# Bottleneck Attention: Hard-Wiring Low-Dimensional Routing in GPT-Style Transformers

Daniel Owen van Dommelen
theapemachine@gmail.com

December 2025

## Abstract

**Background:** Adaptive Low-Rank Training (ALRT) indicates extreme redundancy in Transformer attention: query/key (Q/K) projections compress to effective ranks $\approx 9$–$14$ (mean $\approx 11.2$) out of 512 in prior experiments, suggesting that attention pattern computation may live in a low-dimensional comparison subspace.

**Methods:** We test a minimal architectural hypothesis (*Bottleneck Attention*): decouple residual stream width $d_{\text{model}}$ from the attention subspace dimension $d_{\text{attn}}$. We compute attention with $Q, K, V : \mathbb{R}^{d_{\text{model}}} \to \mathbb{R}^{d_{\text{attn}}}$ where $d_{\text{attn}} \ll d_{\text{model}}$, and project back with $\mathbb{R}^{d_{\text{attn}}} \to \mathbb{R}^{d_{\text{model}}}$. Optionally, we append a learnable per-head null key/value pair $(k_\emptyset, v_\emptyset)$ to the key/value sequence, allowing the attention distribution to assign explicit probability mass $p_\emptyset$ to a no-op sink (*attend nowhere*), and we learn per-head temperature scaling.

**Results:** On a controlled WikiText-2 word-level setup (6-layer GPT-style LM; $d_{\text{model}} = 512$, 8 heads, context 256), reducing $d_{\text{attn}}$ from 512 to 128 (4×) and to 32 (16×, with a null slot) yields only modest degradation in best validation loss (5.3688 $\to$ 5.4800 $\to$ 5.6070). Notably, during early training ($< 500$ steps), bottleneck variants achieve lower validation perplexity than the full-width baseline (e.g., step 200: 582.11 baseline vs 518.64 at $d_{\text{attn}} = 128$; 519.18 at $d_{\text{attn}} = 32$+null), suggesting accelerated initial learning on a constrained optimization manifold.

**Conclusions:** Attention routing in this regime is intrinsically low-dimensional. Because KV-cache memory scales linearly in $d_{\text{attn}}$, reducing $d_{\text{attn}}$ from 512 to 128 reduces per-token inference state by 75% (and to 32 by 93.75%), directly improving long-context inference feasibility.

## 1 Introduction

Transformers remain architecturally conservative despite dramatic changes in scale, hardware, and use cases since their introduction [1]. Many design choices persist because they work, not because they are known to be minimal. ALRT was motivated by a simple suspicion: much of Transformer parameterization is redundant, and this redundancy is not uniform across components. Empirically, ALRT finds strong projection-type asymmetry: attention Q/K projections compress far more aggressively than attention output and MLP projections [9, 10].

This report tests the most literal architectural consequence: if attention pattern computation truly lives in a small subspace, we can *hard-wire* that by decoupling attention's internal dimension from the residual stream width. Unlike dynamic rank resizing (which complicates optimizer state validity), this is a static architectural change trained from scratch.

## 2 Background: ALRT and the low-rank attention signal

ALRT parameterizes linear maps as low-rank factors $W = UV$ and adapts rank during training using stable rank estimation:

$$r_{\text{stable}}(W) = \frac{\|W\|_F^2}{\sigma_{\max}(W)^2}.$$

Across GPT-style language modeling experiments, ALRT reports that Q/K projections are the most compressible, with mean ranks $\approx 11.2$ out of 512, while attention output and MLP projections retain higher effective ranks [9]. A natural architectural interpretation is that attention's *comparison space* (used to compute patterns) is low-dimensional, even if the residual stream is not.

## 3 Method: Bottleneck Attention

### 3.1 Decoupled attention subspace

We replace full-width Q/K/V projections with projections into a smaller attention subspace:

$$W_Q, W_K, W_V \in \mathbb{R}^{d_{\text{model}} \times d_{\text{attn}}}, \qquad W_O \in \mathbb{R}^{d_{\text{attn}} \times d_{\text{model}}}.$$

Attention score and apply operations now scale in $d_{\text{attn}}$, while the residual stream stays at $d_{\text{model}}$. This directly targets attention FLOPs, attention parameters, and KV-cache memory.

