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

Haozhan Tang, Zerui Wang, Yuxian Gu, Song Han, Han Cai
NVIDIA
hcai@nvidia.com
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. The dominant zero-shot methods (YaRN, Self-Extend, DCA) [35, 22, 2] fix a single rescaling factor up front, so an aggressive factor sacrifices short-context fidelity while a conservative one breaks down at long contexts; recent length-aware variants [48, 57] adapt the mapping, but with a fitted or distance-dependent schedule. 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 via a parameter-free analytic schedule, recovering the base model exactly at short inputs while extrapolating cleanly at long ones. An inclusion–exclusion attention merge and on-the-fly RoPE correction enable a fused CuTe implementation. On H100 at 64K–128K, prefill retains 83–88% of FlashAttention-3 throughput across the evaluated Qwen3 sizes and 88–93% of a matched CuTe control; Qwen3-8B single-batch generation reaches 1.04–1.08× FlashAttention-3 throughput. On Qwen3-1.7B/4B/8B [49] 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 [53]) and attains the lowest PG-19 perplexity. Additional evaluations cover Meta-Llama-3-8B, post-trained Qwen3 checkpoints, and the hybrid Jet-Nemotron architecture [17], supporting broader applicability without retraining. The local-window hyperparameter remains robust across the tested settings.

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 [45, 27, 29, 33, 20, 21, 15, 54, 46]. The pressure is most severe in agentic LLMs that interleave reasoning, planning, and tool use across many turns [52, 40, 31], and in coding agents operating over real software repositories [50, 19], where source code, execution traces, and tool outputs routinely accumulate to 100K+ tokens per task.

Training directly at long context remains expensive: efficient kernels like FlashAttention [11] and Ring Attention [28] 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 [3, 14]. Many models are therefore pretrained at a moderate window (4K–32K tokens) and expected to handle

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

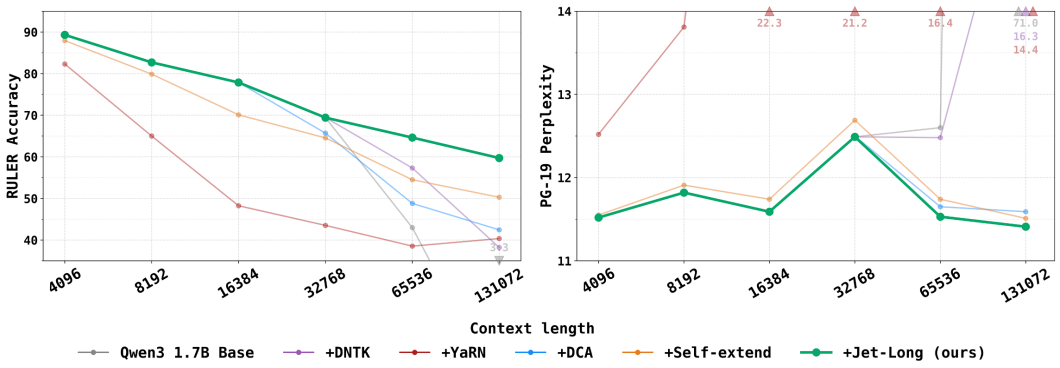


Figure 1: Qwen3-1.7B-Base RULER accuracy, averaged over all 13 tasks, and PG-19 perplexity at the six main context lengths. Jet-Long improves extended-context performance while preserving the base attention rule within the native window. The revised H100 efficiency comparison, with FA3 as the primary reference and a matched CuTe control, is reported in Section 4.8.

longer inputs at inference, a setting known as *context extension* [36, 12]. *Zero-shot context extension* (without fine-tuning) has become the dominant deployment mode for open-weight LLMs [49, 1, 26], since a single released checkpoint must support arbitrary downstream context lengths.

Vanilla Transformer-based LLMs fail to generalize beyond the training window [36, 7, 32], 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 [6] / Dynamic NTK [13], YaRN [35], Self-Extend [22], DCA [2]) 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 the fixed configurations of YaRN, Self-Extend, and DCA, and unlike length-adaptive maps with fitted or distance-dependent schedules [48, 57], Jet-Long derives the factor analytically from the current sequence length and the pretrained window, with no per-model fitting. Cache correction and kernel fusion make the adaptive mapping practical at inference.

Our contributions are:

- **Jet-Long**, a tuning-free bifocal context-extension method that sets the remote group size to the minimal integer compression $G = \max(1, \lceil L/w_{\text{pretrained}} \rceil)$ at the current sequence length L, keeping remote relative rotations on the pretrained grid without per-model fitting and mathematically recovering base attention within the native context.
- An inclusion–exclusion attention merge with on-the-fly RoPE correction that avoids rewriting cached keys. CuTe fusion reaches 0.83–$0.88\times$ FlashAttention-3 [41] prefill throughput on H100 at 64K–128K across three model sizes, and 0.88–$0.93\times$ a matched CuTe control without bifocal attention; 8B decode reaches 1.04–$1.08\times$ FA3. These measurements separate routing cost from improvements in the underlying attention implementation.
- Evaluation on Qwen3-1.7B/4B/8B-Base up to 128K: Jet-Long leads the strongest compared zero-shot baseline on RULER [20] by $+4.79/+2.18/+2.03$ pp, has the best aggregate HELMET-RAG [53] score, and the lowest PG-19 [38] perplexity at each size. Additional three-seed evaluations test transfer to Meta-Llama-3-8B and post-trained Qwen3; Jet-Nemotron [17] demonstrates applicability to hybrid attention.

2 Related work

2.1 Why RoPE-based LLMs fail to extrapolate

Most modern open-weight LLMs use Rotary Position Embedding (RoPE) [43], which applies per-position rotations across geometrically spaced frequencies so attention depends only on relative

position; earlier relative-position schemes such as ALiBi [36], T5's relative-position bias [39], and iRPE [47] 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 [7, 35, 32].

(ii) Attention diffusion and positional bias. As the key set grows, the softmax distribution flattens, dispersing probability mass over irrelevant tokens [18, 35]; separately, models exhibit a U-shaped positional attention bias that under-attends to middle-context information, degrading retrieval accuracy for centrally placed evidence [30].

These motivate two complementary zero-shot axes: interpolating RoPE frequencies or position indices to keep rotation angles in-distribution [7, 6, 13, 35, 22, 2, 25, 42], and attention penalties or temperature scaling to counteract softmax diffusion [35, 18, 25]. 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) [7] linearly compresses positions into the pretrained range. NTK-aware scaled RoPE [6] increases the RoPE base to preserve high-frequency components while interpolating low-frequency ones; Dynamic NTK (DNTK) [13] makes that base a function of the current sequence length so the scaling adapts at decode time. YaRN [35] 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 [25] interpolates at the attention-logit level rather than the embedding.

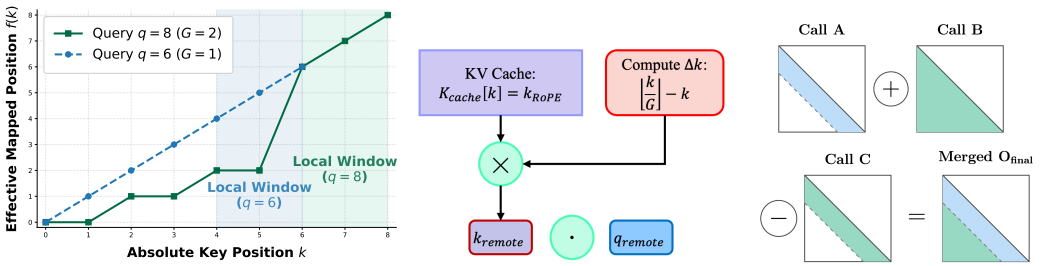
Grouped / chunked-position methods. A second line reuses only in-distribution position indices. Self-Extend [22] pairs a neighbor window with a grouped window in which blocks of tokens share a single position index. Dual Chunk Attention (DCA) [2] partitions the sequence into chunks and uses asymmetric query/key indices in the cross-chunk component so all relative distances stay within the pretrained range. LM-Infinite [18] earlier introduced a Λ-shaped mask retaining an attention sink and a recent window.

