ATTENTION PROJECTION MIXING WITH EXOGENOUS ANCHORS

Jonathan Su
German Swiss International School
270985@learning.gsis.edu.hk

ABSTRACT

Cross-layer reuse of early attention projections can improve optimization and data efficiency, but it creates a structural conflict: the first layer must simultaneously act as a stable, reusable anchor for all deeper layers and as an effective computational block. We demonstrate that this tension constrains the performance of internal-anchor designs. We propose ExoFormer, which resolves the conflict by learning *exogenous anchor projections* outside the sequential layer stack. We introduce a unified normalized mixing framework that mixes queries, keys, values, and gate logits using learnable coefficients (exploring coefficient granularities: elementwise, headwise, and scalar), and we show that normalizing anchor sources is key to stable reuse. ExoFormer variants consistently outperform their internal-anchor counterparts, and the dynamic variant yields ~ 1.5 downstream accuracy points while matching validation loss using $\sim 1.5\times$ fewer tokens than Gated Attention. We explain this efficacy via an *Offloading Hypothesis*: external anchors preserve essential token identity, allowing layers to specialize exclusively in feature transformation. We release code and models to facilitate future research.

1 INTRODUCTION

The Transformer architecture (Vaswani et al., 2017) underpins modern large language models (LLMs) and contextual tasks. Its success relies on the multi-head self-attention mechanism, which enables dynamic, context-dependent interactions across sequences. However, as models scale in depth, ensuring stable training and effective information propagation remains a challenge.

Token information is diluted in deeper layers due to over-smoothing (Shi et al., 2022; Zhou et al., 2021), spurring interest in direct mechanisms for preserving early representations. Existing solutions leave key questions unanswered; ResFormer (Zhou et al., 2025) focuses solely on residualizing *values*, leaving the reuse of queries, keys, and gate logits unexplored.

We introduce a unified framework for cross-layer mixing across all attention pathways, systematically evaluating the contribution of mixing queries (Q), keys (K), and gate logits (G), alongside the established value (V) residual, using various coefficient granularities. A key insight is that applying RMSNorm to residual sources before mixing resolves distributional mismatch, enabling stable reuse.

Reusing first-layer projections reveals a tension: the layer must serve as both a stable anchor for deeper layers and a computational block for feature transformation. This dual objective inherently limits effectiveness in both roles.

We introduce **ExoFormer**, which resolves this tension by learning *dedicated exogenous anchor projections* outside the sequential layer stack. We compare this with NuResFormer (**N**ormalized **u**nified), an internal-anchor baseline. Decoupling the two roles proves beneficial: ExoFormer variants outperform NuResFormer counterparts in perplexity while remaining competitive in downstream accuracy, without increasing width or depth. We explain this via an *Offloading Hypothesis*: the external anchor preserves token identity, allowing sequential layers to specialize in refinement.

Our empirical analysis reveals that ExoFormer layers spend approximately two-thirds of their depth in the refinement stage (versus one-third in standard Transformers), with catastrophic failure upon

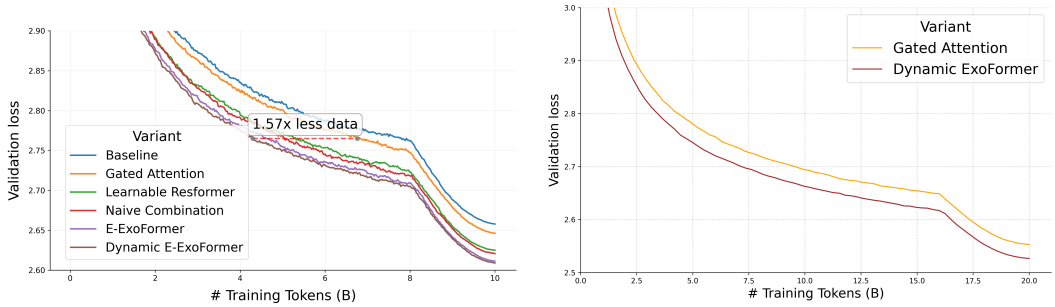


Figure 1: (Left) Validation loss for $\sim 450M$ parameter models. (Right) Validation loss for $\sim 1B$ parameter models.

anchor removal, confirming that layers specialize under the assumption of externalized identity preservation.

2 Related Work

Value Residuals and Gated Attention. Zhou et al. (2025) introduced *Value Residual Learning* (ResFormer), which adds a residual connection from the first layer's value projection (V_1) to the value projections of all subsequent layers (regulated by scalar coefficients λ_1 and λ_2). This simple yet effective method was shown to greatly improve model performance and data efficiency, highlighting the benefit of explicitly reusing early content representations. However, preliminary attempts to residualize queries and keys were found to be unstable (Zhou et al., 2025). Concurrently, gated attention mechanisms have been explored to introduce dynamic, input-dependent modulation to the attention output, improving expressiveness and training stability (Qiu et al., 2025). Our work generalizes residual learning to all attention pathways and stabilizes it via normalized mixing, bridging the gap between residual connections and gated attention.

Cross-layer communication and residual mixing. Recent work has sought to improve cross-layer information flow in Transformers beyond simple residual connections. Zhu et al. (2025) proposed Hyper-Connections that expand residual stream width. More recently, Xie et al. (2026) introduced Manifold-Constrained Hyper-Connections (mHC), restoring an identity-like signal-preservation property to the hyper-connected architecture.

Most closely related to our work, though orthogonal in direction, is **MUDDFormer** (Xiao et al., 2025), which proposes Multiway Dynamic Dense (MUDD) connections. MUDDFormer decouples the input to each Transformer block into four streams (query, key, value, residual) and dynamically aggregates outputs from *all* preceding layers using context-specific weights generated by a small MLP.

While the techniques mentioned above and our work focus on enhancing residual pathways, they operate at different levels and can coexist. MUDDFormer primarily addresses the *input* to the attention projections. In contrast, our unified mixing framework focuses on *mixing* the attention projections themselves. Our approach can be seen as a form of blending *after* projection, whereas MUDDFormer enriches the input *before* projection.

3 Methodology

3.1 Preliminaries and Notation

For the n-th Transformer layer, let $H_{n-1} \in \mathbb{R}^{T \times d_{\text{model}}}$ be the input hidden states (after pre-normalization), where T is the sequence length and d_{model} is the model width. We use h attention heads and per-head dimension d_k such that $d_{\text{model}} = h\, d_k$. We denote the projected queries, keys, values, and gate logits as Q_n, K_n, V_n, G_n. Let $\mathrm{RMSNorm}(\cdot)$ denote per-token RMS normalization applied independently to each attention head (with learnable gain).

