TRAIL: AN EFFICIENT INDUSTRIAL RANKING FRAMEWORK VIA INTEREST EVOLUTION AND TARGET-RELEVANT AGGREGATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Modeling long user behavior sequences is crucial for click-through rate (CTR) prediction, which requires efficiently capturing user interest evolution over long histories and target-relevant behaviors. Target-aware attention methods are efficient and directly model the relevance between target items and historical behaviors, but they mainly focus on matching the target item with individual behaviors, leaving interest evolution underexplored. Self-attentive models capture behavior interactions, but their sequence modeling is usually performed without conditioning on the target item and incurs quadratic computational cost.

We propose TRAIL, an efficient framework for long sequence CTR prediction via interest evolution and target-relevant aggregation. TRAIL first applies the State-Compressed Gated DeltaNet (SC-GDN) to model interest-state evolution over the raw behavior sequence with linear complexity. It then uses Segmented Target Attention (STA) to construct target-relevant segment representations, reducing sequence length while preserving target-relevant behavioral signals. Over the compressed sequence, Global Stacked Target Attention (GSTA) further aggregates target-relevant information with low computational cost. Finally, a Multi-resolution Interest Fusion module integrates multi-level interest representations for CTR prediction. Experiments show that TRAIL achieves the best AUC and GAUC while using only about 15% of the FLOPs required by the full-attention model. Online A/B testing further yields +2.51% advertiser value and +2.36% revenue gains, demonstrating TRAIL's effectiveness and efficiency in industrial long sequence CTR modeling.

1 INTRODUCTION

CTR prediction is a central task in industrial recommender systems, where user behavior sequences provide important signals about long-term preferences and short-term intents He et al. (2023); Hu et al. (2025). As these histories grow longer, modeling interest evolution and target-relevant behaviors becomes increasingly important, while industrial ranking systems must satisfy strict latency and resource constraints Kang & McAuley (2018); Sun et al. (2019); Xia et al. (2023). Existing approaches face different trade-offs: target-aware methods such as DIN Zhou et al. (2018) efficiently match historical behaviors with the target, but provide limited explicit modeling of interactions and transitions among behaviors; self-attentive models Vaswani et al. (2017); Kang & McAuley (2018); Sun et al. (2019); Zhai et al. (2024) capture such dependencies but are typically target-independent and incur quadratic complexity; and retrieval or compression methods Pi et al. (2020); Chang et al. (2023); Chai et al. (2025); Liu et al. (2023) reduce the effective sequence length but may discard fine-grained temporal dependencies and preference shifts.

Motivated by these observations, we propose TRAIL, an efficient framework for long sequence CTR prediction via interest evolution and target-relevant aggregation. TRAIL first applies a State-Compressed Gated DeltaNet (SC-GDN) encoder to model the recurrent evolution of interest states over the raw behavior sequence with linear complexity. The enhanced sequence is then partitioned into segments, where Segmented Target Attention (STA) constructs target-relevant segment representations. Over the compressed sequence, Global Stacked Target Attention (GSTA) further aggregates target-relevant information to form a global interest representation conditioned on the target

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.

item. Finally, Multi-resolution Interest Fusion combines the global representation and contextual signals for CTR prediction.

Our main contributions are summarized as follows:

- **A practical framework for ultra-long sequence CTR prediction.** We present TRAIL, a practical CTR prediction framework that jointly models user interest evolution and target-relevant behaviors over long behavior histories. TRAIL is designed for industrial ranking systems, where both predictive accuracy and strict computational efficiency are required.
- **Efficient full-history interest evolution and target-relevant aggregation.** TRAIL models fine-grained behavior transitions with linear sequence complexity and performs target-aware interaction over compact segment-level interests, preserving behavior dynamics and global target-relevant signals while handling histories with up to 10,000 interactions.
- **Effectiveness in offline and online evaluation.** TRAIL consistently outperforms representative target-aware and long-sequence baselines on KuaiRand-1K and a large-scale industrial advertising dataset, while using only about 15% of the FLOPs of the full-attention baseline. Online A/B testing further yields +2.51% advertiser value and +2.36% revenue gains.

2 Related Work

CTR prediction models have progressively evolved from modeling short behavior histories to exploiting lifelong user sequences. Early target-aware methods, such as DIN Zhou et al. (2018), construct a target-conditioned interest representation by independently weighting historical behaviors; however, they do not explicitly model dependencies or interest transitions among behaviors. DIEN Zhou et al. (2019) and DSIN Feng et al. (2019) further introduce recurrent interest evolution and session-level modeling. Self-attentive models, including SASRec Kang & McAuley (2018), BERT4Rec Sun et al. (2019), BST Chen et al. (2019), and TransAct Xia et al. (2023), improve long-range dependency modeling through behavior interactions, while their attention cost becomes challenging as sequence length increases. To enable long-sequence serving, SIM Pi et al. (2020), TWIN Chang et al. (2023), and TWIN V2 Si et al. (2024) retrieve or compress target-relevant behaviors before ranking, and LONGER Chai et al. (2025) further explores efficient ultra-long sequence modeling. However, early retrieval or compression may discard fine-grained interest transitions before they are sufficiently encoded. More recently, STCA Guan et al. (2026) replaces history self-attention with stacked target-to-history cross-attention, enabling end-to-end modeling of ultra-long behavior sequences with linear complexity. Nevertheless, existing methods still face a fundamental trade-off between modeling effectiveness and serving efficiency when scaling to longer behavior sequences.

3 Method

3.1 Problem Formulation

In industrial recommendation systems, the CTR prediction task in the ranking stage aims to accurately estimate the probability that a given user will click a target item. We formulate CTR prediction as a conditional probability estimation problem. Given a historical behavior sequence $\mathcal{H} = [\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_L]$, a target item $\mathbf{t}$, and a user-context representation $\mathbf{c}$ that encodes user-profile and request-context features, our objective is to learn a function $f(\cdot)$ that estimates the click probability:

$$P(\text{click} = 1 \mid \mathcal{H}, \mathbf{t}, \mathbf{c}) = \sigma\big(f(\mathcal{H}, \mathbf{t}, \mathbf{c}; \theta)\big), \tag{1}$$

where $\sigma(\cdot)$ denotes the sigmoid function, θ represents the model parameters, $\mathbf{h}_n \in \mathbb{R}^d$ is the d-dimensional representation of the n-th historical interaction, and L denotes the sequence length.

