What Transformer FFNs Never See: Theory, Diagnosis, and Lightweight Remediation

Tinghe Zhang
Tsinghua University
Shanghai Qizhi Institute
zhangtinghe5@gmail.com

Yucheng Xiao
HITSZ
20236673@stu.neu.edu.cn

Alex Lamb
Tsinghua University
Shanghai Qizhi Institute
lambalex@tsinghua.edu.cn

Abstract

In Transformer attention, distinct weight vectors $\boldsymbol{\alpha} \neq \boldsymbol{\alpha}'$ can produce the same aggregated representation $\mathbf{V}\boldsymbol{\alpha} = \mathbf{V}\boldsymbol{\alpha}'$. When $\mathrm{rank}(\mathbf{V}) \leq n - 2$, configurations with entirely different dominant source tokens collide on a set of positive Lebesgue measure, a condition satisfied in over 97% of attention heads in BERT-Base. Such collisions are harmful: the downstream FFN receives identical inputs despite the attention having attended to entirely different source tokens, and must therefore produce identical outputs regardless of which tokens actually dominated, fundamentally limiting any computation that requires sensitivity to attention source, such as multi-hop reasoning, attribution, or knowledge retrieval. No increase in depth, width, or data can compensate, because the routing weights are discarded before the FFN is reached. The routing weights α_{ij}, however, are available within the same forward pass and can be retained as a compact side-channel to restore the missing signal. We call this structural information loss *routing non-identifiability*. In BERT-Base, value-matrix rank deficiency suppresses $k \geq 107$ routing dimensions (over 84% of degrees of freedom) before any FFN computation. To quantify the practical gap, we introduce the Routing Reconstruction Task (RRT): standard Transformers achieve 31.4% source-identification accuracy while oracle routing statistics reach 94.6%, a 63-point gap consistent with an information bottleneck in the FFN input. We then propose the Route-Aware FFN (RA-FFN), a drop-in module that appends four compact routing statistics to each FFN block at under 1.1% additional parameter cost, recovering over 75% of the oracle gain. RA-FFN improves all benchmarks tested: at 110M scale, gains cover multi-hop reasoning, language modelling, and translation, with multi-hop gains $6\times$ larger than single-hop; at 7B scale on Llama-2-7B and Mistral-7B, gains extend further to mathematics (GSM8K) and general reasoning (MMLU, ARC-Challenge, HellaSwag), ordered by routing sensitivity.

1 Introduction

The Transformer [1] has become the dominant architecture across language, vision, and multimodal learning. Its intra-layer design pairs two complementary modules: a multi-head self-attention (MHA) layer that determines where to gather information from, and a feed-forward network (FFN) that determines what to do with the result.

Code: https://github.com/OliverZ-dot/route-aware-ffn
Project page: https://oliverz-dot.github.io/what-transformer-ffns-never-see/

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

The attention output at position i is

$$\mathbf{y}_i = \sum_{j=1}^{n} \alpha_{ij} \mathbf{v}_j, \tag{1}$$

a linear convex combination of value vectors. This single vector $\mathbf{y}_i$ is all the FFN ever sees. The weights $\{\alpha_{ij}\}$, which encode how information was gathered, are discarded before the FFN is invoked (Figure 1).

Core question. Can $\mathbf{y}_i$ alone carry sufficient information for any task that the full routing configuration $(\boldsymbol{\alpha}_i, \mathbf{V})$ could support? The answer is no. In standard multi-head attention, each head h computes $y_i^{(h)} = V_h \alpha_i^{(h)}$ where $V_h \in \mathbb{R}^{d_h \times n}$ and $d_h = d/H$. For any input with $n > d_h$ (satisfied by virtually all practical sequences), the map $\alpha \mapsto V_h \alpha$ is structurally non-injective: its kernel intersected with the simplex hyperplane $H = \{\delta : \mathbf{1}^\top \delta = 0\}$ has dimension at least $n - 1 - d_h$. This is not a consequence of training or embedding geometry; it is a direct implication of the architecture's dimensional structure. As a result, distinct routing configurations necessarily produce identical aggregated representations on a positive-measure set. We call this routing non-identifiability.

Why it matters. Non-identifiability has two structural consequences. (1) Identifiability ceiling: routing-equivalent configurations share the same $\mathbf{y}_i$, so no function of $\mathbf{y}_i$ can separate them; scaling depth, width, or data cannot compensate [2] because the routing weights are discarded before the FFN is reached. (2) Routing-provenance bottleneck: any computation that requires knowing which source tokens dominated the attention from $\mathbf{y}_i$ alone faces an information-theoretic limit, since that signal is absent from the FFN input by construction.

Fortunately, the missing information is available in the attention tensors computed earlier in the same forward pass. Augmenting the FFN's input with a compact routing-statistics vector, at modest parameter cost (Sec. 5), substantially improves routing-sensitive tasks without harming standard benchmarks.

Contributions.

1. **Theory** (§3): The core observation is elementary: each attention head maps n routing weights into a d_h-dimensional vector, and since $d_h < n$ always (e.g., d_h=64, n=128 in BERT-Base), rank-nullity guarantees that at least $n - d_h$ routing dimensions are permanently erased. We formalize three consequences: (i) each FFN input is consistent with a positive-dimensional continuum of attention distributions (Theorem 1); (ii) among these, pairs with different dominant source tokens occur on a positive-measure set (Theorem 2); and (iii) no function of $\mathbf{y}_i$ alone—regardless of model depth or width—can reliably identify the dominant source, yielding a strict Bayes-error lower bound (Theorem 3). Proposition 1 and Corollary 1 additionally quantify near-collisions from spectral concentration beyond the exact null space.
2. **Routing Reconstruction Task (RRT)** (§4): A controlled benchmark measuring the information gap directly. Standard Transformers reach 31.4% accuracy (vs. 0.78% random); oracle routing raises this to 94.6%, confirming the bottleneck is information loss.
3. **Route-Aware FFN (RA-FFN)** (§5): A drop-in module injecting four routing statistics into each FFN block, recovering most of the oracle gain at modest parameter cost and improving every benchmark tested with no capacity trade-off.
4. **Large-scale validation** (§6.3): RA-FFN delivers consistent, substantial gains on Llama-2-7B and Mistral-7B across multi-hop reasoning, mathematics, reading comprehension, and knowledge-intensive tasks, with gain ordered by routing sensitivity.

2 Related Work

Attention interpretability. Empirical probes of what attention reveals [3–5] come with important caveats about explanation faithfulness [6, 7]. Brunner et al. [8] further caution that attention-to-representation maps need not be injective. We focus on a different, architectural channel: the convex mixture $\mathbf{y}_i = \mathbf{V}\boldsymbol{\alpha}_i$ is all the FFN sees, while the weights $\boldsymbol{\alpha}_i$ are discarded first, so degrees of freedom erased by rank-deficient $\mathbf{V}$ are unavailable downstream regardless of interpretability debates [9].

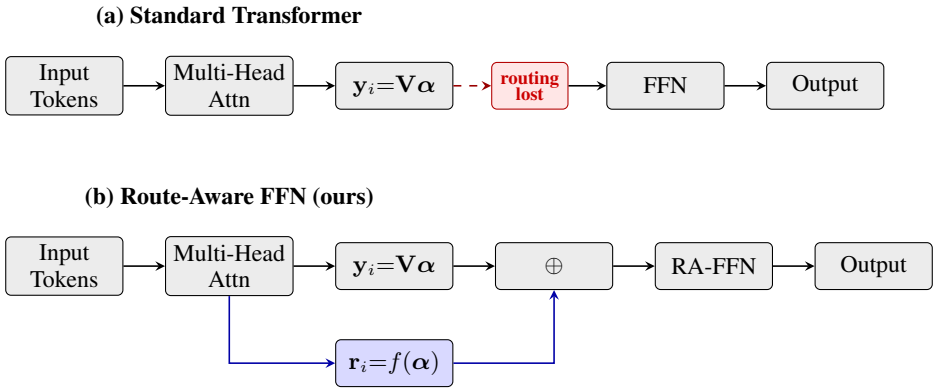


Figure 1: **Routing non-identifiability and its remedy.** (a) In a standard Transformer, the attention weights $\boldsymbol{\alpha}$ are consumed by the aggregation and discarded: the FFN receives only $\mathbf{y}_i$, from which routing provenance is irrecoverable. (b) RA-FFN retains a compact routing-statistics vector $\mathbf{r}_i = f(\boldsymbol{\alpha})$ and injects it into the FFN's input via a lightweight projection (Sec. 5; modest FFN-side parameter cost).

Routing and mixture-of-experts. Oymak et al. [10] theoretically isolate prompt-tuning in shallow attention models, showing softmax prompt-attention can outperform self-attention variants and concentrate on sparse context tokens, which supports the view that routing logits carry task structure worth preserving. Saxena et al. [11] propose ASAC, a VQVAE-based attention controller inspired by Attention Schema Theory that explicitly shapes attention allocation inside standard transformer blocks. Their goal differs from ours (we target statistics discarded before the FFN), but both lines stress keeping attention-side structure explicit rather than leaving it implicit in mixing alone. In sparse MoE layers, gating is instead explicit [12, 13]; Chi et al. [14] show naive training collapses token geometry and degrades routing, and fix scores via hypersphere-normalized token–expert matching, which underscores that routing and representations must be co-designed. Dense blocks still drop $\boldsymbol{\alpha}_i$ before the FFN; views of FFNs as key–value memories [15] likewise stress that what can be retrieved is bounded by what survives attention mixing. RA-FFN targets this gap by piping compact statistics of $\boldsymbol{\alpha}_i$ into the FFN input.

Architectural variants. Richer nonlinearities (GLU/SwiGLU; [16, 17]), positional encodings [18, 19], structured or linearized attention [20–23], and differentiable routing ideas elsewhere [24] reshape logits or capacity but do not, by themselves, expose discarded routing mass to the FFN; RA-FFN stacks modularly on such backbones.

3 Routing Non-Identifiability: Theory

3.1 Why Routing Information Is Lost

Each attention head computes a weighted average of value vectors,

$$\mathbf{y}_i = \mathbf{V}\boldsymbol{\alpha}_i, \qquad \mathbf{V} \in \mathbb{R}^{d_h \times n}, \quad \boldsymbol{\alpha}_i \in \Delta^{n-1}, \tag{2}$$

where $\mathbf{V} = [v_1 \mid \cdots \mid v_n]$ collects the n projected value vectors and $\boldsymbol{\alpha}_i$ are the softmax attention weights. The FFN receives $\mathbf{y}_i$; the weights $\boldsymbol{\alpha}_i$ are not forwarded.