Length-adaptive remapping. Closest to our setting, several training-free methods also condition the position map on input length. AdaGroPE [48] grows the reuse count of relative positions with distance; LaMPE [57] applies a piecewise map over three sequence regions whose interior compression ratio is a per-model fitted sigmoid of the input length; SELF [9] grows Self-Extend's group size logistically. LaMPE's Appendix E also reports additional latency and memory use relative to FlashAttention on A800, motivating attention to execution cost; this is a separate hardware setup from our matched H100 study.

Jet-Long sits in the grouped-position family, with the group size derived from the current sequence length rather than fixed or fitted: identity within the native window, and past it the minimal integer G that keeps remote relative rotations on the pretrained grid (Section 3.1). It therefore reduces mathematically to base attention for $L \leq w_{\text{pretrained}}$, and its correction rotation (Section 3.2) preserves a base-position key cache when G changes mid-generation, without recomputing historical hidden states.

2.3 Long-context training, architectures, and benchmarks

Training-based approaches extend the window via continued pretraining [12] but face quadratic FLOP costs (only partially mitigated by efficient kernels [11, 28]) and scarce long-context data with short-context regression risk [3, 14], motivating the zero-shot setting. Our evaluation does not include a matched comparison with training-based methods such as LongRoPE; comparisons reported for other backbones and protocols do not establish their performance relative to Jet-Long (Appendix A).



(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 use $f(x) = \lfloor x/G \rfloor$ with $G = \max(1, \lceil L/w_{\text{pretrained}} \rceil)$, adapting to context length L. **(b)** Cached keys retain original RoPE coordinates. Correction rotations (Δq on the active query, Δk on each cached key) recover remote phases without rewriting stored keys or recomputing historical hidden states. **(c)** Prefill combines three attention computations (full remote, local-only-with-RoPE, local-only-with-remap) by inclusion–exclusion, using LogSumExp statistics; CuTe fusion reduces execution overhead.

Alternative architectures replace dense softmax with sparse attention [5, 56], linear or kernel-based variants [23, 8], or state-space models [16]; 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 [24], and open-weight models such as Kimi K2 [44] and NVIDIA Nemotron Nano 2 [4] 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 4.5, Jet-Nemotron [17]).

We evaluate on RULER [20] (13 retrieval, tracking, extraction, and QA tasks), HELMET-RAG [53] (the HELMET study's best overall predictor of downstream long-context performance), and PG-19 [38] long-form perplexity, against four zero-shot baselines. Additional evaluations cover Meta-Llama-3-8B and post-trained Qwen3 (Appendices C and D), and applicability to the hybrid Jet-Nemotron backbone (Section 4.5).

3 Methodology

Fixed rescaling and grouping factors force a tradeoff between short-context fidelity and long-context reach; recent length-adaptive maps [48, 57] relax it, but with a fitted or distance-dependent schedule. Jet-Long instead derives the factor analytically from the current sequence length (Section 3.1) and couples the two-window computation with cache correction and fused attention (Sections 3.2–3.3). Section 4.8 measures the resulting inference cost against FlashAttention-3 and a matched CuTe control.

We build on the dual-window (bifocal) decomposition of Self-Extend [22] (a related but architecturally three-way ancestor is DCA [2]): a local window of size w_0 that retains classic RoPE, and a remote window governed by a rewritten position mapping $f(\cdot)$ that maps positions back into the training range. Grouping shares only positional coordinates: every token retains a separate key and value, with no token merging or removal. For a query at position q and a key at position k, the pre-softmax attention score is

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

Let L denote the current sequence length and $w_{\text{pretrained}}$ the pretrained context window. Fresh prefill uses one G determined by the full prompt length; decoding updates L to include the cached prefix and current token. When $L \le w_{\text{pretrained}}$, $f(x) = x$ and Eq. (1) reduces to the unmodified base attention. Jet-Long's contribution lies in the choice of $f(\cdot)$ and its cache-aware, fused realization.

3.1 Dynamic extrapolation factor

To keep remote-window relative rotation angles on the pretrained position grid, 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 [13] and dynamic-YaRN [35] variants), the scaling ratio $s = \max(1, L/w_{\text{pretrained}})$; for discrete grouped methods (e.g., Self-Extend [22]), it is an integer group size G. Jet-Long uses discrete grouping because pretraining uses a finite, discrete set of relative RoPE rotation angles; aliasing positions onto that grid retains integer relative positions within the training range. This is a statement about positional phases, not a guarantee that long-context hidden states follow their training distribution. We ablate this choice against continuous frequency interpolation in Section 4.7. Whereas DNTK [13] adapts the RoPE base β (per-frequency), Jet-Long adapts the discrete group size G on the position-aliasing axis.

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 integer compression that keeps every remote position within the pretrained range, maximizing positional resolution within this mapping family. LaMPE [57] likewise applies floor-based remapping in its middle region, but derives the middle-region compression ratio from an empirically fitted sigmoid of the input length, with model-specific fits presented for Llama2 and Llama3, whereas Eq. (2) is determined analytically by L and $w_{\text{pretrained}}$ with no fitted parameters.

3.2 Key-value cache management and correction rotation

Dynamic extrapolation poses a cache-management challenge: cached keys must be interpreted under the current G even when it changes during generation.

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 $\text{RoPE}(\cdot, \Delta)$ 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:

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

The correction reconstructs transient remote views without rewriting the stored cache. It exactly recovers the current RoPE phases for the cached key vectors, but does not recompute historical hidden states or values. Retained-cache decoding is therefore not equivalent to a full re-prefill of the prefix under the new G. Appendix E measures this approximation through a direct re-prefill comparison and a stratified analysis of group-size transitions.

3.3 Inclusion–exclusion prefill

To avoid materializing the full quadratic attention matrix during long-sequence prefill, Jet-Long combines inclusion–exclusion with FlashAttention's [11] LogSumExp statistics. The merge uses 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 in exact arithmetic: $W_B\mathbf{O}_B - W_C\mathbf{O}_C$ (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 accumulated in FP32) 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 algebraically exact merge implements one joint softmax over the local and remote windows without a dense boolean mask. It retains dense attention and its quadratic arithmetic cost; fusion reduces implementation overhead. Subtractive attention merges can suffer cancellation [51]. Here subtraction occurs in linear space, the shift by M bounds weights in $(0, 1]$ in exact arithmetic, and FP32 accumulation reduces roundoff, without providing an unconditional numerical-stability guarantee.

4 Experiments

4.1 Setup

Models. Our primary evaluation is on Qwen3-1.7B/4B/8B-Base [49], each with a 32,768-token (32K) native window, extended to 131,072 (128K) without fine-tuning. We also evaluate Meta-Llama-3-8B-Base [34], post-trained Qwen3-1.7B/8B with their chat templates, and the hybrid Jet-Nemotron-2B/4B models [17], which interleave softmax and linear-attention layers. The main Qwen3 comparisons use $w_0 = 2048$; baseline configurations and additional sensitivity tests are in Appendices I and F.

