

TRONTOR: To RoPE or Not to RoPE for Long-Context Modeling

Youngjun Son¹, Jongyeon Park¹, Jaejin Lee^{1,2}

¹Graduate School of Data Science, Seoul National University

²Department of Computer Science and Engineering, Seoul National University

{jun041577, iscopark67, jaejin}@snu.ac.kr

Abstract

While Rotary Position Embedding (RoPE) is widely used in large language models, its inductive biases can impair long-context reasoning and lead to attention sinks. In this work, we propose TRONTOR, a method that effectively combines RoPE and No Positional Encoding (NoPE) through a simple fine-tuning phase, bypassing the need for expensive pretraining modifications. Our preliminary experiments demonstrate that mixing RoPE and NoPE at the attention-head level can enhance long-context capabilities, while static selection strategies fail to capture the varying complexity of inputs. Based on this observation, TRONTOR employs a lightweight context-aware routing method that dynamically switches each attention head between RoPE and NoPE at every token step. This router-only configuration provides an efficient operating point, yielding long-context performance gains with minimal inference overhead. For further gains, we extend TRONTOR to TRONTOR-Agg, which applies simple unweighted averaging over multiple RoPE bases within RoPE-active heads while avoiding learnable intra-head routing. Evaluated on the RULER benchmark using Llama-3.2 (1B and 3B) and Qwen3 (1.7B and 4B), TRONTOR improves long-context performance by mitigating attention sinks and dynamically reallocating attention toward relevant mid-context tokens.

1 Introduction

Large language models (LLMs) have achieved significant advancements in language understanding, generation, and reasoning (Grattafiori et al., 2024; Yang et al., 2025a; Achiam et al., 2023; Gemini Team et al., 2024). As their deployment increasingly relies on long-form interactions and retrieval-augmented generation, the effective utilization of extended context has become a crucial determinant of practical capability (Bai et al., 2024a; Li et al., 2024). Nevertheless, models continue to exhibit limitations in long-range reasoning.

Notable issues include the *lost-in-the-middle* phenomenon (Liu et al., 2024a) and the emergence of attention sinks (Xiao et al., 2024; Gu et al., 2025), where the model disproportionately focuses on initial tokens while neglecting crucial mid-context evidence.

Recent studies argue that Rotary Position Embedding (RoPE) (Su et al., 2024), the predominantly used position embedding in language models, imposes structural constraints on long-context modeling. Chiang and Yogatama (2025) hypothesized that RoPE may induce dimension inefficiency within attention heads during long-distance retrieval. Furthermore, Chen et al. (2025) empirically demonstrated that RoPE can contribute to a global U-shaped attention pattern, with reduced attention to intermediate positions relative to early and recent tokens.

In response, approaches such as Ms-PoE (Zhang et al., 2024) and MoICE (Lin et al., 2024) have attempted to modify RoPE by introducing heterogeneous frequency bases or learnable intra-head routers. Meanwhile, evidence suggests that Transformers can implicitly recover positional information without explicit embeddings (Kazemnejad et al., 2023), implying that an explicit, *always-on* positional bias may be redundant or even detrimental in certain contexts.

Motivated by these observations, we ask whether positional information must be imposed uniformly across all attention heads. We answer this question with TRONTOR, which treats positional encoding as a context-dependent choice: a lightweight router selects between RoPE and NoPE for each head-token pair. This design differs from prior head-wise scaling methods that retain RoPE for every head and adapt only its frequency. To further examine the benefit of multi-scale positional modeling, we introduce TRONTOR-Agg, which augments only RoPE-active heads with parameter-free, unweighted aggregation over multiple RoPE bases.

The contributions of this paper are summarized as follows:

- By introducing a lightweight, learnable gating mechanism, we flexibly integrate NoPE at the attention-head level based on context-aware thresholding. This approach effectively mitigates the inherent limitations of RoPE while yielding improvements on the long context benchmarks.
- We question the necessity of complex learnable intra-head routing schemes. Our results show that simple unweighted averaging over multiple RoPE bases can achieve competitive performance without introducing additional intra-head routing parameters.
- Empirical analyses show that dynamically substituting RoPE with NoPE at router-selected head–token pairs reduces excessive attention to initial tokens and reallocates attention toward relevant mid-context evidence.

2 Background

This section presents essential background on RoPE, long-context degradation phenomena, and previous efforts to modify or relax positional inductive biases.

2.1 Rotary Position Embedding

RoPE (Su et al., 2024) has become the predominant positional encoding in contemporary LLM architectures. RoPE injects positional information by rotating the query and key vectors according to their respective token positions. Instead of using a full rotation matrix, RoPE applies rotations to pairs of elements within the embedding. For an attention head with dimension d , the rotation frequencies are defined as $\theta_i = b^{-2i/d}$, where b denotes the base frequency. For a query vector $\mathbf{q}_n$ at position n , the transformed vector $\hat{\mathbf{q}}_n$ is obtained by rotating each pair of dimensions $(2i, 2i + 1)$ by the angle $n\theta_i$:

$$\hat{\mathbf{q}}_{n,[2i,2i+1]} = \begin{pmatrix} \cos(n\theta_i) & -\sin(n\theta_i) \\ \sin(n\theta_i) & \cos(n\theta_i) \end{pmatrix} \begin{pmatrix} \mathbf{q}_{n,2i} \\ \mathbf{q}_{n,2i+1} \end{pmatrix} \quad (1)$$

Extending this operation to the full dimension d results in individual 2D rotations forming a block-diagonal matrix $\mathbf{R}_{\theta,n}$. As a result, when evaluated against a key vector $\mathbf{k}_m$ at position m , the inner product between the rotated query and key vectors

is determined exclusively by their relative distance:

$$\hat{\mathbf{q}}_n^\top \mathbf{k}_m = \mathbf{q}_n^\top \mathbf{R}_{\theta,m-n} \mathbf{k}_m.$$

$\mathbf{R}_{\theta,m-n} = \mathbf{R}_{\theta,n}^\top \mathbf{R}_{\theta,m}$ represents the rotation matrix corresponding to the relative position $(m - n)$.

2.2 Long-Context Awareness

Current LLMs systematically underattend to relevant information in the middle of extended contexts (Liu et al., 2024a). This bias is partly structural: softmax normalization can induce an “attention sink” that absorbs non-informative attention mass through coupling among attention scores (Gu et al., 2025). Furthermore, recent findings indicate that RoPE also contributes to this positional imbalance. For instance, Chiang and Yogatama (2025) note that rotary encodings create representational bottlenecks during extended dependency modeling. Moreover, RoPE can bias positional attention distributions, reducing attention to intermediate tokens through distance-based decay while relatively favoring initial positions (Chen et al., 2025).

2.3 RoPE Variations

Several approaches have been proposed to address positional attention imbalance and mitigate the degradation of RoPE under long-context extrapolation. These methods generally modify the application of positional rotations and can be categorized into frequency-domain scaling and head-wise positional variation.

Scaling in the frequency domain. A prominent research direction modifies RoPE by adjusting rotation frequencies across hidden dimensions to extrapolate beyond the training context length. Positional Interpolation (PI; Chen et al., 2023) achieves this by linearly downscaling position indices. However, linear compression reduces the resolution of high-frequency components, potentially degrading local modeling performance. Liu et al. (2024b) established scaling laws for RoPE-based extrapolation, revealing how different frequency dimensions critically impact length generalization.

