Decoupled Bottleneck Attention:

Low-Rank Semantic Routing as Structural Regularization

Daniel Owen van Dommelen
Independent Research
theapemachine@gmail.com

January 2026

Abstract

We propose **Decoupled Bottleneck Attention (DBA)**, an attention architecture that separates *semantic routing* (low-rank query–key interaction) from *positional geometry* (higher-rank routing), with RoPE applied only to the geometric path.

At 1B parameters (100k steps on FineWeb-Edu), DBA achieves a **37.5% KV-cache reduction** and **12% faster cached decode** while increasing held-out perplexity by **6%** (12.76 → 13.53). Under the behavioral evaluations used here, we observe **no evidence of capability collapse**: on 117 probes, accuracy is comparable (26.5% vs 27.4%); on an expanded 490-probe suite, DBA *outperforms* baseline (39.3% vs 32.0%).

Overall, DBA makes *interaction bandwidth* a tunable axis in attention design and characterizes an efficiency–perplexity tradeoff in this setting.

Keywords: Transformer, attention mechanism, low-rank, structural regularization, KV-cache, memory efficiency

1 Introduction

Modern Transformer architectures [13] achieve strong performance but their inference cost scales poorly with context length. The key–value (KV) cache grows linearly with sequence length and dominates memory bandwidth.

Most efficiency work targets storage rather than interaction: sharing KV heads, compressing cached representations, or reducing precision. These approaches preserve the full-dimensional token–token interaction used to compute attention scores.

This paper asks: how much *semantic interaction bandwidth* is actually required for the behaviors we evaluate? We propose Decoupled Bottleneck Attention (DBA), which separates low-rank semantic routing from higher-rank positional geometry. The architecture and results are summarized in Table 1.

Table 1: DBA in context: comparison of attention efficiency methods.

Method	**Reduces**	**Full interaction?**	**Composable?**
GQA [1]	KV storage	Yes	Yes
MLA [6]	KV storage	Yes (expands)	Yes
Linformer [14]	Sequence complexity	Approximated	No
DBA (ours)	Interaction bandwidth	No	Yes

Existing methods reduce what is *stored* or *approximated*. DBA reduces what is *interacted*: the dimensionality of the query–key subspace used for semantic token–token matching. We refer to this as *interaction bandwidth*.

Empirically, DBA produces a notable pattern in this training regime: held-out perplexity worsens, yet behavioral probe accuracy does not collapse and can improve under expanded evaluation. This work does not claim that perplexity is uninformative; rather, it reports an instance where next-token likelihood and the behavioral measures used here are not tightly coupled, and uses DBA as a controlled way to study interaction bandwidth as an architectural constraint.

2 Related Work

Low-rank and approximate attention. Linformer [14] projects along the sequence dimension; Performer [5] and Reformer [7] use kernel approximations or hashing. DBA instead reduces the query/key interaction dimension within each layer, targeting both compute ($O(n^2 r)$) and KV-cache ($O(nr)$).

Sparse attention. Longformer [3] and BigBird [15] use sparse patterns to reduce compute. DBA reduces per-token cache footprint and is complementary.

KV-cache optimization. MQA/GQA [11, 1] share KV heads; MLA [6] compresses to a latent expanded at runtime. Both preserve full interaction bandwidth. DBA reduces the interaction dimension itself and enables heterogeneous quantization (e.g., Q4 semantic, Q8 geometric).

Expressiveness limits. Low-rank attention can harm representation power [4, 2]. DBA mitigates this by compressing only the semantic path while preserving geometric fidelity.

Semantic subspaces. Menary et al. [8] show attention operates through independent semantic subspaces. DBA explicitly separates semantic and geometric subspaces, which may satisfy their orthogonality requirements.

3 Methodology

3.1 Standard Multi-Head Attention

In standard scaled dot-product attention with H heads:

$$\text{Attn}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V \tag{1}$$

where $Q, K, V \in \mathbb{R}^{n \times d}$ are obtained by linear projection from input $X \in \mathbb{R}^{n \times d_{\text{model}}}$. The KV-cache requires $O(2 \cdot L \cdot n \cdot d)$ memory.

3.2 Bottleneck Attention

We project Q and K to a lower-dimensional space *before* computing attention scores:[1]

$$Q' = XW'_Q, \quad K' = XW'_K \tag{2}$$

where $W'_Q, W'_K \in \mathbb{R}^{d_{\text{model}} \times d_{\text{attn}}}$ with $d_{\text{attn}} \ll d_{\text{model}}$. This reduces dot-product complexity from $O(n^2 \cdot d_{\text{model}})$ to $O(n^2 \cdot d_{\text{attn}})$ and KV-cache proportionally.

[1] Our use of "bottleneck" refers to dimensionality reduction in the query/key space, distinct from Park et al.'s BAM [9].

