), which forms a separate stream mixture for each attention role. DSR computes the mixture weights in a low-dimensional space and applies them to the original hidden representations, using only streams at the current layer. On 1B and 3B language models trained for 1.64B tokens, DSR lowers validation loss relative to mHC by 0.00977 and 0.01830. Static mixtures and partial routing also retain the improvement, with several variants outperforming full dynamic routing. The results distinguish two choices in residual readout: which roles share a mixture, and whether that mixture varies across tokens. Separate mixtures remain effective even when their weights are static.

1 INTRODUCTION

Queries and keys determine which tokens attend to one another; values provide the information passed between them. In a Transformer, all three are usually projected from the same hidden state (Vaswani et al., 2017). This is a natural choice when each layer carries a single residual stream. Multi-stream architectures change that setting. Hyper-Connections (HC) maintains several parallel streams, and Manifold-Constrained Hyper-Connections (mHC) stabilizes their propagation through constrained mixing (Zhu et al., 2025; Xie et al., 2026). Attention now has several representations available at the same depth.

In mHC, these streams are still combined before the query, key, and value projections:

$$\{X_1, \ldots, X_n\} \longrightarrow x_{\text{base}} \longrightarrow \{x_{\text{base}}W_Q,\ x_{\text{base}}W_K,\ x_{\text{base}}W_V\}. \tag{1}$$

The projections can emphasize different features within this aggregate, but they cannot choose its stream composition independently. We ask whether queries, keys, and values benefit from choosing their own mixtures.

To examine this question, we probe a trained mHC model by perturbing each role's stream mixture independently. In deeper layers, the loss gradients for queries and keys tend to align, while those for values point in opposing directions. This pattern is absent at random initialization. The three roles have thus learned different local preferences over the same streams, which a shared mixture must reconcile.

Decoupled Semantic Routing (DSR) allows these preferences to be expressed through separate attention inputs. Each role assigns weights to the mHC aggregate and the individual residual streams. The weights are computed from low-dimensional projections, while the mixture itself is formed from the original hidden vectors. The resulting representations enter the usual attention computation, whose output updates the mHC streams (Figure 1).

MUDDFormer (Xiao et al., 2025) studies separate attention inputs formed from historical layer outputs. DSR brings role-specific aggregation to the parallel streams of an mHC layer. These streams coexist at the same depth, so their readout does not require retaining earlier layer states.

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.

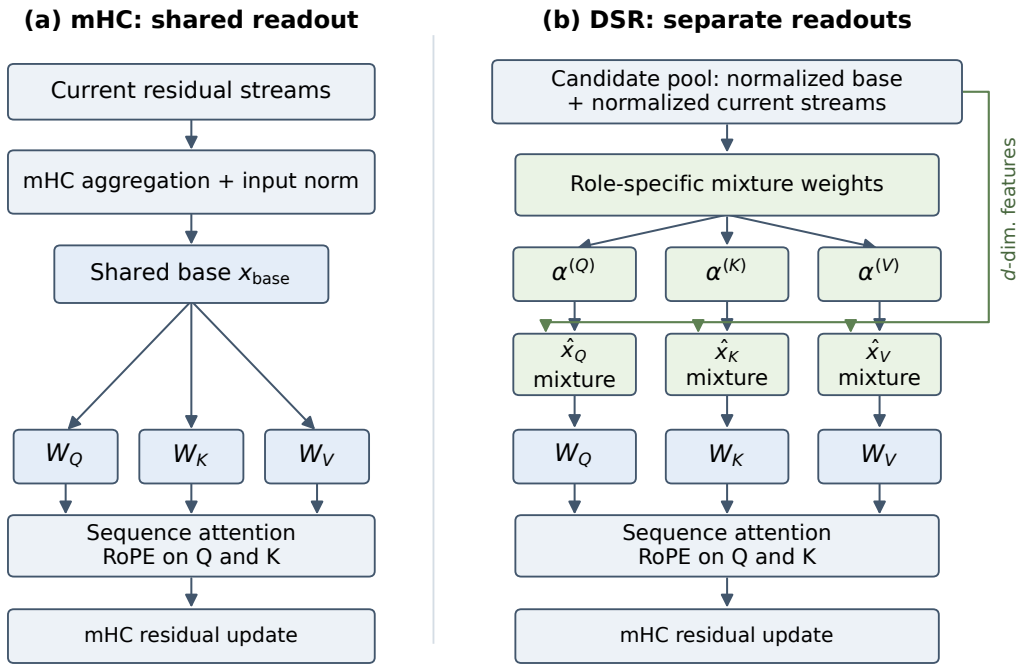


Figure 1: Residual pathways and DSR. mHC aggregates the current streams into one attention input. DSR computes separate query, key, and value weights in a compact space and applies them to the original hidden vectors. The residual update is the same in both architectures.

