T-ROPE: TIME-AWARE ROTARY POSITION EMBEDDING FOR SEQUENTIAL RECOMMENDATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Large-scale recommenders increasingly adopt the sequential generative recipe behind large language models, bringing the Transformer into recommendation along with design choices made for text, including Rotary Position Embedding (RoPE). In language models, RoPE encodes token indices for relative position reasoning, but in recommendation, an interaction index records only event order, saying nothing about elapsed time, behavioral cycles across scales, or calendar phase. We revisit this choice and propose T-RoPE, a time-aware RoPE for sequential generative recommendation that replaces index-only rotation with timestamp-based angles, learnable temporal coefficients, multiscale frequency banks, shifted query alignment, and non-stationary key rotation. We prove that standard RoPE, even on timestamps, remains time-translation invariant and cannot distinguish seasonal contexts, and that T-RoPE breaks this invariance while preserving the RoPE interface. Across five public benchmarks, T-RoPE achieves the best result on every metric on every dataset, improving over the strongest baseline by 78–130% in HR@10 on the sparse PixelRec data and 8–12% across metrics on Amazon Books. On an industrial-scale e-commerce dataset with more than 6B interactions, it improves every metric over the HSTU + Time RAB backbone by 13–82%, with ablations attributing the largest gains to multiscale frequencies ($+56\%$ NDCG@50) and non-stationary keys ($+4\%$). An online A/B test in the Shop app yields positive lifts in conversion rate ($+0.33\%$) and order count ($+0.63\%$). We also provide forward and backward algorithms whose added cost is linear in sequence length and head dimension, keeping time-aware RoPE practical for large generative recommenders.

1 INTRODUCTION

Large language models have made autoregressive sequence modeling a practical way to learn from long, heterogeneous histories (Vaswani et al., 2017; Touvron et al., 2023). Their success comes not from the architecture alone but from a mature engineering stack of scalable training, fast attention kernels, and long-context modeling that makes generative prediction practical at scale. Recommendation models are now borrowing this recipe. Recent sequential generative recommenders represent user behavior as action sequences and autoregressively predict future items, actions, or semantic identifiers (Rajput et al., 2023; Wang et al., 2024; Chen et al., 2024; Hou et al., 2025; OneRec Team et al., 2026; Liang et al., 2026). This recipe has already proven itself at production scale, where HSTU drives live recommendation systems on industrial workloads (Zhai et al., 2024).

This transfer is productive but imperfect, because inductive biases that serve language do not always carry over to recommendation. Generative recommenders inherit components from language models, including RoPE (Su et al., 2024; Tian et al., 2026; Liu et al., 2026b; Movahedi et al., 2026). RoPE injects sequence order by rotating query and key vectors according to an integer position index. For text, this is a natural coordinate, because token offsets carry much of the structure attention needs. For recommendation, the same coordinate loses what matters. An interaction index records that one event preceded another, but not whether the gap was seconds, weeks, or seasons.

This positional encoding gap leaves three temporal signals unrepresented. First, *true time gaps* are erased. A key six months in the past can be positionally identical to a key from yesterday if their sequence offsets match. Second, *multiscale periodicity* is absent. User behavior contains hourly, weekly, monthly, and seasonal rhythms, but index-based RoPE has no wall-clock frequency structure.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Table 1: Temporal mechanisms explicitly provided by positional encoding methods.

Method	Timestamp	Learnable	Multiscale	Shifted	Non-stationary
Standard RoPE (Su et al., 2024)	×	×	×	×	×
TiSASRec (Li et al., 2020)	Partial	×	×	×	×
TO-RoPE (Wei et al., 2025)	✓	Partial	✓	×	×
T-RoPE (ours)	✓	✓	✓	✓	✓

Multiscale denotes an explicit continuous frequency bank covering multiple periods. Learnable denotes learned temporal coefficients inside the encoding (not generic model parameters). Partial denotes a limited or indirect version of the property.

Third, *absolute calendar phase* is inaccessible. Because index-based attention depends only on the relative offset $m - n$, it is time-translation invariant. The same history evaluated at different points in a seasonal cycle produces the same attention pattern.

Existing temporal recommendation methods close only part of this gap, as Table 1 summarizes. TiSASRec injects discretized time-interval features into self-attention, which captures recency but not continuous multiscale periodicity or calendar phase (Li et al., 2020). HSTU adds a time-based Relative Attention Bias (RAB), mapping bucketed time gaps to a learned scalar added to the attention logits, which sharpens recency weighting but leaves the query and key geometry untouched and still discretizes time (Zhai et al., 2024). TO-RoPE brings timestamps into RoPE angles, but the joint geometry of its queries and keys remains time-translation invariant (Wei et al., 2025). As we prove in Section 4, it captures temporal distance but cannot distinguish queries issued in different calendar phases over the same history. Time cannot just sit alongside RoPE as an extra feature or bias, because closing the gap requires structurally changing the way RoPE itself rotates queries and keys.

To address this gap, we propose **T-RoPE** (**T**ime-aware **RoPE**), a RoPE-compatible positional encoding for timestamped user histories. T-RoPE keeps the efficient RoPE interface but replaces index-only rotation with timestamp-based angles, learnable temporal coefficients, multiscale frequency banks, shifted query alignment, and non-stationary key rotation, as shown in Table 1. We list our main contributions below.

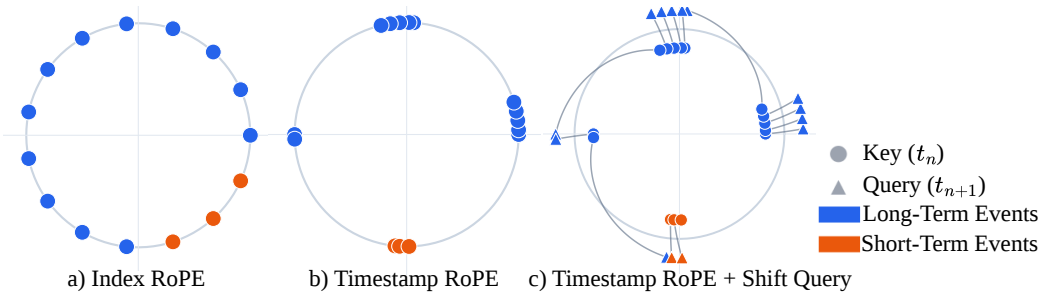
1. **A time-aware RoPE for generative recommendation.** T-RoPE encodes continuous timestamps directly in rotary attention to represent elapsed time, multiscale periodicity, and calendar phase within a single rotary encoding. Index-based RoPE discards these signals, and prior temporal encodings capture them only in part.
2. **Theoretical analysis.** We prove that rotating queries and keys by the same function of their timestamps yields time-translation invariant attention (covering standard RoPE on timestamps and TO-RoPE's timestamp component), and that T-RoPE's non-stationary key provably breaks this invariance.
3. **Results across settings.** On five public benchmarks, an industrial dataset with more than 6B interactions, and an online A/B test in the Shop app, T-RoPE improves over temporal and RoPE-based baselines on every public benchmark and delivers positive production lifts.
4. **An efficient implementation path.** We provide forward and backward algorithms and a cost analysis showing only linear overhead in sequence length and head dimension.

2 PRELIMINARIES

In generative recommendation, the model input is a user's interaction history, a sequence of timestamped events $\{(i_1, t_1), \ldots, (i_L, t_L)\}$ ordered by time $t_1 < t_2 < \cdots < t_L$. Each event contains an item i_m and a Unix timestamp $t_m \in \mathbb{R}_{>0}$ measured in seconds. It may also include an interaction type or other semantic identifiers. The model predicts the next event autoregressively, and we take the prediction target to be the next item i_{L+1}.

A Transformer encoder maps each item i_m to query, key, and value vectors $\mathbf{q}_m, \mathbf{k}_m, \mathbf{v}_m \in \mathbb{R}^D$. Standard RoPE (Su et al., 2024) acts on $D/2$ 2D rotation planes. In plane $d \in \{1, \ldots, D/2\}$, the 2D slice $[\mathbf{q}_{m,2d-1}, \mathbf{q}_{m,2d}]^\top$ is multiplied by a rotation matrix

$$\mathbf{R}(\phi_m) = \begin{bmatrix} \cos\phi_m & -\sin\phi_m \\ \sin\phi_m & \cos\phi_m \end{bmatrix}. \tag{1}$$



a) Index RoPE b) Timestamp RoPE c) Timestamp RoPE + Shift Query

Figure 1: Geometric intuition for T-RoPE rotations. a) Index RoPE places events at uniform sequence positions, hiding elapsed time. b) Timestamp RoPE uses event time, so bursty events cluster and dormant gaps spread on the rotation circle. c) Shifted queries place each query at the next-event timestamp (markers displaced from keys), matching training to the inference-time geometry.