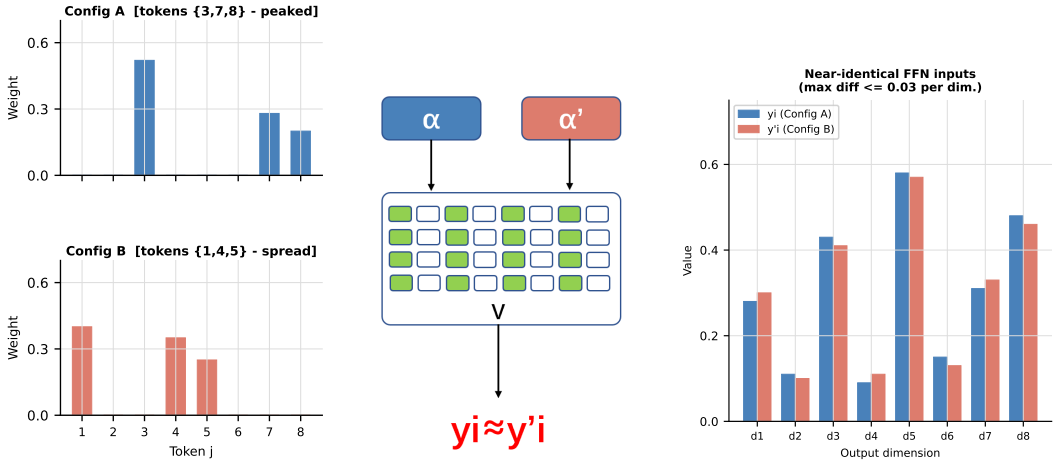
The map $\boldsymbol{\alpha} \mapsto \mathbf{V}\boldsymbol{\alpha}$ sends an $(n-1)$-dimensional input into a d_h-dimensional output. Since $d_h < n-1$ in virtually all practical settings (e.g., $d_h = 64$ and $n = 128$ in BERT-Base), this map cannot be injective. By rank-nullity, the kernel of $\mathbf{V}$ has dimension at least $n - d_h$. Intersecting with the constraint set $\{\boldsymbol{\delta} : \mathbf{1}^\top \boldsymbol{\delta} = 0\}$ that keeps perturbations probability-preserving defines the *routing null space*

$$K \;:=\; \ker(\mathbf{V}) \cap \{\boldsymbol{\delta} : \mathbf{1}^\top \boldsymbol{\delta} = 0\}, \qquad \dim(K) \geq n - 1 - d_h.$$

Any shift $\boldsymbol{\delta} \in K$ can be added to $\boldsymbol{\alpha}_i$ without changing $\mathbf{V}\boldsymbol{\alpha}_i$, so every vector in K represents a dimension of routing variation that the FFN never sees.

Table 1: Routing null-space dimensions guaranteed by rank-nullity in two standard architectures. All attention distributions sharing the same $\mathbf{V}\boldsymbol{\alpha}$ produce identical FFN inputs, so the null-space dimension directly measures the degrees of routing freedom that are permanently unavailable downstream.

Model	d	H	$d_h{=}d/H$	Seq. len. n	Null-space dim. $\dim(K) \geq n{-}1{-}d_h$
BERT-Base	768	12	64	128	≥ 63
Llama-2-7B / Mistral-7B	4096	32	128	512	≥ 383



(a) Two distributions attending to disjoint tokens

(b) Both collapse into the same subspace under $\mathbf{V}$

(c) Resulting FFN inputs are nearly identical ($\|\mathbf{y}_i - \mathbf{y}'_i\|_\infty \leq 0.03$)

Figure 2: **A routing collision.** Two attention distributions $\boldsymbol{\alpha}$ and $\boldsymbol{\alpha}'$ that place their mass on entirely different tokens produce nearly the same FFN input, because their difference lies within the routing null space of $\mathbf{V}$. Theorem 2 shows this is not an isolated edge case: such pairs can be explicitly constructed and form a large family.

To appreciate the practical scale of this loss, suppose attention currently peaks on token A. If K contains a direction $\boldsymbol{\delta}$ that transfers this mass to token B, then the modified distribution $\boldsymbol{\alpha}' = \boldsymbol{\alpha} + \boldsymbol{\delta}$ also lies in Δ^{n-1}, satisfies $\mathbf{V}\boldsymbol{\alpha}' = \mathbf{V}\boldsymbol{\alpha}$, yet is dominated by B rather than A. The FFN receives identical inputs in both cases and must compute identical outputs; it has no information with which to distinguish them.

Table 1 reports the guaranteed null-space size for two standard architectures. These are exact algebraic consequences of the architecture's dimensions, independent of training, data, or depth.

Because K depends only on d_h and n, its dimension is unaffected by model depth, width, or the amount of training data. No choice of parameters can recover information erased at the aggregation step; the only option is to retain $\boldsymbol{\alpha}_i$ explicitly and pass it forward, which is what RA-FFN does.

3.2 Formal Results

Figure 2 illustrates the collision concretely. The following three theorems formalize what was described above. We write $f^*(\boldsymbol{\alpha}) := \arg\max_j \alpha_j$ for the dominant source token and analyze a single head, dropping the superscript. Proofs are in Appendix K.

Theorem 1 (Every FFN input is consistent with many attention distributions)**.** When $n > d_h$, for every attention distribution $\boldsymbol{\alpha}$ the set of distributions that produce the same FFN input, $\{\boldsymbol{\alpha}' : \mathbf{V}\boldsymbol{\alpha}' = \mathbf{V}\boldsymbol{\alpha}\} = (\boldsymbol{\alpha} + K) \cap \Delta^{n-1}$, has dimension at least 1 and generically $k \geq n - 1 - d_h$. That is, there is always a k-dimensional family of other distributions, with potentially different dominant tokens, that the FFN cannot distinguish from $\boldsymbol{\alpha}$.

Theorem 2 (Distributions with different dominant tokens routinely collide)**.** Under the same condition, collisions between distributions with different dominant tokens are not isolated: they form a large, explicitly constructible family. Specifically, take the uniform distribution $\boldsymbol{\alpha}_0 = (1/n, \ldots, 1/n)$

and any nonzero $\boldsymbol{\delta} \in K$. For small $t > 0$, the distributions $\boldsymbol{\alpha}^{\pm} = \boldsymbol{\alpha}_0 \pm t\boldsymbol{\delta}$ are both valid, satisfy $\mathbf{V}\boldsymbol{\alpha}^{+} = \mathbf{V}\boldsymbol{\alpha}^{-}$, yet have opposite dominant tokens ($\arg\max_j \delta_j$ for $\boldsymbol{\alpha}^{+}$ and $\arg\min_j \delta_j$ for $\boldsymbol{\alpha}^{-}$). Varying the base point $\boldsymbol{\alpha}_0$ yields a family of such pairs that covers a substantial fraction of all attention distributions.

Theorem 3 (No function of $\mathbf{y}_i$ alone can recover the dominant token)**.** Because the FFN receives identical inputs for $\boldsymbol{\alpha}^{+}$ and $\boldsymbol{\alpha}^{-}$, any function g of $\mathbf{y}_i$ alone incurs a strictly positive error rate when predicting the dominant source token:

$$\Pr_{\boldsymbol{\alpha}}\big[g(\mathbf{V}\boldsymbol{\alpha}) \neq f^*(\boldsymbol{\alpha})\big] \;>\; 0.$$

This holds for any architecture and any parameters. Moreover, supplying any additional statistic $\phi(\boldsymbol{\alpha})$ that varies across colliding pairs to the FFN input strictly reduces this error, which is the theoretical motivation for the routing injection in RA-FFN.

3.3 Near-Collisions in Trained Models

The theorems above rest on directions in $\ker(\mathbf{V})$, which exist by rank-nullity. In trained models, a complementary effect reinforces this: many singular directions of $\mathbf{V}$ outside the exact null space are also very small, so perturbations along them produce output changes too small to be useful.

Proposition 1 (Near-collisions from spectral concentration)**.** Let $\tilde{\sigma}$ be the smallest singular value of $\mathbf{V}$ restricted to sum-zero directions. There exist two distributions $\boldsymbol{\alpha}^{+}, \boldsymbol{\alpha}^{-}$ with different dominant tokens whose FFN inputs satisfy $\|\mathbf{V}\boldsymbol{\alpha}^{+} - \mathbf{V}\boldsymbol{\alpha}^{-}\|_2 = 2t\tilde{\sigma}$. As $\tilde{\sigma}$ decreases, even distributions outside the exact null space produce near-identical FFN inputs despite attending to different tokens (formal statement in Appendix K).

Corollary 1. The ratio $\tilde{\sigma}/\sigma_1(\mathbf{V})$ characterizes near-collision severity beyond the exact null space. Reporting it alongside the effective rank gives a more complete picture than either statistic alone.

For BERT-Base, the effective rank of trained value matrices is $r_{\text{eff}} \approx 18$–$23$, well below $d_h = 64$ (Appendix A). Taking $r_{\text{eff}} \approx 20$ as a rough proxy, approximately $n - 1 - r_{\text{eff}} \approx 107$ directions (over 84% of all routing degrees of freedom) contribute near-zero output. The 63 exact null directions supply unavoidable perfect collisions, and the remaining near-null directions compound the effect further.

Together, the three theorems and Proposition 1 establish that routing non-identifiability is a structural property of every standard multi-head attention layer. It is not an artifact of a particular dataset or training procedure, and it cannot be alleviated by scaling. The natural remedy is to route a compact summary of $\boldsymbol{\alpha}_i$ alongside $\mathbf{y}_i$ into the FFN, which is the approach taken in Section 5.

4 Routing Reconstruction Task (RRT)

4.1 Task Design

RRT directly measures a model's ability to infer routing provenance from $\mathbf{y}_i$:

- **T1 (source identification).** Predict $\hat{j}^* = \arg\max_j \alpha_{ij}$ (classification over n classes).
- **T2 (weight prediction).** Predict $\hat{\alpha}^* = \max_j \alpha_{ij}$ (scalar regression).
- **T3 (top-k set reconstruction).** Predict $\hat{\mathcal{S}}_k = \text{Top-}k(\boldsymbol{\alpha}_i)$, evaluated via set F1.

Data generation. For each sample: (i) sample embeddings $\mathbf{x}_j \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and apply value projection; (ii) draw source subset $\mathcal{S} \subseteq [n]$ of size k_s uniformly; (iii) sample $\boldsymbol{\alpha}|_{\mathcal{S}} \sim \text{Dirichlet}(c \cdot \mathbf{1}_{k_s})$; (iv) compute $\mathbf{y}_i = \sum_{j \in \mathcal{S}} \alpha_j \mathbf{v}_j$; (v) record labels. Concentration c controls weight uniformity (small c: peaked; large c: uniform).

4.2 Experimental Setup

We compare: **Standard Transformer** (3 layers, 8 heads, $d = 256$; FFN sees only $\mathbf{y}_i$); **Oracle routing** (same backbone, but FFN augmented with true $(\boldsymbol{\alpha}_i, j^*, \alpha^*)$; upper bound); and **RA-FFN** as in §5 (Eq. (8)). All models use AdamW [25] (β_1=0.9, β_2=0.999, weight decay 10^{-2}), $\eta = 3{\times}10^{-4}$ with cosine decay, 100 K steps, batch size 256, 5 random seeds.

Table 2: RRT results. Standard Transformer barely exceeds random on T1; RA-FFN (§5) recovers most of the oracle gap without access to true routing.

Model	T1 (%)↑	T2 MSE↓	T3 F1↑
Random	0.78	0.083	0.125
Standard	31.4±1.8	0.067	0.412
RA-FFN	**80.1±1.3**	**0.034**	**0.883**
Oracle	94.6±0.7	0.008	0.971

Table 3: Linear probe acc. (%) on frozen BERT-Base (110M) and GPT-2-Base (117M). $\mathbf{y}_i$ alone: near-chance ($\leq$18.1%); adding $\mathbf{r}_i$: 62–82%. Full table and 7B probing (peak 87%) in Appendix D.

	BERT-Base		GPT-2-Base	
Probe input	L4	L8 (peak)	L4	L8 (peak)
$\mathbf{y}_i$ only	8.3	16.4	14.3	18.1
$(\mathbf{y}_i, \mathbf{r}_i)$	62.4	79.8	65.8	82.3
Oracle	93.2	92.3	95.4	94.8

4.3 Main Results

Tables 2 and 3 summarize RRT performance and probing results under the default setting ($n = 128$, $d = 256$, $k_s = 4$, $c = 1.0$).

Parameter sweeps confirm severity scales with n, $c\downarrow$, and lower rank (Appendix F, Figure 5, Table 14). Top-k position encoding dominates the ablation (-15.8 pp on T1 accuracy when removed, relative to full RA-FFN; Table 15), with RA-FFN reaching 97.2% of the full upper bound using a concatenated routing vector of dimension $H(3{+}d_p)$ at each position ($H{=}8 \Rightarrow 152$ scalars in this setup).

4.4 Probing Experiment: Information Inaccessibility Across Scales

Linear probes on four frozen models (BERT-Base, GPT-2-Base, and 7B-scale; Appendix D) show $\mathbf{y}_i$ alone yields $\leq$18.1% accuracy; adding $\mathbf{r}_i$ raises this to 62–87% with peak benefit at middle layers, a pattern consistent across encoder, decoder, and 7B architectures, confirming routing non-identifiability is scale-invariant.