We evaluate DSR on 1B and 3B language models under a common training budget. Both achieve lower validation loss than mHC. The ablations separate two choices that are often made together: sharing a mixture across attention roles, and adapting it to each token. Learned static mixtures and routing only a subset of roles are also effective. This points to a simple extension of multi-stream models: preserve their residual propagation while allowing attention roles to combine the available representations differently.

2 Related Work

Residual propagation. Residual connections accumulate layer updates in a shared state. PreNorm and PostNorm (Xiong et al., 2020), LayerScale (Touvron et al., 2021), and DeepNorm (Wang et al., 2022) study how normalization and scaling affect optimization. Highway Networks (Srivastava et al., 2015) introduce gates between transformed features and identity paths. These methods primarily control how updates enter the residual pathway.

Parallel residual streams. HC expands the residual pathway into multiple streams with learned connections (Zhu et al., 2025). mHC constrains residual mixing to the doubly stochastic manifold using Sinkhorn-Knopp iterations (Xie et al., 2026). DSR builds on mHC, forming a separate mixture of its streams for each attention role.

Reading across depth. DenseFormer learns weighted combinations of earlier layer representations (Pagliardini et al., 2024). Attention Residuals uses input-dependent attention over prior layer outputs, with a block variant to reduce storage and communication costs (Kimi Team et al., 2026). MoDA lets attention heads read key–value pairs from preceding layers (Zhu et al., 2026). These methods select among representations retained across depth.

MUDDFormer learns dynamic connections over historical layers separately for query, key, value, and residual inputs (Xiao et al., 2025). Its role-specific aggregation is closely related to DSR. Here,

the representations being combined are parallel residual streams at a single depth, and their propagation follows the mHC update.

3 METHOD

3.1 MULTI-STREAM ATTENTION INPUTS

Let $X^{(l)} \in \mathbb{R}^{n\times d}$ contain the n residual streams at layer l, each of hidden dimension d. We describe one sequence position and omit its batch and token indices. mHC computes an aggregation vector $h_{\text{pre}} \in [0,1]^n$, an expansion vector $h_{\text{post}} \in [0,2]^n$, and a residual mixing matrix $H_{\text{res}} \in \mathbb{R}^{n\times n}$. With $\mathcal{N}_l$ denoting the sublayer's input normalization, its update is

$$u^{(l)} = \sum_{i=1}^{n} h_{\text{pre},i} X_i^{(l)}, \qquad x_{\text{base}} = \mathcal{N}_l(u^{(l)}), \tag{2}$$

$$X^{(l+1)} = H_{\text{res}}^{\top} X^{(l)} + h_{\text{post}}\, \mathcal{F}(x_{\text{base}}). \tag{3}$$

The outer product of the column vector h_{post} and row vector $\mathcal{F}(x_{\text{base}})$ distributes the sublayer output across streams. Sinkhorn-Knopp iterations approximately enforce doubly stochastic residual mixing. For attention, mHC uses the same normalized base for all three projections in Equation 1.

3.2 DECOUPLED SEMANTIC ROUTING

DSR forms its three attention inputs from the following representations:

$$X_{\text{pool}} = [x_{\text{base}};\, \text{RMSNorm}(X_1);\, \ldots;\, \text{RMSNorm}(X_n)] \in \mathbb{R}^{(n+1)\times d}. \tag{4}$$

The first row is the normalized mHC aggregate; the remaining rows are the normalized streams. The construction is applied independently at each token.

Computing the mixture weights. Three projections map the candidates to a routing dimension $d_R \ll d$:

$$Q_R = X_{\text{pool}} W_q^R, \qquad K_R = X_{\text{pool}} W_k^R, \qquad V_R = X_{\text{pool}} W_v^R, \tag{5}$$

where $W_q^R, W_k^R, W_v^R \in \mathbb{R}^{d\times d_R}$. These projections determine the stream weights and are distinct from the attention projections. We use normalized routing vectors $\bar{z} = z/\max(\|z\|_2, \epsilon)$.

The projected base is compared with each candidate to obtain the role-specific weights:

$$\begin{aligned} \alpha^{(Q)} &= \text{softmax}\big(\tau_Q \bar{Q}_{R,0} \bar{K}_R^{\top} + b_Q\big), \\ \alpha^{(K)} &= \text{softmax}\big(\tau_K \bar{K}_{R,0} \bar{Q}_R^{\top} + b_K\big), \\ \alpha^{(V)} &= \text{softmax}\big(\tau_V \bar{V}_{R,0} \bar{V}_R^{\top} + b_V\big). \end{aligned} \tag{6}$$

Here $b_r \in \mathbb{R}^{n+1}$ is a learned candidate bias and τ_r a learned scale. The softmax gives each role a distribution over the streams and their aggregate. Query and key use cross-scoring, while value is scored in its own space; Section 4.4 also examines same-role scoring.

Forming the attention inputs. The routing weights are applied directly to the candidate vectors:

$$\hat{x}_r = \sum_{i=0}^{n} \alpha_i^{(r)} X_{\text{pool},i}, \qquad r \in \{Q,K,V\}. \tag{7}$$

The role-specific inputs $\hat{x}_r$ therefore remain in $\mathbb{R}^d$; the low-dimensional projections affect their weights, not the features being combined.

