How Training Window Length Shapes Neural Language Model Weights

Zacharie Bugaud [1]

Abstract

We study how the training context window length w is associated with changes in the learned weights of language models. By training identical architectures across three families (Transformer, 81.1M; GRU, 39.5M; RetNet, 41.5M) on ten window lengths ($w \in \{128, \ldots, 65536\}$) with a fixed 500M-token budget, we characterize the geometry of window-induced weight changes. Our primary finding: in a matched-step regime where all models receive identical optimizer steps and tokens per step, adjacent-window final-weight angular distance forms an approximate plateau ($\sim$0.175, corresponding to $\sim$31.5°), spanning a 32$\times$ range from $w = 2048$ to $w = 65{,}536$. This regularity is reproduced across two independent Transformer seeds with near-identical magnitudes. Two qualifications matter. *Window-effect vectors* (the weight displacement induced by doubling w, defined as $\delta_{w \to 2w,s} = (\theta_{2w,s} - \theta_{0,s}) - (\theta_{w,s} - \theta_{0,s})$ where $\theta_{0,s}$ is the shared initialization for seed s) are completely orthogonal across seeds (cosine ≈ 0): the *magnitude* of the window-induced change is reproducible, but its *direction* in weight space is initialization-dependent. Consecutive window doublings within the same seed produce *anti-correlated* displacement vectors (cosine ≈ -0.45), though we show this is largely a geometric artifact of finite differences sharing a middle term. Functional validation via output KL divergence at multiple evaluation lengths ($T \in \{1024, 2048, 4096\}$) provides evidence that adjacent-window models (i.e., models trained at consecutive powers of two, e.g., w and $2w$) are moderately more similar than cross-seed models (sym-KL $\approx$ 0.27 vs. 0.31), with the gap stable across evaluation lengths and bootstrap 95% CIs non-overlapping for most same-w adjacent/cross-seed comparisons, though some overlaps occur at extreme window sizes. Zero-shot weight transfer reveals strong asymmetry: short-to-long fails catastrophically while long-to-short degrades substantially. RoPE frequency utilization is uniform across all windows in both final weights and learned updates ($N_{\text{eff}} \approx 32$ per head), providing no evidence for norm-level frequency condensation.

1. Introduction

The context window w, the maximum sequence length during training, is a fundamental hyperparameter of neural language models (Vaswani et al., 2017). It determines the longest-range dependencies the model can observe, the computational cost per step, and the model's ability to generalize to longer sequences at inference (Press et al., 2022).

Despite the centrality of the context window, remarkably little is known about how w shapes the *learned weights* of the model. Most work on context length focuses on evaluation-time behavior: can a model trained on $w = 2048$ process sequences of length 8192? The answer is generally no for RoPE-based models (Su et al., 2024; Chen et al., 2023), but the weight-level mechanisms underlying this failure remain poorly understood.

We conduct a controlled mechanistic study: train identical architectures across three model families on ten window lengths with matched data budgets, then systematically compare the resulting weights. The main control is the matched-step regime, which controls optimizer steps and tokens per step for $w \geq 2048$, while validating results across two independent Transformer seeds. We find four main patterns:

1. **Stable per-octave distance plateau** (Section 4). In the matched-step regime, adjacent-window angular distance forms an approximate plateau ($\sim$0.175 in final weights, $\sim$0.24 in update vectors) across a 32$\times$ window range. Both seeds reproduce the same distance magnitudes ($r > 0.99$ correlation). However,

[1]Astera Institute, Berkeley, CA, USA. Correspondence to: Zacharie Bugaud <zacharie@astera.org>.

Accepted to the Mechanistic Interpretability Workshop, MechInterp@ICML 2026, Seoul, South Korea.

the *direction* of the window-induced displacement is initialization-dependent: window-effect vectors are completely orthogonal across seeds (cosine ≈ 0), meaning the magnitude regularity does not reflect a universal update direction.

2. **Anti-correlated consecutive doublings** (Section 4). Within the same seed, consecutive window-doubling displacement vectors are anti-correlated (cosine ≈ -0.45 in the matched-step regime). However, an analytic null model shows that $\sim$95% of this anti-correlation is explained by the geometric structure of finite differences sharing a middle term. Non-overlapping displacement pairs (e.g., 2048→4096 vs. 8192→16384) have cosine ≈ 0, suggesting that the anti-correlation does not reflect oscillatory trajectories.

3. **Transfer asymmetry** (Section 5). Short-to-long weight transfer fails catastrophically at lengths beyond training, while long-to-short transfer degrades substantially (0.9 nat gap at length 128, corresponding to $\sim$2.5$\times$ higher perplexity). Brief fine-tuning (100 steps) closes the gap, suggesting the specialization is shallow in the loss landscape.

4. **Null frequency condensation in both weights and updates** (Section 3). RoPE frequency utilization (measured via weight norms) is nearly uniform across all window lengths in both final weights and learned updates ($N_{\text{eff}} \approx 32/32$ per head), providing no evidence that the length barrier operates through norm-level frequency selection.

The paper presents findings in order of investigation: frequency condensation hypothesis (Section 3), broader weight geometry (Section 4), transfer experiments (Section 5), and cross-architecture comparison (Section 6).

These results motivate testing whether full-model updates outperform positional-parameter-only updates for context extension. The strongest claims are limited to the *matched-step regime* ($w \geq 2048$), and distance regularities are validated across two independent seeds. We distinguish reproducible *magnitude* patterns (well-supported) from claims about the *direction* or *interpretation* of weight changes (which our data do not support). Throughout, we frame the contribution as a careful empirical characterization of weight signatures *associated with* training context length, not yet a causal explanation of long-context behavior.

2. Experimental Setup

Architectures. We train three architectures (total unique parameters in parentheses; all use tied input/output embeddings):

- **Transformer** (81.1M): Decoder-only with SwiGLU (Shazeer, 2020), RMSNorm (Zhang & Sennrich, 2019), multi-head attention ($h = 12$), RoPE (Su et al., 2024) (base $b_{\text{RoPE}} = 10{,}000$), tied embeddings. $d = 768$, $L = 6$, $d_{\text{ff}} = 2048$, vocabulary $|\mathcal{V}| = 50{,}257$ (GPT-2 tokenizer). Embeddings: $50{,}257 \times 768 = 38.6$M (tied with output projection); non-embedding parameters: $6 \times (4 \times 768^2 + 3 \times 768 \times 2048) = 42.5$M.

- **GRU-LM** (39.5M): cuDNN-optimized Gated Recurrent Unit (Cho et al., 2014) with learned input/output projections. $d = 640$, $L = 3$. No explicit positional encoding.