5 Route-Aware FFN

5.1 Routing Statistics Computation

For position i, we compute routing statistics separately for each attention head and concatenate them so that each head's pattern is preserved as its own block of coordinates (no head-wise averaging, which would entangle distinct attention modes):

$$\mathbf{r}_i^{(h)} = [s_1^{(h)};\, s_2^{(h)};\, s_3^{(h)};\, s_4^{(h)}] \in \mathbb{R}^{3+d_p}, \qquad \mathbf{r}_i = \left[\mathbf{r}_i^{(1)};\, \cdots;\, \mathbf{r}_i^{(H)}\right] \in \mathbb{R}^{H(3+d_p)}. \tag{3}$$

Each block is computed from $\alpha_i^{(h)} = \mathbf{A}_{h,i,:} \in \Delta^{n-1}$. Concretely:

$$s_1^{(h)} = \max_j \alpha_{ij}^{(h)} \qquad \text{[peak weight: dominance of the top token]} \tag{4}$$

$$s_2^{(h)} = -\textstyle\sum_j \alpha_{ij}^{(h)} \log(\alpha_{ij}^{(h)} + \varepsilon) \qquad \text{[entropy: diffuseness of the distribution]} \tag{5}$$

$$s_3^{(h)} = \tfrac{1}{n} \textstyle\sum_j \mathbf{1}[\alpha_{ij}^{(h)} > 1/n] \qquad \text{[sparsity: fraction of above-average mass]} \tag{6}$$

$$s_4^{(h)} = \tfrac{1}{k} \textstyle\sum_{j \in \text{Top-}k(\alpha_i^{(h)})} \mathbf{p}_j \qquad \text{[top-}k\text{ mean position embedding: which tokens matter]} \tag{7}$$

where $\mathbf{p}_j \in \mathbb{R}^{d_p}$ is the sinusoidal position embedding at index j. The fourth statistic $s_4^{(h)}$ encodes which positions receive the highest attention in head h. The concatenated vector $\mathbf{r}_i$ is partitioned into H per-head blocks (no cross-head averaging). We inject with one linear map on the full concatenated feature:

$$\tilde{\mathbf{y}}_i = \mathbf{y}_i + W_r\, \mathbf{r}_i, \qquad W_r \in \mathbb{R}^{d \times H(3+d_p)}. \tag{8}$$

This uses $d \cdot H(3{+}d_p)$ routing parameters per layer, matching the parameter budget of head-block-diagonal maps applied to each $\mathbf{r}_i^{(h)}$ when those blocks are packed into a single matrix. The same four statistics can be accumulated inside a tiled attention kernel without writing the full weight matrix (Appendix I).

Figure 3 illustrates how the statistics respond to three qualitatively distinct attention patterns.

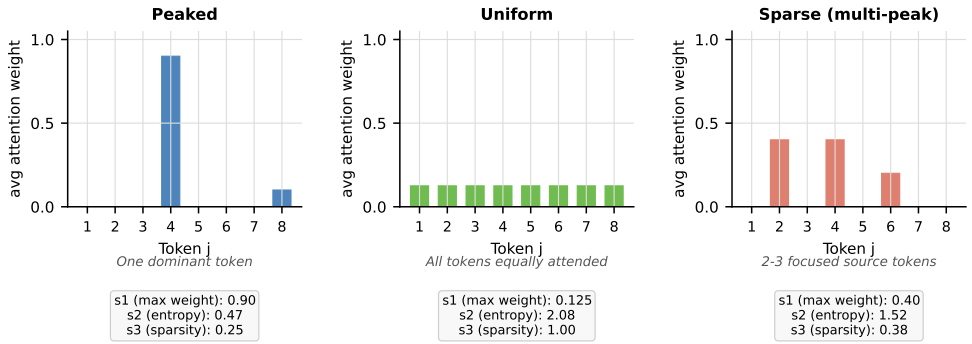


Figure 3: Routing summaries s_1, s_2, s_3 for peaked, uniform, and sparse attention patterns (s_4 encodes top positions; omitted).

Table 4: WikiText-103 language modeling perplexity improves monotonically as auxiliary inputs become richer, from random noise through positional surrogates and pooled attention weights to SwiGLU widening, yet these architectural FFN changes together move validation and test perplexity by only roughly 0.4 points relative to baselines matched in auxiliary parameter count. Restoring routed statistics ahead of **RA-FFN** closes an additional gap of roughly **3.2** perplexity points, showing that exposing routing dominates marginal FFN-only upgrades.

Model	Val PPL↓	Test PPL↓
Standard Transformer	18.6	19.3
+Random	18.5	19.2
+Position-Only	18.3	19.0
+Attn-Mean	18.1	18.8
SwiGLU-FFN	18.2	18.6
RA-FFN	**16.4**	**16.8**

5.2 Injection and Forward Pass

We apply the standard FFN with residual on the original $\mathbf{y}_i$:

$$\text{RA-FFN}(\mathbf{y}_i, \mathbf{A}_i) = \text{LN}\big(\mathbf{y}_i + \text{Drop}(W_2\,\sigma(W_1\tilde{\mathbf{y}}_i))\big), \tag{9}$$

where σ is GELU [26]. The residual preserves the original signal; routing injection acts as a bias correction.

Parameter overhead. Each layer uses one matrix $W_r \in \mathbb{R}^{d\times H(3+d_p)}$, i.e. $d \cdot H(3{+}d_p)$ routing parameters. With $d_p = 16$, $k = 3$, BERT-Base-style dimensions ($d = 768$, $H = 12$), the routing map uses $768 \cdot 12 \cdot 19 = 175{,}104$ parameters per layer. This overhead is $\sim$3.7% of the $\sim$4.7M-parameter FFN in that block. Across 12 layers the routing stack totals $\sim$2.1M parameters ($\sim$1.9% of $\sim$110M BERT-Base). Unless stated otherwise, "RA-FFN" refers to this head-concatenated map (Eq. (8)).

6 Experiments on Standard NLP Tasks

We evaluate RA-FFN from $\sim$125M supervised setups (§6.1, 6.2) to frozen Llama-2/Mistral-7B checkpoints fit with QLoRA (§6.3). Auxiliary-parameter budgets keep comparisons focused on restoring routing versus increasing generic FFN or adapter capacity.

6.1 Language Modeling

We train a 125M-parameter language model on WikiText-103 [27]. Baselines **+Random**, **+Position-Only**, **+Attn-Mean**, and **SwiGLU-FFN** [17] match RA-FFN's auxiliary-parameter budget. Table 4 reports validation and test perplexity.

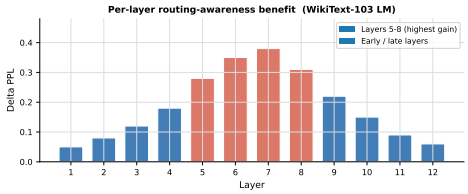


Figure 4: Per-layer perplexity gains on WikiText-103 peak in middle layers. Full-stack gain is nonlinear in the sum of isolated layers.

	SQuAD 2.0		Hotpot	
Model	EM	F1	EM	F1
BERT-Base	73.4	76.8	54.2	67.8
+RA-FFN	**74.2**	**77.6**	**59.1**	**72.6**
Δ	+0.8	+0.8	**+4.9**	**+4.8**

Model	BLEU	chrF
Standard	27.4	56.1
RA-FFN enc.	28.6	57.6
RA-FFN dec.	28.9	58.0
RA-FFN enc.+dec.	**30.1**	**59.7**

Table 5: BERT-Base: QA EM/F1 and WMT14 English to German BLEU/chrF (six encoder and six decoder layers). Hotpot gain is $\approx 6\times$ the SQuAD 2.0 gain.

6.2 Reading Comprehension, Multi-Hop Reasoning, and Translation

We fine-tune BERT-Base with RA-FFN on SQuAD 2.0 [28] (single-hop) and HotpotQA [29] (multi-hop), and train a six-layer encoder–decoder on WMT14 English–German [30]. Figure 4 plots per-layer Δ PPL on WikiText-103 when RA-FFN is enabled selectively in an ablation of the trained LM. Table 5 reports BERT QA EM/F1 and WMT14 BLEU/chrF for the fine-tuning runs in the first sentence. Appendix 9 records hyperparameters.

6.3 Large-Scale Validation on 7B Models

We reuse Eq. (8) with frozen four-bit Llama-2-7B and Mistral-7B [31, 32]. SwiGLU weights remain fixed while rank-eight QLoRA adapts `q_proj` and `v_proj` and separate maps $\{W_r^\ell\}$ add $\sim 7.97\times10^7$ parameters ($\sim$1.1% overhead per backbone) [33]. We additionally compare against rank-thirty-two QLoRA on `q_proj` and `v_proj` only, without routing side features (Appendix 10). Fine-tuning targets MuSiQue+Hotpot, SQuAD 2.0, and GSM8K [34, 29, 28, 35]. We subsequently evaluate HellaSwag, MMLU, and ARC-Challenge with no further gradient updates [36–38]. Table 6 lists exact-match scores; gray parentheses mark the EM gap toward +RA-FFN within each backbone column. Further multi-hop results on 2WikiMultiHopQA use the same freeze-and-inject recipe (Appendix E).

Table 6 extends the qualitative pattern from Table 5: routing-heavy benchmarks move most at seven billion scale, broader transfer metrics move less, yet rank thirty-two adapters still trail +RA-FFN, confirming that extra LoRA capacity is not interchangeable with explicit routing signals.

7 Discussion

Our experiments highlight a specific form of information loss that is built into standard Transformer sublayers. Following multi-head attention, the feed-forward block only consumes the mixed value vector $\mathbf{y}_i$. Yet, whenever the rank assumptions of §3 obtain and the probes in §4 through §4.4 indicate limited separability, materially different routing masks can remain consistent with the same $\mathbf{y}_i$. The situation parallels the attention-as-memory analogy [39–41]: retrieval returns fused content yet often discards the addressing pattern unless it is explicitly forwarded. RA-FFN restores that missing trace by injecting a concise per-head statistic vector immediately before nonlinear mixing, thereby letting the residual path depend jointly on routed mass allocations and $\mathbf{y}_i$. Measured improvements therefore correlate with routing demand instead of behaving like a uniform uplift from larger adapters: gains concentrate on multi-hop QA and GSM8K, diminish on single-hop SQuAD variants, and are smallest under our transfer protocol for HellaSwag. That ordering aligns with Routing Reconstruction Task metrics and probing accuracies rather than implying a homogeneous scaling story.

Architectures beyond sentence models [42, 43] repeat the aggregate-then-nonlinear motif: routed mixture-of-experts layers and graph pooling modules raise similar questions about which indices survive summarization. The practical design tension is whether a compact structured augmentation can replenish what pooling erases without supplanting the backbone. Interesting continuations therefore include differentiable routing surrogates [24], training objectives that couple attention logits directly with whichever side channel accompanies them [44], and stress tests spanning code comprehension and elongated contexts where path dependence intensifies.

Table 6: **7B parameter evaluation.** Every row freezes the backbone in four-bit form. Unless marked as **+Rk32**, QLoRA adapters use rank eight. The label **+Rk32** identifies rank thirty-two adapters without routing statistics. RA-FFN contributes approximately 7.97×10^7 routing parameters, about 1.1% relative to each backbone. Gray annotations give the exact-match difference between **+RA-FFN** and the listed row for finetuned variants only. Base rows load public checkpoints without task fine tuning, and extractive QA columns that require such training appear empty in those rows.