Static and shared readouts. Role-specific mixtures can also be learned without token-dependent scores. In the static variant, each layer learns logits $\beta_r \in \mathbb{R}^{n+1}$ and sets

$$\alpha^{(r)} = \text{softmax}(\beta_r), \qquad r \in \{Q,K,V\}. \tag{8}$$

The weights are shared across tokens, while the representations in Equation 7 still depend on the input. With four streams, this requires 15 mixture logits per layer. Conversely, a shared readout ties $\alpha^{(Q)} = \alpha^{(K)} = \alpha^{(V)}$ while allowing the weights to depend on the token. These variants distinguish role separation from input dependence.

3.3 ATTENTION AND COMPUTATIONAL COST

DSR supplies the three bases to standard grouped-query attention (GQA):

$$Q = \hat{x}_Q W_Q, \qquad K = \hat{x}_K W_K, \qquad V = \hat{x}_V W_V. \tag{9}$$

After reshaping into query heads and key/value groups, rotary position embeddings are applied to Q and K. Each query head attends using its assigned key/value group and the usual causal mask. The attention output $y = \mathrm{Attn}(\hat{x}_Q, \hat{x}_K, \hat{x}_V)$, including the projection W_O, replaces $\mathcal{F}(x_{\mathrm{base}})$ in Equation 3. Feed-forward sublayers use the original mHC aggregate.

For each token and attention layer, routing adds

$$O(3(n+1)dd_R) + O(3(n+1)d_R) + O(3(n+1)d) \tag{10}$$

for the projections, scores, and weighted sums, respectively. The number of candidates is fixed at $n+1$, independent of the number of preceding layers. Measured training costs are reported in Section 4.3.

4 EXPERIMENTS

We study the effect of separate stream mixtures on language modeling, then use ablations and routing statistics to examine where the improvement arises.

4.1 EXPERIMENTAL SETUP

Our 3B model has 38 layers and hidden dimension 2560; the 1B model has 48 layers and hidden dimension 1024 (approximately 0.9B parameters). Both use GQA with four key/value groups, 2048-token sequences, and mHC with four streams and three Sinkhorn-Knopp iterations. DSR uses $d_R = 64$ unless specified otherwise.

All reported runs are trained from scratch on FineWeb (Penedo et al., 2024) with the Qwen2.5 tokenizer for 10,000 steps, or 1.64B tokens. Within each scale, data, optimizer, schedule, and token budget are held fixed. For the 1B models, a paired evaluation uses the same held-out batches for every model. Training and evaluation details are given in Appendix C.

4.2 ROLE PREFERENCES UNDER A SHARED READOUT

We first examine the stream preferences of a trained mHC model. For a layer with attention input x and streams $X_1, \ldots, X_n$, we perturb each role's input independently:

$$x^r = x + \sum_{i=1}^{n} \delta_i^r (X_i - x), \qquad r \in \{Q, K, V\}. \tag{11}$$

At $\delta^r = 0$, the forward pass is unchanged. The validation-loss gradient $g_r = \partial \mathcal{L} / \partial \delta^r|_{\delta=0}$ measures sensitivity to the mixture. We subtract its stream mean, $\tilde{g}_r = g_r - \bar{g}_r \mathbf{1}$, and compare roles using $C_{ab} = \cos(\tilde{g}_a, \tilde{g}_b)$. Positive cosine indicates aligned local preferences; negative cosine indicates opposing preferences.

We apply the probe to the final 1B mHC model using 500 validation batches (approximately 4.1M tokens). Figure 2 summarizes gradient alignment across layers, excluding the degenerate first layer. Error bars are nominal 95% Student-t intervals across layers, without a correction for dependence between adjacent layers.

Across layers, mean cosine is $+0.44$ for Q–K, -0.29 for Q–V, and -0.28 for K–V. The pattern is clearest in layers 32–48: Q–K is $+0.48$ with interval $[+0.13, +0.83]$, Q–V is -0.49 $[-0.81, -0.16]$, and K–V is -0.46 $[-0.79, -0.14]$. Only Q–K has an interval excluding zero in the shallow third; none does in the middle third. Gradient norms also grow substantially with depth, making the deeper-layer disagreement particularly relevant to the shared input.

At random initialization, all three cosine intervals include zero in every depth bin. The cancellation gap, $1 - \|\sum_r g_r\|_2 / \sum_r \|g_r\|_2$, averages 0.05 at initialization and 0.46 after training. The contrast with initialization suggests that the opposing preferences reflect the learned representations. We next examine the effect of allowing each role to form its own mixture.