- **RetNet-LM** (41.5M): Linear attention with per-head exponential decay (Sun et al., 2023). $d = 512$, $h = 8$, $L = 6$, $d_{\text{ff}} = 1536$. Decay rates $\gamma \in [0.9, 0.999]$.

Training protocol. Each model is trained on OpenWebText (Gokaslan & Cohen, 2019) for a fixed budget of 500M tokens with AdamW (Loshchilov & Hutter, 2019) ($\beta = (0.9, 0.95)$, weight decay 0.1), cosine LR schedule (peak 6×10^{-4}, linear warmup for first 5% of steps, minimum LR 6×10^{-5}), and BFloat16 precision on 8$\times$ NVIDIA H100 80GB GPUs using FlashAttention-2. No dropout is used. The intended intervention is the context window $w \in \{2^7, 2^8, \ldots, 2^{16}\}$ (i.e., 128 to 65,536), spanning a 512$\times$ range; however, changing w necessarily changes sequence composition and batch structure (see below). Transformer models are trained with 2 seeds; GRU and RetNet with 1 seed each (30 total models + 10 second-seed transformers = 40 models).

Controlled comparison and step-count analysis. By fixing the data budget, we ensure that differences in learned weights are not due to different amounts of training data. Batch size B and gradient accumulation G are adjusted per window (Table 1) such that models at $w \geq 2048$ all receive the *same* number of optimizer steps ($\sim$1907) and the same tokens per step (262,144), while shorter-window models receive more steps (e.g., $\sim$15,258 at $w = 128$). This creates two regimes:

- **Matched-step regime** ($w \in \{2048, 4096, 8192, 16384, 32768, 65536\}$): All 6 models receive $\sim$1907 steps with 262,144 tokens/step. The primary varying factor is window length, though the number of independent sequences per step decreases from 128 to 4 as w grows (Table 1). This regime spans a 32$\times$ range and is the focus of our strongest claims.

- **Unmatched regime** ($w \leq 1024$): These models receive 2–8$\times$ more optimizer steps with fewer tokens

Table 1. Training configuration per window length. Tokens/step = $B \times G \times w$.

w	B	G	Seqs/step	Tokens/step	Steps
128	16	16	256	32,768	15,258
256	16	8	128	32,768	15,258
512	8	16	128	65,536	7,629
1,024	8	16	128	131,072	3,814
2,048	8	16	128	262,144	1,907
4,096	8	8	64	262,144	1,907
8,192	8	4	32	262,144	1,907
16,384	4	4	16	262,144	1,907
32,768	2	4	8	262,144	1,907
65,536	1	4	4	262,144	1,907

per step. Comparisons involving these windows are confounded by optimization trajectory length.

We present results for both regimes but mark unmatched-regime conclusions as requiring further validation (Section 7).

We use simple concatenation packing without document-boundary attention masks. We acknowledge the following confounds even within the matched-step regime: (1) the number of independent sequences (and hence gradient diversity) varies from 128 to 4, (2) document-boundary frequency within each sequence changes with w, and (3) longer sequences may span multiple unrelated documents. These limitations should be kept in mind when interpreting the matched-step results.

3. RoPE Frequency Utilization: A Null Result

3.1. Background and Hypothesis

RoPE (Su et al., 2024) defines $d_h/2$ frequency bands per head (with head dimension d_h) with angular frequencies $\theta_j = \text{base}^{-2j/d_h}$, for $j = 0, \ldots, d_h/2 - 1$ (in our Transformer, $d_h = 64$, giving 32 bands per head). One hypothesis is that models trained on short windows should *condense* their weight mass onto high-frequency bands (which complete multiple rotations within w), while longer-window models should distribute mass more uniformly across all bands. We test this by measuring the L2 norm of Q and K weight blocks corresponding to each frequency band pair. Specifically, for head h in layer ℓ, let $W^Q_{\ell,h} \in \mathbb{R}^{d_h \times d}$ be the query projection and let the j-th frequency band correspond to rows $2j{:}2j{+}2$. We define the per-band utilization as $p_j = \|W^Q_{\ell,h}[2j{:}2j{+}2, :]\|^2_F / \sum_{j'} \|W^Q_{\ell,h}[2j'{:}2j'{+}2, :]\|^2_F$ and compute the effective number of active frequencies $N_{\text{eff}} = \exp(H)$ where $H = -\sum_j p_j \log p_j$ is the entropy of the utilization profile. Values are averaged over heads and layers unless stated otherwise.

Table 2. RoPE frequency utilization across window lengths (averaged over layers). Top: final weights. Bottom: learned updates ($\Delta W = W_{\text{final}} - W_{\text{init}}$). Both show near-uniform utilization.

	$w=128$	$w=512$	$w=2048$	$w=8192$	$w=32768$	$w=65536$
Final weights:						
N_{eff} (Q)	31.7	31.9	31.8	31.9	31.9	31.9
N_{eff} (K)	31.6	31.9	31.9	31.9	32.0	32.0
Learned updates (ΔW):						
N_{eff} (Q)	31.8	—	31.7	31.7	—	31.7
N_{eff} (K)	31.7	—	31.4	31.5	—	31.8
Initialization (same for all):						
N_{eff} (Q/K)				32.0		

3.2. Results

Finding 1: Weight norms show no frequency condensation. Table 2 shows N_{eff} per head is approximately 31.7–32.0 out of a maximum 32 bands for all models from $w = 128$ to $w = 65{,}536$. The number of top-ranked frequency bands required to explain $\geq 50\%$ of total norm mass is similarly stable at 14–16 per head across all windows and layers. This also holds within the matched-step regime: models at $w = 2048$ through $w = 65{,}536$ all show $N_{\text{eff}} \approx 32$ despite identical optimization budgets.

3.3. Interpretation

The null result matters because models do *not* learn to suppress weight norms for frequency bands beyond their training window's Nyquist-like limit. **The same pattern appears in learned updates, not only in final weights** (Table 2, bottom rows), arguing against the simplest explanation that final-weight uniformity is solely inherited from initialization norms. The learned updates ΔW_Q, ΔW_K also distribute mass uniformly across all frequency bands ($N_{\text{eff}} \approx 31.5$–$31.8$), regardless of window length.

Scope and caveats. Weight-norm uniformity does *not* rule out subtler forms of frequency specialization. Specifically:

- **Phase relationships** between Q and K weights may encode window-specific structure invisible to norm analysis.
- **Attention pattern analysis** (e.g., per-frequency-band logit contributions) could reveal window-dependent behavior.
- **Downstream layers** that consume attention output may selectively amplify certain frequency bands.
- **Activation-level effects:** Even with uniform weight norms, the network may produce non-uniform frequency utilization at the activation level through selective amplification or cancellation.

