Jet-Long: Efficient Long-Context Extension with Dynamic Bifocal RoPE

Haozhan Tang, Zerui Wang, Yuxian Gu, Song Han, Han Cai

NVIDIA
https://github.com/jet-ai-projects/jet-long

Abstract: Modern LLMs are increasingly deployed in long-context applications such as retrieval-augmented generation, repository-level coding, and agentic workflows whose accumulated reasoning and tool traces routinely push the input an order of magnitude past the pretraining window, making zero-shot context extension the dominant deployment path for open-weight checkpoints. Most existing zero-shot methods fix a single rescaling factor up front, so an aggressive factor sacrifices short-context fidelity while a conservative one breaks down at long contexts. We propose **Jet-Long**, a tuning-free zero-shot method that pairs a local RoPE-faithful window with a long-range window whose rescaling factor adapts dynamically to the current sequence length, recovering the base model exactly at short inputs while extrapolating cleanly at long ones. An inclusion–exclusion attention merge and an on-the-fly RoPE correction rotation make the bifocal construction essentially free at inference; fused into a single CuTe kernel, long-context prefill reaches up to 1.39× FA2 throughput on H100 (approaching the Hopper-only FA4), and single-batch generation incurs $\leq 4\%$ overhead at every length. On Qwen3-1.7B/4B/8B [1] up to 128K context, Jet-Long leads RULER by +4.79/+2.18/+2.03 pp over the strongest baseline at 1.7B/4B/8B, achieves the best overall accuracy on HELMET-RAG (a benchmark identified by HELMET as the most efficient predictor of downstream long-context performance [2]) and attains the lowest PG-19 perplexity. Jet-Long also generalizes to hybrid attention architectures such as Jet-Nemotron [3] for further long-context improvement without retraining, and remains hyperparameter-resilient for ease of deployment.

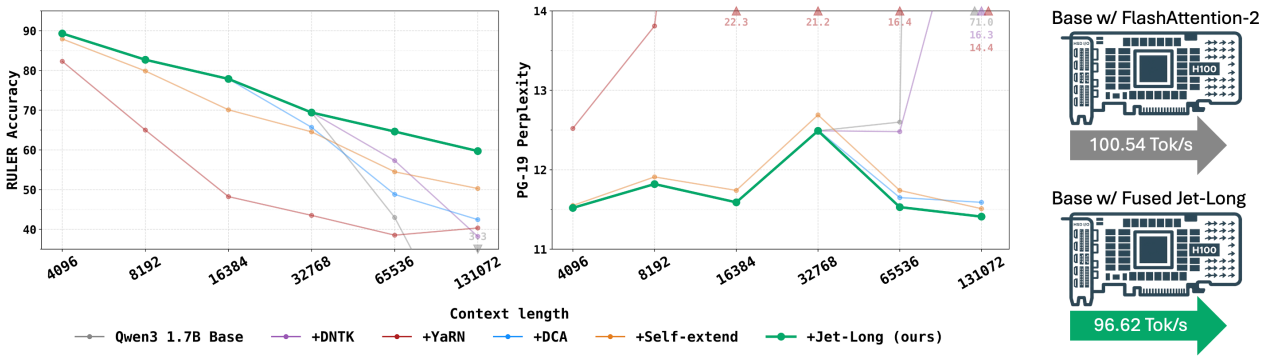


Figure 1 | Comparison between Jet-Long and baseline methods, applied on Qwen3-1.7B-Base, on per-length accuracy aggregated over all 13 RULER tasks and per-length perplexity in PG-19, alongside single-batch generation throughput on H100 at 128K context (the worst length we test). Jet-Long achieves the highest accuracy and lowest perplexity at extended context lengths, preserves the base model's pretrained performance within the training context, and incurs $\leq 4\%$ latency overhead relative to FlashAttention-2 [4].

1. Introduction

Large language models (LLMs) are now deployed in long-context applications including long-document QA, repository-level code understanding, retrieval-augmented generation, and multi-step agentic workflows [5, 6, 7, 8, 9, 10, 11, 12, 13]. The pressure is most severe in agentic LLMs that interleave reasoning, planning, and tool use across many turns [14, 15, 16], and in coding agents operating over real software repositories [17, 18], where source code, execution traces, and tool outputs routinely accumulate to 100K+ tokens per task.

Corresponding author(s): Han Cai (hcai@nvidia.com).

Training directly at long context remains expensive: efficient kernels like FlashAttention [19] and Ring Attention [20] make memory linear but compute stays quadratic in sequence length, and long-context data is scarce while long-context fine-tuning often degrades short-context behavior [21, 22]. Models are therefore pretrained at a moderate window (4K–32K tokens) and expected to handle longer inputs at inference, a setting known as *context extension* [23, 24]. *Zero-shot context extension* (without fine-tuning) has become the dominant deployment mode for open-weight LLMs [1, 25, 26], since a single released checkpoint must support arbitrary downstream context lengths.

Vanilla Transformer-based LLMs fail to generalize beyond the training window [23, 27, 28], with two failure modes (out-of-distribution RoPE rotations and softmax-attention diffusion) detailed in Section 2. A growing body of zero-shot methods (NTK [29] / Dynamic NTK [30], YaRN [31], Self-Extend [32], DCA [33]) addresses one or the other; these constitute the zero-shot baselines we benchmark against.

We propose **Jet-Long**, a tuning-free zero-shot context extension method that pairs a local RoPE-faithful window with a long-range window whose rescaling factor is *dynamic* in the current sequence length. Unlike YaRN, Self-Extend, and DCA, which fix a single grouping size or factor up front, Jet-Long's dynamic schedule preserves short-context behavior at short inputs while extrapolating cleanly at long ones. An inclusion–exclusion attention merge and on-the-fly correction rotation leave the KV cache unchanged and make the construction essentially free at inference.

Our contributions are:

- **Jet-Long**, a tuning-free bifocal context-extension method whose remote-window rescaling factor adapts dynamically to the current sequence length, keeping every remote rotation in-distribution while reproducing the base model exactly within its native context.
- An inclusion–exclusion attention merge that routes local and remote windows through three FlashAttention [19] passes, paired with an on-the-fly RoPE correction rotation that leaves the KV cache untouched during generation; fused into a single CuTe kernel, the construction reaches 1.28–1.39× FA2 prefill on H100 (approaching the Hopper-only FA4 [34]) and incurs $\leq 4\%$ overhead on generation at every length.
- Empirical evaluation on Qwen3-1.7B/4B/8B up to 128K context: Jet-Long leads the strongest zero-shot baseline on RULER [9] by $+4.79/+2.18/+2.03$ pp and is best or tied on HELMET-RAG [2] and PG-19 [35] perplexity; the single hyperparameter w_0 is robust to choice; and the construction transfers unchanged to the hybrid Jet-Nemotron [3] architecture.

2. Related Work

2.1. Why RoPE-based LLMs fail to extrapolate

Most modern open-weight LLMs use Rotary Position Embedding (RoPE) [36], which applies per-position rotations across geometrically spaced frequencies so attention depends only on relative position; earlier relative-position schemes such as ALiBi [23], T5's relative-position bias [37], and iRPE [38] have largely been supplanted. Two failure modes prevent these models from extrapolating beyond their training window.

(i) Position out-of-distribution. At sequence positions never seen during training, the low-frequency RoPE components produce rotation angles outside the training distribution, causing attention scores to behave erratically [27, 31, 28].