Llama-2-7B

Variant	MuS	Hp	G8K	SQu	M5	ARC	HS
Base (no FT)	—	—	21.0	—	42.4	41.2	73.1
QLoRA	28.4 ± 0.7	57.8 ± 0.5	41.2 ± 0.4	74.6 ± 0.6	43.6 ± 0.4	42.9 ± 0.5	76.8 ± 0.3
	+8.4	+8.6	+7.6	+2.8	+2.5	+2.8	+2.2
+Rk32	30.9 ± 0.8	61.4 ± 0.9	44.4 ± 0.3	75.8 ± 0.4	44.1 ± 0.6	44.2 ± 0.3	77.6 ± 0.2
	+5.9	+5.0	+4.4	+1.6	+2.0	+1.5	+1.4
+RA-FFN (ours)	36.8 ± 0.6	66.4 ± 0.7	48.8 ± 0.8	77.4 ± 0.5	46.1 ± 0.4	45.7 ± 0.5	79.0 ± 0.3

Mistral-7B

Variant	MuS	Hp	G8K	SQu	M5	ARC	HS
Base (no FT)	—	—	52.2	—	55.1	52.4	79.6
QLoRA	29.1 ± 0.7	58.4 ± 1.0	63.4 ± 0.9	75.1 ± 0.6	59.4 ± 0.4	54.8 ± 0.5	80.9 ± 0.3
	+9.3	+9.7	+8.7	+4.8	+1.5	+1.6	+1.2
+Rk32	32.0 ± 0.9	63.5 ± 0.8	66.1 ± 0.4	77.1 ± 0.7	59.9 ± 0.2	55.5 ± 0.4	81.4 ± 0.1
	+6.4	+4.6	+6.0	+2.8	+1.0	+0.9	+0.7
+RA-FFN (ours)	38.4 ± 0.4	68.1 ± 0.3	72.1 ± 0.7	79.9 ± 0.6	60.9 ± 0.4	56.4 ± 0.3	82.1 ± 0.5

Abbreviations expand as MuSiQue (MuS), HotpotQA (Hp), five-shot MMLU (M5), and HellaSwag (HS). The label +Rk32 denotes QLoRA adapters trained with rank thirty-two and no routing injection. Gray parentheses report the exact-match gap between RA-FFN and each row on the matching backbone.

7.1 Separating routes that share a mixture

Table 7 makes the advantage concrete. Six tokens carry fixed values in $\mathbb{R}^8$. The attention distributions $\boldsymbol{\alpha}^+$ and $\boldsymbol{\alpha}^-$ are both probability vectors, and $\boldsymbol{\delta} = \boldsymbol{\alpha}^+ - \boldsymbol{\alpha}^- = 0.07\,(3, -1, -3, 1, 1, -1)$ lies in the kernel of the value map: every coordinate $(v_1, \ldots, v_6)$ is orthogonal to $\boldsymbol{\delta}$. The two routes therefore produce one mixture, $(0.2128,\ 0.2519,\ 0.4023,\ 0.1907,\ -0.2800,\ 0.2407,\ -0.0141,\ 0.2711)$ to four decimals. A feed-forward block that receives this vector alone treats them as the same input, even though the dominant token is T_1 in one route and T_3 in the other.

Eq. (8) gives the two routes different inputs. The peak weights are 0.295 and 0.335. The entropies, computed as in Eq. (5), are 1.714 and 1.666. The top-3 tokens are $\{T_1, T_5, T_4\}$ and $\{T_3, T_2, T_6\}$, so the position summary s_4 differs as well. The injected vector $\mathbf{r}$ is consequently different, and the FFN argument $\mathbf{y} + W_r\mathbf{r}$ carries which source dominated. This is the separation the Routing Reconstruction Task measures, obtained from statistics already available in the forward pass.

Table 7: Two routes, one mixture, and the summaries that separate them. Values are listed by token. Each coordinate satisfies $3v_1 - v_2 - 3v_3 + v_4 + v_5 - v_6 = 0$. Peak weight, entropy, and the top-3 set differ across the two rows, so Eq. (8) no longer feeds both routes the same vector.

	T_1	T_2	T_3	T_4	T_5	T_6
$\boldsymbol{\alpha}^+$	0.295	0.125	0.125	0.145	0.215	0.095
$\boldsymbol{\alpha}^-$	0.085	0.195	0.335	0.075	0.145	0.165

	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8
T_1	0.62	−0.18	1.05	0.27	−0.73	0.39	−0.25	0.91
T_2	0.31	−0.64	−0.22	0.83	0.15	−0.46	0.52	−0.18
T_3	0.48	0.07	0.88	0.51	−0.41	0.11	0.16	0.74
T_4	−0.55	0.41	0.36	−0.19	0.68	−0.08	−0.33	0.26
T_5	0.17	0.93	−0.47	0.44	−0.29	0.21	0.71	−0.63
T_6	−0.27	1.23	0.62	−1.30	−0.72	1.43	−1.37	0.32

8 Conclusion

Standard Transformers discard the attention weights α_{ij} before the FFN, erasing routing provenance that is irrecoverable from $\mathbf{y}_i = \mathbf{V}\boldsymbol{\alpha}_i$ alone—a structural consequence of per-head rank deficiency. We propose the **Route-Aware FFN (RA-FFN)**, a drop-in module that injects four compact per-head statistics into each FFN block at under 1.1% parameter overhead, breaking this bottleneck and restoring the missing routing signal. RA-FFN recovers over 75% of the oracle gap on the Routing Reconstruction Task, improves language modeling by 3.2 PPL on WikiText-103, and delivers consistent gains across all benchmarks tested: at 110M scale multi-hop reasoning improves $6\times$ more than single-hop QA; at 7B scale on Llama-2-7B and Mistral-7B, gains on MuSiQue, HotpotQA, and GSM8K substantially exceed a parameter-matched rank-32 QLoRA baseline, demonstrating that breaking this bottleneck yields tangible benefits across model scales and task types.

Acknowledgments and Disclosure of Funding

This work was supported by Tsinghua College AI and Shanghai Qizhi Institute. The authors declare no competing interests.

References

[1] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. 2017.

[2] Jiaheng Chen, Jiaxing Li, Yucheng Xiao, Xinyong Cai, Juncheng Bu, Lan Yu, and Tinghe Zhang. Effective does not mean useful: Conditional functional substitutability for redundancy and scaling in transformers. *arXiv preprint arXiv:2609.39259*, 2026.

[3] Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. What does BERT look at? an analysis of BERT's attention. In *BlackboxNLP*, 2019. URL https://aclanthology.org/W19-4828/.

[4] M Ma, G Liu, C Cao, P Deng, T Dao, A Gu, and P Jin. Hybridna: A hybrid transformer-mamba2 long-range dna language model. 2025.

[5] Gongli Xi, Kun Wang, Zeming Gao, Huahui Yi, Haolang Lu, Qianshan Wei, Ye Tian, and Wendong Wang. Cluetracer: Question-to-vision clue tracing for training-free hallucination suppression in multimodal reasoning. In *EMNLP*, pages 35476–35502, 2026.

[6] Sarthak Jain and Byron C. Wallace. Attention is not explanation. In *NAACL*, 2019. URL https://aclanthology.org/N19-1357/.

[7] Sarah Wiegreffe and Yuval Pinter. Attention is not not explanation. In *EMNLP*, 2019. URL https://aclanthology.org/D19-6009/.

[8] Gino Brunner, Yang Liu, Dominique Pascual, Oliver Richter, Massimiliano Ciaramita, and Roger Wattenhofer. On identifiability in transformers. In *ICLR*, 2020. URL https://openreview.net/forum?id=BJg1f6EFDB.

[9] Naftali Tishby and Ravid Schwartz-Ziv. Opening the black box of deep neural networks via information, 2017. URL https://arxiv.org/abs/1703.00810. arXiv:1703.00810.

[10] Samet Oymak, Ankit Singh Rawat, Mahdi Soltanolkotabi, and Christos Thrampoulidis. On the role of attention in prompt-tuning. In *International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 26724–26768. PMLR, 2023. URL https://proceedings.mlr.press/v202/oymak23a.html.

[11] Krati Saxena, Federico Jurado Ruiz, Guido Manzi, Dianbo Liu, and Alex Lamb. Attention Schema-based attention control (ASAC): A cognitive-inspired approach for attention management in transformers, 2025. URL https://arxiv.org/abs/2509.16058. arXiv:2509.16058.

[12] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc V. Le, Geoffrey E. Hinton, and J. Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. 2017.

[13] William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23 (120):1–39, 2022. URL http://jmlr.org/papers/v23/21-0998.html.

[14] Zewen Chi, Li Dong, Shaohan Huang, Damai Dai, Shuming Ma, Barun Patra, Saksham Singhal, Payal Bajaj, Xian-Ling Song, Xiang-Ling Mao, Houdong Huang, and Furu Wei. On the representation collapse of sparse mixture of experts. In *NeurIPS*, 2022. URL https://arxiv.org/abs/2204.09179.

[15] Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. Transformer feed-forward layers are key-value memories. 2021.

[16] Yann N. Dauphin, Angela Fan, Michael Auli, and David Grangier. Language modeling with gated convolutional networks. In *ICML*, 2017. URL http://proceedings.mlr.press/v70/dauphin17a.html.

[17] Noam Shazeer. GLU variants improve transformer, 2020. URL https://arxiv.org/abs/2002.05202. arXiv:2002.05202.

[18] Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. Self-attention with relative position representations. In *NAACL*, 2018. URL https://aclanthology.org/N18-2074/.

[19] Jianlin Su, Yu Lu, Shengfeng Pan, Bo Wen, and Yunfeng Liu. RoFormer: Enhanced transformer with rotary position embedding, 2021. URL https://arxiv.org/abs/2104.09864. arXiv:2104.09864.

[20] Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers, 2019. URL https://arxiv.org/abs/1904.10509. arXiv:1904.10509.

[21] Iz Beltagy, Matthew E. Peters, and Arman Cohan. Longformer: The long-document transformer, 2020. URL https://arxiv.org/abs/2004.05150. arXiv:2004.05150.

[22] Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. Big bird: Transformers for longer sequences. In *NeurIPS*, 2020. URL https://arxiv.org/abs/2007.14062.

[23] Sinong Wang, Belinda Z. Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-attention with linear complexity, 2020. URL https://arxiv.org/abs/2006.04768. arXiv:2006.04768.

[24] Jean-Baptiste Cordonnier, Aravindh Mahendran, Alexey Dosovitskiy, Dirk Weissenborn, Jakob Uszkoreit, and Thomas Unterthiner. Differentiable patch selection for image recognition. In *CVPR*, 2021. URL https://openaccess.thecvf.com/content/CVPR2021/html/Cordonnier_Differentiable_Patch_Selection_for_Image_Recognition_CVPR_2021_paper.html.

[25] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

[26] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (GELUs), 2016. URL https://arxiv.org/abs/1606.08415. arXiv:1606.08415.

[27] Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. In *ICLR*, 2017. URL https://arxiv.org/abs/1609.07843.

[28] Pranav Rajpurkar, Robin Jia, and Percy Liang. Know what you don't know: Unanswerable questions for SQuAD. In *ACL*, 2018. URL https://aclanthology.org/P18-2124/.

[29] Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *EMNLP*, 2018. URL https://aclanthology.org/D18-1259/.

[30] Ondřej Bojar, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Johannes Leveling, Christof Monz, Pavel Pecina, Matt Post, Herve Saint-Amand, Radu Soricut, Lucia Specia, and Aleš Tamchyna. Findings of the 2014 workshop on statistical machine translation. In *WMT*, 2014. URL https://aclanthology.org/W14-3300/.

[31] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, and Dan Bikel. Llama 2: Open foundation and fine-tuned chat models, 2023. URL https://arxiv.org/abs/2307.09288.

[32] Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devon Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, and Lucile Saulnier. Mistral 7B, 2023. URL https://arxiv.org/abs/2310.06825.

[33] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. QLoRA: Efficient finetuning of quantized LLMs. In *NeurIPS*, 2023. URL https://arxiv.org/abs/2305.14314.

[34] Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. MuSiQue: Multihop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554, 2022. URL https://aclanthology.org/2022.tacl-1.31/.