3.3 Decoupled Bottleneck Attention

The key insight is that *semantic routing* and *positional geometry* have different bandwidth requirements. DBA decomposes the attention score into two additive components:

$$\text{Score} = \underbrace{\frac{Q_{\text{sem}} K_{\text{sem}}^\top}{\sqrt{d_{\text{sem}}/H}}}_{\text{Semantic routing}} + \underbrace{\frac{Q_{\text{geo}} K_{\text{geo}}^\top}{\sqrt{d_{\text{geo}}/H}}}_{\text{Geometric routing}} \tag{3}$$

The semantic path uses $d_{\text{sem}} = 256$ total dimensions (8 per head with $H = 32$). The geometric path uses $d_{\text{geo}} = 1024$ (32 per head). RoPE [12] is applied *only* to the geometric path:

$$Q_{\text{geo}}, K_{\text{geo}} \leftarrow \text{RoPE}(Q_{\text{geo}}, K_{\text{geo}}, \text{position}) \tag{4}$$

The value projection is not compressed:

$$V = XW_V, \quad W_V \in \mathbb{R}^{d_{\text{model}} \times d_{\text{attn}}}, \quad d_{\text{attn}} = d_{\text{sem}} + d_{\text{geo}} = 1280 \tag{5}$$

This preserves representational capacity downstream of routing: DBA constrains *which* tokens interact, not *what* information is aggregated.

Semantic/geometric gating. Our implementation optionally enables a learnable per-head gate $g_h = \sigma(\gamma_h) \in [0, 1]$:

$$Q'_{\text{sem},h} = 2g_h \cdot Q_{\text{sem},h}, \quad Q'_{\text{geo},h} = 2(1 - g_h) \cdot Q_{\text{geo},h} \tag{6}$$

Ablations show gating provides substantial improvement (-5.67 PPL at 10k steps). We validate this at scale with a separate 100k-step gated run (Section 4.5). The primary comparison uses the ungated variant to isolate the decoupling contribution.

4 Experiments

We present experiments in chronological order: (1) architecture selection at 22 layers/10k steps, (2) main results at 22 layers/100k steps, (3) ablations at 12 layers/10k steps, and (4) gated DBA results.

4.1 Experimental Setup

All models use d_{model}=2048, 32 heads, and are trained on FineWeb-Edu with AdamW.[2] Table 2 summarizes configurations.

4.2 Architecture Selection (22L 1B 10k)

We selected **sem8** for 100k-step runs based on 10k-step comparisons (Table 3, Figure 1): best efficiency score (1.51×) with 37.5% KV-cache reduction and competitive downstream accuracy (1.2% gap).

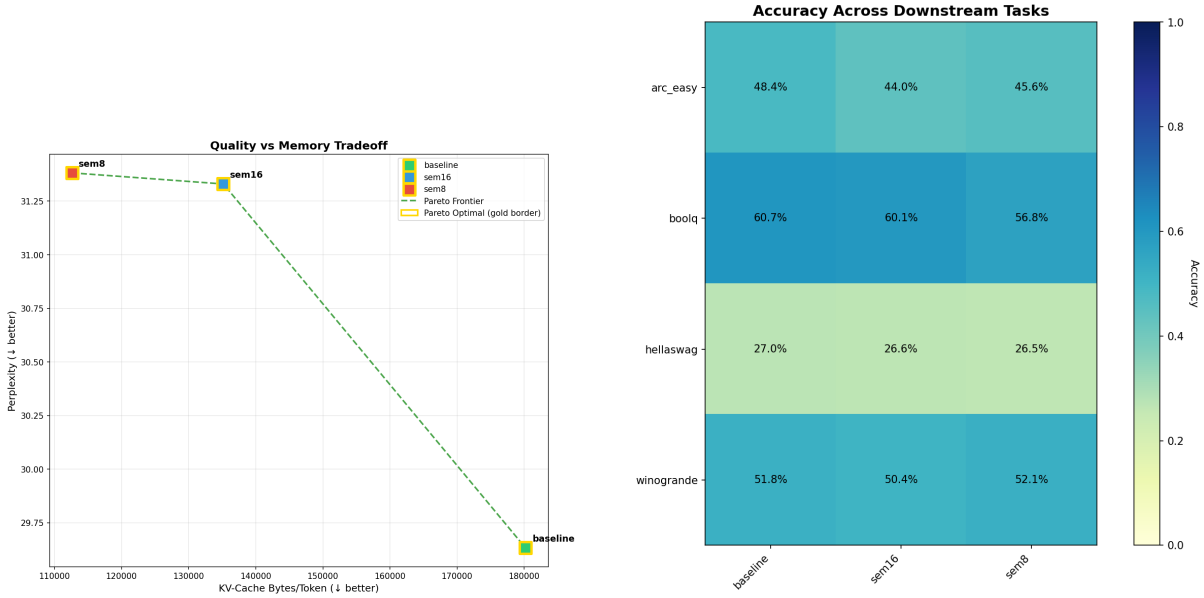
[2]Code and checkpoints: `https://github.com/theapemachine/caramba`.

Table 2: Experimental configurations. All use vocab = 50304, batch 32, d_{model}=2048.