Thus, our conclusion is limited to norm-level frequency condensation: the prediction is not supported by the data.

Table 3. Final-weight angular distances, same-window cross-seed baseline, and L2 distances. Init angular distance between seeds = 0.438; init L2 distance = 499. Note that cross-seed distances are dominated by initialization ($0.438/0.471 \approx 93\%$), so the ratio column should not be interpreted as a clean "window vs. training variability" effect size.

Window pair	Adj. ang	Seed ang	Ang ratio	Adj. L2	Seed L2
128 ↔ 256	0.322	0.472	0.68	367	499
256 ↔ 512	0.341	0.473	0.72	—	—
512 ↔ 1024	0.283	0.475	0.60	308	466
1024 ↔ 2048	0.213	0.470	0.45	230	433
2048 ↔ 4096	0.171	0.470	0.36	186	436
4096 ↔ 8192	0.173	0.470	0.37	188	437
8192 ↔ 16384	0.176	0.471	0.37	192	441
16384 ↔ 32768	0.175	0.471	0.37	191	440
32768 ↔ 65536	0.167	0.472	0.35	183	440

Testing subtler frequency effects will require activation-level analyses.

4. Weight Space Geometry

4.1. Methodology

We compute pairwise angular distances between models trained at different windows: $d_{\text{ang}} = \frac{1}{\pi} \arccos\left(\frac{\langle \mathbf{w}_1, \mathbf{w}_2 \rangle}{\|\mathbf{w}_1\| \|\mathbf{w}_2\|}\right)$ where $\mathbf{w}_i$ is the flattened concatenation of all weight tensors. This distance is normalized to $[0, 1]$ (a value of 0.175 corresponds to $0.175\pi \approx 31.5°$).

4.2. Results

Finding 2: Constant per-octave weight perturbation in matched-step regime. Figure 1 shows the 10×10 angular distance matrix for all Transformer models. Adjacent-window distances decrease from 0.322 ($w = 128 \leftrightarrow 256$) to 0.170 ($w = 32\text{K} \leftrightarrow 64\text{K}$). Within the matched-step regime ($w \geq 2048$), adjacent distances form an approximate plateau: 0.174, 0.176, 0.179, 0.179, 0.170 (range 0.170–0.179). Thus, each doubling corresponds to a roughly constant perturbation of $\sim$0.175 ($\approx 31.5°$). The decrease from 0.32 to 0.17 occurs in the unmatched regime ($w \leq 1024$) where shorter-window models receive up to $8\times$ more optimizer steps. Both Transformer seeds (42 and 137) produce nearly identical adjacent-distance patterns ($r > 0.99$ correlation over all nine pairs; five matched-step pairs alone: range 0.170–0.179), supporting the *magnitude* regularity.

Caution on ratio interpretation. Cross-seed final-weight angular distance ($\sim$0.471) is close to the *initialization* distance (0.438), meaning most of the cross-seed difference reflects different random initializations rather than training-induced variation. Independently initialized networks are also expected to have near-orthogonal weight vectors due to permutation symmetries of hidden units, regardless of functional similarity. Therefore, the ratio of adjacent-window to cross-seed distance (Table 3, column "Ang ratio") should not be interpreted as a clean effect size measuring "window effect divided by training variability." The main finding is the *stability* of the adjacent-window distance plateau, not its magnitude relative to the seed baseline.

Embedding decomposition. Token embeddings constitute 48% of unique parameters (38.6M/81.1M; stored as tied input/output projection). We verify that the plateau is **not** driven by embedding drift by decomposing distances into three components:

- **Non-embedding only** (42.5M attention + MLP): matched-step adjacent distance 0.178–0.190, seed distance $\sim$0.375.
- **Embedding only** (38.6M): matched-step adjacent distance 0.178–0.187, seed distance $\sim$0.50.
- **Full model**: matched-step adjacent distance 0.169–0.174, seed distance $\sim$0.471.

All three subsets exhibit the same qualitative pattern: stable plateau in the matched-step regime. The findings are not an artifact of embedding dominance.

Update-vector analysis. To remove initialization effects, we compute update vectors $\Delta_w = \theta_w - \theta_0$ (final weights minus initialization). Update-vector norms are nearly constant in the matched-step regime ($\|\Delta_w\| \approx$ 253–256 for $w \geq 2048$), indicating that matched-step models traverse similar total distances from initialization. Adjacent update-vector angular distances in the matched-step regime are 0.230–0.242, higher than final-weight distances (0.170–0.179) because final weights include the shared initialization component that pulls models toward each other. Cross-seed update-vector distances are $\sim$0.502, indistinguishable from random; however, this is expected from permutation symmetry (independently initialized networks have unaligned coordinate systems) and does not imply that the learned functions are random.

Window-effect vector reproducibility across seeds. A central question is: does window length determine a *reproducible direction* of weight change, or only a reproducible *magnitude*? We test this by computing the window-effect vector $\delta_{w \to 2w,s} = \Delta_{2w,s} - \Delta_{w,s}$ (the weight displacement induced by doubling w) and measuring cosine similarity across seeds (Table 4).

The *magnitudes* $\|\delta\|$ are highly reproducible across seeds (differences $< 4\%$), but the *directions* are completely orthogonal (cosine ≈ 0). This means window length determines a reproducible *amount* of weight change but not a universal *direction* of change in raw weight coordinates.

Table 4. Window-effect vector analysis. $\delta_{w\to 2w,s} = \Delta_{2w,s} - \Delta_{w,s}$ measures the weight displacement caused by doubling w within a given seed s. Cross-seed cosine measures whether this displacement is reproducible in direction.

Window pair	$\cos(\delta_{s42}, \delta_{s137})$	$\|\delta_{s42}\|$	$\|\delta_{s137}\|$
$128 \to 256$	0.005	365	358
$512 \to 1024$	0.008	309	305
$2048 \to 4096$	0.000	186	187
$4096 \to 8192$	0.000	188	185
$8192 \to 16384$	−0.001	192	187
$16384 \to 32768$	0.001	192	190
$32768 \to 65536$	−0.001	184	188

Table 5. Cosine similarity between consecutive window-doubling vectors within the same seed. Negative values indicate the next doubling partially reverses the previous one.

Pairs	Seed 42	Seed 137
128→256 vs 256→512	−0.488	−0.474
512→1024 vs 1024→2048	−0.302	−0.296
2048→4096 vs 4096→8192	−0.444	−0.440
4096→8192 vs 8192→16384	−0.464	−0.457
8192→16384 vs 16384→32768	−0.485	−0.472
16384→32768 vs 32768→65536	−0.485	−0.481

The directional orthogonality is expected from permutation symmetry: the same functional change maps to different coordinate directions when the hidden units are relabeled by different initializations. Thus, raw-coordinate directions are not comparable across initializations without alignment.