### 3.2 "Attend nowhere" via a null key/value (optional)

Softmax attention forces each query to allocate probability mass across keys. We optionally append a learnable per-head null key/value $(k_\varnothing, v_\varnothing)$ to the Key/Value sequence. For a query position $t$, the attention distribution becomes $p(\cdot \mid t) = \mathrm{softmax}(\ldots, q_t^\top k_\varnothing)$, allowing explicit probability mass $p_\varnothing$ on a no-op sink. We initialize $v_\varnothing = 0$ so "attend nowhere" starts as "contribute nothing".

Table 1: WikiText-2 (word-level) results. Best validation metrics over training.

| Model | $d_{\text{attn}}$ | Params (M) | Best val loss | Best va |
|---|---|---|---|---|
| Baseline | 512 | 36.06 | 5.3688 | 21 |
| Bottleneck | 128 | 31.34 | 5.4800 | 23 |
| Extreme+Null | 32 | 30.16 | 5.6070 | 27 |

### 3.3 Learned per-head temperature (optional)

When $d_{\text{attn}}$ is small, head dimension $d_{\text{head}} = d_{\text{attn}}/n_{\text{head}}$ becomes tiny and dot products can become noisy. We include a learnable per-head logit scale multiplying the standard $1/\sqrt{d_{\text{head}}}$ factor.

## 4 Experimental Setup

**Dataset.** WikiText-2 word-level setup using a minimal whitespace tokenizer with per-line `<eos>` boundaries and `<unk>` for OOV. A single input file is tokenized and split 90/10 into train/validation within the script.

**Model.** Decoder-only GPT-style Transformer with pre-norm blocks; 6 layers; $d_{\text{model}} = 512$; 8 heads; $d_{\text{ff}} = 2048$; context length 256; dropout 0.1; weight tying between input embedding and LM head.

**Optimization.** AdamW with lr=3e-4, weight decay=0.01, betas=(0.9,0.95), batch size 32, 200 steps/epoch, 30 epochs (6000 steps), gradient clipping 1.0. Evaluation every 200 steps (50 eval iters). Hardware: Apple MPS backend.

## 5 Results

Table 1 summarizes best validation metrics. Reducing $d_{\text{attn}}$ yields a smooth degradation curve rather than a catastrophic cliff.

### 5.1 Early convergence anomaly

Figure 1 shows validation perplexity vs training step. Notably, at the first evaluation point (step 200), the bottleneck variants outperform the baseline (e.g., baseline 582.11 vs 518.64 for $d_{\text{attn}} = 128$). This pattern persists through the first several hundred steps (inset), contradicting the intuition that wider attention must be retained to "find" a good solution.

### 5.2 Pareto curve

Figure 2 shows best validation loss vs $d_{\text{attn}}$ (log scale). Figure 3 shows parameters vs best validation loss.

## 6 Discussion

**Attention as low-dimensional routing.** The viability of $d_{\text{attn}} = 32$ suggests that attention can route effectively using

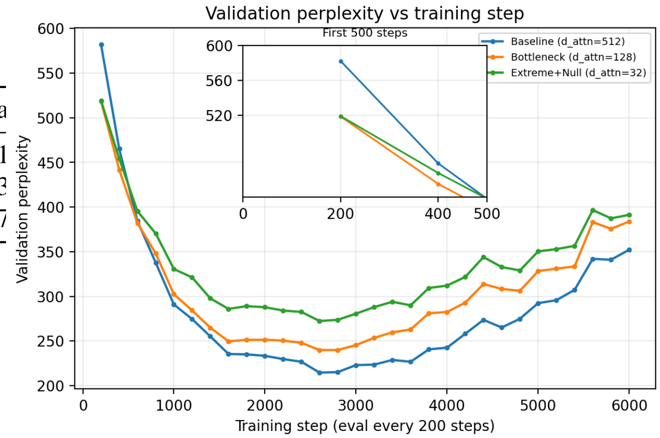


Figure 1: Validation perplexity vs training step (eval every 200 steps). Inset: first 500 steps. Bottleneck variants learn faster early despite lower attention dimensionality.

