Delta Attention Residuals

Selective depth readout alongside a preserved residual stream

Paper by Cheng Luo, Zefan Cai, and Junjie Hu

Research paper discussion | 4 October 2026
arXiv:2605.18855v1 (13 May 2026)

Versioned paper | Approximately 25 minutes + backup discussion

The useful design is an extra readout from depth

Core computation

$$\hat{\mathbf{h}}_l = \tilde{\mathbf{h}}_l + \underbrace{\sum_{i \in \mathcal{S}_l} \alpha_{i \to l} \mathbf{v}_i}_{\text{selected historical contributions}}$$

- Preserve the natural residual stream; route into the next sublayer.
- At 7.57B: validation PPL 17.43 $\rightarrow$ 16.00, with lower throughput.
- Separate measured gains from claims about novelty and initialization.

Paper, Eq. (2), PDF p. 3; Table 3, PDF p. 6.

Standard residuals expose a sum of earlier contributions

$$\mathbf{h}_{i+1} = \mathbf{h}_i + \mathbf{v}_i, \qquad \mathbf{v}_i = f_i(\mathbf{h}_i)$$

$$\mathbf{h}_l = \mathbf{h}_0 + \sum_{i<l} \mathbf{v}_i$$

- A sublayer is attention or MLP; f_i includes its Pre-Norm operation.
- Every past output enters the running sum with coefficient one.
- The paper asks whether a sublayer can selectively revisit a contribution.

Motivation

Shared prefixes can make cumulative states difficult to distinguish.

Paper, Secs. 2.1–2.2, PDF pp. 2–3. The geometric argument is a motivation.

Original AttnRes already routes over sublayer outputs

Original Full AttnRes explicitly defines its sources as

$$\mathbf{k}_i = \mathbf{v}_i = \begin{cases} \mathbf{h}_1, & i = 0, \\ f_i(\mathbf{h}_i), & 1 \leq i < l. \end{cases}$$

- Original Block AttnRes sums outputs *within* each block.
- These are not all full-history cumulative hidden states.
- Thus the present paper's baseline description misstates the prior work.

Reading implication

Focus on the additive residual path and the tested implementations.

External primary-source comparison: Attention Residuals, arXiv:2603.15031v1, Secs. 3.1–3.2, Eqs. (3), (5), PDF p. 4.

A learned query scores depth sources for each token

$$e_{i\to l} = \mathbf{w}_l^\top \,\mathrm{RMSNorm}(\mathbf{v}_i), \qquad \alpha_{i\to l} = \frac{\exp e_{i\to l}}{\sum_{j\in\mathcal{S}_l} \exp e_{j\to l}}$$

$$\mathbf{r}_l = \sum_{i\in\mathcal{S}_l} \alpha_{i\to l}\mathbf{v}_i$$

- $\mathbf{w}_l \in \mathbb{R}^d$: one learned query per receiving sublayer.
- Keys are normalized; values retain their original magnitudes.
- Queries are parameters, while source content makes weights token-dependent.
- Softmax runs over depth sources, independently at each sequence position.

Paper, Eqs. (1)–(2), PDF pp. 2–3; Fig. 3, PDF p. 4.

Readout changes the sublayer input, not the residual write

$$\hat{\mathbf{h}}_l = \tilde{\mathbf{h}}_l + \mathbf{r}_l \qquad \text{read from depth}$$

$$\mathbf{v}_l = f_l(\hat{\mathbf{h}}_l) \qquad \text{compute new contribution}$$

$$\tilde{\mathbf{h}}_{l+1} = \tilde{\mathbf{h}}_l + \mathbf{v}_l \qquad \text{update the original stream}$$

- Attention and MLP have separate routing queries and norms.
- Delta AttnRes appends each attention/MLP output to the source list.
- Adding $\mathbf{r}_l$ again in the last line would implement a different method.

Paper, Eq. (2), PDF p. 3; forward pseudocode, Fig. 3, PDF p. 4.