Anti-correlated consecutive doublings: a geometric artifact. Within the same seed, consecutive window-doubling vectors $\delta_{w\to 2w}$ and $\delta_{2w\to 4w}$ are *anti-correlated* (Table 5). However, this anti-correlation is largely a geometric consequence of finite differences sharing a middle term.

The anti-correlation is consistent across both seeds (values agree to within ± 0.02). Most of this pattern follows from the finite-difference geometry. Consecutive finite differences $\delta_{w\to 2w} = \Delta_{2w} - \Delta_w$ and $\delta_{2w\to 4w} = \Delta_{4w} - \Delta_{2w}$ share the middle term Δ_{2w} with opposite signs. Given the pairwise cosine structure of the update vectors (consecutive: $\cos \approx 0.73$; skip-1: $\cos \approx 0.70$), the analytic formula $\cos(\delta_i, \delta_{i+1}) = \frac{c_1 + c_2 - c_3 - 1}{2\sqrt{(1-c_1)(1-c_2)}}$ where $c_1 = \cos(\Delta_w, \Delta_{2w})$, $c_2 = \cos(\Delta_{2w}, \Delta_{4w})$, and $c_3 = \cos(\Delta_w, \Delta_{4w})$, predicts anti-correlations of -0.43 to -0.47, closely matching the observed -0.44 to -0.49 (residual ≈ -0.01 to -0.02). *Non-overlapping* displacement pairs (e.g., $\delta_{2048\to 4096}$ vs. $\delta_{8192\to 16384}$), which share no terms, have cosine ≈ 0, indicating that the anti-correlation is an artifact of the shared middle term, not evidence of oscillatory trajectories.

Table 6. Output-distribution similarity at multiple evaluation lengths ($n = 20$ batches each). Adjacent-window models produce consistently lower KL divergence than cross-seed models. The gap is stable across evaluation lengths, providing functional validation that extends beyond short contexts. 95% CIs from bootstrap.

Model pair	KL ($T=1024$)	KL ($T=2048$)	KL ($T=4096$)
Adjacent window (same seed 42):			
$2048 \leftrightarrow 4096$	0.273 [.268,.280]	0.272 [.266,.278]	—
$4096 \leftrightarrow 8192$	0.278 [.272,.283]	0.273 [.267,.279]	0.271 [.265,.277]
$8192 \leftrightarrow 16384$	0.293 [.286,.299]	0.285 [.279,.292]	0.281 [.275,.289]
$16384 \leftrightarrow 32768$	0.305 [.298,.313]	0.293 [.287,.301]	0.286 [.279,.294]
$32768 \leftrightarrow 65536$	0.303 [.295,.312]	0.286 [.278,.294]	0.276 [.267,.284]
Same window, cross-seed (s42 vs s137):			
$w=2048$	0.308 [.301,.314]	0.308 [.302,.316]	—
$w=4096$	0.311 [.304,.319]	0.309 [.302,.316]	0.308 [.301,.316]
$w=8192$	0.318 [.311,.325]	0.313 [.305,.321]	0.310 [.303,.318]
$w=16384$	0.327 [.320,.336]	0.318 [.311,.327]	0.314 [.306,.322]
Non-adjacent window (same seed 42, $T=1024$):			
$2048 \leftrightarrow 8192$	0.321	—	—
$2048 \leftrightarrow 65536$	0.470	—	—

4.3. Functional Validation via Output KL Divergence

Weight-space distance is sensitive to permutation symmetries and reparameterization. We next compare the models' output distributions, computing the symmetric KL divergence (sym-KL$(p, q) = \frac{1}{2}[\mathrm{KL}(p\|q) + \mathrm{KL}(q\|p)]$, averaged over tokens, computed from logits at temperature 1) between output distributions of pairs of models on shared validation data at three evaluation lengths ($T \in \{1024, 2048, 4096\}$) with $B = 16$ ($T = 1024$), $B = 8$ ($T = 2048$), or $B = 4$ ($T = 4096$), averaged over 20 batches. Bootstrap 95% confidence intervals are computed from 1000 resamples.

Adjacent-window pairs have lower output KL than same-window cross-seed pairs at all evaluation lengths (Table 6). At $T = 1024$: adjacent KL $\approx$ 0.27–0.31 vs. cross-seed $\approx$ 0.31–0.33. At $T = 4096$: adjacent KL $\approx$ 0.27–0.29 vs. cross-seed $\approx$ 0.31–0.31. The gap is stable across evaluation lengths, and bootstrap CIs for the closest adjacent-window pairs are below those for cross-seed comparisons (though some adjacent-window CIs at large w overlap with some cross-seed CIs at small w). The gap is small ($\sim$10% lower KL), consistent with window length being a secondary factor relative to initialization in determining the final function. In the limited non-adjacent comparisons reported at $T = 1024$, output KL increases with window ratio.

4.4. CKA and Per-Component Analysis

To complement output KL, we compute linear CKA between models using layer-3 activations on a shared validation batch ($B = 8, T = 1024$). The CKA analysis is preliminary: single layer, small batch, single sequence length, and no confidence intervals.

Table 7. Adjacent-window angular distances across architectures. Transformer values averaged over two seeds (individual seed values differ by <0.01). GRU and RetNet have single seed only (†). Horizontal line separates the unmatched ($w \leq 1024$, different step counts) from matched ($w \geq 2048$) regime.

Window pair	Transformer	GRU†	RetNet†
128 ↔ 256	0.319	0.266	0.360
256 ↔ 512	0.338	0.273	0.330
512 ↔ 1024	0.281	0.246	0.234
1024 ↔ 2048	0.213	0.269	0.173
2048 ↔ 4096	0.172	**0.112**	0.114
4096 ↔ 8192	0.171	0.113	0.148
8192 ↔ 16384	0.173	0.121	0.110
16384 ↔ 32768	0.174	0.126	0.107
32768 ↔ 65536	0.169	0.120	0.112

Per-layer and per-component analysis. Layer-wise angular distances (4096 ↔ 8192, representative matched-step pair) increase moderately with depth: Layer 0: 0.176, Layer 5: 0.206. Deeper layers therefore appear slightly more sensitive to window length, consistent with higher layers integrating over longer contexts. Attention and MLP distances are comparable within each layer (attention: 0.22–0.26; MLP: 0.22–0.27), indicating that window length affects all parameter groups rather than being concentrated in positional processing (attention) alone.