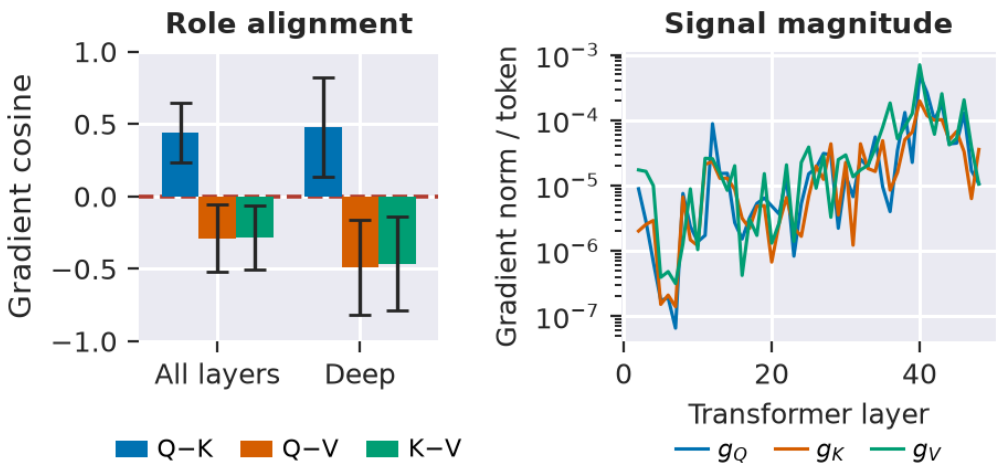


Figure 2: Gradient probe on the trained 1B mHC baseline. Left: stream-centered gradient cosine, pooled over nondegenerate layers and over layers 32–48; bars show nominal 95% Student-t intervals across layers. Right: per-token centered-gradient norms, which increase by roughly two orders of magnitude with depth.

4.3 Main Results

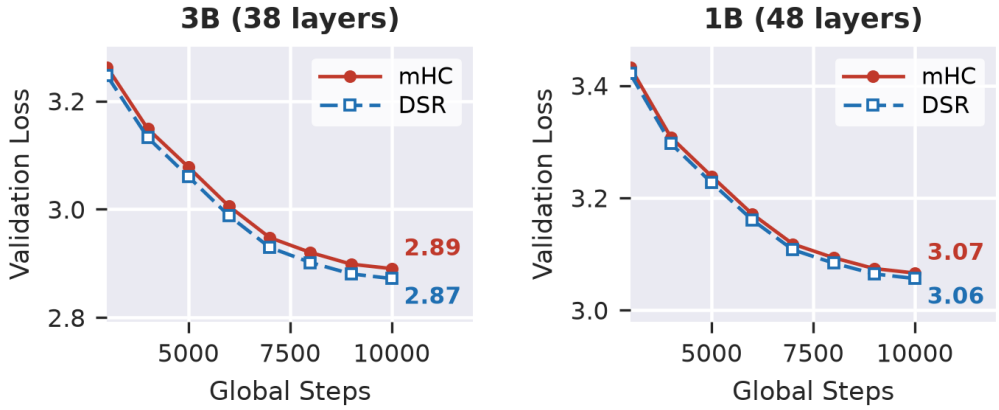


Figure 3: Validation loss during the second half of training. Solid curves show mHC and dashed curves show DSR for the 3B model (left) and 1B model (right). All runs use the same 1.64B-token budget. Table 1 reports the final values.

Table 1: Validation loss for mHC and DSR after 1.64B training tokens. Training settings are matched within each model scale.

Model	Layers	mHC	DSR	Δ
3B	38	2.89035	**2.87205**	−0.01830
1B	48	3.06608	**3.05631**	−0.00977

DSR achieves lower validation loss at both scales (Table 1). The reduction is 0.01830 for the 3B model and 0.00977 for the 1B model. In the paired evaluation of the 1B models on 50 held-out

batches (8.19M tokens), the mean mHC-minus-DSR difference is 0.0098, with standard error 0.0003 ($t = 34$).

The learning curves provide a complementary view. Between steps 3,000 and 10,000, the 1B gap changes from 0.0118 to 0.0098, while the 3B gap grows from 0.0159 to 0.0183. DSR retains its advantage over this interval in both configurations. The two models differ in width, depth, and tokens per parameter, so we treat them as two model configurations rather than a controlled test of depth scaling.

Training cost. At 1B, Full DSR processes 22,744 tokens/s compared with 34,002 for mHC, and uses 78.4 rather than 49.3 GiB of peak memory per GPU. With static routing, throughput rises to 25,523 tokens/s and peak memory falls to 71.1 GiB. Table 2 reports the remaining variants. The lower loss at a fixed token budget thus comes with additional computation and memory.

4.4 Ablation studies

We train all ablations with the 1B configuration and the same optimization settings and token budget. Every routed variant uses the candidate set in Equation 4. Full DSR gives all three roles separate input-dependent mixtures. *Shared weights* uses one learned input-dependent mixture for all roles; *static weights* learns a separate mixture per layer and role that is constant across tokens. *QK-only* routes queries and keys separately while values use x_{base}; *V-only* routes values while queries and keys use the base. *QK-coupled* learns one shared query/key mixture and a separate value mixture. We also test same-role scoring ($Q \to Q$, $K \to K$, $V \to V$), routing dimension 32, and an mHC baseline with 20 Sinkhorn-Knopp iterations. Appendix B specifies the weights used by each variant.