Building on such insights, NTK-aware scaled RoPE (Peng and Quesnelle, 2023) introduces a non-linear scaling scheme that extrapolates high-frequency dimensions to maintain local precision while interpolating low-frequency components. YaRN (Peng et al., 2024) further refines this by applying length-dependent temperature scaling to

attention logits. Taking a signal processing approach, FoPE (Hua et al., 2025) mitigates spectral damage by modeling each dimension as a multi-frequency Fourier series and zeroing out inadequately trained low-frequency components. Additionally, LongRoPE (Ding et al., 2024) and its successor LongRoPE2 (Shang et al., 2025) utilize evolutionary search to discover non-uniform, dimension-specific scaling factors, successfully extending context windows to millions of tokens with near-lossless performance.

Head-wise scaling. Recent analyses have shown that individual attention heads specialize in distinct functional roles (Nam et al., 2025; Yang et al., 2025c). This observation prompts the question of whether attention heads should receive differentiated positional information tailored to their specific behaviors. Ms-PoE (Zhang et al., 2024) assigns fixed RoPE scaling factors to individual heads, allowing them to operate across distinct positional ranges. Similarly, HARPE (Lian et al., 2025) assigns a distinct RoPE base to each attention head, enabling direct single-stage continual pretraining at the target context length. Furthermore, Attention Buckets (Chen et al., 2024) aggregates predictions from parallel RoPE-base runs using confidence-derived weights and MoICE (Lin et al., 2024) aggregates outputs from multiple sets of bases using a weighted sum with learnable parameters. While effective, these approaches fundamentally assume that each head must retain a RoPE-based positional signal, thereby preserving RoPE’s inductive biases.

No positional encoding (NoPE). Kazemnejad et al. (2023) demonstrated that transformers trained without explicit positional encoding (NoPE) can outperform conventional schemes, theoretically proving that NoPE can implicitly capture positional information. Further investigating this capability, Wang et al. (2024) confirmed that NoPE is generally advantageous for length generalization. However, they observed that at extreme context lengths, NoPE eventually degrades due to the *distraction of attention distributions*, where attention weights become excessively uniform and fail to focus on crucial tokens. To harness the strengths of NoPE while mitigating these limits, several approaches have emerged. Recent hybrid architectures interleave RoPE and NoPE at the layer level during pre-training (Yang et al., 2025b; Bae et al., 2025). In these models, RoPE layers provide an inductive bias for short-range aggregation, whereas NoPE

layers facilitate long-range retrieval through pure vector-similarity alignment.

However, they rely on costly pre-training and statically fix the positional roles of each layer. In contrast, TRONTOR introduces a dynamic, head-wise selection mechanism that adaptively determines whether to apply RoPE or NoPE. This approach achieves a flexible combination of positional encodings with minimal tuning, requiring only a lightweight router to be trained while the LLM backbone remains frozen.

Notably, our work focuses specifically on head-wise scaling. Since this routing mechanism is orthogonal to frequency-domain scaling, TRONTOR can be integrated with existing frequency scaling methods applied to the target model.

Table 1: Impact of NoPE replacement on RULER in preliminary experiments. Bold-faced numbers indicate a better result than the baseline.

Layer Index	Method	Average Score ($\uparrow$)	
		4k	8k
0	PAS	87.80	73.30
	Random	91.00	77.90
2	PAS	91.20	77.50
	Random	90.60	78.90
15	PAS	92.70	82.90
	Random	91.60	81.50
17	PAS	93.00	79.60
	Random	93.70	83.50
22	PAS	93.20	78.40
	Random	92.60	80.40
Baseline	—	92.50	78.70

3 Observations

We begin with an exploratory analysis that demonstrates the potential of selectively mixing RoPE and NoPE at the attention-head level. To validate the hypothesis that eliminating redundant positional information from specific attention heads can enhance model performance, preliminary experiments were conducted on the pre-trained Llama3.2-3B-Instruct (Grattafiori et al., 2024) model. In these experiments, a single layer was systematically targeted at a time, converting a subset of its attention heads—targeting less than 20% of the total heads (up to four heads)—from RoPE to NoPE without further training. This approach enabled isolation of the impact of selective positional encoding at the head level. To determine which heads to convert, two distinct strategies are employed: (1) Position-Awareness Score (PAS) guided selection, which targets up to four heads with the lowest position

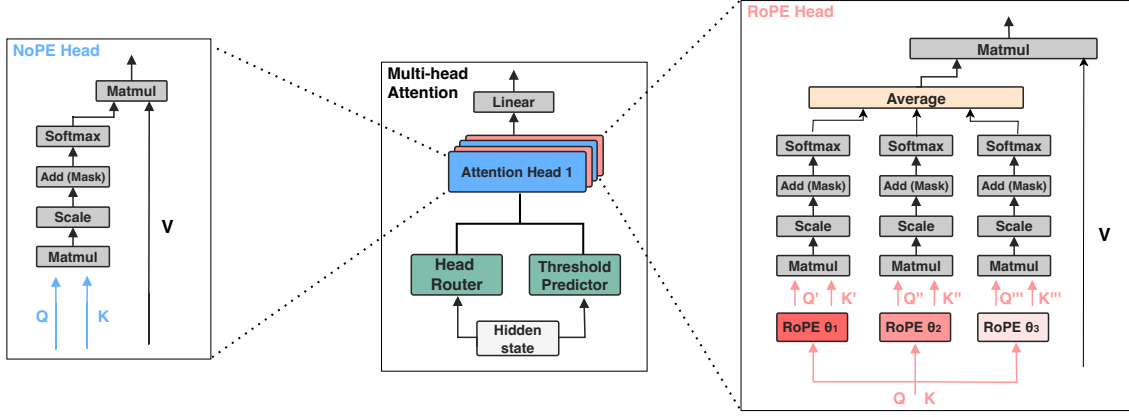


Figure 1: Overall architecture of TRONTOR and TRONTOR-Agg. The head router and threshold predictor constitute TRONTOR, while the multi-base RoPE branch is additionally activated in TRONTOR-Agg.

awareness score (Zhang et al., 2024) in the selected layer; and (2) random selection as a control. Detailed formulations and procedures for these selection strategies are provided in Appendix A.

Evaluation was conducted on a subset of the RULER benchmark (Hsieh et al., 2024), focusing on Needle-In-A-Haystack (NIAH) and Variable Tracking (VT) tasks at 4k and 8k context lengths. Table 1 presents average accuracy per context length, particularly illustrating the case where exactly four heads were replaced. The results confirm that NoPE replacement offers potential for performance gains, although its effects vary by layer. Specifically, removing RoPE in early layers (e.g., 0–2) reduces performance, whereas removing it in mid-to-late layers often improves results.

The observation that random selection occasionally outperforms PAS indicates that static metrics cannot reliably capture this potential. Note that these random selection results represent the best outcomes observed across 10 independent seeds. Since relying on such random chance is practically unfeasible for stable deployment, the key to unlocking consistent performance gains lies in identifying the optimal subset of heads to convert. This insight directly motivates the introduction of a dynamic head router that adaptively switches between RoPE and NoPE based on the input context to capture optimal positional configurations consistently.

4 Methods

Drawing on our findings, we present TRONTOR, a lightweight mechanism designed to execute the head-level routing. Figure 1 provides an overview of the architecture.