Configuration	d_{sem}	d_{geo}	**Head**	d_{ff}	**Steps**	**Seeds**
22L Baseline	—	—	64	4096	10k	1337
22L sem16	512	1024	16+32	4096	10k	1337
22L sem8	256	1024	8+32	4096	10k	1337
22L Baseline	—	—	64	5632	100k	42
22L Decoupled	256	1024	8+32	5632	100k	42
12L Baseline	—	—	64	5632	10k	1337–39
12L Bottleneck	—	—	40	5632	10k	1337–39
12L Decoupled	256	1024	8+32	5632	10k	1337–39
12L GQA 32Q/4KV	—	—	64	5632	10k	1337–39
22L Gated	256	1024	8+32	5632	100k	42

Table 3: Architecture comparison at 10k steps.

Metric	baseline	sem16	sem8
Perplexity (↓)	**29.63**	31.33	31.38
Tokens/sec (↑)	71	75	**78**
KV bytes/token (↓)	180,224	135,168	**112,640**
KV reduction	—	25%	37.5%
Efficiency score	1.00	1.26	**1.51**



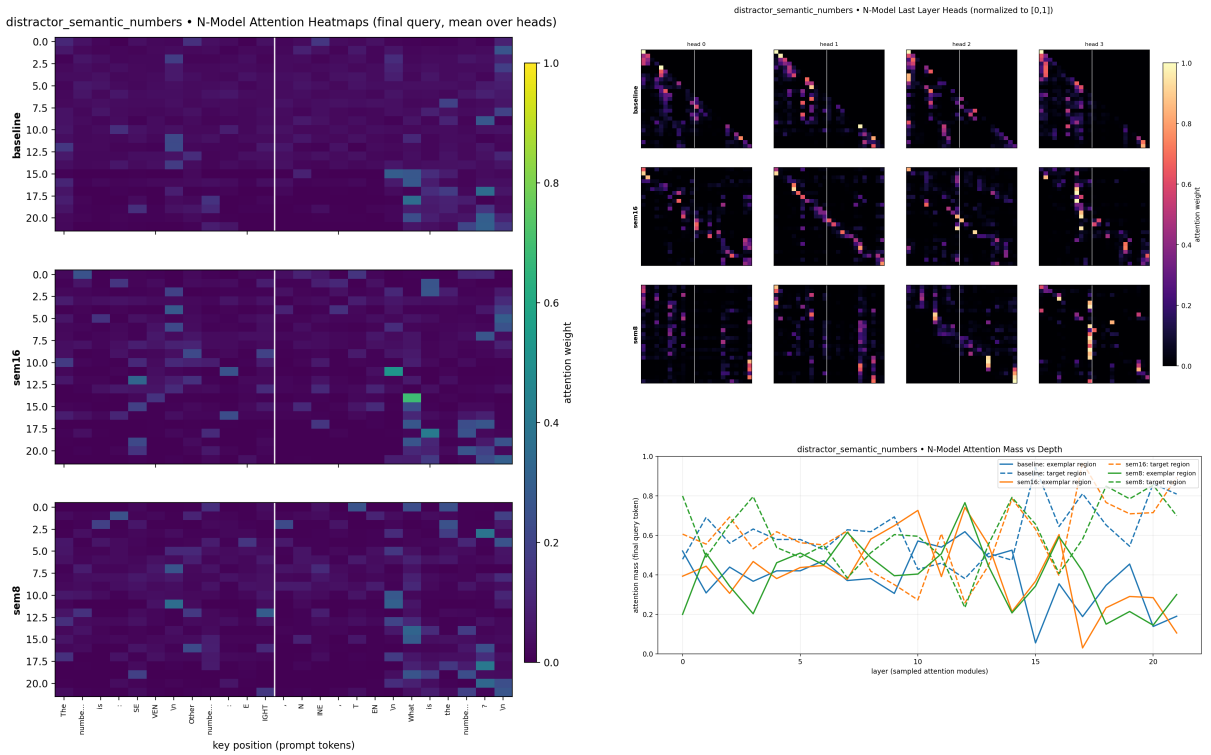
(a) Pareto frontier.

(b) Downstream accuracy.

