The FFT Strikes Again: A Plug and Play Efficient Alternative to Self-Attention

Jacob Fein-Ashley
University of Southern California
feinashl@usc.edu

Neelesh Gupta
University of Southern California
neeleshg@usc.edu

Rajgopal Kannan
DEVCOM Army Research Office
rajgopal.kannan.civ@army.mil

Viktor Prasanna
University of Southern California
prasanna@usc.edu

Abstract

The quadratic cost of self-attention makes context length the chief bottleneck in Transformer inference. We introduce **SPECTRE**, a drop-in frequency-domain mixer whose per-layer cost scales only as $\mathcal{O}(L \log L)$. SPECTRE projects tokens with a real FFT, applies a learned *diagonal* gate, inverts the transform, and—optionally—adds a lightweight wavelet refinement for local detail. The rest of the model is untouched, so fine-tuning just the new weights suffices. On PG-19 and ImageNet-1k, SPECTRE matches or exceeds quadratic attention while running up to 7× faster than FlashAttention-2 and enabling 32k-token inference on a single GPU. It replaces the quadratic wall with a logarithmic ramp for long-range reasoning.

1 Introduction

Long contexts unlock stronger reasoning. From multi-turn dialogue and book-length summarization to high-resolution vision, many modern tasks demand that Transformers attend over tens of thousands of tokens. Yet the *quadratic* $\mathcal{O}(n^2 d)$ cost of self-attention turns the context itself into the primary inference bottleneck, straining both latency and memory on commodity hardware.

Can we keep global context without paying a quadratic bill? A rich line of work accelerates attention via sparse patterns, kernel approximations, or low-rank structure, but often sacrifices exactness, requires non-standard optimization, or fails to support streaming generation. In contrast, the frequency domain offers an orthogonal route: the Fourier transform *diagonalizes* circular convolutions, converting global mixing into cheap, element-wise products. Unfortunately, prior spectral mixers either rely on fixed filters or must recompute an FFT at every time step—blunting their theoretical advantage.

We answer this challenge with *SPECTRE*, a *drop-in* replacement for self-attention that (i) projects tokens onto an orthonormal Fourier basis, (ii) applies content-adaptive diagonal (and optional low-rank) gates, and (iii) returns to token space via an inverse transform—achieving $\mathcal{O}(n \log n)$ complexity without altering the surrounding architecture. A novel **Prefix–FFT cache** enables streaming decoding analogous to the standard KV-cache, while a switchable **Wavelet Refinement Module** restores the local detail often lost in purely spectral methods.

Preprint. Under review.

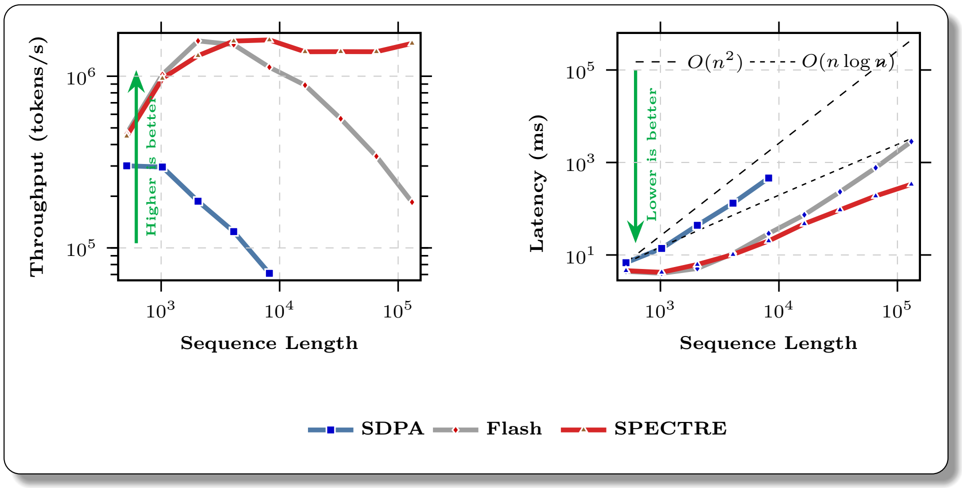

Figure 1: **Inference scaling of a `Llama-3.2-1B` model equipped with three different attention kernels.** We fine-tune an identical backbone with *(i)* standard softmax-dot-product attention (**SDPA**, blue), *(ii)* **FlashAttention-2** [Dao et al., 2023] (grey), and *(iii)* the proposed **SPECTRE** mixer (red). After training, we measure *tokens-per-second throughput* (left) and *single-batch latency* (right) on an NVIDIA A100-80 GB for sequence lengths $L \in \{512,\ 1\text{k},\ 4\text{k},\ 8\text{k},\ 32\text{k},\ 128\text{k}\}$. Dashed black lines show the ideal $\mathcal{O}(n^2)$ and $\mathcal{O}(n \log n)$ slopes. Higher throughput and lower latency are better (green arrows). SPECTRE retains the accuracy of the backbone yet delivers near-$\mathcal{O}(n \log n)$ runtime—remaining flat up to 32k tokens and sustaining a 7× speed-up over FlashAttention-2 at the extreme 128k-token setting.

Contributions.

- **We propose** *SPECTRE*, a frequency-domain token mixer whose per-layer cost scales as $\mathcal{O}(n \log n)$ while slotting directly into existing Transformer checkpoints.
- **We introduce** *content-adaptive spectral gating*, combining learned diagonal and Toeplitz updates to match the expressivity of quadratic attention.
- **We design** the *Prefix–FFT cache*, the first FFT-based key–value cache that supports efficient autoregressive decoding within a fixed memory budget.
- **We add** a lightweight *Wavelet Refinement Module* that reinstates fine local structure for $< 2\%$ extra compute.
- **We demonstrate** up to 7× faster inference than FlashAttention-2 at 32k tokens, while matching or surpassing accuracy on PG-19 language modelling and ImageNet-1k classification.