4.1 Model Architecture

TRONTOR includes a head router and threshold predictor.

Head router. Unlike static approaches, TRONTOR predicts whether to apply RoPE for each head at every token step. Let $x_t \in \mathbb{R}^d$ denote the input hidden state for the t -th token, where d represents the hidden dimension of the model. The router computes a head-specific score $s_t \in \mathbb{R}^H$ for H attention heads:

$$s_t = \text{LayerNorm}(W_2 \cdot \sigma(W_1 \cdot x_t)) \quad (2)$$

Here, σ denotes the activation function, specifically SiLU (Elfwing et al., 2018), and W_1 and W_2 are learnable weight matrices. The h -th component of this vector, $s_{t,h}$, represents the model’s preference for bypassing RoPE for the h -th head, with higher values indicating a greater likelihood that the head functions as a NoPE head. Consistent with our findings that removing RoPE in early layers degrades performance, the router is disabled for the first three layers.

Threshold predictor. A fixed decision boundary prevents the model from adapting to varying language contexts. Inspired by the method proposed by Wang and Tang (2025), we employ a context-sensitive mechanism via a *threshold predictor*. The threshold τ_t is derived from query vectors $q_{t,h}$ to reflect the search intent at time t . Specifically, τ_t is determined by a global context vector c_t , obtained by averaging queries across all heads. Subsequently, the selection probability $P_{t,h}$ is derived by comparing the head-specific score $s_{t,h}$ against

this threshold as follows:

$$\begin{aligned} c_t &= \frac{1}{H} \sum_{h=1}^H q_{t,h}, & \tau_t &= W_3 \cdot c_t + b \\ P_{t,h} &= \text{Sigmoid} \left(\frac{s_{t,h} - \tau_t}{T} \right) \end{aligned} \quad (3)$$

where W_3 and b are learnable parameters, and the temperature scaling factor T is set to 0.1.

The model makes a binary decision to activate NoPE behavior if $P_{t,h} > 0.5$, which is equivalent to $s_{t,h} > \tau_t$, and retains RoPE otherwise. Although this discretization is non-differentiable, we employ the straight-through estimator (STE) (Bengio et al., 2013) during training. Additionally, to maintain training stability, the predictor’s parameters are initialized so that the initial probability of selecting NoPE is approximately 5%. This intentional bias allows the framework to gradually adapt its handling of positional information, activating NoPE only in contexts where it provides a clear semantic benefit.

Extension with intra-head RoPE aggregation.

The formulation above defines TRONTOR, in which each RoPE-active head uses the default RoPE base of the backbone model. We further extend TRONTOR by applying multiple RoPE bases within each RoPE-active head; we refer to this extended variant as TRONTOR-Agg. Specifically, TRONTOR-Agg computes attention separately with each RoPE base and takes a simple unweighted average of the resulting outputs. Unlike MoICE (Lin et al., 2024), this aggregation requires no learnable weighting parameters. For a set of M RoPE bases $\mathcal{B} = \{b_1, \dots, b_M\}$, the aggregated output o_h for head h at the t -th token is defined as follows:

$$o_h = \frac{1}{M} \sum_{m=1}^M \text{Attention}(\hat{q}_{t,h,m}, \hat{\mathbf{K}}_{t,h,m}, \mathbf{V}_{t,h}). \quad (4)$$

In this formulation, $\hat{q}_{t,h,m}$ refers to the query vector of the current token at time step t , which is rotated by applying RoPE with base b_m and $\hat{\mathbf{K}}_{t,h,m}$ denotes the sequence of key vectors from the beginning of the context up to the current step t , where each vector in the sequence is also rotated using the same base b_m . In contrast, $\mathbf{V}_{t,h}$ represents the sequence of value vectors up to step t in their original latent space, as they remain unrotated. Following the search scheme in Chen et al. (2024), we construct a candidate base pool of

$\{500, 515, 535, 635, 760, 800, 900\} \times 10^3$. Specifically, we select $M = 3$ bases from this pool for the 1B model, while utilizing all $M = 7$ bases for larger models.

Final output integration. In both TRONTOR and TRONTOR-Agg, the outputs of all H attention heads are concatenated and passed through the output projection to produce the final multi-head attention output. The overall procedure of TRONTOR-Agg is summarized in Algorithm 1.

4.2 Training the Model

To ensure the router makes clear binary decisions while maintaining a preference for the default positional bias, a specialized auxiliary loss function is introduced, consisting of a sparsity term and an uncertainty term.

Sparsity loss. A sudden and complete shift to NoPE across all heads could disrupt the model’s positional understanding. To ensure stable adaptation, RoPE is maintained as the default state, and NoPE is activated only when the specific context provides clear semantic benefits. This is achieved by regularizing the mean activation probability of NoPE:

$$\mathcal{L}_{\text{sparse}} = \frac{1}{N} \sum P_{t,h} \quad (5)$$

Here, $P_{t,h}$ denotes the predicted probability of activating NoPE, and N represents the total number of routing decisions. This term acts as a soft constraint to prevent excessive deviation from the pre-trained positional structure.

Uncertainty loss. To prevent the router from remaining in an ambiguous state (i.e., $P_{t,h} \approx 0.5$) and to facilitate clear binary switching, an uncertainty penalty is imposed:

$$\mathcal{L}_{\text{uncert}} = 1 - 4 \cdot \frac{1}{N} \sum (P_{t,h} - 0.5)^2 \quad (6)$$

This loss is minimized when $P_{t,h}$ approaches 0 or 1 and maximized when $P_{t,h}$ is 0.5, thereby encouraging the router to make confident decisions.

The total auxiliary loss $\mathcal{L}_{\text{aux}}$ is then defined as a weighted sum of these two terms:

$$\mathcal{L}_{\text{aux}} = \lambda_{\text{sparse}} \mathcal{L}_{\text{sparse}} + \lambda_{\text{uncert}} \mathcal{L}_{\text{uncert}}. \quad (7)$$

The total training loss $\mathcal{L}$ is defined as the sum of the negative log-likelihood loss $\mathcal{L}_{\text{NLL}}$ and the auxiliary loss $\mathcal{L}_{\text{aux}}$, formulated as $\mathcal{L} = \mathcal{L}_{\text{NLL}} + \mathcal{L}_{\text{aux}}$.

Algorithm 1: Dynamic Head Routing with Intra-head RoPE Aggregation

Require: Input hidden state x_t for the t -th token, and its projections $q_{t,h}, k_{t,h}, v_{t,h}$ for each head h
Require: Key and value sequences up to step t : $\mathbf{K}_{t,h} = \{k_{1,h}, \dots, k_{t,h}\}, \mathbf{V}_{t,h} = \{v_{1,h}, \dots, v_{t,h}\}$
Require: Set of M RoPE bases $\mathcal{B} = \{b_1, \dots, b_M\}$
Require: Head router ϕ , Threshold predictor ψ , Temperature T , Output projection W_O