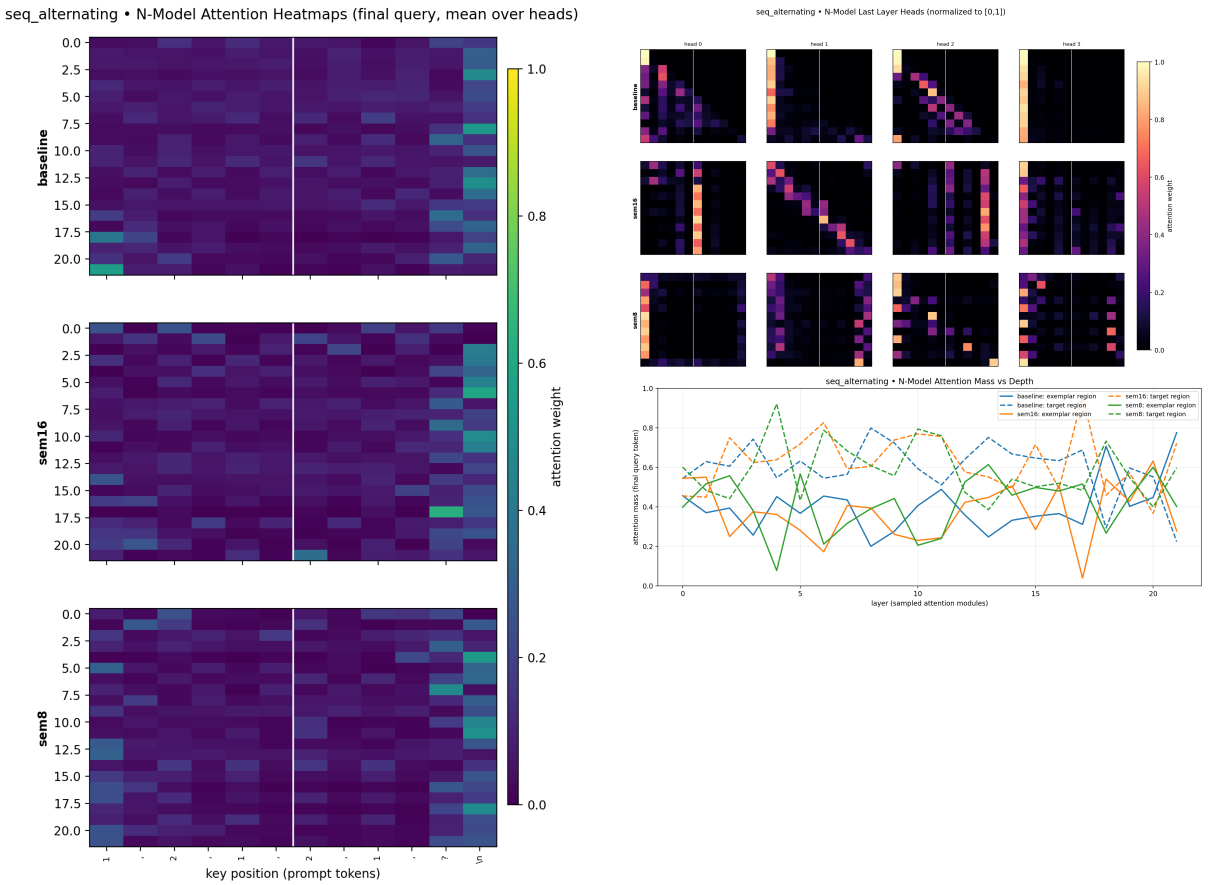
Figure 1: Architecture comparison at 10k steps.

Table 4: Downstream accuracy at 10k steps.

Task	baseline	sem16	sem8
Winogrande	51.8%	50.4%	**52.1%**
ARC-Easy	**48.4%**	44.0%	45.6%
BoolQ	**60.7%**	60.1%	56.8%
HellaSwag	**27.0%**	26.6%	26.5%
Micro avg	**37.1%**	36.5%	35.9%



(a) DBA wins: sem8 focuses on target "SEVEN"; baseline attends to distractors.

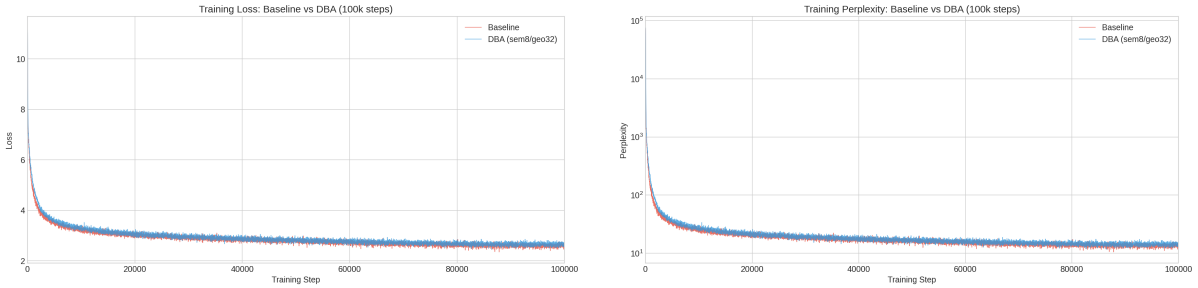


(b) Baseline wins: Baseline shows sharper attention to "2" positions; DBA shows diffuse patterns.

Figure 2: Attention patterns at 10k steps. Left: layer×token heatmaps. Right: last-layer head patterns and attention mass vs depth.

4.3 Main Results (22 Layers, 100k Steps)

Training dynamics. Training loss converges similarly: baseline 2.67 vs DBA 2.68. Held-out perplexity differs: **baseline 12.76 PPL vs DBA 13.53 PPL** (6% relative increase). Early runs showed transient loss spikes at peak LR; we mitigated this with conservative peak LR and gradient clipping.