The resulting dot product between rotated query and key in plane d is

$$(\mathbf{q}'_{m,d})^\top \mathbf{k}'_{n,d} = (\mathbf{q}_{m,d})^\top \mathbf{R}(\phi_n - \phi_m)\mathbf{k}_{n,d}, \tag{2}$$

which depends only on the angle difference $(\phi_n - \phi_m)$. Integer indices set $\phi_m = m\theta_d$, where θ_d is the fixed angular frequency of plane d. So the angle difference becomes $(n-m)\theta_d$ and the dot product depends only on the relative offset $n - m$.

3 The T-RoPE Method

T-RoPE keeps RoPE's computational interface but changes the meaning of the rotation, treating the user history as a timestamped event sequence rather than a text-like list. The construction has five parts: real-time rotation, learned per-layer scale coefficients, multiple temporal scales across dimensions, query alignment to the prediction time, and a key rotation that breaks the relative-time symmetry hiding calendar phase.

3.1 Timestamp-Based Rotation

Adjacent sequence positions are not equally spaced in time. In recommendation, neighboring events can be seconds, days, or months apart, yet the index m treats every gap as an offset of one position (Figure 1a). T-RoPE therefore replaces the integer index with the absolute Unix timestamp t_m in seconds, used directly before rotation without centering or normalizing. As Figure 1b shows, events close in sequence index but far apart in time no longer occupy neighboring phases, while bursty events stay clustered. For dimension d, the angle is $\phi_{m,d} = t_m/\beta_d$, where β_d is a base time frequency. Under this substitution the relative phase between two events becomes a function of their true temporal gap,

$$(\mathbf{q}'_{m,d})^\top \mathbf{k}'_{n,d} \propto \cos\left(\frac{t_n - t_m}{\beta_d}\right), \tag{3}$$

so a large gap $|t_n - t_m|$ weakens attention even between adjacent events in the truncated history.

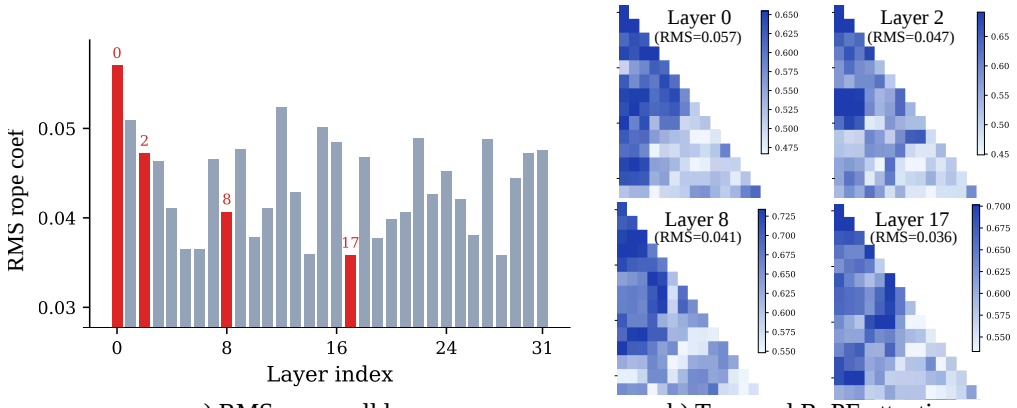
3.2 Learnable Temporal Coefficients

Not all temporal scales deserve equal weight. Dominant cycles differ by domain and across layers within one model. Grocery purchases may follow cycles of 7 or 30 days, fashion demand may be seasonal, and marketplace activity may follow payday or promotion intervals. T-RoPE learns the strength of each cycle rather than fixing it.

To learn these strengths, T-RoPE introduces learnable coefficients $\alpha^q_d, \alpha^k_d \in \mathbb{R}$, separate for queries and keys,

$$\phi^q_{m,d} = \alpha^q_d \cdot \frac{t_m}{\beta_d}, \qquad \phi^k_{n,d} = \alpha^k_d \cdot \frac{t_n}{\beta_d}. \tag{4}$$

Independent query and key coefficients permit asymmetric specialization, but in practice we find the opposite. Across most dimensions the learned query and key coefficients align in both sign and



a) RMS across all layers b) Temporal RoPE attention

Figure 2: Layer-wise visualization of T-RoPE. a) Per-layer RMS magnitude of learned query coefficients, with the selected layers highlighted. b) Temporal RoPE attention components for selected layers on a synthetic event sequence. Larger RMS values correspond to stronger temporal modulation.

magnitude, as Figure 3a shows, so both emphasize the same temporal bands. This alignment does not carry across layers, however. Figure 2 shows the per-layer RMS coefficient magnitude varying widely (a), and the temporal attention on the same sequence shifting substantially across the selected layers (b), so each layer learns its own temporal sensitivity rather than a single shared bias.

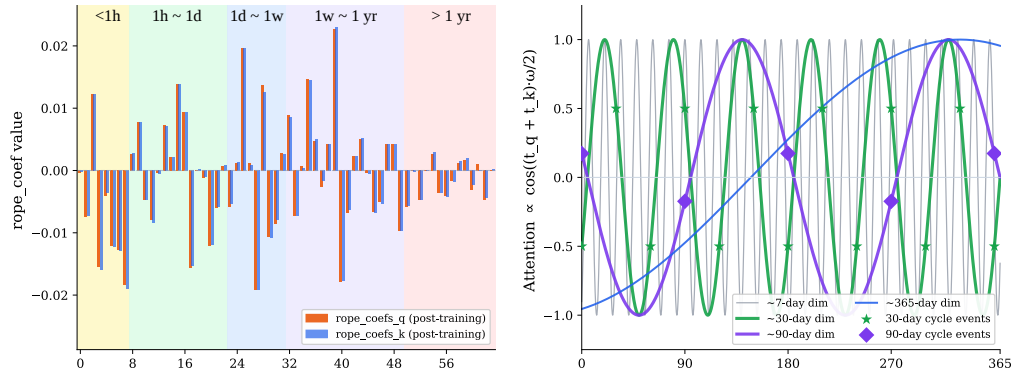
3.3 Multiscale Frequency Bank

A single base frequency resolves only one time scale, but user behavior spans many, from browsing sessions over minutes and household purchases over weeks or months to seasonal demand over a year. Capturing behavior across these scales requires several frequencies at once.

T-RoPE assigns a distinct base frequency β_d to each rotation plane d using geometric spacing,

$$\beta_d = \exp\left[\log \beta_{\min} + \frac{d-1}{D/2-1}\left(\log \beta_{\max} - \log \beta_{\min}\right)\right], \quad d = 1, \ldots, D/2, \tag{5}$$

where $\beta_{\min}$ and $\beta_{\max}$ are hyperparameters (defaults 10^2 and 10^8 seconds), giving a logarithmic bank from 10^{-2} to 10^{-8} radians/second. The bank fixes the available frequencies, and the learnable coefficients from Section 3.2 set how strongly each one contributes. As Figure 3 shows, even within a single layer training upweights a few temporal bands rather than all frequencies uniformly (a), and different frequencies lock onto different user cycles (b). The learned coefficients adjust the multiscale



a) Geomspaced Base Frequencies + Learned Coefficients b) Per-Dimension Resonance for Periodic Events

Figure 3: Multigranular temporal encoding. a) Learned query and key temporal coefficients over frequency dimensions; shaded regions group dimensions by time period, from subhour to multiyear scales. Training upweights selected temporal bands rather than all frequencies uniformly. b) Representative resonance curves for those bands, where short-period dimensions oscillate rapidly while 30-day, 90-day, and yearly dimensions align with recurring user cycles.

frequency bank so that the timestamp encoding can represent periodic patterns without explicitly encoding named calendar events such as Christmas.

3.4 Shifted Query Alignment

The rotation so far encodes when each history event occurred but not when the predicted event will occur. Since position m is supervised to predict the event at t_{m+1} (and a deployed model receives the current time t_{now} at inference), we adopt HSTU's RAB Zhai et al. (2024) and rotate the query at this target time rather than at the history time. As illustrated in Figure 1c, T-RoPE assigns t_{m+1} to the query at position m while keys keep their own timestamps,

$$\phi^q_{m,d} = \alpha^q_d \frac{t_{m+1}}{\beta_d}, \quad \phi^k_{n,d} = \alpha^k_d \frac{t_n}{\beta_d}, \tag{6}$$

which is implemented as a one-step shift of the timestamp-angle tensor along the sequence dimension. As Figure 4 shows, the final training position keeps its own timestamp since it serves only as a label, while at inference the final query slot instead receives the current session time t_{now}, so the same item history can be evaluated under different prediction seasons (e.g., a holiday season versus a routine-replenishment context) without changing the architecture.