- 1: $s_t \leftarrow \phi(x_t)$ # Compute routing scores for all H heads
- 2: $c_t \leftarrow \frac{1}{H} \sum_{h=1}^H q_{t,h}$
- 3: $\tau_t \leftarrow \psi(c_t)$ # Predict dynamic threshold
- 4: **for** each head $h \in \{1, \dots, H\}$ **do**
- 5: $P_{t,h} \leftarrow \text{Sigmoid}((s_{t,h} - \tau_t)/T)$
- 6: **if** $P_{t,h} > 0.5$ **then**
- 7: $o_h \leftarrow \text{Attention}(q_{t,h}, \mathbf{K}_{t,h}, \mathbf{V}_{t,h})$
- 8: **else**
- 9: **for** each base $b_m \in \mathcal{B}$ **do**
- 10: $\hat{q}_{t,h,m} \leftarrow \text{RoPE}(q_{t,h}, b_m)$ # Apply RoPE with base b_m on query
- 11: $\hat{\mathbf{K}}_{t,h,m} \leftarrow \text{RoPE}(\mathbf{K}_{t,h}, b_m)$ # Apply RoPE with base b_m on key vectors
- 12: $\text{attn}_{h,m} \leftarrow \text{Attention}(\hat{q}_{t,h,m}, \hat{\mathbf{K}}_{t,h,m}, \mathbf{V}_{t,h})$
- 13: **end for**
- 14: $o_h \leftarrow \frac{1}{M} \sum_{m=1}^M \text{attn}_{h,m}$
- 15: **end if**
- 16: **end for**
- 17: $O_t \leftarrow \text{Concat}(o_1, o_2, \dots, o_H)$
- 18: $Y_t \leftarrow W_O \cdot O_t$ # Final output projection
- 19: **return** Y_t

5 Experiments

This section outlines the experimental setup and provides a comparative analysis with several baseline models to evaluate TRONTOR’s performance.

5.1 Experimental Setup

The model and baselines. TRONTOR is implemented on Llama-3.2-Instruct-1B/3B (Grattafiori et al., 2024) and Qwen3-1.7B/4B (Yang et al., 2025a) to assess its effectiveness across architectures and scales. We evaluate performance against five configurations: (1) Baseline: the original pre-trained model; (2) MoICE (Lin et al., 2024): a multi-scale method computing a weighted sum of outputs from multiple RoPE bases; (3) AVG: an ablation applying simple unweighted intra-head aggregation across all heads; (4) TRONTOR; and (5) TRONTOR-Agg, as described in Section 4.

Among existing head-wise scaling approaches, we focus on MoICE because it is a recent learned multi-base baseline closely aligned with our setting.

Dataset and training. We use OpenHermes-2.5-1k-longest dataset (Teknum, 2023) during router training, following MoICE (Lin et al., 2024). Other training hyperparameters, such as the learning rate and batch size, are also set to be identical to those used in MoICE. The pre-trained LLM is frozen, and only the lightweight routing layers (less than 0.1% of total parameters) are optimized. The en-

tire training process is completed in less than 10 minutes using 4 NVIDIA B200 GPUs.

Hyperparameters. Our training objective aimed to maintain the proportion of NoPE heads at approximately 5% (typically 0 to 2 heads per layer), an empirically determined ratio that ensures a highly selective adaptation. Through an extensive grid search on the auxiliary loss coefficients using the Llama-3.2-1B-Instruct model, we identified the optimal values as $\lambda_{\text{sparse}} = 1.75$ and $\lambda_{\text{uncert}} = 0.5$. We confirmed that these values yield robust performance across different model sizes and various tasks. A detailed analysis of the hyperparameter search is provided in Appendix D.

5.2 Performance on RULER Tasks

Table 2 shows the evaluation results. Long-context performance is evaluated using the RULER benchmark across six context lengths, with tasks categorized as Needle-in-a-Haystack (NIAH), Variable Tracking (VT), Word Extraction (WE), and Question Answering (QA). NIAH reports the average retrieval task accuracy (*niah_single_1*, *niah_single_2*, *niah_single_3*, *niah_multiquery*, and *niah_multivalue*). WE is calculated as the average of *ruler_cwe* and *ruler_fwe*, while QA is the average of *ruler_qa_hotpot* and *ruler_qa_squad*. VT reflects performance on the *ruler_vt* task. All results are averaged over three independent runs with different random seeds.

Table 2: Detailed evaluation results on the RULER benchmark. The configurations are described in Section 5.1. The best result in each task–length setting is indicated in **bold**.

(a) Llama-3.2 Models							(b) Qwen3 Models						
1B-Instruct							1.7B						
Task	Length	Baseline	MoICE	AVG	TRONTOR	TRONTOR-Agg	Task	Length	Baseline	MoICE	AVG	TRONTOR	TRONTOR-Agg
NIAH	4k	99.60	99.52	99.60	99.65	99.62	NIAH	4k	99.89	99.85	99.89	99.82	99.80
	8k	98.56	99.23	99.20	98.87	99.29		8k	99.77	99.74	99.77	99.75	99.74
	16k	95.49	96.87	96.48	96.55	97.81		16k	99.65	99.60	99.71	99.65	99.77
	32k	88.98	91.42	91.29	90.58	92.50		32k	98.79	99.05	99.19	99.27	99.27
	64k	84.52	85.20	85.24	85.89	85.97		64k	58.96	60.23	60.19	60.07	60.40
	128k	73.66	74.71	74.68	74.59	74.84		128k	28.93	32.37	32.28	32.72	32.94
WE	4k	58.88	58.79	58.70	59.42	59.38	WE	4k	81.27	82.10	81.68	82.14	82.71
	8k	40.41	40.22	40.21	42.18	42.14		8k	68.34	68.92	69.04	69.81	69.77
	16k	30.48	34.14	34.39	34.86	34.53		16k	59.26	58.94	58.89	59.43	59.31
	32k	20.35	24.45	24.83	24.76	24.80		32k	46.98	47.66	47.33	48.20	48.60
	64k	15.66	17.65	17.58	17.72	18.01		64k	36.76	37.40	37.52	37.60	38.37
	128k	11.73	13.56	13.56	13.27	13.73		128k	43.81	45.18	45.34	45.25	46.03
QA	4k	49.16	49.58	49.40	49.68	49.84	QA	4k	47.86	48.25	48.67	48.22	49.05
	8k	41.47	42.29	42.53	42.63	43.05		8k	37.63	38.05	37.94	38.05	38.18
	16k	37.53	38.01	38.38	37.68	39.38		16k	44.32	46.62	46.50	46.27	46.90
	32k	35.44	35.98	35.88	37.09	37.06		32k	41.70	42.67	43.28	42.95	43.64
	64k	32.33	33.72	33.81	33.27	34.23		64k	28.37	30.19	30.24	32.05	32.41
	128k	25.38	27.64	27.48	27.92	28.00		128k	25.86	27.54	27.61	27.16	27.92
VT	4k	82.24	84.64	84.40	84.37	85.18	VT	4k	96.13	96.57	96.40	96.53	96.57
	8k	74.12	75.50	75.64	77.07	77.24		8k	93.55	95.10	94.67	95.31	95.31
	16k	70.63	73.66	74.31	75.18	76.82		16k	90.52	91.62	91.88	92.32	92.50
	32k	78.80	80.52	80.40	79.92	80.92		32k	79.01	82.17	81.94	83.13	83.50
	64k	63.12	65.28	65.39	66.45	67.33		64k	21.68	24.33	24.98	25.23	25.28
	128k	41.21	43.56	44.02	44.59	45.72		128k	7.04	8.81	8.49	9.40	9.86
3B-Instruct							4B						
Task	Length	Baseline	MoICE	AVG	TRONTOR	TRONTOR-Agg	Task	Length	Baseline	MoICE	AVG	TRONTOR	TRONTOR-Agg
NIAH	4k	99.92	99.97	99.97	99.92	99.97	NIAH	4k	99.91	99.95	99.95	99.92	99.95
	8k	99.85	99.95	99.96	99.78	99.90		8k	99.90	100.00	99.95	99.95	100.00
	16k	99.64	99.60	99.62	99.60	99.64		16k	99.65	99.80	99.80	99.87	99.90
	32k	99.46	99.68	99.68	99.74	99.59		32k	99.85	99.89	99.91	99.87	99.91
	64k	98.54	98.74	98.74	98.98	98.98		64k	62.91	63.81	63.94	63.81	64.05
	128k	97.34	97.85	97.94	97.54	97.94		128k	31.72	32.30	32.49	32.81	33.56
WE	4k	93.40	92.92	93.04	93.70	93.08	WE	4k	86.88	86.95	87.13	86.87	87.62
	8k	68.08	71.56	71.48	71.26	70.78		8k	85.44	86.28	86.22	85.90	86.48
	16k	49.66	50.42	50.32	51.02	51.73		16k	83.45	83.05	83.66	84.80	84.56
	32k	41.31	41.93	41.71	42.06	42.65		32k	78.00	78.06	78.13	79.16	79.44
	64k	39.08	39.83	39.71	41.24	44.64		64k	47.75	49.15	48.17	48.34	49.15
	128k	30.54	32.48	31.80	31.90	32.73		128k	40.71	42.49	43.18	44.26	44.57
QA	4k	62.31	62.13	62.01	65.42	66.03	QA	4k	64.15	65.15	64.87	64.52	64.78
	8k	57.20	57.54	57.53	57.69	57.28		8k	58.28	59.18	59.32	59.23	60.27
	16k	53.57	54.40	54.80	53.98	54.14		16k	57.13	58.84	58.20	59.14	59.33
	32k	48.24	49.32	49.12	49.54	49.88		32k	54.66	57.04	56.95	55.81	56.72
	64k	42.49	43.59	43.50	44.06	44.62		64k	41.44	42.75	43.10	42.94	43.86
	128k	38.64	40.23	40.38	41.06	41.34		128k	40.48	41.43	41.28	42.56	42.93
VT	4k	90.03	92.88	92.40	92.54	93.40	VT	4k	100.00	100.00	100.00	100.00	100.00
	8k	85.24	88.52	88.28	87.68	89.52		8k	99.93	99.95	99.93	99.95	99.95
	16k	82.92	87.64	87.36	86.76	88.52		16k	99.60	99.70	99.64	99.60	99.70
	32k	84.08	85.67	85.75	86.32	90.44		32k	98.19	98.79	99.02	99.10	98.95
	64k	87.36	87.38	88.02	88.04	88.55		64k	74.12	77.24	77.61	78.46	78.92
	128k	68.44	71.20	70.76	70.92	72.20		128k	35.04	37.30	37.16	38.87	38.72