Table 2: Readout ablations at 1B under the same data and training schedule. Throughput is averaged over training; memory is the per-GPU peak. Appendix D reports the training-loss statistics.

Configuration	Val. loss ↓	Tokens/s ↑	Peak GiB ↓
mHC, no routing	3.06608	**34002**	**49.3**
mHC, 20 Sinkhorn iterations	3.07011	32733	**49.3**
Full DSR	3.05631	22744	78.4
Shared weights	3.06092	25067	74.6
Static weights	3.05568	25523	71.1
QK-only	**3.05381**	22274	77.2
V-only	3.05458	24877	76.0
QK-coupled	3.05791	21572	78.4
Same-role scoring	3.05765	22785	78.4
Full DSR, $d_R = 32$	3.05895	23214	76.7

Sharing one learned mixture among all three roles already lowers loss from 3.06608 to 3.06092. Allowing separate mixtures reduces it further to 3.05631. Thus the comparison favors giving the attention roles independent access to the same set of representations.

The improvement also appears with simpler forms of routing. Static per-role weights reach 3.05568, slightly below Full DSR, while QK-only and V-only reach 3.05381 and 3.05458. These variants preserve different mixtures for at least some of the roles, even though they remove part or all of the token-dependent routing. The static result is particularly useful: 720 mixture logits across 48 layers suffice to retain the loss improvement, without the dynamic routing projections. A learned preference over streams can therefore capture much of the benefit at this training budget. The learned distributions in the next section provide a closer view of this behavior.

QK-coupled averages the query and key routing logits before softmax. It has higher loss than Full DSR (paired $t = 8.7$), despite the aligned query and key gradients in the probe. V-only also gives queries and keys a shared input, but uses the original mHC base; its loss is 0.0033 lower than QK-coupled (paired $t = 16$). The choice of representation within a shared mixture therefore matters as well as the grouping of roles. This is compatible with the local nature of the probe: it measures preferences around a trained mHC model, while the routed models learn their representations and mixtures together.

Changing the scoring rule or reducing d_R to 32 preserves an improvement over mHC, with losses of 3.05765 and 3.05895. The default routing projections add 9.4M parameters at 1B and 18.7M at 3B (1.0% and 0.6%); QK-only uses two thirds of these additional parameters. Finally, using 20 rather than 3 Sinkhorn iterations gives an mHC loss of 3.07011. The baseline gap therefore remains with a larger iteration count.

4.5 Learned routing patterns

The learned mixtures remain broad. Averaged over 3,276,800 validation tokens, their depth-bin entropies range from 1.489 to 1.607 nats, against a maximum of $\ln 5 = 1.609$. Relative standard deviations across tokens are 3–5% in shallow and middle layers and 13–24% in deeper layers. The router therefore tends to adjust the balance among streams rather than select one stream exclusively. Its modest variation across tokens helps explain why static mixtures perform well.

Differences between roles become more pronounced with depth. The mean ℓ_1 distance between query and key weights grows from 0.094 in the shallow third to 0.286 in the deep third; the query–value distance grows from 0.089 to 0.399. Most variation is still shared by the three roles within a layer, with smaller role-specific differences superimposed on it. Appendix A gives the full distributions and statistics.

5 Conclusion

Parallel residual streams give a Transformer several representations at each depth. DSR allows queries, keys, and values to draw on those representations in different proportions. The gradient probe reveals differing local preferences, and the language-model experiments show lower validation loss with separate readouts. The gains persist with static and partial routing: separate attention inputs can be useful even without token-dependent mixture weights. This makes role-specific readout a simple way to extend a multi-stream architecture while retaining its residual update.

Limitations

Our experiments focus on pretraining loss after 1.64B tokens, with one training seed per configuration. The paired evaluations measure variation across held-out batches; behavior across training seeds, under longer training, and on downstream tasks remains open.

AI USE STATEMENT

Generative AI tools assisted with experiment design by suggesting the static-weight and QK-coupled ablations, with implementation of ablation variants, and with evaluation and analysis scripts. They also assisted with statistical analysis, result interpretation, manuscript editing, organization, and critical review, and creation of the routing-distribution, gradient-probe, and revised architecture figures. The authors developed the decoupled-readout design and gradient probe, implemented the full method, specified the other ablations, and created the original architecture diagram. Experimental values come from model runs. Generative AI was not used for proofs or manuscript translation.

We checked numerical results against training logs, verified configurations against training scripts, and reviewed AI-assisted text, code, and figures. We take responsibility for the final manuscript and its associated artifacts.