The training objective minimizes the binary cross-entropy loss:

$$\mathcal{L} = -\frac{1}{|\mathcal{D}|} \sum_{(\mathcal{H}, \mathbf{t}, \mathbf{c}, y) \in \mathcal{D}} \big[y \log \hat{y} + (1 - y) \log(1 - \hat{y}) \big], \tag{2}$$

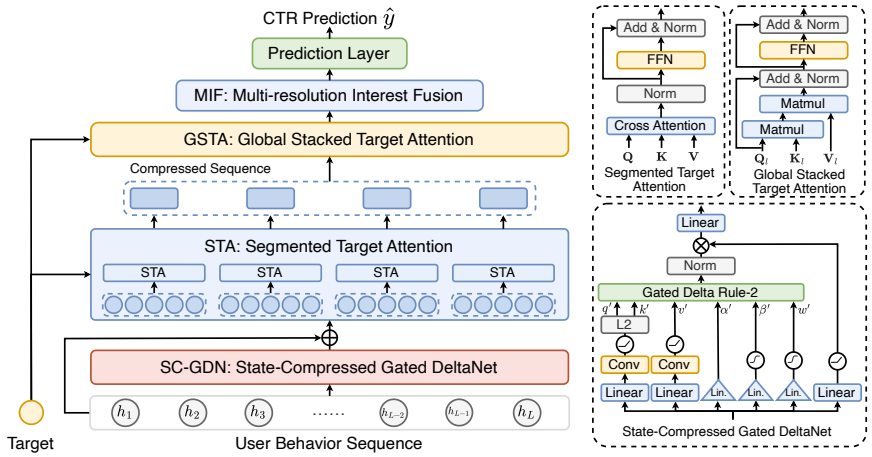


Figure 1: Overview of the TRAIL framework. TRAIL models interest-state evolution over raw behavior sequences, constructs target-relevant segment representations, aggregates target-relevant information over the compressed sequence, and combines complementary interest signals for CTR prediction.

where $\mathcal{D}$ denotes the training dataset, $y \in \{0, 1\}$ is the ground-truth label, and $\hat{y} = P(\text{click} = 1 \mid \mathcal{H}, \mathbf{t}, \mathbf{c})$.

3.2 Overview

As illustrated in Figure 1, TRAIL consists of four components. (1) A State-Compressed Gated DeltaNet (SC-GDN) encoder models fine-grained interest-state evolution over the raw behavior sequence $\mathcal{H}$, producing a sequence enhanced with interest state information, denoted as $\tilde{\mathcal{H}}$. (2) Segmented Target Attention (STA) constructs target-relevant segment representations from $\tilde{\mathcal{H}}$ within fixed-size segments, reducing the effective length from L to $L' = L/w$. (3) Global Stacked Target Attention (GSTA) aggregates target-relevant information over the compressed sequence $\mathcal{H}'$ to produce a global interest representation conditioned on the target item. (4) A Multi-resolution Interest Fusion module integrates the global interest with complementary signals (average interest, salient interest, recent interest, and user-context features) and feeds them into a prediction layer for CTR estimation.

3.3 Model Structure

3.3.1 Input.

We first construct the input tokens for the user behavior sequence and the target item. Each historical item is represented by its ID embedding and multimodal embeddings. We concatenate the embeddings of these features and apply a linear projection to obtain the d-dimensional behavior and target tokens. We also include a recency feature, representing the time difference between the historical event and the request time, encoded as positional embeddings. Formally, we obtain the user sequence tokens $\mathcal{H} = [\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_L] \in \mathbb{R}^{L \times d}$ and the target item token $\mathbf{t} \in \mathbb{R}^d$.

3.3.2 Interest-State Encoding via State-Compressed Gated Delta Networks.

Existing efficient long-sequence CTR models usually reduce computation by shortening long behavior histories, but this may weaken fine-grained user-interest evolution. Full-sequence self-attention can model behavior dependencies, yet its quadratic complexity is impractical for long histories. To address this, TRAIL introduces a lightweight linear-complexity interest-state encoder to enrich behavior representations before target-aware aggregation.

We build this encoder based on Gated DeltaNet-2 (GDN2) Hatamizadeh et al. (2026), a gated recurrent sequence model that maintains a key–value state and updates it through input-dependent gates. Although GDN2 avoids the quadratic pairwise interactions of self-attention, the recurrent state can still be costly for long histories. To reduce this cost, we propose **State-Compressed Gated DeltaNet (SC-GDN)**, which performs the original GDN2 recurrence in compact key and value subspaces while retaining all other operations, including local convolution and gate activations.

Given a behavior representation $\mathbf{h}_n \in \mathbb{R}^d$, SC-GDN projects its query, key, and value into compact spaces:

$$\mathbf{q}'_n = \mathbf{h}_n \mathbf{W}'_Q, \qquad \mathbf{k}'_n = \mathbf{h}_n \mathbf{W}'_K, \qquad \mathbf{v}'_n = \mathbf{h}_n \mathbf{W}'_V, \tag{3}$$

where $\mathbf{W}'_Q, \mathbf{W}'_K \in \mathbb{R}^{d \times d'_k}$ and $\mathbf{W}'_V \in \mathbb{R}^{d \times d'_v}$. Therefore, $\mathbf{q}'_n, \mathbf{k}'_n \in \mathbb{R}^{d'_k}$ and $\mathbf{v}'_n \in \mathbb{R}^{d'_v}$.

In addition to $\mathbf{q}'_n$, $\mathbf{k}'_n$, and $\mathbf{v}'_n$, the recurrent update takes three learned variables derived from $\mathbf{h}_n$: the decay variable $\boldsymbol{\alpha}'_n$, the erase variable $\boldsymbol{\beta}'_n$, and the write variable $\mathbf{w}'_n$. Their dimensions are $\boldsymbol{\alpha}'_n, \boldsymbol{\beta}'_n \in \mathbb{R}^{d'_k}$ and $\mathbf{w}'_n \in \mathbb{R}^{d'_v}$. Denoting Gated Delta Rule-2 as GDR2, the recurrent computation is written as:

$$\begin{aligned} (\mathbf{o}_n, \mathbf{S}_n) &= \mathrm{GDR2}\left(\mathbf{q}'_n, \mathbf{k}'_n, \mathbf{v}'_n, \boldsymbol{\alpha}'_n, \boldsymbol{\beta}'_n, \mathbf{w}'_n, \mathbf{S}_{n-1}\right), \\ \mathbf{S}_n &\in \mathbb{R}^{d'_k \times d'_v}, \quad \mathbf{o}_n \in \mathbb{R}^{d'_v}. \end{aligned} \tag{4}$$

Since $d'_k, d'_v < d$, $\mathbf{S}_n$ is the $O(d'_k d'_v)$ compressed key–value state over the first n behaviors, and $\mathbf{o}_n$ is its readout for query $\mathbf{q}'_n$.

Finally, the recurrent output is projected back to the model dimension and combined with the original behavior representation:

$$\tilde{\mathbf{h}}_n = \mathbf{h}_n + \mathrm{Proj}(\mathbf{o}_n), \quad \tilde{\mathcal{H}} = [\tilde{\mathbf{h}}_1, \ldots, \tilde{\mathbf{h}}_L]. \tag{5}$$

The resulting sequence preserves the original sequence length while enriching each behavior representation with the evolving interest state. The enhanced sequence $\tilde{\mathcal{H}}$ is then fed into the subsequent target-aware compression module.

3.3.3 Fine-grained Target-aware Compression.

Although SC-GDN enriches each token in the enhanced sequence $\tilde{\mathcal{H}}$ with user interest evolution, it still contains L behavior tokens, making downstream interaction modeling over long histories computationally expensive and susceptible to irrelevant behaviors. To address this, we introduce the **Segmented Target Attention (STA)** module, which constructs a shorter segment-level sequence while preserving target-relevant information.

