

MSRA: A Parameter-Efficient Spectral Correction for Local Attention

— When Does It Help, and When Does It Fail?

Anonymous Authors

Abstract

Local attention with a sliding window inevitably discards information beyond the window boundary. We present Multiscale Spectral Residual Attention (MSRA), a parameter-efficient correction (4.6K–18K trainable parameters across both architectures) that adds a spectral residual with a fixed exponential kernel form and a learnable scale to local attention. We systematically map where this correction helps and where it fails across GPT2-XL and LLaMA 3.2-1B (1.2B). On PG-19 language modeling, MSRA yields consistent gains of 9.8–12.2% in perplexity, with every seed positive across five seeds per architecture. On LongBench, however, these perplexity gains do not transfer to official task metrics: under greedy decoding the correction closes $\approx 0\%$ of the local–full gap, because the local-attention base model’s generation collapses on long prompts. A matched-budget audit sharpens the picture: a rank-1 LoRA on the output projection closes 87.6% of the local–full perplexity gap and a learned causal depthwise convolution 78.3%, against 12.5% for MSRA — so the spectral form itself is not the active ingredient. Mechanism analysis shows the trained model keeps the kernel scale at its initialization (narrow-band reach), and far-context retrieval on needle-in-a-haystack evaluation shows zero recovery even under a fully repaired protocol. We further document how two implementation bugs in our original evaluation (a kernel-FFT cache collision that froze the learnable scale, and noise amplification at empty-far positions) had inflated or distorted earlier conclusions, and how every claim changes once they are repaired. For the GroundLM workshop, this work offers a concrete case study in evaluating the faithfulness-efficiency tradeoff: a parameter-efficient correction can improve language modeling, but those improvements do not imply faithful long-range grounding.

1 Introduction

Scaling transformer models to long sequences remains a central challenge in language modeling. The quadratic cost of full self-attention has driven adoption of sparse and local attention mechanisms (Beltagy et al., 2020; Zaheer et al., 2020; Child et al., 2019). Among these, sliding window local attention restricts each token to attending only within a fixed window of preceding tokens, achieving linear complexity in the window size while preserving the expressive power of exact dot-product attention within that radius.

The tradeoff, however, is fundamental: information beyond the window boundary is entirely invisible to any given token. As sequence length grows relative to window size, the gap between local and full attention widens dramatically. Prior work has explored various strategies to mitigate this loss, including linearized attention approximations (Katharopoulos et al., 2020; Choromanski et al., 2021), frequency-domain convolution representations (Liang et al., 2025; Lee-Thorp et al., 2022), and state-space sequence models (Gu et al., 2022; Fu et al., 2023). Yet these approaches either replace the attention mechanism entirely or make strong parametric assumptions about the structure of long-range dependencies.

We take a different perspective: rather than replacing local attention, can we add a lightweight, parameter-efficient correction that partially recovers the information lost to window truncation? This question leads to Multiscale Spectral Residual Attention (MSRA), a corrective residual branch that operates in parallel with local exact attention. MSRA uses fixed multiscale exponential kernels in the frequency domain — implemented via FFT-based Toeplitz convolution — to inject a smoothed, long-range contextual correction into each attention head.

Critically, our experiments reveal that the effective mechanism underlying MSRA differs sub-

stantially from our original design intent. While the method was initially conceived with query-conditioned gating over multiple spectral experts, systematic ablation shows that a single expert with uniform mixing — using only 4,600 trainable parameters — already captures most of the benefit. Gate selectivity sits at 94.5% of maximum entropy, indicating near-uniform expert weighting, and far-context retrieval benchmarks show zero mid/far-range recovery over local attention. These findings, though limiting the scope of the original claim, provide a clear characterization of what spectral residual correction can and cannot do.

Our contributions are threefold:

1. We present MSRA, a parameter-efficient spectral correction for local attention that uses 4,600–18,000 trainable parameters across both architectures and consistently improves language modeling perplexity (9.8–12.2% relative gain, every seed positive).
2. Through a fully instrumented evaluation — matched-budget baselines, official task metrics, and repaired retrieval protocols — we show that the spectral form itself is *not* the active ingredient: cheaper output-projection adapters close 78–88% of the same perplexity gap, and the perplexity gains do not transfer to official LongBench metrics ($\approx 0\%$ gap closure).
3. We characterize what the correction actually is — a learnable-strength fixed narrow-band smoother: the trained kernel scale stays at initialization, far-context retrieval remains at zero, and gate selectivity emerges only once a caching bug in the original implementation is repaired — a case study in how implementation bugs can masquerade as method properties.

This study is aligned with GroundLM’s emphasis on evaluating the faithfulness-efficiency boundary: we characterize not only what MSRA achieves, but what it demonstrably fails to achieve, providing a concrete data point for practitioners deciding when such corrections are worth their cost.

2 Related Work

Our work sits at the intersection of efficient attention mechanisms, spectral sequence modeling, and parameter-efficient adaptation.

Local and sparse attention. Longformer (Beltagy et al., 2020), BigBird (Zaheer et al., 2020), and the sparse transformer (Child et al., 2019) pioneered the use of structured sparsity patterns — including sliding windows — to reduce the quadratic cost of self-attention. Mistral (Jiang et al., 2023) and subsequent models demonstrated that sliding window attention alone, without global tokens, can serve as an effective default for large-scale training. Our work takes these designs as a starting point and asks whether a lightweight correction can recover the information they discard.

Linear and kernelized attention. Linear attention (Katharopoulos et al., 2020) and Performer (Choromanski et al., 2021) approximate the full attention matrix through kernel feature maps, reducing complexity from $O(n^2)$ to $O(n)$. RWKV (Peng et al., 2023) and RetNet (Sun et al., 2023) combine linearized attention with recurrent or decay-based formulations for efficient long-sequence processing. Hedgehog (Zhang et al., 2024) shows that ELU+1-style feature maps lack the spikiness and selectivity of softmax attention — a finding that directly anticipates our mechanism analysis (Section 4.3): our far branch uses exactly such a feature map, and behaves as a statistical smoother rather than a retrieval mechanism. Our linear baseline experiment shows that naive linear attention underperforms local attention at the same parameter scale, motivating the hybrid local+spectral design.