REPRODUCIBILITY STATEMENT

The routing equations and gradient probe are described in Sections 3 and 4.2. Appendix C gives the training and evaluation settings, and Appendix A defines the routing statistics. Training code and configuration files will be released with the final version.

REFERENCES

Kimi Team, Guangyu Chen, Yu Zhang, Jianlin Su, Weixin Xu, Siyuan Pan, Yaoyu Wang, Yucheng Wang, Guanduo Chen, Bohong Yin, Yutian Chen, Junjie Yan, Ming Wei, Y. Zhang, Fanqing Meng, Chao Hong, Xiaotong Xie, Shaowei Liu, Enzhe Lu, Yunpeng Tai, Yanru Chen, Xin Men, Haiqing Guo, Y. Charles, Haoyu Lu, Lin Sui, Jinguo Zhu, Zaida Zhou, Weiran He, Weixiao Huang, Xinran Xu, Yuzhi Wang, Guokun Lai, Yulun Du, Yuxin Wu, Zhilin Yang, and Xinyu Zhou. Attention Residuals, 2026. URL https://arxiv.org/abs/2603.15031.

Matteo Pagliardini, Amirkeivan Mohtashami, Francois Fleuret, and Martin Jaggi. DenseFormer: Enhancing Information Flow in Transformers via Depth Weighted Averaging, 2024. URL https://arxiv.org/abs/2402.02622.

Guilherme Penedo, Hynek Kydlíček, Loubna Ben Allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, and Thomas Wolf. The FineWeb Datasets: Decanting the Web for the Finest Text Data at Scale. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, 2024. URL https://arxiv.org/abs/2406.17557.

Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. Highway Networks, 2015. URL https://arxiv.org/abs/1505.00387.

Hugo Touvron, Matthieu Cord, Alexandre Sablayrolles, Gabriel Synnaeve, and Hervé Jégou. Going Deeper with Image Transformers. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. URL https://arxiv.org/abs/2103.17239.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention Is All You Need. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017. URL https://arxiv.org/abs/1706.03762.

Hongyu Wang, Shuming Ma, Li Dong, Shaohan Huang, Dongdong Zhang, and Furu Wei. DeepNet: Scaling Transformers to 1,000 Layers, 2022. URL https://arxiv.org/abs/2203.00555.

Da Xiao, Qingye Meng, Shengping Li, and Xingyuan Yuan. MUDDFormer: Breaking Residual Bottlenecks in Transformers via Multiway Dynamic Dense Connections. In *International Conference on Machine Learning (ICML)*, 2025. URL https://arxiv.org/abs/2502.12170.

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, 2026. URL https://arxiv.org/abs/2512.24880.

Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Chen Xing, Huishuai Zhang, Yanyan Lan, Liwei Wang, and Tie-Yan Liu. On Layer Normalization in the Transformer Architecture. In *International Conference on Machine Learning (ICML)*, 2020. URL https://arxiv.org/abs/2002.04745.

Defa Zhu, Hongzhi Huang, Zihao Huang, Yutao Zeng, Yunyao Mao, Banggu Wu, Qiyang Min, and Xun Zhou. Hyper-Connections, 2025. URL https://arxiv.org/abs/2409.19606.

Lianghui Zhu, Yuxin Fang, Bencheng Liao, Shijie Wang, Tianheng Cheng, Zilong Huang, Chen Chen, Lai Wei, Yutao Zeng, Ya Wang, Yi Lin, Yu Li, and Xinggang Wang. Mixture-of-Depths Attention, 2026. URL https://arxiv.org/abs/2603.15619.