First, the enhanced sequence $\tilde{\mathcal{H}}$ produced by SC-GDN is partitioned into L' non-overlapping segments, each containing w consecutive tokens:

$$[\mathcal{S}_1, \ldots, \mathcal{S}_{L'}] = \mathrm{Split}(\tilde{\mathcal{H}}, w), \tag{6}$$

where w denotes the segment size, $L' = L/w$ is the number of resulting segments, and the maximum sequence length L is chosen to be divisible by w, with shorter histories padded accordingly. This segmentation strategy serves a dual purpose: (1) **Complexity Reduction:** It decouples local pattern extraction from global dependency modeling, reducing the sequence length for subsequent layers; (2) **Target-aware Filtering:** Within each segment, target-aware attention aggregates target-relevant behaviors and filters locally irrelevant signals.

To keep the target-aware compression efficient, we adopt a **low-dimensional projection** that maps the query and keys to reduced dimensions d_q and d_k, respectively, with $d_q = d_k < d$. This lowers their projection cost from $O(Ld^2)$ to $O(Ldd_q)$, while retaining d-dimensional values.

For each segment $\mathcal{S}_i$, we then compute a target-aware attention score. The query is the projection of the target $\mathbf{t}$ through $\mathbf{W}_q \in \mathbb{R}^{d \times d_q}$, while the keys and values are obtained from the segment tokens via $\mathbf{W}_k \in \mathbb{R}^{d \times d_k}$ and $\mathbf{W}_v \in \mathbb{R}^{d \times d}$:

$$\mathbf{Q} = \mathbf{t}\mathbf{W}_q, \quad \mathbf{K} = \mathcal{S}_i \mathbf{W}_k, \quad \mathbf{V} = \mathcal{S}_i \mathbf{W}_v, \tag{7}$$

$$\mathbf{s}_i = \mathrm{LN}\left(\mathrm{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_k}}\right)\mathbf{V}\mathbf{W}_o\right). \tag{8}$$

Here, LN is layer normalization; all projections, including $\mathbf{W}_o \in \mathbb{R}^{d\times d}$, are shared across segments. Padding tokens are excluded from the softmax using a token-level mask.

The projected segment representations $\{\mathbf{s}_i\}$ are further refined by an FFN:

$$\mathbf{s}_i \leftarrow \mathrm{LN}\big(\mathrm{FFN}(\mathbf{s}_i) + \mathbf{s}_i\big). \tag{9}$$

Finally, the segments are fused into a compact sequence $\mathcal{H}' = [\mathbf{s}_1, \ldots, \mathbf{s}_{L'}] \in \mathbb{R}^{L'\times d}$, encapsulating target-relevant patterns at a segment-level granularity.

3.3.4 Global Stacked Target Attention over Compressed Sequence.

After compression, the sequence length is reduced by a factor of w, allowing us to apply deep interaction layers at a relatively low cost. We employ a **Global Stacked Target Attention (GSTA)** module that stacks L_{stack} target-attention layers to capture target-mediated cross-segment dependencies and iteratively refine the target-conditioned global interest representation over the compressed segments. Let $l = 1, \ldots, L_{\text{stack}}$ index the GSTA layers. The input target representation to layer l is $\mathbf{t}_{l-1}$, where $\mathbf{t}_0 = \mathbf{t}$ initializes the first layer. The attention process is defined as:

$$\mathbf{Q}_l = \mathbf{t}_{l-1}\mathbf{W}_q^l, \quad \mathbf{K}_l = \mathcal{H}'\mathbf{W}_k^l, \quad \mathbf{V}_l = \mathcal{H}'\mathbf{W}_v^l, \tag{10}$$

$$\mathbf{o}_l = \mathrm{softmax}\left(\frac{\mathbf{Q}_l\mathbf{K}_l^\top}{\sqrt{d}}\right)\mathbf{V}_l\mathbf{W}_o^l. \tag{11}$$

Here, $\mathbf{W}_q^l, \mathbf{W}_k^l, \mathbf{W}_v^l, \mathbf{W}_o^l \in \mathbb{R}^{d\times d}$ are the projection matrices at layer l, and the softmax is computed only over segment representations containing valid behaviors. The attention output is added back to the target through a residual connection and layer normalization, followed by a feed-forward network (FFN) with a second residual connection:

$$\tilde{\mathbf{t}}_l = \mathrm{LN}(\mathbf{t}_{l-1} + \mathbf{o}_l), \quad \mathbf{t}_l = \mathrm{LN}\big(\tilde{\mathbf{t}}_l + \mathrm{FFN}(\tilde{\mathbf{t}}_l)\big). \tag{12}$$

The final output $\mathbf{z} = \mathbf{t}_{L_{\text{stack}}} \in \mathbb{R}^d$ serves as the target-conditioned global interest representation. With stacking ($L_{\text{stack}} \geq 2$), GSTA progressively refines this representation through target-aware aggregation over the compressed segments. Specifically, each layer uses the current target representation to attend to segment values from $\mathcal{H}'$ and incorporates the aggregated information back into the target representation. Subsequent layers then perform attention with this enriched representation, allowing the model to gradually filter and refine global interest information relevant to the target item. This design captures global signals over compressed segments efficiently, without explicitly computing all pairwise segment-to-segment attentions.

3.3.5 Multi-resolution Interest Fusion.

The Multi-resolution Interest Fusion (MIF) module combines complementary views of the compressed sequence $\mathcal{H}' = [\mathbf{s}_1, \ldots, \mathbf{s}_{L'}]$ to capture user interests from multiple perspectives:

- **Global Context ($\mathbf{z}$):** the target-conditioned global interest from GSTA, capturing long-term, cross-segment dependencies.
- **Average Interest ($\bar{\mathbf{h}} = \mathrm{meanpool}(\mathcal{H}')$):** the mean over segments, summarizing the overall interest distribution.
- **Salient Interest ($\hat{\mathbf{h}} = \mathrm{maxpool}(\mathcal{H}')$):** max pooling over segment representations, preserving the strongest signals across the history.
- **Recent Interest ($\mathbf{s}_{L'-r+1}, \ldots, \mathbf{s}_{L'}$):** the most recent r valid segment representations are retained individually to complement the pooled summaries, with zero padding when fewer than r are available.

These views are concatenated with the projected user-context representation $\mathbf{c} \in \mathbb{R}^d$ into a fused representation:

$$\mathbf{f} = \big[\,\mathbf{z}\,;\,\bar{\mathbf{h}}\,;\,\hat{\mathbf{h}}\,;\,\mathbf{s}_{L'-r+1}\,;\,\cdots\,;\,\mathbf{s}_{L'}\,;\,\mathbf{c}\,\big], \tag{13}$$

The fused representation $\mathbf{f} \in \mathbb{R}^{(4+r)d}$ is projected by $\mathbf{W}_f \in \mathbb{R}^{(4+r)d\times Td}$, reshaped into T tokens of dimension d, and passed through R RankMixer Zhu et al. (2025) blocks followed by a prediction tower for CTR estimation:

$$\hat{y} = \sigma\big(\mathrm{Tower}(\mathrm{RankMixer}(\mathrm{reshape}(\mathbf{f}\mathbf{W}_f)))\big), \tag{14}$$

where r is the number of recent segments and $\sigma(\cdot)$ is the sigmoid.