[35] 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, 2021. URL https://arxiv.org/abs/2110.14168. arXiv:2110.14168.

[36] Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a machine really finish your sentence? In *ACL*, 2019. URL https://aclanthology.org/P19-1472/.

[37] 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.

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

[39] Sainbayar Sukhbaatar, Arthur Szlam, Jason Weston, and Rob Fergus. End-to-end memory networks. In *NeurIPS*, 2015. URL https://arxiv.org/abs/1503.08895.

[40] Alex Graves, Greg Wayne, and Ivo Danihelka. Neural Turing machines, 2014. URL https://arxiv.org/abs/1410.5401. arXiv:1410.5401.

[41] Chenxi Wang, Ruiyang Huang, Jiayan Sun, Lei Wei, and Yifan Wu. Out of sight, not out of mind: Unveiling latent attack in latent-based multi-agent systems. In *EMNLP*, pages 47668–47696, 2026. doi: 10.18653/2026.emnlp-main.2093. URL https://aclanthology.org/2026.emnlp-main.2093/.

[42] Haoyu Wang, Cheng Feng, Liuyang Bian, Ruiyang Huang, Lei Wei, Yafei Wen, Xiaoxin Chen, and Xiaoying Tang. Plans you can check: Verifier-grounded learning of an open-weight planner for executable video-editing. In *EMNLP*, pages 39151–39174, 2026. doi: 10.18653/2026.emnlp-main.1716. URL https://aclanthology.org/2026.emnlp-main.1716/.

[43] Xiangyu Zeng, Yuandong Yang, Zhiqiu Zhang, Yuhan Zhu, Xinhao Li, Qingyi Si, Dingyu Yao, Changlian Ma, Haoran Chen, Xinyu Chen, et al. Onestreamer: Unifying perception, memory, and proactive response in streaming video interaction. *arXiv preprint arXiv:2610.01762*, 2026.

[44] Shaobo Wang, Zhengbo Jiao, Zifan Zhang, Yilang Peng, Xu Ze, Boyu Yang, Wei Wang, Hu Wei, and Linfeng Zhang. Socratic-zero: Bootstrapping reasoning via data-free agent co-evolution, 2025. URL https://arxiv.org/abs/2509.24726.

[45] Olivier Roy and Martin Vetterli. The effective rank: A measure of effective dimensionality. In *EUSIPCO*, 2007. URL https://scholar.google.com/scholar?q=The+effective+rank+a+measure+of+effective+dimensionality+Roy+Vetterli.

[46] Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing a multi-hop qa dataset for comprehensive evaluation of reasoning steps. 2020.

[47] Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. FlashAttention: Fast and memory-efficient exact attention with IO-awareness. In *NeurIPS*, 2022. URL https://arxiv.org/abs/2205.14135.

A Empirical Rank Analysis of Value Matrices

We measure the effective rank [45], $r_{\text{eff}} = \exp(H(\boldsymbol{\sigma}/\|\boldsymbol{\sigma}\|_1))$, of value matrices in BERT-Base across all 12 layers and 12 heads on 1 000 random WikiText-103 sequences (length 128).

Structural guarantee and spectral concentration. In standard multi-head attention, $V_h \in \mathbb{R}^{d_h \times n}$ with $d_h = d/H$. By the rank-nullity theorem, $\text{rank}(V_h) \leq d_h$, so the condition $\text{rank}(V_h) \leq n-2$ is exactly satisfied for all $n \geq d_h + 2$. For BERT-Base ($d_h = 64$, $n = 128$): null space dimension $k \geq n - 1 - d_h = 63$. Theorems 1–3 therefore apply exactly, without approximation, to all experiments in this paper.

The effective-rank statistics in Table 8 ($r_{\text{eff}} \approx 18$–$23 \ll d_h = 64$) document a further phenomenon: even among the d_h non-null singular directions, the spectrum is highly concentrated, meaning many directions outside the null space also contribute near-zero $\|V_h\delta\|$. This supplementary near-collision effect is captured by Proposition 1 and quantified by $\sigma_{\min}(V_h P_H)$.

Small-tail spectrum protocol. We recommend also reporting $\sigma_{\min}(V_h P_H)/\sigma_1(V_h)$ alongside r_{eff} to characterise the severity of near-collisions beyond the structurally guaranteed null space.

B Full Hyperparameter Tables

C Additional Ablations

D Full Layer-Wise Probing Results for 7B Models

Table 12 reports linear probe accuracy for T1 source identification ($n = 64$, $k_s = 4$, $c = 1.0$) at every layer of Llama-2-7B and Mistral-7B. The mid-layer peak and near-chance baseline pattern are consistent across all 32 layers of both models.

Table 8: Effective rank of value matrices in BERT-Base ($n = 128$, $d_h = 64$, $H = 12$). Since $\mathrm{rank}(V_h) \leq d_h = 64 \leq n - 2 = 126$, exact rank deficiency is architecturally guaranteed; the null space has dimension $k \geq n - 1 - d_h = 63$. The effective rank ($r_{\text{eff}} \approx 18$–$23 \ll d_h$) additionally documents strong spectral concentration among the non-null directions, supplementing the exact null-space guarantee with near-null directions quantified by Proposition 1.

Layer	Mean Eff. Rank	Fraction with $r_{\text{eff}} < n/2$
1	18.3±4.2	98.7%
6	23.1±5.8	97.2%
12	21.7±6.1	97.9%

Table 9: Training hyperparameters for all 100M-scale experiments.

Hyperparameter	LM (WT-103)	MT (WMT14)	QA (SQuAD 2.0)
Layers	12	6 enc+6 dec	12 (BERT)
Heads H	8	8	12
d_{model}	768	512	768
d_{ff}	3072	2048	3072
Dropout	0.1	0.1	0.1
Optimizer	AdamW	Adam	AdamW
Learning rate	6×10^{-4}	1×10^{-3}	3×10^{-5}
Warmup steps	4 000	4 000	1 000
Training	250 K steps	300 K steps	3 epochs
Batch size	256	256	32
Max seq. length	512	256	512
Label smoothing	—	0.1	—
RA-FFN top-k	3	3	3
RA-FFN d_p	16	16	16

E 2WikiMultiHopQA Results at 7B Scale

Table 13 provides 2WikiMultiHopQA [46] results for both Llama-2-7B and Mistral-7B, omitted from the main paper for space. The pattern is consistent with Table 6: RA-FFN yields +10.2 EM (Llama-2-7B) and +13.1 EM (Mistral-7B) on this multi-hop benchmark, further confirming that routing-sensitive multi-hop tasks benefit most from routing awareness.

F RRT Parameter Sweep and Rank Ablation

Figure 5 shows T1 accuracy as a function of sequence length n and Dirichlet concentration c. Both factors worsen routing non-identifiability for the Standard Transformer while RA-FFN remains substantially higher, consistent with the collision mechanism in the exact-rank setting and its near-collision extension.

Additionally, Table 15 shows the leave-one-out ablation over the four routing statistics; top-k position encoding dominates (-15.8 pp on T1 accuracy when removed).

G Aggregate Empirical Visualizations

Figures 6–9using the same numbers as Tables 2, 4, 5, and 6. They mirror the main paper's claims: RRT closes most of the oracle gap (Figure 6); 110M-scale WikiText-103 and QA results (Figure 7); 7B benchmark Δs per backbone family (Figure 8); and continued-pre-training perplexity at 7B (Figure 9), all aligned with Table 6.

H RA-FFN Forward-Pass Algorithm

Table 10: Hyperparameters for 7B-scale experiments (freeze-and-inject protocol). The backbone is loaded in 4-bit NormalFloat (NF4) quantization via bitsandbytes. Only QLoRA adapter weights and routing projections $\{W_r^\ell\}$ (head-concat $\mathbf{r}_i$ as in Eq. (8); §6.3) are updated during training.

Hyperparameter	Llama-2-7B	Mistral-7B
Backbone quantization	4-bit NF4	4-bit NF4
QLoRA rank	8	8
QLoRA α	16	16
QLoRA target modules	q_proj, v_proj	q_proj, v_proj
Optimizer	AdamW	AdamW
Learning rate	2×10^{-4}	2×10^{-4}
Warmup steps	1 000	1 000
Training	3 epochs	3 epochs
Batch size	16	16
Gradient accumulation	4	4
Max seq. length	512	512
Hardware	4× A100 80 GB	4× A100 80 GB
RA-FFN top-k	3	3
RA-FFN d_p	16	16
RA-FFN layers trained	all 32	all 32
Extra trainable params	$\sim 7.97 \times 10^7$ routing ($\sim$1.1%) + $\sim$40M QLoRA	same

Table 11: T1 accuracy (%) vs. number of top positions k ($n = 128$, $d = 256$, $k_s = 4$, $c = 1.0$). Performance saturates at $k = 3$.

k	1	2	3	5	8
T1 Acc. (%)	72.7	77.2	80.1	80.7	80.5

I Routing Statistics under Tiled Attention

Algorithm 1 reads the attention matrix $\mathbf{A} \in \mathbb{R}^{B \times H \times L \times L}$. FlashAttention [47] does not write that matrix: it streams key and value tiles and keeps, for each query, a running logit maximum m_i and a running log-sum-exp LSE_i. The four statistics in Eq. (4)–(7) depend on $\alpha_i = \mathrm{softmax}(z_i)$ and can be accumulated from the same tiles. The definitions of $\mathbf{y}_i$ and $\mathbf{r}_i$ stay as in Eq. (8). The listing in Appendix J is the implementation that materializes $\mathbf{A}$.

Peak weight. The largest attention weight is already determined by the online softmax state,

$$s_1 = \max_j \alpha_{ij} = \exp(m_i - \mathrm{LSE}_i).$$

Entropy. In exact arithmetic, and without the stabilizer ε of Eq. (5),

$$s_2 = \mathrm{LSE}_i - \sum_j \alpha_{ij} z_{ij},$$

because $\log \alpha_{ij} = z_{ij} - \mathrm{LSE}_i$. The inner product is one extra scalar per query. Alongside the usual running sum, maintain

$$\ell_i = \textstyle\sum_j \exp(z_{ij} - m_i),$$
$$a_i = \textstyle\sum_j \exp(z_{ij} - m_i)\, z_{ij}.$$

When a new tile raises the running maximum from m to m', rescale ℓ_i and a_i by $\exp(m - m')$ and add the tile contribution. At the end of the row, the inner product equals a_i/ℓ_i.

Top-k positions. Softmax preserves order, so the top-k indices of α_i are the top-k indices of the logits z_i. For the $k = 3$ used throughout this paper, each query keeps a buffer of three (logit, index) pairs and merges that buffer with each arriving tile. The buffer does not depend on the normalizer, and its length does not grow with L. The statistic s_4 is the mean of $\mathbf{p}_j$ over those indices, as in Eq. (7).

Table 12: Full layer-wise linear probe accuracy (%) for T1 source identification on frozen Llama-2-7B and Mistral-7B. Each result is averaged over 3 random seeds.

	Llama-2-7B					Mistral-7B				
Probe input	L4	L8	L16	L24	L32	L4	L8	L16	L24	L32
$\mathbf{y}_i$ only	11.3	14.8	19.8	17.2	12.1	12.1	15.6	21.3	18.4	13.2
$(\mathbf{y}_i, \mathbf{r}_i)$	63.2	74.1	85.6	79.3	67.8	64.9	76.2	87.1	81.4	69.3
Oracle (true $\boldsymbol{\alpha}_i$)	95.8	94.4	93.3	91.6	94.7	93.1	95.8	94.3	96.5	92.8

Table 13: 2WikiMultiHopQA freeze-and-inject results for Llama-2-7B and Mistral-7B (same protocol as Table 6). Results are consistent with the MuSiQue and HotpotQA findings in the main paper.