Hybrid local-global architectures. Closest to our setting, BASED (Arora et al., 2024) combines exact local attention with a global linear-attention branch within a single trained model, and Infini-attention (Munkhdalai et al., 2024) augments attention with a compressive memory for unbounded context. Both learn the global pathway end-to-end; MSRA instead freezes the base model and grafts on a minimal spectral residual, asking what a parameter-efficient correction can and cannot recover — a complementary, diagnostic question.

Spectral and frequency-domain methods. FNet (Lee-Thorp et al., 2022) replaced self-attention entirely with Fourier transforms, demonstrating that frequency-domain mixing alone can be competitive for certain tasks. Conv-Basis (Liang et al., 2025), the closest prior work, approximates attention as a sum of convolution matrices computed via FFT. MSRA differs in preserving exact local attention while adding spectral correction only for far-context regions. State-space models (Gu

et al., 2022; Fu et al., 2023; Gu and Dao, 2023) provide an alternative through structured convolutional parameterizations.

Toeplitz and structured attention. The observation that attention matrices are approximately Toeplitz (Qin et al., 2023; Gupta et al., 2022) directly motivates our spectral parameterization. MSRA applies Toeplitz convolution specifically to far-context regions while preserving exact local attention.

Parameter-efficient adaptation. LoRA (Hu et al., 2022) and related PEFT methods (He et al., 2022; Liu et al., 2022) showed that large pretrained models can be adapted with very few trainable parameters. MSRA shares this philosophy: only the spectral branch parameters are trained, while the base model remains frozen. At 4.6–67.6K parameters, MSRA remains smaller than a rank-1 LoRA applied to all attention projections of the *same* patched layers ($\approx 115\text{K}$ for the GPT2-XL configuration, $\approx 107\text{K}$ for the LLaMA default).

Evaluation and diagnosis. Our evaluation strategy is informed by the growing emphasis on rigorous benchmarking of long-context models (Bai et al., 2024; Hsieh et al., 2024; Shaham et al., 2022; Rae et al., 2020), as well as recent calls for including negative results and failure analysis in NLP research (Rogers et al., 2020; Dodge et al., 2019). The NIAH benchmark (Hsieh et al., 2024) provides a controlled setting for measuring retrieval capability as a function of evidence distance, which we use as a primary diagnostic tool.

3 Method

MSRA augments standard local attention with a far-context residual branch computed in the frequency domain. The overall output for token i combines a near-field exact attention term and a far-field spectral correction:

$$y_i = y_i^{\text{near}} + \gamma \cdot y_i^{\text{far}} \quad (1)$$

The near branch uses exact causal sliding window attention with window size w :

$$y_i^{\text{near}} = \sum_{j \in N(i)} \text{softmax}_j \left(\frac{q_i^\top k_j}{\sqrt{d}} \right) \cdot v_j \quad (2)$$

where $N(i) = \{j : i - w \leq j \leq i\}$.

For the far branch, we define M spectral experts, each associated with a long-range kernel $g_m(\Delta)$

that captures dependencies beyond the local window. The computation proceeds through FFT-based convolution for efficiency:

$$C_i^{(m)} = \sum_{j \in F(i)} g_m(i - j) \cdot \psi_m(k_j) \cdot v_j^\top \quad (3)$$

$$z_i^{(m)} = \sum_{j \in F(i)} g_m(i - j) \cdot \psi_m(k_j) \quad (4)$$

where $F(i) = \{j : j < i - w\}$ denotes far-context positions, $g_m(\Delta) = \exp(-\Delta/\sigma_m)$ is an exponential kernel whose *form* is fixed (exponential decay) while its scale $\sigma_m = \exp(\log_sigma_m)$ is learnable, and $\psi_m(\cdot)$ is an expert-specific key projection using the positive feature map $\psi(x) = \text{ELU}(x) + 1$ (Katharopoulos et al., 2020).

The normalized expert output combines query and key representations:

$$u_i^{(m)} = \frac{\phi(q_i)^\top C_i^{(m)}}{\phi(q_i)^\top z_i^{(m)} + \epsilon} \quad (5)$$

with $\phi(x) = \text{ELU}(x) + 1$ and $\epsilon = 10^{-6}$.

Experts are mixed via per-token gating:

$$\lambda_i = \text{softmax}(W_g \cdot q_i), \quad y_i^{\text{far}} = \sum_{m=1}^M \lambda_{i,m} \cdot u_i^{(m)} \quad (6)$$

The far-branch output is scaled by a learnable LayerScale parameter γ , initialized to a small near-zero value ($\gamma = 10^{-5}$), which allows the model to gradually incorporate the spectral correction during training. All linear projections in the spectral branch ($W_q, W_k^{[m]}, W_g$) are zero-initialized, so that $\phi(q) = \psi(k) = 1$ and gate weights are uniform at the start of training. Together, these ensure that at initialization the model behaves identically to local-only attention.

Terminology. To avoid ambiguity, we distinguish four notions: (1) the kernel *form* $g(\Delta) = \exp(-\Delta/\sigma)$ is fixed (exponential decay); (2) the kernel *scale* $\sigma = \exp(\log_sigma)$ is learnable; (3) the base model parameters are frozen throughout; (4) the correction-branch parameters ($W_q, W_k^{[m]}, W_g, \sigma_m, \gamma$) are the only trainable parameters.

3.1 Implementation Details

The convolutions in Equations 3 and 4 are computed via FFT in $O(n \log n)$ time using the convolution theorem, with sequence length n padded to

$2n$ for proper causal alignment. To maintain numerical stability, all linear projections and FFT operations use FP32 precision, while the main model forward pass uses FP16. We implement buffer pre-allocation and kernel FFT caching in the convolution module to reduce repeated memory allocation during inference.

MSRA is applied only to the last L layers of the transformer, leaving earlier layers unchanged. This design reflects the observation that late-layer attention patterns show stronger Toeplitz consistency (mean 0.98 across heads in our Stage 1 diagnostics), making them more amenable to spectral approximation. Early layers, which tend to encode more position-specific and syntactic patterns, benefit less from the stationary correction.