3.4 Efficiency Discussion

TRAIL supports efficient long sequence CTR prediction by modeling interest evolution with linear complexity over L behaviors and performing target-relevant aggregation only after the sequence is reduced to $L' = L/w$ segment representations. We use ρ to denote the common FFN expansion ratio. The computational complexities of the four modules are summarized as follows:

$$\begin{aligned} C_{\text{SC-GDN}} &= O(L\,[d(d'_k + d'_v) + d'_k d'_v])\,, \\ C_{\text{STA}} &= O\big(Ld(d + d_q) + L'\rho d^2\big)\,, \\ C_{\text{GSTA}} &= O\big(L_{\text{stack}}(L' + \rho)d^2\big)\,, \\ C_{\text{MIF}} &= O\big(T(4 + r + R\rho)d^2\big)\,, \\ C_{\text{TRAIL}} &= C_{\text{SC-GDN}} + C_{\text{STA}} \\ &\quad + C_{\text{GSTA}} + C_{\text{MIF}} = O(Ld^2). \end{aligned} \tag{15}$$

Under fixed projection dimensions, segment size, number of stacked target-attention layers, and MIF configuration, TRAIL scales linearly with L. In contrast, full-sequence self-attention incurs $O(L^2 d)$ pairwise token interactions. This linear scaling enables TRAIL to model interest evolution over the full raw history while keeping target-relevant aggregation computationally tractable. Detailed FLOPs derivations and module-wise counts are provided in Appendix.

4 Experiments

4.1 Experimental Setup

Dataset. We evaluate TRAIL on both a public dataset and a large-scale industrial dataset. The public dataset is KuaiRand-1K Gao et al. (2022), a short-video recommendation benchmark from Kuaishou, containing 1,000 users, 4.37M videos, and 11.7M interaction records. We use its standard recommendation logs for supervised CTR prediction and adopt a strict chronological split, dividing the interactions in temporal order into training, validation, and test sets with an 8:1:1 ratio to prevent temporal data leakage. The industrial dataset is collected from a large-scale production advertising system and comprises hundreds of millions of daily ad impression logs. Each sample corresponds to an ad exposure event, including user profile features, candidate ad features, contextual features, and the user's historical behavior sequence, with a binary click label. To ensure temporal validity and prevent data leakage, we adopt a strict time-based split, using 30 consecutive days for training and the subsequent 3 days for testing. For both datasets, user behavior sequences are constructed using only events before the target impression.

Baselines. We compare TRAIL with representative CTR sequence modeling baselines, including the target-aware DIN Zhou et al. (2018), two-stage retrieval model TWIN Chang et al. (2023), the full-sequence self-attention model HSTU Zhai et al. (2024), and recent long-sequence models STCA Guan et al. (2026) and LONGER Chai et al. (2025).

Evaluation Metrics. We report AUC and GAUC for offline evaluation. AUC measures overall ranking performance, while GAUC computes user-level AUC and weights it by the number of samples, better reflecting personalized recommendation quality. For online A/B testing on the advertising system, we report advertiser value (ADVV) and revenue as business metrics.

Implementation Details. All experiments are conducted using NVIDIA H20 GPUs, with a default batch size of 4096 that is reduced when memory is insufficient. For a fair comparison, all models use an embedding dimension and hidden dimension of 128, 2 attention layers, 2 attention heads, a dropout rate of 0.1, and a maximum history length of $L = 1000$. For TRAIL, we set the SC-GDN compressed key and value dimensions to $d'_k = 16$ and $d'_v = 64$, the local convolution size to $k_{\text{conv}} = 3$, the STA low-dimensional projection size to $d_q = 16$, the STA segment size to $w = 10$, the FFN expansion ratio to $\rho = 4$, and the number of GSTA layers to $L_{\text{stack}} = 2$. For Multi-resolution Interest Fusion, we retain $r = 5$ recent segments and use $T = 8$ RankMixer tokens of dimension d with $R = 2$ layers. For KuaiRand-1K experiments, we apply early stopping on validation AUC with patience of 5. Each offline experiment is repeated five times with different random seeds, and the standard deviations of AUC and GAUC are below 0.00025.

Table 1: Overall performance comparison on KuaiRand-1K and the industrial dataset.

Model	KuaiRand-1K				Industrial Dataset				FLOPs (M)
	AUC	ΔAUC	GAUC	ΔGAUC	AUC	ΔAUC	GAUC	ΔGAUC	
DIN	0.8136	-0.76%	0.7262	-1.28%	0.7728	-1.04%	0.7138	-1.08%	28
TWIN	0.8129	-0.83%	0.7257	-1.33%	0.7709	-1.23%	0.7119	-1.27%	–
HSTU	0.8188	-0.24%	0.7348	-0.42%	0.7803	-0.29%	0.7213	-0.33%	1354
STCA	0.8193	-0.19%	0.7357	-0.33%	0.7810	-0.22%	0.7222	-0.24%	920
LONGER	0.8195	-0.17%	0.7365	-0.25%	0.7817	-0.15%	0.7229	-0.17%	172
TRAIL	**0.8212**	–	**0.7390**	–	**0.7832**	–	**0.7246**	–	199

4.2 Overall Performance

Table 1 summarizes the comparison between TRAIL and representative baselines on KuaiRand-1K and the industrial dataset. TRAIL achieves the best AUC and GAUC on both datasets, reaching 0.8212 and 0.7390 on KuaiRand-1K, and 0.7832 and 0.7246 on the industrial dataset.

Compared with DIN and TWIN, TRAIL improves AUC by 0.76% and 0.83%, and GAUC by 1.28% and 1.33% on KuaiRand-1K, respectively. The gains remain consistent on the industrial dataset, with AUC improvements of 1.04% and 1.23%, and GAUC improvements of 1.08% and 1.27%, respectively. These results indicate that modeling interest evolution over the complete behavior history provides more informative representations than independent behavior matching or retrieval-based modeling.

TRAIL also outperforms HSTU, STCA, and LONGER with 199M FLOPs, corresponding to about 15% of the cost of HSTU and 22% of STCA. Compared with HSTU and STCA, TRAIL improves both AUC and GAUC on the two datasets, while incurring only 27M additional FLOPs over LONGER and achieving consistently better performance. This result suggests that encoding interest evolution before segment-level aggregation provides a favorable trade-off between prediction quality and efficiency.

4.3 Ablation Study

We ablate the key components of TRAIL on KuaiRand-1K to examine their contributions to long-history modeling. Specifically, we evaluate five variants: replacing the target-aware aggregation in STA with intra-segment mean pooling (STA-Mean), removing STA entirely, removing SC-GDN before sequence compression, replacing GSTA with a single layer (GSTA-1L), and removing the MIF module. The results are shown in Table 2.

Table 2: Ablation study on KuaiRand-1K.