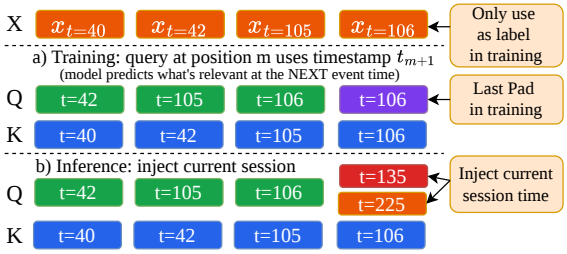


Figure 4: Shifted query alignment. a) During training, each query at position m uses the next-event timestamp t_{m+1}, while the key keeps its historical timestamps t_m; the last slot is used only as a label. b) During inference, the final query slot is assigned the current session time t_{now}. Injecting different current times changes the temporal phase of the query while leaving the item history unchanged.

3.5 Non-Stationary Key Encoding

The preceding components give RoPE real timestamps, multiple scales, and the prediction-time query, but they keep one symmetry from standard RoPE. Because queries and keys rotate the same way, the dot product depends only on the angle difference $\phi^q_m - \phi^k_n = (t_{m+1} - t_n)/\beta_d$, which a constant time shift leaves unchanged. The model can tell an item occurred seven days before the query but not where the query falls in the calendar cycle. T-RoPE removes this time-translation invariance next.

T-RoPE breaks this symmetry by applying the *transpose* of the standard rotation matrix to keys. For plane d and key timestamp t_n, $\mathbf{k}'_{n,d} = \mathbf{R}(-\phi^k_{n,d})\,\mathbf{k}_{n,d} = \mathbf{R}(\phi^k_{n,d})^\top \mathbf{k}_{n,d}$. Since $\mathbf{R}(-\phi) = \mathbf{R}(\phi)^\top$ for 2D rotation matrices, this keeps the same norm-preserving rotation primitive while changing the phase relation inside attention. The resulting dot product is

$$(\mathbf{q}'_{m,d})^\top \mathbf{k}'_{n,d} = \mathbf{q}^\top_{m,d}\mathbf{R}(-\phi^q_{m,d})\mathbf{R}(-\phi^k_{n,d})\,\mathbf{k}_{n,d} = \mathbf{q}^\top_{m,d}\mathbf{R}(-(\phi^q_{m,d} + \phi^k_{n,d}))\,\mathbf{k}_{n,d}. \tag{7}$$

The angle now depends on the *sum* $(t_{m+1} + t_n)/\beta_d$ rather than the gap, so attention shifts when the same history falls in a different calendar region. We prove in Section 4 that this is sufficient for absolute-time sensitivity.

3.6 Complete T-RoPE Formulation

Combining the components, each head receives a timestamp-aware, multiscale, prediction-aligned, and non-stationary rotation. The final T-RoPE rotated vectors are

$$\mathbf{q}'_m = \bigoplus_{d=1}^{D/2} \mathbf{R}(\phi^q_{m,d})\,\mathbf{q}_{m,d}, \qquad \mathbf{k}'_n = \bigoplus_{d=1}^{D/2} \mathbf{R}(-\phi^k_{n,d})\,\mathbf{k}_{n,d}, \tag{8}$$

where $\oplus$ denotes concatenation over planes, giving the attention score

$$a_{mn} = \sum_{d=1}^{D/2} \mathbf{q}^\top_{m,d}\mathbf{R}(-(\alpha^q_d t_{m+1} + \alpha^k_d t_n)/\beta_d)\,\mathbf{k}_{n,d}. \tag{9}$$

This learned linear combination of the prediction timestamp t_{m+1} and the historical timestamp t_n across all planes lets T-RoPE represent both local recency and repeated temporal structure.

Due to space limits, the full forward and backward implementation of T-RoPE is deferred to Appendix B.1. The theoretical FLOP overhead (Appendix B.2) and measured latency (Appendix B.3) show that T-RoPE adds only cost linear in sequence length and head dimension.

4 THEORETICAL ANALYSIS

We now isolate the property the method is meant to fix, namely whether the attention score can change when the same relative history is moved to a different calendar time, since a model that sees only temporal gaps can learn recency but cannot distinguish two queries at different phases of the same annual cycle. The argument proceeds in three steps. We first define time-translation invariance as the property that shifting every timestamp equally leaves attention unchanged; show that standard RoPE stays invariant even on timestamps when query and key share the same relative geometry; and show that T-RoPE's non-stationary key breaks this symmetry, making periodic calendar effects representable. Full proofs and the supporting remarks are deferred to Appendix A.

Definition 1 (Time-translation invariance)**.** A positional encoding is *time-translation invariant* if, for all fixed item embedding vectors $\mathbf{q}, \mathbf{k}$ and all constant offsets $c \in \mathbb{R}$,

$$\mathrm{Attn}(\mathbf{q}, \mathbf{k}; t_{m+1}, t_n) = \mathrm{Attn}(\mathbf{q}, \mathbf{k}; t_{m+1} + c,\ t_n + c).$$

This definition captures the exact limitation we want to avoid. If it holds, attention can depend on the elapsed time between query and key, but it cannot depend on the absolute calendar phase shared by both timestamps.

Theorem 1 (Shared-coefficient timestamp RoPE is time-translation invariant)**.** *Let RoPE be applied with any fixed base frequencies $\{\beta_d\}$ and any learnable shared coefficients $\alpha_d = \alpha_d^q = \alpha_d^k$. Suppose the timestamp-based angles are $\phi_{m,d}^q = \alpha_d\, t_{m+1}/\beta_d$ and $\phi_{n,d}^k = \alpha_d\, t_n/\beta_d$, and the key uses the standard rotation $\mathbf{k}'_{n,d} = \mathbf{R}(\phi_{n,d}^k)\mathbf{k}_{n,d}$. Then $\mathrm{Attn}(\mathbf{q}, \mathbf{k};\, t_{m+1}, t_n)$ depends only on $(t_{m+1} - t_n)$.*

The theorem explains why timestamp substitution alone is insufficient. In each plane the query–key dot product reduces to a rotation by $\alpha_d(t_n - t_{m+1})/\beta_d$, which depends only on the gap and is unchanged by a common shift. Timestamp RoPE can measure how far a past event is from the prediction time, but it still erases the shared calendar location of the two timestamps.

Remark 1 (Independent standard-RoPE coefficients). The same phase analysis shows that unequal coefficients can also break the symmetry, so the condition above is not the only route. Independent query and key coefficients can also break invariance. Our claim is only that T-RoPE's non-stationary key is a RoPE-compatible *sufficient* mechanism that does so directly.

Theorem 2 (T-RoPE with non-stationary keys has absolute-time sensitivity)**.** *Under T-RoPE, each per-plane summand of the attention score (Eq. 9) rotates by the angle $-(\alpha_d^q\, t_{m+1} + \alpha_d^k\, t_n)/\beta_d$, which combines t_{m+1} and t_n additively. If the trained model satisfies $\alpha_d^q + \alpha_d^k \neq 0$ for dimension d, then T-RoPE is* not *time-translation invariant. For any shift c satisfying $(\alpha_d^q + \alpha_d^k)c/\beta_d \notin 2\pi\mathbb{Z}$, there exist timestamps t_{m+1}, t_n and query and key vectors such that $\mathrm{Attn}(t_{m+1}, t_n) \neq \mathrm{Attn}(t_{m+1} + c, t_n + c)$.*

Unlike Theorem 1, the non-stationary key turns the per-plane phase into the *sum* $(\alpha_d^q t_{m+1} + \alpha_d^k t_n)/\beta_d$ rather than a difference, so a common shift moves it by $(\alpha_d^q + \alpha_d^k)c/\beta_d$ and the score changes unless this is a multiple of 2π. Because the coefficients are learnable per dimension, the model chooses which timescales stay stationary and which become absolute-time sensitive, and in our deployed model this induces visible layer-dependent temporal bias (Figure 2) rather than the degenerate $\alpha_d^q = -\alpha_d^k$ case. Absolute-time sensitivity matters because many recommendation signals are periodic, which the next corollary connects to calendar cycles.