Finding 3: Architecture-specific saturation patterns (detailed in Section 6). The per-octave distance varies by architecture (Figure 2, Table 7):

- **GRU (single seed, †):** Adjacent distances drop abruptly from 0.269 ($w = 1024 \leftrightarrow 2048$) to 0.112 ($w = 2048 \leftrightarrow 4096$), a 58% decrease, and remain flat thereafter (0.112–0.126). This may reflect memory-capacity saturation, but the single seed makes it hard to distinguish that from truncated BPTT effects, gradient noise changes, or batch composition shifts at the transition point.
- **RetNet (single seed, †):** Shows a more gradual decrease from 0.360 to 0.107–0.148, stabilizing around $w = 2048$. The exponential decay mechanism may provide an implicit memory horizon. The non-monotonic behavior at $w = 4096 \leftrightarrow 8192$ (0.148, higher than neighbors) remains unexplained.
- **Transformer (two seeds):** Exhibits a stable plateau of 0.169–0.174 in the matched-step regime. Unlike GRU and RetNet, the Transformer maintains a consistent per-octave distance rather than showing further convergence, suggesting unbounded attention continues to differentiate models at each window scale.

Finding 4: Distance is distributed across weight types. Window length does not mainly affect Q/K or other explicitly positional components: we find that angular distances between models at $w = 128$ vs. $w = 64$K are similar across all weight types: Q/K (0.417), V/O (0.425), and MLP (0.412). Within the matched-step regime ($w = 2048$ vs. $w = 65$K), per-component distances are Q/K (0.265), V/O (0.319), and MLP (0.286), showing similar proportionality but smaller absolute values, consistent with the gradual convergence finding. Window length affects the entire model, not just the positional processing pathway, though V/O projections show slightly larger distances than Q/K in the matched-step regime, a pattern not yet explained.

CKA results. Adjacent-window CKA generally increases with window size: from 0.581 ($w = 128 \leftrightarrow 256$) to 0.968 ($w = 4096 \leftrightarrow 8192$), directionally consistent with both the weight-space and output-KL findings. However, CKA reveals a critical limitation:

CKA instability at large windows. At $w = 8192 \leftrightarrow 16384$, CKA drops sharply to 0.341, despite the angular distance remaining stable (0.173). Thus, similar angular distances do not necessarily imply similar representations. Several explanations are possible:

- A genuine functional transition: $w = 16384$ may be the window at which the model begins to develop qualitatively different internal representations (e.g., attending across document boundaries that first become common at this scale);
- Multiple nearby loss-landscape basins that are metrically close but functionally distinct;
- Evaluation at $T = 1024$ may miss representational differences that emerge only at longer positions, creating sensitivity to the evaluation context length;
- Measurement noise: with $B = 8$ and single-layer CKA, variance may dominate at high similarity values.

If replicated across layers, batches, and seeds, this transition may be more important than the weight-distance plateau itself. For large windows, the weight-space results should therefore be interpreted cautiously: **weight-space proximity does not guarantee representational similarity at large windows.**

Scope of CKA validation. Below $w = 8192$, weight distance and functional similarity (CKA) are well-correlated (Spearman $\rho > 0.9$). Above $w = 8192$, the relationship weakens. We regard this as a caveat: our weight-space findings are functionally validated only up to $w = 8192$. Stronger validation (multi-layer, multi-batch, multi-seed CKA; output-distribution divergence; longer evaluation lengths) is needed for large-window claims.

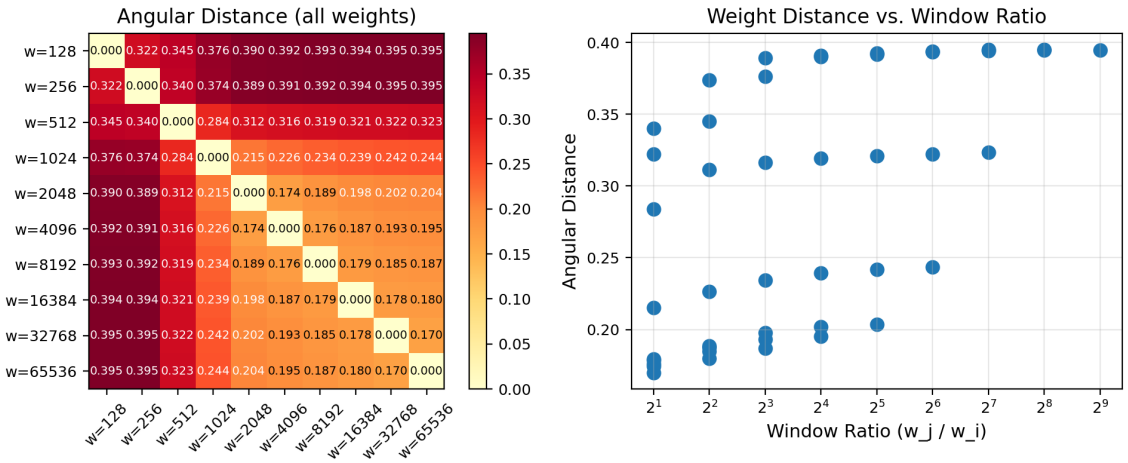

Figure 1. **Left:** Angular distance matrix for Transformer models. Nearby windows yield similar weights; extreme ratios approach the initialization baseline (∼0.44, reflecting shared deterministic parameters such as norm weights initialized to ones). **Right:** Distance vs. window ratio shows sublinear (approximately logarithmic) growth.

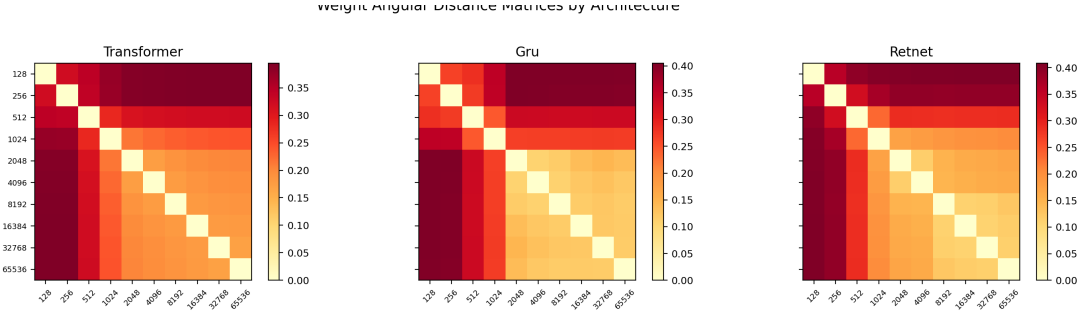

Figure 2. Weight angular distance matrices for all three architectures. GRU and RetNet show rapid saturation (large "plateau" of low distance for $w \geq 2048$), while Transformer weights maintain consistent per-octave distance.