(a) Training loss. (b) Perplexity.

Figure 3: A100 training curves (1B/100k steps).

Training efficiency. DBA reduces attention parameters by 37.5% (10.4% total). KV-cache elements drop by 37.5% (2560 vs 4096 per token). On A100, DBA trains 24% faster (29.3k vs 23.7k tok/s) and uses 34% less GPU memory (44 GB vs 67 GB)—a **1.88× improvement in throughput-per-GB**.

Table 5: Measured inference efficiency (Apple Silicon, fp16, batch=1).

Metric	Baseline	DBA	Change
KV-cache (bytes/token)	180,224	112,640	−37.5%
Cached decode (tok/s)	76.9	85.8	+12%
Long-prompt decode (tok/s)	21.9	24.6	+12%

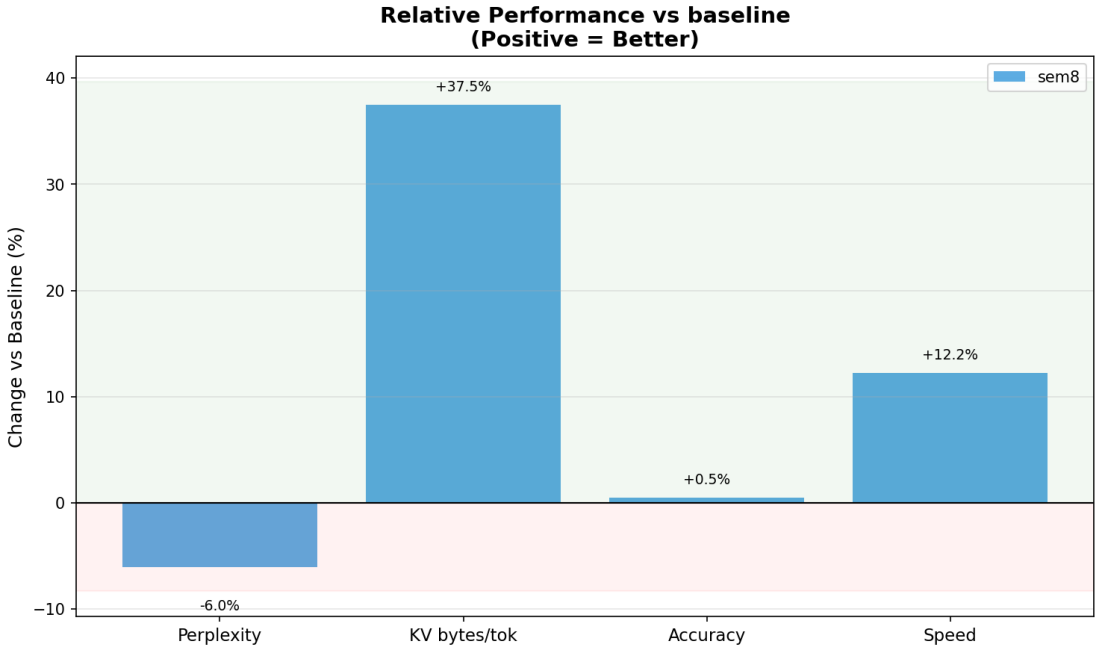


Figure 4: Relative performance vs baseline (100k; auto-generated). Positive is better.

4.3.1 Behavioral Probes: No Evidence of Collapse Under Probes

We evaluated 117 behavioral probes across 15 categories targeting exact copy, few-shot learning, distractor filtering, reasoning, arithmetic, and long-context retrieval.

Table 6: Behavioral benchmark results (117 probes; weighted scoring).

Model	Exact	Cont.	None	Hard	Soft	Weighted
baseline	1	32	84	**0.9%**	28.2%	10.4%
sem8	0	35	82	0.0%	**29.9%**	**12.8%**

Despite 6% higher perplexity, we do not observe a broad degradation across the probe suite; differences are task-specific. DBA outperforms on reasoning and world-knowledge; baseline performs better on distractor filtering and exact copy.

To stress-test this finding, we evaluated an expanded benchmark suite (490 probes, 18 categories) with additional adversarial, binding, and multi-hop tests. Under this suite, DBA scores 39.3% vs 32.0% soft accuracy, head-to-head 7–3 (Table 7).

Table 7: Extended behavioral benchmark (490 probes, 18 categories).

Metric	**Baseline**	**DBA**
Soft accuracy	32.0%	**39.3%**
Weighted accuracy	12.0%	**13.1%**
Head-to-head wins	3	**7**

Attention visualization. Baseline heads show diffuse attention spreading into distractor tokens; DBA heads concentrate attention near salient targets (Figure 5).

Table 8: Head-to-head differences. Tests where exactly one model passed.