Long-context benchmarks. We report results on three complementary suites: (i) RULER [20], a synthetic recall stress test averaging over thirteen tasks; (ii) HELMET-RAG [53], 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 [38], 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 B. We further ablate the only Jet-Long hyperparameter, the local protected window size w_0, in Section 4.6.

Inference efficiency. We benchmark fused CuTe prefill and single-batch generation on H100, using FlashAttention-3 [41] as the primary hardware-appropriate reference, FA2 [10] as a secondary reference, and FA4 [55] as a complementary implementation. A matched CuTe control without bifocal attention separates the extension cost from the underlying kernel implementation.

Lengths and uncertainty. The main Base-model comparisons use six lengths: 4K, 8K, 16K, 32K, 64K, and 128K. Their additional 96K results are reported in the appendix. RULER uses 500 examples per task, except for the 256K–1M evaluation, which uses 100 (Section 4.4); the cross-family and post-trained evaluations use three independently generated datasets (seeds 42/44/45), reporting mean $\pm$ half-range. This measures dataset-sampling variation, not a confidence interval or training variance (Appendix B). Compute details and the frozen efficiency stack are in Appendices J and H.

Table 1: Long-context performance on RULER (accuracy averaged over 13 tasks and six lengths: 4K, 8K, 16K, 32K, 64K, 128K), HELMET-RAG, and PG-19 perplexity (geometric mean over the same six lengths, lower is better), on three Qwen3 base sizes. Each Avg column averages the three model-size scores. 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 [13]	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 [35]	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 [2]	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 [22]	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**

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 configuration in this table (DNTK at 1.7B; Self-Extend at 4B and 8B) is 4.79, 2.18, and 2.03 pp. Additional fixed-configuration sensitivity tests improve Self-Extend but preserve Jet-Long's lead at extended lengths (Appendix F).

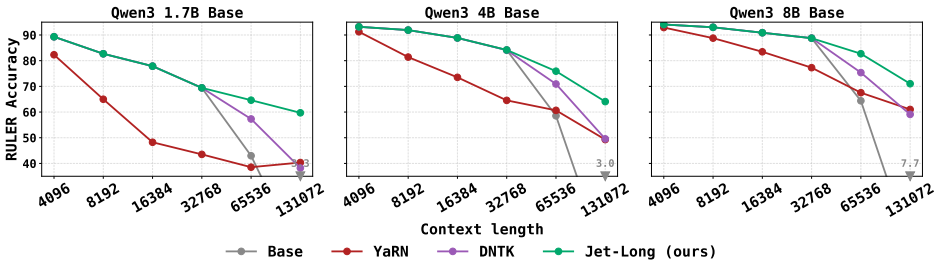


Figure 3: RULER accuracy as a function of context length, averaged over the 13 RULER tasks, on Qwen3-1.7B/4B/8B-Base. Jet-Long preserves the pretrained model's performance within the training window (32K) and outperforms YaRN and DNTK at all extended context lengths (the full comparison vs. DCA and Self-Extend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

The per-task diagnostics in Appendix G.2 show the largest gains on multi-key retrieval and variable tracking. Jet-Long does not lead every task: common-word extraction remains difficult for the 1.7B model. Full PG-19 results, including the additional 96K evaluation, are in Appendix G.1.

4.3 Cross-family and post-trained evaluation

Table 2 reports three-seed RULER results at 64K, 96K, and 128K. Meta-Llama-3-8B-Base has an 8K native window and RoPE base $\theta = 500{,}000$, testing transfer beyond Qwen3. Jet-Long trails Self-Extend by 0.71 pp at 64K, within the observed sampling spread, and leads by 4.50/11.99 pp at 96K/128K. On post-trained Qwen3-1.7B, Jet-Long leads at all three displayed lengths. On post-trained Qwen3-8B, DNTK leads at 64K, while Jet-Long has the highest mean at 96K and 128K. These prompted-generation results broaden checkpoint coverage; they do not replace direct instruction-following or agent evaluations. Full results from 4K through 128K and additional baseline configurations are in Appendices C and D.

Table 2: RULER accuracy: mean ± half-range over seeds 42/44/45, with 500 examples per task for all 13 tasks. Best mean per model and length in **bold**. † marks at least 10% empty generations in a seed; these scores include generation failures. Full seven-length results and baseline configurations are in Appendices C and D.

Length	Unextended	DNTK	YaRN	DCA	Self-Extend	Jet-Long
			Meta-Llama-3-8B-Base			
64K	0.00 ± 0.00	28.94 ± 0.64	31.87 ± 0.37	30.30 ± 0.40	$\mathbf{58.28} \pm 0.97$	57.57 ± 1.17
96K	0.00 ± 0.00	19.32 ± 0.11	26.15 ± 0.89	16.32 ± 0.63	47.86 ± 1.38	$\mathbf{52.36} \pm 1.64$
128K	0.00 ± 0.00	11.59 ± 0.28	12.21 ± 0.79	12.29 ± 0.25	31.86 ± 0.99	$\mathbf{43.85} \pm 0.67$
			Qwen3-1.7B, post-trained			
64K	42.89 ± 0.31	51.78 ± 0.35	$27.26 \pm 0.38^{\dagger}$	44.13 ± 0.44	$48.51 \pm 1.76^{\dagger}$	$\mathbf{54.88} \pm 1.23$
96K	13.15 ± 0.42	$36.28 \pm 0.32^{\dagger}$	$34.36 \pm 0.14^{\dagger}$	41.07 ± 0.65	48.30 ± 0.49	$\mathbf{50.50} \pm 0.89$
128K	2.37 ± 1.11	$27.70 \pm 1.04^{\dagger}$	$25.78 \pm 0.19^{\dagger}$	38.51 ± 1.03	$42.75 \pm 0.87^{\dagger}$	$\mathbf{47.99} \pm 0.90$
			Qwen3-8B, post-trained			
64K	61.22 ± 0.41	$\mathbf{77.63} \pm 0.73$	64.85 ± 0.46	64.57 ± 0.43	73.65 ± 0.79	74.91 ± 0.23
96K	19.70 ± 0.59	70.56 ± 0.70	61.54 ± 0.77	62.78 ± 0.23	71.81 ± 0.88	$\mathbf{72.94} \pm 0.36$
128K	3.00 ± 0.20	63.62 ± 0.64	56.92 ± 0.68	61.05 ± 0.46	68.54 ± 0.16	$\mathbf{68.88} \pm 0.21$

Table 3: RULER on Qwen3-30B-A3B-Instruct-2507: mean ± half-range over three dataset seeds, with 100 examples per task. The native context is 256K; 512K and 1M are 2× and 4× extensions. Best mean per length in **bold**.

Method	256K (native)	512K (2×)	1M (4×)
Unextended	74.71 ± 0.04	67.22 ± 0.32	26.82 ± 0.11
YaRN	62.20 ± 1.27	59.84 ± 0.75	57.71 ± 0.59
DNTK	74.91 ± 0.31	69.83 ± 0.27	52.99 ± 0.30
DCA	$\mathbf{75.36} \pm 0.92$	65.70 ± 0.14	56.90 ± 0.66
Jet-Long	74.72 ± 0.35	$\mathbf{71.99} \pm 0.56$	$\mathbf{64.03} \pm 0.97$

4.4 Extension from 256K to 1M

We extend Qwen3-30B-A3B-Instruct-2507 [37] from its native 256K context to 512K and 1M. Table 3 reports mean ± half-range over three independently generated dataset seeds, with 100 examples for each of the 13 RULER tasks per seed and length. Jet-Long achieves the highest mean at both extended lengths, exceeding DNTK by 2.16 pp at 512K and YaRN by 6.32 pp at 1M. The unextended model retains 67.22 at 512K and falls to 26.82 at 1M. Self-Extend is omitted because its two rotated key-cache views incur prohibitive memory or runtime costs in the standard implementation; a practical accuracy comparison requires a fused implementation.