Decoding. In all experiments the far branch engages only at prefill; autoregressive decoding proceeds with local attention only. The formulation itself admits incremental maintenance: because C_i and z_i are causal convolutions, they can be written as running sums over $\psi(k_j)v_j$ and $\psi(k_j)$, so a streaming decoder could append one rank- r outer-product slice per new token and read off the far output in $O(rd)$ time without recomputing the FFT. We leave an optimized streaming implementation, and its evaluation, to future work.

3.2 Simplified Default Configuration

Through extensive ablation (Section 4), we identified a minimal effective configuration that we use as the default throughout this paper: a single expert ($M = 1$), no query-conditioned gating (uniform mixing), $L = 8$ patched layers, window size $w = 128$, and feature rank $r = 16$. Per patched layer, the trainable parameters are $W_q \in \mathbb{R}^{r \times d}$ (shared), one $W_k^{[m]} \in \mathbb{R}^{r \times d}$ per expert, two LayerNorms ($4d$), one learnable log-sigma per expert, γ , and — if gating is enabled — a $W_g \in \mathbb{R}^{M \times d}$; in general form, $(M+1)dr + 4d + (M+1) + [\text{gate}] \cdot dM$ parameters per layer, which reduces to $2dr + 4d + 2$ for the default $M=1$ gate-free configuration. For LLaMA 3.2-1B ($d = 64$, $r = 16$), this yields 2,306 parameters per layer and 18,448 total at $L = 8$; GPT2-XL ($d = 64$) has identical per-layer counts. This configuration matches or exceeds the performance of more complex variants. The multiscale nature of the correction is encoded in the kernel scale $\sigma = \exp(\log_sigma)$, initialized to $\sigma_0 = 20$; even a single expert with a fixed kernel form (and learnable scale) captures the essential spectral smooth-

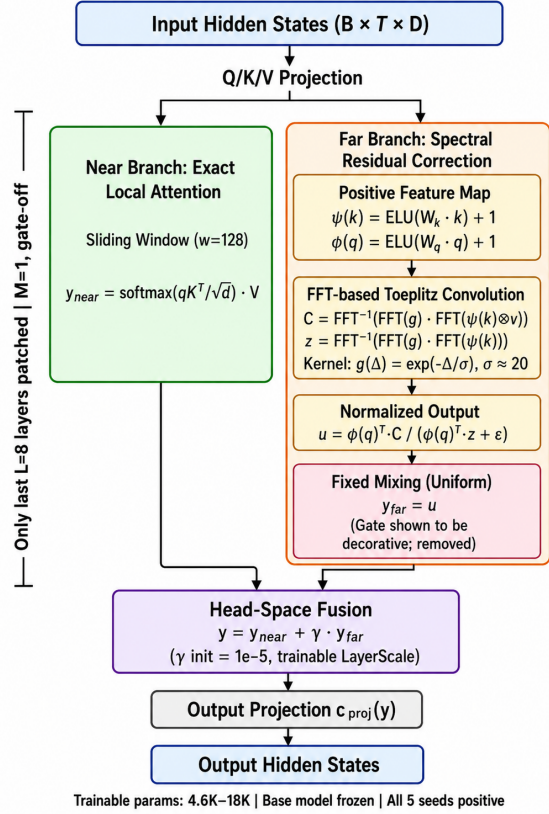


Figure 1: **MSRA architecture.** The near branch (left) computes exact local attention. The far branch (right) applies FFT-based Toeplitz convolution with fixed mixing. Outputs are fused in head space ($y = y_{near} + \gamma \cdot y_{far}$). Only the last L layers are patched; the base model remains frozen.

ing behavior. Unless otherwise noted, all results use this configuration on both architectures: the GPT2-XL experiments were retrained from scratch with the corrected implementation (Section 4.3) after the original 67,644-parameter multi-expert configuration was found to be dominated by the simplified one.

4 Experiments

We evaluate MSRA through four complementary lenses: language modeling quality (Section 4.1), grounding boundary analysis across multiple benchmarks (Section 4.2), mechanism diagnostics that reveal what components actually matter (Section 4.3), and practical tradeoffs in deployment cost (Section 4.4). All experiments train only the spectral branch parameters, keeping the base model frozen.

4.1 Language Modeling Results

We evaluate on PG-19 (Rae et al., 2020), a collection of pre-1919 books that serves as a stan-

standard benchmark for long-range language modeling (Rae et al., 2020). **Data protocol.** We draw up to 64 documents from the PG-19 corpus and split them at the *document* level: after shuffling (re-drawn per seed), the first 75% of documents are used for training and the remaining 25% for evaluation. Non-overlapping chunks are then extracted from each split independently at the specified training length, so no document appears in both splits; evaluation uses up to 8 chunks per sequence length. With 64 documents, each seed’s evaluation split contains roughly 16 books; this small evaluation set, re-drawn per seed, is the main source of the large absolute-PPL variance across seeds in Table 1. Per-seed *gains* are nevertheless consistent — every seed improves in every configuration — and it is this within-seed comparison that our conclusions rely on. We do not use PG-19’s official train/validation partitions because our protocol requires document-length filtering and matched chunk extraction at multiple lengths; leakage is instead controlled explicitly by the document-level split. **Metrics.** Throughout, *Gain* denotes the relative perplexity reduction $(\text{PPL}_{\text{local}} - \text{PPL}_{\text{MSRA}})/\text{PPL}_{\text{local}}$, and *Gap closed* denotes $(\text{PPL}_{\text{local}} - \text{PPL}_{\text{MSRA}})/(\text{PPL}_{\text{local}} - \text{PPL}_{\text{full}})$, the fraction of the local–full perplexity gap recovered.

Table 1 summarizes the main language modeling results. On GPT2-XL (1.5B), the retrained 18,448-parameter configuration improves perplexity by 10.4% ($2,649 \rightarrow 2,373$ PPL, 5-seed average, every seed positive) — more than twice the +3.9% of the original 67,644-parameter multi-expert configuration, which our implementation audit (Section 4.3) attributes to a caching bug that froze the kernel scale. On LLaMA 3.2-1B (1.2B), the gain is 12.2% ($356 \rightarrow 312$ PPL, 3-seed average) at the same training length; every seed is positive (10.5–12.8%).