(ii) Attention diffusion and positional bias. As the key set grows, the softmax distribution flattens, dispersing probability mass over irrelevant tokens [39, 31]; separately, models exhibit a U-shaped positional attention bias that under-attends to middle-context information, degrading retrieval accuracy for centrally placed evidence [40].

These motivate two complementary zero-shot axes: interpolating RoPE frequencies or position indices to keep rotation angles in-distribution [27, 29, 30, 31, 32, 33, 41, 42], and attention penalties or temperature scaling to counteract softmax diffusion [31, 39, 41]. Jet-Long targets position-OOD via dynamic aliasing onto the pretrained rotation grid.

2.2. Zero-shot context extension methods

Frequency-rescaling methods. Position Interpolation (PI) [27] linearly compresses positions into the pretrained range. NTK-aware scaled RoPE [29] increases the RoPE base to preserve high-frequency components while interpolating low-frequency ones; Dynamic NTK (DNTK) [30] makes that base a function of the current sequence length so the scaling adapts at decode time. YaRN [31] combines per-dimension frequency partitioning with an attention-temperature correction. Beyond pure rescaling, ReRoPE [42] caps relative distances past a window threshold (implemented as a two-pass within/beyond-window attention blend at prefill), and GALI [41] interpolates at the attention-logit level rather than the embedding.

Grouped / chunked-position methods. A second line reuses only in-distribution position indices. Self-Extend [32] pairs a neighbor window with a grouped window in which blocks of tokens share a single position index. Dual Chunk Attention (DCA) [33] partitions the sequence into chunks and uses asymmetric query/key indices in the cross-chunk component so all relative distances stay within the pretrained range. LM-Infinite [39] earlier introduced a Λ-shaped mask retaining an attention sink and a recent window.

Jet-Long sits in the grouped-position family but makes the group size a function of the current sequence length: identity within the native window, and just large enough past it to keep every remote rotation in-distribution (Section 3.1).

2.3. Long-context training, architectures, and benchmarks

Training-based approaches extend the window via continued pretraining [24] but face quadratic FLOP costs (only partially mitigated by efficient kernels [19, 20]) and scarce long-context data with short-context regression risk [21, 22], motivating the zero-shot setting.

Alternative architectures replace dense softmax with sparse attention [43, 44], linear or kernel-based variants [45, 46], or state-space models [47]; sparser or non-softmax distributions naturally curb the attention diffusion and lost-in-the-middle bias that plague dense softmax at long context. A complementary line removes positional encoding entirely: NoPE outperforms explicit position encodings on length-generalization benchmarks [48], and open-weight models such as Kimi K2 [49] and NVIDIA Nemotron Nano 2 [50] adopt NoPE in their hybrid layers. Both lines typically require training from scratch or substantial fine-tuning and therefore complement rather than compete with our zero-shot setting; Jet-Long itself extends smoothly to hybrid designs (Section 4.3, Jet-Nemotron [3]).

We evaluate on RULER [9] (synthetic recall over 13 tasks), HELMET-RAG [2] (the HELMET study's best overall predictor of downstream long-context performance), and PG-19 [35] long-form perplexity, against the strongest zero-shot baselines; we additionally test transfer to the hybrid Jet-Nemotron backbone (Section 4.3).

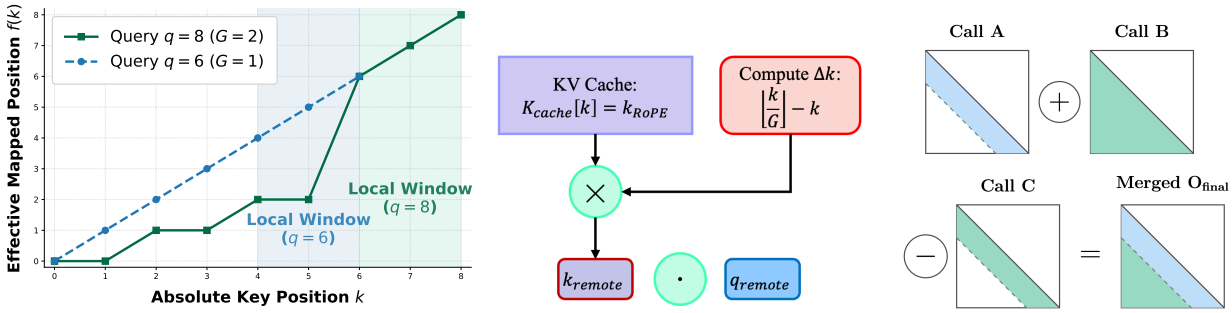
3. Methodology

The zero-shot methods of Section 2.2 fix a single rescaling factor up front, forcing a tradeoff between short-context fidelity and long-context reach. Jet-Long resolves this by making the factor *dynamic* in the current sequence length (Section 3.1); the resulting two-window computation matches FlashAttention [19] within the pretraining window and exceeds it at long context (Section 4.6), and leaves the KV cache untouched at decode (Sections 3.2–3.3).

We build on the dual-window (bifocal) decomposition of Self-Extend [32] (a related but architecturally three-way ancestor is DCA [33]): a local window of size w_0 that retains classic RoPE (preserving the model's pretraining behavior exactly), and a remote window governed by a rewritten position mapping $f(\cdot)$ that maps positions back into the training range. For a query at position q and a key at position k, the pre-softmax attention score is

$$S(q,k) = \begin{cases} \mathrm{RoPE}(\mathbf{x}_q, q)^\top \mathrm{RoPE}(\mathbf{x}_k, k) & \text{if } q-k \le w_0, \\ \mathrm{RoPE}(\mathbf{x}_q, f(q))^\top \mathrm{RoPE}(\mathbf{x}_k, f(k)) & \text{if } q-k > w_0. \end{cases} \tag{1}$$

Let L denote the current sequence length and $w_{\text{pretrained}}$ the pretrained context window. When $L \le w_{\text{pretrained}}$, $f(x) = x$ and Eq. (1) reduces to the unmodified base model. Jet-Long's contribution lies in the choice of $f(\cdot)$ and in the inference-time machinery that makes the two windows interact for free.



(a) Dynamic bifocal mapping (b) KV cache correction at decode (c) Inclusion–exclusion prefill

Figure 2 | Overview of Jet-Long. **(a)** The local window of width w_0 keeps classic RoPE; remote keys are routed through a dynamic position map $f(x) = \lfloor x/G \rfloor$ with $G = \max(1, \lceil L/w_{\text{pretrained}} \rceil)$, so the remote group size adapts to the current context length L. **(b)** At generation time, the KV cache stores positions in the original coordinate. An on-the-fly rotation pair (Δq on the active query, Δk on each cached key) is fused into FlashAttention to realize the remote view, leaving the cache unchanged. **(c)** Prefill is computed by an inclusion–exclusion combination of three FlashAttention calls (full remote, local-only-with-RoPE, local-only-with-remap), stabilized via LogSumExp and fused into a single CuTe kernel.

3.1. Dynamic extrapolation factor

To keep remote-window rotation angles in-distribution, the group size G must scale with the current sequence length L, as illustrated in Figure 2a.

The natural dynamic factor is, for continuous-style RoPE rescaling (e.g., DNTK [30] and dynamic-YaRN [31] variants), the scaling ratio $s = \max(1, L/w_{\text{pretrained}})$; for discrete grouped methods (e.g., Self-Extend [32]), it is an integer group size G. Jet-Long uses discrete grouping because an LLM only encounters a finite, discrete set of relative RoPE rotation angles during pretraining; aliasing positions onto that pretrained grid keeps every remote-window angle exactly in-distribution, an integer relative position the model has actually been trained on. We ablate this choice against continuous frequency interpolation in Section 4.5. Whereas DNTK [30] adapts the RoPE base β (per-frequency), Jet-Long adapts the discrete group size G on the position-aliasing axis, keeping every remote angle on the model's training grid.