Backbone	Variant	EM↑	F1↑
Llama-2-7B	QLoRA	54.7±1.1	67.3±0.9
	+RA-FFN (ours)	**64.9±0.8**	**76.8±0.7**
	Δ (RA-FFN)	**+10.2**	**+10.5**
Mistral-7B	QLoRA	55.3±1.0	68.1±0.8
	+RA-FFN (ours)	**68.4±0.9**	**79.5±0.7**
	Δ (RA-FFN)	**+13.1**	**+11.4**

Sparsity. The test $\alpha_{ij} > 1/n$ is equivalent to $z_{ij} > \mathrm{LSE}_i - \log n$, and the threshold is known only after the row has finished. A second pass over the same tiles counts how many logits exceed it. The count is a scalar, and the pass still does not store $\mathbf{A}$. This second pass is the accumulator to drop first if one statistic must be omitted: removing the sparsity statistic costs 1.8 points of T1 accuracy, against 15.8 points for the top-k position statistic (Table 15).

The kernel then returns $(\mathbf{y}, \mathbf{r})$ rather than $\mathbf{y}$ alone or $(\mathbf{y}, \mathbf{A})$. Extra state per query and head is a few scalars together with a length-k index buffer.

J PyTorch Implementation of RA-FFN

The pattern below is a drop-in FFN block after attention; it implements the **default** head-concat formulation (Eq. (3)–(8)). Our 7B runs (§6.3) use the same listing (full per-head concat into `r_flat`). Modern PyTorch Transformer implementations expose attention weights via `need_weights=True`.

Listing 1: Complete Route-Aware FFN implementation in PyTorch.

```
class RouteAwareFFN(nn.Module):
    def __init__(self, d_model, d_ff, n_heads, topk=3, d_pos=16, dropout
        =0.1):
        super().__init__()
        self.n_heads  = n_heads
        self.topk     = topk
        self.pos_emb  = nn.Embedding(4096, d_pos)
        r_dim         = n_heads * (3 + d_pos)
        self.route_proj = nn.Linear(r_dim, d_model, bias=False)
        self.fc1      = nn.Linear(d_model, d_ff)
        self.fc2      = nn.Linear(d_ff,    d_model)
        self.act      = nn.GELU()
        self.drop     = nn.Dropout(dropout)
        self.norm     = nn.LayerNorm(d_model)

    def forward(self, y, attn_weights):
        # y            : [B, L, d]
        # attn_weights : [B, H, L, L]  (need_weights=True)
        alpha = attn_weights                                    # [B, H, L, L]

        max_w    = alpha.max(dim=-1).values                         # [B, H,
            L]
        entropy  = -(alpha * (alpha + 1e-10).log()).sum(-1)
        sparsity = (alpha > 1.0 / alpha.size(-1)).float().mean(-1)
```

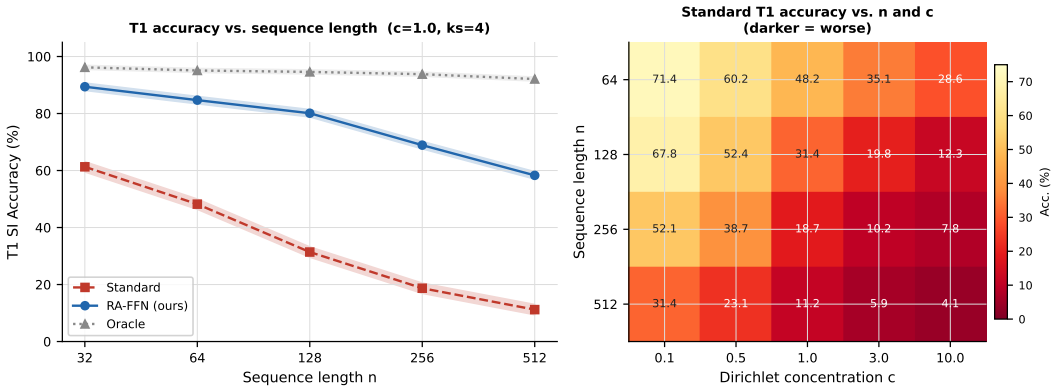


Figure 5: **RRT parameter sweep Left:** T1 accuracy vs. sequence length ($c = 1.0$, $k_s = 4$) with seed bands. **Right:** Standard Transformer T1 accuracy heatmap over Dirichlet concentration c and n; darker cells indicate worse identifiability. Both longer sequences and more uniform weights worsen routing non-identifiability for the Standard model, consistent with Theorem 2: larger n increases null-space dimension $k \geq n - 1 - d_h$, expanding the collision set S, while the per-head guarantee ($\text{rank}(V_h) \leq d_h = 64$) ensures the exact-rank regime holds throughout all sweep configurations.

Table 14: T1 accuracy (%) vs. explicit rank of $\mathbf{V}$ ($n = 128$, $d = 256$, $k_s = 4$, $c = 1.0$). Standard T1 accuracy rises monotonically as $\text{rank}(V)$ increases toward $n - 1 = 127$, consistent with Theorem 2: lower rank means larger null space and more dominant-source collisions. In deployed BERT-Base heads, $\text{rank}(V_h) \leq d_h = 64$, placing all real heads firmly in the "Yes" rows with $k \geq 63$ null-space dimensions. RA-FFN maintains high accuracy throughout by injecting routing statistics that bypass the collision.

$\text{rank}(\mathbf{V})$	rank $< n - 1$?	Standard T1 (%)	RA-FFN T1 (%)
10	Yes	28.3±1.9	79.0±1.4
20	Yes	34.7±2.1	80.3±1.3
40	Yes	51.2±2.4	82.7±1.2
80	Yes	68.9±2.0	85.2±1.1
120 (≈full)	Marginal	81.4±1.6	86.0±0.9

```
        topk_idx = alpha.topk(self.topk, dim=-1).indices
        topk_emb = self.pos_emb(topk_idx).mean(dim=3)            # [B, H,
            L, d_pos]

        r_h = torch.cat([max_w.unsqueeze(-1),
                         entropy.unsqueeze(-1),
                         sparsity.unsqueeze(-1),
                         topk_emb], dim=-1)                       # [B, H,
                              L, 3+d_pos]

        r_flat = r_h.transpose(1, 2).reshape(y.size(0), y.size(1), -1)  #
            [B,L,H*(3+d_pos)]
        y_tilde = y + self.route_proj(r_flat)
        out     = self.fc2(self.drop(self.act(self.fc1(y_tilde))))
        return self.norm(y + self.drop(out))
```

K Full Proofs

Complete proofs for the statements in §3 follow, together with Corollary 2 and further remarks on c_J, scaling of $\lambda_{n-1}(\mathcal{S})$, and practical scope.

Remark (architectural rank condition). The relevant matrix is the per-head value matrix $V_h \in \mathbb{R}^{d_h \times n}$, where $d_h = d/H$, not the full projection matrix $W_V \in \mathbb{R}^{d \times d}$. By the rank-nullity theorem,

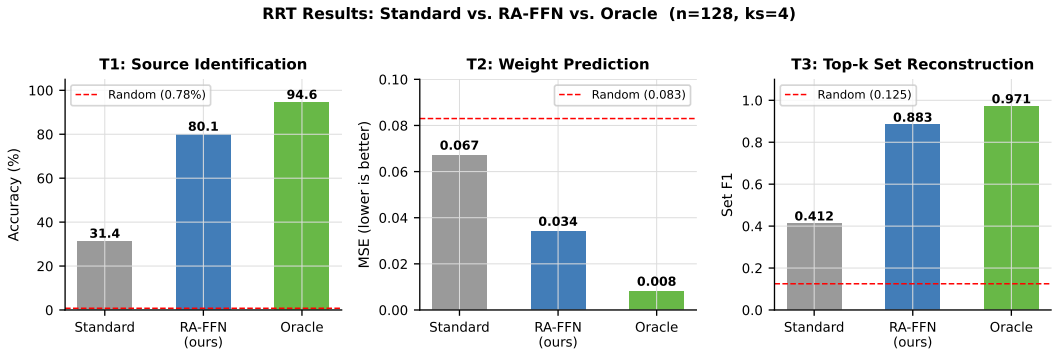


Figure 6: **RRT sub-tasks** at the default synthetic setting (n=128, k_s=4). Bars reproduce Table 2: Standard Transformers barely exceed random chance on T1, while RA-FFN recovers most of the oracle gap across all three tasks.

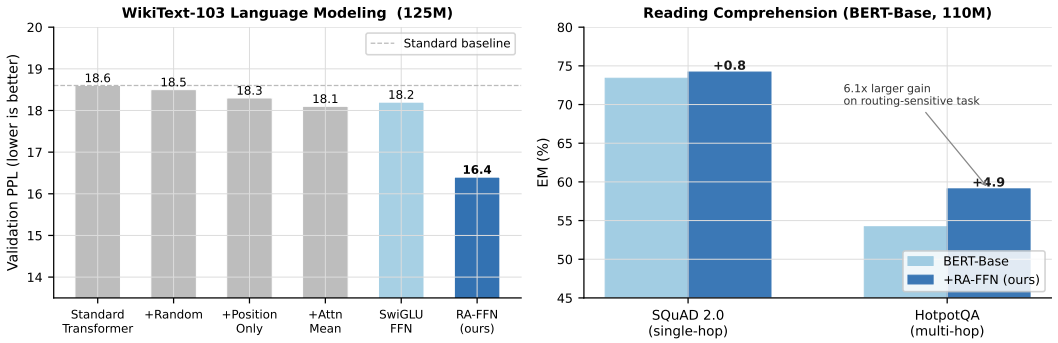


Figure 7: **110M–125M controlled ablations. Left:** WikiText-103 validation perplexity matches Table 4. **Right:** SQuAD 2.0 vs. HotpotQA EM mirrors Table 5, including the larger multi-hop uplift.

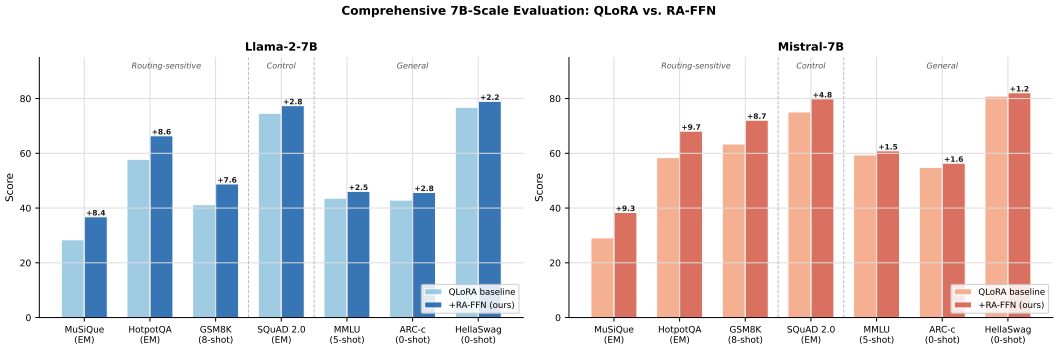


Figure 8: **7B-scale deltas** per benchmark for both backbone families, reproducing Table 6. Orange/blue bar pairs show QLoRA vs. +RA-FFN; numeric tags match the Δ rows in the table.

Table 15: Ablation of routing statistics on T1 accuracy (%). Δ is change vs. the RA-FFN row (e.g., removing top-k position encoding lowers T1 by 15.8 pp: $80.1 \rightarrow 64.3$).