Block grouping reduces the number of selectable sources

With sublayer boundary indices τ_b, the intended block definition is

$$\Delta_b = \tilde{\mathbf{h}}_{\tau_{b+1}} - \tilde{\mathbf{h}}_{\tau_b} = \sum_{i=\tau_b}^{\tau_{b+1}-1} \mathbf{v}_i.$$

	Delta AttnRes	Delta Block
Selectable unit	One sublayer output	A grouped contribution
Source count	$O(2L)$	$O(K)$ blocks
Benefit	Finest selection	Lower readout cost
Tradeoff	More activations	Less granular selection

Residual preservation does not preserve individual access inside a block.

Paper, Appendix C, Eq. (3), PDF pp. 13–14. Here K is block count; the paper overloads B for count and width.

Block sources require a saved boundary state

Intended computation

```
delta = current - block_start
block_start = current
```

Boundary states: $1, 3, 6$.
Correct deltas: $2, 3$.

Published computation

```
delta = current - last_source
```

After storing $3 - 1 = 2$,
the next result is $6 - 2 = 4$.

- Fig. 3 and the public `delta_block` branch use the latter pattern.
- Formula and implementation need reconciliation before reproduction.

Paper, Fig. 3, PDF p. 4; Eq. (3), p. 13. Public code at fc11c3e, modeling_qwen3_attnres.py, lines 559–610 and 1104–1108. Scalar example is our analysis.

A zero query does not generally make readout an identity

For N available sources:

$$\mathbf{w}_l = 0 \implies \hat{\mathbf{h}}_l = \tilde{\mathbf{h}}_l + \frac{1}{N}\sum_i \mathbf{v}_i.$$

A zero-valued null source changes $1/N$ to $1/(N+1)$.

- The added mean need not vanish; a null source is not a zero gate.
- Complete source coverage can yield a scale change removable by ideal RMSNorm (backup).
- A gate $g_l = 0$ in $\hat{\mathbf{h}}_l = \tilde{\mathbf{h}}_l + g_l \mathbf{r}_l$ would guarantee identity.

Our substitution into Eq. (2). Paper's identity claim: Sec. 2.3, PDF p. 4, and Sec. 3.5, p. 8. Zero-gate construction is an analytical suggestion, not a main-table result.

Training is matched by steps and data within each scale

Architecture	d / L	Batch / length	Tokens in 10K steps
220M	768 / 12	32 / 1024	0.328B
533M	1024 / 24	32 / 1024	0.328B
1044M	1280 / 36	32 / 1024	0.328B
Qwen3-0.6B (508M)	1024 / 28	64 / 2048	1.311B
Qwen3-8B (7.57B)	4096 / 36	64 / 2048	1.311B

- FineWeb-Edu; AdamW; 500-step warmup + cosine decay.
- LR 6×10^{-4}; 3×10^{-4} at 7.57B.
- 8 H100s, BF16, `torch.compile`; 7.57B uses FSDP + checkpointing.
- Same steps do not mean the same wall-clock or compute budget.

Paper, Secs. 3.1 and 3.3, PDF pp. 4–6. Token totals calculated as steps $\times$ batch $\times$ length.

Delta variants reduce PPL at all three custom scales

Validation PPL ↓	220M	533M	1044M
Baseline	38.71	32.00	29.70
AttnRes (reimplementation)	37.39	31.75	31.76
Full AttnRes (reimplementation)	37.30	31.68	33.36
Delta Block	37.08	31.16	29.19
Delta AttnRes	**36.83**	**31.05**	**29.13**

- Delta AttnRes reductions vs. baseline: 4.86%, 2.97%, 1.92%.
- Delta Block stays close to the finest-grained variant.
- The tested replacement-routing baselines deteriorate at 1044M.

Paper, Table 1, PDF p. 5. Lower PPL is better; these are fixed-step comparisons.