4.5 Hybrid attention extension results

To test generalization beyond softmax-only transformers, we apply Jet-Long to the hybrid Jet-Nemotron architecture [17], which interleaves softmax and linear-attention layers. Table 4 reports per-length RULER accuracy on Jet-Nemotron-2B and 4B.

Within the native window, the construction reduces to base attention. At 128K, Jet-Long improves the hybrid models from 8.54 to 33.78 at 2B and from 5.65 to 33.14 at 4B ($+25.24/+27.49$ pp). The six-length averages improve by $7.54/8.33$ pp; Appendix G.3 includes 96K. This demonstrates architectural applicability to hybrid LLMs; the experiment compares against the unextended models and does not establish superiority over other hybrid extension methods.

4.6 Ablation: local window size w_0

Table 5 sweeps the protected local window on Qwen3-4B/8B at 64K and 128K; Appendix G.4 adds 96K. Setting $w_0 = 0$ leaves self-attention locally and grouped remote attention active, reducing RULER to 1.08–4.91 across the full sweep. Local protection is therefore essential in this construction. Every $w_0 \in \{512, 1024, 2048, 4096\}$ stays within 2 pp of the best at each tested length and within 1 pp on the three-length average, supporting the default $w_0 = 2048$. The tested components serve

Table 4: RULER accuracy over the six main context lengths for Jet-Long applied to the hybrid Jet-Nemotron architecture [17], 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	128K	*Avg*
Jet-Nemotron-2B	82.35	68.93	61.21	46.59	20.34	8.54	47.99
+ Jet-Long	82.35	68.93	61.21	46.58	40.33 (+19.99 pp)	33.78 (+25.24 pp)	55.53 (+7.54 pp)
Jet-Nemotron-4B	84.08	69.24	62.35	48.87	17.48	5.65	47.95
+ Jet-Long	84.08	69.24	62.35	48.88	39.93 (+22.45 pp)	33.14 (+27.49 pp)	56.27 (+8.33 pp)

Table 5: Local-window ablation on Qwen3-4B/8B-Base at 64K and 128K. Cells report RULER accuracy; red values give the gap in pp from the row's best score (**bold**). Appendix G.4 adds 96K and the three-length averages.

Length	w_0=0	256	512	1024	2048	4096	8192
Qwen3-4B-Base + Jet-Long							
64K	4.10 (-72.55 pp)	**76.65**	76.63 (-0.02 pp)	76.49 (-0.16 pp)	76.57 (-0.09 pp)	75.91 (-0.74 pp)	74.69 (-1.96 pp)
128K	1.08 (-64.55 pp)	64.13 (-1.50 pp)	**65.63**	65.57 (-0.05 pp)	64.74 (-0.89 pp)	64.06 (-1.56 pp)	63.52 (-2.11 pp)
Qwen3-8B-Base + Jet-Long							
64K	4.91 (-78.60 pp)	**83.51**	83.23 (-0.28 pp)	83.31 (-0.20 pp)	83.24 (-0.27 pp)	82.71 (-0.80 pp)	81.40 (-2.11 pp)
128K	1.55 (-69.47 pp)	69.48 (-1.55 pp)	70.49 (-0.53 pp)	70.68 (-0.35 pp)	71.01 (-0.01 pp)	**71.03**	69.65 (-1.38 pp)

distinct roles: local protection and remote mapping affect accuracy, while the fusion comparison in Section 4.8 isolates execution cost.

4.7 Ablation: interpolate frequency or alias position

Table 6 compares discrete position aliasing with YaRN-style frequency interpolation within the same bifocal construction on Qwen3-1.7B-Base. Aliasing leads by 6.99 pp at 64K and 4.30 pp at 128K. Our hypothesis is that aliasing's loss of within-group positional resolution explains some exceptions: tokens sharing a coordinate remain distinct content vectors, but their fine offsets cannot be distinguished by remote RoPE alone. At 128K, interpolation leads on MK-NIAH-2, MK-NIAH-3, CWE, and QA-2. This pattern motivates distance-adaptive combinations of interpolation and aliasing, but does not establish a causal explanation or validate a hybrid mapping. The mapping ablation itself is limited to 128K.

4.8 Inference efficiency

We measure end-to-end throughput on an H100 80GB with CUDA graphs, using a frozen software stack. We use Hopper-optimized FA3 [41] as the primary H100 reference, FA2 [10] as a secondary reference, and FA4 [55] as a complementary prefill comparison. The matched control, labeled "CuTe (without bifocal attention)," omits Jet-Long's local/remote attention rule for context extension. Comparing it with fused Jet-Long measures the cost of that rule on the same backend. At extended lengths, fused and unfused Jet-Long retain the same bifocal attention rule, so their comparison measures the speedup from fusion. Table 7 reports representative Qwen3-8B-Base measurements; Appendix H includes 96K results and the software configuration.

Across Qwen3-1.7B/4B/8B at 64K–128K, fused prefill reaches 0.83–$0.88\times$ FA3 throughput, 0.88–$0.93\times$ the matched CuTe control, and 1.43–$1.56\times$ the unfused implementation. Thus, the bifocal attention rule incurs a 7–12% prefill throughput cost relative to the matched control; the 1.28–$1.45\times$ gain over FA2 largely reflects the underlying CuTe implementation. Fusion substantially reduces the cost of bifocal attention, while a measurable prefill cost remains relative to FA3.

For Qwen3-8B at 64K–128K, fused decode reaches 1.04–$1.08\times$ FA3 and 0.994–$1.034\times$ the matched control, while remaining within 3.6% of FA2 throughput. At native lengths, $G = 1$ bypasses extension; switching this fallback from FA2 to FA3 yields 0.97–$1.00\times$ FA3 throughput across sizes. The extended-length branch is unchanged, and the measurements above use the same frozen stack throughout.

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: Qwen3-8B-Base throughput (tok/s) on H100. Parentheses give throughput relative to FA3 at the same length. Decode measurements are reported at extended lengths; dashes denote cells not included in this table. FA4 is compared for prefill only.

Method	4K	32K	64K	128K
Prefill				
FA2	31211 (0.92×)	17796 (0.78×)	12238 (0.69×)	7433 (0.61×)
FA3	33915 (1.00×)	22869 (1.00×)	17701 (1.00×)	12269 (1.00×)
FA4	33455 (0.99×)	22358 (0.98×)	16690 (0.94×)	11400 (0.93×)
CuTe (without bifocal attention)	33235 (0.98×)	22431 (0.98×)	16860 (0.95×)	11494 (0.94×)
Jet-Long (unfused)	31242 (0.92×)	17810 (0.78×)	10909 (0.62×)	6932 (0.57×)
Jet-Long CuTe	33830 (1.00×)	22759 (1.00×)	15612 (0.88×)	10382 (0.85×)
Decode				
FA2	—	—	74.80 (1.09×)	60.16 (1.07×)
FA3	—	—	68.33 (1.00×)	56.05 (1.00×)
CuTe (without bifocal attention)	—	—	71.48 (1.05×)	58.38 (1.04×)
Jet-Long (unfused)	—	—	14.20 (0.21×)	8.32 (0.15×)
Jet-Long CuTe	—	—	73.90 (1.08×)	58.03 (1.04×)

5 Conclusion

We presented **Jet-Long**, a tuning-free zero-shot context-extension method combining a protected local RoPE window, analytically determined remote grouping, and cache-aware fused execution. On Qwen3-1.7B/4B/8B-Base up to 128K, it improves RULER over the strongest compared baseline by 4.79/2.18/2.03 pp, achieves the best aggregate HELMET-RAG score, and the lowest PG-19 perplexity at each size. Three-seed evaluations further support transfer to Meta-Llama-3-8B and post-trained Qwen3, while Jet-Nemotron demonstrates applicability to hybrid attention. Fusion reaches 0.83–0.88× FA3 prefill throughput at extended lengths across the three Qwen3 sizes; a matched CuTe control isolates the remaining routing cost. Correction rotations avoid cache rewrites but retain historical hidden states, an approximation assessed in Appendix E.