Comparison. Both TRONTOR and TRONTOR-Agg achieve strong overall performance across different model scales and context lengths. The router-only TRONTOR variant yields higher aggregate performance than the vanilla Baseline for all four backbone models without requiring multi-base intra-head aggregation. Its improvements are partic-

ularly pronounced in long-context settings, including the VT tasks for the Llama-3.2 models, while it generally maintains or improves performance at shorter context lengths (4k and 8k). Using the three-seed mean for each setting, two-sided paired *t*-tests over the 24 matched task–length settings confirm that the aggregate improvement of TRON-

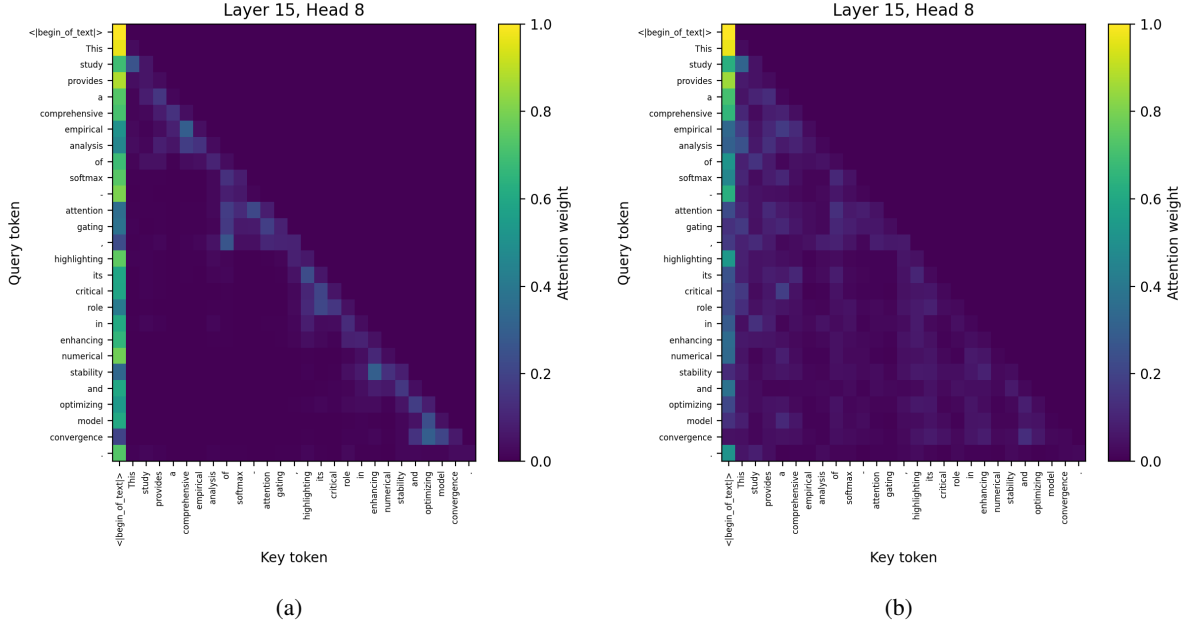


Figure 2: Visualization of attention-sink mitigation through dynamic routing in TRONTOR. (a) The attention map under standard RoPE. (b) The attention map under TRONTOR’s token-wise dynamic routing.

TOR over the Baseline is statistically significant for all four backbones ($p < 0.05$). Moreover, TRONTOR achieves aggregate performance competitive with MoICE, indicating that dynamic switching between RoPE and NoPE can itself provide an effective mechanism for long-context adaptation. For intra-head aggregation, the AVG ablation performs comparably to MoICE across most tasks, suggesting that simple unweighted averaging of attention outputs obtained with different RoPE bases can provide performance comparable to learned weighting in the evaluated settings. Building on this observation, TRONTOR-Agg combines dynamic RoPE/NoPE routing with intra-head aggregation and provides further performance gains over TRONTOR. Under the same paired testing protocol, its aggregate improvements over TRONTOR, the Baseline, MoICE, and AVG are statistically significant for all four backbone models ($p < 0.05$). Because TRONTOR activates only one positional-encoding path for each head–token pair, it avoids evaluating the multiple RoPE attention branches used by MoICE. In our efficiency analysis using Llama-3.2-1B with a 32k-token input, TRONTOR increases the total decode time for 20 generated tokens by only approximately 5% relative to the Baseline, while requiring approximately 40% less decode time than MoICE. Although TRONTOR-Agg performs additional multi-base aggregation, it still requires 33% less decode time than MoICE.