At 1044M, grouping recovers much of the lost throughput

Method	PPL ↓	k token/s ↑	Peak GB / GPU
Baseline	29.70	108	22.5
AttnRes	31.76	71	31.6
Full AttnRes	33.36	49	52.0
Delta Block	29.19	86	28.4
Delta AttnRes	29.13	34	77.7

Block vs. per-sublayer delta routing

0.21% higher PPL, 2.53× throughput, 36.6% of the memory.

Versus baseline: throughput −20.4%, memory +26.2%.

Paper, Table 1, PDF p. 5. 8 H100s, BF16, compiled; peak memory measured at batch 4/GPU and length 1024. Ratios calculated from the table.

At 7.57B, lower PPL costs about 1.53× the time

Method	PPL ↓	k token/s ↑	Peak GB / GPU
Baseline	17.43	21.4	41.6
AttnRes	18.58	12.5	44.0
Delta Block	**16.00**	14.0	42.7

- Versus baseline: PPL -8.20%, throughput -34.6%, memory $+2.6\%$.
- At fixed token count, steady-state time ratio is $21.4/14.0 = 1.53$.
- 589.8K extra parameters ($\sim$0.008%) do not imply negligible runtime.
- Trained from scratch for 1.311B input tokens; only three variants tested.

Paper, Table 3 and Sec. 3.3, PDF pp. 6–7. 8 H100s, FSDP, BF16, compilation, gradient checkpointing.

More blocks do not guarantee better perplexity

220M, $L = 12$

Blocks	Sources	PPL ↓
1	2	37.44
2	3	37.08
4	5	36.98
6	7	36.92
12	13	37.34

Best block result near 4–6 blocks.

533M, $L = 24$

Blocks	Sources	PPL ↓
2	3	31.23
4	5	31.27
6	7	31.19
12	13	31.22
24	25	31.18

PPL varies by only 0.09.

More sources generally increase memory; throughput is not monotonic here.

Paper, Table 4, PDF p. 7. *B* means block count in this table. Table 1 and Table 4 contain different runs and should not be pooled.

Fine-tuning compares architecture–optimizer combinations

- Start from pretrained Qwen3-0.6B; continue on FineWeb-Edu.
- 20K steps, batch 32, 4 H100s, 500-step warmup.
- Delta Block uses a null source and separate parameter-group LRs.

Method	Backbone LR	Routing LR
Baseline FT	5×10^{-5}	—
AttnRes	5×10^{-5}	5×10^{-5}
Delta Block + null	5×10^{-5}	5×10^{-3}

The 100× routing LR and null source are not separately ablated.

Paper, Sec. 3.5, PDF p. 8. Eight 0-shot tasks evaluated with lm-evaluation-harness.

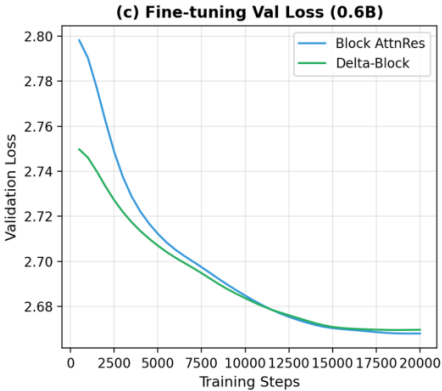
The +0.6-point mean gain is driven mainly by ARC-Easy

0-shot score (%)	Baseline FT	Delta Block	Change (pp)
HellaSwag	50.1	50.0	−0.1
ARC-Easy	61.5	**66.5**	+5.0
ARC-Challenge	36.7	37.3	+0.6
PIQA	69.2	69.4	+0.2
WinoGrande	57.0	56.7	−0.3
BoolQ	70.3	70.3	0.0
MMLU	45.2	44.8	−0.4
LAMBADA	50.0	49.8	−0.2
Mean	55.0	**55.6**	+0.6

3 wins, 1 tie, 4 losses vs. standard FT; AttnRes mean is 54.1%.

Paper, Table 5, PDF p. 8. Single training runs, no error bars (checklist, PDF pp. 17–18).