We consider three coefficient granularities for λ: Scalar (S) (a single coefficient shared across all channels), Headwise (H) (one coefficient per head, broadcast across d_k), and Elementwise (E) (one coefficient per channel).

3.2 Multi-Head Attention

We describe attention as a sequence of stages, with two standard enhancements integrated implicitly: rotary position embeddings (RoPE) (Su et al., 2024) and query/key normalization (QKNorm) (Henry et al., 2020). For simplicity, the multi-head mechanism is presented using unified tensors with an implicit head dimension.

Stage 1: QKV Linear Projections. Given $H_{n-1} \in \mathbb{R}^{T \times d_{\text{model}}}$, we compute the projected tensors:

$$Q_n = H_{n-1} W_n^Q, \quad K_n = H_{n-1} W_n^K, \quad V_n = H_{n-1} W_n^V, \tag{1}$$

where $W_n^Q, W_n^K, W_n^V \in \mathbb{R}^{d_{\text{model}} \times d_{\text{model}}}$. The resulting Q_n, K_n, V_n are structured to contain h heads implicitly.

Stage 2: Scaled Dot-Product Attention (SDPA). Attention is computed per head, which is represented here as a single operation:

$$\begin{aligned} A_n &= \operatorname{softmax}\left(\frac{Q_n K_n^\top}{\sqrt{d_k}}\right) \in \mathbb{R}^{T \times T}, \\ U_n &= A_n V_n \in \mathbb{R}^{T \times d_{\text{model}}}, \end{aligned} \tag{2}$$

where the operations encompass the independent computations across h heads.

Stage 3: Final Output Projection. The output of the attention computation is projected:

$$O_n = U_n W_n^O, \qquad W_n^O \in \mathbb{R}^{d_{\text{model}} \times d_{\text{model}}}. \tag{3}$$

In NuResFormer/ExoFormer, the above stages use mixed tensors $\widehat{Q}_n, \widehat{K}_n, \widehat{V}_n$ from Eq. equation 10.

3.3 Gated Attention

We adopt the elementwise head-specific multiplicative gating formulation as described by Qiu et al. (2025). The general form is:

$$Y' = g(Y, X, W_\theta, \sigma) = Y \odot \sigma(X W_\theta), \tag{4}$$

where Y is the feature tensor to be modulated, X is the input used to compute the gating scores, W_θ are learnable parameters, σ is an activation function (sigmoid unless stated otherwise), and $\odot$ denotes elementwise multiplication.

In our attention block, we apply gating to the concatenated multi-head output (i.e., after Stage 2 and before Stage 3). Let $G_n \in \mathbb{R}^{T \times (h d_k)}$ denote the gate logits, computed from H_{n-1}:

$$G_n = H_{n-1} W_n^G, \qquad W_n^G \in \mathbb{R}^{d_{\text{model}} \times d_{\text{model}}}. \tag{5}$$

Then the gated attention output is

$$\widetilde{U}_n = U_n \odot \sigma(G_n), \qquad O_n = \widetilde{U}_n W_n^O. \tag{6}$$

Normalization placement. We note that in ablations, using post-norm instead of pre-norm substantially degraded gated attention performance.

3.4 Unified Mixing Formulation

We now present a general framework for mixing. The core idea is to enrich each layer's attention pathways with a set of persistent *anchor projections* $\{Q_{\text{anc}}, K_{\text{anc}}, V_{\text{anc}}, G_{\text{anc}}\}$ that is reused, via learnable mixing, across all layers.

Current-layer projections. For layer n with input H_{n-1}, we compute the standard projections:

$$Q_n = H_{n-1}W_n^Q, \quad K_n = H_{n-1}W_n^K, \quad V_n = H_{n-1}W_n^V, \quad G_n = H_{n-1}W_n^G. \tag{7}$$

Anchor projections. The anchors are a fixed set of projections defined once for the entire model. In our work, we explore two instantiations:

- **NuResFormer:** The anchors are the projections from the very first attention layer. That is,

$$Q_{\text{anc}} = Q_1, \quad K_{\text{anc}} = K_1, \quad V_{\text{anc}} = V_1, \quad G_{\text{anc}} = G_1, \tag{8}$$

- **ExoFormer:** The anchors are produced by a dedicated, external projection module on the input embeddings:

$$Q_{\text{anc}} = H_0W_{\text{anc}}^Q, \quad K_{\text{anc}} = H_0W_{\text{anc}}^K, \quad V_{\text{anc}} = H_0W_{\text{anc}}^V, \quad G_{\text{anc}} = H_0W_{\text{anc}}^G, \tag{9}$$

  where $W_{\text{anc}}^Q, W_{\text{anc}}^K, W_{\text{anc}}^V, W_{\text{anc}}^G \in \mathbb{R}^{d_{\text{model}} \times d_{\text{model}}}$ are independent learnable weight matrices.

Mixing with normalized sources. For each component $S \in \{Q, K, V, G\}$, we mix the anchor projection with the current-layer projection using learned coefficients. To stabilize the mixture, we apply RMS normalization to the anchor source before scaling:

$$\widehat{S}_n = \lambda_{n,1}^S \odot \text{RMSNorm}(S_{\text{anc}}) + \lambda_{n,2}^S \odot S_n, \qquad \forall S \in \{Q, K, V, G\}, \tag{10}$$

where $\lambda_{n,1}^S, \lambda_{n,2}^S$ are coefficient tensors of the chosen granularity (scalar, headwise, or elementwise). All λ parameters are initialized to 0.5 and are learnable.

We apply QKNorm and RoPE to the mixed queries and keys, $\widehat{Q}_n$ and $\widehat{K}_n$. We then compute the scaled dot-product attention per head using the mixed projections $\widehat{Q}_n, \widehat{K}_n, \widehat{V}_n$, apply the gating operation using $\widehat{G}_n$, and finally pass the result to the output projection.

3.5 First Layer Tension and ExoFormer

Cross-layer residual reuse makes early information available at every depth. This is powerful, but it implicitly forces the first layer to satisfy two pressures:

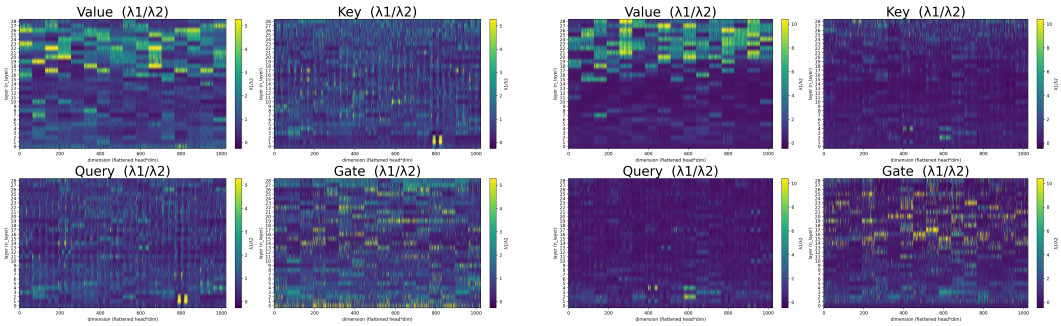
1. **Reusable anchor:** produce a broadly useful reference representation that remains valuable throughout depth.
2. **Progressive computation:** produce features that are easy for downstream layers to transform into increasingly task-relevant abstractions.