By turning the quadratic wall into a *logarithmic ramp*, SPECTRE brings efficient, long-range reasoning within reach of everyday computation budgets.

2 Background

Why look beyond quadratic attention? Multi-head self-attention gives Transformers their global receptive field, but its $\mathcal{O}(n^2 d)$ time–memory footprint (n tokens, d channels) quickly overwhelms GPUs and edge accelerators [Vaswani et al., 2017, Beltagy et al., 2020]. A host of linear-time surrogates exist—sparse kernels, low-rank factors, state-space models—yet most trade exactness for speed or break autoregressive caching.

The spectral shortcut. The discrete Fourier transform (DFT) furnishes an *orthonormal spectral basis* that *diagonalizes* any circulant operator: $F_n C F_n^*$ is diagonal for every circulant matrix C [Oppenheim and Schafer, 1999]. Consequently, global convolution becomes an element-wise multiplication in the frequency domain, slashing complexity to $\mathcal{O}(nd \log n)$ once an FFT is available.

Fast Fourier transform and its real cousin. The Cooley–Tukey FFT reduces a length-n DFT from $\mathcal{O}(n^2)$ to $\mathcal{O}(n \log n)$ [Cooley and Tukey, 1965]; its split-radix variant is near optimal [Heideman et al., 1984]. For *real* inputs, Hermitian symmetry implies that only $(\lfloor n/2 \rfloor + 1)$ complex coefficients are unique. Real FFT (**RFFT**) kernels exploit this fact, halving memory and delivering ∼1.8× higher throughput on modern GPUs [Frigo and Johnson, 2005]—the main reason SPECTRE chooses the RFFT.

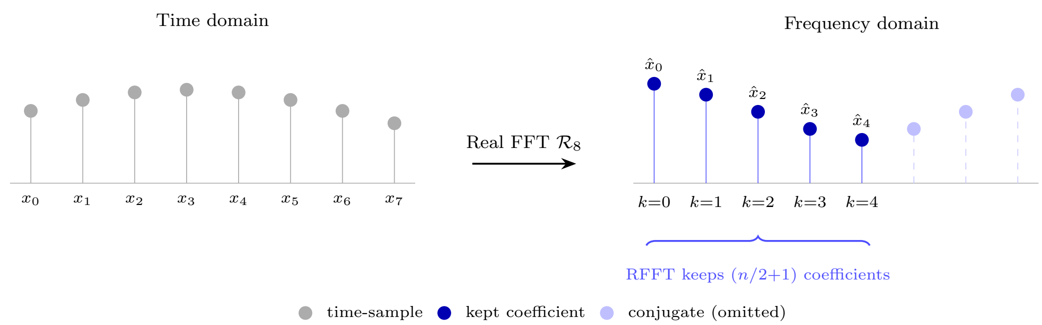

Figure 2: Intuition behind the real FFT. An 8-sample real sequence (left) is mapped, via $\mathcal{R}_8$, to the frequency domain (right). Hermitian symmetry means that the shaded half of the spectrum is redundant; the RFFT therefore stores only $(n/2{+}1)$ coefficients, cutting memory and compute in half.

Spectral token mixers. FNet replaced attention with a fixed global FFT [Lee-Thorp et al., 2021], proving the concept but losing content adaptivity. Follow-ups added learned complex gates or low-rank updates [Lee et al., 2021] yet still recomputed an FFT each step. SPECTRE advances the line by (i) learning *content-adaptive* diagonal and Toeplitz gates *per token*, and (ii) introducing a streaming *Prefix–FFT cache* that preserves frequency-domain efficiency during decoding.

Local detail via multi-resolution. Fourier bases are global; sharp discontinuities smear across frequencies. Wavelets supply localised, orthogonal atoms with logarithmic frequency tiling [Mallat, 1989]. SPECTRE therefore adds an optional *Wavelet Refinement Module* (WRM) that restores fine structure at $\mathcal{O}(nd)$ cost when enabled.

Prefix–FFT *KV*-caching. Autoregressive decoders typically keep a *key–value (KV) cache* so that each new token only attends to its predecessors [Brown et al., 2020]. In SPECTRE, the cache lives in the *frequency* domain: the non-redundant RFFT coefficients of the value stream are stored in a **Prefix–FFT cache**. Thanks to a running-FFT update with eviction, adding one token costs only $\mathcal{O}\big((N_{\max}/2)d\big)$ instead of recomputing an $\mathcal{O}(n \log n)$ transform. The KV footprint, therefore, matches that of quadratic attention while keeping log-linear arithmetic.

Persistent memory bank. Some information—user profile fields, system instructions, stable task prompts—should survive the sliding window entirely. SPECTRE appends a small, fixed set of *persistent memory vectors* $\mathbf{M} \in \mathbb{R}^{N_{\text{mem}} \times d}$ to every sequence. Their RFFT, $\widehat{\mathbf{M}} \in \mathbb{C}^{(\frac{N_{\text{mem}}}{2}+1) \times d}$, is computed *once* per session and concatenated with the prompt's coefficients during pre-fill. Because $\widehat{\mathbf{M}}$ is static, per-step latency is unchanged, and the extra memory cost is merely $\mathcal{O}(N_{\text{mem}} d)$ with $N_{\text{mem}} \ll N_{\max}$ (e.g. 16–64).

Take-away. Log-linear spectral mixing, constant-time Prefix–FFT caching, and a lightweight persistent memory bank, together with SPECTER, equip SPECTRE with global context, streaming generation, and long-term recall—all at a fraction of the compute budget demanded by quadratic attention.