The plotted fine-tuning losses become very close

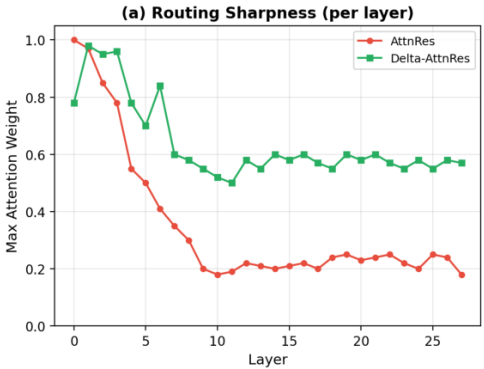


- Green is lower early; late losses nearly overlap.
- The graph contains no standard-FT baseline curve.
- Its 2.66–2.80 axis does not show the 3.96 spike claimed in the text.

This plot cannot establish exact identity initialization.

Paper, Fig. 5(c) and Sec. 3.5, PDF p. 8. Cropped original panel with axes and legend preserved.

Routing becomes more concentrated in the tested 0.6B model

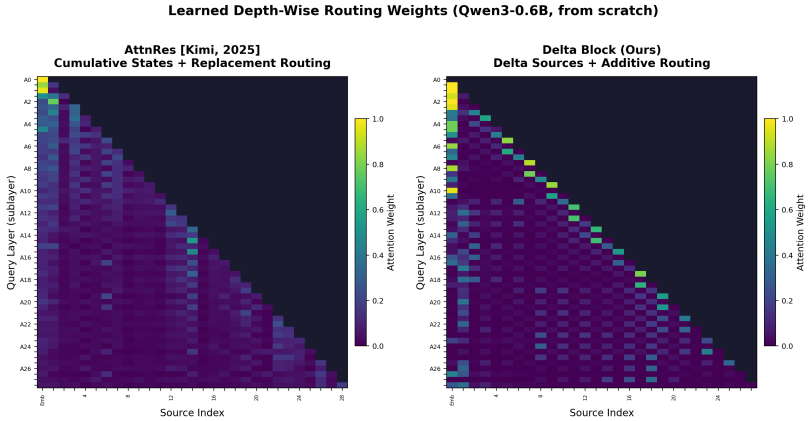


- Deep-layer maximum weights: roughly 0.2 vs. 0.6.
- Average maxima: 0.35 vs. 0.62 (1.77×).
- Validation PPL: 32.22 baseline, 32.38 AttnRes, 31.45 Delta Block.

Caption identifies Delta Block; the image legend says Delta-AttnRes.

Paper, Fig. 4(a), Table 2, PDF p. 6. Qwen3-0.6B architecture, $N = L = 28$, trained from scratch.

Heatmaps show localized choices and embedding reuse



Rows: receiving sublayers. Columns: sources. Color: attention weight (0–1).

Paper, Fig. 6, PDF p. 9. Right panel shows stronger local peaks; blank upper triangle is unavailable future depth.

The experiments support a design, not a unique cause

1. **Quality:** delta variants improve the reported fixed-step PPLs.
2. **Cost:** block grouping offers a useful compromise; 7.57B throughput still falls 34.6% vs. baseline.
3. **Interpretation:** source construction, additive paths, and optimization change together.

Most informative next comparison

Fix the source definition; compare additive vs. replacement readout with matched initialization, learning rates, and wall-clock budget.

Synthesis of Secs. 2–4. Proposed comparison is our analysis, not a reported experiment.

References and discussion

Main paper
Cheng Luo, Zefan Cai, and Junjie Hu.
Delta Attention Residuals. arXiv:2605.18855v1, 2026.
arxiv.org/abs/2605.18855v1

Targeted prior-work comparison
Kimi Team. *Attention Residuals*. arXiv:2603.15031v1, 2026.
arxiv.org/abs/2603.15031v1

Implementation snapshot
wdlctc/delta-attention-residuals-code, commit fc11c3e

All main results are taken from the paper. Formula checks and proposed controls are labeled as our analysis.