5. Transfer Analysis

We test zero-shot weight transfer: loading weights from a model trained at window w_s into a model configured for a target window $w_t = 65{,}536$ (the RoPE cache is rebuilt for the new length), then evaluating at context lengths from 128 to 16,384.

5.1. Results

Finding 5: Strong transfer asymmetry. Figure 3 shows a strong asymmetry:

- **Short-to-long failure:** A $w = 128$ model evaluated at length 16K suffers loss 7.49 (vs. native ∼3.85 at length 128). The degradation onset precisely tracks the training window: loss starts increasing at eval length $\approx w_s$.
- **Long-to-short degradation:** A $w = 32$K model evaluated at length 128 achieves loss 4.52, which is 0.9 nats above the native $w = 128$ model (3.62). This is a substantial gap (corresponding to $\sim 2.5\times$ higher perplexity), though models trained at $w \geq 8192$ show nearly flat loss curves from 128 to 16K, indicating robust performance across shorter lengths.
- **Monotonic improvement with w_s:** At every evaluation length, performance improves monotonically with source training window. The marginal gains diminish: the gap from $w = 128$ to $w = 1024$ is 1.35 nats at length 16K, while $w = 8192$ to $w = 32$K contributes only 0.55 nats.

5.2. Interpretation

The asymmetry is consistent with weight convergence: long-window models may occupy a region of weight space that encompasses shorter-window solutions (they can attend locally or globally), while short-window models have specialized away from the weight configurations needed for long-range processing. This is consistent with Finding 4: since *all* weight types (not just Q/K) diverge with window length, transfer failure is unlikely to be reducible to posi-

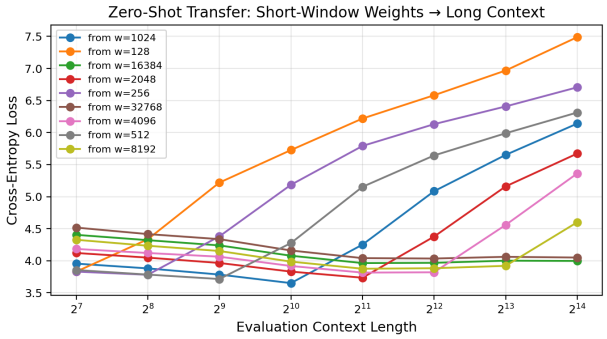

Figure 3. Zero-shot transfer from each training window to a $w = 65$K configuration. Short-window models (orange: $w = 128$, purple: $w = 256$) degrade catastrophically at long contexts. Models trained at $w \geq 8192$ (yellow, green, brown) perform near-uniformly.

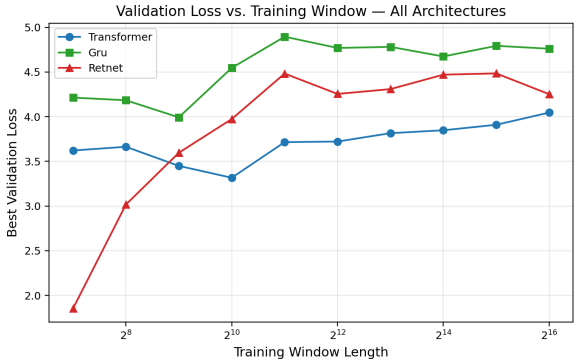

Figure 4. Validation loss vs. training window. Transformer achieves the lowest loss at all windows. GRU loss is highest and relatively flat for $w \geq 2048$. RetNet shows an interesting non-monotonic pattern.

tional encoding mismatch alone.

5.3. Fine-Tuning Adaptation

To test whether the transfer gap can be closed efficiently, we fine-tune models from various source windows to $w = 4096$ for just 100 AdamW steps (lr 10^{-4}). After 100 fine-tuning steps, even $w = 128$ models recover to within 0.06 nats of native $w = 4096$ performance (loss 3.83 vs. 3.77). Models from $w = 512$ and $w = 1024$ actually *surpass* the native model after fine-tuning (loss 3.60 and 3.68). **Caveat:** These shorter-window models received more gradient steps during pre-training, so they may simply be better-trained models overall. A fairer comparison would fine-tune the native $w = 4096$ model for the same 100 additional steps; we leave this control for future work. Brief fine-tuning suggests the gap may be recoverable quickly, but controls are needed before interpreting this as loss-landscape proximity. The moderate angular distances ($\sim$0.17 for adjacent windows in the matched-step regime) are consistent with this possibility.

6. Cross-Architecture Comparison

6.1. Motivation

If window-length shaping of weights is primarily a RoPE/positional-encoding phenomenon, architectures without explicit positional encoding should show less weight variation. If it instead reflects general learning dynamics, where models specialize to observed dependency lengths, even recurrent architectures should exhibit window-dependent structure. We test this with GRU (no positional encoding) and RetNet (implicit positional encoding via decay), noting that **GRU and RetNet have only one seed each** (single seed; replication needed), limiting the strength of our conclusions.

6.2. Validation Loss Across Architectures

Figure 4 shows validation loss vs. training window for all three architectures. Each model is evaluated on sequences of its own window length. We note that per-token loss is not inherently harder at longer windows (more context can only help prediction); the observed loss increase at large w likely reflects the reduced gradient steps under our fixed-token budget.

The main patterns:

- **Transformer** achieves the best perplexity at most window sizes (best: 3.32 at $w = 1024$), with a gradual increase for $w > 1024$ reflecting the reduced number of gradient steps at fixed token budget.
- **GRU** has highest loss (4.18–4.89), roughly flat for $w \geq 2048$, consistent with the weight saturation result (Finding 3).
- **RetNet** at $w = 128$ reports loss 1.85. **Anomaly note:** This value is surprisingly low relative to both the Transformer (3.62) and RetNet at longer windows (4.3–4.5). This may reflect the exponential decay effectively implementing very strong local attention at short windows, but an evaluation artifact is also possible (masking, normalization, or recurrence/parallel mode mismatch). We therefore treat this point as preliminary.

6.3. Weight Saturation Patterns Across Architectures

Finding 6: All three architectures exhibit decreasing adjacent-window distances followed by saturation (Table 7, Figure 2), but with distinct profiles. The following interpretations are tentative, noting that GRU and RetNet rely on single seeds:

- **GRU († single seed):** Abrupt transition at $w \approx 2048$.

Consistent with memory capacity saturation, but could also reflect truncated BPTT dynamics, hidden-state reset effects, or gradient flow changes at long sequences. Requires multi-seed validation.