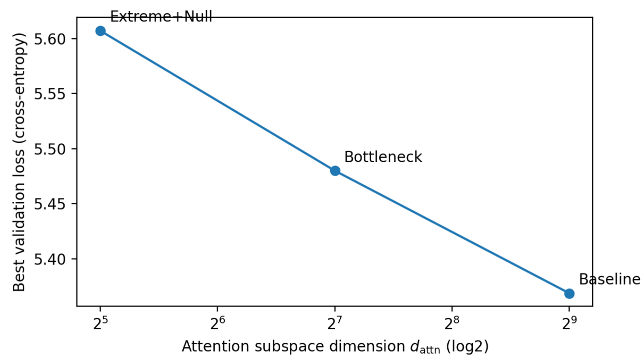


Figure 2: Best validation loss vs $d_{\text{attn}}$ ($\log_2$ x-axis).

a small comparison/mixing subspace, consistent with ALRT's observation that Q/K matrices collapse to tiny effective rank. A useful mental model is "wide residual stream + narrow router."

**Why degradation is smooth.** If high-dimensional attention geometry were essential for core competence, we would expect collapse once $d_{\text{attn}}$ falls below a critical threshold. Instead, loss increases gradually, suggesting many attention patterns in this regime are representable with relatively few degrees of freedom.

**Early learning signal.** The early-phase perplexity advantage (Figure 1) suggests that full-width attention may be not only redundant but also *optimization-suboptimal* in this regime. One interpretation is that over-parameterized Q/K spaces introduce many weakly-aligned gradient directions, increasing noise and slowing early feature acquisition. Constraining the routing manifold may act as an implicit regularizer that accelerates early training.

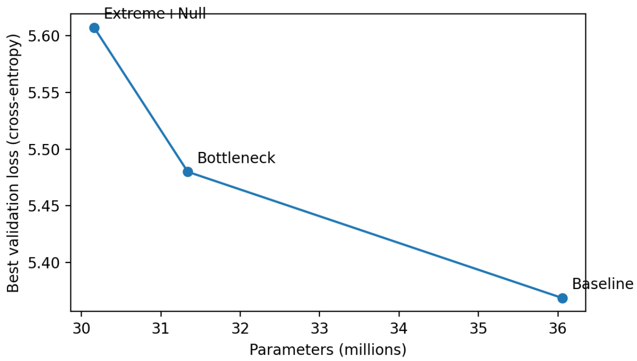


Figure 3: Parameters (M) vs best validation loss.

**KV-cache memory: concrete scaling.** For standard multi-head attention, per-token KV state per layer is

$$\text{KV per token per layer} = 2 \cdot n_{\text{head}} \cdot d_{\text{head}} = 2d_{\text{attn}}.$$

Therefore total KV-cache memory for batch size $B$, context length $T$, $L$ layers, and element size $b$ bytes is

$$M_{\text{KV}} = 2BTL\, d_{\text{attn}}\, b,$$

which scales *linearly* in $d_{\text{attn}}$. Reducing $d_{\text{attn}}$ from 512 to 128 cuts KV-cache by 75% (4× smaller); reducing to 32 cuts KV-cache by 93.75% (16× smaller). In a hypothetical 128k-context regime, a 4× reduction in per-token KV state enables $\approx 4\times$ larger batch size (or $\approx 4\times$ longer context) for the same memory budget, directly targeting a dominant modern inference bottleneck.