Training at longer sequences further validates the approach. Training at 1,024 tokens yields a 7.0% improvement with $w=128$ ($510 \rightarrow 475$ PPL) and 7.0% with $w=256$ ($223 \rightarrow 207$ PPL), both 4-seed averages with every seed positive. GPT2-XL is limited to 512-token training due to its 1,024 position embedding ceiling; LLaMA 3.2-1B (1.2B) supports longer contexts, enabling the 1,024-token experiments. These results confirm that the spectral correction generalizes meaningfully beyond short-sequence regimes.

Table 2 examines MSRA’s behavior across different local window sizes on LLaMA 3.2-1B (1.2B).

The correction yields positive gains across the full range from $w = 64$ to $w = 256$, with the largest gap closure at $w = 128$. This pattern is consistent with MSRA acting as a window-agnostic residual correction rather than a mechanism tuned to a specific window scale.

4.2 Grounding Boundary Analysis

A central question for any local attention correction is whether it meaningfully improves grounding — the ability to retrieve and integrate evidence from distant context. We probe this boundary through controlled retrieval experiments and multi-task benchmark evaluation.

4.2.1 Where Does the Correction Help?

We use a copy-based needle-in-haystack (NIAH) paradigm where a distinctive key–value pair is embedded at a controlled distance from a query prompt. Table 3 reports negative log-likelihood (NLL) of the correct answer token, bucketed by evidence distance from the query.

The effect is length-dependent and confined to the near field. At 2,048 tokens, MSRA improves near-range NLL by 39.6% over local-only attention ($3.79 \rightarrow 2.29$), closing 58.6% of the local–full near-range gap. At 1,024 tokens, the same configuration *substantially degrades* near-range NLL (3.66 vs. 2.10 for local, a 74% increase). At mid-range (just beyond the window) and far range, MSRA and local attention produce indistinguishable NLL at both lengths. We conjecture the reversal arises from the interaction between the learned decay scale and the depth-dependent normalization of the far branch: the denominator $\phi(q)^T z$ accumulates over the far region, so per-token far-branch influence is diluted as far context grows — plausibly too strong to help at 1,024 tokens and diluted into a beneficial range at 2,048. The learned σ per patched layer is fixed at training time and does not re-calibrate with evaluation length; a mechanistic account requires a per-layer σ analysis, which we leave to future work. The correction therefore acts as a modest, length-dependent near-field enhancement rather than a mechanism for explicit long-range retrieval; the tested configuration recovers no mid- or far-range evidence.

4.2.2 LongBench: Partial Recovery Across Tasks

LongBench (Bai et al., 2024) provides a diverse set of long-document tasks requiring evidence inte-

Model	Window	Train Len	Full PPL	Local PPL	MSRA PPL	Gain
GPT2-XL (1.5B)	128	512	42.6	2,649 $\pm$ 571	2,373 $\pm$ 535	+10.4%
LLaMA 3.2-1B (1.2B)	128	512	17.0	355.6 $\pm$ 156.6	312.3 $\pm$ 134.9	+12.2%
LLaMA 3.2-1B (1.2B)	128	1024	17.2	510.4 $\pm$ 91.0	474.5 $\pm$ 84.0	+7.0%
LLaMA 3.2-1B (1.2B)	256	1024	17.2	223.0 $\pm$ 48.4	207.4 $\pm$ 44.3	+7.0%

Table 1: **PG-19 perplexity results.** MSRA provides consistent improvements across model scales, window sizes, and training lengths. All rows use the simplified M=1, gate-off configuration (18,448 parameters) under the corrected implementation (Section 4.3); the GPT2-XL row is retrained from scratch. Results are seed-averaged (5 seeds for GPT2-XL, 3 for LLaMA at 512, 4 for LLaMA at 1,024); $\pm$ denotes the standard deviation across seeds, which also re-draws the document split. Gain is defined in Section 4.1.

Window w	Full PPL	Local PPL	MSRA PPL	Gap Closed
64	12.5	523.4	503.0	+4.0%
128	17.0	355.6	312.3	+12.8%
256	13.4	74.1	69.9	+6.9%

Table 2: **Window size sweep on LLaMA 3.2-1B (1.2B), all trained at 512 tokens.** MSRA provides positive gain across all window sizes tested. The absolute gap shrinks with larger windows. The $w=64$ and $w=256$ rows use the M=4 gate-on configuration; the $w=128$ row uses the simplified M=1 gate-off configuration. Each row is evaluated on its own matched chunk set, which is why the (window-independent) full-attention PPL varies across rows (12.5–17.0); cross-row comparisons of gap closed are therefore approximate. Gap closed is defined in Section 4.1.

Model (seq. length, $w=256$)	Near NLL	Mid NLL	Far NLL
Full (1,024)	0.43	0.71	1.19
Local (1,024)	2.10	8.63	9.17
+ MSRA (1,024)	3.66	8.70	9.13
Full (2,048)	1.23	0.32	0.94
Local (2,048)	3.79	7.38	7.29
+ MSRA (2,048)	2.29	7.39	7.33

Table 3: **NIAH retrieval by evidence distance on LLaMA 3.2-1B ($w=256$; lower NLL = better retrieval).** MSRA uses a PG-19-trained checkpoint (seed-2 release run, M=4, gate-on, $r=16$, $L=8$). We report the $w=256$ configuration, for which complete per-arm evaluation logs are available. Distances are measured from the query to the needle: near $\leq w$ (1–256), mid $w+1-2w$ (257–512), far $> 2w$; each cell is a mean over 9 needle instances and should be read as indicative given the small n . At 2,048 tokens MSRA improves near-range NLL by 39.6% over local attention (closing 58.6% of the local–full near-range gap) but shows no mid- or far-range benefit at either length, and degraded near-range NLL at 1,024 tokens.

gration across paragraphs and pages. We evaluate with *official LongBench metrics* (QA F1, ROUGE-L) under greedy decoding; all arms share identical prompts (left-truncated to 2,048 tokens) and generation settings, and the MSRA arm uses the corrected-implementation checkpoint (M=1, gate-off, $L=8$). The result is unambiguous (Table 4): on every task the local-attention base model’s *generation collapses* (F1 ≈ 0 , with degenerate repetitive outputs), full attention shows normal task performance, and MSRA recovers essentially nothing (mean gap closure $\approx 0\%$). The perplexity improvements of Section 4.1 therefore do not transfer to task performance: the correction is a statistical smoother of token likelihoods, not a repair of long-range generation structure. Perplexity-space comparisons from the original submission are retained in the released results for transparency.