A Learned Routing Statistics

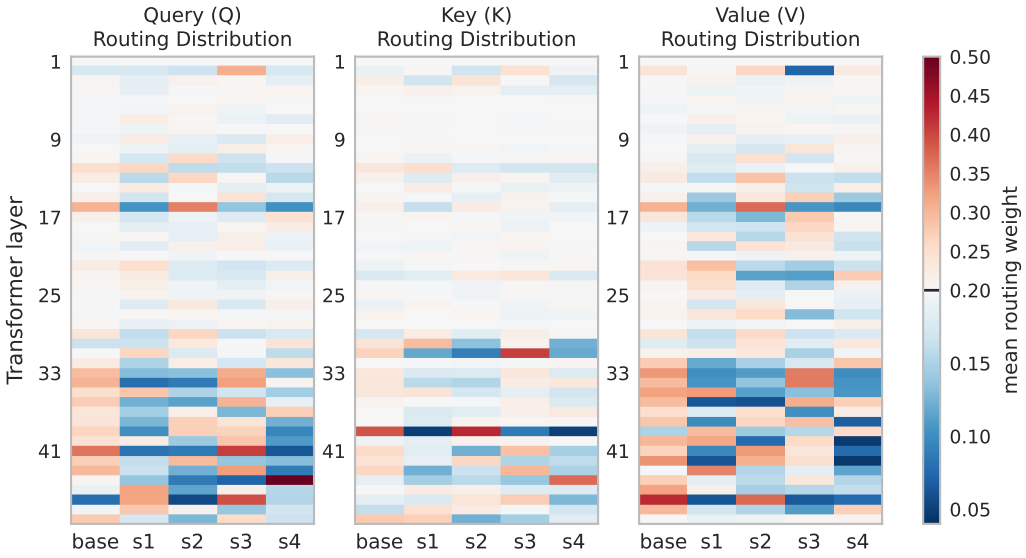


Figure 4: Learned routing weights of Full DSR at the 1B scale, averaged over 3,276,800 tokens. Color is centered on 0.20, the weight a uniform distribution would assign, so white denotes no preference and red (blue) denotes more (less) weight than uniform. Routing stays close to uniform in the shallow and middle layers and develops visible structure only in the deeper third, where Value concentrates on the base candidate while avoiding `stream_4` and Query favors `stream_3`.

We average the five routing weights over 100 validation iterations (400 micro-batches, 3,276,800 tokens), separately for each layer and role. Figure 4 shows these means relative to the uniform weight of 0.20.

Concentration and token dependence. The mean absolute deviation from uniform is 0.036. The largest individual deviation is 0.30: Query assigns weight 0.50 to `stream_4` in layer 44. Across the 144 layer–role pairs, the modal candidate has the highest weight for a median of 99% of tokens (range: 32–100%). The lower end occurs where the leading two candidates are nearly tied, so small changes in their weights can switch the ordering.

Table 3 summarizes the distributions by depth. Entropy ranges from 1.489 to 1.607 nats, close to the maximum $\ln 5 = 1.609$. Token-wise relative standard deviations are 3–5% in shallow and middle layers and 13–24% in deeper layers. A fairly consistent stream preference across tokens agrees with the static-weight result, which reaches loss 3.05568 compared with 3.05631 for Full DSR.

Variation across layers and roles. Let $w_{\ell,r,c}$ denote the mean weight of candidate c for role r at layer ℓ. The distance between two roles is $D_\ell(a, b) = \sum_c |w_{\ell,a,c} - w_{\ell,b,c}|$. From the shallow to the deep third, Q–K distance rises from 0.094 to 0.286 (3.1×), Q–V from 0.089 to 0.399 (4.5×), and K–V from 0.140 to 0.463 (3.3×). Their correlations with layer index are +0.59, +0.58, and +0.65, respectively. The depth bins exclude the degenerate first layer, as in Table 3.

Decomposing the deviation from uniform into a component shared within each layer and a role-specific component gives mean absolute deviations of 0.028 and 0.012. Layer preferences thus account for most of the visible structure, while role differences become clearer toward the end of the network.

Candidate-wise role contrasts. Across 48 layers, Key assigns 0.024 less weight to the base than Value ($t = 2.4$); the other fourteen candidate-wise role contrasts have $|t| < 1.6$. The candidate-wise comparisons are exploratory and use uncorrected t statistics. The Value-minus-Key base-weight gap

Table 3: Learned routing statistics by depth. *Base* is the weight on the shared base candidate; *relative std* is the per-token standard deviation of the weight divided by its mean, averaged over candidates, and measures how much the routing varies with the input; *entropy* is in nats against a maximum of $\ln 5 = 1.609$.

Depth	Role	Base	Streams	Relative std	Entropy
Shallow (2–16)	Q	0.210	0.790	0.045	1.594
	K	0.206	0.794	0.034	1.607
	V	0.211	0.789	0.035	1.590
Middle (17–31)	Q	0.205	0.795	0.042	1.603
	K	0.204	0.796	0.047	1.595
	V	0.216	0.784	0.047	1.588
Deep (32–48)	Q	0.249	0.751	0.186	1.506
	K	0.229	0.771	0.129	1.564
	V	0.284	0.716	0.237	1.489

grows from 0.005 in the shallow third ($t = 0.8$) to 0.055 in the deep third ($t = 2.0$). Averaged across the network, Query and Key assign `stream_3` weights 0.013 and 0.007 above uniform, while Value assigns 0.004 below uniform. Value has the largest average base preference, 0.037 above uniform.

Relation to the gradient probe. In the deep third, Query and Key have the closest mean routing distributions (ℓ_1 distance 0.286), and Key and Value the most distant (0.463). Queries and keys are thus closer to each other than either is to values, consistent with the grouping observed in the mHC gradient probe.

B Definitions of the Readout Variants

All variants combine the same $(n + 1)$ candidates in Equation 4. Let $s_Q, s_K, s_V \in \mathbb{R}^{n+1}$ denote the three logits before softmax in Equation 6, and let $e_0 = (1, 0, \ldots, 0)$ select the mHC base. Each input is $\hat{x}_r = \alpha^{(r)} X_{\text{pool}}$.

Table 4: Mixture weights for the readout ablations. σ denotes softmax, s_r is a token-dependent logit vector, and β_r is a learned vector shared across tokens.