Test	Baseline	DBA
Only Baseline passed (6):		
copy_long_sequence	✓ Stops at 15	× Continues
distractor_words	✓ "BLUE"	× "GREEN"
double_negation	✓ "true"	× "false"
analogy_size	✓ "short"	× "wide"
multiply_zero	✓ 0	× 1
recent_vs_distant	✓ "NEW"	× "OLD"
Only DBA passed (7):		
compare_simple	× "true"	✓ "false"
compare_equal	× "true"	✓ "false"
negation_simple	× "true"	✓ "false"
days_week	× 6	✓ 7
antonym_hot	× "cool"	✓ "cold"
single_digit (1+1)	× 1	✓ 2
attention_focus_noise	× "NOISE"	✓ "APPLE"

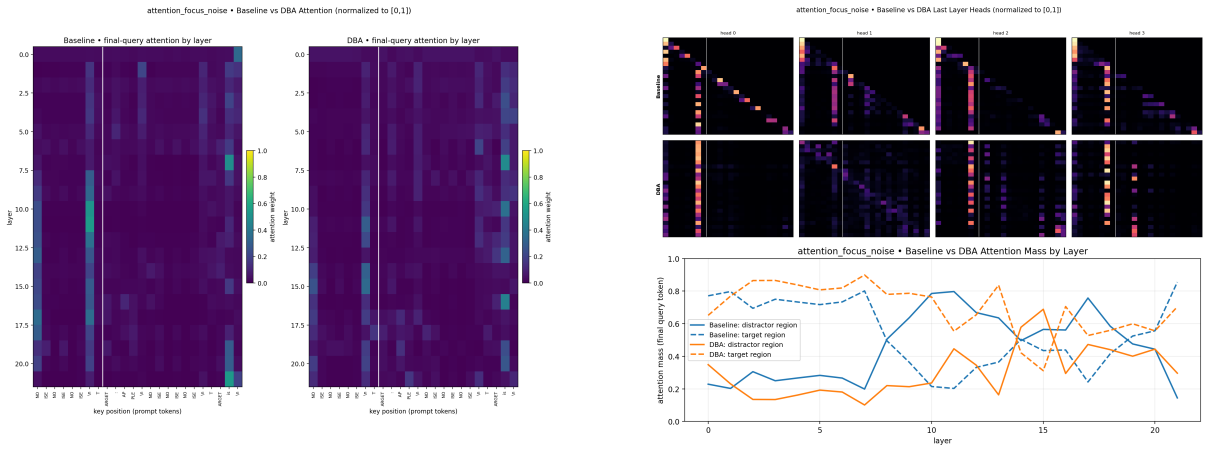


Figure 5: Attention patterns for `attention_focus_noise` (100k checkpoints). Left: layer×token heatmaps (baseline vs DBA). Right: last-layer head patterns and attention mass vs depth. Baseline shows diffuse attention; DBA concentrates on target.

4.3.2 Long-Context Inference

At long context lengths, the reduced KV-cache changes decode-time scaling. We measured single-token decode throughput from 2k to 131k context (Table 9).

Table 9: Context sweep: Baseline vs DBA decode (MPS, fp16, batch=1).

Context	Baseline (ms)	DBA (ms)	Speedup
2k	86.3	40.2	2.1×
4k	55.5	41.4	1.3×
8k	80.6	91.3	0.9×
16k	154.1	118.8	1.3×
32k	1566.4	216.9	7.2×
65k	67145.5	417.9	160.7×
131k	12740.9	952.1	13.4×

At 32k+ context, we observe large slowdowns for the baseline relative to DBA. The 7–160×

range reflects a regime where the baseline appears to hit memory bandwidth limits that a smaller KV-cache can mitigate.

4.4 Design Variant Ablations (12 Layers, 10k Steps)

Table 10: DBA design variants (12L, seed 1337).

Variant	null	tie_qk	gate	RoPE	PPL (Δ)
Decoupled (default)	×	×	×	geo	49.71 (—)
+ Null token	✓	×	×	geo	49.71 (+0.00)
+ Tied Q-K	×	✓	×	geo	56.33 (+6.62)
+ Gate	×	×	✓	geo	**44.04** (–5.67)
– RoPE	×	×	×	none	60.43 (+10.72)

Among the tested design variants at this scale, **gating produces the largest perplexity improvement** (−5.67 PPL). Tied Q-K degrades perplexity (+6.62 PPL). Removing RoPE degrades perplexity (+10.72 PPL). The null token has no measurable effect at this scale.

4.5 Gated DBA Results (22 Layers, 100k Steps)

The gating ablation showed a 12% relative PPL improvement at 10k steps—the largest effect of any design variant. To validate whether this improvement persists at scale, we trained a gated DBA variant at 22 layers for 100k steps (Table 2, final row). Training is in progress; results will be reported upon completion.