To maximize positional resolution, G is the smallest integer that keeps the compressed sequence within the pretrained window $w_{\text{pretrained}}$:

$$G = \max\left(1, \left\lceil \frac{L}{w_{\text{pretrained}}} \right\rceil\right) \tag{2}$$

The remote mapping is a floor division of absolute positions:

$$f(x) = \left\lfloor \frac{x}{G} \right\rfloor \tag{3}$$

By recomputing G as the sequence grows, Jet-Long applies the minimum compression that keeps every remote position in-distribution, maximizing positional resolution at every length.

3.2. Key-value cache management and correction rotation

Dynamic extrapolation poses a cache-management challenge: if G changes during generation, rewriting the KV cache with new extrapolated phases would require discarding and recomputing the entire cache.

As depicted in Figure 2b, Jet-Long avoids this overhead by maintaining a strict invariant: the cache stores only uncompressed base keys at their exact absolute positions k,

$$\mathbf{k}_{\text{cache}}[k] = \text{RoPE}(\mathbf{x}_k, k) \tag{4}$$

When the remote window needs the query at $f(q)$ and the key at $f(k)$, instead of recomputing those vectors

from scratch we apply a correction rotation on the fly using the position offsets

$$\Delta q = f(q) - q = \left\lfloor \frac{q}{G} \right\rfloor - q, \qquad \Delta k = f(k) - k = \left\lfloor \frac{k}{G} \right\rfloor - k. \tag{5}$$

This relies on standard RoPE composing additively in angle, $R_a R_b = R_{a+b}$ (per-position scalings beyond a pure rotation would break this and require recomputing keys from $\mathbf{x}_k$). Applying $\mathrm{RoPE}(\cdot, \Delta)$ to a vector encoding position p therefore produces the vector at position $p + \Delta$, so we apply the offset directly to the active query and cached keys before attention:

$$\mathbf{q}_{\text{remote}} = \mathrm{RoPE}(\mathbf{q}_{\text{local}}, \Delta q), \qquad \mathbf{k}_{\text{remote}} = \mathrm{RoPE}(\mathbf{k}_{\text{cache}}[k], \Delta k). \tag{6}$$

This constant-time operation reconstructs the extrapolated vectors in registers; the physical cache is never touched, so streaming generation runs across length boundaries without stalling.

3.3. Inclusion–exclusion prefill

To avoid materializing the full quadratic attention matrix, which exhausts memory during long-sequence prefill, Jet-Long achieves exact distance-based routing by combining the inclusion–exclusion principle with FlashAttention's [19] LogSumExp statistics. The merge requires three standard attention calls (Figure 2c), each returning an output $\mathbf{O}_X$ and an LSE vector $\boldsymbol{\ell}_X$: **(A)** sliding-window attention (local w_0) with base queries and keys, **(B)** full causal attention with remote queries and keys, and **(C)** sliding-window attention (local w_0) with remote queries and keys. Calls B and C apply the same remote rotation to local keys, so their local-subset contributions cancel exactly: $W_B\mathbf{O}_B - W_C\mathbf{O}_C$ (resp. $W_B - W_C$) collects only the remote-only term, while Call A supplies the base-local term. Stabilizing element-wise (per query position) via $M = \max(\boldsymbol{\ell}_A, \boldsymbol{\ell}_B, \boldsymbol{\ell}_C)$ and weights $W_X = \exp(\boldsymbol{\ell}_X - M)$, the final output (numerator and denominator both in FP32 to avoid catastrophic cancellation) is

$$\mathbf{O}_{\text{final}} = \frac{W_A\mathbf{O}_A + W_B\mathbf{O}_B - W_C\mathbf{O}_C}{W_A + W_B - W_C}. \tag{7}$$

This merge realizes exact distance-based routing without a boolean mask matrix, retaining FlashAttention's memory efficiency and near-FA2 throughput.

4. Experiments

4.1. Setup

Models. Our primary evaluation is on the Qwen3 base model family [1], namely Qwen3-1.7B-Base, Qwen3-4B-Base, and Qwen3-8B-Base, each of which has a 32,768-token (32K) native training window. We extend the usable context to lengths up to 131,072 (128K) at inference time, without any fine-tuning. To verify that Jet-Long generalizes beyond pure softmax-attention transformers, we additionally evaluate on the hybrid Jet-Nemotron-2B and Jet-Nemotron-4B models [3], which interleave softmax and linear-attention layers. We use $w_0 = 2048$ throughout the main results, and per-baseline hyperparameters (DNTK, YaRN, DCA, Self-Extend) are listed in Appendix C.

Long-context benchmarks. We report results on three complementary suites: (i) RULER [9], a synthetic recall stress test averaging over thirteen tasks; (ii) HELMET-RAG [2], averaged over the four default sub-tasks (NaturalQuestions, TriviaQA, PopQA, HotpotQA), which the HELMET study reports as the most efficient predictor of downstream long-context performance and which serves as our application-grounded benchmark; and (iii) PG-19 [35], on which we report long-form perplexity. RULER and HELMET-RAG accuracies are percentages, with gaps between methods reported in percentage points (pp); PG-19 is reported as token-level perplexity (ppl, lower is better). All RULER and HELMET-RAG generations use greedy decoding with the default RULER/HELMET prompts and per-task max-output-token limits; PG-19 is teacher-forced perplexity at stride 1024 over 100 books. Formal definitions and the geometric-mean aggregation used in the Avg columns are given in Appendix A. We further ablate the only Jet-Long hyperparameter, the local protected window size w_0, in Section 4.4.

Table 1 | Long-context performance on RULER (accuracy averaged over 13 tasks and 7 lengths from 4K to 128K), HELMET-RAG, and PG-19 perplexity (geometric mean over the same 7 lengths, lower is better), on three Qwen3 base sizes. Best per column in **bold**.

	RULER				HELMET-RAG				PG-19 ppl (↓)			
Method	1.7B	4B	8B	Avg	1.7B	4B	8B	Avg	1.7B	4B	8B	Avg
Base	60.93	69.94	73.13	68.00	36.20	44.33	47.24	42.59	16.13	14.84	12.80	14.59
DNTK [30]	69.14	79.75	83.54	77.48	41.27	50.81	55.55	49.21	12.60	10.78	9.13	10.84
YaRN [31]	52.99	70.11	78.49	67.20	32.24	43.63	53.41	43.09	16.39	12.08	9.91	12.79
DCA [33]	67.80	80.19	81.08	76.36	41.77	51.91	56.12	49.93	11.77	9.89	8.77	10.14
Self-Extend [32]	67.86	80.84	84.71	77.80	**43.01**	52.98	56.86	50.95	11.85	9.95	8.81	10.20
Jet-Long (ours)	**73.93**	**83.02**	**86.74**	**81.23**	42.28	**53.61**	**57.34**	**51.08**	**11.72**	**9.85**	**8.73**	**10.10**

Inference efficiency. We implement Jet-Long as a fused CuTe kernel that merges the three attention calls of Section 3.3 into a single launch, and benchmark prefill and single-batch generation throughput on a single H100 against the highly optimized FlashAttention-2 [4] and FlashAttention-4 [34] baselines applied to the unmodified base models.