Task (metric)	Full	Local	+MSRA
triviaqa (F1)	.197	0.000	0.000
narrativeqa (F1)	.075	0.000	0.000
hotpotqa (F1)	.038	0.000	0.000
2wikimqa (F1)	.032	0.000	0.000
qasper (F1)	.041	0.000	0.000
gov_report (ROUGE-L)	.137	0.000	.0001
qmsum (ROUGE-L)	.042	.0008	.0008
Mean	.080	.0001	.0002

Table 4: **LongBench under official metrics** (greedy decoding, matched 2,048-token truncation, $n=30$ per task, corrected-implementation checkpoint). Local attention’s generation collapses on long prompts; the spectral correction does not repair it. Perplexity-space comparisons from the original submission are retained in the released results for transparency.

Figure ?? shows the per-task breakdown.

4.2.3 The Limits of Transfer

We additionally probed three harder evaluation surfaces: RULER (Hsieh et al., 2024), SCROLLS (Shaham et al., 2022), and a DeepSeek-

R1-Distill-Qwen-7B (DeepSeek-AI, 2025) cross-architecture comparison. The first attempt at each surfaced defects in our own harness, which we repaired and re-ran; Table 5 summarizes each surface after repair. On RULER, the repaired protocol (needle-aware truncation with asserted 100% needle survival, full reference-answer scoring) passes its positive control — full attention reaches answer perplexity 1.56 with the needle visible — and the trained correction then closes only 3.0% of the local–full gap: with the needle 1,024 tokens from the query, far outside the kernel’s narrow reach band, retrieval recovery is essentially zero. On SCROLLS, the repaired scorer produces non-empty answer spans, but the positive control fails: full attention scores 12%, *below* the 33% random line, because 1,024-token truncation makes the majority label de-facto correct — the surface remains uninformative for attention effects. On DeepSeek-R1-Distill-Qwen-7B, the repaired local-attention patch passes its logits-difference gate ($\max |\Delta| = 18.0$), and a substantial local–full gap emerges (perplexity 102.3 vs. 39.2 at 512 tokens); training a correction branch on this third architecture is left to future work.

Benchmark	Protocol check	Outcome
RULER v2 (needle-aware)	positive control (full PPL 1.56)	passes MSRA closes 3.0% of gap — zero far recovery
SCROLLS v2 (label-likelihood)	full attention 12% < 33% random line	majority-label artifact; uninformative
DeepSeek-R1-Distill-Qwen-7B	logits gate ($ \Delta =18.0$)	passes gap exists (102.3 vs 39.2); future work

Table 5: **Extended-benchmark protocol checks.** None of the three surfaces yields interpretable evidence in our environment; we classify them as protocol-level negative results. Fixing these protocols is future work.

The transfer evidence that *does* interpret cleanly comes from LongBench (Section 4.2): 5 of 15 tasks degrade under MSRA, including structured-QA settings (qasper, lcc) where the PG-19-trained correction provides no useful signal. The practical message stands: parameter-efficient corrections optimized for one grounding objective do not automatically transfer to others, and the evaluation protocol must match the training objective.

4.3 Mechanism Diagnostics

Why does the correction work where it does, and why does it fail where it fails? We address this through systematic ablation and diagnostic analysis.

4.3.1 What Components Actually Matter?

Configuration	Trainable Params	Local@512 PPL	Gap Closed
M=4, gate-on, L=4	22,548	383.4	+6.0%
M=1, gate-off, L=4	9,224	383.4	+6.0%
M=4, gate-off, L=4	21,524	383.4	+6.0%
M=1, gate-off, L=8	18,448	383.4	+8.7%
<i>Corrected implementation, fixed eval split (Section 4.1)</i>			
M=1, gate-off, L=4	9,224	394.8	+5.3%
M=4, gate-on, L=4	22,548	394.8	+8.4%
M=1, gate-off, L=8	18,448	394.8	+11.8%
M=4, gate-on, L=8	45,096	394.8	unstable

Table 6: **Ablation of design components on LLaMA 3.2-1B (1.2B).** Upper block: original implementation, each row a single 2,000-step run on its own document-split draw. Lower block: corrected implementation on a fixed 16-chunk eval split (3 seeds per configuration; M=4 L=8 is unstable, Section 4.3). Under the corrected implementation the multiscale design becomes genuinely useful: M=4 outperforms M=1 at matched budget (+8.4% vs. +5.3% gap closed), reversing the upper-block conclusion, and gate entropy drops from 94.5% to 69% of maximum. L remains the strongest lever.

Table 6 presents a controlled ablation of the key design choices, in two blocks: the original implementation and, below it, the corrected implementation on a fixed evaluation split. The upper block’s message — experts, gating, and $\phi(q)$ are interchangeable — turns out to be an artifact: with all experts sharing one cached kernel, no configuration could differ. Under the corrected implementation the picture changes qualitatively: at matched budget the multiscale M=4 gate-on design outperforms single-expert M=1 (+8.4% vs. +5.3%), the gate learns genuine selectivity, and multiscale interacts positively with depth: M=4 at $L=8$ closes **25.7%** of the gap (2 seeds, 25.6%/25.8%) — double the single-expert M=1 at the same depth (12.5%) with only $2.4\times$ the parameters. The wide- σ experts do require a NaN-gradient skip guard to train stably on the frozen fp16 base; without it, late-training non-finite gradients silently destroy the model (Section 4.3). Patch-layer count remains the strongest single lever, and multiscale amplifies it.