Jet-Long addresses position-OOD at the RoPE level, while the complementary attention-diffusion failure mode is naturally tackled by architectural alternatives [5, 56, 23, 8, 16]. Because Jet-Long requires a softmax-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 4.5.

Acknowledgments and Disclosure of Funding

This work received no third-party funding. We declare no other competing interests.

References

[1] 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.

[2] 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.

[3] 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.

[4] 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.

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

[6] 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/.

[7] 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.

[8] 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.

[9] Phat Thanh Dang, Saahil Thoppay, Wang Yang, Qifan Wang, Vipin Chaudhary, and Xiaotian Han. SELF: Self-extend the context length with logistic growth function. *arXiv preprint arXiv:2505.17296*, 2025.

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

[11] 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.

[12] 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.

[13] 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/.

[14] 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.

[15] 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.

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

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] 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.

[22] 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.

[23] 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.

[24] 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.

[25] 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.

[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] 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.

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

[29] 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.

[30] 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.

[31] 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.

[32] 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.

[33] 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.

[34] Meta. Meta Llama 3 model card. https://huggingface.co/meta-llama/Meta-Llama-3-8B, 2024.

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

[36] 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.

[37] Qwen Team. Qwen3-30B-A3B-Instruct-2507 model card. https://huggingface.co/Qwen/Qwen3-30B-A3B-Instruct-2507, 2025.

[38] 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.

[39] 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.

[40] 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.

[41] Jay Shah, Ganesh Bikshandi, Ying Zhang, Vijay Thakkar, Pradeep Ramani, and Tri Dao. FlashAttention-3: Fast and accurate attention with asynchrony and low-precision. *arXiv preprint arXiv:2407.08608*, 2024.

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

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

[44] 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.

[45] 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.

[46] 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.

[47] 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.

[48] Xinhao Xu, Jiaxin Li, Hui Chen, Zijia Lin, Jungong Han, and Guiguang Ding. Extending LLM context window with adaptive grouped positional encoding: A training-free method. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 573–587, 2025.

[49] 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.

[50] 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.

[51] Hengshuai Yao, Xing Chen, Ahmed Murtadha, Jin Li, Yasin Abbasi Yadkori, Shuai Shao, Changling Liu, Guan Wang, Mingli Yuan, William Chen, and Sen Song. Why attend to everything? Focus is the key. *arXiv preprint arXiv:2604.03260*, 2026.

[52] 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.

[53] 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.

[54] 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.

[55] 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.

[56] 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.

[57] Sikui Zhang, Guangze Gao, Ziyun Gan, Chunfeng Yuan, Zefeng Lin, Houwen Peng, Bing Li, and Weiming Hu. LaMPE: Length-aware multi-grained positional encoding for adaptive long-context scaling without training. In *Findings of the Association for Computational Linguistics: ACL 2026*, pages 32134–32149, 2026.

A Limitations

Comparison scope. Our controlled comparisons focus on training-free extension. We do not provide a matched comparison with training-based systems such as LongRoPE [12], nor a direct comparison with GALI [25] or ReRoPE [42]. Results reported for related baselines on other backbones and protocols cannot establish Jet-Long's relative performance against these methods. The Jet-Nemotron experiment establishes applicability to interleaved softmax and linear attention through comparison with the unextended models; it does not establish superiority over competing extension methods on hybrid architectures.

Evaluation coverage. RULER, HELMET-RAG, and PG-19 cover diagnostic long-context tasks, retrieval-augmented question answering, and language modeling. The post-trained experiments use chat templates but still evaluate RULER; they do not establish broad instruction-following, graph-reasoning, or agentic performance. The additional model families and evaluation through 1M expand the tested range, but do not guarantee transfer to arbitrary models, lengths, or workloads. Reported seed variation reflects dataset sampling rather than training variance, and mean $\pm$ half-range is not a confidence interval.

Cache-history approximation. Correction rotations recover the phases required by the current group size for the cached key vectors. They do not recompute historical hidden states or values after G changes. Retained-cache decoding therefore differs from full re-prefill. Appendix E characterizes this difference at a tested boundary and in a stratified task analysis; these finite observations do not guarantee unchanged decisions at every transition.

Positional resolution and future adaptation. Discrete aliasing preserves separate token content but removes within-group positional distinctions from remote RoPE coordinates. Its aggregate advantage over frequency interpolation does not hold on every subtask, and the mapping comparison is limited to 64K and 128K. Distance-adaptive combinations of interpolation and aliasing, and long-context fine-tuning with Jet-Long, remain untested. Fine-tuning would also require evaluating short-context retention and generalization across group sizes.

Systems and extreme-length constraints. Jet-Long retains dense attention: fusion reduces implementation overhead but does not remove quadratic prefill computation or linear KV-cache storage. Throughput depends on hardware, kernels, and workload; our H100 comparisons include a measurable routing cost relative to a matched CuTe control. At 512K and 1M, the standard Self-Extend implementation could not be evaluated practically under our memory and runtime constraints (Section 4.4), so the extreme-length comparison covers a smaller set of methods.

B 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 each model, PG-19 perplexity is aggregated by the geometric mean across lengths: $n = 6$ for the main comparison (4K, 8K, 16K, 32K, 64K, 128K), and $n = 7$ for the appendix table that additionally includes 96K,

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

This averages log perplexity uniformly across lengths. The Avg columns spanning model sizes in Table 1 are arithmetic means of the three displayed model scores. The six-length main and seven-length appendix aggregates use different length sets and should not be equated.

Dataset-sampling variation. For the repeated RULER evaluations, let a_s be the mean task score for dataset seed s. We report the mean across seeds and the half-range,

$$\bar{a} = \frac{1}{m} \sum_{s=1}^{m} a_s, \qquad h = \frac{\max_s a_s - \min_s a_s}{2}, \qquad m = 3. \tag{11}$$

The notation $\bar{a} \pm h$ describes variation across independently generated evaluation datasets. It is not a standard deviation, standard error, confidence interval, or estimate of training variance. The cross-family and post-trained comparisons use seeds 42/44/45; the extreme-length evaluation also uses three dataset seeds. The main Base-model tables and the additional single-seed sensitivity tests do not carry these error bars.

C Cross-family validation

We evaluate Meta-Llama-3-8B-Base, whose native context window is 8K and RoPE base is $\theta = 500{,}000$, complementing the Qwen3 experiments. Table 8 reports all seven lengths from 4K to 128K, with means and half-ranges over three independently generated RULER datasets (seeds 42, 44, and 45), using 500 examples for each of the 13 tasks per seed and length. The half-range is $(\max_s a_s - \min_s a_s)/2$, where a_s is the aggregate score for dataset seed s. It measures dataset-sampling variation rather than training variance or a confidence interval; the same convention applies to the post-trained table below and the extreme-length results in Section 4.4.