All experiments are run on NVIDIA H100 Tensor Core GPUs.

4.2. Main results on long-context extension

The average scores on the three benchmarks are reported in Table 1. Jet-Long is best on every RULER and PG-19 column across all three Qwen3 sizes (over Base and the four extrapolation baselines), and best on HELMET-RAG at 4B and 8B (0.73 pp behind Self-Extend at 1.7B). The RULER lead over the strongest baseline (DNTK at 1.7B; Self-Extend at 4B and 8B) is 4.79, 2.18, and 2.03 pp.

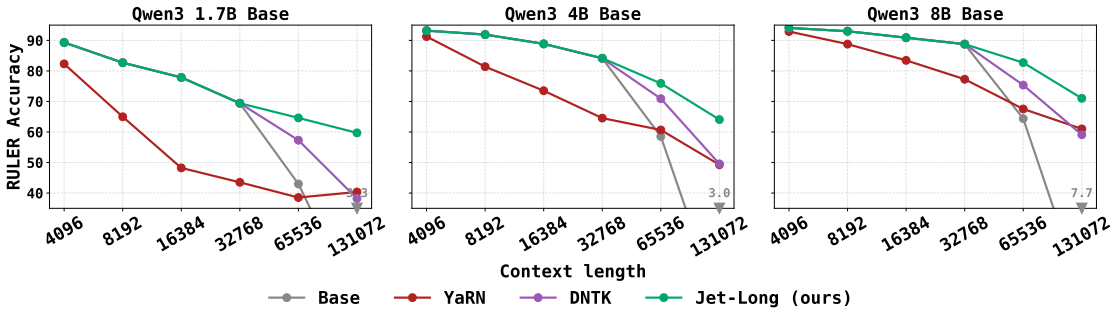


Figure 3 | RULER accuracy as a function of context length, averaged over the 13 RULER tasks, on Qwen3-1.7B/4B/8B-Base. Jet-Long preserves the pretrained model's performance within the training window (32K) and outperforms YaRN and DNTK at all extended context lengths (the full comparison vs. DCA and Self-Extend is in Table 1).

As Figure 3 shows, Jet-Long matches or substantially outperforms YaRN and DNTK at every length across all three sizes for RULER accuracy by length. Table 2 confirms the same pattern on PG-19: within the 32K training window, Jet-Long is mathematically equivalent to the base model (the dynamic factor reduces to identity, Section 3.1); past 32K the bare base collapses (ppl of 71.00 / 104.66 / 79.37 at 128K for 1.7B / 4B / 8B) while Jet-Long stays at 11.41 / 9.62 / 8.51, the lowest among all extrapolation methods.

We also report the per-task accuracy at context length L=65536 in Table 3. Jet-Long is best or tied for best on 8 of the 13 tasks at both 1.7B and 8B, with the largest leads on Multi-Key NIAH (MK-NIAH-2 at 1.7B: 61.80 vs 20.00; MK-NIAH-3 at 8B: 70.80 vs 39.80) and Variable Tracking (VT at 1.7B: 73.56 vs 54.24); the only 1.7B exception is CWE, on which every method (including Base) scores below 1%. Aggregated over the 13 tasks, Jet-Long beats the strongest baseline by 7.31 pp at 1.7B and 6.07 pp at 8B.

Table 2 | PG-19 perplexity by context length across three Qwen3 base models. Anchored growing-window ppl on 100 PG-19 books, stride 1024. Base shown in gray as reference (collapses past its 32K native window). Best per column among extrapolation methods in **bold** (ties shared); the Avg column is the geometric mean across the seven lengths.

Method	4K	8K	16K	32K	64K	96K	128K	Avg
Qwen3-1.7B-Base	11.52	11.82	11.59	12.49	12.60	17.99	71.00	16.39
+ DNTK	**11.52**	**11.82**	**11.59**	**12.49**	12.48	14.75	16.31	12.89
+ YaRN	12.52	13.81	22.28	21.18	16.45	15.31	14.45	16.23
+ DCA	**11.52**	**11.82**	**11.59**	12.50	11.65	11.78	11.59	11.77
+ Self-Extend	11.55	11.91	11.74	12.69	11.74	11.83	11.51	11.85
+ Jet-Long (ours)	**11.52**	**11.82**	**11.59**	**12.49**	**11.53**	**11.65**	**11.41**	**11.71**
Qwen3-4B-Base	9.74	9.90	9.75	10.45	10.40	21.44	104.66	15.64
+ DNTK	**9.74**	**9.90**	**9.75**	**10.45**	11.43	12.88	13.93	11.05
+ YaRN	10.12	11.10	11.11	14.37	13.33	13.48	12.98	12.27
+ DCA	**9.74**	**9.90**	**9.75**	**10.45**	9.74	9.89	9.76	9.89
+ Self-Extend	9.76	9.97	9.87	10.62	9.87	9.94	9.63	9.95
+ Jet-Long (ours)	**9.74**	**9.90**	**9.75**	**10.45**	**9.67**	**9.81**	**9.62**	**9.85**
Qwen3-8B-Base	8.64	8.77	8.61	9.25	9.18	19.21	79.37	13.56
+ DNTK	**8.64**	**8.77**	**8.61**	**9.25**	9.17	10.22	10.50	9.28
+ YaRN	8.93	9.21	9.16	11.51	10.75	10.69	10.16	10.02
+ DCA	**8.64**	**8.77**	**8.61**	**9.25**	8.70	8.81	8.68	8.78
+ Self-Extend	8.66	8.83	8.70	9.38	8.77	8.80	8.55	8.81
+ Jet-Long (ours)	**8.64**	**8.77**	**8.61**	**9.25**	**8.63**	**8.71**	**8.51**	**8.73**

4.3. Hybrid attention extension results

To test generalization beyond softmax-only transformers, we apply Jet-Long to the hybrid Jet-Nemotron architecture [3], which interleaves softmax and linear-attention layers. Table 4 reports per-length RULER accuracy on Jet-Nemotron-2B and 4B.

Within 32K the construction reduces to base attention, so Jet-Long inherits the base model's in-distribution behavior (matching to within rounding). Past 32K the bare hybrid base collapses (8.54 / 5.65 at 128K for 2B / 4B), while Jet-Long retains 33.78 and 33.14 (+25.24 / +27.49 pp). Averaged over the seven lengths, Jet-Long lifts RULER from 42.93 to 52.94 (+10.01 pp) at 2B and from 42.16 to 53.47 (+11.31 pp) at 4B. The bifocal decomposition and dynamic factor generalize to hybrid LLM architectures.

4.4. Ablation: local window size w_0

To check whether w_0 requires per-deployment tuning, we sweep it on Qwen3-4B/8B at three out-of-window lengths (64K, 96K, 128K) in Table 5 (1.7B omitted for compute). The control w_0=0 shrinks the local window to attention-to-self only and collapses RULER to near-zero, confirming the local window is necessary. Past that boundary, Jet-Long is hyperparameter-resilient: every $w_0 \in \{512, 1024, 2048, 4096\}$ stays within 2 pp of the per-row best at any single length and within 1 pp on the per-model average; practitioners can pick w_0=2048 without per-deployment tuning. The slight degradation at w_0=8192 (1.4–2.1 pp gap) reflects the local window consuming a larger fraction of context.