# 7 Related work

**Low-rank adaptation and training.** LoRA [2] demonstrates that many learned transformations can be modified via low-dimensional updates. ALRT extends this line of thought by measuring and adapting rank during training [9, 10]. AdaRankGrad [7] provides complementary theoretical and algorithmic evidence that learning signals themselves become low-rank: it proves that estimated layer gradients decrease in rank during training and asymptotically approach rank one, and leverages this structure to reduce optimizer memory. Unlike AdaRankGrad (which compresses *updates*), Bottleneck Attention applies low-dimensionality as a static architectural constraint, yielding inference-time KV-cache savings.

**Efficient attention and KV-cache reduction.** Efficient-attention methods address quadratic attention cost via sparsity or kernel/low-rank approximations (e.g., Reformer [3], Linformer [4], Performer [5]). Recent architectures such as DeepSeek-V2's Multi-Head Latent Attention (MLA) compress the Key/Value representation into a low-rank latent to reduce inference memory [6]. Bottleneck Attention is complementary: it reduces the dimensionality of the *interaction space* (including the Query projection and the score/apply computation), not only the stored KV state.

# 8 Limitations and future work

**Long-context scaling risk (Johnson–Lindenstrauss).** These experiments use context length 256 and learned absolute positional embeddings. If attention must reliably discriminate among $N$ candidate keys, random-projection theory suggests the dimensionality required to preserve pairwise distances scales as $O(\log N)$ (e.g., Johnson–Lindenstrauss-type bounds [8]). Thus $d_{\text{attn}} = 32$ may saturate at much longer sequence lengths, and the optimal $d_{\text{attn}}$ may increase (slowly) with context length.

**Position encoding.** Interactions with RoPE/ALiBi and long-context regimes remain future work.

**Variance and scale.** We report single runs at modest scale; multi-seed sweeps and larger models are needed for confidence intervals and generalization.

# 9 Conclusion

ALRT indicates that attention pattern computation is intrinsically low-rank, with Q/K compressing to mean ranks $\approx 11.2/512$. Bottleneck Attention operationalizes this as a static architectural change: compute attention in a reduced subspace $d_{\text{attn}} \ll d_{\text{model}}$ while preserving a wide residual stream. On WikiText-2 (word-level), $d_{\text{attn}} = 128$ and even $d_{\text{attn}} = 32$ (with a null slot) remain viable with modest loss penalties, and show an early-phase perplexity advantage. Together, these results suggest that the standard convention $d_{\text{attn}} = d_{\text{model}}$ is substantial structural bloat in this regime.

# Reproducibility

Code, scripts, and exact reproduction instructions (including pinned dependencies and commands) are provided at: `https://github.com/TheApeMachine/bottleneck-attention`

# Conflict of interest statement

The author declares no competing interests.

# Funding statement

This research received no external funding.

# Ethics statement

This work does not involve studies with human participants, human data, human tissue, or animals; therefore, no ethical approval was required.

## Data availability statement

The data supporting this study are derived from the publicly available WikiText-2 corpus. Training scripts, configuration, and logs sufficient to reproduce the reported results are available in the repository listed above.

## References

[1] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[2] Edward J Hu, Yelong Shen, Phil Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

[3] Nikita Kitaev, Łukasz Kaiser, and Anselm Levskaya. Reformer: The efficient transformer. In *International Conference on Learning Representations (ICLR)*, 2020.

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

[5] Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Łukasz Kaiser, and others. Rethinking attention with performer's random features. In *International Conference on Learning Representations (ICLR)*, 2021.

[6] DeepSeek-AI. DeepSeek-V2: A strong, economical, and efficient mixture-of-experts language model. *arXiv preprint arXiv:2405.04434*, 2024.

[7] 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.

[8] William B Johnson and Joram Lindenstrauss. Extensions of Lipschitz mappings into a Hilbert space. In *Conference in Modern Analysis and Probability*, pages 189–206. American Mathematical Society, 1984.

[9] Daniel Owen van Dommelen. Adaptive low-rank training via spectral rank estimation: Dynamic parameter efficiency in transformer language models (technical report). Internal technical report, December 2025.

[10] Daniel Owen van Dommelen. Adaptive low-rank training — technical report v2. Internal technical report, December 2025.