3 Method

We introduce the *Spectral Projection and Content-adaptive Transformer Engine* (**SPECTRE**), a frequency-domain alternative to multi-head self-attention. SPECTRE preserves the Transformer's global receptive field while reducing both runtime and memory usage to $\mathcal{O}(n\,d\log n)$, where n is the sequence length and d is the (per-head) embedding dimension. The SPECTRE layer operates in three main steps:

(i) project tokens onto an orthonormal spectral basis,
(ii) apply content-adaptive diagonal (or optional low-rank) gating in that basis, and
(iii) perform an inverse transform back to token space.

3.1 Preliminaries

Let $X = [x_1, \ldots, x_n] \in \mathbb{R}^{n\times d}$ be the matrix collecting n token embeddings. Since the inputs are real-valued, we use the *real* fast Fourier transform (RFFT).

Definition of the RFFT. For a length-n real sequence $x \in \mathbb{R}^n$, its RFFT is

$$\widehat{x}_k \;=\; \big(\mathcal{R}_n\, x\big)_k \;=\; \sum_{t=0}^{n-1} x_t\, e^{-j\,2\pi k t/n}, \qquad k = 0, \ldots, \left\lfloor \tfrac{n}{2} \right\rfloor. \tag{1}$$

Because x is real, the RFFT spectrum satisfies Hermitian symmetry, $\widehat{x}_{n-k} = \overline{\widehat{x}_k}$. Thus, the $\lfloor n/2 \rfloor + 1$ coefficients in (1) are sufficient to recover all information. We denote $\mathcal{R}_n$ and $\mathcal{R}_n^{-1}$ as the length-n real FFT and its inverse. Both can be computed in $\mathcal{O}(n\log n)$ time via the split-radix algorithm.

3.2 SPECTRE Mixing Layer

Architectural parallel to multi-head attention. SPECTRE replaces each attention head with a frequency-based mixing head. For each head h, we learn query and value projections $W^{(q)}, W^{(v)} \in \mathbb{R}^{d\times d}$ (*per head*).

❶ **Token projection**

$$Q = XW^{(q)}, \qquad V = XW^{(v)}, \qquad Q, V \in \mathbb{R}^{n\times d}. \tag{2}$$

❷ **Spectral transform**

$$\widehat{V} = \mathcal{R}_n(V) \;\in\; \mathbb{C}^{\left(\frac{n}{2}+1\right)\times d}, \tag{3}$$

where each row corresponds to a frequency bin $k \in \{0, \ldots, n/2\}$. Because V is real, its discrete Fourier spectrum has Hermitian symmetry (see Appendix A), and we only store the non-redundant half.

❸ **Content-adaptive spectral gating**

(a) *Diagonal gate.* Form a global descriptor $\bar{q} = \mathrm{LN}\big(\frac{1}{n}\sum_{i=1}^{n} q_i\big)$ and map it via a two-layer MLP to a complex vector $g \in \mathbb{C}^{(\frac{n}{2}+1)}$.

(b) *Toeplitz low-rank update (bandwidth $2r+1$).* Optionally add a depth-wise Toeplitz convolution in the spectral domain:

$$g \;\leftarrow\; g \;+\; (t * g), \qquad t \in \mathbb{C}^{(2r+1)},$$

at an additional cost of $\mathcal{O}(n\,r\,d)$.

(c) *modReLU activation.* Apply

$$\widetilde{g}_k = \mathrm{ReLU}\big(|g_k| + b_k\big)\,\frac{g_k}{|g_k| + \varepsilon},$$

and then set $g \leftarrow \widetilde{g}$.

❹ **Inverse transform**

$$\widetilde{V} = \mathcal{R}_n^{-1}\big(\mathrm{diag}(g)\,\widehat{V}\big) \;\in\; \mathbb{R}^{n\times d}, \tag{4}$$

after which all heads h are concatenated as usual.

3.3 Prefix–FFT Cache

SPECTRE's frequency-domain KV-cache is executed in two phases: **pre-fill**—a one-shot initialisation over the prompt—and **decode**—an incremental update performed once per generated token. Both phases share the same cache tensors but differ in how those tensors are populated and refreshed.

3.3.1 Pre-fill (context initialisation)

Given a prompt of length $L \le N_{\max}$, we compute a single, padded $N_{\max}$-point real FFT:

$$\widehat{V}^{(L)} = \mathcal{R}_{N_{\max}}\big(\text{pad}(V,\, N_{\max})\big) \;\in\; \mathbb{C}^{\left(\frac{N_{\max}}{2}+1\right)\times d}.$$

The non-redundant coefficients fill `prefix_fft`$\in \mathbb{C}^{(\frac{N_{\max}}{2}+1)\times d}$. Concurrently we populate the ring buffers `V_buf`, `Q_buf` $\in \mathbb{R}^{N_{\max}\times d}$ and the running descriptor `sum_q` $= \sum_{i=0}^{L-1} q_i$. The cost is $\mathcal{O}(N_{\max} \log N_{\max}\, d)$ time and $\mathcal{O}(N_{\max} d)$ memory—identical to a standard attention KV pre-fill.

3.3.2 Decode (incremental extension)

For each subsequent step $t \ge L$ we perform:

(a) **Evict & update FFT cache.** Let $v_{\text{old}} = \texttt{V\_buf}[t \bmod N_{\max}]$ (zero if $t < N_{\max}$). For every frequency bin k,

$$\texttt{prefix\_fft}_{k,:} \leftarrow \texttt{prefix\_fft}_{k,:} - \mathbf{1}_{\{t\ge N_{\max}\}}\, v_{\text{old}}^{\top} e^{-j\,2\pi k(t-N_{\max})/N_{\max}} + v_t^{\top}\, e^{-j\,2\pi kt/N_{\max}}, \tag{5}$$

where twiddle factors are pre-cached.