4.5. Ablation: interpolate frequency or alias position

Section 3.1 requires only that the remote mapping $f(\cdot)$ keep rotation angles in-distribution, not how. Table 6 compares position aliasing ($f(x) = \lfloor x/G \rfloor$, the default Jet-Long) against YaRN-style frequency interpolation (rescaled RoPE frequencies, unchanged positions) on Qwen3-1.7B-Base. Aliasing wins by 6.99 pp at 64K and 4.30 pp at 128K, consistent with the LLM having learned a discrete grid of relative angles. The alias margin shrinks at extreme lengths because larger G gives continuous interpolation more to compensate for: on FWE from +20.3 pp at 64K to +2.5 at 128K, on QA-1 from +11.2 to +0.6, and frequency interpolation overtakes aliasing on MK-NIAH-2 and QA-2 at 128K. Aliasing remains the better default for $\leq$ 128K; hybrid mappings merit investigation at longer contexts.

Table 3 | RULER per-task accuracy at $L = 65536$ on Qwen3-1.7B-Base and Qwen3-8B-Base. Tasks: S1/S2/S3 = Single-NIAH, MK1/MK2/MK3 = Multi-Key NIAH, MV/MQ = Multi-Value/Query NIAH, VT = Variable Tracking, CWE/FWE = Common/Frequent-Word Extraction, QA1/QA2 = Question Answering. Best per column in **bold** (ties shared).

Method	S1	S2	S3	MK1	MK2	MK3	MV	MQ	VT	CWE	FWE	QA1	QA2	**Avg**
Qwen3-1.7B-Base														
Base	99.80	49.60	82.40	47.20	10.40	0.20	62.10	58.15	48.56	**0.58**	52.00	25.60	22.20	42.98
DNTK	**100.00**	93.00	99.40	78.40	16.40	**1.60**	81.60	81.05	54.24	0.38	70.93	41.20	26.80	57.31
YaRN	98.40	75.40	89.00	67.60	6.00	0.00	49.65	46.80	0.00	0.56	23.13	21.20	23.20	38.53
DCA	**100.00**	49.20	89.20	39.40	20.00	1.00	73.10	66.40	50.88	0.22	**75.93**	**44.60**	24.40	48.79
Self-Extend	**100.00**	98.00	99.60	74.20	5.80	0.60	**84.70**	**86.15**	29.68	0.18	70.80	31.60	27.00	54.49
Jet-Long	**100.00**	**100.00**	**100.00**	**93.20**	**61.80**	**1.60**	84.45	82.25	**73.56**	0.32	75.47	39.40	**28.00**	**64.62**
Qwen3-8B-Base														
Base	**100.00**	92.20	79.00	79.40	41.60	12.40	83.15	72.15	92.80	28.30	86.13	41.20	28.60	64.38
DNTK	**100.00**	99.80	99.60	91.80	78.80	28.60	93.10	93.90	98.88	16.84	81.67	54.80	**41.80**	75.35
YaRN	**100.00**	97.80	96.20	70.00	45.20	7.20	89.85	79.00	94.16	27.62	77.07	55.20	38.60	67.53
DCA	**100.00**	99.80	98.20	76.80	46.60	20.40	94.00	54.25	83.44	6.94	**87.73**	50.80	34.80	65.67
Self-Extend	**100.00**	99.80	99.80	87.40	57.80	39.80	96.55	**94.05**	**99.52**	41.44	81.60	**59.00**	39.60	76.64
Jet-Long	**100.00**	**100.00**	**100.00**	**94.20**	**92.20**	**70.80**	**96.60**	92.15	98.60	**47.32**	87.53	56.00	39.80	**82.71**

Table 4 | RULER accuracy by context length for Jet-Long applied to the hybrid Jet-Nemotron architecture [3], against the bare base model. The two configurations are mathematically identical within the 32K native window; the green tag reports the absolute accuracy-point gap of Jet-Long over Base for $L > 32$K.

Method	4K	8K	16K	32K	64K	96K	128K	*Avg*
Jet-Nemotron-2B	82.35	68.93	61.21	46.59	20.34	12.52	8.54	42.93
+ Jet-Long	82.35	68.93	61.21	46.58	40.33 (+19.99%)	37.40 (+24.88%)	33.78 (+25.24%)	52.94 (+10.01%)
Jet-Nemotron-4B	84.08	69.24	62.35	48.87	17.48	7.44	5.65	42.16
+ Jet-Long	84.08	69.24	62.35	48.88	39.93 (+22.45%)	36.69 (+29.25%)	33.14 (+27.49%)	53.47 (+11.31%)

4.6. Inference efficiency

We measure end-to-end throughput (tok/s) on a single H100 with CUDA graphs on Qwen3-8B-Base, comparing FA2 [4], the H100-native FA4 [34], the multi-launch Jet-Long (unfused) variant of Sections 3.3–3.2, and our fused Jet-Long CuTe kernel; 1.7B and 4B follow the same pattern (Table 8, Appendix B).

Within 32K all four configurations match FA2 to within ±1% since Jet-Long reduces to standard attention. Past 32K the naive multi-launch baseline pays the bifocal cost: three FlashAttention passes plus an out-of-kernel merge drop prefill to 0.89–0.93× FA2, and unfused per-token correction rotation drops generation to 0.14–0.34×. The fused CuTe kernel removes both overheads: prefill recovers to 1.28–1.39× FA2 past 32K (approaching FA4's 1.53× at 128K, without an H100-specific kernel), and generation stays $\geq 0.96\times$ FA2 at every length, with the residual $\leq 4\%$ overhead from the per-token correction rotation and the dynamic-G bookkeeping. The accuracy gains in Table 1 therefore come essentially free at inference.

5. Conclusion

We presented **Jet-Long**, a tuning-free zero-shot context-extension method that pairs a local RoPE-faithful window with a long-range window whose rescaling factor adapts dynamically to the current sequence length, and made the construction essentially free at inference through an inclusion–exclusion attention merge with on-the-fly correction rotation in a fused CuTe kernel. On Qwen3-1.7B/4B/8B at context lengths up to 128K, Jet-Long shows superior performance on RULER (+4.79/+2.18/+2.03 pp at 1.7B/4B/8B), HELMET-RAG, and PG-19 perplexity, generalizes to the hybrid linear-attention Jet-Nemotron backbone, and remains highly resilient to its only hyperparameter, the local window size w_0.

Jet-Long addresses position-OOD at the RoPE level, while the complementary attention-diffusion failure mode is naturally tackled by architectural alternatives [43, 44, 45, 46, 47]. Because Jet-Long requires a softmax-

Table 5 | Ablation on the local protected window size w_0. RULER avg accuracy at $L \in \{65536, 98304, 131072\}$ for Jet-Long applied to Qwen3-4B/8B-Base, sweeping w_0 from $\{0, 256, 512, 1024, 2048, 4096, 8192\}$. Best per row in **bold**; non-best cells show the accuracy-point gap from the per-row best in red.