Jet-Long's mean at 64K is 0.71 pp below Self-Extend, with overlapping reported spreads. At 96K and 128K, it exceeds the strongest compared baseline by 4.50 and 11.99 pp, respectively. Self-Extend with $G = 19$ leads at 32K, while Jet-Long leads at 16K. Within the 8K native window, Jet-Long mathematically recovers base attention; its measured means differ from the unextended model by 0.21 and 0.16 pp at 4K and 8K. We report these implementation-level measurements without claiming numerically identical outputs.

Table 8: RULER on Meta-Llama-3-8B-Base (native window 8K): mean ± half-range over three dataset seeds at all seven lengths. DCA uses chunk size s and Self-Extend uses group size G; both use local window $w = 1{,}024$. Bold marks the highest mean at each length.

Method	4K	8K	16K	32K	64K	96K	128K
Unextended	**93.36** ± 0.04	**91.54** ± 0.55	0.03 ± 0.02	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
DCA ($s = 5{,}120$)	93.26 ± 0.17	89.18 ± 0.35	69.45 ± 0.81	56.16 ± 0.78	30.30 ± 0.40	16.32 ± 0.63	12.29 ± 0.25
DCA ($s = 7{,}168$)	93.25 ± 0.18	91.49 ± 0.52	65.43 ± 0.45	41.97 ± 0.56	13.58 ± 0.59	8.41 ± 0.32	6.00 ± 0.18
DNTK	93.26 ± 0.19	91.50 ± 0.50	84.31 ± 0.36	64.95 ± 0.54	28.94 ± 0.64	19.32 ± 0.11	11.59 ± 0.28
Self-Extend ($G = 32$)	88.46 ± 0.17	84.47 ± 0.55	81.46 ± 0.71	73.38 ± 0.22	**58.28** ± 0.97	47.86 ± 1.38	31.86 ± 0.99
Self-Extend ($G = 19$)	89.52 ± 0.18	86.28 ± 0.50	83.24 ± 0.60	**74.90** ± 0.38	53.99 ± 0.64	34.20 ± 0.53	14.64 ± 0.20
YaRN	77.14 ± 0.57	63.39 ± 0.77	57.34 ± 0.13	37.80 ± 0.42	31.87 ± 0.37	26.15 ± 0.89	12.21 ± 0.79
Jet-Long	93.15 ± 0.22	91.38 ± 0.61	**86.96** ± 0.28	73.18 ± 0.93	57.57 ± 1.17	**52.36** ± 1.64	**43.85** ± 0.67

D Post-trained checkpoints

Table 9 evaluates the post-trained Qwen3-1.7B and Qwen3-8B checkpoints using their chat templates across all seven lengths from 4K to 128K. Each entry aggregates all 13 RULER tasks with 500

examples per task on each of three independently generated datasets (seeds 42, 44, and 45). These checkpoints and prompts differ from the Base-checkpoint experiments in the main comparison.

Jet-Long has the highest mean at 64K, 96K, and 128K on Qwen3-1.7B. On Qwen3-8B, DNTK leads at 64K, while Jet-Long has the highest mean at 96K and 128K. Within the native 32K window, Jet-Long's measured means differ from the unextended model by at most 0.17 pp for 1.7B and 0.11 pp for 8B, despite the attention rules being mathematically equivalent. Daggers identify baseline cells with at least 10% empty generations in at least one seed; these scores reflect generation failures as well as task capability. DCA and Jet-Long have no such flags. The results demonstrate prompted long-context performance on RULER, with the broader evaluation limits discussed in Appendix A.

Table 9: RULER on post-trained Qwen3 checkpoints with their chat templates: mean ± half-range over three dataset seeds at all seven lengths. Bold marks the highest mean for each model and length. † denotes an empty-generation rate of at least 10% in at least one seed.

Method	4K	8K	16K	32K	64K	96K	128K
Qwen3-1.7B (post-trained)							
Unextended	81.50 ± 0.04	77.56 ± 0.73	**70.25** ± 1.28	**63.85** ± 0.87	42.89 ± 0.31	13.15 ± 0.42	2.37 ± 1.11
DCA	**81.51** ± 0.08	77.60 ± 0.84	70.20 ± 1.32	57.85 ± 0.49	44.13 ± 0.44	41.07 ± 0.65	38.51 ± 1.03
DNTK	81.49 ± 0.12	**77.63** ± 0.71	70.21 ± 1.40	63.82 ± 1.00	51.78 ± 0.35	36.28 ± 0.32†	27.70 ± 1.04†
Self-Extend	80.86 ± 0.10	75.49 ± 1.00	69.11 ± 1.95	57.83 ± 0.90	48.51 ± 1.76†	48.30 ± 0.49	42.75 ± 0.87†
YaRN	72.06 ± 1.59	60.18 ± 0.49	17.21 ± 1.35†	27.45 ± 1.63†	27.26 ± 0.38†	34.36 ± 0.14†	25.78 ± 0.19†
Jet-Long	81.38 ± 0.07	77.48 ± 0.76	70.11 ± 1.41	63.68 ± 0.81	**54.88** ± 1.23	**50.50** ± 0.89	**47.99** ± 0.90
Qwen3-8B (post-trained)							
Unextended	84.60 ± 0.18	78.83 ± 1.09	81.21 ± 1.33	77.59 ± 0.46	61.22 ± 0.41	19.70 ± 0.59	3.00 ± 0.20
DCA	84.60 ± 0.18	78.84 ± 1.15	**81.23** ± 1.50	76.29 ± 0.62	64.57 ± 0.43	62.78 ± 0.23	61.05 ± 0.46
DNTK	84.63 ± 0.18	78.88 ± 1.07	81.21 ± 1.39	**77.76** ± 0.32	**77.63** ± 0.73	70.56 ± 0.70	63.62 ± 0.64
Self-Extend	82.36 ± 0.12	77.75 ± 1.02	77.75 ± 0.42	76.81 ± 0.42	73.65 ± 0.79	71.81 ± 0.88	68.54 ± 0.16
YaRN	**84.81** ± 0.29	**81.03** ± 0.96	74.17 ± 0.50	76.19 ± 0.35	64.85 ± 0.46	61.54 ± 0.77	56.92 ± 0.68
Jet-Long	84.51 ± 0.17	78.81 ± 1.13	81.19 ± 1.47	77.48 ± 0.39	74.91 ± 0.23	**72.94** ± 0.36	**68.88** ± 0.21

E Cache correction and group-size transitions

Retained cache versus full recomputation. The correction rotations in Section 3.2 exactly reconstruct the current RoPE phases for stored key vectors. Historical hidden states and values remain unchanged, so this operation is not equivalent to recomputing the entire prefix under the new G. We compare retained-cache decoding (A) with complete re-prefill at every step (B) on Qwen3-1.7B-Base at the length boundary $L = 98{,}304 \rightarrow 98{,}305$, where $w_{\text{pretrained}} = 32{,}768$ and G changes from 3 to 4.

Table 10 shows a larger logit discrepancy after the transition, while all tested greedy decisions agree. At the boundary itself, the maximum absolute logit difference increases from 0.1719 to 0.8750. The nonzero discrepancy in the no-transition control reflects numerical differences between prefill and decode kernels. These observations characterize a finite test; they do not establish equivalence of the two decoding procedures.

Table 10: Retained-cache decoding (A) versus full re-prefill (B) around a $G : 3 \rightarrow 4$ transition on Qwen3-1.7B-Base. Agreement counts refer to the tested greedy next-token decisions.