Corollary 1 (Seasonality representability)**.** *Let $P > 0$ be a temporal period (e.g., 365 days), and define the effective query angular frequency $\omega_d^q = \alpha_d^q/\beta_d$. T-RoPE can represent a sinusoidal dependence on the prediction-time calendar phase with period P if there exists a dimension d and nonzero integer harmonic r such that $\omega_d^q = 2\pi r/P$.*

Intuitively, with the key timestamp fixed the plane-d score carries a $\cos(\alpha_d^q t_{m+1}/\beta_d + \text{const})$ term, so choosing $\omega_d^q = 2\pi r/P$ makes attention periodic in the prediction time with period P.

5 EXPERIMENTS

We evaluate T-RoPE in three settings. First, public benchmarks across domains and sizes test general accuracy. Second, a Shop ablation isolates which temporal mechanisms matter under non-stationary commerce. Third, an online A/B test measures production retrieval and business impact.

Table 2: Sequential recommendation results (%). Best per row in **bold**, second best underlined. The **green** value beside our result is its relative improvement (%) over the second-best method in that row.

Dataset	Metric	Method					
		SASRec	TiSASRec	HSTU	HSTU+ Time RAB	HSTU+ TO-RoPE	**HSTU+ T-RoPE (ours)**
ML-20M	HR@10	31.36	31.97	30.87	32.05	31.19	**32.37** +1.0%
	HR@50	56.87	57.50	56.19	58.01	56.69	**58.55** +0.9%
	NDCG@10	18.04	18.38	17.84	18.47	17.93	**18.62** +0.8%
	NDCG@50	23.68	24.04	23.44	24.20	23.57	**24.42** +0.9%
	MRR	15.42	15.68	15.31	15.79	15.32	**15.91** +0.8%
Amazon Books	HR@10	2.97	3.19	5.88	6.89	6.69	**7.65** +11.0%
	HR@50	7.26	7.66	12.40	14.27	13.30	**15.45** +8.3%
	NDCG@10	1.57	1.71	3.33	3.93	3.96	**4.40** +11.1%
	NDCG@50	2.50	2.68	4.75	5.54	5.40	**6.11** +10.3%
	MRR	1.41	1.53	2.94	3.46	3.51	**3.87** +10.3%
Pixel200K	HR@10	4.72	4.65	4.22	7.21	4.16	**7.41** +2.8%
	HR@50	10.85	10.86	10.26	16.82	10.09	**17.37** +3.3%
	NDCG@10	2.65	2.61	2.33	4.08	2.30	**4.15** +1.7%
	NDCG@50	3.98	3.93	3.63	6.14	3.58	**6.30** +2.6%
	MRR	2.39	2.35	2.12	3.70	2.11	**3.76** +1.6%
Pixel1M	HR@10	7.12	7.16	6.89	12.46	6.81	**13.35** +7.1%
	HR@50	15.62	15.77	15.31	26.16	15.14	**27.44** +4.9%
	NDCG@10	4.04	4.05	3.88	7.22	3.83	**7.81** +8.2%
	NDCG@50	5.88	5.92	5.70	10.19	5.64	**10.88** +6.8%
	MRR	3.61	3.62	3.46	6.42	3.43	**6.95** +8.3%
Pixel8M	HR@10	4.89	4.96	5.01	9.14	4.99	**11.47** +25.5%
	HR@50	10.69	10.82	10.93	18.87	10.86	**23.02** +22.0%
	NDCG@10	2.75	2.79	2.82	5.36	2.82	**6.85** +27.8%
	NDCG@50	4.00	4.06	4.11	7.47	4.09	**9.35** +25.2%
	MRR	2.44	2.48	2.51	4.78	2.50	**6.11** +27.8%

5.1 Validating T-RoPE on Public Datasets

To validate T-RoPE against existing methods with reproducible results, we evaluate on five public recommendation benchmarks whose statistics are listed in Table 6 (Appendix C). We compare against five existing methods: **SASRec** (Kang & McAuley, 2018) (no temporal encoding), **TiSASRec** (Li et al., 2020) (time-interval attention biases), **HSTU** (Zhai et al., 2024) (position-only RAB), **HSTU + Time RAB** (Zhai et al., 2024), and **HSTU + TO-RoPE** (Wei et al., 2025) (HSTU with Time+Pos RAB + TO-RoPE). All of these, together with our **HSTU + T-RoPE** (HSTU with Time+Pos RAB and T-RoPE), are implemented in the same PyTorch training framework and share identical item-ID inputs, preprocessing filters, maximum sequence lengths, leave-one-out protocol, candidate evaluation procedure, and training hyperparameters, making the comparison fair and consistent. Full hyperparameter settings are reported in Appendix D.

Detailed results across the five benchmarks are reported in Table 2, covering HR@10, HR@50, NDCG@10, NDCG@50, and MRR. HSTU + T-RoPE achieves the best result on all five metrics on every dataset, and the improvement scales with how much temporal structure the data contain. On the dense ML-20M benchmark, whose histories are fairly regular, gains are modest but consistent. The effect is clearer on sparse and commercial data, where T-RoPE improves over HSTU + Time RAB by 8–12% across all five metrics on Amazon Books, and over TO-RoPE by 78–130% in HR@10 and 72–112% in HR@50 across the PixelRec splits. The largest margins appear on the sparse and commercial data, where absolute calendar phase carries the most signal.

Table 3: Ablation study on the internal Shop e-commerce dataset. The shaded top row is the full T-RoPE model. Rows 2–5 remove components cumulatively down to the HSTU + Time RAB backbone, and rows 6–8 remove single components from the full model. Metric values are percentages, with the best value in each column shown in **bold**. Parenthetical values show each row's relative change from the full model. Green indicates an improvement over T-RoPE, while red indicates a shortfall.

Configuration	Recall@50	NDCG@50	Recall@200	NDCG@200	MRR
T-RoPE	**31.41**	**22.94**	41.77	**24.82**	**23.14**
- Non-Stationary (§3.5)	30.97 (−1.40%)	22.02 (−4.01%)	40.49 (−3.06%)	23.29 (−6.16%)	21.88 (−5.45%)
- shifted (§3.4)	30.90 (−1.62%)	20.59 (−10.24%)	40.92 (−2.03%)	22.55 (−9.15%)	20.90 (−9.68%)
- multiscale freq. bank (§3.3)	25.36 (−19.26%)	13.23 (−42.33%)	36.98 (−11.47%)	15.42 (−37.87%)	11.96 (−48.31%)
- abs. time embedding (§3.1 + §3.2)	25.68 (−18.24%)	13.81 (−39.80%)	37.00 (−11.42%)	15.96 (−35.70%)	12.70 (−45.12%)
- Shifted Query RoPE	30.87 (−1.72%)	21.54 (−6.10%)	40.50 (−3.04%)	23.42 (−5.64%)	22.46 (−2.94%)
- Key RoPE	31.19 (−0.70%)	20.58 (−10.29%)	**41.90** (+0.31%)	22.63 (−8.82%)	20.90 (−9.68%)
- Time RAB	29.08 (−7.42%)	20.76 (−9.50%)	39.69 (−4.98%)	22.95 (−7.53%)	16.89 (−27.01%)

5.2 Ablation Study on an Industrial-Scale Dataset

To identify which design choices drive the gains, we ablate T-RoPE components on Shop, an internal e-commerce dataset with roughly 150M users, 4.8M items, and over 6B interactions spanning multiple years. Larger and more temporally heterogeneous than the public benchmarks, it better tests the multiscale and non-stationary mechanisms. The model uses 128-dimensional embeddings, 12 HSTU blocks, 8 attention heads, 32 attention dimensions per head, and a maximum sequence length of 512. We use a one-day temporal holdout (training on interactions before the holdout date, every meaningful interaction on the held-out day counting as a positive) and report recall@K rather than HR@K, since a user may have several relevant interactions. The detailed results are shown in Table 3, which removes components cumulatively from the full T-RoPE model down to the HSTU + Time RAB backbone, with the key findings discussed below.

The multiscale frequency bank and shifted queries supply the core gains. The multiscale bank has the largest effect. Removing it cuts NDCG@50 by 36% when comparing rows 3 and 4. This shows that multiscale rotation turns raw timestamps into usable frequency structure. Without it, the timestamp is just noise, and the single frequency embedding in row 4 falls about 4% below the HSTU + Time RAB backbone in row 5. Shifted query alignment provides the next largest gain by adding seasonality to the query at inference. It improves NDCG@50 by 7% when comparing rows 2 and 3.

The non-stationary key drives absolute-time sensitivity, complemented by key and query rotation. Adding the non-stationary key (§3.5, row 1) outperforms the shifted model (row 2) on all five metrics, improving recall@200 by 3.2% and NDCG@50 by 4.2%, which confirms that it injects useful absolute-time sensitivity. Key and query rotation each carry such sensitivity on their own (Remark 1), so we further ablate them one at a time. The shifted-query rotation has the larger effect on retrieval, since removing it (row 6) lowers recall@200 by 3.0%, whereas removing the key rotation (row 7) instead costs ranking precision, cutting NDCG@50 by 10.3%. The two are therefore complementary, and the non-stationary key combines both sides and pushes furthest.