Variant	AUC	ΔAUC	GAUC	ΔGAUC
TRAIL	**0.8212**	–	**0.7390**	–
STA-Mean	0.8200	-0.12%	0.7370	-0.20%
w/o STA	0.8207	-0.05%	0.7383	-0.07%
w/o SC-GDN	0.8206	-0.06%	0.7381	-0.09%
GSTA-1L	0.8209	-0.03%	0.7387	-0.03%
w/o MIF	0.8207	-0.05%	0.7377	-0.13%

We make the following observations. **1) Replacing STA aggregation with mean pooling.** STA-Mean causes the largest degradation, with a 0.12% AUC drop and a 0.20% GAUC drop. Since it keeps the same segmentation structure and compressed length, the decline shows that compression alone is insufficient; target-aware selection is crucial for retaining relevant behaviors. **2) Removing STA.** Removing STA reduces AUC and GAUC by 0.05% and 0.07%, respectively. In this case, global target-aware modeling is applied directly to the full enhanced sequence without local target-aware filtering, demonstrating the importance of constructing target-relevant segment representations before global aggregation. **3) Removing SC-GDN.** Removing SC-GDN leads to 0.06% AUC and 0.09% GAUC decreases, suggesting that modeling interest-state evolution before compression enriches behavior representations and produces more expressive segment-level interests. **4) Replacing GSTA with a single target-attention layer.** This variant reduces both AUC and GAUC by 0.03%, indicating that stacked global aggregation progressively refines target-relevant interests from the compressed sequence. **5) Removing MIF.** Removing MIF reduces AUC and GAUC by 0.05% and 0.13%, respectively, indicating that the global target-conditioned representation from GSTA benefits from complementary signals. The

larger GAUC drop further suggests that combining these interest resolutions improves personalized ranking across users. Overall, these ablations validate TRAIL's core design: modeling interest evolution over the full history, constructing compact target-aware segment representations, aggregating global target-relevant information, and fusing multi-resolution interests for prediction.

4.4 Sensitivity Analysis

Beyond component-level ablations, we analyze the sensitivity of TRAIL to key hyperparameters in STA and SC-GDN on KuaiRand-1K. For STA, we examine the segment size and projection dimension. For SC-GDN, we study the compact state dimensions and local convolution size. The reported FLOPs correspond to the total per-sample modeling cost of TRAIL under each configuration.

Effect of STA Configuration. We study the sensitivity of STA to the segment size w and the projection size d_q. As shown in Table 3, $w = 10$ achieves the best observed overall performance. Smaller segments provide limited behavioral context and produce longer segment sequences, whereas larger segments improve compression but may lose fine-grained behavioral information. The FLOPs reveal a clear trade-off between prediction performance and computational cost. Reducing w from 10 to 5 increases the cost from 199.3M to 242.0M without improving accuracy, whereas larger segments reduce computation but gradually degrade performance. Thus, w should be adjusted according to the original sequence length L to balance local interest modeling and sequence compression. For the projection dimension, increasing $d_q = d_k$ beyond 16 raises the FLOPs from 199.3M to as much as 228.2M, while providing relatively limited performance gains. Therefore, under $L = 1000$, we set $d_q = 16$ and $w = 10$ to maintain computational efficiency while preserving competitive prediction accuracy.

Table 3: Sensitivity of STA configurations on KuaiRand-1K.

w	5	10	20	25	50
AUC	0.8207	0.8212	0.8210	0.8208	0.8201
GAUC	0.7382	0.7390	0.7389	0.7384	0.7372
FLOPs (M)	242.0	199.3	177.9	173.6	165.1
$d_q = d_k$	8	16	32	64	128
AUC	0.8210	0.8212	0.8214	0.8216	0.8216
GAUC	0.7387	0.7390	0.7394	0.7396	0.7395
FLOPs (M)	197.2	199.3	203.4	211.7	228.2

Effect of SC-GDN Configuration. We next study the sensitivity of SC-GDN to the compressed key dimension d'_k, value dimension d'_v, and local convolution size. As shown in Table 4, increasing d'_k beyond 16 results in only marginal performance gains, indicating that a compact key state is sufficient for modeling behavior transition patterns. The FLOPs further show that the compact state dimensions dominate the computational cost of SC-GDN. Increasing d'_v improves performance from 16 to 64, while $d'_v = 128$ raises the total cost from 199.3M to 315.4M for only a small GAUC gain. Therefore, we use $d'_v = 64$ as a better trade-off between effectiveness and efficiency. For the local convolution size, removing convolution degrades performance, while a moderate size of 3 improves both metrics with only 0.6M additional FLOPs and achieves the best overall result. Larger kernels introduce additional cost without further performance gains, suggesting that lightweight local context modeling is sufficient before recurrent state updates. These results indicate that a lightweight SC-GDN configuration can effectively capture interest evolution with compact states and moderate local context modeling.

Table 4: Sensitivity of SC-GDN configurations on KuaiRand-1K.

d'_k	8	16	32	64
AUC	0.8210	0.8212	0.8212	0.8212
GAUC	0.7385	0.7390	0.7389	0.7389
FLOPs (M)	188.4	199.3	221.0	264.3
d'_v	16	32	64	128
AUC	0.8208	0.8210	0.8212	0.8212
GAUC	0.7384	0.7386	0.7390	0.7392
FLOPs (M)	122.9	147.4	199.3	315.4
Conv. size	None	3	5	10
AUC	0.8209	0.8212	0.8212	0.8210
GAUC	0.7386	0.7390	0.7387	0.7385
FLOPs (M)	198.7	199.3	199.6	200.6

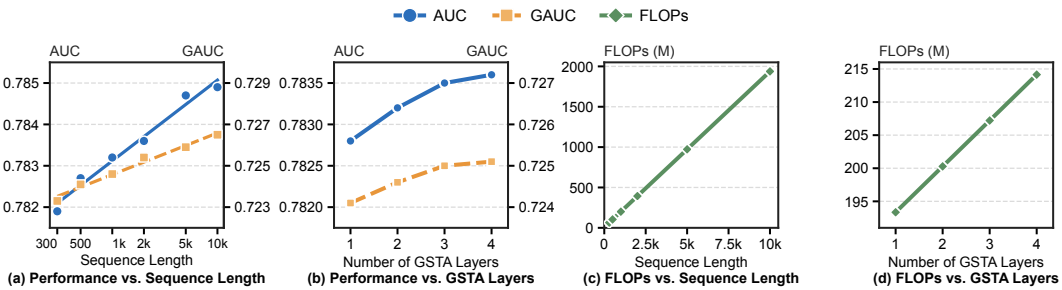


Figure 2: Scaling analysis on the industrial dataset.

4.5 Scaling Analysis

We analyze how TRAIL scales with behavior sequence length and the number of GSTA layers on the industrial dataset. Figure 2 reports the corresponding performance and per-sample FLOPs.

Scaling with Sequence Length. As shown in Figure 2(a), increasing the maximum behavior sequence length consistently improves both AUC and GAUC, with diminishing marginal gains at longer lengths. Meanwhile, Figure 2(c) shows that the FLOPs scale linearly with sequence length, consistent with the computational complexity of TRAIL.

Scaling with GSTA Depth. As shown in Figure 2(b), increasing the number of GSTA layers improves both AUC and GAUC. Meanwhile, Figure 2(d) shows that the FLOPs increase moderately because these layers operate on a reduced sequence. Overall, TRAIL can use longer behavior histories and deeper target-aware aggregation while maintaining manageable computational cost.

4.6 Online A/B Test

After validating TRAIL through offline experiments and component ablations, we further conduct an online A/B test in a large-scale production advertising system to evaluate whether the offline improvements translate into real production gains. The experiment lasts for 7 days, with 10% of the total traffic allocated to the experimental group under strict traffic control. We report two business metrics: advertiser value (ADVV) and revenue.