Variant	$\alpha^{(Q)}$	$\alpha^{(K)}$	$\alpha^{(V)}$
Full DSR	$\sigma(s_Q)$	$\sigma(s_K)$	$\sigma(s_V)$
Shared weights	$\sigma(s_{\text{shared}})$	$\sigma(s_{\text{shared}})$	$\sigma(s_{\text{shared}})$
Static weights	$\sigma(\beta_Q)$	$\sigma(\beta_K)$	$\sigma(\beta_V)$
QK-only	$\sigma(s_Q)$	$\sigma(s_K)$	e_0
V-only	e_0	e_0	$\sigma(s_V)$
QK-coupled	$\sigma((s_Q + s_K)/2)$	$\sigma((s_Q + s_K)/2)$	$\sigma(s_V)$

Shared scoring. The shared variant uses two learned projections $A, B \in \mathbb{R}^{d \times d_R}$ to compare the base with every candidate:

$$s_{\text{shared},i} = \tau_{\text{shared}} \left\langle \overline{x_{\text{base}} A},\, \overline{X_{\text{pool},i} B} \right\rangle + b_{\text{shared},i}. \tag{12}$$

One resulting mixture supplies all three attention projections. Full DSR uses three routing projections and separate role-specific weights.

Static scoring. Each layer learns three vectors of five logits. This gives $3(n+1)L = 720$ mixture parameters in the 48-layer model. These logits replace the dynamic routing projections, logit scales, and candidate biases; the stream normalization is retained. The static mixture is initialized with weight 0.8 on the base and 0.05 on each of the four streams.

Partial and coupled readouts. QK-only and V-only keep the unrouted roles at x_{base}. QK-coupled computes s_Q and s_K as in Full DSR, averages them, and then applies softmax; it does not average the two softmax distributions. The value mixture remains independent.

Same-role scoring and routing dimension. Same-role scoring replaces the query–key cross-comparisons by $\tau_Q \bar{Q}_{R,0} \bar{Q}_R^\top + b_Q$ and $\tau_K \bar{K}_{R,0} \bar{K}_R^\top + b_K$. Value scoring and feature mixing are unchanged. The $d_R = 32$ variant halves the routing dimension in Equation 5.

C Training and Evaluation Details

Models. The 3B model has 38 layers, hidden dimension 2560, 20 query heads, and feed-forward dimension 6912. The 1B model has 48 layers, hidden dimension 1024, 16 query heads, and feed-forward dimension 4096. Both use grouped-query attention with 4 key/value groups, SwiGLU feed-forward layers without linear biases, RMSNorm with $\epsilon = 10^{-6}$, rotary position embeddings with base $10{,}000$, and 2048-token sequences.

Data and optimization. All models are trained from scratch on FineWeb with the Qwen2.5 tokenizer (151,936 entries) for 10,000 iterations at a global batch size of 80 sequences, giving 1.64B tokens, approximately 0.55 tokens per parameter for the 3B model and 1.82 for the 1B model. Training uses AdamW ($\beta_1 = 0.9$, $\beta_2 = 0.95$, $\epsilon = 10^{-8}$), weight decay 0.1, a cosine schedule with a peak learning rate of 2×10^{-4} and 500 warmup steps, gradient clipping at ℓ_2 norm 1.0, bfloat16 precision, and a fixed seed of 42.

Routing and distributed training. Routing projections are initialized from $\mathcal{N}(0, 0.02^2)$, with initial logit scales of 1 and candidate biases of 0. Routing vectors are normalized in FP32 with $\epsilon = 10^{-6}$, and logit scales are clamped to $[0.1, 20]$. The stream RMSNorm has a learned gain shared across the four streams. The 3B configuration uses a tensor-parallel size of 2 and a data-parallel size of 4 on 8 GPUs, whereas the 1B configuration uses a tensor-parallel size of 1 and a data-parallel size of 4 on 4 GPUs; both configurations use 5 gradient-accumulation steps.

Evaluation. Validation loss is evaluated every 1,000 iterations over 50 iterations of a held-out split of the same corpus; the paired re-evaluation uses validation samples 36,000–40,000, i.e. 50 global batches of 80 sequences (8.19M tokens), with identical batch fingerprints across runs. Runs are segmented at 1,000-iteration boundaries and resumed from checkpoints.

D TRAINING-LOSS STATISTICS

Table 5 reports the mean and standard deviation of training loss over the final 2% of iterations for each ablation.

Table 5: Training loss over the final 2% of iterations for the 1B runs.

Configuration	Tail mean	Tail standard deviation
mHC, no routing	3.05588	0.01372
mHC, 20 Sinkhorn iterations	3.06039	0.01367
Full DSR	3.04634	0.01365
Shared weights	3.05086	0.01373
Static weights	3.04584	0.01367
QK-only	3.04402	0.01368
V-only	3.04494	0.01389
QK-coupled	3.04764	0.01372
Same-role scoring	3.04735	0.01361
Full DSR, $d_R = 32$	3.04870	0.01365