While these roles appear misaligned (universal anchors favor invariance, while progressive computation necessitates change), they can theoretically coexist because it is an *optional pathway* modulated by learned mixing coefficients ($\lambda_{n,1}, \lambda_{n,2}$). However, this structural constraint places pressure on the first layer to make compromises, inherently limiting its effectiveness in both roles. This tension is empirically supported by the analysis in Figure 5, showing that NuResFormer's first layer adopts a permissive gating policy compared to standard baselines, indicating a compromise on selectivity in favor of serving as a stable anchor.

Our most successful approach, termed **ExoFormer**, instantiates the general framework of Section 3.4 with the dedicated exogenous projections of Eq. equation 9.

3.6 Dynamic Mixing (DM) Module

Building upon the unified formulation, we take inspiration from MUDDFormer's Depth-wise Aggregate (DA) module (Xiao et al., 2025) and introduce a dynamic variant where the learnable parameters are modulated by context-dependent scaling factors computed from the layer input H_{n-1} using a small MLP. This allows the model to adapt its mixing strategy based on the specific context.



(a) Dynamic E-ExoFormer (full residual norms)

(b) E-ExoFormer (full residual norms)

Figure 2: Heatmaps showing the learned mixing coefficient ratio $\lambda_{n,1}/\lambda_{n,2}$ for each residualized component $\{Q, K, V, G\}$ across layers (y-axis) and channels/heads (x-axis) for model variants ~ 450M. This ratio quantifies the model's reliance on the anchor relative to the current layer's projection; a higher value indicates stronger reuse of the early signal.

Dynamic Coefficient Generation. For each layer n, we compute modulation scalars from its input H_{n-1} (pre-normalized) using a two-layer MLP with GELU activation and sigmoid output:

$$\mathcal{DM}_n(H_{n-1}) = \sigma\Big(\mathrm{GELU}\big(H_{n-1}W^{\mathrm{DM}}_{n,1}\big) \cdot W^{\mathrm{DM}}_{n,2} + b^{\mathrm{DM}}_n\Big) \tag{11}$$

The trainable parameters for the Dynamic Mixing module at layer n are:

$$\theta^{\mathrm{DM}}_n = \big\{W^{\mathrm{DM}}_{n,1} \in \mathbb{R}^{d_{\mathrm{model}} \times 16}, W^{\mathrm{DM}}_{n,2} \in \mathbb{R}^{16\times 8}, b^{\mathrm{DM}}_n \in \mathbb{R}^8\big\}$$

The output dimension of this module is 8, corresponding to the dynamic scaling factors:

$$\{\gamma^Q_{n,1}, \gamma^Q_{n,2}, \gamma^K_{n,1}, \gamma^K_{n,2}, \gamma^V_{n,1}, \gamma^V_{n,2}, \gamma^G_{n,1}, \gamma^G_{n,2}\}$$

The output layer weights $W^{\mathrm{DM}}_{n,2}$ and bias b^{DM}_n are zero-initialized, ensuring that initial sigmoid outputs are 0.5. Consequently, the base λ parameters must be initialized at 1.0 to achieve effective identity mixing at initialization.

Modulated Mixing. For each component we compute using the dynamic scaling factors:

$$\widehat{S}_n = (\lambda^S_{n,1}\gamma^S_{n,1}) \odot \mathrm{RMSNorm}(S_{\mathrm{anc}}) \quad + (\lambda^S_{n,2}\gamma^S_{n,2}) \odot S_n, \qquad \forall S \in \{Q, K, V, G\}, \tag{12}$$

where $\gamma^S_{n,i}$ are broadcast appropriately based on the residual granularity (elementwise, headwise, or scalar), and $\lambda^S_{n,i}$ are the learnable base parameters.

4 Experiments

4.1 Experimental Setup

All models use a modern pre-normalized Transformer architecture with SwiGLU activations (Shazeer, 2020), QKNorm (Henry et al., 2020), and rotary position embeddings (Su et al., 2024) trained on FineWeb-Edu (Penedo et al., 2024) (10B tokens for $\sim$ 450M models; 20B for $\sim$ 1B models. Further details and evaluation setup can be found in the appendix.

4.2 The Role of Anchor Normalization

RMSNorm serves two roles: first, acting as a scale-dependent rational function, it introduces a nonlinear transformation into the otherwise linear residual pathway; and second, as a mild isotropization

Table 1: Full performance comparison of model variants on 6 multiple-choice downstream tasks. Parameter counts are 453M (with Gated Attention), 454M (without), and 457M (external anchor). Naïve Combination refers to the unmodified addition of Gated Attention and ResFormer. Prefixes E, H, and S denote elementwise, headwise, and scalar mixing granularity. "Only Q/K Norms" applies RMSNorm solely to anchor queries/keys, while "No Norm" omits anchor normalization.

Model	ARC-c	ARC-e	Hella.	OBQA	PIQA	Wino.	Avg. Acc	PPL
Baselines								
Base Transformer	30.97	63.38	42.02	33.60	67.30	51.54	48.14	14.79
Gated Attention	30.89	64.69	42.73	34.00	67.14	53.35	48.80	14.64
ResFormer (Value Residual)	33.62	64.86	43.49	34.40	**68.88**	52.64	49.65	14.32
Naïve Combination	32.94	64.52	43.47	32.60	68.34	52.64	49.09	14.25
Internal Anchor								
E-NuResFormer (Only Q/K Norms)	31.74	64.27	43.83	33.00	68.23	52.41	48.91	14.21
H-NuResFormer (Only Q/K Norms)	**34.73**	64.06	44.45	34.20	67.57	53.04	49.68	14.21
S-NuResFormer (Only Q/K Norms)	32.85	**66.08**	43.87	34.20	68.28	53.67	49.83	14.22
E-NuResFormer (No Norm, {V,G})	32.42	64.94	43.80	35.00	67.52	51.54	49.20	14.24
H-NuResFormer (No Norm, {V,G})	31.23	64.18	43.50	34.00	67.68	54.06	49.11	14.22
S-NuResFormer (No Norm, {V,G})	31.83	64.44	43.66	34.80	68.44	52.49	49.28	14.24
E-NuResFormer	33.70	65.07	44.23	33.60	68.34	53.12	49.68	14.15
H-NuResFormer	33.79	65.11	44.09	33.40	67.41	52.72	49.42	14.17
S-NuResFormer	32.42	64.02	43.93	33.20	67.90	53.20	49.11	14.17
External Anchor								
E-ExoFormer (No Norm)	32.08	63.76	43.49	34.40	68.77	**55.64**	49.69	14.30
Dynamic E-ExoFormer	33.36	65.87	**44.54**	34.40	68.28	55.17	**50.27**	**14.09**
E-ExoFormer	32.42	64.65	44.28	**36.40**	67.74	53.59	49.85	14.13
H-ExoFormer	32.17	65.87	44.00	32.40	68.23	52.72	49.23	14.14
S-ExoFormer	32.17	65.66	44.33	33.60	68.06	51.14	49.16	14.15