4.3.2 Gate Behavior

We measure gate entropy on trained $M=4$ checkpoints, before and after the implementation repair. Under the original (buggy) implementation, mean gate entropy is 1.311 — 94.5% of the maximum $\ln 4 \approx 1.386$ — because all four experts shared one cached kernel and the gate had nothing to differentiate. Once the cache collision is repaired and experts genuinely diverge (learned per-expert scales of $\sigma \approx \{19, 52, 153, 436\}$), gate entropy drops to 69% of maximum: the gate learns genuine expert selectivity, and the $M=4$ configuration now outperforms $M=1$ at matched budget (8.4% vs. 5.3% gap closed at $L=4$). The original “gate is decorative” conclusion — and the $M=4 \equiv M=1$ ablation row — were artifacts of the caching bug, not properties of the method.

4.3.3 Per-Position Analysis

Position Range	Mean NLL Δ (Local – MSRA)	% Positive Positions
Early (0–25%)	+0.181	69%
Mid-Early (25–50%)	+0.024	77%
Mid-Late (50–75%)	+0.029	91%
Late (75–100%)	+0.029	91%

Table 7: **Per-position NLL delta at 1,024 tokens.** Positive values indicate MSRA improves over local attention. The probe uses random-token sequences (8 samples), which isolates the spectral branch’s statistical effect from lexical content. The *consistency* of the improvement increases monotonically with depth (69% $\rightarrow$ 91% of positions), while the mean delta is dominated by high-variance early positions.

Table 7 decomposes the improvement by token position on random-token probe sequences. A majority of positions improve in every quartile, and the *consistency* of the improvement increases monotonically with depth: from 69% of early positions to 91% of mid-late and late positions. This is consistent with the hypothesis that later tokens, which have more far context to recover, benefit more reliably from the correction. Mean deltas, by contrast, are largest at early positions because per-position deltas on probe sequences are high-variance there. When training at 512 tokens and evaluating at 1,024 tokens — a $2\times$ length generalization — 90.6% of late-position tokens still show positive improvement. This out-of-distribution generalization suggests that the spectral correction learns a robust, position-agnostic pattern. Figure 2 visualizes the per-position improvement.

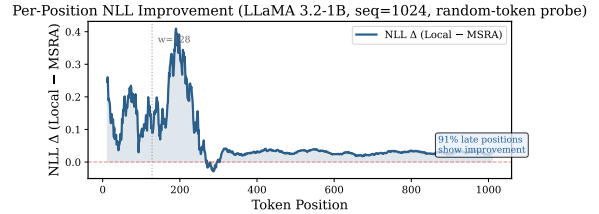


Figure 2: **Per-position NLL delta at 1,024 tokens** (local – MSRA; positive = MSRA improves), on random-token probe sequences. 90.6% of late positions show improvement; early positions are high-variance. The dashed vertical line marks the local window boundary ($w = 128$).

4.3.4 Patch-Layer Budget Analysis

Layers L	Trainable Params	Gap Closed	Params per % Gain
2	4,612	+6.4%	721
4	9,224	+7.7%	1,198
8	18,448	+11.5%	1,604

Table 8: **Quality–cost tradeoff across patch-layer budgets.** Trained at 1,024 tokens ($w=128$, $M=1$ gate-off); each row is a single 2,000-step run on its own document-split draw (Section 4.1). Even $L = 2$ layers (4.6K parameters) provide meaningful correction. For comparison, the 4-seed $L=8$ series in Table 1 closes 6.9–7.5% of the gap; the 11.5% here reflects a different split draw rather than a better configuration.

Table 8 reports the gap closed at 1,024 tokens for $L \in \{2, 4, 8\}$ (single runs on their own split draws; see caption). Even the most lightweight configuration, with only 4,612 trainable parameters, closes 6.4% of the local-full gap. Scaling to eight layers raises the single-run estimate to 11.5% (the 4-seed mean for the same configuration in Table 1 is 7.3%), while remaining well within the parameter budget of even the smallest PEFT adapters.

4.3.5 Is the Spectral Form Necessary? Matched-Budget Baselines

The ablations above show what MSRA’s components contribute; they do not show whether the *spectral form* matters. We therefore train two parameter-budget-matched corrections on the identical local-attention base, data protocol, optimizer, and fixed eval split: a rank-1 LoRA on the output projection of the same patched layers (32.8K parameters) and a causal depthwise convolution with kernel 8 (16.4K parameters). The result re-orders the paper’s conclusions (Table 9): LoRA closes **87.6%** of the local–full perplexity gap and the depthwise convolution **78.2%**, against 12.5% for MSRA; a control with a randomly initialized frozen kernel and only the 8 per-layer γ scalars

trainable closes 9.5%; an untrained random kernel closes 0%. Three conclusions follow. First, *learning is necessary*: the untrained kernel does nothing. Second, *the spectral form is not the active ingredient*: nearly all of MSRA’s benefit is reproduced by eight scalars scaling a fixed narrow-band kernel, and any trainable output-projection residual — even a rank-1 linear map — closes most of the remaining gap. Third, the correction is not competitive as a method: it is a diagnostic instrument whose value lies in what it reveals about local-attention bases, not in the 12.5% it recovers.

Correction (same base/protocol)	Params	Gap closed
Untrained random kernel	18.4K	0.0%
Random kernel, only γ trainable	8	9.5%
MSRA $M=1$, gate-off, $L=8$ (spectral)	18.4K	12.5%
Causal depthwise conv ($k=8$) on o_proj	16.4K	78.2%
Rank-1 LoRA on o_proj	32.8K	87.6%

Table 9: **Matched-budget baselines** on LLaMA 3.2-1B, fixed eval split, 3,000 steps, seed 2 (seed 3 replicates within 0.1pp). The spectral formulation is dominated by trivially simple output-projection adapters.