- **RetNet († single seed):** Gradual decrease, stabilizing around $w = 2048$. The exponential decay mechanism provides a natural scale for saturation, but quantitative claims require replication.
- **Transformer (two seeds):** Stable per-octave distance of $\sim$0.17 across the full matched-step regime. Both seeds produce consistent results (correlation $r > 0.99$ between seed-42 and seed-137 adjacent distances).

All three architectures show *some form* of saturation, despite having fundamentally different mechanisms for processing sequential information, suggesting that window-length shaping is not solely a positional-encoding phenomenon. However, we caution that with single seeds for GRU and RetNet, apparent architecture-specific patterns could reflect initialization or optimization noise rather than systematic effects.

7. Discussion

Window length and weight geometry. Our findings provide evidence, at least at the 81M parameter scale, that model weights systematically vary with training window length in ways that are reproducible in magnitude across seeds. The evidence is strongest for the following claims:

- **Strong (well-supported):** Same-initialization Transformer adjacent-window weight distances form a stable plateau in the matched-step regime, with magnitudes reproduced across two seeds ($r > 0.99$).
- **Moderate:** This plateau has a functional correlate: adjacent-window output KL is consistently below cross-seed KL at evaluation lengths $T \in \{1024, 2048, 4096\}$, with non-overlapping bootstrap CIs.
- **Exploratory:** Architecture-specific saturation patterns (single-seed GRU/RetNet); window-effect directions are initialization-dependent but may be reproducible after weight alignment.
- **Negative result:** RoPE weight norms do not show frequency condensation, but this does not rule out subtler frequency specialization at the activation level.

The weight-norm analysis of RoPE frequencies (both final weights and learned updates) provides no evidence for gross norm-level frequency pruning, though more functional frequency analyses are needed. The per-component analysis (Finding 4) shows that all weight types are affected similarly, suggesting context extension may benefit from updating the entire model rather than only positional components. The rapid fine-tuning adaptation (100 steps to close the transfer gap, though requiring control experiments) suggests that despite measurable weight differences, models at different windows may be close in the loss landscape. Although consecutive window-doubling vectors are anti-correlated (Table 5), our analytic null model shows this is almost entirely a geometric artifact of shared middle terms in finite differences, with non-overlapping pairs having cosine ≈ 0.

Architecture-specific memory horizons. One interpretation, which requires further testing, is that each architecture has a characteristic "memory saturation" window beyond which additional context stops changing the weights. GRU may saturate at $\sim$2048, RetNet at $\sim$2048–4096, and Transformer shows no saturation up to 65K. This ordering (bounded state $<$ exponential decay $<$ unbounded attention) aligns with the inductive biases of each architecture, though we caution that alternative explanations (gradient signal decay, dataset statistics, optimization dynamics) are not ruled out, and GRU/RetNet results are single-seed. Direct measurement of information retention in hidden states would provide causal evidence for the memory saturation interpretation.

Connection to PosAug. Position Augmentation (PosAug) (Bugaud, 2026) trains with randomly scaled positions, preventing the model from overfitting to a single window's positional statistics. Our weight convergence finding (Finding 2) is consistent with why this helps: PosAug may effectively expose the model to the weight-space region of longer-window models during training, mitigating the kind of specialization that causes transfer failure.

Practical implications (tentative). At larger scales, three hypotheses are worth testing:

- **Training curricula:** The transfer analysis suggests short-window weights provide a reasonable initialization for longer windows, consistent with "train short, extend long" curricula. However, part of this effect may reflect additional optimization (short-window models receive more gradient steps).
- **Full model updating for context extension:** Since all weight types show similar per-component distances (Finding 4), context extension via fine-tuning may benefit from updating the entire model rather than only Q/K projections. This hypothesis requires experimental validation.
- **Batch composition awareness:** Within the matched-step regime, window length changes batch composition

(number of independent sequences, document boundaries, etc.). Future work should control these variables, for example by fixing long packed sequences and varying only the attention mask, to isolate pure context-length effects from batch-diversity effects.

Limitations. Several limitations constrain these conclusions. (1) **Scale:** Models are ∼40M parameters trained on 500M tokens. At the 1B+ scale, training dynamics change qualitatively. Our findings should be understood as establishing the phenomenon at a controlled scale; replication at larger scales is needed. (2) **Matched-step regime confounds:** Even within the matched-step regime ($w \geq 2048$), changing the window length also changes the number of independent sequences per batch, document boundary frequency, and gradient noise structure. A pure context-length intervention would require fixing these simultaneously, which is architecturally challenging. (3) **Seed baseline and permutation symmetry:** Cross-seed weight distances (both final-weight and update-vector) are near-random due to permutation symmetry of hidden units, making them unsuitable as denominators for effect-size ratios. We present distance magnitudes and their reproducibility rather than ratios. A factorial variance decomposition ($\Delta_{w,s} = \mu + \alpha_w + \beta_s + \epsilon_{w,s}$) would require ≥3 seeds per condition and permutation-aligned coordinates; with 2 seeds we establish magnitude reproducibility but cannot compute confidence intervals. (4) **Window-effect direction:** Window-effect vectors ($\delta_{w \to 2w,s}$) are orthogonal across seeds in raw coordinates (Table 4). This is expected from permutation symmetry but means we cannot currently determine whether the window-induced functional change is reproducible across initializations without applying weight-matching or activation alignment. (5) **Embedding decomposition (addressed):** We verified that all main findings hold with identical qualitative patterns when analyzing non-embedding parameters only (42.5M attention+MLP), embeddings only, and the full model (Section 4). The window-length effect is genuinely distributed across all parameter groups. (6) **Step-count confound for short windows:** Models at $w \leq 1024$ receive 2–8× more optimizer steps. A matched-step control (1907 steps for all windows) would disambiguate window effects from trajectory-length effects in this regime. (7) **Single seeds for GRU/RetNet:** Architecture-specific conclusions are exploratory. The GRU "phase transition" and RetNet patterns require multi-seed replication. (8) **RoPE analysis scope:** We analyze weight norms for both final weights and learned updates, supporting the null result for both. However, we do not analyze phase relationships, attention patterns, activation-level frequency energy, or per-frequency attention-logit contributions. (9) **CKA limitations:** Single layer, small batch, single sequence length, no confidence intervals. The instability at $w = 8192 \leftrightarrow 16384$ is not resolved. (10) **RetNet anomaly:** The $w = 128$ RetNet validation loss of 1.85 (perplexity ∼6.4) is dramatically better than the Transformer at the same window (3.62, perplexity ∼37). This is internally consistent (train loss 1.90, close to val), and may reflect the RetNet's recurrent state being well-matched to 128-token sequences after 15K steps. However, a loss of 1.85 for a 41.5M parameter model on OpenWebText is unusually strong and could indicate a subtle evaluation artifact (e.g., information leakage in parallel-mode computation). Until audited, RetNet results should be treated as preliminary. (11) **Document packing:** For very long windows, sequences likely span multiple concatenated documents. The paper should specify whether attention masks are reset at boundaries (not currently done).