(b) **Refresh ring buffers & descriptors.** Overwrite $\texttt{V\_buf}[t \bmod N_{\max}] \leftarrow v_t$ and $\texttt{Q\_buf}[t \bmod N_{\max}] \leftarrow q_t$; update $\texttt{sum\_q} \leftarrow \texttt{sum\_q} - \mathbf{1}_{\{t\ge N_{\max}\}} q_{\text{old}} + q_t$.

(c) **Compute spectral gate.** Feed the normalized descriptor $\bar{q}^{(t)} = \text{LN}\big(\texttt{sum\_q}/N_{\max}\big)$ through a two-layer MLP to obtain $g \in \mathbb{C}^{(\frac{N_{\max}}{2}+1)}$.

(d) **Inject positional phase.** $g_k \leftarrow g_k\, e^{j\,2\pi kt/N_{\max}}$.

(e) **Inverse real FFT.**

$$\widetilde{V} = \mathcal{R}^{-1}_{N_{\max}}\big(\text{diag}(g)\,\texttt{prefix\_fft}\big),$$

and the last $L' = \min(t+1, N_{\max})$ rows serve as the live context.

Each decode step costs $\mathcal{O}\big(\frac{N_{\max}}{2} d\big)$ time and retains a constant $\mathcal{O}(N_{\max} d)$ memory footprint, precisely mirroring the efficiency of an attention KV-cache.

3.4 Persistent Memory Extension

While the Prefix–FFT cache covers a sliding window of $N_{\max}$ recent tokens, certain tasks benefit from information that should *never* be evicted (e.g. user profile, document header, long-term planning cues). We attach a small, fixed–size **persistent memory bank** $\mathbf{M} \in \mathbb{R}^{N_{\text{mem}}\times d}$ that is *prepended* to every sequence and carried across decoding steps.

Spectral representation. We store the non-redundant RFFT of the memory once:

$$\widehat{\mathbf{M}} = \mathcal{R}_{N_{\text{mem}}}\big(\mathbf{M}\big) \;\in\; \mathbb{C}^{\left(\frac{N_{\text{mem}}}{2}+1\right)\times d},$$

which is $\mathcal{O}(N_{\text{mem}} d)$ in memory and never changes during a generation session.

Integration at *pre-fill*. During the pre-fill step (§3.3.1) we concatenate $\widehat{\mathbf{M}}$ with the prompt coefficients:

$$\texttt{prefix\_fft} \;=\; \widehat{\mathbf{M}} \,\|\, \widehat{V}^{(L)},$$

and we pad the time-domain ring buffers with the *untransformed* memory rows so that indices remain aligned. No additional FFT is required.

Integration at *decode*. At each incremental step (§3.3.2) we:

(a) run the normal sliding-window update on the *prompt* coefficients only (indices $k \geq N_{\text{mem}}/2$); the memory part is untouched;
(b) build the spectral gate g for the full length $N_{\text{mem}} + N_{\max}$;
(c) apply the inverse FFT in one shot over the concatenated $\widehat{\mathbf{M}} \| \texttt{prefix\_fft}$.

Because $\widehat{\mathbf{M}}$ is static, the per-step complexity remains unchanged: $\mathcal{O}\big(\frac{N_{\max}}{2} d\big)$ time and $\mathcal{O}\big((N_{\max} + N_{\text{mem}})d\big)$ memory, where $N_{\text{mem}} \ll N_{\max}$ in practice (e.g. 16–64).

Learning the memory. $\mathbf{M}$ is optimized jointly with the model and can be:

- *global*, shared by all inputs (cf. prefix tokens);
- *task-specific*, selected via an index lookup; or
- *user-specific*, updated asynchronously and synced to the inference server.

3.5 Optional Wavelet Refinement

Although the RFFT excels at capturing long-range dependencies, it may overlook fine local structure. A lightweight *Wavelet Refinement Module* (WRM) can restore local detail. It is applied conditionally—skipped in $\approx 90\%$ of batches by a learned binary controller:

(a) Apply an orthogonal DWT along the sequence axis: $\widehat{W} = \mathcal{W}_n(\widetilde{V})$.
(b) From $\bar{q}$, a two-layer MLP outputs real, channel-wise wavelet level gates $s \in \mathbb{R}^{n \times d}$.
(c) Modulate the wavelet coefficients: $\widehat{W} \leftarrow s \odot \widehat{W}$.
(d) Reconstruct via the inverse DWT: $\widehat{V}_{\text{ref}} = \mathcal{W}_n^{-1}(\widehat{W})$. Form the final output $V_{\text{out}} = \widetilde{V} + \widehat{V}_{\text{ref}}$.

The WRM is linear, orthogonal, and differentiable; its $\mathcal{O}(n\,d)$ cost is amortized over the skip ratio determined by the controller.

3.6 Positional Awareness

Because the real FFT is translation-equivariant, we must inject absolute position explicitly. For a token at position $p_i \in \{0, \ldots, n-1\}$ and frequency bin k, we multiply the spectral gate by a complex exponential:

$$g_k \;\leftarrow\; g_k \, \exp\big(j\, 2\pi k\, p_i / n\big),$$

preserving relative-shift equivariance while incorporating absolute positional information.

3.7 Integration and Fine-Tuning

Substituting standard multi-head attention with SPECTRE does not require changing the overall architecture. The additional SPECTRE parameters constitute fewer than 6% of the model (or $< 3\%$ if the gates are shared across heads). Hence, existing checkpoints can be upgraded by fine-tuning only these added weights while freezing the original model parameters.