Beyond RULER, TRONTOR-Agg remains competitive on LongBench and standard short-context language-understanding benchmarks, suggesting that its long-context improvements do not cause systematic degradation on shorter-context tasks. Detailed results are provided in Appendix B and C, respectively.

5.3 Mitigating attention sinks

Figure 2 visualizes the attention maps of Layer 15, Head 8 in the Llama-3.2-3B model. In Figure 2(a), the original RoPE-based head concentrates its attention on the initial tokens as the sequence length grows, causing mid-context tokens to receive less attention. By contrast, Figure 2(b) shows the same head under token-wise dynamic routing, with query positions selected by the router operating as NoPE. In this case, attention is redistributed away from the sequence prefix and toward semantically relevant tokens located in the middle of the context.

We also measured the average attention probability mass allocated to the initial tokens (the first 1, 5, and 10 tokens) across all layers for the heads designated as NoPE. To ensure robustness, these measurements were averaged over multiple random seeds and 10 diverse input sequences, covering domains such as instruction-following, code, and dialogue. We compared the attention weights of these heads when operating as standard RoPE versus when routed to NoPE. As summarized in Table 3, we observed a clear mitigation of the attention

sink across different initial token windows, with an absolute attention weight reduction of 47.54% for the first token and 40.06% for the first 5 tokens. This evidence supports our claim that NoPE routing effectively mitigates the attention sink and redistributes attention capacity to the middle tokens.

Table 3: Comparison of average attention probability mass allocated to the initial tokens when heads operate as RoPE versus when routed to NoPE.

Initial Token Window	RoPE (%)	NoPE (%)	Absolute Reduction (%)
First 1	79.81	32.27	47.54
First 5	80.74	40.68	40.06
First 10	81.22	50.27	30.95

6 Conclusion

In this paper, we introduce TRONTOR, a framework that addresses the limitations of “always-on” static positional encodings by employing a lightweight, context-aware router. Moving beyond conventional layer-level modifications, TRONTOR explores the potential of adaptively switching between RoPE and NoPE at a fine-grained, head level. Empirical evaluations demonstrate that this head-wise dynamic routing alone improves performance over the vanilla baseline with minimal overhead and remains competitive with existing multi-scale approaches. When additional computation is available, TRONTOR-Agg further improves performance by augmenting RoPE-active heads with simple unweighted intra-head aggregation. The analysis of our results indicates that TRONTOR effectively mitigates the attention-sink phenomenon and redistributes attention capacity to enhance mid-context sensitivity. To support reproducibility, our implementation is available at <https://github.com/mcrl/TRONTOR>.

Limitations

Although TRONTOR improves long-context modeling, several limitations remain. First, our work focuses on head-wise positional adaptation and operates on top of the frequency scaling strategy provided by each backbone. We therefore do not directly compare against or replace frequency-domain scaling methods such as NTK-aware scaling and LongRoPE. Since these approaches are largely orthogonal to head-wise routing, integrating TRONTOR with more advanced frequency scaling methods remains an important direction for future

work. Second, we initialize the NoPE routing ratio to 5% to prevent premature collapse toward excessive NoPE activation. Although the final routing behavior is learned dynamically and the method remains robust across a range of low initialization ratios (below 10%), a broader sensitivity analysis across model scales, tasks, and routing sparsity regimes is needed to determine the optimal initialization strategy.

Acknowledgements

This work was partially supported by the National Research Foundation of Korea (NRF) under Grant No. RS-2023-00222663 (Center for Optimizing Hyperscale AI Models and Platforms) and under Grant No. A400-20260031, and by the Institute for Information and Communications Technology Promotion (IITP) under Grant No. 2018-0-00581 (CUDA Programming Environment for FPGA Clusters) and No. RS-2025-02304554 (Efficient and Scalable Framework for AI Heterogeneous Cluster Systems), all funded by the Ministry of Science and ICT (MSIT) of Korea. It was also partially supported by the Korea Health Industry Development Institute (KHIDI) under Grant No. RS-2025-25454559 (Frailty Risk Assessment and Intervention Leveraging Multimodal Intelligence for Networked Deployment in Community Care), funded by the Ministry of Health and Welfare (MOHW) of Korea. Additional support was provided by the BK21 Plus Program for Innovative Data Science Talent Education (Department of Data Science, Seoul National University, No. 5199990914569) and the BK21 FOUR Program for Intelligent Computing (Department of Computer Science and Engineering, Seoul National University, No. 4199990214639), both funded by the Ministry of Education (MOE) of Korea. This work was also partially supported by the Advanced GPU Utilization Support Program, funded by the Ministry of Science and ICT (MSIT) of Korea and operated by the National IT Industry Promotion Agency (NIPA), as well as by MSIT through NIPA as part of the Sovereign AI Foundation Model Project. It was also carried out under the project’s Data Track, led by MSIT and supported by the National Information Society Agency (NIA). Research facilities were provided by the Institute of Computer Technology (ICT) at Seoul National University.

Ethical Considerations

This study uses publicly available models and dataset. Generative AI tools were used solely for grammatical correction. All scientific content, analyses, and conclusions were developed and verified by the authors. We have also reviewed broader ethical implications, such as potential misuse and inherent model biases, and conclude that our architectural modifications do not introduce new risks beyond those already present in the foundational models.

References

- Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.
- Kyunghoon Bae, Eunbi Choi, Kibong Choi, Stanley Jungkyu Choi, Yemuk Choi, Kyubeen Han, Seokhee Hong, Junwon Hwang, Taewan Hwang, Joonwon Jang, et al. 2025. Exaone 4.0: Unified large language models integrating non-reasoning and reasoning modes. *arXiv preprint arXiv:2507.11407*.
- Yushi Bai, Xin Lv, Jiajie Zhang, Yuze He, Ji Qi, Lei Hou, Jie Tang, Yuxiao Dong, and Juanzi Li. 2024a. Longalign: A recipe for long context alignment of large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 1376–1395.
- Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, et al. 2024b. Longbench: A bilingual, multitask benchmark for long context understanding. In *Proceedings of the 62nd annual meeting of the association for computational linguistics (volume 1: Long papers)*, pages 3119–3137.
- Yoshua Bengio, Nicholas Léonard, and Aaron Courville. 2013. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*.
- Shouyuan Chen, Sherman Wong, Liangjian Chen, and Yuandong Tian. 2023. Extending context window of large language models via positional interpolation. *arXiv preprint arXiv:2306.15595*.
- Yuhan Chen, Ang Lv, Ting-En Lin, Changyu Chen, Yuchuan Wu, Fei Huang, Yongbin Li, and Rui Yan. 2024. Fortify the shortest stave in attention: Enhancing context awareness of large language models for effective tool use. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11160–11174.
- Yuhan Chen, Ang Lv, Jian Luan, Bin Wang, and Wei Liu. 2025. Hope: A novel positional encoding without long-term decay for enhanced context awareness and extrapolation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 23044–23056.
- Ting-Rui Chiang and Dani Yogatama. 2025. The rotary position embedding may cause dimension inefficiency in attention heads for long-distance retrieval. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 13552–13562.
- Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.
- Yiran Ding, Li Lina Zhang, Chengruidong Zhang, Yuanyuan Xu, Ning Shang, Jiahang Xu, Fan Yang, and Mao Yang. 2024. LongRoPE: Extending LLM context window beyond 2 million tokens. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 11091–11104. PMLR.
- Stefan Elfving, Eiji Uchibe, and Kenji Doya. 2018. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural networks*, 107:3–11.
- Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, et al. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*.
- Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.
- Xiangming Gu, Tianyu Pang, Chao Du, Qian Liu, Fenghuo Zhang, Cunxiao Du, Ye Wang, and Min Lin. 2025. When attention sink emerges in language models: An empirical view. In *International Conference on Learning Representations*, volume 2025, pages 97114–97144.
- Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, Yang Zhang, and Boris Ginsburg. 2024. Ruler: What’s the real context size of your long-context language models? *Preprint*, arXiv:2404.06654.
- Ermo Hua, Che Jiang, Xingtai Lv, Kaiyan Zhang, Youbang Sun, Yuchen Fan, Xuekai Zhu, Biqing Qi, Ning Ding, and Bowen Zhou. 2025. Fourier position embedding: Enhancing attention’s periodic extension for length generalization. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 24932–24949. PMLR.