Time RAB and T-RoPE are complementary. Removing Time RAB from the full model (row 8) lowers all metrics, with the starkest drop of 27.0% in MRR. This suggests RAB still captures useful relative-recency signals that T-RoPE does not. Even so, T-RoPE without RAB still improves NDCG@50 by roughly 50% over the baseline, so the two mechanisms contribute independently and combine for the best result.

Table 4: Online A/B test results in the Shop app. Values are relative lifts against the control group.

Metric	Definition	Relative lift	P-Value
CVR	Conversion rate	$+0.33\%$	0.037
Shop orders	Count of orders placed on the Shop app	$+0.63\%$	0.094

5.3 Online A/B Test

To measure real production impact, we deploy T-RoPE in an online A/B test in the Shop app recommender, scaling up the offline architecture while keeping the T-RoPE design unchanged. Table 4 reports relative lift against the production control, an identical model without T-RoPE.

The online results align with the offline gains. CVR shows the strongest signal ($+0.33\%$, $p = 0.037$), indicating that time-aware retrieval converts more sessions into interactions. This carries to higher-intent behavior. The Shop orders also increase $+0.63\%$ ($p = 0.094$), directional evidence of downstream value. Overall, time-aware RoPE yields positive business gains in production.

6 Related Work

Sequential recommendation models Sequential recommendation models user preferences from ordered interaction histories. Early deep models used recurrent networks (GRU4Rec (Hidasi et al., 2016), NARM (Li et al., 2017)) and convolution (Caser (Tang & Wang, 2018)). The Transformer (Vaswani et al., 2017) reshaped the field: SASRec (Kang & McAuley, 2018) showed that unidirectional self-attention outperforms RNNs, and BERT4Rec (Sun et al., 2019) added bidirectional masked-item prediction. Later work enriched attention with item attributes (FDSA (Zhang et al., 2019)), replaced it with learnable frequency-domain filters (FMLP-Rec (Zhou et al., 2022)), or added self-supervised pretraining (S^3-Rec (Zhou et al., 2020)); at production scale, HSTU (Zhai et al., 2024) reaches trillions of parameters via hierarchical action-based transduction. A newer thread casts recommendation as autoregressive generation over discrete semantic tokens: TIGER (Rajput et al., 2023) uses RQ-VAE item codes, LETTER (Wang et al., 2024) adds collaborative signals to tokenization, ActionPiece (Hou et al., 2025) tokenizes at action granularity, PSID (Zhang et al., 2025) relies on pure semantic IDs, and HLLM (Chen et al., 2024; Zhang et al., 2026b;a; Liu et al., 2026a; He et al., 2026) pairs an item-level with a user-level LLM. T-RoPE is a drop-in positional encoding compatible with any of these Transformer backbones.

Temporal information in sequential recommendation Temporal dynamics are a core problem in sequential recommendation. Early collaborative filtering showed that preferences and item popularity shift substantially over calendar time (Koren, 2009). TiSASRec (Li et al., 2020) discretizes the elapsed gap $|t_i - t_j|$ into buckets injected as additive attention biases, capturing relative recency but only the gap magnitude, with no calendar-phase awareness. Knowledge-guided methods (Wang et al., 2020) learn time-decay kernels over item relations, and attention-mixture approaches (Tran et al., 2023) condition on calendar features such as hour of day. TO-RoPE (Wei et al., 2025) blends position indices and timestamps inside RoPE angles, making attention sensitive to temporal distance, but because query and key share the same timestamp coefficient it remains time-translation invariant (Theorem 1, Remark 2). T-RoPE instead introduces a non-stationary key with learnable query/key coefficients, breaking that invariance and giving the absolute-time awareness needed for periodic patterns such as seasonal purchase cycles.

7 Conclusion

We introduced T-RoPE, a time-aware rotary embedding that rotates attention by when events happen rather than by their order, so the model reasons directly over elapsed time and recurring calendar cycles. We proved that shared-geometry timestamp encodings are time-translation invariant and cannot model seasonal preferences, motivating T-RoPE's non-stationary keys. We validated it on public benchmarks and a large commercial dataset, where it consistently outperforms baselines, with the largest gains under high temporal heterogeneity. Grounding rotary attention in time is thus principled and effective for recommendation.

AI USE STATEMENT

We used AI only to aid and polish writing at the sentence level, including edits to clarity, grammar, and phrasing. We did not use generative AI for literature retrieval or discovery, research ideation, theoretical development, methodology or experiment design, code implementation, data generation or processing, data analysis, or interpretation of results. The authors reviewed every AI-assisted edit and take full responsibility for the final content of this work.

ETHICS STATEMENT

Public experiments used established recommendation benchmarks obtained from the cited sources. The industrial experiments use data for which the authors had authorized access, and the online experiment compared two recommender variants in the normal operation of the Shop app. The industrial and online studies followed the company's processes for data access, privacy, and experimentation governance. We report their results only in aggregate and release neither individual user records nor personally identifying information. The proprietary data and production infrastructure therefore cannot be made public.

REPRODUCIBILITY STATEMENT

We support reproduction of the public-benchmark results through the complete method specification in Section 3, the experimental protocol in Section 5.1, and the implementation details in Appendix B. Appendix B.1 gives forward and backward algorithms, Appendix C lists the public datasets and their statistics, and Appendix D reports the architecture, optimization settings, random seed, and hardware. Appendix A contains full proofs of the theoretical claims, while Appendices B.2 and B.3 document computational cost and measured throughput.

REFERENCES

Junyi Chen, Lu Chi, Bingyue Peng, and Zehuan Yuan. HLLM: Enhancing sequential recommendations via hierarchical large language models for item and user modeling. *arXiv preprint arXiv:2409.12740*, 2024. URL https://arxiv.org/abs/2409.12740.

Yu Cheng, Yunzhu Pan, Jiaqi Zhang, Yongxin Ni, Aixin Sun, and Fajie Yuan. An image dataset for benchmarking recommender systems with raw pixels. In *Proceedings of the 2024 SIAM International Conference on Data Mining (SDM)*, pp. 418–426, 2024. doi: 10.1137/1.9781611978032.49.

F. Maxwell Harper and Joseph A. Konstan. The MovieLens datasets: History and context. *ACM Transactions on Interactive Intelligent Systems*, 5(4):1–19, 2015. doi: 10.1145/2827872.

Yinhan He, Liam Collins, Bhuvesh Kumar, Jundong Li, Neil Shah, and Donald Loveland. Implicit reasoning for large language model-based generative recommendation. *arXiv preprint arXiv:2606.14142*, 2026. URL https://arxiv.org/abs/2606.14142.

Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. Session-based recommendations with recurrent neural networks. In *International Conference on Learning Representations (ICLR)*, 2016.

Yupeng Hou, Jianmo Ni, Zhankui He, Noveen Sachdeva, Wang-Cheng Kang, Ed H. Chi, Julian McAuley, and Derek Zhiyuan Cheng. ActionPiece: Contextually tokenizing action sequences for generative recommendation. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, 2025. URL https://arxiv.org/abs/2502.13581.

Yupeng Hou, Jiacheng Li, Xiangjun Fu, Zhankui He, An Yan, Xiusi Chen, and Julian McAuley. Bridging language and items for retrieval and recommendation: Benchmarking LLMs as semantic encoders. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 3251–3265, 2026. doi: 10.18653/v1/2026.acl-long.147.

Wang-Cheng Kang and Julian McAuley. Self-attentive sequential recommendation. In *Proceedings of the IEEE International Conference on Data Mining (ICDM)*, pp. 197–206, 2018. doi: 10.1109/ICDM.2018.00035.

Yehuda Koren. Collaborative filtering with temporal dynamics. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 447–456, 2009. doi: 10.1145/1557019.1557072.

Jiacheng Li, Yujie Wang, and Julian McAuley. Time interval aware self-attention for sequential recommendation. In *Proceedings of the 13th ACM International Conference on Web Search and Data Mining (WSDM)*, pp. 322–330, 2020. doi: 10.1145/3336191.3371786.

Jing Li, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Tao Lian, and Jun Ma. Neural attentive session-based recommendation. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management (CIKM)*, pp. 1419–1428, 2017. doi: 10.1145/3132847.3132926.