Length	w_0=0	256	512	1024	2048	4096	8192
Qwen3-4B-Base + Jet-Long							
64K	4.10 (-72.55%)	**76.65**	76.63 (-0.02%)	76.49 (-0.16%)	76.57 (-0.09%)	75.91 (-0.74%)	74.69 (-1.96%)
96K	2.07 (-69.13%)	70.71 (-0.48%)	**71.19**	70.77 (-0.42%)	71.09 (-0.11%)	70.57 (-0.63%)	69.11 (-2.09%)
128K	1.08 (-64.55%)	64.13 (-1.50%)	**65.63**	65.57 (-0.05%)	64.74 (-0.89%)	64.06 (-1.56%)	63.52 (-2.11%)
Avg	2.42 (-68.74%)	70.50 (-0.65%)	**71.15**	70.95 (-0.21%)	70.80 (-0.35%)	70.18 (-0.97%)	69.11 (-2.04%)
Qwen3-8B-Base + Jet-Long							
64K	4.91 (-78.60%)	**83.51**	83.23 (-0.28%)	83.31 (-0.20%)	83.24 (-0.27%)	82.71 (-0.80%)	81.40 (-2.11%)
96K	2.54 (-72.96%)	74.29 (-1.20%)	74.94 (-0.56%)	75.20 (-0.29%)	**75.49**	74.84 (-0.65%)	73.82 (-1.68%)
128K	1.55 (-69.47%)	69.48 (-1.55%)	70.49 (-0.53%)	70.68 (-0.35%)	71.01 (-0.01%)	**71.03**	69.65 (-1.38%)
Avg	3.00 (-73.58%)	75.76 (-0.82%)	76.22 (-0.36%)	76.40 (-0.19%)	**76.58**	76.19 (-0.39%)	74.96 (-1.63%)

Table 6 | Ablation on the remote-window mapping for Qwen3-1.7B-Base: position aliasing (default Jet-Long) vs. YaRN-style frequency interpolation, reported per task at L=65536 and L=131072. Best per column within each length block in **bold** (ties shared); aliasing wins on the overall average at both lengths.

Variant	S1	S2	S3	MK1	MK2	MK3	MV	MQ	VT	CWE	FWE	QA1	QA2	Avg
L=65536														
Aliasing	**100.00**	**100.00**	**100.00**	**93.20**	**61.80**	**1.60**	**84.45**	**82.25**	**73.56**	0.32	**75.47**	**39.40**	**28.00**	**64.62**
Frequency	**100.00**	95.40	98.40	75.20	49.00	1.20	81.40	81.10	64.36	**0.78**	55.13	28.20	19.00	57.63
L=131072														
Aliasing	**100.00**	**98.60**	**100.00**	**91.40**	24.00	0.20	**75.90**	**76.30**	**70.12**	0.30	**82.40**	**33.40**	23.60	**59.71**
Frequency	**100.00**	84.40	97.80	72.60	**33.00**	**0.40**	69.80	66.50	56.32	**0.52**	79.93	32.80	**26.20**	55.41

Table 7 | Prefill and generation throughput (tok/s) on H100 for Qwen3-8B-Base. Parenthesized values are speedups against FA2 [4] at the matching length. FA4 [34] is omitted from the generation rows because no H100 generation kernel has been released for it.

Method	4K	8K	16K	32K	64K	96K	128K
Qwen3-8B-Base (Prefill)							
FA2 (baseline)	31211 (1.00×)	28091 (1.00×)	23652 (1.00×)	17796 (1.00×)	12238 (1.00×)	9332 (1.00×)	7433 (1.00×)
FA4	33455 (1.07×)	30831 (1.10×)	27321 (1.16×)	22358 (1.26×)	16690 (1.36×)	13693 (1.47×)	11400 (1.53×)
Jet-Long (unfused)	31242 (1.00×)	28116 (1.00×)	23602 (1.00×)	17810 (1.00×)	10909 (0.89×)	8548 (0.92×)	6932 (0.93×)
Jet-Long CuTe	31225 (1.00×)	28080 (1.00×)	23552 (1.00×)	17837 (1.00×)	15605 (1.28×)	12465 (1.34×)	10339 (1.39×)
Qwen3-8B-Base (Generation)							
FA2 (baseline)	105.31 (1.00×)	103.18 (1.00×)	99.20 (1.00×)	84.83 (1.00×)	74.80 (1.00×)	67.01 (1.00×)	60.16 (1.00×)
Jet-Long (unfused)	30.69 (0.29×)	30.50 (0.30×)	30.53 (0.31×)	28.89 (0.34×)	14.20 (0.19×)	10.49 (0.16×)	8.32 (0.14×)
Jet-Long CuTe	105.29 (1.00×)	103.14 (1.00×)	99.15 (1.00×)	84.84 (1.00×)	73.90 (0.99×)	65.00 (0.97×)	58.03 (0.97×)

with-RoPE base, natural extensions include other softmax-with-RoPE variants such as Multi-head Latent Attention [26] and sparse attention, and architectures interleaving softmax with sparse or linear-attention sub-layers, beyond the Jet-Nemotron experiments in Section 4.3.

References

[1] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

[2] Howard Yen, Tianyu Gao, Minmin Hou, Ke Ding, Daniel Fleischer, Peter Izsak, Moshe Wasserblat, and Danqi Chen. HELMET: How to evaluate long-context language models effectively and thoroughly. *arXiv preprint arXiv:2410.02694*, 2024.

[3] Yuxian Gu, Qinghao Hu, Shang Yang, Haocheng Xi, Junyu Chen, Song Han, and Han Cai. Jet-Nemotron: Efficient language model with post neural architecture search. *arXiv preprint arXiv:2508.15884*, 2025.

[4] Tri Dao. FlashAttention-2: Faster attention with better parallelism and work partitioning. *arXiv preprint arXiv:2307.08691*, 2023.

[5] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. LLaMA 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[6] Aixin Liu, Aoxue Mei, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, et al. DeepSeek-v3.2: Pushing the frontier of open large language models. *arXiv preprint arXiv:2512.02556*, 2025.

[7] Jiaheng Liu, Dawei Zhu, Zhiqi Bai, Yancheng He, Huanxuan Liao, Haoran Que, Zekun Wang, Chenchen Zhang, Ge Zhang, Jiebin Zhang, et al. A comprehensive survey on long context language modeling. *arXiv preprint arXiv:2503.17407*, 2025.

[8] Xiaoran Liu, Ruixiao Li, Mianqiu Huang, Zhigeng Liu, Yuerong Song, Qipeng Guo, Siyang He, Qiqi Wang, Linlin Li, Qun Liu, et al. Thus spake long-context large language model. *arXiv preprint arXiv:2502.17129*, 2025.

[9] Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, Yang Zhang, and Boris Ginsburg. RULER: What's the real context size of your long-context language models? *arXiv preprint arXiv:2404.06654*, 2024.

[10] Carlos E Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. SWE-bench: Can language models resolve real-world GitHub issues? *arXiv preprint arXiv:2310.06770*, 2023.

[11] Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yixin Dai, Jiawei Sun, Haofen Wang, Haofen Wang, et al. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*, 2(1):32, 2023.

[12] Tan Yu, Anbang Xu, and Rama Akkiraju. In defense of RAG in the era of long-context language models. *arXiv preprint arXiv:2409.01666*, 2024.

[13] Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, et al. A survey on large language model based autonomous agents. *Frontiers of Computer Science*, 18(6):186345, 2024.

[14] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*, 2022.

[15] Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. *Advances in neural information processing systems*, 36:68539–68551, 2023.

[16] Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, et al. AgentBench: Evaluating LLMs as agents. *arXiv preprint arXiv:2308.03688*, 2023.

[17] John Yang, Carlos E Jimenez, Alexander Wettig, Kilian Lieret, Shunyu Yao, Karthik Narasimhan, and Ofir Press. SWE-agent: Agent-computer interfaces enable automated software engineering. *Advances in Neural Information Processing Systems*, 37:50528–50652, 2024.