Table 5: Online lift of TRAIL in a large-scale production advertising system.

Metric	Lift
ADVV	+2.51%
Revenue	+2.36%

As shown in Table 5, TRAIL achieves a +2.51% lift in ADVV and a +2.36% lift in revenue. The online gains indicate that modeling interest evolution over long behavior histories and constructing target-relevant interests can improve advertising ranking under production serving constraints. These results show that TRAIL improves not only offline ranking accuracy but also production-level advertising effectiveness.

5 Conclusion

In this paper, we presented TRAIL, an efficient framework for long sequence CTR prediction via interest evolution and target-relevant aggregation. TRAIL first models interest-state evolution over complete behavior histories and then performs target-aware compression and global aggregation, preserving informative long-sequence signals while controlling serving cost. Experiments on KuaiRand-1K and a large-scale industrial dataset show that TRAIL achieves strong ranking performance with only 199M FLOPs per sample, while online A/B testing yields +2.51% advertiser value and +2.36% revenue gains. These results demonstrate the effectiveness of modeling interest evolution before target-aware aggregation for industrial long-sequence CTR prediction.

AI USE STATEMENT

In this work, generative AI tools were used to assist with language polishing and grammatical refinement. The research ideas, methodology, experiments, analyses, results, and conclusions were developed by the authors. All AI-assisted edits were reviewed by the authors, who take full responsibility for the final content of this work.

REPRODUCIBILITY STATEMENT

This paper provides: (1) descriptions of the datasets, data splits, baselines, and evaluation protocols in the Experimental Setup section; (2) a detailed description of the proposed TRAIL framework and its underlying methodology in the Method section; and (3) training configurations, model dimensions, and hyperparameter settings in the Implementation Details section. Results are reported on both a public benchmark and a large-scale industrial dataset using multiple ranking metrics.

REFERENCES

Zheng Chai, Qin Ren, Xijun Xiao, Huizhi Yang, Bo Han, Sijun Zhang, Di Chen, Hui Lu, Wenlin Zhao, Lele Yu, Xionghang Xie, Shiru Ren, Xiang Sun, Yaocheng Tan, Peng Xu, Yuchao Zheng, and Di Wu. LONGER: Scaling up long sequence modeling in industrial recommenders. In *Proceedings of the Nineteenth ACM Conference on Recommender Systems*, pp. 247–256, 2025. doi: 10.1145/3705328.3748065. URL https://doi.org/10.1145/3705328.3748065.

Jianxin Chang, Chenbin Zhang, Zhiyi Fu, Xiaoxue Zang, Lin Guan, Jing Lu, Yiqun Hui, Dewei Leng, Yanan Niu, Yang Song, and Kun Gai. TWIN: TWo-stage interest network for lifelong user behavior modeling in CTR prediction at Kuaishou. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 3785–3794, 2023. doi: 10.1145/3580305.3599922. URL https://doi.org/10.1145/3580305.3599922.

Qiwei Chen, Huan Zhao, Wei Li, Pipei Huang, and Wenwu Ou. Behavior sequence transformer for e-commerce recommendation in Alibaba. In *Proceedings of the 1st International Workshop on Deep Learning Practice for High-Dimensional Sparse Data*, pp. 1–4, 2019. doi: 10.1145/3326937.3341261. URL https://doi.org/10.1145/3326937.3341261.

Yufei Feng, Fuyu Lv, Weichen Shen, Menghan Wang, Fei Sun, Yu Zhu, and Keping Yang. Deep session interest network for click-through rate prediction. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pp. 2301–2307, 2019. doi: 10.24963/ijcai.2019/319. URL https://doi.org/10.24963/ijcai.2019/319.

Chongming Gao, Shijun Li, Yuan Zhang, Jiawei Chen, Biao Li, Wenqiang Lei, Peng Jiang, and Xiangnan He. KuaiRand: An unbiased sequential recommendation dataset with randomly exposed videos. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pp. 3953–3957, 2022. doi: 10.1145/3511808.3557624. URL https://doi.org/10.1145/3511808.3557624.

Lin Guan, Jia-Qi Yang, Zhishan Zhao, Beichuan Zhang, Bo Sun, Xuanyuan Luo, Jinan Ni, Xiaowen Li, Yuhang Qi, Zhifang Fan, Hangyu Wang, Qiwei Chen, Yi Cheng, Feng Zhang, and Xiao Yang. Make it long, keep it fast: End-to-end 10k-sequence modeling at billion scale on Douyin. In *Proceedings of the ACM Web Conference 2026*, pp. 1193–1203, 2026. doi: 10.1145/3774904.3792811. URL https://doi.org/10.1145/3774904.3792811.

Ali Hatamizadeh, Yejin Choi, and Jan Kautz. Gated DeltaNet-2: Decoupling erase and write in linear attention, 2026. URL https://arxiv.org/abs/2605.22791.

Zhicheng He, Weiwen Liu, Wei Guo, Jiarui Qin, Yingxue Zhang, Yaochen Hu, and Ruiming Tang. A survey on user behavior modeling in recommender systems. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, pp. 6656–6664, 2023. doi: 10.24963/ijcai.2023/746. URL https://doi.org/10.24963/ijcai.2023/746.

Xian Hu, Ming Yue, Zhixiang Feng, Junwei Pan, Junjie Zhai, Ximei Wang, Xinrui Miao, Qian Li, Xun Liu, Shangyu Zhang, Letian Wang, Hua Lu, Zijian Zeng, Chen Cai, Wei Wang, Fei Xiong,

Pengfei Xiong, Jintao Zhang, Zhiyuan Wu, Chunhui Zhang, Anan Liu, Jiulong You, Chao Deng, Yuekui Yang, Shudong Huang, Dapeng Liu, and Haijie Gu. Practice on long behavior sequence modeling in Tencent advertising, 2025. URL https://arxiv.org/abs/2510.21714.

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

Qi Liu, Xuyang Hou, Haoran Jin, Xiaolong Chen, Jin Chen, Defu Lian, Zhe Wang, Jia Cheng, and Jun Lei. Deep group interest modeling of full lifelong user behaviors for CTR prediction, 2023. URL https://arxiv.org/abs/2311.10764.

Qi Pi, Guorui Zhou, Yujing Zhang, Zhe Wang, Lejian Ren, Ying Fan, Xiaoqiang Zhu, and Kun Gai. Search-based user interest modeling with lifelong sequential behavior data for click-through rate prediction. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pp. 2685–2692, 2020. doi: 10.1145/3340531.3412744. URL https://doi.org/10.1145/3340531.3412744.

Zihua Si, Lin Guan, ZhongXiang Sun, Xiaoxue Zang, Jing Lu, Yiqun Hui, Xingchao Cao, Zeyu Yang, Yichen Zheng, Dewei Leng, Kai Zheng, Chenbin Zhang, Yanan Niu, Yang Song, and Kun Gai. TWIN V2: Scaling ultra-long user behavior sequence modeling for enhanced CTR prediction at Kuaishou. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, pp. 4890–4897, 2024. doi: 10.1145/3627673.3680030. URL https://doi.org/10.1145/3627673.3680030.