operator, RMSNorm projects representations onto the unit sphere while preserving directional information and removing scale differences. The unit vector retains the relative alignment between dimensions, which encodes semantic and syntactic features.

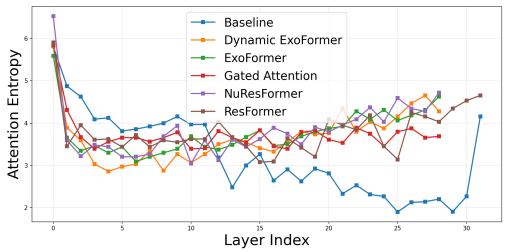
The necessity of applying RMSNorm to anchor sources is empirically supported by an analysis of the learned mixing coefficients in unnormalized models. For instance, in the ExoFormer variant without residual normalization, the proportion of near-zero coefficients (below 0.001) for λ_1 (the strength of anchor signal) is approximately triple that of the normalized variant. This suppression of the anchor pathway suggests the model is actively compensating for distributional mismatch.

We hypothesize that the naïve combination underperforms for the above reason. $\sigma(G)$ must compensate for distributional mismatch rather than filtering, resulting in instability during training (Figure 1) and worse downstream accuracy than ResFormer in isolation.

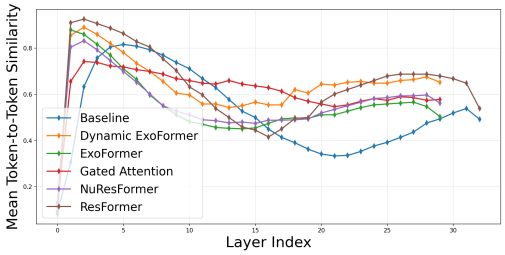
4.3 Extending Mixing to Q, K, and G Pathways

Stabilizing Q/K Residuals. Unnormalized Q/K residuals proved unstable without QKNorm, causing divergent loss. QKNorm alone stabilizes training by compressing Q/K scale and mitigating distributional mismatch; adding RMSNorm to anchor sources further improves performance and enables positive reuse. In contrast, gating logit (G) residuals were stable even without normalization, likely because sigmoid compression naturally buffers distributional shifts.

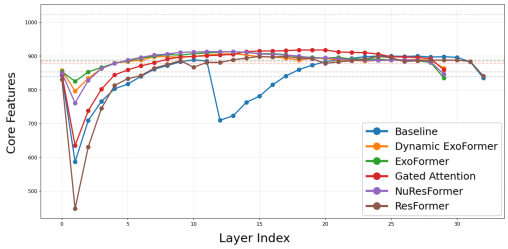
Granularity Effects. NuResFormer achieves peak downstream accuracy with scalar mixing (49.83%), while elementwise mixing yields the best perplexity (14.15) but slightly lower accuracy, suggesting fine-grained coefficients may overfit in-distribution loss. ExoFormer reverses this trend:



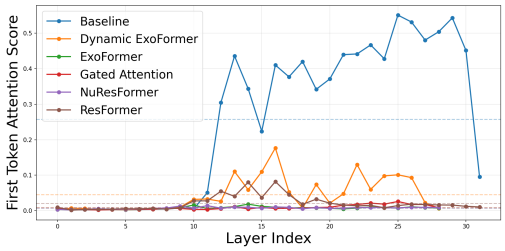
(a) Attention entropy by layer.



(b) Average cosine similarity between different token representations within the same layer.



(c) PCA core features by layer to explain 99% variance.



(d) First-token attention (attention sink) by layer.

Figure 3: Attention-pattern and representation analysis across model variants $\sim$450M. *Elementwise* is used unless stated otherwise. Some graphs include input embeddings for comparison.

elementwise mixing achieves both highest accuracy (49.85%) and lowest perplexity (14.13), indicating that decoupling anchors enables effective exploitation of parametric freedom without generalization loss.

Emergent Head Structure in Elementwise Mixing. Visualizing the learned elementwise coefficients reveals emergent patterns specific to each attention component (Figure 2). For the value pathway (V), the heatmaps exhibit *sharp boundaries that align precisely with head blocks*. This structure emerges despite the optimizer having the freedom to set each channel independently, aligning with previous research on heads as specialized submodules for routing information (Voita et al., 2019). Conversely, queries (Q), keys (K), and gate logits (G) exhibit *finer-grained, intra-head structure*, such as alternating bands of high and low reuse ("striping"), suggesting a form of sub-head specialization. Notably, the positions of these high/low bands in Q often align with those in K.

Flexibility Enabled by Dynamic Mixing. As shown in Figure 2a, $\lambda_{n,1}/\lambda_{n,2}$ for queries, keys, and gating logits are substantially more uniformly distributed across channels and layers in the dynamic variant. Rather than converging to a fixed, layer-specific reuse policy, the model can modulate the strength of the anchor pathway for each component in real time based on the input sequence, enabling more freedom in expression in the mixing coefficients themselves.

4.4 Mix-Compress-Refine Theory, Gated Attention, and ResFormer

As shown in Figure 3, the baseline Transformer's behavior aligns with the Mix-Compress-Refine theory proposed by Queipo-de-Llano et al. (2025): (1) it begins with a high attention entropy phase for broad, contextual integration of token information, (2) transitions into a compression valley marked by dominant attention sinks and a drastic drop in core features that halt mixing and reduce representational dimensionality to filter out useless information, (3) concludes with a sudden rise in attention entropy as sinks dissipate, enabling the refined, token-specific processing necessary for generation.

While these three stages emerge naturally in standard Transformers, they are not optimized for computational efficiency. The model expends computational capacity collapsing contextual information, only to subsequently reconstruct it, indicating inefficiency in the standard architecture (Figure 3c).

Table 2: Metrics averaged across layers for ~ 450M models.