Mingfu Liang, Yufei Li, Jay Xu, Kavosh Asadi, Xi Liu, Shuo Gu, Kaushik Rangadurai, Frank Shyu, Shuaiwen Wang, Song Yang, Zhijing Li, Jiang Liu, Mengying Sun, Fei Tian, Xiaohan Wei, Chonglin Sun, Jacob Tao, Shike Mei, Wenlin Chen, Santanu Kolay, Sandeep Pandey, Hamed Firooz, and Luke Simon. GR2: Generative reasoning re-ranker. *arXiv preprint arXiv:2602.07774*, 2026. URL https://arxiv.org/abs/2602.07774.

Enze Liu, Zhen Tian, and Wayne Xin Zhao. Collaborative memory augmentation for generative recommendation. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2026a. URL https://arxiv.org/abs/2608.01315.

Xiaoran Liu, Yuerong Song, Zhigeng Liu, Zengfeng Huang, Qipeng Guo, Zhaoxiang Liu, Shiguo Lian, Ziwei He, and Xipeng Qiu. Beyond real: Imaginary extension of rotary position embeddings for long-context LLMs. In *International Conference on Learning Representations (ICLR)*, 2026b. URL https://arxiv.org/abs/2512.07525.

Sajad Movahedi, Timur Carstensen, Arshia Afzal, Frank Hutter, Antonio Orvieto, and Volkan Cevher. Selective rotary position embedding. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://arxiv.org/abs/2511.17388.

OneRec Team et al. OneReason technical report. *arXiv preprint arXiv:2606.06260*, 2026. URL https://arxiv.org/abs/2606.06260.

Shashank Rajput, Nikhil Mehta, Anima Singh, Raghunandan Hulikal Keshavan, Trang Yu, Lukasz Heldt, Lichan Hong, Yi Tay, Vinh Q. Tran, Jonah Samost, Maciej Kula, Ed H. Chi, and Maheswaran Sathiamoorthy. Recommender systems with generative retrieval. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, 2023.

Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. RoFormer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024. doi: 10.1016/j.neucom.2023.127063.

Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management (CIKM)*, pp. 1441–1450, 2019. doi: 10.1145/3357384.3357895.

Jiaxi Tang and Ke Wang. Personalized top-N sequential recommendation via convolutional sequence embedding. In *Proceedings of the 11th ACM International Conference on Web Search and Data Mining (WSDM)*, pp. 565–573, 2018. doi: 10.1145/3159652.3159656.

Qingyuan Tian, Wenhong Zhu, Xiaoran Liu, Xiaofeng Wang, and Rui Wang. MrRoPE: Mixed-radix rotary position embedding. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://arxiv.org/abs/2601.22181.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. LLaMA: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023. URL https://arxiv.org/abs/2302.13971.

Viet-Anh Tran, Guillaume Salha-Galvan, Bruno Sguerra, and Romain Hennequin. Attention mixtures for time-aware sequential recommendation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pp. 1821–1826, 2023. doi: 10.1145/3539618.3591951.

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

Chenyang Wang, Min Zhang, Weizhi Ma, Yiqun Liu, and Shaoping Ma. Make it a chorus: Knowledge- and time-aware item modeling for sequential recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pp. 109–118, 2020. doi: 10.1145/3397271.3401131.

Wenjie Wang, Honghui Bao, Xinyu Lin, Jizhi Zhang, Yongqi Li, Fuli Feng, See-Kiong Ng, and Tat-Seng Chua. Learnable item tokenization for generative recommendation. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management (CIKM)*, pp. 2400–2409, 2024. doi: 10.1145/3627673.3679569.

Xiaokai Wei, Jiajun Wu, Daiyao Yi, Reza Shirkavand, and Michelle Gong. Rotate both ways: Time-and-order RoPE for generative recommendation. *arXiv preprint arXiv:2510.20455*, 2025. URL https://arxiv.org/abs/2510.20455.

Jiaqi Zhai, Lucy Liao, Xing Liu, Yueming Wang, Rui Li, Xuan Cao, Leon Gao, Zhaojie Gao, Xinyang Gu, Zhiling He, et al. Actions speak louder than words: Trillion-parameter sequential transducers for generative recommendations. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, pp. 58484–58509. PMLR, 2024.

Ruohan Zhang, Jiacheng Li, Julian McAuley, and Yupeng Hou. Purely semantic indexing for LLM-based generative recommendation and retrieval. *arXiv preprint arXiv:2509.16446*, 2025. URL https://arxiv.org/abs/2509.16446.

Tingting Zhang, Pengpeng Zhao, Yanchi Liu, Victor S. Sheng, Jiajie Xu, Deqing Wang, Guanfeng Liu, and Xiaofang Zhou. Feature-level deeper self-attention network for sequential recommendation. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI-19)*, pp. 4320–4326, 2019. doi: 10.24963/ijcai.2019/600.

Yingyi Zhang, Junyi Li, Yejing Wang, Wenlin Zhang, Xiaowei Qian, Sheng Zhang, Yue Feng, Yichao Wang, Yong Liu, Xiangyu Zhao, and Xianneng Li. RAGR: Review-augmented generative recommendation. *arXiv preprint arXiv:2605.17267*, 2026a. URL https://arxiv.org/abs/2605.17267.

Zhiyang Zhang, Junda She, Kuo Cai, Bo Chen, Shiyao Wang, Xinchen Luo, Qiang Luo, Ruiming Tang, Han Li, Kun Gai, and Guorui Zhou. Unleashing the native recommendation potential: LLM-based generative recommendation via structured term identifiers. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 19659–19673, 2026b. doi: 10.18653/v1/2026.findings-acl.984.

Kun Zhou, Hui Wang, Wayne Xin Zhao, Yutao Zhu, Sirui Wang, Fuzheng Zhang, Zhongyuan Wang, and Ji-Rong Wen. S^3-Rec: Self-supervised learning for sequential recommendation with mutual information maximization. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management (CIKM)*, pp. 1893–1902, 2020. doi: 10.1145/3340531.3411954.

Kun Zhou, Hui Yu, Wayne Xin Zhao, and Ji-Rong Wen. Filter-enhanced MLP is all you need for sequential recommendation. In *Proceedings of the ACM Web Conference 2022 (WWW)*, pp. 2388–2399, 2022. doi: 10.1145/3485447.3512111.

APPENDIX

A FULL PROOFS

A.1 PROOF OF THEOREM 1

Restatement. With shared coefficients $\alpha_d = \alpha_d^q = \alpha_d^k$ and angles $\phi_{m,d}^q = \alpha_d t_{m+1}/\beta_d$, $\phi_{n,d}^k = \alpha_d t_n/\beta_d$ under the standard key rotation $\mathbf{k}'_{n,d} = \mathbf{R}(\phi_{n,d}^k)\mathbf{k}_{n,d}$, the attention score depends only on $(t_{m+1} - t_n)$ and is therefore time-translation invariant.

Proof. We give the full argument for standard key rotation with shared query and key timestamp coefficients $\alpha_d^q = \alpha_d^k \triangleq \alpha_d$.

Standard RoPE with $\phi_{m,d}^q = \alpha_d t_{m+1}/\beta_d$ and $\phi_{n,d}^k = \alpha_d t_n/\beta_d$ (standard key rotation $\mathbf{R}(\phi_{n,d}^k)$) gives

$$(\mathbf{q}'_{m,d})^\top \mathbf{k}'_{n,d} = \mathbf{q}_{m,d}^\top \mathbf{R}(\phi_{n,d}^k - \phi_{m,d}^q)\,\mathbf{k}_{n,d} = \mathbf{q}_{m,d}^\top \mathbf{R}(\alpha_d(t_n - t_{m+1})/\beta_d)\,\mathbf{k}_{n,d},$$

which depends only on $(t_n - t_{m+1})$. Summing over planes and heads preserves time-translation invariance.

For comparison, if standard key rotation uses independent coefficients α_d^q and α_d^k, the phase is

$$\phi_{n,d}^k - \phi_{m,d}^q = (\alpha_d^k t_n - \alpha_d^q t_{m+1})/\beta_d.$$

Under a global shift by c, the phase changes by $(\alpha_d^k - \alpha_d^q)c/\beta_d$. Therefore this standard-RoPE family is invariant only for active planes where $\alpha_d^q = \alpha_d^k$. Opposite nonzero coefficients depend on the timestamp sum and are not time-translation invariant. □

Remark 2 (Relation to TO-RoPE). The same argument covers the timestamp component of TO-RoPE, which shares one timestamp coefficient across query and key. That component depends only on $(\tau_m - \tau_n)$ and is therefore time-translation invariant, so it cannot represent seasonal shifts on its own.