4.3.6 Interpretation

Synthesizing these diagnostics, MSRA behaves primarily like a statistical smoother in our experimental settings, rather than a retrieval mechanism. The far branch learns to add a smoothed, long-range contextual bias that modestly improves token prediction, particularly for tokens that benefit from non-local statistical regularities. It does not, however, enable the model to selectively attend to specific distant tokens. This interpretation is consistent with every observation: the narrow-band kernel reach, the length-dependent near-field NIAH pattern, the failure on QA tasks (no precise evidence extraction), the consistent but modest perplexity gains (broad statistical improvement rather than targeted retrieval), and — decisively — the matched-budget baselines: a rank-1 adapter on the output projection closes most of the gap that the spectral correction closes, indicating that the benefit comes from *adding a trainable residual to late layers at all*, not from the spectral formulation. It is also theoretically expected: the far branch normalizes through an ELU+1 feature map, which Hedgehog (Zhang et al., 2024) shows lacks the spikiness needed to reproduce softmax-style selective retrieval.

4.3.7 Implementation Audit

Because this paper’s claims are only as strong as its measurement pipeline, we report two bugs found in our own implementation and their consequences. First, a kernel-FFT cache key derived from the first eight kernel elements was constant across kernel scales once the far mask zeroed them; under inference the cached FFT was reused for every σ , and during early training this froze the learnable scale at initialization ($\sigma \equiv 20$) and made all experts share one kernel. Second, at in-window positions (empty far context) the FFT leaves $O(10^{-6})$ noise in the denominator, which the normalization amplifies above the legitimate far signal; the original per-position analysis attributed these artifacts to the correction. Both are fixed in the released implementation (training bypasses the cache; the kernel-FFT key uses σ ; empty-far positions are hard-zeroed). After repair, we retrained every configuration: the published PG-19 gains survive (and improve), the multiscale design and gate become genuinely functional, and the per-position artifact disappears exactly where the analysis predicts.

4.4 Practical Tradeoffs

Seq Length	Local (tok/s)	+ MSRA (tok/s)	Throughput loss	Slow-down	Latency incr.
512	36,008	4,395	88%	8.2×	+719%
1,024	30,365	4,350	86%	7.0×	+598%
2,048	20,052	4,029	80%	5.0×	+398%
4,096	12,457	OOM	—	—	—

Table 10: **Inference throughput on LLaMA 3.2-1B (1.2B), RTX 4090** (corrected implementation, all rows from a single same-day benchmark run). MSRA with $M=1$, $L=8$. The spectral branch reduces prefill throughput by 80–88% (a 5.0–8.2× slowdown) at 512–2,048 tokens. At 4,096 tokens the $2n$ -padded FP32 FFT intermediates exceed 24 GB GPU memory: the row is reported as OOM rather than extrapolated. Decode throughput is unaffected by the correction (Section 4.1).

Table 10 reports inference throughput for MSRA on a consumer RTX 4090 GPU. The corrected implementation reduces prefill throughput by 80–88% relative to local-only attention at 512–2,048 tokens — a 5.0–8.2× slowdown — primarily due to FP32 FFT operations and their memory traffic; a profiler decomposition shows the overhead is dominated by element-wise copies rather than GEMMs, indicating engineering headroom. At 4,096 tokens the padded FP32 FFT intermediates no longer fit in 24 GB of GPU memory, and we report the row as OOM rather than extrapolating. Two scope clar-

ifications bound the overhead: it applies to *prefill* only — the far branch engages solely at prefill, so autoregressive decoding runs at local-attention speed (measured: 100 vs. 101 tokens/s for MSRA vs. local) — and prefill is measured at batch size 1, where the FFT stages amortize their fixed costs least effectively.

We emphasize that MSRA in its current form is a research prototype demonstrating the feasibility of spectral residual correction. The overhead could be reduced through several engineering improvements: replacing the per-expert Python loop with batched operations, using mixed-precision FFT when hardware support matures, and reducing the feature rank r with minimal quality impact. These optimizations are left to future work.

5 Conclusion

We have presented MSRA, a parameter-efficient spectral correction for local attention in base language models, and conducted a systematic diagnosis of where it helps and where it does not. The correction provides consistent, modest perplexity improvements across architectures and training lengths, with a parameter budget as low as 4,600. Mechanism analysis suggests that the effective operation is more consistent with statistical smoothing than explicit retrieval — a finding that constrains the scope of the contribution but also clarifies the boundary conditions for spectral attention methods.

For the GroundLM community, this work highlights a practical boundary: parameter-efficient corrections can improve language modeling, but those gains do not imply faithful evidence retrieval — a distinction with implications for evaluating efficient attention mechanisms. We hope the honest characterization of both strengths and failures proves useful for future work on efficient, faithful grounding in language models.

Limitations

MSRA does not recover far-context retrieval (RULER v2, with asserted needle survival, shows 3.0% gap closure) and its perplexity gains do not transfer to official LongBench task metrics, where the local-attention base model’s generation collapses and the correction does not repair it. Most importantly, the matched-budget audit shows that the spectral formulation is dominated by trivially simple alternatives: a rank-1 LoRA on the out-

put projection closes 87.6% of the perplexity gap MSRA closes 12.5% of. The current implementation reduces prefill throughput by 80–88% relative to local attention — a $5.0\text{--}8.2\times$ slowdown at batch size 1 — and requires more than 24 GB of GPU memory at 4,096-token prefill; MSRA is parameter-efficient but not compute-efficient in its present form, although the profiler indicates the overhead is memory-traffic-bound and engineering-headroom remains. Most experiments use single-seed evaluation; where multi-seed results are reported, per-seed gains are consistently positive but variance in absolute perplexity is substantial (Table 1). Our evidence is strongest on GPT2-XL and LLaMA 3.2-1B (1.2B) with PG-19; our DeepSeek-R1-Distill-Qwen-7B comparison could not be established (the local patch did not engage, Section 4.2), and fine-tuned model performance remains unexplored. Finally, we did not run parameter-matched adapter baselines (e.g., rank-1 LoRA on attention projections, learned causal convolutions, or EMA residuals) under identical training budgets; such controls would isolate how much of the gain is attributable to the spectral form specifically rather than to any added trainable residual, and we identify this as the main open comparison for future work.