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*, pp. 1441–1450, 2019. doi: 10.1145/3357384.3357895. URL https://doi.org/10.1145/3357384.3357895.

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.

Xue Xia, Pong Eksombatchai, Nikil Pancha, Dhruvil Deven Badani, Po-Wei Wang, Neng Gu, Saurabh Vishwas Joshi, Nazanin Farahpour, Zhiyuan Zhang, and Andrew Zhai. TransAct: Transformer-based realtime user action model for recommendation at Pinterest. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 5249–5259, 2023. doi: 10.1145/3580305.3599918. URL https://doi.org/10.1145/3580305.3599918.

Jiaqi Zhai, Lucy Liao, Xing Liu, Yueming Wang, Rui Li, Xuan Cao, Leon Gao, Zhaojie Gong, Fangda Gu, Jiayuan He, Yinghai Lu, and Yu Shi. Actions speak louder than words: Trillion-parameter sequential transducers for generative recommendations. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 58484–58509. PMLR, 2024. URL https://proceedings.mlr.press/v235/zhai24a.html.

Guorui Zhou, Xiaoqiang Zhu, Chengru Song, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. Deep interest network for click-through rate prediction. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1059–1068, 2018. doi: 10.1145/3219819.3219823. URL https://doi.org/10.1145/3219819.3219823.

Guorui Zhou, Na Mou, Ying Fan, Qi Pi, Weijie Bian, Chang Zhou, Xiaoqiang Zhu, and Kun Gai. Deep interest evolution network for click-through rate prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 5941–5948, 2019. doi: 10.1609/aaai.v33i01.33015941. URL https://doi.org/10.1609/aaai.v33i01.33015941.

Jie Zhu, Zhifang Fan, Xiaoxie Zhu, Yuchen Jiang, Hangyu Wang, Xintian Han, Haoran Ding, Xinmin Wang, Wenlin Zhao, Zhen Gong, Huizhi Yang, Zheng Chai, Zhe Chen, Yuchao Zheng, Qiwei Chen, Feng Zhang, Xun Zhou, Peng Xu, Xiao Yang, Di Wu, and Zuotao Liu. RankMixer: Scaling up ranking models in industrial recommenders. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*, pp. 6309–6316, 2025. doi: 10.1145/3746252.3761507. URL https://doi.org/10.1145/3746252.3761507.

A Computational Complexity Analysis

This appendix derives the per-sample forward FLOPs of the TRAIL modeling components reported in the overall-performance table of the main paper. We count one multiplication and one addition as two FLOPs, i.e., a matrix product $[m\times k]\cdot[k\times n]$ costs $2mkn$ FLOPs. Elementwise nonlinearities, normalization, masking, pooling, and reshaping are lower-order terms and are omitted from the main count. The reported cost excludes the input embedding layers and final prediction tower. TRAIL contains four modeling modules: State-Compressed Gated DeltaNet (SC-GDN), Segmented Target Attention (STA), Global Stacked Target Attention (GSTA), and Multi-resolution Interest Fusion (MIF).

Unless otherwise specified, we follow the notation in the main text: L denotes the original behavior length, w the STA segment size, $L' = L/w$ the compressed segment length, d the hidden dimension, d_q and d_k the STA query and key projection dimensions with $d_q = d_k$, d'_k and d'_v the compressed SC-GDN key and value dimensions, L_{stack} the number of GSTA layers, and ρ the common FFN expansion ratio. For SC-GDN, k_{conv} denotes the local convolution size. For MIF, r denotes the number of recent segments, T the number of RankMixer tokens, and R the number of RankMixer layers. In our default setting, $L = 1000$, $w = 10$, $L' = 100$, $d = 128$, $d'_k = 16$, $d'_v = 64$, $k_{\text{conv}} = 3$, $d_q = 16$, $L_{\text{stack}} = 2$, $\rho = 4$, $r = 5$, $T = 8$, and $R = 2$.

A.1 State-Compressed Gated DeltaNet (SC-GDN)

SC-GDN is applied before sequence compression and operates on the original length-L behavior sequence. It contains low-dimensional projections, local short convolution, the recurrent GDN2 state update, output gating, and output projection.

Input projections. For each behavior token $\mathbf{h}_n \in \mathbb{R}^d$, SC-GDN projects query and key into compact key spaces and value into a compact value space:

$$\mathbf{q}'_n, \mathbf{k}'_n \in \mathbb{R}^{d'_k}, \qquad \mathbf{v}'_n \in \mathbb{R}^{d'_v}. \tag{16}$$

The query, key, and value projections cost:

$$C^{\text{SC}}_{qkv} = 2L\left(2dd'_k + dd'_v\right). \tag{17}$$

Erase and write gates. SC-GDN computes an erase gate $\boldsymbol{\beta}'_n \in \mathbb{R}^{d'_k}$ and a write gate $\mathbf{w}'_n \in \mathbb{R}^{d'_v}$. Their linear projections cost:

$$C^{\text{SC}}_{\beta} + C^{\text{SC}}_{w} = 2L\left(dd'_k + dd'_v\right). \tag{18}$$

Decay branch. The channel-wise decay variable $\boldsymbol{\alpha}'_n \in \mathbb{R}^{d'_k}$ uses a two-layer projection $d \to d'_v \to d'_k$, giving:

$$C^{\text{SC}}_{\alpha} = 2L\left(dd'_v + d'_v d'_k\right). \tag{19}$$

Short convolution. The local convolution is a depthwise causal convolution over the projected query, key, and value sequences. With kernel size k_{conv}, its cost is:

$$C^{\text{SC}}_{\text{conv}} = 2Lk_{\text{conv}}(2d'_k + d'_v). \tag{20}$$

Recurrent state update. We count the SC-GDN recurrence using the original token-by-token recurrent update. The recurrent state is

$$\mathbf{S}_n \in \mathbb{R}^{d'_k \times d'_v}. \tag{21}$$

Following GDN2, the update is

$$\bar{\mathbf{S}}_n = \mathrm{Diag}(\boldsymbol{\alpha}'_n)\mathbf{S}_{n-1}, \tag{22}$$

$$\mathbf{r}_n = \bar{\mathbf{S}}_n^\top(\boldsymbol{\beta}'_n \odot \mathbf{k}'_n), \tag{23}$$

$$\mathbf{S}_n = \bar{\mathbf{S}}_n + \mathbf{k}'_n(\mathbf{w}'_n \odot \mathbf{v}'_n - \mathbf{r}_n)^\top, \tag{24}$$

$$\mathbf{o}_n = \mathbf{S}_n^\top \mathbf{q}'_n. \tag{25}$$

Per token, the diagonal state scaling, the computation of $\mathbf{r}_n$, the outer-product state update, and the output computation cost $d'_k d'_v$, $2d'_k d'_v$, $2d'_k d'_v$, and $2d'_k d'_v$ FLOPs, respectively. Therefore,

$$C^{\mathrm{SC}}_{\mathrm{rec}} = 7Ld'_k d'_v. \tag{26}$$

Vector products such as $\boldsymbol{\beta}'_n \odot \mathbf{k}'_n$ and $\mathbf{w}'_n \odot \mathbf{v}'_n$ are lower-order terms.