A.2 PROOF OF THEOREM 2

Restatement. Under T-RoPE the per-plane attention equals $\mathbf{q}_{m,d}^\top \mathbf{R}(-(\alpha_d^q t_{m+1} + \alpha_d^k t_n)/\beta_d)\,\mathbf{k}_{n,d}$; if $\alpha_d^q + \alpha_d^k \neq 0$ for some dimension d, then T-RoPE is not time-translation invariant.

Proof. We expand Eq. 7 carefully using the rotation group property $\mathbf{R}(a)\mathbf{R}(b) = \mathbf{R}(a+b)$,

$$\begin{aligned}
(\mathbf{q}'_{m,d})^\top \mathbf{k}'_{n,d} &= \Big(\mathbf{R}(\phi_{m,d}^q)\mathbf{q}_{m,d}\Big)^\top \big(\mathbf{R}(-\phi_{n,d}^k)\mathbf{k}_{n,d}\big) \\
&= \mathbf{q}_{m,d}^\top \mathbf{R}(\phi_{m,d}^q)^\top \mathbf{R}(-\phi_{n,d}^k)\,\mathbf{k}_{n,d} \\
&= \mathbf{q}_{m,d}^\top \mathbf{R}(-\phi_{m,d}^q)\mathbf{R}(-\phi_{n,d}^k)\,\mathbf{k}_{n,d} \\
&= \mathbf{q}_{m,d}^\top \mathbf{R}(-\phi_{m,d}^q - \phi_{n,d}^k)\,\mathbf{k}_{n,d}.
\end{aligned}$$

Substituting $\phi_{m,d}^q = \alpha_d^q t_{m+1}/\beta_d$ and $\phi_{n,d}^k = \alpha_d^k t_n/\beta_d$ gives

$$(\mathbf{q}'_{m,d})^\top \mathbf{k}'_{n,d} = \mathbf{q}_{m,d}^\top \mathbf{R}(-(\alpha_d^q t_{m+1} + \alpha_d^k t_n)/\beta_d)\,\mathbf{k}_{n,d}.$$

Under the time shift $(t_{m+1}, t_n) \to (t_{m+1} + c, t_n + c)$, the argument changes by $(\alpha_d^q + \alpha_d^k)c/\beta_d$. If $(\alpha_d^q + \alpha_d^k)c/\beta_d \notin 2\pi\mathbb{Z}$, the shifted and unshifted rotation matrices are different. Since two different rotation matrices define different bilinear forms, there exist nondegenerate vectors $\mathbf{q}_{m,d}$ and $\mathbf{k}_{n,d}$ for which the dot product changes after the shift. This gives a counterexample to the universal equality required by time-translation invariance. If the phase shift is in $2\pi\mathbb{Z}$, that particular shift can coincide, but such modulo coincidences do not establish invariance for all shifts. □

Algorithm 1 T-RoPE forward pass

Require: queries Q, keys $K \in \mathbb{R}^{B\times L\times H\times D}$; timestamps $\mathbf{t} \in \mathbb{R}^{B\times L}$; coefficients $\alpha^q, \alpha^k \in \mathbb{R}^{D/2}$

$\beta \leftarrow \mathrm{geomspace}(10^2, 10^8, D/2)$
$\Phi^q \leftarrow \mathrm{FP32}(\alpha^q)[\mathrm{None}, \mathrm{None}, :] \cdot \mathbf{t}[:, :, \mathrm{None}]/\beta[\mathrm{None}, \mathrm{None}, :]$
$\Phi^q \leftarrow \textsc{RollLeft}(\Phi^q, \mathrm{dim} = 1)$
$\Phi^q[:, L, :] \leftarrow \mathrm{FP32}(\alpha^q)[:] \cdot \mathbf{t}[:, L, \mathrm{None}]/\beta[:]$ ▷ last query keeps its timestamp
$\Phi^k \leftarrow \mathrm{FP32}(\alpha^k)[\mathrm{None}, \mathrm{None}, :] \cdot \mathbf{t}[:, :, \mathrm{None}]/\beta[\mathrm{None}, \mathrm{None}, :]$
$Q' \leftarrow \textsc{ApplyRotFromTheta}(Q, \Phi^q, \mathrm{standard})$
$K' \leftarrow \textsc{ApplyRotFromTheta}(K, \Phi^k, \mathrm{nonstationary})$
return Q', K'

Remark 3 (Learnable degeneracy and timescale selection). The condition $\alpha_d^q + \alpha_d^k \neq 0$ for some d is genuinely needed. If training converges to $\alpha_d^q = -\alpha_d^k$ for all d, T-RoPE degenerates to the time-translation invariant case. This is not purely a failure mode, since learnable per-dimension coefficients let the model choose which timescales stay stationary and which become absolute-time sensitive. In our deployed model the degenerate case does not arise, and the learned coefficients induce visible layer-dependent temporal bias (Figure 2).

A.3 Proof of Corollary 1

Restatement. For a period $P > 0$ and effective query angular frequency $\omega_d^q = \alpha_d^q/\beta_d$, T-RoPE can represent a sinusoidal dependence on the prediction-time calendar phase with period P whenever $\omega_d^q = 2\pi r/P$ for some dimension d and nonzero integer harmonic r.

Proof. The attention in dimension d can be written as

$$f_d(t_{m+1}, t_n) = \mathbf{q}_{m,d}^\top \mathbf{R}(-(\alpha_d^q t_{m+1} + \alpha_d^k t_n)/\beta_d)\, \mathbf{k}_{n,d}.$$

Expanding in terms of $\cos$ and $\sin$,

$$f_d = (q_1k_1 + q_2k_2)\cos\left(\frac{\alpha_d^q t_{m+1} + \alpha_d^k t_n}{\beta_d}\right) + (q_2k_1 - q_1k_2)\sin\left(\frac{\alpha_d^q t_{m+1} + \alpha_d^k t_n}{\beta_d}\right).$$

Holding t_n fixed, the key term $\alpha_d^k t_n/\beta_d$ is a constant phase offset. Let $\omega_d^q = \alpha_d^q/\beta_d$. If $\omega_d^q = 2\pi r/P$ for a nonzero integer harmonic r, then both the sine and cosine terms are periodic in t_{m+1} with period P. Thus T-RoPE has the capacity to express sinusoidal modulation over prediction time at the target calendar period. □

B Implementation Details

B.1 Forward and backward algorithms

Each layer using T-RoPE maintains a geometric frequency bank $\beta \in \mathbb{R}^{D/2}$ and two trainable coefficient vectors $\alpha^q, \alpha^k \in \mathbb{R}^{D/2}$. The frequency bank uses $\beta_{\min} = 10^2$ seconds and $\beta_{\max} = 10^8$ seconds by default, and the coefficients are initialized independently from $\mathcal{N}(0, \sigma^2)$ with $\sigma = 1.0$.

Algorithm 1 expands the forward pass. Because we use absolute Unix timestamps in seconds, which are on the order of 10^9, T-RoPE computes Φ^q and Φ^k in FP32 even when the surrounding model uses BF16. The rotated outputs are cast back to the model dtype after the trigonometric functions have been evaluated.

Algorithm 3 summarizes the custom autograd path used for learnable angles. A fixed RoPE backward only needs dX, which is obtained by applying the conjugate rotation to the upstream gradient. T-RoPE additionally returns $d\Phi$ because Φ depends on trainable coefficients. This is the main computational difference from raw RoPE.