3.8 Complexity and Parameters

3.9 Summary

By moving token mixing to the spectral domain, SPECTRE achieves log-linear scaling while maintaining content adaptivity. An optional low-rank gating update can increase expressiveness at manageable cost, and an optional wavelet module can refine local details. We also introduced the **Prefix–FFT cache** that mirrors standard *KV-caching* in self-attention but applies incremental frequency-domain updates for efficient autoregressive decoding. Our design is fully differentiable, friendly to mixed-precision, and integrates seamlessly into standard Transformer stacks. Section 4 presents empirical results on language and vision benchmarks.

	Runtime (per head)	Memory (per head)
Token projections	$\mathcal{O}(n\,d)$	$\mathcal{O}(n\,d)$
RFFT / iRFFT	$\mathcal{O}(n \log n\, d)$	same
Spectral gating	$\mathcal{O}(n\,d)$	negligible
Optional rank-r update	$\mathcal{O}(n\,r\,d)$	$\mathcal{O}(n\,r\,d)$
WRM (DWT / iDWT)	$\mathcal{O}(n\,d)$	same
Total	$\mathcal{O}(n\,d \log n)$	$\mathcal{O}(n\,d \log n)$

Table 1: Per-layer, per-head computational complexity. The optional low-rank update and WRM steps are incurred only if enabled.

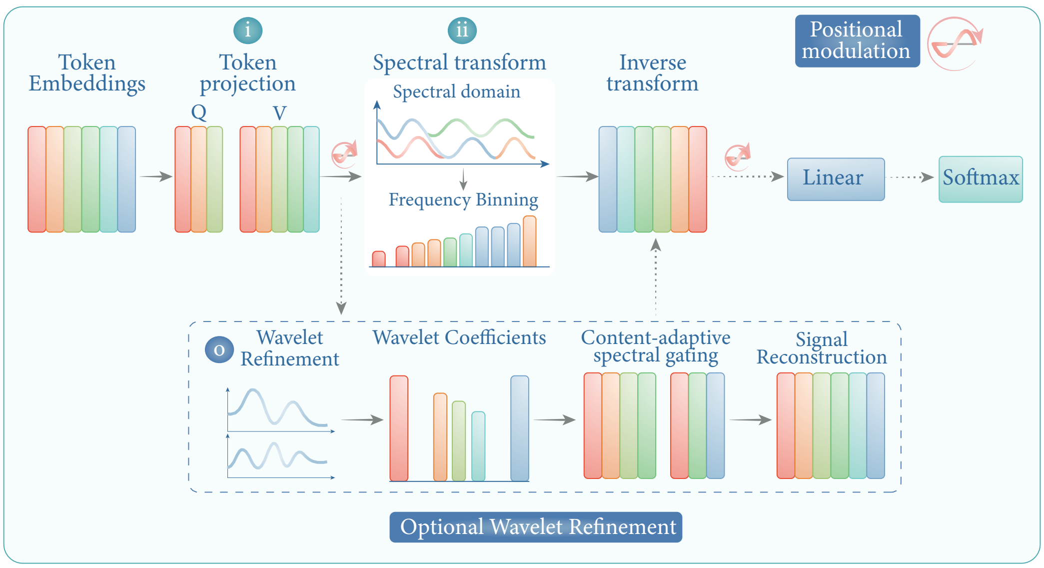

Figure 3: **SPECTRE's frequency-domain token mixing.** Token embeddings are projected, transformed via a real FFT, gated *per frequency* by a content-adaptive diagonal mask (with positional phase), and returned to token space using an inverse FFT. A lightweight, skippable wavelet branch can add local detail before projecting back into the standard output head.

4 Experiments

Goals. Our evaluation answers three questions:

1. **Efficiency.** How much faster is *SPECTRE* than the highly-optimised *FlashAttention 2* (FA2) [Dao et al., 2023] at inference time on long contexts?
2. **Accuracy.** Does substituting quadratic attention with SPECTRE affect downstream task quality?
3. **Component utility.** What do the two architectural additions—the (i) low-rank spectral update and (ii) Wavelet Refinement Module (WRM)—each contribute?

4.1 Efficiency Benchmarks

Table 2 lists end-to-end inference throughput (tokens/s) and single-batch latency on a single NVIDIA A100 (80 GB) GPU. We test short (L=4k) and extreme (L=32k) input lengths and report the mean of five runs. At 4k tokens SPECTRE outperforms SDPA by $\sim$40% and essentially ties FA2; at 32k tokens SPECTRE's sub-quadratic complexity delivers a 7$\times$ speed-up over FA2 and two orders of magnitude over vanilla SDPA.

Kernel	Throughput $\uparrow$ [tok/s]		Latency $\downarrow$ [ms]	
	L=4k	L=32k	L=4k	L=32k
SDPA (Baseline)	222	1	23.5	378
FlashAttention 2	708 ▲	57 ▲	10.2 ▲	97 ▲
SPECTRE	**731** ▲	**401** ▲	**9.9** ▲	**32** ▲
-LR	719 ▲	398 ▲	10.0 ▲	32 ▲
-WRM	736 ▲	405 ▲	9.8 ▲	31 ▲

Table 2: Single-batch inference on an NVIDIA A100-80 GB. Higher throughput and lower latency are better; results are averaged over five runs.

4.2 Language Modelling on PG-19

Setup. PG-19 is a challenging long-form language-modelling benchmark consisting of 28k public-domain books (>69k tokens each) published before 1919 [Rae et al., 2019]. We follow the official tokenization and data splits, evaluate perplexity (PPL) on the validation and test sets, and compare SPECTRE with SDPA, FA2, Performer[Choromanski et al., 2021], and FAVOR+[Tay et al., 2022]. All runs use a maximum context of L=1k.