5 Discussion

5.1 Why Does Constrained Semantic Interaction Preserve Behavior?

DBA yields a consistent pattern in this setting: held-out perplexity increases, while the behavioral probe performance we report is preserved and can improve under expanded evaluation. The following non-exclusive interpretations are consistent with this observation: (1) *Perplexity may reward modeling low-value correlations*—high-bandwidth attention may allocate capacity to weak or local correlations that improve next-token likelihood without improving the probe outcomes measured here; (2) *Routing decisions may be intrinsically low-entropy*—if attention primarily performs coarse token selection, high semantic interaction bandwidth may be unnecessary [8, 10]; (3) *Depth-wise deferral*—our attention visualizations show DBA early layers exhibit diffuse attention while later layers concentrate more sharply, which is consistent with constrained bandwidth shifting routing/synthesis toward later depth.

These explanations are hypotheses rather than established mechanisms. In particular, our visualizations are correlational and do not isolate causality. Targeted ablations (e.g., hybrid stacks that switch between full attention and DBA at a chosen depth, or analysis of learned gate values when gating is enabled) would help distinguish among these possibilities.

5.2 Limitations

Perplexity increase. The 6% held-out perplexity increase is consistent across the reported comparison. Applications that prioritize next-token likelihood may not tolerate this difference.

Termination failures. DBA degrades on exact copy tasks requiring precise stopping. The EOS signal may compete with final-layer semantic aggregation.

Evaluation scope. The 117-probe suite detects capability collapse, not comprehensive task coverage.

Scale and variance. Large-scale results use a single seed. Multi-seed validation exists only at 12L/550M scale.

Long context. Models train on 2k context. Behavior under RoPE extrapolation is not validated.

Generality. DBA is evaluated only for autoregressive language modeling.

6 Conclusion

Decoupled Bottleneck Attention demonstrates that semantic routing and positional geometry can be factored into separate routing paths within attention. In this setting (1B parameters, 100k steps), DBA achieves 37.5% KV-cache reduction, 12% faster inference decode, and 1.88× better training throughput-per-GB, while increasing held-out perplexity by 6%. Under the behavioral evaluations used here, performance is preserved—and under expanded evaluation across 490 probes, *improves* relative to baseline.

This work establishes *interaction bandwidth* as a tunable axis in attention design, orthogonal to existing techniques like KV sharing and quantization. DBA can be composed with both to further explore the efficiency–perplexity frontier. The behavioral preservation observed here is consistent with the hypothesis that some next-token likelihood gains can come from modeling correlations that are not necessary for the probe outcomes measured in this paper, but this hypothesis is not proven by the current experiments.

Our experiments were conducted at 1B scale with limited training compute. Whether the perplexity gap narrows, holds, or widens at competitive scale remains to be determined. The behavioral outcomes reported here motivate larger-scale investigation and targeted mechanistic ablations.

Statements and Declarations

Conflict of Interest. The author declares no competing interests.

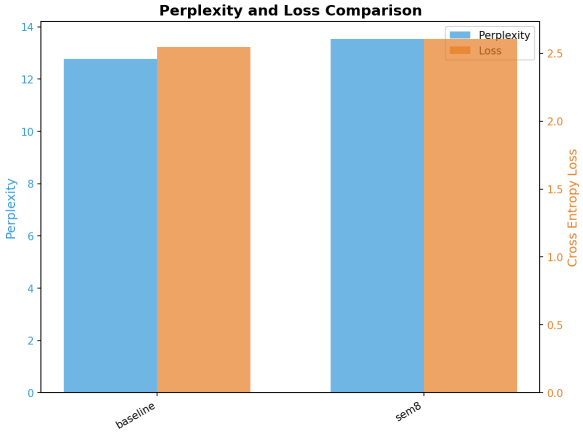
Data Availability. All code, checkpoints, and logs are available at `https://github.com/theapemachine/caramba`.

Funding. This research was conducted without external funding.

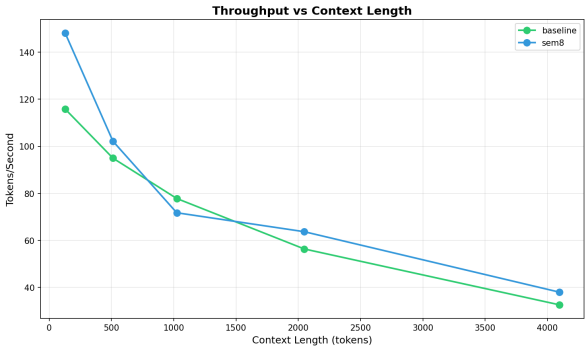
A Decoupled Ablations

We evaluated four DBA design variants at 12 layers (Table 10). The **null token** variant adds a learnable "sink" token but shows no measurable effect at this scale. **Tied Q-K** shares semantic query and key projections, which hurts performance (+6.62 PPL), indicating that asymmetric query–key interactions are important. **Gating** is the best variant (−5.67 PPL): a learnable per-head gate allows each head to balance semantic and geometric contributions. Finally, removing **RoPE** from the geometric path is catastrophic (+10.72 PPL), confirming that positional structure is essential even when decoupled from semantic routing.