Backup: RMSNorm gives a conditional invariance argument

If the sources exactly partition the residual, including the embedding,

$$\sum_{i=1}^{N} \mathbf{s}_i = \tilde{\mathbf{h}}_l, \qquad \alpha_i = 1/N,$$

then

$$\hat{\mathbf{h}}_l = (1 + 1/N)\tilde{\mathbf{h}}_l.$$

- For positive c, $\mathrm{RMSNorm}(c\mathbf{h}) = \mathrm{RMSNorm}(\mathbf{h})$ when $\epsilon = 0$.
- The sublayer may therefore be unchanged despite nonzero readout.
- This requires complete coverage, correct source bookkeeping, and the normalization assumption.
- It is distinct from claiming that `depth_route` is itself an identity.

Our conditional derivation from Eq. (2); relevant paper claims on PDF pp. 4 and 8.

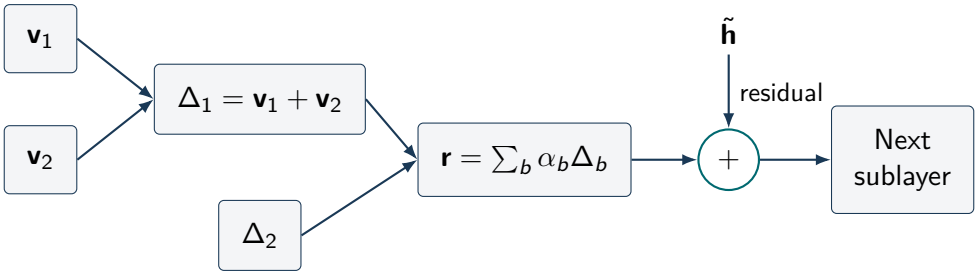
Backup: 0.2 is not uniform over roughly 28 sources

$$\max_i \alpha_i^{\text{uniform}} = \frac{1}{N}; \qquad N = 28 \implies 1/N \simeq 0.036.$$

- A maximum of 0.2 is diffuse relative to 0.6, but well above uniform.
- Candidate count grows with depth; maximum weight alone is incomplete.
- A normalized entropy $H(\alpha)/\log N$ would help compare depths.
- Larger peaks do not isolate whether source choice caused the PPL gain.

Our interpretation of Fig. 4 and Sec. 4.1, PDF pp. 6 and 8–9. Normalized entropy is a suggested diagnostic.

Backup: grouped sources trade resolution for memory



- The receiver can select a block sum, not its individual components.
- The direct residual input remains available alongside the readout.

Teaching schematic based on Eq. (3) and Fig. 8, PDF pp. 13–14; not an original paper figure. It illustrates the intended block definition, not the inconsistent public bookkeeping.

Backup: small-model efficiency has its own tradeoffs

Scale	Method	PPL	k token/s	Peak GB
220M	Baseline	38.71	289	5.6
	Delta Block	37.08	306	5.9
	Delta AttnRes	36.83	246	7.5
533M	Baseline	32.00	243	13.9
	Delta Block	31.16	198	16.1
	Delta AttnRes	31.05	80	33.6

Delta Block is faster than baseline at 220M in this table, but slower at larger scales. Parameter overhead alone does not predict throughput.

Paper, Table 1, PDF p. 5. All listed memory figures are per GPU.

Backup: do not merge Table 1 and Table 4 measurements

Scale	Delta Block row	PPL	k token/s	Peak GB
220M	Table 1	37.08	306	5.9
	Table 4, $B = 4$	36.98	343	9.1
533M	Table 1	31.16	198	16.1
	Table 4, $B = 4$	31.27	160	15.4

- The paper says $B = 4$ is the main-table default.
- The repository maps main and ablation entries to separate runs.
- Report each table on its own terms; measurement differences remain underexplained.

Paper, Tables 1 and 4, PDF pp. 5 and 7; public WANDB_RUNS.md at fc11c3e.