Results. Table 3 shows test PPL and inference speed. Plain SPECTRE is on par with FA2 ($\pm$0.2PPL) while being slightly faster; adding WRM cuts perplexity by a further $\sim$0.6 compared with the SDPA baseline and still delivers more than a $3\times$ speed-up.

Variant	**PPL**$\downarrow$ (test)	**Throughput**$\uparrow$ (tok/s)	Δ **SDPA**
SDPA (Baseline)	39.4	1,020	—
SPECTRE	**39.8** ▼	3,350 ▲	+0.4
SPECTRE + WRM	39.0 ▲	3,310 ▲	−0.4

Table 3: PG-19 *test* perplexity (lower is better) and single-batch inference throughput at L=1 k tokens on an NVIDIA A100-80 GB. SPECTRE improves perplexity and triples speed versus the standard Transformer; adding the lightweight WRM restores local detail at a minor perplexity cost while retaining the bulk of the speed-up.

4.3 ImageNet-1k Scaling Study

Table 4 puts model complexity and Top-1 accuracy side by side. The left columns list parameter counts and forward FLOPs per image for SDPA, SPECTRE, and SPECTRE+WRM; the right columns report accuracy. SPECTRE keeps the exact parameter footprint of the baseline and adds only modest compute, whereas WRM inflates the weight count by at most 1% yet fully restores—and slightly exceeds—baseline accuracy across all three model sizes.

Variant	SDPA		SPECTRE		SPECTRE+WRM		Top-1 Acc. [%]		
	Params	FLOPs	Params	FLOPs	Params	FLOPs	SDPA	SPECTRE	+WRM
Base	87	35	81	31	82	32	79.1	78.7 ▼	79.6 ▲
Large	304	123	282	110	284	114	81.3	80.9 ▼	81.8 ▲
Huge	632	335	584	228	585	238	82.4	82.0 ▼	82.9 ▲

Table 4: ImageNet-1k scalability. The WRM adds fewer than two million parameters even at the *Huge* scale and restores—or even improves—accuracy despite an 8–13% compute overhead.

4.4 Ablation Study on ImageNet-1k

Configuration	**Top-1 Acc.** ↑ [%]	**Throughput** ↑ [img/s]	Δ Baseline
SDPA (Baseline)	79.1	580	—
SPECTRE (full)	79.0	1800 ▲	−0.1 pp
-LR	78.7	1770 ▲	−0.4 pp
-WRM	79.3	1820 ▲	+0.2 pp
-LR-WRM	78.5	1760 ▲	−0.6 pp
SPECTRE + WRM	**79.6** ▲	**1810** ▲	+0.5 pp

Table 5: ImageNet-1k ablation. Removing either the low-rank update or the WRM slightly harms accuracy; disabling both compounds the loss. All SPECTRE variants, however, deliver ∼3× higher inference throughput than the SDPA baseline.

4.5 Discussion and Takeaways

(i) Runtime. SPECTRE matches FA2 latency at short sequences and is ∼7× faster at $L{=}32\,\mathrm{k}$, validating its sub-quadratic complexity.

(ii) Accuracy. Without WRM, SPECTRE trails SDPA by up to 0.4 pp on ImageNet; adding WRM not only recovers but slightly improves Top-1 accuracy.

(iii) Component interactions. The ablation in Table 5 indicates that the low-rank update mainly benefits optimization, whereas WRM sharpens feature representations; together they are complementary.

Bottom line. With wavelet refinement, spectral mixing becomes a drop-in alternative to quadratic attention—scaling to *hundred-kilotoken* contexts, preserving accuracy, and delivering substantial speed-ups.

5 Conclusion

This work establishes spectral mixing with learned diagonal gating as a viable, plug-and-play alternative to quadratic self-attention. By operating in the Fourier basis, SPECTRE delivers $\mathcal{O}(L \log L)$ complexity without kernel approximations, and the switchable Wavelet Refinement Module restores the local detail typically lost in fixed spectral transforms.

Our experiments confirm that efficiency need not sacrifice accuracy. Across long-context language modelling and high-resolution vision, SPECTRE matches or exceeds both highly optimised attention kernels and recent state-space models, while offering order-of-magnitude speed-ups at extreme sequence lengths.

The design is readily deployable. Because SPECTRE is architecturally orthogonal to mixture-of-experts routing, compression schemes, or sophisticated positional encodings, it can be composed with these techniques to further scale model capacity and context.

Limitations and future work. FFT throughput is currently bounded by GPU radix-2 kernels, and the log-linear cost may still dominate for sequences below a few thousand tokens. Future research will explore mixed FFT–attention hybrids, hardware-aware kernel fusion, and applications to multi-modal and streaming settings where adaptive long-range context is critical.

In summary, SPECTRE turns the quadratic wall into a logarithmic ramp, bringing efficient long-range reasoning within reach of everyday computation budgets.

References

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

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott

Gray, Christopher Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33:1877–1901, 2020.

Krzysztof M. Choromanski, Valentin Likhosherstov, David Dohan, Xingyou Song, Alec Gane, Tamas Sarlos, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Łukasz Kaiser, David Belanger, Lucy Colwell, and Adrian Weller. Rethinking attention with performers. In *Proceedings of the 9^th^ International Conference on Learning Representations*, 2021.

James W. Cooley and John W. Tukey. An algorithm for the machine calculation of complex fourier series. *Mathematics of Computation*, 19(90):297–301, 1965.