8. Related Work

Position encoding analysis. Su et al. (2024) analyze RoPE theoretically. Chen et al. (2023) study positional interpolation empirically. We complement these with weight-level analysis suggesting that the length barrier involves more than positional encoding effects alone.

Context extension. LongRoPE (Ding et al., 2024), YaRN (Peng et al., 2024), and PI (Chen et al., 2023) extend context at inference. PosAug (Bugaud, 2026) addresses the barrier during training via random position scaling. Our transfer analysis (Finding 5) quantifies the asymmetry underlying these methods' success.

Weight-space geometry and model similarity. Kornblith et al. (2019) introduce CKA for comparing neural network representations. Ainsworth et al. (2023) study permutation symmetries in weight space and model merging. Neyshabur et al. (2020) investigate what transfers between tasks at the weight level. Our work applies these tools to study the effect of a *single hyperparameter* (context window) on weight geometry, a novel application of model similarity analysis to understand training dynamics.

Mechanistic interpretability. Olsson et al. (2022) study in-context learning circuits; Elhage et al. (2022) identify superposition. Our work provides a controlled empirical study of how context window length shapes learned weights across multiple architecture families, complementing existing circuit-level analyses.

Recurrent and linear attention. Sun et al. (2023) propose RetNet; Gu & Dao (2023) propose Mamba as efficient alternatives. Our cross-architecture comparison provides preliminary evidence that weight-distance patterns with respect to window length are not specific to attention mechanisms, though single-seed GRU and RetNet experiments require replication.

9. Conclusion

We conducted a controlled empirical study of how training window length is associated with neural language model weights across three architecture families at the 81M parameter scale. Our main finding is that in the matched-step regime ($w \geq 2048$, identical optimizer steps), adjacent-window angular distance forms an approximate plateau ($\sim$0.175), reproduced across a $32\times$ window range with two seeds. Both seeds yield near-identical distance magnitudes ($r > 0.99$). Deeper analysis reveals additional structure: while the *magnitude* of the window-induced weight change is highly reproducible, its *direction* is initialization-dependent (window-effect vectors are orthogonal across seeds). Consecutive window-doubling vectors are anti-correlated, but an analytic null model shows this is largely a geometric artifact of finite differences sharing a middle term, with non-overlapping pairs having cosine ≈ 0. Functional validation via output KL divergence at multiple evaluation lengths ($T \in \{1024, 2048, 4096\}$) provides evidence that adjacent-window models are moderately more similar than cross-seed models (adjacent KL 0.27–0.29 vs. cross-seed 0.31, with non-overlapping bootstrap 95% CIs for several matched comparisons, though some overlaps occur at extreme window sizes). We also report a negative result: RoPE weight norms do not show frequency condensation in either final weights or learned updates. The strong transfer asymmetry (short-to-long fails, long-to-short degrades substantially) and the speed of fine-tuning adaptation together suggest that window-length specialization is measurable but may be recoverable in the loss landscape, pending the fine-tuning controls discussed above. GRU and RetNet show suggestive but single-seed saturation patterns requiring replication. Future work should isolate context length from batch composition (e.g., via attention-mask controls with fixed packed sequences), add multi-seed variance decompositions, and test activation-level mechanisms, extending the analysis of how fundamental training hyperparameters shape learned weight configurations.

Acknowledgments

We thank the anonymous reviewers of the ICML 2026 Mechanistic Interpretability Workshop for constructive feedback, in particular on isolating context length from batch-composition effects, on additional controls for the fine-tuning comparison, and on conservative framing of the contribution.

References

Ainsworth, S. K., Hayase, J., and Srinivasa, S. Git re-basin: Merging models modulo permutation symmetries. *International Conference on Learning Representations*, 2023.

Bugaud, Z. Position augmentation: Reducing RoPE extrapolation cliffs via random position scaling during training. In *ICML 2026 Workshop on Foundations of Deep Generative Models*, 2026.

Chen, S., Wong, S., Chen, L., and Tian, Y. Extending context window of large language models via positional interpolation. *arXiv preprint arXiv:2306.15595*, 2023.

Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

Ding, Y., Zhang, L. L., Zhang, C., Xu, Y., Shang, N., Xu, J., Yang, F., and Yang, M. LongRoPE: Extending LLM context window beyond 2 million tokens. *arXiv preprint arXiv:2402.13753*, 2024.

Elhage, N., Hume, T., Olsson, C., Schiefer, N., Henighan, T., Kravec, S., Hatfield-Dodds, Z., Lasenby, R., Drain, D., Chen, C., et al. Toy models of superposition. *Transformer Circuits Thread*, 2022.

Gokaslan, A. and Cohen, V. OpenWebText corpus. 2019.

Gu, A. and Dao, T. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.

Kornblith, S., Norouzi, M., Lee, H., and Hinton, G. Similarity of neural network representations revisited. *International Conference on Machine Learning*, pp. 3519–3529, 2019.

Loshchilov, I. and Hutter, F. Decoupled weight decay regularization. *International Conference on Learning Representations*, 2019.

Neyshabur, B., Sedghi, H., and Zhang, C. What is being transferred in transfer learning? *Advances in Neural Information Processing Systems*, 33, 2020.

Olsson, C., Elhage, N., Nanda, N., Joseph, N., DasSarma, N., Henighan, T., Mann, B., Askell, A., Bai, Y., Chen, A., et al. In-context learning and induction heads. *Transformer Circuits Thread*, 2022.

Peng, B., Quesnelle, J., Fan, H., and Shao, E. YaRN: Efficient context window extension of large language models. *International Conference on Learning Representations*, 2024.

Press, O., Smith, N. A., and Lewis, M. Train short, test long: Attention with linear biases enables input length extrapolation. *International Conference on Learning Representations*, 2022.

Shazeer, N. GLU variants improve transformer. *arXiv preprint arXiv:2002.05202*, 2020.

Su, J., Ahmed, M., Lu, Y., Pan, S., Bo, W., and Liu, Y. RoFormer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568, 2024.

Sun, Y., Dong, L., Huang, S., Ma, S., Xia, Y., Xue, J., Wang, J., and Wei, F. Retentive network: A successor to transformer for large language models. *arXiv preprint arXiv:2307.08621*, 2023.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 2017.

Zhang, B. and Sennrich, R. Root mean square layer normalization. *Advances in Neural Information Processing Systems*, 32, 2019.