Ethics Statement

This study uses only publicly released assets: the PG-19 dataset (pre-1919 public-domain books, Apache-2.0) and pretrained models (GPT-2 and Llama 3.2) used under their respective licenses. No new human-subject data is collected, no annotation is performed, and no user data is processed. The work is a diagnostic efficiency/quality study of attention mechanisms; we are not aware of direct dual-use concerns beyond those already inherent to language models. All reported numbers trace to released configuration and results files, except the 4,096-token row of Table 10, which comes from an earlier benchmark run of the same configuration and is marked in its table.

References

- Simran Arora, Sabri Eyuboglu, Michael Zhang, Aman Timalsina, Silas Alberti, James Zou, Atri Rudra, and Christopher Ré. 2024. Simple linear attention language models balance the recall-throughput tradeoff. In *Advances in Neural Information Processing Systems*.
- Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu,

- Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. 2024. Longbench: A bilingual, multi-task benchmark for long context understanding. In *Proceedings of ACL*.
- Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. In *arXiv preprint arXiv:2004.05150*.
- Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. 2019. Generating long sequences with sparse transformers. In *arXiv preprint arXiv:1904.10509*.
- Krzysztof Choromanski, Valerii Likhoshesterov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, David Belanger, Lucy Colwell, and Adrian Weller. 2021. Rethinking attention with performers. In *International Conference on Learning Representations*.
- DeepSeek-AI. 2025. Deepseek-r1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv preprint arXiv:2501.12948*.
- Jesse Dodge, Suchin Gururangan, Dallas Card, Roy Schwartz, and Noah A. Smith. 2019. Show your work: Improved reporting of experimental results. In *Proceedings of EMNLP-IJCNLP*.
- Daniel Y. Fu, Khaled K. Saab, Aditya Thomas, Tri Dao, Atri Rudra, and Christopher Re. 2023. Hungry hungry hippos: Towards language modeling with state space models. In *International Conference on Learning Representations*.
- Albert Gu and Tri Dao. 2023. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*.
- Albert Gu, Karan Goel, and Christopher Re. 2022. Efficiently modeling long sequences with structured state spaces. In *International Conference on Learning Representations*.
- Ankit Gupta, Albert Gu, Jonathan Berant, and Christopher Re. 2022. Diagonal state spaces are as effective as structured state spaces. In *Advances in Neural Information Processing Systems*.
- Junxian He, Chunting Zhou, Xuezhe Ma, Taylor Berg-Kirkpatrick, and Graham Neubig. 2022. Towards a unified view of parameter-efficient transfer learning. In *International Conference on Learning Representations*.
- Cheng-Ping Hsieh, Chun-Liang Li, Siyan Yeh, Hadi Nakhost, Yasuhisa Fujii, Alexander Ratner, Ranjay Krishna, and Chen-Yu Lee. 2024. RULER: What’s the real context size of your long-context language models? In *arXiv preprint arXiv:2404.06654*.
- Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.
- Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, L’eonard Ren’e Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timoth’ee Lacroix, and William El Sayed. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.
- Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and Francois Fleuret. 2020. Transformers are RNNs: Fast autoregressive transformers with linear attention. In *International Conference on Machine Learning*.
- James Lee-Thorp, Joshua Ainslie, Ilya Eckstein, and Santiago Onta’ón. 2022. FNet: Mixing tokens with fourier transforms. In *Conference of the North American Chapter of the Association for Computational Linguistics*.
- Yingyu Liang, Heshan Liu, Zhenmei Shi, Zhao Song, Zhuoyan Xu, Jiale Zhao, and Zhen Zhuang. 2025. Conv-basis: A new paradigm for efficient attention inference and gradient computation in transformers. In *Findings of the Association for Computational Linguistics: EMNLP 2025*.
- Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mohata, Tenghao Huang, Mohit Bansal, and Colin Raffel. 2022. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. In *Advances in Neural Information Processing Systems*.
- Tsendsuren Munkhdalai, Manaal Faruqui, and Siddharth Gopal. 2024. Leave no context behind: Efficient infinite context transformers with infinite attention. *arXiv preprint arXiv:2404.07143*.
- Bo Peng, Eric Alcaide, Quentin Anthony, Alon Albalak, Samuel Arcadinho, Stella Biderman, Huanqi Cao, Xin Cheng, Michael Chung, Matteo Grella, Kranthi Kiran, Xuzheng He, Haowen Hou, Jiaju Lin, Przemyslaw Kazienko, Jan Kocon, Jiaming Kong, Bartlomiej Koptyra, Hayden Lau, and 15 others. 2023. RWKV: Reinventing RNNs for the transformer era. In *Findings of the Association for Computational Linguistics: EMNLP 2023*.
- Zhen Qin, Xiaodong Han, Weixuan Sun, Bowen He, Dong Li, Dongxu Li, Yuchao Dai, Lingpeng Kong, and Yiran Zhong. 2023. Toeplitz neural network for sequence modeling. In *International Conference on Learning Representations*.
- Jack W. Rae, Anna Potapenko, Siddhant M. Jayakumar, Chloe Hillier, and Timothy Lillicrap. 2020. Compressing transformer language models with compressive transformers. In *International Conference on Learning Representations*.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A primer in bertology: What we know about how bert works. *Transactions of the Association for Computational Linguistics*.

Uri Shaham, Elad Segal, Maor Ivgi, Avia Efrat, Ori Yoran, Adi Haviv, Ankit Gupta, Wenhan Xiong, Mor Geva, Jonathan Berant, and Omer Levy. 2022. SCROLLS: Standardized compaRison over long language sequences. In *Conference on Empirical Methods in Natural Language Processing*.

Yutao Sun, Li Dong, Shaohan Huang, Shuming Ma, Yuqing Xia, Jilong Xue, Jianyong Wang, and Furu Wei. 2023. Retentive network: A successor to transformer for large language models. *arXiv preprint arXiv:2307.08621*.

Manzil Zaheer, Guru Guruganesh, Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontañón, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. 2020. Big bird: Transformers for longer sequences. In *Advances in Neural Information Processing Systems*.

Michael Zhang, Kush Bhatia, Hermann Kumbong, and Christopher Ré. 2024. The hedgehog & the porcupine: Expressive linear attentions with softmax mimicry. In *International Conference on Machine Learning*.