[18] Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiawu Zheng, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, et al. MetaGPT: Meta programming for a multi-agent collaborative framework. In *The Twelfth International Conference on Learning Representations*, 2024.

[19] Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. FlashAttention: Fast and memory-efficient exact attention with IO-awareness. *Advances in neural information processing systems*, 35:16344–16359, 2022.

[20] Hao Liu, Matei Zaharia, and Pieter Abbeel. Ring attention with blockwise transformers for near-infinite context. *arXiv preprint arXiv:2310.01889*, 2023.

[21] Chenxin An, Jun Zhang, Ming Zhong, Lei Li, Shansan Gong, Yao Luo, Jingjing Xu, and Lingpeng Kong. Why does the effective context length of LLMs fall short? *arXiv preprint arXiv:2410.18745*, 2024.

[22] Yao Fu, Rameswar Panda, Xinyao Niu, Xiang Yue, Hannaneh Hajishirzi, Yoon Kim, and Hao Peng. Data engineering for scaling language models to 128k context. *arXiv preprint arXiv:2402.10171*, 2024.

[23] Ofir Press, Noah A Smith, and Mike Lewis. Train short, test long: Attention with linear biases enables input length extrapolation. *arXiv preprint arXiv:2108.12409*, 2021.

[24] Yiran Ding, Li Lyna Zhang, Chengruidong Zhang, Yuanyuan Xu, Ning Shang, Jiahang Xu, Fan Yang, and Mao Yang. LongRoPE: Extending LLM context window beyond 2 million tokens. *arXiv preprint arXiv:2402.13753*, 2024.

[25] Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K Arora, Yu Bai, Bowen Baker, Haiming Bao, et al. gpt-oss-120b & gpt-oss-20b model card. *arXiv preprint arXiv:2508.10925*, 2025.

[26] Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. DeepSeek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024.

[27] Shouyuan Chen, Sherman Wong, Liangjian Chen, and Yuandong Tian. Extending context window of large language models via positional interpolation. *arXiv preprint arXiv:2306.15595*, 2023.

[28] Xiaoran Liu, Hang Yan, Shuo Zhang, Chenxin An, Xipeng Qiu, and Dahua Lin. Scaling laws of RoPE-based extrapolation. *arXiv preprint arXiv:2310.05209*, 2023.

[29] bloc97. NTK-aware scaled RoPE allows LLaMA models to have extended (8k+) context size without any fine-tuning and minimal perplexity degradation. Reddit post on r/LocalLLaMA, 2023. https://www.reddit.com/r/LocalLLaMA/comments/14lz7j5/ntkaware_scaled_rope_allows_llama_models_to_have/.

[30] emozilla. Dynamically scaled RoPE further increases performance of long context LLaMA with zero fine-tuning. Reddit post on r/LocalLLaMA, 2023. https://www.reddit.com/r/LocalLLaMA/comments/14mrgpr/dynamically_scaled_rope_further_increases/.

[31] Bowen Peng, Jeffrey Quesnelle, Honglu Fan, and Enrico Shippole. YaRN: Efficient context window extension of large language models. *arXiv preprint arXiv:2309.00071*, 2023.

[32] Hongye Jin, Xiaotian Han, Jingfeng Yang, Zhimeng Jiang, Zirui Liu, Chia-Yuan Chang, Huiyuan Chen, and Xia Hu. LLM maybe LongLM: Self-extend LLM context window without tuning. *arXiv preprint arXiv:2401.01325*, 2024.

[33] Chenxin An, Fei Huang, Jun Zhang, Shansan Gong, Xipeng Qiu, Chang Zhou, and Lingpeng Kong. Training-free long-context scaling of large language models. *arXiv preprint arXiv:2402.17463*, 2024.

[34] Ted Zadouri, Markus Hoehnerbach, Jay Shah, Timmy Liu, Vijay Thakkar, and Tri Dao. FlashAttention-4: Algorithm and kernel pipelining co-design for asymmetric hardware scaling. *arXiv preprint arXiv:2603.05451*, 2026.

[35] Jack W Rae, Anna Potapenko, Siddhant M Jayakumar, and Timothy P Lillicrap. Compressive transformers for long-range sequence modelling. *arXiv preprint arXiv:1911.05507*, 2019.

[36] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. RoFormer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

[37] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020.

[38] Kan Wu, Houwen Peng, Minghao Chen, Jianlong Fu, and Hongyang Chao. Rethinking and improving relative position encoding for vision transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10033–10041, 2021.

[39] Chi Han, Qifan Wang, Hao Peng, Wenhan Xiong, Yu Chen, Heng Ji, and Sinong Wang. LM-Infinite: Zero-shot extreme length generalization for large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 3991–4008, 2024.

[40] Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. *Transactions of the association for computational linguistics*, 12:157–173, 2024.

[41] Yan Li, Tianyi Zhang, Zechuan Li, and Soyeon Caren Han. A training-free length extrapolation approach for LLMs: Greedy attention logit interpolation (GALI). *arXiv preprint arXiv:2502.02659*, 2025.

[42] Jianlin Su. Rectified rotary position embeddings. https://github.com/bojone/rerope, 2023.

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

[44] Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. BigBird: Transformers for longer sequences. *Advances in neural information processing systems*, 33:17283–17297, 2020.

[45] Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are RNNs: Fast autoregressive transformers with linear attention. In *International conference on machine learning*, pages 5156–5165. PMLR, 2020.

[46] Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, et al. Rethinking attention with performers. *arXiv preprint arXiv:2009.14794*, 2020.

[47] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.

[48] Amirhossein Kazemnejad, Inkit Padhi, Karthikeyan Natesan Ramamurthy, Payel Das, and Siva Reddy. The impact of positional encoding on length generalization in transformers. *Advances in Neural Information Processing Systems*, 36:24892–24928, 2023.

[49] Kimi Team, Yifan Bai, Yiping Bao, Y Charles, Cheng Chen, Guanduo Chen, Haiting Chen, Huarong Chen, Jiahao Chen, Ningxin Chen, et al. Kimi K2: Open agentic intelligence. *arXiv preprint arXiv:2507.20534*, 2025.

[50] Aarti Basant, Abhijit Khairnar, Abhijit Paithankar, Abhinav Khattar, Adithya Renduchintala, Aditya Malte, Akhiad Bercovich, Akshay Hazare, Alejandra Rico, Aleksander Ficek, et al. NVIDIA Nemotron Nano 2: An accurate and efficient hybrid Mamba-transformer reasoning model. *arXiv preprint arXiv:2508.14444*, 2025.

A. Metric definitions

Percentage points (pp). For two scores $a_1, a_2 \in [0, 100]$ reported as percentages, the absolute gap in percentage points is

$$\Delta_{\text{pp}} = a_1 - a_2, \tag{8}$$

which we use throughout the paper to describe RULER and HELMET-RAG accuracy differences. This is distinct from the relative percent change $(a_1 - a_2)/a_2$.

Perplexity (ppl). For a held-out token sequence $x_1, \ldots, x_T$ scored under a language model with conditional probabilities $p(x_t \mid x_{<t})$, perplexity is the exponentiated mean negative log-likelihood per token,

$$\text{ppl} = \exp\left(-\frac{1}{T}\sum_{t=1}^{T} \log p(x_t \mid x_{<t})\right); \tag{9}$$

lower is better. On PG-19 we use anchored growing-window evaluation: at each context length L in $\{4\text{K}, 8\text{K}, 16\text{K}, 32\text{K}, 64\text{K}, 96\text{K}, 128\text{K}\}$, the next 1024 tokens are scored conditional on the preceding L tokens, and ppl is averaged across the 100 books in the PG-19 evaluation split.