Model	Attention Sink	Token Similarity	PCA Core Features
Baseline	0.2580	**0.5364**	839.0
Dynamic ExoFormer	0.0453	0.6606	885.0
ExoFormer	0.0078	0.5682	**890.1**
Gated	0.0091	0.6327	880.0
NuResFormer	**0.0077**	0.5785	887.9
ResFormer	0.0212	0.6558	853.8

Gated Attention and residual mixing improve performance by targeting stage 2. We propose that the performance gains from Gated Attention and ResFormer may partly stem from their effect on the model's second, compression stage.

The gating mechanism addresses the model's need to filter irrelevant context (core function of stage 2). By selectively modulating information flow at every layer, gated attention provides a form of distributed filtering, reducing the need for a sharp, dedicated compression phase. This interpretation is supported by the absence of a clear compression valley and a drastic reduction in attention sink magnitude as shown in Figure 3.

Similarly, in models employing residual mixing, the residual pathways may provide an implicit, learned alternative to abrupt sink-based filtering. The learned blending reduces the model's reliance on extreme, attention-sink-driven compression to isolate useful signals. This is supported by the milder compression valley observed in such models and their notable reduction in attention sink magnitude (Figure 3)

4.5 The Offloading Hypothesis: Specialization via Exogenous Anchors

The architectural decoupling in ExoFormer enables an interesting functional specialization, which we formalize as the Offloading Hypothesis. By providing a dedicated, high-fidelity source of token identity, the exogenous anchor allows sequential layers to offload the preservation of static features and specialize almost exclusively in the final "refinement" stage (Stage 3).

This specialization is evident in token-similarity trajectories (Figure 3b), where a local minimum marks the onset of Stage 3. Crucially, ExoFormer variants and NuResFormer spend approximately two-thirds of their layers in this refinement stage, the highest proportion of any model, compared to one-third for the baseline. In a standard architecture, the residual stream must concurrently carry two conflicting types of information: (1) distinct, static features that preserve token identity ("What token am I?"), and (2) transformed, task-relevant features that evolve through the layers to support next-token prediction. Over-smoothing in standard models catastrophically loses the first type, crippling the model's ability to route information effectively. ExoFormer circumvents this tension by introducing a persistent external anchor that is reinjected at every layer, guaranteeing access to high-fidelity token identity.

When we remove the exogenous anchor during inference, the model suffers a catastrophic failure: core features plummet drastically and reach 321 in the final layer, while token-to-token similarity peaks at 93%. The layers were optimized under the specific assumption that identity preservation was offloaded to the anchor. When that anchor is removed, they have no mechanism to maintain token distinctiveness.

5 Conclusion

We introduced ExoFormer, a novel Transformer architecture that decouples token identity preservation from computational refinement by learning dedicated exogenous anchor projections. Through a unified normalized mixing framework across queries, keys, values, and gates, ExoFormer resolves the architectural tension inherent in reusing first-layer projections. These results validate the Offloading Hypothesis, suggesting that externalizing identity preservation enables sequential layers to

specialize exclusively in high-level feature transformation, providing an efficient architectural path for enhancing large language models.

REFERENCES

Noah Amsel, David Persson, Christopher Musco, and Robert Gower. The polar express: Optimal matrix sign methods and their application to the muon algorithm. *CoRR*, abs/2505.16932, 2025. doi: 10.48550/ARXIV.2505.16932. URL https://doi.org/10.48550/arXiv.2505.16932.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. PIQA: reasoning about physical commonsense in natural language. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pp. 7432–7439. AAAI Press, 2020. doi: 10.1609/AAAI.V34I05.6239. URL https://doi.org/10.1609/aaai.v34i05.6239.

Lizhang Chen, Jonathan Li, Kaizhao Liang, Baiyu Su, Cong Xie, Nuo Wang Pierse, Chen Liang, Ni Lao, and Qiang Liu. Cautious weight decay. *CoRR*, abs/2510.12402, 2025. doi: 10.48550/ARXIV.2510.12402. URL https://doi.org/10.48550/arXiv.2510.12402.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. Palm: Scaling language modeling with pathways. *J. Mach. Learn. Res.*, 24:240:1–240:113, 2023. URL https://jmlr.org/papers/v24/22-1144.html.

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. *CoRR*, abs/1803.05457, 2018. URL http://arxiv.org/abs/1803.05457.

Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/67d57c32e20fd0a7a302cb81d36e40d5-Abstract-Conference.html.

Alexandre de Brébisson and Pascal Vincent. The z-loss: a shift and scale invariant classification loss belonging to the spherical family. *CoRR*, abs/1604.08859, 2016. URL http://arxiv.org/abs/1604.08859.

Alex Henry, Prudhvi Raj Dachapally, Shubham Shantaram Pawar, and Yuxuan Chen. Query-key normalization for transformers. In Trevor Cohn, Yulan He, and Yang Liu (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020*, volume EMNLP 2020 of *Findings of ACL*, pp. 4246–4253. Association for Computational Linguistics, 2020. doi: 10.18653/V1/2020.FINDINGS-EMNLP.379. URL https://doi.org/10.18653/v1/2020.findings-emnlp.379.

Jingyuan Liu, Jianlin Su, Xingcheng Yao, Zhejun Jiang, Guokun Lai, Yulun Du, Yidao Qin, Weixin Xu, Enzhe Lu, Junjie Yan, Yanru Chen, Huabin Zheng, Yibo Liu, Shaowei Liu, Bohong Yin, Weiran He, Han Zhu, Yuzhi Wang, Jianzhou Wang, Mengnan Dong, Zheng Zhang, Yongsheng Kang, Hao Zhang, Xinran Xu, Yutao Zhang, Yuxin Wu, Xinyu Zhou, and Zhilin Yang. Muon is scalable for LLM training. *CoRR*, abs/2502.16982, 2025. doi: 10.48550/ARXIV.2502.16982. URL https://doi.org/10.48550/arXiv.2502.16982.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? A new dataset for open book question answering. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii (eds.), *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pp. 2381–2391. Association for Computational Linguistics, 2018. doi: 10.18653/V1/D18-1260. URL https://doi.org/10.18653/v1/d18-1260.

Guilherme Penedo, Hynek Kydlícek, Loubna Ben Allal, Anton Lozhkov, Margaret Mitchell, Colin A. Raffel, Leandro von Werra, and Thomas Wolf. The fineweb datasets: Decanting the web for the finest text data at scale. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/370df50ccfdf8bde18f8f9c2d9151bda-Abstract-Datasets_and_Benchmarks_Track.html.