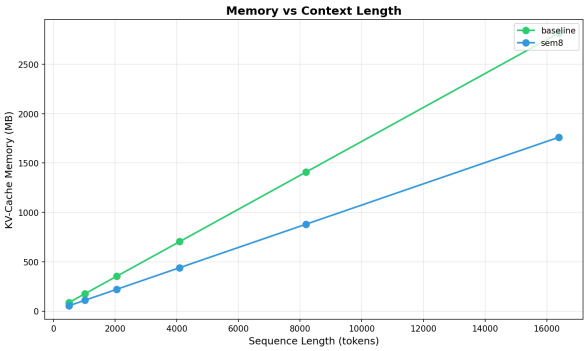
B Auto-generated Benchmark Artifacts (100k)



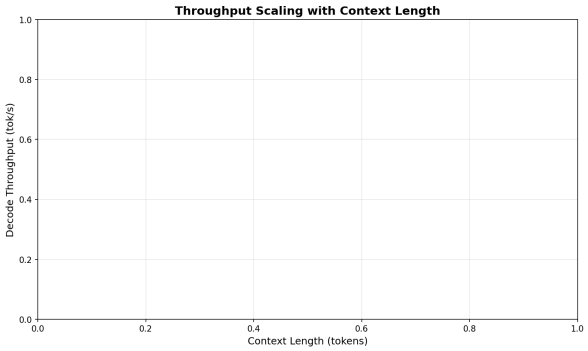
(a) Perplexity comparison.



(b) Latency vs context.



(c) Memory vs context.



(d) Context sweep throughput.

Figure 6: Auto-generated benchmark figures from the 100k run.

Table 11: Multi-model benchmark comparison

Metric	baseline	sem8
Perplexity (↓)	**12.76**	13.53
PPL Loss	2.5463	2.6050
Tokens/sec (↑)	**76**	85
TTFT (ms)	538.7	498.1
Prefill (ms)	525.8	482.4
KV Bytes/tok (↓)	**180224**	112640
Peak Mem (MB)	0	0
Task Accuracy (↑)	**42.6%**	41.9%
Behavioral (Exact)	**15.1%**	16.9%
Efficiency Score	1.00	1.51

Table 12: Improvements vs baseline (baseline)

Metric	sem8
PPL Δ	+0.77
Speedup	+12.2%
Mem. reduction	1.60×

Table 13: Behavioral Test Results (Weighted Scoring)

Model	Exact	Cont.	None	Hard	Soft	Weighted
baseline	1	32	84	**0.9%**	28.2%	10.4%
sem8	0	35	82	0.0%	**29.9%**	**12.8%**

Table 14: Behavioral V2 Test Results (Weighted Scoring)

Model	Exact	Cont.	None	Hard	Soft	Weighted
baseline	76	91	323	15.5%	**34.1%**	13.6%
sem8	82	75	333	**16.7%**	32.0%	**17.7%**

References

[1] Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebrón, and Sumit Sanghai. Gqa: Training generalized multi-query transformer models from multi-head checkpoints. *EMNLP*, 2023.

[2] Noah Amsel, Gilad Yehudai, and Joan Bruna. Quality over quantity in attention layers: When adding more heads hurts. In *ICLR*, 2025. OpenReview: `https://openreview.net/forum?id=y9Xp9NozPR`.

[3] Iz Beltagy, Matthew E. Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020.

[4] Srinadh Bhojanapalli et al. Low-rank bottleneck in multi-head attention models. *ICML*, 2020.

[5] Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Łukasz Kaiser, David Belanger, Lucy Colwell, and Adrian Weller. Rethinking attention with performers. In *ICLR*, 2021. arXiv:2009.14794.

[6] DeepSeek-AI. Deepseek-v2: A strong, economical, and efficient mixture-of-experts language model. *arXiv preprint arXiv:2405.04434*, 2024.

[7] Nikita Kitaev, Łukasz Kaiser, and Anselm Levskaya. Reformer: The efficient transformer. In *ICLR*, 2020. arXiv:2001.04451.

[8] Stephen Menary, Samuel Kaski, and Andre Freitas. Transformer normalisation layers and the independence of semantic subspaces. *arXiv preprint arXiv:2406.17837*, 2024. Shows Pre-Norm requires orthogonal semantic subspaces; analyzes attention as routing between independent subspaces.

[9] Jongchan Park et al. Bam: Bottleneck attention module. *BMVC*, 2018.

[10] Yehonathan Refael, Jonathan Svirsky, Boris Shustin, Wasim Huleihel, and Ofir Lindenbaum. Adarankgrad: Adaptive gradient-rank and moments for memory-efficient llms training and fine-tuning. *arXiv preprint arXiv:2410.17881*, 2024.

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

[12] Jianlin Su et al. Roformer: Enhanced transformer with rotary position embedding. *arXiv preprint arXiv:2104.09864*, 2021.

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

[14] Sinong Wang, Belinda Z Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768*, 2020.

[15] Manzil Zaheer, Guru Guruganesh, 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. *arXiv preprint arXiv:2007.14062*, 2020.