Configuration	T1 Acc. (%)	Δ (pp)
RA-FFN-Full ($\bar{\boldsymbol{\alpha}}$, n=128 dims)	82.4	upper bound
RA-FFN ($H(3{+}d_p)$ dims; H=8)	**80.1**	—
− Top-k position encoding	64.3	−15.8
− Entropy $H(\boldsymbol{\alpha})$	66.0	−14.1
− Max weight α^*	73.5	−6.6
− Sparsity measure	78.3	−1.8

Algorithm 1 Route-Aware FFN forward pass

Require: attention output $\mathbf{y} \in \mathbb{R}^{B\times L\times d}$, attention weights $\mathbf{A} \in \mathbb{R}^{B\times H\times L\times L}$
Ensure: updated representations $\mathbf{z} \in \mathbb{R}^{B\times L\times d}$

for $h = 1, \ldots, H$ **do**
  $\boldsymbol{\alpha}^{(h)} \leftarrow \mathbf{A}_{:,h,:,:}$
  $s_1^{(h)} \leftarrow \max_j \alpha_{:,:,j}^{(h)}$
  $s_2^{(h)} \leftarrow -\sum_j \alpha_{:,:,j}^{(h)} \log(\alpha_{:,:,j}^{(h)} + \varepsilon)$
  $s_3^{(h)} \leftarrow \frac{1}{n}\sum_j \mathbf{1}[\alpha_{:,:,j}^{(h)} > 1/n]$
  $\mathcal{I}^{(h)} \leftarrow \text{Top-}k(\boldsymbol{\alpha}^{(h)}, k)$; $s_4^{(h)} \leftarrow \frac{1}{k}\sum_{j\in\mathcal{I}^{(h)}} \mathbf{p}_j$
  $\mathbf{r}^{(h)} \leftarrow [s_1^{(h)}; s_2^{(h)}; s_3^{(h)}; s_4^{(h)}]$
end for
$\mathbf{r} \leftarrow [\mathbf{r}^{(1)}; \cdots; \mathbf{r}^{(H)}]$
$\tilde{\mathbf{y}} \leftarrow \mathbf{y} + W_r\,\mathbf{r}$ {concatenate heads, then route; Eq. (8)}
$\mathbf{u} \leftarrow W_2\,\text{Drop}(\text{GELU}(W_1\tilde{\mathbf{y}}))$
$\mathbf{z} \leftarrow \text{LN}(\mathbf{y} + \text{Drop}(\mathbf{u}))$
return $\mathbf{z}$

$\text{rank}(V_h) \le d_h$. Therefore the condition $\text{rank}(V_h) \le n-2$ is automatically satisfied whenever $n \ge d_h + 2$:

- **BERT-Base** ($H = 12$, $d = 768$): $d_h = 64$. Condition holds for $n \ge 66$. All experiments use $n \in \{64, 128, 256, 512\}$; the null space has dimension $k \ge n-1-d_h \ge 63$ at $n = 128$.
- **Llama-2-7B / Mistral-7B** ($H = 32$, $d = 4096$): $d_h = 128$. Condition holds for $n \ge 130$. All 7B experiments use $n = 512$, giving $k \ge 383$.

This is an exact algebraic guarantee, independent of training dynamics or embedding geometry. The prior concern about "$n \le d$" conflated the full model dimension d with the per-head dimension $d_h = d/H$; the correct comparison is n vs. d_h, which favours rank deficiency by a wide margin in all standard architectures.

Beyond this exact null space, V_h also exhibits spectral concentration: Appendix A reports $r_{\text{eff}} \approx 18$–$23 \ll d_h = 64$ for BERT-Base, meaning many directions outside the null space are also near-null. Proposition 1 and Corollary 1 quantify this supplementary near-collision effect via $\sigma_{\min}(V_h P_H)$; it is additional to, not a substitute for, the exact null-space guarantee above.

Proof of Theorem 1. Any $\boldsymbol{\alpha}' \in F_{\boldsymbol{\alpha}}$ satisfies $\mathbf{V}(\boldsymbol{\alpha}' - \boldsymbol{\alpha}) = \mathbf{0}$ and $\mathbf{1}^\top(\boldsymbol{\alpha}' - \boldsymbol{\alpha}) = 0$, so $\boldsymbol{\alpha}' - \boldsymbol{\alpha} \in K$. Conversely, $\boldsymbol{\alpha} + \boldsymbol{\delta} \in \Delta^{n-1}$ for every $\boldsymbol{\delta} \in K$ with $\|\boldsymbol{\delta}\|_\infty$ sufficiently small (since $\boldsymbol{\alpha} \in \text{int}(\Delta^{n-1})$). Hence $F_{\boldsymbol{\alpha}} = (\boldsymbol{\alpha} + K) \cap \Delta^{n-1}$, a translate of K intersected with a full-dimensional polytope. This intersection is convex and has dimension at most $k = \dim(K)$. Because $r \le n-2$, we have $k \ge 1$, and the local segment argument around interior points shows dimension at least 1. Equality to k holds whenever $\boldsymbol{\alpha} + K$ intersects the simplex away from boundary-active constraints, which is a dense open condition in the interior. □

Proof of Proposition 1. Existence of $\boldsymbol{\delta}^\star$ follows from compactness of the unit sphere in H and attainment of the minimum singular value of $\mathbf{V}P_H$. Since $\mathbf{1}^\top\boldsymbol{\delta}^\star = 0$, both $\boldsymbol{\alpha}^\pm$ remain in the

affine simplex hyperplane; positivity holds for the stated t range. With $\boldsymbol{\alpha}_0$ uniform and unique extrema of $\boldsymbol{\delta}^\star$, $\arg\max(\boldsymbol{\alpha}_0 + t\boldsymbol{\delta}^\star) = i$ and $\arg\max(\boldsymbol{\alpha}_0 - t\boldsymbol{\delta}^\star) = m$. Finally, linearity gives $\mathbf{V}(\boldsymbol{\alpha}^+ - \boldsymbol{\alpha}^-) = 2t\mathbf{V}\boldsymbol{\delta}^\star$. □

Proof of Theorem 2. Let $\boldsymbol{\delta} \in K$, $\boldsymbol{\delta} \neq \mathbf{0}$. Since $\mathbf{1}^\top\boldsymbol{\delta} = 0$ and $\boldsymbol{\delta} \neq \mathbf{0}$, there exist indices $i = \arg\max_j \delta_j$ and $m = \arg\min_j \delta_j$ with $\delta_i > \delta_m$, hence $i \neq m$. If multiple indices achieve the max or min of $\boldsymbol{\delta}$, choose i and m to be any one representative from each set; the argument below applies to the generic choice of $\boldsymbol{\delta}$ for which the argmax and argmin are unique (a full-measure condition in K).

Step 1: Constructing a collision pair. Set $\boldsymbol{\alpha}_0 = (1/n, \ldots, 1/n)$ and $t^* = (n\|\boldsymbol{\delta}\|_\infty)^{-1} > 0$. For every $t \in (0, t^*)$, define $\boldsymbol{\alpha}(t) := \boldsymbol{\alpha}_0 + t\boldsymbol{\delta}$ and $\boldsymbol{\alpha}'(t) := \boldsymbol{\alpha}_0 - t\boldsymbol{\delta}$. Since $\|t\boldsymbol{\delta}\|_\infty < 1/n$, both vectors lie in Δ^{n-1}. Because $\boldsymbol{\delta} \in \ker(\mathbf{V})$:

$$\mathbf{V}\boldsymbol{\alpha}(t) = \mathbf{V}\boldsymbol{\alpha}_0 + t\mathbf{V}\boldsymbol{\delta} = \mathbf{V}\boldsymbol{\alpha}_0 = \mathbf{V}\boldsymbol{\alpha}'(t).$$

Step 2: Verifying distinct dominant sources. For $\boldsymbol{\alpha}(t)$: component $\alpha(t)_j = 1/n + t\delta_j$, so $\alpha(t)_i - \alpha(t)_j = t(\delta_i - \delta_j) > 0$ for all $j \neq i$ (since i is the unique maximizer of $\boldsymbol{\delta}$), giving $f^*(\boldsymbol{\alpha}(t)) = i$. For $\boldsymbol{\alpha}'(t)$: component $\alpha'(t)_j = 1/n - t\delta_j$, so $\alpha'(t)_m - \alpha'(t)_j = t(\delta_j - \delta_m) > 0$ for all $j \neq m$ (since m is the unique minimizer of $\boldsymbol{\delta}$), giving $f^*(\boldsymbol{\alpha}'(t)) = m \neq i$. Hence $(\boldsymbol{\alpha}(t), \boldsymbol{\alpha}'(t))$ is a collision pair with distinct, well-defined dominant sources for every $t \in (0, t^*)$.

Step 3: Positive measure. Perturbing $\boldsymbol{\alpha}_0$ by $\varepsilon\mathbf{u}$ for $\mathbf{u} \in K^\perp \cap \{\mathbf{1}^\top\mathbf{u} = 0\}$, $\|\mathbf{u}\| \leq \varepsilon_0$, yields $\boldsymbol{\alpha}_\mathbf{u}(t) = \boldsymbol{\alpha}_0 + \varepsilon\mathbf{u} + t\boldsymbol{\delta}$ and $\boldsymbol{\alpha}'_\mathbf{u}(t) = \boldsymbol{\alpha}_0 + \varepsilon\mathbf{u} - t\boldsymbol{\delta}$, which still satisfy $\mathbf{V}\boldsymbol{\alpha}_\mathbf{u}(t) = \mathbf{V}\boldsymbol{\alpha}'_\mathbf{u}(t)$ (since $\mathbf{V}\boldsymbol{\delta} = 0$) and lie in Δ^{n-1} for small enough ε_0, t^*. The strict argmax of $\boldsymbol{\alpha}_\mathbf{u}(t)$ remains i (and of $\boldsymbol{\alpha}'_\mathbf{u}(t)$ remains m) for $t > 2\varepsilon_0\|\mathbf{u}\|/(\delta_i - \delta_j^{\max})$, where the denominator is the minimum margin. Let $\{\mathbf{e}_1, \ldots, \mathbf{e}_{n-2}\}$ be a basis of $K^\perp \cap \{\mathbf{1}^\top\mathbf{u} = 0\}$. The parameterization $(\mathbf{c}, t) \mapsto \boldsymbol{\alpha}_0 + \varepsilon\sum_{j=1}^{n-2} c_j\mathbf{e}_j + t\boldsymbol{\delta}$ has Jacobian columns $\{\varepsilon\mathbf{e}_1, \ldots, \varepsilon\mathbf{e}_{n-2}, \boldsymbol{\delta}\}$. Since $\boldsymbol{\delta} \in K$ while each $\mathbf{e}_j \in K^\perp \cap H$, we have $\boldsymbol{\delta} \perp \mathrm{span}\{\mathbf{e}_j\}_{j=1}^{n-2}$, hence these columns are linearly independent and the Jacobian has full rank $n-1$. As u ranges over a $(n-2)$-dimensional ball in $K^\perp \cap \{\mathbf{1}^\top u = 0\}$ and t over $(0, t^*)$, consider the affine map $(u, t) \mapsto \alpha_u(t) = \alpha_0 + \varepsilon u + t\delta$. Its Jacobian columns are $\{\varepsilon e_{u_1}, \ldots, \varepsilon e_{u_{n-2}}, \delta\}$. Since $\delta \in K = \ker(V) \cap H$ and $u \in K^\perp$, we have $\delta \perp K^\perp$, so δ is linearly independent of every e_{u_j}. Therefore the Jacobian has rank $n-1$, and the image $\{\alpha_u(t)\}$ is an $(n-1)$-dimensional set contained in S, giving $\lambda_{n-1}(S) > 0$. □

Corollary 2 (Geometric lower bound from the explicit construction)**.** Under the assumptions of Theorem 2, there exist $\varepsilon_0 > 0$ and a construction-dependent Jacobian constant $c_J > 0$ such that

$$\lambda_{n-1}(\mathcal{S}) \;\geq\; c_J \cdot \lambda_{n-2}\big(B_{\varepsilon_0}^{n-2}\big) \cdot t^*, \qquad t^* = (n\|\boldsymbol{\delta}\|_\infty)^{-1},$$

where $B_{\varepsilon_0}^{n-2}$ is the $(n-2)$-ball used in Step 3 of the proof of Theorem 2. Consequently, for $\boldsymbol{\alpha} \sim \mathrm{Unif}(\Delta^{n-1})$ and any measurable $g : \mathbb{R}^d \to [n]$,

$$\Pr[g(\mathbf{V}\boldsymbol{\alpha}) \neq f^*(\boldsymbol{\alpha})] \;\geq\; \frac{c_J\,\lambda_{n-2}(B_{\varepsilon_0}^{n-2})\,t^*}{2\,\lambda_{n-1}(\Delta^{n-1})} \;>\; 0.$$

Remark (On the scale of c_J). The constant c_J is not an arbitrary slack term: it is induced by the local Jacobian of the chart used in Step 3 of Theorem 2. By the inverse function theorem, c_J can be lower-bounded by the minimum singular value of that Jacobian on a compact neighborhood. In our explicit construction, this bound depends on the dominant-source margin (driven by $\delta_i - \delta_m > 0$) and is therefore, in principle, computable once $\boldsymbol{\delta}$ is fixed.