Zihan Qiu, Zekun Wang, Bo Zheng, Zeyu Huang, Kaiyue Wen, Songlin Yang, Rui Men, Le Yu, Fei Huang, Suozhi Huang, Dayiheng Liu, Jingren Zhou, and Junyang Lin. Gated attention for large language models: Non-linearity, sparsity, and attention-sink-free. *CoRR*, abs/2505.06708, 2025. doi: 10.48550/ARXIV.2505.06708. URL https://doi.org/10.48550/arXiv.2505.06708.

Enrique Queipo-de-Llano, Alvaro Arroyo, Federico Barbero, Xiaowen Dong, Michael M. Bronstein, Yann LeCun, and Ravid Shwartz-Ziv. Attention sinks and compression valleys in llms are two sides of the same coin. *CoRR*, abs/2510.06477, 2025. doi: 10.48550/ARXIV.2510.06477. URL https://doi.org/10.48550/arXiv.2510.06477.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pp. 8732–8740. AAAI Press, 2020. doi: 10.1609/AAAI.V34I05.6399. URL https://doi.org/10.1609/aaai.v34i05.6399.

Noam Shazeer. GLU variants improve transformer. *CoRR*, abs/2002.05202, 2020. URL https://arxiv.org/abs/2002.05202.

Han Shi, Jiahui Gao, Hang Xu, Xiaodan Liang, Zhenguo Li, Lingpeng Kong, Stephen M. S. Lee, and James T. Kwok. Revisiting over-smoothing in BERT from the perspective of graph. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL https://openreview.net/forum?id=dUV91uaXm3.

Jianlin Su, Murtadha H. M. Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024. doi: 10.1016/J.NEUCOM.2023.127063. URL https://doi.org/10.1016/j.neucom.2023.127063.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp.

5998–6008, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. In Anna Korhonen, David R. Traum, and Lluís Màrquez (eds.), *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pp. 5797–5808. Association for Computational Linguistics, 2019. doi: 10.18653/V1/P19-1580. URL https://doi.org/10.18653/v1/p19-1580.

Da Xiao, Qingye Meng, Shengping Li, and Xingyuan Yuan. Muddformer: Breaking residual bottlenecks in transformers via multiway dynamic dense connections. In *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=qkhgzNiEdj.

Zhenda Xie, Yixuan Wei, Huanqi Cao, Chenggang Zhao, Chengqi Deng, Jiashi Li, Damai Dai, Huazuo Gao, Jiang Chang, Kuai Yu, Liang Zhao, Shangyan Zhou, Zhean Xu, Zhengyan Zhang, Wangding Zeng, Shengding Hu, Yuqing Wang, Jingyang Yuan, Lean Wang, and Wenfeng Liang. mhc: Manifold-constrained hyper-connections, 2026. URL https://arxiv.org/abs/2512.24880.

Greg Yang, Edward J. Hu, Igor Babuschkin, Szymon Sidor, Xiaodong Liu, David Farhi, Nick Ryder, Jakub Pachocki, Weizhu Chen, and Jianfeng Gao. Tensor programs V: tuning large neural networks via zero-shot hyperparameter transfer. *CoRR*, abs/2203.03466, 2022. doi: 10.48550/ARXIV.2203.03466. URL https://doi.org/10.48550/arXiv.2203.03466.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? In Anna Korhonen, David R. Traum, and Lluís Màrquez (eds.), *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pp. 4791–4800. Association for Computational Linguistics, 2019. doi: 10.18653/V1/P19-1472. URL https://doi.org/10.18653/v1/p19-1472.

Daquan Zhou, Bingyi Kang, Xiaojie Jin, Linjie Yang, Xiaochen Lian, Qibin Hou, and Jiashi Feng. Deepvit: Towards deeper vision transformer. *CoRR*, abs/2103.11886, 2021. URL https://arxiv.org/abs/2103.11886.

Zhanchao Zhou, Tianyi Wu, Zhiyun Jiang, Fares Obeid, and Zhenzhong Lan. Value residual learning. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, pp. 28341–28356. Association for Computational Linguistics, 2025. URL https://aclanthology.org/2025.acl-long.1375/.

Defa Zhu, Hongzhi Huang, Zihao Huang, Yutao Zeng, Yunyao Mao, Banggu Wu, Qiyang Min, and Xun Zhou. Hyper-connections. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=9FqARW7dwB.

A Anchor Reliance and Hidden State Similarity

B Gate Activation Profiles and First-Layer Selectivity

NuResFormer's first layer exhibits **a high mean gate activation** (approximately 0.4-0.5), indicating its gating mechanism is not highly suppressive, allowing roughly half of the attention output to pass through. This contrasts sharply with standalone Gated Attention, where the first-layer mean activation is significantly lower (approximately 0.2). Following this initial peak, gate activations fall rapidly in intermediate layers before rising steadily again in deeper layers.

This pattern provides direct empirical support for the architectural tension hypothesized. When the first layer also serves as the residual anchor (as in NuResFormer), it faces conflicting objectives: its

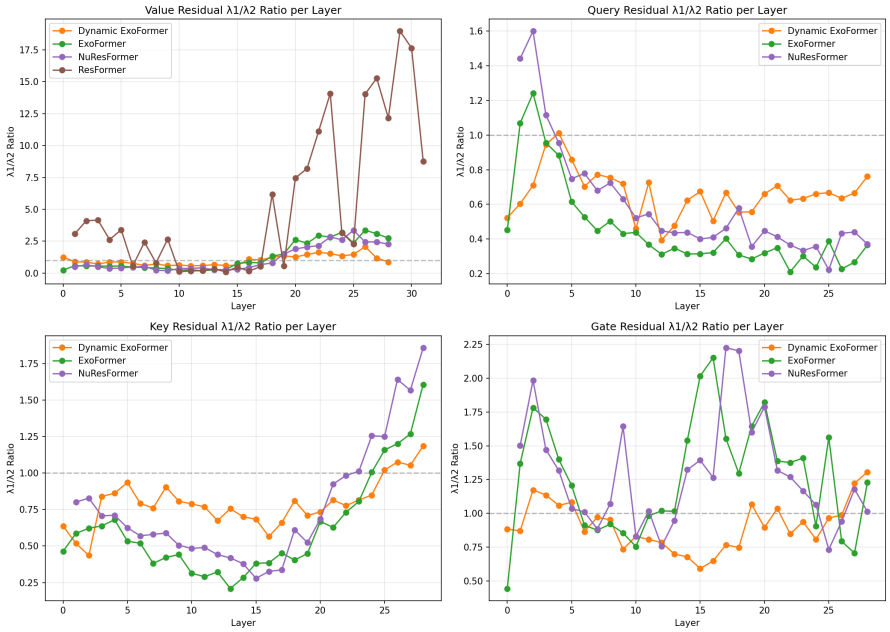
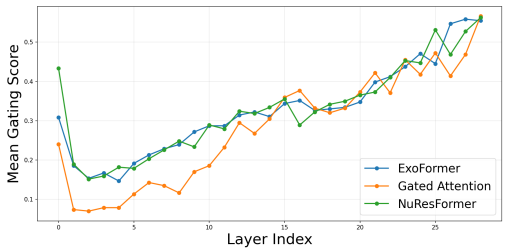
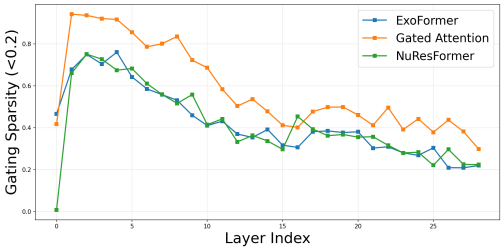