Regime	**Mean** $\max \lvert\Delta\text{logit}\rvert$	$\text{KL}(A \parallel B)$	**Argmax agreement**
Before transition	0.1781	$\sim 10^{-3}$	5/5
After transition	0.8000	1.84×10^{-2}	5/5
No-transition control	0.2094	$\sim 5 \times 10^{-4}$	10/10

Stratified task analysis. We also analyze a subset of RULER generations containing 9,247 examples that cross a G boundary. To account for completion length, we stratify by model, task, target length, and completion-length bin, retaining 49 bins with at least 20 examples per arm. Table 11 reports the crossed-minus-non-crossed score difference. Its mean is -0.74 pp and its reported 95% confidence interval includes zero; the median is zero, and crossing scores lower in 24 of 49 bins. This

observational comparison does not identify a causal transition effect or rule out degradation on other workloads.

Table 11: Stratified RULER analysis of group-size transitions. Differences are scores for crossed minus non-crossed generations.

Statistic	Result
Mean score difference	-0.74 pp
95% confidence interval	$[-1.83, +0.35]$ pp
Median score difference	0.00 pp
Bins where crossing scored lower	24/49

F Baseline sensitivity

The configurations in Appendix I support the main Qwen3 Base-model comparisons. Table 12 provides a separate controlled rerun on Qwen3-1.7B-Base (500 examples per task, seed 42), testing additional DCA and Self-Extend configurations held fixed across 64K, 96K, and 128K. Here s/w denotes DCA's chunk/local-window sizes and G/w denotes Self-Extend's group/window sizes.

Reducing Self-Extend's group size from 8 to 5 improves its score by 3.07–3.93 pp; Jet-Long still leads the best compared configuration by 6.37, 3.53, and 5.81 pp at 64K, 96K, and 128K. Increasing DCA's chunk size does not improve this comparison. The Llama-family configurations in Table 8 also illustrate that a setting's effect depends on the model: the alternative larger DCA chunk and smaller Self-Extend group both reduce the scores at 64K–128K. The smaller Self-Extend group instead improves the 16K and 32K scores. These tests assess sensitivity to the listed settings rather than an exhaustive hyperparameter search.

Table 12: Additional baseline configurations on Qwen3-1.7B-Base, with 500 RULER examples per task and dataset seed 42. Each configuration is fixed across lengths.

Method and configuration	64K	96K	128K
DCA ($s = 20{,}480$, $w = 4{,}096$)	48.80	43.45	42.62
DCA ($s = 28{,}672$, $w = 4{,}096$)	47.66	40.83	38.52
Self-Extend ($G = 8$, $w = 1{,}024$)	54.44	54.41	50.25
Self-Extend ($G = 5$, $w = 1{,}024$)	58.37	57.48	54.12
Jet-Long	**64.74**	**61.01**	**59.93**

G Additional lengths and detailed diagnostics

The main Qwen3 comparisons use six lengths: 4K, 8K, 16K, 32K, 64K, and 128K. This appendix retains the additional 96K evaluations and the detailed task and window sweeps. Seven-length aggregates in this appendix include 96K and therefore differ from the six-length aggregates in the main text.

G.1 PG-19 perplexity by length

G.2 RULER per-task diagnostics

Table 14 gives the per-task diagnostic results at 64K. These diagnostics and the local-window sweep below are reported separately from the additional baseline-sensitivity runs in Appendix F.

G.3 Hybrid attention across seven lengths

Table 15 extends Table 4 with the 96K evaluation. Its averages include all seven lengths.

Table 13: 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**

Table 14: 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**

G.4 Full local-window sweep

Table 16 includes the additional 96K results and averages across the three extended lengths. The control $w_0 = 0$ retains self-attention in the local branch and leaves grouped remote attention active.

H Inference efficiency measurements and setup

Table 17 provides the reported Qwen3-8B-Base measurements from the revised H100 comparison, including the additional 96K length. These selected measurements complement Table 7; the summary ranges in Section 4.8 cover Qwen3-1.7B/4B/8B. All ratios use FA3 as the reference. The matched CuTe control without bifocal attention omits Jet-Long's local/remote attention treatment while retaining the underlying backend. As explained in Section 4.8, comparing it with fused Jet-Long measures the cost of the context-extension rule, while comparing fused and unfused Jet-Long at extended lengths measures the benefit of optimizing that rule.

Table 15: RULER accuracy by context length for Jet-Long applied to the hybrid Jet-Nemotron architecture [17], 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 pp)	37.40 (+24.88 pp)	33.78 (+25.24 pp)	52.94 (+10.01 pp)
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 pp)	36.69 (+29.25 pp)	33.14 (+27.49 pp)	53.47 (+11.31 pp)

Table 16: Full local-window ablation on Qwen3-4B/8B-Base at 64K, 96K, and 128K. Cells report RULER accuracy; red values give the gap in pp from the row's best score (**bold**). Avg aggregates the three extended lengths.

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

Table 17: Selected Qwen3-8B-Base prefill and decode throughput (tok/s) on H100 from the revised comparison. Parentheses give throughput relative to FA3 at the matching length. Dashes denote unreported decode cells; FA4 is compared for prefill only.

Method	4K	32K	64K	96K	128K
Prefill					
FA2	31211 (0.92×)	17796 (0.78×)	12238 (0.69×)	9332 (0.64×)	7433 (0.61×)
FA3	33915 (1.00×)	22869 (1.00×)	17701 (1.00×)	14615 (1.00×)	12269 (1.00×)
FA4	33455 (0.99×)	22358 (0.98×)	16690 (0.94×)	13693 (0.94×)	11400 (0.93×)
CuTe (without bifocal attention)	33235 (0.98×)	22431 (0.98×)	16860 (0.95×)	13810 (0.94×)	11494 (0.94×)
Jet-Long (unfused)	31242 (0.92×)	17810 (0.78×)	10909 (0.62×)	8548 (0.58×)	6932 (0.57×)
Jet-Long CuTe	33830 (1.00×)	22759 (1.00×)	15612 (0.88×)	12575 (0.86×)	10382 (0.85×)
Decode					
FA2	—	—	74.80 (1.09×)	67.01 (1.09×)	60.16 (1.07×)
FA3	—	—	68.33 (1.00×)	61.44 (1.00×)	56.05 (1.00×)
CuTe (without bifocal attention)	—	—	71.48 (1.05×)	64.14 (1.04×)	58.38 (1.04×)
Jet-Long (unfused)	—	—	14.20 (0.21×)	10.49 (0.17×)	8.32 (0.15×)
Jet-Long CuTe	—	—	73.90 (1.08×)	65.00 (1.06×)	58.03 (1.04×)

Hardware, precision, and software. Measurements use an NVIDIA H100 80GB HBM3 (SM90), driver 535.216.03, BF16 model weights and Q/K/V, and FP32 softmax and LogSumExp statistics. The software stack is PyTorch 2.9.1+cu128, CUDA 12.8, `nvcc` 12.6.85, `nvidia-cutlass-dsl` 4.6.0.dev0, `quack-kernels` 0.5.3, `cuda-python` 13.3.1, and `apache-tvm-ffi` 0.1.13rc1. The attention implementations use the same Dao-AILab/flash-attention repository, with the following versions and pinned commits:

FA2	`v2.8.3`
FA3 (`hopper/`)	`00756db9d921da0846453283ddfbeb7457abd09b`
FA4	`fa4-v4.0.0.beta23`, commit `2409214`
CUTLASS	`7127592069c2fe01b041e174ba4345ef9b279671`

Interpreting the controls. FA3 is the primary H100 baseline, and FA4 is a complementary implementation with an SM90 path. At 64K–128K across the three Qwen3 sizes, fused Jet-Long prefill reaches 0.90–0.94× FA4 in addition to the FA3 and matched-control comparisons in Section 4.8. The CuTe control itself improves on FA2, so a comparison with FA2 alone would conflate the routing

Table 18: 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 [13]	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 [35]	HuggingFace `rope_type=yarn`, `factor`= 4.0, `max_position_embeddings`= 131,072, `original_max_position_embeddings`= 32,768 (target context 32K $\times$ 4 = 128K)
DCA [2]	`chunk_size`= 20,480, `local_window`= 4,096
Self-Extend [22]	`group_size`= 8, `window_size`= 1,024, `scale_base`= -1

cost with a change of backend. For Qwen3-8B decode, Jet-Long's throughput is close to the matched control and exceeds FA3 at the reported extended lengths. These throughput comparisons are specific to the measured hardware, software stack, and model configurations.