- Amirhossein Kazemnejad, Inkit Padhi, Karthikeyan Natesan Ramamurthy, Payel Das, and Siva Reddy. 2023. The impact of positional encoding on length generalization in transformers. *Advances in Neural Information Processing Systems*, 36:24892–24928.
- Jiaqi Li, Mengmeng Wang, Zilong Zheng, and Muhan Zhang. 2024. Loogle: Can long-context language models understand long contexts? In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16304–16333.
- Haoran Lian, Junmin Chen, Wei Huang, Yizhe Xiong, Wenping Hu, Guiguang Ding, Hui Chen, Jianwei Niu, Zijia Lin, Fuzheng Zhang, et al. 2025. Breaking the stage barrier: A novel single-stage approach to long context extension for large language models. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 4897–4909.
- Hongzhan Lin, Ang Lv, Yuhan Chen, Chen Zhu, Yang Song, Hengshu Zhu, and Rui Yan. 2024. Mixture of in-context experts enhance llms’ long context awareness. *Advances in Neural Information Processing Systems*, 37:79573–79596.
- Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024a. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173.
- Xiaoran Liu, Hang Yan, Chenxin An, Xipeng Qiu, and Dahua Lin. 2024b. Scaling laws of rope-based extrapolation. In *International Conference on Learning Representations*, volume 2024, pages 48160–48185.
- Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pages 2381–2391.
- Andrew Nam, Henry Conklin, Yukang Yang, Tom Griffiths, Jonathan D Cohen, and Sarah-Jane Leslie. 2025. Causal head gating: A framework for interpreting roles of attention heads in transformers. In *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pages 115609–115631. Curran Associates, Inc.
- Bowen Peng and Jeffrey Quesnelle. 2023. Ntk-aware scaled rope allows llama models to have extended (8k+) context size without any fine-tuning and minimal perplexity degradation. https://www.reddit.com/r/LocalLLaMA/comments/14lz7j5/ntkaware_scaled_rope_allows_llama_models_to_have/.
- Bowen Peng, Jeffrey Quesnelle, Honglu Fan, and Enrico Shippole. 2024. Yarn: Efficient context window extension of large language models. In *International Conference on Learning Representations*, volume 2024, pages 31932–31951.
- Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106.
- Ning Shang, Li Lyna Zhang, Siyuan Wang, Gaokai Zhang, Gilsinia Lopez, Fan Yang, Weizhu Chen, and Mao Yang. 2025. LongRoPE2: Near-lossless LLM context window scaling. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 54203–54218. PMLR.
- Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. 2024. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063.
- Teknum. 2023. Openhermes 2.5: An open dataset of synthetic data for generalist llm assistants.
- Jie Wang, Tao Ji, Yuanbin Wu, Hang Yan, Tao Gui, Qi Zhang, Xuan-Jing Huang, and Xiaoling Wang. 2024. Length generalization of causal transformers without position encoding. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 14024–14040.
- Shenqi Wang and Guangzhi Tang. 2025. Context-aware sparse spatiotemporal learning for event-based vision. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 13713–13719. IEEE.
- Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. 2024. Efficient streaming language models with attention sinks. In *International Conference on Learning Representations*, volume 2024, pages 21875–21895.
- An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. 2025a. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.
- Bowen Yang, Bharat Venkitesh, Dwaraknath Gnaneshwar Talupuru, Hangyu Lin, David Cairuz, Phil Blunsom, and Acyr Locatelli. 2025b. Rope to nope and back again: A new hybrid attention strategy. In *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pages 64133–64157. Curran Associates, Inc.
- Yukang Yang, Declan Iain Campbell, Kaixuan Huang, Mengdi Wang, Jonathan D. Cohen, and Taylor Whittington Webb. 2025c. Emergent symbolic mechanisms support abstract reasoning in large language models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 70515–70549. PMLR.
- Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pages 4791–4800.

Zhenyu Zhang, Runjin Chen, Shiwei Liu, Zhewei Yao, Olatunji Ruwase, Beidi Chen, Xiaoxia Wu, Zhangyang Wang, et al. 2024. Found in the middle: How language models use long contexts better via plug-and-play positional encoding. *Advances in Neural Information Processing Systems*, 37:60755–60775.

A Manual Head Selection Strategies

In preliminary analysis, we explore two distinct strategies to identify target heads for RoPE removal: Position-Awareness-Guided selection and Random selection.

Position-Awareness-Guided Selection. Motivated by evidence that attention heads exhibit varying sensitivities to positional information, we adopt the head-wise Position Awareness Score (PAS) proposed by Zhang et al. (2024). PAS serves as a proxy for positional sensitivity by quantifying the fraction of tokens whose attention scores exceed a threshold relative to the mean attention score. Formally, the position-awareness score $\mathcal{S}_P \in \mathbb{R}^H$ is defined as:

$$\mathcal{S}_P = \frac{1}{l} \sum_{i=1}^l \left(A_i \geq \alpha \frac{1}{l} \sum_{j=1}^l A_j \right) \quad (8)$$

where A_i denotes the attention score assigned to the i -th token for the most recent query, l is the sequence length, and α is a threshold hyperparameter (fixed at $\alpha = 3$ following prior work). Intuitively, a higher $\mathcal{S}_P$ indicates stronger positional sensitivity. In our experiments, we identify heads with the lowest PAS values and convert them to NoPE, allowing us to assess the impact of removing RoPE from heads that empirically encode positional information weakly.

Random Selection. As a control condition, we evaluate a random head-selection strategy where heads are converted to NoPE independently of any positional metric. Specifically, within a targeted layer, we randomly sample a subset of heads for conversion. This baseline allows us to disentangle the effects of principled head selection from those arising merely from a general reduction of positional bias.

B Computational Analysis

For a more precise inference-efficiency analysis, we measured Peak VRAM, Prefill Time, and Decode Time using a randomly generated input of 32k

Metric	Baseline	TRONTOR	TRONTOR-Agg	MolCE
Peak VRAM (GB)	13.26	15.02	18.36	21.22
Prefill Time (ms)	329.66	352.74	569.92	741.74
Total Decode Time (20 tokens, ms)	5158.07	5429.55	6034.69	8952.23

Table 4: Inference efficiency comparison at 32k context length. Decode time is measured for generating 20 tokens after the prefill phase.