Algorithm 2 APPLYROTFROMTHETA

Require: input X, angles Φ, rotation type $r \in \{\text{standard}, \text{nonstationary}\}$
$C \leftarrow \cos\Phi$, $S \leftarrow \sin\Phi$
Split rotary dimensions of X into planes (x_0, x_1)
if $r = \text{standard}$ **then**
    $y_0 \leftarrow x_0 C - x_1 S$; $y_1 \leftarrow x_0 S + x_1 C$
else
    $y_0 \leftarrow x_0 C + x_1 S$; $y_1 \leftarrow -x_0 S + x_1 C$
end if
Preserve all non-rotary tail dimensions of X unchanged
return concatenated output Y

Algorithm 3 Backward pass for learnable T-RoPE angles

Require: upstream gradient G, saved input X, saved angles Φ, rotation type r
$C \leftarrow \cos\Phi$, $S \leftarrow \sin\Phi$
Split rotary dimensions of X and G into planes (x_0, x_1) and (g_0, g_1)
if $r = \text{standard}$ **then**
    $dx_0 \leftarrow g_0 C + g_1 S$; $dx_1 \leftarrow -g_0 S + g_1 C$
    $d\Phi \leftarrow g_0(-x_0 S - x_1 C) + g_1(x_0 C - x_1 S)$
else
    $dx_0 \leftarrow g_0 C - g_1 S$; $dx_1 \leftarrow g_0 S + g_1 C$
    $d\Phi \leftarrow g_0(-x_0 S + x_1 C) + g_1(-x_1 S - x_0 C)$
end if
$d\Phi \leftarrow$ SUMHEADS$(d\Phi)$ ▷ sum contributions from all heads
return $dX, d\Phi$

The gradients for the learnable coefficients follow directly from $\Phi^q_{b,l,d} = \alpha^q_d t^q_{b,l}/\beta_d$ and $\Phi^k_{b,l,d} = \alpha^k_d t_{b,l}/\beta_d$,

$$\frac{\partial \mathcal{L}}{\partial \alpha^q_d} = \sum_{b,l} \frac{\partial \mathcal{L}}{\partial \Phi^q_{b,l,d}} \frac{t^q_{b,l}}{\beta_d}, \qquad \frac{\partial \mathcal{L}}{\partial \alpha^k_d} = \sum_{b,l} \frac{\partial \mathcal{L}}{\partial \Phi^k_{b,l,d}} \frac{t_{b,l}}{\beta_d}, \tag{10}$$

where $t^q_{b,l} = t_{b,l+1}$ for $l < L$ and $t^q_{b,L} = t_{b,L}$ under the shifted query boundary rule. Because $t_{b,l+1}$ is a Unix timestamp measured in seconds, the gradients are usually large and must be clipped or rescaled to stabilize training.

B.2 PARAMETER AND FLOP OVERHEAD

Let B be the batch size and L the sequence length, so $N = BL$ is the total number of tokens. Let H be the number of attention heads, D the per-head dimension, $P = D/2$ the number of rotary planes (each plane is a 2×2 rotation over one channel pair), and M the number of Transformer layers that use T-RoPE. Because each T-RoPE module keeps separate query and key coefficients, it adds $2P = D$ learnable parameters, so the full model adds MD parameters in total. In our HSTU experiments ($M = 2$ layers, $D = 128$), this is only $2 \cdot 128 = 256$ additional parameters.

We count multiply, add/subtract, and divide as one FLOP each, and report $\sin$ and $\cos$ separately as transcendental operations. The non-stationary key rotation has the same arithmetic count as a standard rotation. It only changes signs in the rotation equations. The rotation cost below corresponds to Algorithm 2 applied once to Q and once to K.

Operation	Arithmetic FLOPs	Transcendental ops
Build Φ^q, Φ^k	$4NP$	–
Rotate Q, K	$12NHP$	$2NP$ sin and $2NP$ cos
Fixed-RoPE backward dQ, dK	$12NHP$	$2NP$ sin and $2NP$ cos
Extra learnable-angle $d\Phi$	$18NHP + 2NP(H-1)$	–
Coefficient gradients	$\approx 4NP$	–

Thus, relative to raw RoPE with a fixed precomputed rotation cache, T-RoPE adds $4NP$ forward arithmetic FLOPs to build timestamp-dependent angles, plus runtime trigonometric evaluations if the kernel computes $\sin \Phi$ and $\cos \Phi$ on the fly. During training, the extra cost over fixed raw RoPE is the $d\Phi$ computation and reduction, followed by the coefficient-gradient chain above. These extra terms are linear in sequence length and head dimension, while self-attention remains quadratic in sequence length.

B.3 Static latency analysis.

Table 5 reports static latency based on sampled training and validation iterator throughput during the second epoch for the Pixel1M and Shop examples. T-RoPE has modest observed throughput overhead relative to baseline HSTU and is comparable to TO-RoPE.

Table 5: Static throughput snapshots for baseline HSTU, TO-RoPE, and T-RoPE. Values are mean $\pm$ sample standard deviation in iterations per second.

Dataset example	Model	Training it/s	Validation it/s
Pixel1M	Baseline HSTU	17.58 ± 0.01	58.59 ± 0.87
Pixel1M	HSTU + TO-RoPE	16.84 ± 0.05	57.33 ± 0.74
Pixel1M	HSTU + T-RoPE	16.95 ± 0.01	57.06 ± 0.44
Shop	Baseline HSTU	37.19 ± 0.20	92.66 ± 0.69
Shop	HSTU + TO-RoPE	32.17 ± 0.35	82.62 ± 0.92
Shop	HSTU + T-RoPE	32.91 ± 0.05	86.05 ± 0.71

C Dataset Statistics

Table 6 reports the size of the five public benchmarks used in Section 5.1.

Table 6: Statistics of the five public benchmarks.

Dataset	#User	#Item	#Interaction
ML-20M (Harper & Konstan, 2015)	138,493	26,744	20,000,263
Amazon Books (Hou et al., 2026)	776,370	495,063	9,488,297
Pixel200K (Cheng et al., 2024)	200,000	96,282	3,965,656
Pixel1M (Cheng et al., 2024)	1,001,822	100,541	19,886,579
Pixel8M (Cheng et al., 2024)	8,886,078	407,082	158,488,682

D Hyperparameter Settings

Table 7 summarises the architecture and training hyperparameters. All models use 2 Transformer blocks, 128 in-batch negatives, and a learning rate of 10^{-3}. The values shown are the defaults shared by ML-20M, Amazon Books, Pixel200K, Pixel1M, and Pixel8M. The sequence-length exception is listed in the footnote. Here d is the embedding dimension, H the number of attention heads, A the attention dimension, and RAB the relative attention bucket type (Pos = position only; T+P = time + position); "—" denotes an absent component.

For public benchmark training, we use AdamW with learning rate 10^{-3}, betas $(0.9, 0.98)$, weight decay 0.001, batch size 128, seed 42, and dropout 0.2 for all dropout settings. Models are trained for at least 10 and at most 100 epochs, with a ReduceLROnPlateau scheduler on validation HR@10. The objective is SampledSoftmaxLoss with 128 sampled negatives and temperature 0.05. Histories are truncated to the dataset maximum sequence length. All public training and evaluation runs use a single NVIDIA H100.

Table 7: Hyperparameter settings (defaults for all five datasets; see footnote for the sequence-length exception).

Method	d	H	A	RAB	Bkts	RoPE
SASRec	512	4	—	—	—	—
TiSASRec	512	4	—	—	—	—
HSTU	512	4	128	Pos	—	—
HSTU+Time RAB	512	4	128	T+P	128	—
HSTU+TO-RoPE	512	4	128	T+P	128	TO-RoPE
HSTU+T-RoPE	512	4	128	T+P	128	T-RoPE

† Max sequence length $L = 200$ for ML-20M; $L = 50$ for all other datasets.

E Additional Visualizations

The output attention of T-RoPE is visualized in Figure 5, which shows that the day-, week-, month-, and year-scale event blocks become increasingly grouped by their real time spacing as each component is added, until full T-RoPE cleanly separates the blocks by elapsed time.

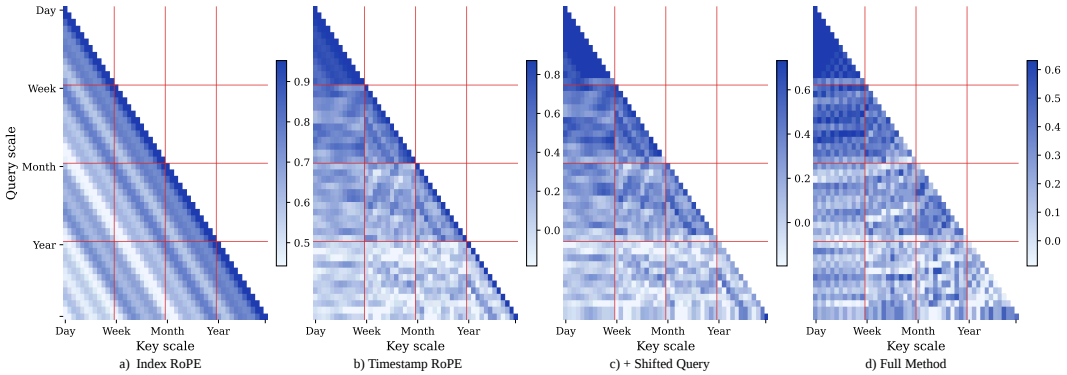


Figure 5: Attention geometry across T-RoPE design stages on a synthetic multiscale sequence. The sequence contains four event blocks whose internal spacing corresponds to day-, week-, month-, and year-scale behavior; red lines mark block boundaries. a) Index RoPE. b) Timestamp RoPE. c) Timestamp RoPE with shifted queries. d) Full T-RoPE with non-stationary keys and learned temporal coefficients. The full method separates the blocks by time geometry rather than only by sequence position.