Native-window fallback. For $L \leq 32$K, $G = 1$ recovers standard attention and bypasses the extension code. The revised implementation uses FA3 for this path instead of the earlier FA2 fallback. The $L > 32$K branch is unchanged; all reported revised measurements were rerun under the frozen stack above. This implementation change affects efficiency measurements, while the attention definition and accuracy results are unchanged.

I Baseline configurations

The four zero-shot baseline configurations in the main Qwen3 Base-model comparison are held constant across Qwen3-1.7B/4B/8B and the six main evaluation lengths, as outlined in Table 18. These three Base checkpoints have a pretrained context window of 32,768 tokens and a RoPE base $\theta = 10^6$. Additional configurations are evaluated separately in Appendix F; Llama-specific DCA and Self-Extend settings appear in Table 8.

J Compute resources

The original Qwen3 Base-model and Jet-Nemotron experiments use NVIDIA H100 80GB Tensor Core GPUs. A RULER evaluation sweep over the 13 tasks, 500 samples each, across seven context lengths including the appendix-only 96K evaluation takes roughly 40–96 H100 GPU-hours per (model, method) pair, with the spread driven by model size (Qwen3-1.7B/4B/8B) and the longer-context tasks dominating the budget. Each HELMET-RAG run takes roughly 8–16 H100 GPU-hours per (model, method) pair, as does each PG-19 long-form perplexity sweep. The added cross-family and post-trained evaluations use three dataset seeds with 500 examples per task, while the 256K–1M study uses 100 examples per task and three seeds. These are additional evaluations beyond the original sweeps. Hardware and software for the revised throughput measurements are specified in Appendix H.

K Broader impacts

Positive impacts. Jet-Long extends existing checkpoints without additional training, which can lower the training cost of adapting models to long-document QA, retrieval-augmented generation, and other long-input applications. Preserving the native-window attention rule allows the same checkpoint to support both short and extended inputs. The fused implementation reduces the overhead of the bifocal construction, although long-context inference still requires substantial memory and computation. Benefits for coding agents, energy consumption, or substitution of larger models are potential applications rather than outcomes directly measured in this work.

Potential negative impacts. The long-context capabilities Jet-Long unlocks inherit the standard risks of the underlying open-weight base model (e.g., scaled disinformation, longer surveillance

or data-extraction prompts, amplification of pretraining bias over longer contexts). One context-extension-specific concern is that extending the input window enables agents to ingest longer streams of personal data within a single session, which can amplify privacy risks if intermediate KV caches are not zeroed; downstream deployers should treat KV caches with the same retention policies as raw input. Jet-Long does not introduce new risk axes beyond those of the base checkpoint it is applied to and provides no additional access to model weights or training data; mitigation should rely on the standard practices around responsible deployment of the underlying open-weight LLMs.

L Licenses for existing assets

Our experiments rely entirely on publicly released pretrained language models and benchmark datasets, all of which are properly cited in the main text. The Qwen3 base models [49] we evaluate on (Qwen3-1.7B-Base, Qwen3-4B-Base, Qwen3-8B-Base) are released by Alibaba under the Apache License 2.0 (https://huggingface.co/Qwen/Qwen3-8B-Base), which permits research and commercial use with attribution. The hybrid Jet-Nemotron-2B and Jet-Nemotron-4B models [17] are released by NVIDIA at https://github.com/NVlabs/Jet-Nemotron: the inference code is licensed under the Apache License 2.0, while the model weights are licensed under the NVIDIA non-commercial research license (https://github.com/NVlabs/Jet-Nemotron/blob/main/LICENSE/jet_nemotron_models), which permits research and evaluation but reserves commercial use to NVIDIA; our use is strictly for academic research and does not constitute commercial deployment, and we do not redistribute model weights. For benchmarks, RULER [20] is released by NVIDIA under the Apache License 2.0 (https://github.com/NVIDIA/RULER); HELMET [53] is released by Princeton NLP under the MIT License (https://github.com/princeton-nlp/HELMET); and PG-19 [38] is released by Google DeepMind under the Apache License 2.0 (https://github.com/google-deepmind/pg19). All three benchmarks are used strictly for academic evaluation as intended by their creators. We do not redistribute any of these datasets and refer readers to the original sources for access.

The additional post-trained Qwen3 checkpoints and Qwen3-30B-A3B-Instruct-2507 are released under Apache License 2.0 (https://huggingface.co/Qwen/Qwen3-8B; https://huggingface.co/Qwen/Qwen3-30B-A3B-Instruct-2507). Meta-Llama-3-8B is distributed under the Meta Llama 3 Community License Agreement and its accompanying acceptable-use policy (https://huggingface.co/meta-llama/Meta-Llama-3-8B). We use the released checkpoints for evaluation and do not redistribute their weights.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We elucidate our contributions in the Abstract and Introduction.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Appendix A, the first appendix section, discusses comparison scope, evaluation coverage, the cache-history approximation, positional resolution, and systems constraints.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

Justification: We do not have theoretical results.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Experimental details are in Section 4.1; baseline configurations and sensitivity tests are in Appendices I and F. Section 4.4 and Appendices C and D describe the additional evaluations, and Appendix H gives the pinned efficiency stack.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The evaluated datasets are publicly available. The paper links the Jet-Long code repository, and the supplementary material includes a PyTorch implementation. Experimental settings and the pinned efficiency stack are described in the main text and appendix.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

Answer: [Yes]

Justification: This is discussed in Section 4.1.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: The cross-family, post-trained, and 256K–1M RULER experiments report mean and half-range over three independently generated datasets (Sections 4.3 and 4.4; Appendices C and D). Appendix B defines this descriptive measure and distinguishes it from a confidence interval or training variance. The original Base-model tables and single-seed sensitivity tests do not report uncertainty; we do not claim statistical significance from their small margins.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.

- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Appendix J reports H100 GPU-hour estimates for the original benchmark sweeps and sample counts for the added evaluations. Appendix H specifies hardware, precision, software versions, and kernel commits for the revised throughput measurements.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have carefully reviewed our research, ensuring it conforms with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: Appendix K discusses potential deployment benefits and negative impacts, distinguishing measured long-context results from possible benefits for agentic workflows or energy consumption.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: At present, our proposed approach poses no immediate risk of misuse.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: All pre-trained models and datasets used in our experiments are publicly available and properly cited in the main text. We discuss their licenses and our usage in Appendix L.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: We do not release new datasets or model checkpoints. The accompanying implementation and linked repository provide a reproducibility artifact for the method; the experimental settings and dependencies are documented in the paper and appendix.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: No crowdsourcing experiments and no research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: No crowdsourcing experiments and no research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: LLMs were only used for grammar and for occasional code debugging.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.