Task	Llama-3.2-1B-Instruct			Llama-3.2-3B-Instruct			Qwen3-1.7B			Qwen3-4B		
	Baseline	MoICE	TRONTOR-Agg	Baseline	MoICE	TRONTOR-Agg	Baseline	MoICE	TRONTOR-Agg	Baseline	MoICE	TRONTOR-Agg
Single-Document QA	25.17	25.69	25.94	37.32	38.16	38.84	13.40	14.28	14.35	15.64	16.58	17.20
Multi-Document QA	25.43	26.16	27.14	24.42	26.40	27.52	13.68	15.04	15.62	16.86	18.31	19.60
Summarization	20.75	21.12	22.05	24.19	24.46	26.13	20.37	21.80	22.91	24.35	24.69	25.35
Few-shot Learning	51.95	52.35	52.89	60.16	61.04	61.78	54.25	56.75	57.06	59.70	60.24	61.36
Synthetic Tasks	3.41	6.46	7.71	40.17	41.93	43.67	58.72	60.19	61.75	63.75	64.37	65.14
Code Completion	26.20	26.38	27.19	33.08	35.23	36.12	6.59	8.46	9.32	12.83	15.06	14.96

Table 5: Evaluation results on the LongBench benchmark across Llama-3.2 and Qwen3 models.

tokens on the Llama-3.2-1B model. The decode time specifically reflects the latency for generating 20 tokens subsequent to the 32k token prefill phase.

First, to isolate the computational cost of our components, we evaluate a TRONTOR variant, which applies our context-aware routing to select between default RoPE and NoPE per head, without any intra-head aggregation. As shown in Table 4, TRONTOR introduces a marginal overhead compared to the vanilla Baseline, with less than an 8% increase in prefill time and minimal impact on decode latency and memory. This demonstrates that the dynamic head routing mechanism itself is highly lightweight and efficient.

The primary computational overhead in TRONTOR-Agg arises from the intra-head aggregation phase, which necessitates the computation and blending of three distinct RoPE bases used in this experiment. Although this aggregation introduces an increase in cost compared to the Baseline, it is a worthwhile trade-off for the substantial performance improvements achieved in long-context scenarios. More importantly, when compared to MoICE, TRONTOR-Agg proves to be highly efficient. Our method achieves an execution speedup of over 30% in total decode time, alongside a substantial reduction of approximately 15% in peak VRAM consumption. Optimizing the low-level kernels for the intra-head aggregation to further accelerate inference speed remains an important direction for future work.

C Additional Benchmark Results

While our primary evaluation focuses on the RULER benchmark to demonstrate long-context capabilities, we provide supplementary results in this section to offer a comprehensive performance assessment. Specifically, we evaluate the LLaMA-based implementation of TRONTOR-Agg on LongBench (Bai et al., 2024b), another standard long-context benchmark, as well as on several general language understanding tasks that handle short-context documents. LongBench categorizes its de-

tailed sub-tasks into six main areas, and we report the average performance for each: Code Completion (*longbench_lcc*, *longbench_repobench-p*), Few-shot Learning (*longbench_lsht*, *longbench_samsum*, *longbench_trec*, *longbench_triviaqa*), Multi-Document QA (*longbench_2wikimqa*, *longbench_dureader*, *longbench_hotpotqa*, *longbench_musique*), Single-Document QA (*longbench_multifieldqa_en*, *longbench_multifieldqa_zh*, *longbench_narrativeqa*, *longbench_qasper*), Summarization (*longbench_gov_report*, *longbench_multi_news*, *longbench_qmsum*, *longbench_vcsun*), and Synthetic Tasks (*longbench_passage_count*, *longbench_passage_retrieval_en*, *longbench_passage_retrieval_zh*).

Task	Llama-3.2-1B-Instruct			Llama-3.2-3B-Instruct		
	Baseline	MoICE	TRONTOR-Agg	Baseline	MoICE	TRONTOR-Agg
HellaSwag	45.09	45.12	45.45	53.29	54.08	54.08
OpenBookQA	24.40	24.60	24.60	29.00	29.47	30.20
WinoGrande	59.67	59.75	59.78	68.82	69.12	69.05
ARC-Easy	68.43	68.27	68.43	75.38	75.80	76.27
ARC-Challenge	35.75	35.41	35.72	43.77	43.94	44.33

Table 6: Evaluation results on General Tasks for standard language understanding across Llama-3.2 1B and 3B models.

The reported score for each category represents the average performance of its corresponding sub-tasks. As shown in Table 5, TRONTOR-Agg consistently improves over the Baseline and outperforms MoICE across nearly all grouped LongBench tasks. This confirms that the retrieval improvements observed in RULER stably generalize to diverse long-context evaluation scenarios. We also demonstrate that TRONTOR-Agg effectively maintains, and in some cases marginally improves, performance even on tasks involving short documents. Table 6 presents the evaluation results on standard general language understanding benchmarks, including HellaSwag (Zellers et al., 2019), OpenBookQA (Mihaylov et al., 2018), WinoGrande (Sakaguchi et al., 2021), ARC-Easy, and ARC-Challenge (Clark et al., 2018). Compared to the baseline, TRONTOR-Agg not only preserves

fundamental reasoning skills without catastrophic forgetting but also exhibits slight but consistent performance gains across most tasks. These findings validate that TRONTOR-Agg robustly enhances long-context capabilities while safely preserving general language understanding in shorter contexts.

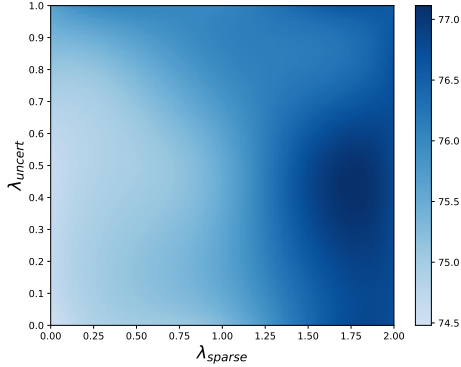


Figure 3: Hyperparameter search results for λ_{sparse} and λ_{uncert} . The color intensity represents the model score, with the optimal performance achieved at $\lambda_{sparse} = 1.75$ and $\lambda_{uncert} = 0.5$.

D Hyperparameter Search for Auxiliary Loss

To determine the optimal coefficient values that achieve the balance between sparsity and performance, we performed a grid search for the auxiliary loss. We evaluated λ_{sparse} over $[0, 2]$ with a 0.25 interval and λ_{uncert} over $[0, 1]$ with a 0.1 interval. For this analysis, we utilized the Llama-3.2-1B-Instruct model with an 8k context length on the RULER Variable Tracking (VT) task. As mentioned in the main text, our training objective was to maintain the proportion of NoPE heads at approximately 5%, which means typically only 0 to 2 heads per layer are activated as NoPE heads. The complete results of this grid search are presented as a heatmap in Figure 3.

Model performance across most configurations surpassed the baseline score of 74.12. A distinct performance peak was observed at $\lambda_{sparse} = 1.75$ and $\lambda_{uncert} = 0.5$. Although the parameter search was restricted to the 1B model and the VT task due to resource limitations, we empirically confirmed that these values yield robust performance across different model scales and various evaluation scenarios.

E System configuration

Detailed system configurations are provided in Table 7.

CPU	2 × AMD EPYC 9365 36-Core Processor
GPU	4 × NVIDIA B200
OS	Ubuntu 24.04.3 LTS
CUDA Version	13.0
PyTorch Version	2.11.0+cu128

Table 7: System configuration