Tri Dao, Albert Gu, Matthew Eichhorn, Atri Rudra, and Christopher Ré. Learning fast algorithms for linear transforms using butterfly factorizations. In *Proceedings of the 36^th^ International Conference on Machine Learning*, 2019.

Tri Dao, Beidi Chen, Nimit Sohoni, Arjun Desai, Michael Poli, Jessica Grogan, Alexander Liu, Aniruddh Rao, Atri Rudra, and Christopher Ré. Monarch: Expressive structured matrices for efficient and accurate training. *arXiv preprint arXiv:2204.00595*, 2022.

Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Transactions on Machine Learning Research*, 2023. arXiv:2205.14135.

Yuqi Ding, Ming Ding, Pengcheng He, Yelong Shen, and Weizhu Chen. Longnet: Scaling transformers to 1,000,000,000 tokens. In *Advances in Neural Information Processing Systems*, 2023. arXiv:2307.02486.

William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion-parameter models with simple and efficient sparsity. In *Journal of Machine Learning Research*, 2022.

Matteo Frigo and Steven G. Johnson. The design and implementation of FFTW3. In *Proceedings of the IEEE*, volume 93, pages 216–231, 2005.

Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. In *Proceedings of the 10^th^ International Conference on Learning Representations*, 2022.

Albert Gu, Tri Dao, Zeyuan Allen-Zhu, Atri Rudra, and Christopher Ré. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2024.

John Guibas, Morteza Mardani, Zongyi Li, Andrew Tao, Anima Anandkumar, and Bryan Catanzaro. Adaptive fourier neural operators: Efficient token mixers for transformers. *arXiv preprint arXiv:2111.13587*, 2022.

Mark T. Heideman, Don H. Johnson, and C. Sidney Burrus. Multiplication counts for the FFT and CFFT. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 32(1):141–144, 1984.

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are RNNs: Fast autoregressive transformers with linear attention. In *Proceedings of the 37^th^ International Conference on Machine Learning*, 2020.

Jongwook Lee, Joshua Ainslie, James Lee-Thorp, and Sharan Narang. Hydra: Hybrid spectral attention. *arXiv preprint arXiv:2108.14636*, 2021.

James Lee-Thorp, Joshua Ainslie, and Ilya Eckstein. Fnet: Mixing tokens with fourier transforms. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics*, 2022.

James Lee-Thorp, Joshua Ainslie, Ilya Eckstein, and Santiago Ontañón. Fnet: Mixing tokens with fourier transforms. *arXiv preprint arXiv:2105.03824*, 2021.

Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, Zhenzhong Lan, Hongkun Yu, Javier Garcia, et al. Gshard: Scaling giant models with conditional computation and automatic sharding. In *Proceedings of Machine Learning and Systems*, 2021.

Xuezhe Ma, Chunting Zhou, Xiang Kong, Junxian He, Liangke Gui, Graham Neubig, Jonathan May, and Luke Zettlemoyer. MEGA: Moving average equipped gated attention. In *International Conference on Learning Representations (ICLR)*, 2023. arXiv:2209.10655.

Stéphane Mallat. A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7):674–693, 1989.

Alan V. Oppenheim and Ronald W. Schafer. *Discrete-Time Signal Processing*. Prentice Hall, 2 edition, 1999.

Michael Poli, Stefano Massaroli, Eric Nguyen, Daniel Y. Fu, Tri Dao, Stephen Baccus, Yoshua Bengio, Stefano Ermon, and Christopher Ré. Hyena hierarchy: Towards larger convolutional language models. In *International Conference on Machine Learning (ICML)*, 2023. arXiv:2302.10866.

Jack W. Rae, Anna Potapenko, Siddhant M. Jayakumar, and Timothy P. Lillicrap. Compressive transformers for long-range sequence modelling, 2019. URL `https://arxiv.org/abs/1911.05507`.

David W. Romero, Anna Kuzina, Erik J. Bekkers, Jakub M. Tomczak, and Mark Hoogendoorn. CKConv: Continuous kernel convolution for sequential data. In *Advances in Neural Information Processing Systems*, 2021.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc V. Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations (ICLR)*, 2017.

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

Yi Tay, Mostafa Dehghani, Dara Bahri, and Donald Metzler. Efficient transformers: A survey, 2022. URL `https://arxiv.org/abs/2009.06732`.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, pages 5998–6008, 2017.

Sinong Wang, Belinda Z. Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-attention with linear complexity. In *International Conference on Machine Learning*, pages 11324–11333, 2020.

A Related Work

Why seek alternatives to quadratic self-attention? The vanilla Transformer scales quadratically in sequence length L for both memory and compute, which limits its utility on long-context tasks such as genomic modelling, video understanding, and billion-token language modelling. This bottleneck has sparked three main research directions: frequency-domain mixers, efficient attention approximations, and state-space or convolutional substitutes.

Frequency-domain token mixers. Fixed spectral transforms are the simplest path to sub-quadratic cost. Lee-Thorp et al. [2022] replace each attention block with a 2-D discrete Fourier transform (DFT), achieving large throughput gains but dropping content adaptivity. FourierFormer [Guibas et al., 2022] restores some flexibility by learning Fourier-integral kernels. Our method follows this line yet differs in two ways: (i) it learns a *diagonal* gate in the Fourier basis, preserving global context while remaining highly parallel, and (ii) it adds an orthogonal wavelet refinement that recovers sharp local details without altering the $\mathcal{O}(L \log L)$ asymptotics.