Figure 4: The ratio $\lambda_{n,1}/\lambda_{n,2}$ plotted for each component (Q, K, V, G) across layers for models using *elementwise* mixing for model variants $\sim 450M$. Values greater than 1 indicate stronger reliance on the anchor signal, while values less than 1 indicate preference for current-layer projections.



(a) Mean gating score ($\sigma(G)$) per layer. Higher values indicate less suppression of attention output.



(b) Gating sparsity per layer (% activations < 0.2). Higher sparsity indicates more selective, suppressive gating.

Figure 5: Analysis of gating behavior across model variants $\sim 450M$.

gate logits G_1 must perform effective, context-dependent selection for the first layer's own computation while also producing a reusable anchor signal G_{anc} for all subsequent layers. The high first-layer gate activation suggests a resolution: the layer adopts a permissive gating policy to ensure the anchor gate logits retain broad, generally useful information, sacrificing some first-layer selectivity in the process.

ExoFormer exhibits an attenuated version of the same profile (Figure 5a); its first-layer activation is elevated compared to standalone gating but lower than NuResFormer's. As shown in Figure 5b, offloading the anchor role allows ExoFormer to partially restore the first layer's capacity for selective gating.

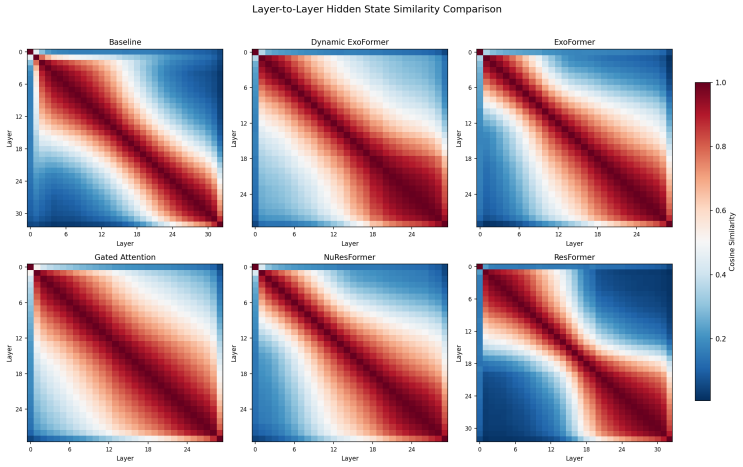


Figure 6: Pairwise cosine similarity of hidden states across transformer layers for models using *elementwise* mixing $\sim 450M$. Brighter colors indicate higher similarity.

C Complexity Analysis

We present a complexity analysis of ExoFormer variants. Let L be the number of Transformer layers and d the model dimension (d_{model}). We omit RMSNorm because it is negligible in terms of parameters and computation.

C.1 Parameter Overhead

We analyze the parameter overhead of ExoFormer variants relative to a baseline Transformer with Gated Attention. Throughout, we ignore input and output embedding parameters, as they are shared across all models.

Baseline Parameters: The baseline Transformer with Gated Attention has parameters per layer for the attention projections (queries, keys, values, gate logits and output) and the two-layer SwiGLU feed-forward network. The attention and output projections require $5d^2$ parameters (five $d \times d$ matrices), and the FFN requires $6d^2$ parameters (assuming expansion factor 4). Thus, ignoring the input and output embedding, the total parameters for L layers are:

$$P_{\text{base}} = L \cdot \left(5d^2 + 6d^2\right) = 11Ld^2. \tag{13}$$

Exogenous Anchor Parameters: ExoFormer introduces dedicated projection matrices for the exogenous anchor: four $d \times d$ matrices, one for each attention component (Q, K, V, G):

$$\Delta P_{\text{anchor}} = 4d^2. \tag{14}$$

Static Mixing Parameters: For static mixing with elementwise granularity, each layer learns two mixing coefficients (one for anchor, one for current projection) for each of the four components. This amounts to $8d$ parameters per layer:

$$\Delta P_{\text{static}} = L \cdot 8d. \tag{15}$$

Dynamic Mixing Parameters: The Dynamic Mixing (DM) module adds a small MLP per layer. With $d_{\text{DM}} = 16$ and $d_{\text{out}} = 8$, the dominant term is from the first weight matrix ($d \cdot d_{\text{DM}}$). The second weight matrix and biases (totaling 128 parameters) are negligible for typical d:

$$\Delta P_{\text{DM}} \approx L \cdot (d \cdot d_{\text{DM}}). \tag{16}$$

Total Parameter Overhead: Static ExoFormer adds:

$$\Delta P_{\text{static ExoFormer}} = 4d^2 + 8Ld. \tag{17}$$

Dynamic ExoFormer additionally includes the DM module:

$$\Delta P_{\text{dynamic ExoFormer}} = 4d^2 + 8Ld + Ld \cdot d_{\text{DM}}. \tag{18}$$

Parameter Ratio: The extra parameter ratio relative to baseline simplifies to:

$$R_P^{(\text{static})} = \frac{4d^2 + 8Ld}{11Ld^2} = \frac{4}{11L} + \frac{8}{11d}, \tag{19}$$

$$R_P^{(\text{dynamic})} = \frac{4d^2 + 8Ld + Ld \cdot d_{\text{DM}}}{11Ld^2} = \frac{4}{11L} + \frac{8 + d_{\text{DM}}}{11d}. \tag{20}$$

For values ($L = 32, d = 1024, d_{\text{DM}} = 16$), this corresponds to approximately 1.2% overhead for static ExoFormer and 1.3% for dynamic ExoFormer, a modest increase given the observed performance gains.

C.2 Computational Overhead (FLOPs)

We estimate the floating-point operations (FLOPs) per token during the forward pass, considering matrix-vector multiplications as the dominant factor ($2d^2$ FLOPs per $d \times d$ matrix). We include the cost of elementwise operations for mixing, although they are computationally smaller ($O(d)$) compared to projections ($O(d^2)$).