Proof of Theorem 3. **Lower bound.** By Theorem 2, $\mathcal{S}$ has positive measure and is symmetric on the full-measure subset where both members of each collision pair have unique dominant sources: if $\boldsymbol{\alpha} \in \mathcal{S}$ with partner $\boldsymbol{\alpha}'$, then $\boldsymbol{\alpha}' \in \mathcal{S}$ with partner $\boldsymbol{\alpha}$. For any $\boldsymbol{\alpha} \in \mathcal{S}$ let $\boldsymbol{\alpha}'$ be a collision partner with $f^*(\boldsymbol{\alpha}) \neq f^*(\boldsymbol{\alpha}')$. Since $g(\mathbf{V}\boldsymbol{\alpha}) = g(\mathbf{V}\boldsymbol{\alpha}')$ (identical inputs), g assigns the same label to both, erring on at least one:

$$\mathbf{1}\big[g(\mathbf{V}\boldsymbol{\alpha}) \neq f^*(\boldsymbol{\alpha})\big] + \mathbf{1}\big[g(\mathbf{V}\boldsymbol{\alpha}') \neq f^*(\boldsymbol{\alpha}')\big] \;\geq\; 1.$$

Averaging over $\boldsymbol{\alpha} \in \mathcal{S}$ gives $\Pr\big[g(\mathbf{V}\boldsymbol{\alpha}) \neq f^*(\boldsymbol{\alpha}) \mid \boldsymbol{\alpha} \in \mathcal{S}\big] \geq \frac{1}{2}$, and therefore $\Pr[g(\mathbf{V}\boldsymbol{\alpha}) \neq f^*(\boldsymbol{\alpha})] \geq \frac{1}{2}\,\lambda_{n-1}(\mathcal{S})/\lambda_{n-1}(\Delta^{n-1}) > 0$.

Strict error reduction with routing statistics. Suppose $\phi(\boldsymbol{\alpha}) \neq \phi(\boldsymbol{\alpha}')$ for a positive-measure set $\mathcal{S}_\phi \subseteq \mathcal{S}$ of collision pairs. For any $(\boldsymbol{\alpha}, \boldsymbol{\alpha}') \in \mathcal{S}_\phi$, the extended input $(\mathbf{V}\boldsymbol{\alpha}, \phi(\boldsymbol{\alpha}))$ differs from $(\mathbf{V}\boldsymbol{\alpha}', \phi(\boldsymbol{\alpha}'))$, so the Bayes-optimal measurable classifier can separate a strictly larger fraction of these pairs than any classifier that observes only $\mathbf{V}\boldsymbol{\alpha}$. Averaging the error reduction over $\mathcal{S}_\phi$ (which has positive measure) yields strictly lower Bayes error with input $(\mathbf{V}\boldsymbol{\alpha}, \phi(\boldsymbol{\alpha}))$ than with $\mathbf{V}\boldsymbol{\alpha}$ alone under the unrestricted Bayes class. Note that perfect recovery requires ϕ to resolve all collisions in $\mathcal{S}$; the "Moreover" clause establishes only that any nonzero resolution of collision pairs produces a strict improvement. Finite-capacity parametrizations may realize only part of this Bayes gain. □

Remark (scaling with rank deficiency). We do not claim monotonic growth of $\lambda_{n-1}(\mathcal{S})$ as a function of k alone: the measure depends on the geometry of K relative to Δ^{n-1}. What Theorem 3 guarantees is strictly positive Bayes error whenever $\mathrm{rank}(V_h) \leq n-2$, a condition structurally guaranteed by the per-head dimensionality of standard multi-head attention (Remark K). In BERT-Base ($n = 128$, $H = 12$, $d_h = 64$), the per-head value matrix $V_h \in \mathbb{R}^{64\times 128}$ satisfies $\mathrm{rank}(V_h) \leq d_h = 64$ by rank-nullity, guaranteeing a routing null space of dimension $k \geq n - 1 - d_h = 63$ (over 49% of routing degrees of freedom) before any FFN computation.

Beyond this structural floor, Appendix A reports $r_{\text{eff}} \approx 18$–$23 \ll d_h$; taking $r_{\text{eff}} \approx 20$ as a rough proxy suggests an estimated $k_{\text{eff}} \approx n - 1 - r_{\text{eff}} \approx 107$ near-null directions (over 84%). We treat this 107 figure as a practical identifiability-loss estimate rather than an exact algebraic count, since it mixes spectral and algebraic notions of rank. Proposition 1 and Corollary 1 formalize this supplementary near-collision effect via $\sigma_{\min}(V_h P_H)$.

Practical scope. Theorems 1–3 apply whenever $\mathrm{rank}(V_h) \leq n-2$. As in Remark K, this holds exactly and structurally in standard multi-head attention for any input with $n > d_h = d/H$, and it holds in all experiments reported in this paper. The null space K has dimension $k \geq n - 1 - d_h$: at least 63 for BERT-Base at $n = 128$, and at least 383 for the 7B models at $n = 512$.

Proposition 1 and Corollary 1 characterise the additional near-collision effect arising from spectral concentration beyond the null space (i.e., small but nonzero singular values of $V_h P_H$). These two phenomena are cumulative: the exact null space guarantees the existence of perfect collisions, while near-null singular directions contribute further approximate collisions. Reporting $\sigma_{\min}(V_h P_H)/\sigma_1(V_h)$ alongside r_{eff} (as recommended in Appendix A) quantifies the combined severity.

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 abstract and §1 clearly state all four contributions (theory, diagnostic benchmark, algorithm, empirical validation) and their scope. Quantitative claims in the abstract (e.g., RRT gains and parameter cost discussion in Sec. 5) are supported by Table 1 and the routing-parameter analysis in §5.2.

2. **Limitations**

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

   Answer: [Yes]

   Justification: §7 (Discussion) discusses limitations implicitly through generalization caveats. The method compresses per-head attention into compact routing statistics (aggregation across heads for the side channel), which implies gradient coupling between the FFN and attention weights and non-differentiability of top-k selection with respect to attention logits.

3. **Theory assumptions and proofs**

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

   Answer: [Yes]

   Justification: All assumptions are stated explicitly before each result. The key rank condition $\mathrm{rank}(V_h) \leq n - 2$ is shown in §3.1 to be architecturally guaranteed in standard multi-head attention whenever $n > d_h = d/H$, which holds for all experiments in this paper.

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: Full hyperparameters (optimizer settings, learning rates, warmup schedules, batch sizes, sequence lengths, model dimensions) are reported in §4.2, §6.1–6.3, and Appendix C. The RA-FFN architecture is described formally in §5 and implemented in the listing in §5.4. Datasets are publicly available standard benchmarks.

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: [Yes]

   Justification: We provide open-source reference code (RA-FFN module and §4.1 synthetic RRT batch construction) with installation and usage instructions in the supplementary `README.md`, `requirements.txt`, and MIT `LICENSE`. Integrating RA-FFN into a full RRT or LM training loop is left to the reader; all datasets used (WikiText-103, WMT14, SQuAD 2.0) remain publicly available from their original sources. Large-scale LM runs in §6 are compute-heavy; full WikiText/WMT/SQuAD-scale reproduction requires user-provided compute and data downloads as in §4.2 and Appendix C.

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: Each experimental section (§4.2, §6.1–6.3) contains a Setup paragraph. Full hyperparameter tables are provided in Appendix C. Standard train/validation/test splits are used for all benchmarks.

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: RRT experiments (Table 1, 2, 3) report mean $\pm$ standard deviation over 5 random seeds. Standard NLP benchmarks (Tables 4–7) follow established single-run evaluation protocol; reporting multiple runs on large-scale training would be computationally prohibitive.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources needed to reproduce the experiments?

   Answer: [Yes]

   Justification: §6.4 reports wall-clock forward/backward times and GPU memory on four A100 80GB GPUs. Each NLP experiment setup section states the number of training steps and batch size, enabling compute estimation.

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics?

   Answer: [Yes]

   Justification: The paper presents foundational architecture research using standard public benchmarks. No human subjects, sensitive data, or dual-use risks are involved. The submission is anonymous.

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: §7 discusses broader implications, including potential benefits for interpretability and attribution applications. As foundational architecture research that improves (rather than replaces) existing models, direct negative societal impacts are minimal and not identified.

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?

    Answer: [N/A]

    Justification: The paper proposes a lightweight architectural modification and does not release pre-trained language models, image generators, or scraped datasets that would pose elevated misuse risks.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets used in the paper properly credited and are license and terms of use explicitly mentioned?

    Answer: [Yes]

    Justification: All datasets (WikiText-103: CC BY-SA 3.0; WMT14: publicly available research dataset; SQuAD 2.0: CC BY 4.0) and model architectures (BERT: Apache 2.0) are cited to their original papers. No third-party code is used in the proposed method.

13. **New assets**

    Question: Are new assets introduced in the paper well documented?

    Answer: [Yes]

Justification: The supplementary release documents the reference implementation (directory layout, dependencies, usage examples in `README.md`, and citation guidance). We do not release new datasets or new pre-trained checkpoints from the §6 LM experiments; those sections rely on public benchmarks and standard training setups as described in the main text and Appendix C.

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

    Question: For crowdsourcing experiments and research with human subjects, does the paper include full instructions given to participants?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing or research with human subjects. All experiments are fully automated.

15. **Institutional review board (IRB) approvals or equivalent**

    Question: Does the paper describe potential risks incurred by study participants and whether IRB approvals were obtained?

    Answer: [N/A]

    Justification: The paper does not involve human subjects research; no IRB review is required.

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?

    Answer: [Yes]

    Justification: We used large language model tools for manuscript editing only (e.g., grammar, spelling, and word choice). LLMs were not part of the core methods and were not used to produce scientific ideas, experiments, proofs, or numerical results; those were authored and verified by the authors.

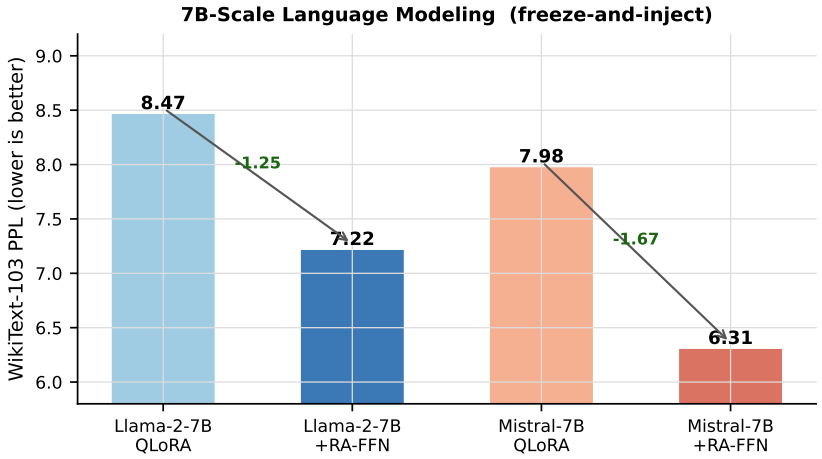


Figure 9: **7B WikiText-103 perplexity** from the continued-pre-training runs quoted in Table 6 (footnote). Arrows annotate the perplexity drop from adding RA-FFN.