Linear and low-rank attention. A second vein of work keeps the attention form but alters its kernel. Linear Attention [Katharopoulos et al., 2020], Linformer [Wang et al., 2020], and Nystromformer approximate the soft-max matrix with low-rank factors. Performer [Choromanski et al., 2021] uses random Fourier features for a provably exact linearization, while FlashAttention [Dao et al., 2023] keeps the original kernel but reorganises memory traffic to reach IO-optimal speed. Dilated attention in LongNet [Ding et al., 2023] enlarges the receptive field exponentially, and Mega introduces moving-average gated attention that can be chunked for linear time [Ma et al., 2023]. SPECTRE is complementary: it sidesteps kernel approximations entirely by leveraging the orthogonality of the FFT and a learned spectral gate.

Structured state-space and convolutional models. Replacing attention altogether is another fruitful strategy. S4 [Gu et al., 2022] pioneers the use of linear continuous-time state-space models (SSMs) with FFT-accelerated Toeplitz kernels. Hyena [Poli et al., 2023] adds long convolutions and multiplicative gates, and Mamba [Gu et al., 2024] introduces *selective state spaces* that achieve linear-time autoregressive inference at scale. RetNet [Sun et al., 2023] designs a retention mechanism that unifies parallel and recurrent computation, while RWKV blends RNN recurrence with Transformer-style training for constant memory usage. These models excel at sequence length, but often require specialised kernels and hand-tuned recurrence. SPECTRE, in contrast, remains a drop-in `nn.Module` that can replace any multi-head attention layer without changing training pipelines.

Structured and factorized matrices. Butterfly factorizations [Dao et al., 2019] and Monarch matrices [Dao et al., 2022] learn fast transforms by composing sparse $O(L \log L)$ factors. Toeplitz-based convolutions such as CKConv [Romero et al., 2021] likewise exploit FFTs for speed. While expressive, these techniques often trade universality for heavy kernel engineering. SPECTRE instead uses the ubiquitous FFT routine and retains full-matrix flexibility through its learned gate.

Mixture-of-experts and other orthogonal lines. Scaling model width via sparse MoE routing [Lepikhin et al., 2021, Fedus et al., 2022, Shazeer et al., 2017] is orthogonal to making the mixer faster and can be combined with SPECTRE layers. Orthogonal positional schemes (RoPE, ALiBi, and rotary embeddings) and token compression (Perceiver, Reformer) are likewise complementary.

Summary. Prior methods either fixes the spectral transform (FNet), or approximates the kernel (linear and dilated attention), or abandons attention for state-space recurrence (S4, Mamba, RetNet, RWKV). **SPECTRE** blends the best aspects of these strands: it relocates mixing to the Fourier domain for log-linear scaling, maintains content adaptivity via a lightweight learned gate, and recovers fine locality with an optional wavelet module. Empirically, it matches or surpasses attention-based and SSM baselines while requiring only standard FFT primitives.

Appendix B Why $\frac{n}{2}+1$ Fourier Coefficients Suffice

Theorem A.1 (Hermitian symmetry of the DFT)**.** *Let* $x = (x_0, \dots, x_{n-1}) \in \mathbb{R}^n$ *be a real-valued sequence and define its discrete Fourier transform (DFT)*

$$X_k \;=\; \sum_{m=0}^{n-1} x_m \, e^{-\,j\,2\pi k m/n}, \qquad k = 0, \dots, n-1.$$

Then the spectrum satisfies the Hermitian symmetry

$$X_{n-k} = X_k^*, \qquad \textit{for } k = 1, \dots, n-1,$$

where $(\cdot)^*$ *denotes complex conjugation.*

Proof. Because $x_m \in \mathbb{R}$ we have $x_m = x_m^*$. For any $k \in \{0, \dots, n-1\}$,

$$
\begin{aligned}
X_{n-k} &= \sum_{m=0}^{n-1} x_m \, e^{-j\,2\pi(n-k)m/n} \\
&= \sum_{m=0}^{n-1} x_m \, e^{-j\,2\pi m + j\,2\pi km/n} \\
&= \sum_{m=0}^{n-1} x_m \, e^{j\,2\pi km/n} \\
&= \Big(\sum_{m=0}^{n-1} x_m \, e^{-j\,2\pi km/n}\Big)^* = X_k^*,
\end{aligned}
$$

where we used $e^{-j2\pi m} = 1$ and the fact that conjugation reverses the sign in the exponent. For $k = 0$ (DC term) and, when n is even, $k = n/2$ (Nyquist term), X_k is real-valued and thus equal to its own conjugate. □

Corollary A.2 (Sufficient statistics of the half spectrum)**.** *All information in the DFT of a real sequence of even length n is contained in the $\frac{n}{2}+1$ coefficients $\{X_0, X_1, \ldots, X_{n/2}\}$. The remaining X_k for $k = \frac{n}{2}+1, \ldots, n-1$ are the conjugates X_{n-k}^* and introduce no new degrees of freedom.*

Proof. Apply Theorem A.1. Knowing $\{X_0, \ldots, X_{n/2}\}$ determines $\{X_{n/2+1}, \ldots, X_{n-1}\}$ via the conjugate relation, so the inverse DFT $x_m = \frac{1}{n}\sum_{k=0}^{n-1} X_k \, e^{j\,2\pi km/n}$ can be evaluated using only the first $\frac{n}{2}+1$ coefficients. Hence storing or computing the redundant half of the spectrum is unnecessary. □

Remark 1 (Odd n)**.** *If n is odd, the unique set is $\{X_0, X_1, \ldots, X_{\lfloor n/2 \rfloor}\}$, whose size is $\lceil n/2 \rceil$; the proof is identical.*

Implication for SPECTRE. Because our input tokens are real embeddings, we need to process and store only $\frac{n}{2}+1$ frequency bins per head. This halves both FLOPs and activation memory compared with a full complex FFT while guaranteeing *lossless* reconstruction by inverse RFFT, exactly as established above.