Geometric-mean aggregation across lengths. For the Avg column of PG-19 perplexity tables, we report the geometric mean across the $n = 7$ lengths,

$$\overline{\text{ppl}} = \left(\prod_{i=1}^{n} \text{ppl}_i\right)^{1/n} = \exp\left(\frac{1}{n}\sum_{i=1}^{n} \log \text{ppl}_i\right), \tag{10}$$

which corresponds to averaging the underlying log-likelihoods uniformly across lengths and is therefore the natural aggregate for an exponentiated metric.

B. Full inference efficiency results

Table 8 reports the complete prefill and generation throughput sweep for all three Qwen3 sizes (1.7B, 4B, and 8B) on H100, complementing the 8B-only Table 7 in Section 4.6. The same four configurations are compared (FA2, FA4, Jet-Long (unfused), Jet-Long CuTe), and the parenthesized speedups are again computed against FA2 at the matching length. The qualitative pattern is identical across sizes: Jet-Long CuTe matches FA2 inside the 32K native window, recovers and surpasses FA2 at long-context prefill (1.28–1.45×), and stays at near-FA2 generation throughput everywhere ($\geq 0.96\times$ across all lengths and sizes), while the unfused multi-launch variant pays a real cost in generation that the fused kernel eliminates.

Table 8 | Full prefill and generation throughput (tok/s) on H100 across Qwen3-1.7B/4B/8B. Same setup as Table 7; parenthesized values are speedups against FA2 at the matching length. FA4 is omitted from the generation rows because no H100 generation kernel has been released for it.

Method	4K	8K	16K	32K	64K	96K	128K
Qwen3-1.7B-Base (Prefill)							
FA2 (baseline)	96354 (1.00×)	84446 (1.00×)	71654 (1.00×)	54123 (1.00×)	34910 (1.00×)	25490 (1.00×)	19956 (1.00×)
FA4	107892 (1.12×)	98257 (1.16×)	88780 (1.24×)	71660 (1.32×)	50100 (1.44×)	39089 (1.53×)	31845 (1.60×)
Jet-Long (unfused)	94977 (0.99×)	84205 (1.00×)	71501 (1.00×)	54171 (1.00×)	30588 (0.88×)	23148 (0.91×)	18560 (0.93×)
Jet-Long CuTe	94864 (0.98×)	84233 (1.00×)	71595 (1.00×)	54051 (1.00×)	45554 (1.30×)	35125 (1.38×)	28353 (1.42×)
Qwen3-4B-Base (Prefill)							
FA2 (baseline)	43384 (1.00×)	38737 (1.00×)	31661 (1.00×)	22798 (1.00×)	14449 (1.00×)	10397 (1.00×)	8053 (1.00×)
FA4	48086 (1.11×)	44725 (1.15×)	39017 (1.23×)	30107 (1.32×)	20927 (1.45×)	16019 (1.54×)	12997 (1.61×)
Jet-Long (unfused)	43573 (1.00×)	38778 (1.00×)	31679 (1.00×)	22852 (1.00×)	12647 (0.88×)	9429 (0.91×)	7506 (0.93×)
Jet-Long CuTe	43454 (1.00×)	38683 (1.00×)	31544 (1.00×)	22773 (1.00×)	19193 (1.33×)	14423 (1.39×)	11640 (1.45×)
Qwen3-8B-Base (Prefill)							
FA2 (baseline)	31211 (1.00×)	28091 (1.00×)	23652 (1.00×)	17796 (1.00×)	12238 (1.00×)	9332 (1.00×)	7433 (1.00×)
FA4	33455 (1.07×)	30831 (1.10×)	27321 (1.16×)	22358 (1.26×)	16690 (1.36×)	13693 (1.47×)	11400 (1.53×)
Jet-Long (unfused)	31242 (1.00×)	28116 (1.00×)	23602 (1.00×)	17810 (1.00×)	10909 (0.89×)	8548 (0.92×)	6932 (0.93×)
Jet-Long CuTe	31225 (1.00×)	28080 (1.00×)	23552 (1.00×)	17837 (1.00×)	15605 (1.28×)	12465 (1.34×)	10339 (1.39×)
Qwen3-1.7B-Base (Generation)							
FA2 (baseline)	224.31 (1.00×)	216.64 (1.00×)	200.36 (1.00×)	161.76 (1.00×)	134.74 (1.00×)	115.73 (1.00×)	100.54 (1.00×)
Jet-Long (unfused)	40.28 (0.18×)	40.68 (0.19×)	40.49 (0.20×)	38.32 (0.24×)	19.36 (0.14×)	14.08 (0.12×)	11.06 (0.11×)
Jet-Long CuTe	224.72 (1.00×)	216.94 (1.00×)	200.92 (1.00×)	161.88 (1.00×)	133.95 (0.99×)	112.12 (0.97×)	96.62 (0.96×)
Qwen3-4B-Base (Generation)							
FA2 (baseline)	141.01 (1.00×)	137.12 (1.00×)	130.63 (1.00×)	107.08 (1.00×)	91.57 (1.00×)	80.08 (1.00×)	70.72 (1.00×)
Jet-Long (unfused)	30.81 (0.22×)	31.10 (0.23×)	31.49 (0.24×)	29.36 (0.27×)	14.37 (0.16×)	10.78 (0.13×)	8.49 (0.12×)
Jet-Long CuTe	141.09 (1.00×)	137.20 (1.00×)	130.69 (1.00×)	107.09 (1.00×)	90.08 (0.98×)	77.22 (0.96×)	67.59 (0.96×)
Qwen3-8B-Base (Generation)							
FA2 (baseline)	105.31 (1.00×)	103.18 (1.00×)	99.20 (1.00×)	84.83 (1.00×)	74.80 (1.00×)	67.01 (1.00×)	60.16 (1.00×)
Jet-Long (unfused)	30.69 (0.29×)	30.50 (0.30×)	30.53 (0.31×)	28.89 (0.34×)	14.20 (0.19×)	10.49 (0.16×)	8.32 (0.14×)
Jet-Long CuTe	105.29 (1.00×)	103.14 (1.00×)	99.15 (1.00×)	84.84 (1.00×)	73.90 (0.99×)	65.00 (0.97×)	58.03 (0.97×)

C. Baseline configurations

The four zero-shot baselines compared in Tables 1–4 use a single configuration each, held constant across Qwen3-1.7B/4B/8B and across all evaluation lengths (4K–128K), as outlined in Table 9. All Qwen3 models have a pretrained context window of 32,768 tokens and a RoPE base $\theta = 10^6$.

Table 9 | Baseline hyperparameters used for the comparisons in Section 4.1. Held constant across Qwen3-1.7B/4B/8B and across evaluation lengths.

Method	Configuration
DNTK [30]	HuggingFace `rope_type=dynamic`, `factor`= 4.0, `original_max_position_embeddings`= 32,768; the NTK base β is computed at runtime from the scaling factor over RoPE base $\theta = 10^6$
YaRN [31]	HuggingFace `rope_type=yarn`, `factor`= 4.0, `max_position_embeddings`= 131,072, `original_max_position_embeddings`= 32,768 (target context 32K × 4 = 128K)
DCA [33]	`chunk_size`= 20,480, `local_window`= 4,096
Self-Extend [32]	`group_size`= 8, `window_size`= 1,024, `scale_base`= −1