Baseline FLOPs: The baseline usage corresponds to the parameters utilized at every layer. With $P_{\text{base}} = 11Ld^2$ (excluding embeddings), the computational cost per token is:

$$C_{\text{base}} \approx 2 \cdot P_{\text{base}} = 22Ld^2. \tag{21}$$

Exogenous Anchor FLOPs: The exogenous anchor projections are computed only once per token using the input embeddings, regardless of the network depth. For the four projection matrices ($W^Q_{\text{anc}}, W^K_{\text{anc}}, W^V_{\text{anc}}, W^G_{\text{anc}}$), the cost is:

$$\Delta C_{\text{anchor}} \approx 2 \cdot (4d^2) = 8d^2. \tag{22}$$

Crucially, this cost is constant and does not scale with the number of layers L.

Dynamic Mixing FLOPs: The Dynamic Mixing module operates at every layer. The computational cost includes the projection of the MLP ($d \to d_{\text{DM}}$) and the elementwise mixing operations. The mixing involves two multiplications (coefficients $\times$ anchor/current projections) and one addition per element for the four components (Q, K, V, G), totaling $12d$ FLOPs per layer:

$$\Delta C_{\text{DM}} \approx L \cdot (2 \cdot d \cdot d_{\text{DM}} + 12d). \tag{23}$$

Total FLOPs Overhead: The total computational overhead ratio is:

$$R_{\text{FLOPs}} = \frac{\Delta C_{\text{anchor}} + \Delta C_{\text{DM}}}{C_{\text{base}}} = \frac{8d^2 + L(2dd_{\text{DM}} + 12d)}{22Ld^2} = \frac{8}{22L} + \frac{d_{\text{DM}}}{11d} + \frac{6}{11d}. \tag{24}$$

For values ($L = 32, d = 1024, d_{\text{DM}} = 16$), the overhead is dominated by the anchor projection ($\approx 1.1\%$), followed by the dynamic module projection ($\approx 0.14\%$), and the elementwise mixing operations ($\approx 0.05\%$), resulting in a total FLOPs increase of approximately 1.33%.

C.3 EFFICIENCY-PERFORMANCE TRADE-OFF

The FLOPs analysis presented above provides a lower-bound estimate of the computational overhead. In practice, the real-world slowdown may be higher (we observe an increase in latency per token of approximately 8–15% for our reference implementation prioritizing modularity and interpretability over speed, compared to the theoretical 1.3% FLOPs increase).

We emphasize that our primary contribution is a novel architecture that unifies attention projection mixing, not a production-optimized layer ready for deployment. The net performance benefit even in our setting, evidenced by accuracy gains of 1.5 points and data efficiency improvements of 1.5× against Gated Attention, remains positive even when accounting for the measured runtime overhead, suggesting that an optimized kernel implementation would close the gap with the theoretical 1.3% overhead.

D HYPERPARAMETERS

All models use a modern pre-normalized Transformer architecture with SwiGLU activations (Shazeer, 2020), QKNorm (Henry et al., 2020), and rotary position embeddings (Su et al., 2024). We follow prior work by initializing projection and classification layers to zero (Yang et al., 2022), removing bias terms (except for the dynamic mixing module) (Chowdhery et al., 2023), applying z-loss regularization (de Brébisson & Vincent, 2016), and disabling dropout. Training is performed on the FineWeb-Edu dataset (Penedo et al., 2024). Specifically, the ~ 450M parameter models are trained on 10B tokens, while the $\sim$1B parameter models are trained on 20B tokens.

We optimize matrix parameters with Muon (Polar Express variant (Amsel et al., 2025)), applying a cautious weight decay of 0.1 (Chen et al., 2025), while 1D parameters are trained using AdamW without weight decay. We selected Muon as a state-of-the-art optimizer for large-scale LLM training (Liu et al., 2025), though we anticipate that the observed improvements should be optimizer-agnostic. Gradient norms are clipped to 1.0, and all models follow the same optimization setup, to ensure fair comparison across architectures. Training uses a global batch size of 262,144 tokens and a sequence length of 2,048.

Full hyperparameters are provided in the Appendix. All experiments are conducted on a single NVIDIA H100 80GB GPU using native BF16 precision, with FlashAttention (Dao et al., 2022) enabled.

Evaluation Details We evaluate each benchmark example using a 5-shot prompt. To reduce length-related bias, we report length-normalized accuracy whenever possible. Perplexity is measured on the FineWeb-Edu validation set containing 100 million tokens.

We report results on 6 multiple-choice benchmarks: ARC_CHALLENGE, ARC_EASY (Clark et al., 2018), HELLASWAG (Zellers et al., 2019), OPENBOOKQA (Mihaylov et al., 2018), PIQA (Bisk et al., 2020), and WINOGRANDE (Sakaguchi et al., 2020).

Table 3: Training hyperparameters for models with different configurations for the main $\sim 450M$ models. Layer depth was reduced for gated variants to maintain comparable parameter counts. All models were trained on the FineWeb-Edu dataset.

Hyperparameter	No Gate	With Gate	Decoupled (ExoFormer)
Parameters (M)	454	453	457
Layers	32	29	29
Attention Heads		16	
Hidden Dimension		1024	
FFN Dimension		4096	
Tie Word Embedding		False	
Vocabulary Size		57,601	
Activation Function		SwiGLU	
Position Embedding		RoPE (θ = 500,000)	
Sequence Length		2048	
Batch Size (tokens)		262,144	
Training Tokens		10B	
Warmup Steps		1000	
Warmdown Steps		7630 (20%)	
Total Steps		38,147	
Optimization			
Optimizer		Muon + AdamW	
Muon Learning Rate		0.01	
AdamW Learning Rate		0.003	
Learning Rate Schedule		Linear	
Adam β		(0.9, 0.95)	
Muon Momentum		0.95	
Gradient Clip		1.0	
Dropout		0.0	
Cautious Weight Decay		True	
Muon Weight Decay		0.1	
AdamW Weight Decay		0.0	
Z Loss Weight		1e-5	
RMSNorm Epsilon		1e-6	
QK Normalization		True	

Table 4: Training hyperparameters for 1B parameter models. Most hyperparameters remain identical to those in Table 3 except for depth, width, and learning rates. Batch size was kept constant, requiring an increase in total steps to reach 20B training tokens.

Hyperparameter	Gated Attention	Dynamic ExoFormer
Parameters (B)	1.01	1.02
Layers	32	
Hidden Dimension	1536	
FFN Dimension	6144	
Optimization		
Muon Learning Rate	0.003	
AdamW Learning Rate	0.001	
Training Tokens	20B	
Total Steps	76,293	