Output gate and output projection. The output gate uses another projection $d \to d'_v \to d'_v$, and the output projection maps d'_v back to d. Thus,

$$C^{\mathrm{SC}}_{\mathrm{gate}} = 2L\left(dd'_v + d'_v d'_v\right), \tag{27}$$

$$C^{\mathrm{SC}}_{o} = 2Ld'_v d. \tag{28}$$

Total SC-GDN cost. Combining the above terms,

$$\begin{aligned} C_{\text{SC-GDN}} = {} & C^{\mathrm{SC}}_{qkv} + C^{\mathrm{SC}}_{\beta} + C^{\mathrm{SC}}_{w} + C^{\mathrm{SC}}_{\alpha} + C^{\mathrm{SC}}_{\mathrm{conv}} \\ & + C^{\mathrm{SC}}_{\mathrm{rec}} + C^{\mathrm{SC}}_{\mathrm{gate}} + C^{\mathrm{SC}}_{o}. \end{aligned} \tag{29}$$

Substituting $L = 1000$, $d = 128$, $d'_k = 16$, $d'_v = 64$, and $k_{\mathrm{conv}} = 3$ gives

$$C_{\text{SC-GDN}} \approx 1.12 \times 10^8 \text{ FLOPs}. \tag{30}$$

A.2 Segmented Target Attention (STA)

STA compresses the SC-GDN-enhanced sequence from length L to $L' = L/w$ target-aware segment representations.

Target query projection. The target query is projected from d to the total low-dimensional space d_q:

$$C^{\mathrm{STA}}_{q} = 2dd_q. \tag{31}$$

History key and value projections. The key projection maps each historical token to the low-dimensional space d_k, while the value projection maps each token to d:

$$C^{\mathrm{STA}}_{k} = 2Ldd_k, \tag{32}$$

$$C^{\mathrm{STA}}_{v} = 2Ld^2. \tag{33}$$

Target-aware scoring. STA computes one target query against all L historical keys:

$$C^{\mathrm{STA}}_{\mathrm{score}} = 2Ld_k. \tag{34}$$

Segment aggregation. Within each segment, STA aggregates values using target-aware weights. Across all segments, this costs

$$C^{\mathrm{STA}}_{\mathrm{agg}} = 2Ld. \tag{35}$$

Output projection. The compressed value representation is projected back to dimension d:

$$C^{\mathrm{STA}}_{o} = 2L'd^2. \tag{36}$$

Feed-forward network. STA uses a two-layer FFN with expansion ratio ρ on the compressed sequence:

$$C_{\text{ffn}}^{\text{STA}} = 2L' \left(d \cdot \rho d + \rho d \cdot d\right) = 4\rho L' d^2. \tag{37}$$

Total STA cost. The STA cost is

$$\begin{aligned} C_{\text{STA}} = C_q^{\text{STA}} + C_k^{\text{STA}} + C_v^{\text{STA}} + C_{\text{score}}^{\text{STA}} \\ + C_{\text{agg}}^{\text{STA}} + C_o^{\text{STA}} + C_{\text{ffn}}^{\text{STA}}. \end{aligned} \tag{38}$$

With $L = 1000$, $L' = 100$, $d = 128$, $d_q = 16$, and $\rho = 4$,

$$C_{\text{STA}} \approx 6.66 \times 10^7 \text{ FLOPs}. \tag{39}$$

A.3 Global Stacked Target Attention (GSTA)

GSTA performs stacked target attention over the compressed length-L' sequence. Each layer uses one target query and attends to all compressed segment representations.

Per-layer projections. For one GSTA layer, the target query projection costs

$$C_q^{\text{GSTA}} = 2d^2. \tag{40}$$

The key and value projections over the compressed sequence cost

$$C_{kv}^{\text{GSTA}} = 4L' d^2. \tag{41}$$

The output projection costs

$$C_o^{\text{GSTA}} = 2d^2. \tag{42}$$

Per-layer attention. The target query attends to L' compressed tokens. Score computation and value aggregation cost

$$C_{\text{attn}}^{\text{GSTA}} = 4L' d. \tag{43}$$

Per-layer FFN. The FFN has expansion ratio ρ and is applied to the target representation:

$$C_{\text{ffn}}^{\text{GSTA}} = 2(d \cdot \rho d + \rho d \cdot d) = 4\rho d^2. \tag{44}$$

Total GSTA cost. The per-layer cost is

$$C_{\text{layer}}^{\text{GSTA}} = (4L' + 4 + 4\rho)d^2 + 4L' d. \tag{45}$$

With L_{stack} layers,

$$C_{\text{GSTA}} = L_{\text{stack}} \left((4L' + 4 + 4\rho)d^2 + 4L' d\right). \tag{46}$$

Substituting $L_{\text{stack}} = 2$, $L' = 100$, $d = 128$, and $\rho = 4$ gives

$$C_{\text{GSTA}} \approx 1.39 \times 10^7 \text{ FLOPs}. \tag{47}$$

A.4 Multi-resolution Interest Fusion (MIF)

MIF concatenates the target-conditioned global interest, average interest, salient interest, r recent segment representations, and user-context features. The resulting fusion input has dimension $(4 + r)d$. Projecting it into T RankMixer tokens of dimension d costs

$$C_{\text{proj}}^{\text{MIF}} = 2(4 + r)Td^2. \tag{48}$$

In the Dense RankMixer implementation, token mixing is performed by reshaping and transposing the representation. The dominant cost comes from the two-layer per-token FFN. With R layers and expansion ratio ρ, its cost is

$$C_{\text{RM}}^{\text{MIF}} = 4R\rho Td^2. \tag{49}$$

Mean and max pooling over the L' segments require $O(L'd)$ lower-order operations and are omitted from the concrete FLOPs count under our convention. Therefore,

$$C_{\text{MIF}} = C_{\text{proj}}^{\text{MIF}} + C_{\text{RM}}^{\text{MIF}}. \tag{50}$$

Substituting $r = 5$, $T = 8$, $d = 128$, $R = 2$, and $\rho = 4$ gives

$$C_{\text{MIF}} \approx 6.55 \times 10^6 \text{ FLOPs}. \tag{51}$$

A.5 TOTAL FLOPS

The per-sample FLOPs of TRAIL are the sum of SC-GDN, STA, GSTA, and MIF:

$$C_{\mathrm{TRAIL}} = C_{\text{SC-GDN}} + C_{\mathrm{STA}} + C_{\mathrm{GSTA}} + C_{\mathrm{MIF}}. \tag{52}$$

Under the default setting,

$$\begin{aligned} C_{\mathrm{TRAIL}} &\approx 1.12 \times 10^8 + 6.66 \times 10^7 \\ &\quad + 1.39 \times 10^7 + 6.55 \times 10^6 \\ &\approx 1.99 \times 10^8 \ \mathrm{FLOPs}. \end{aligned} \tag{53}$$

Thus, TRAIL requires approximately 199M FLOPs per sample under the reported configuration, excluding the input embedding layers and final prediction tower. Because $L' = L/w$ and all projection dimensions, kernel sizes, segment size, GSTA depth, and MIF dimensions are fixed constants in the default setting, the total cost scales linearly with the original sequence length L.