OneTrans: Unified Feature Interaction and Sequence Modeling with One Transformer in Industrial Recommender (ByteDance, 2025)

Challenges

- **Fragmented scaling: sequence modeling vs. feature interaction.** Prior industrial ranking stacks typically follow an "encode-then-interaction pipeline" where user behavior sequences are compressed first and cross features are modeled later, which "hinders bidirectional information exchange" and prevents unified scaling/optimization.
- **Latency and execution fragmentation.** Splitting modules limits reuse of LLM-style serving optimizations (e.g., KV caching, memory-efficient attention, mixed precision), increasing end-to-end latency.
- **Heterogeneous feature types.** Ranking inputs mix homogeneous sequential behavior events with heterogeneous non-sequential (user/item/context) features, creating stability and parameter-allocation challenges.

Key Initiatives (Core Contributions)

- **Unified Transformer backbone (OneTrans).** A single Transformer-style stack that *jointly* performs user behavior sequence modeling and non-sequential feature interaction.
- **Unified tokenizer.** Converts sequential features (S) and non-sequential features (NS) into one token sequence, inserting learnable `[SEP]` tokens between behavior sequences.
- **Mixed parameterization.** Sequential tokens share one set of QKV/FFN weights while each non-sequential token receives token-specific QKV/FFN, bridging RecSys heterogeneity with Transformer computation.
- **Efficiency mechanisms: pyramid + cross-request KV caching.** Pyramid stacking progressively prunes sequential query tokens; cross-request KV caching amortizes user-side computation across candidates and across requests.
- **Scaling + online impact.** Industrial-scale results show efficient scaling and statistically significant online business lifts, including a headline "5.68% lift in per-user GMV".

Methodology

System Architecture (Figure 2)

Problem setup The paper focuses on the industrial **ranking stage** in a cascaded recommender. The model predicts CTR/CVR for each user–candidate pair based on (i) multi-behavior sequential histories and (ii) non-sequential user/item/context features.

OneTrans block A pre-norm causal Transformer block with RMSNorm, Mixed Causal Attention, and Mixed FFN. A unified causal mask enables NS-tokens to attend over the full S-token history while maintaining efficient Transformer-style execution.

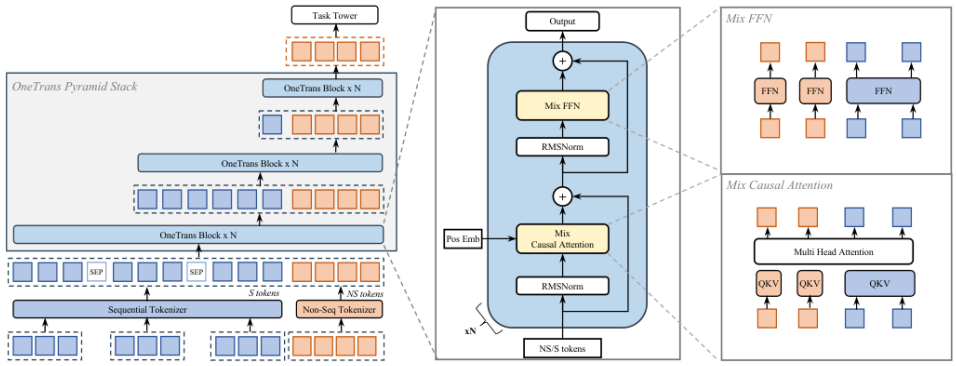
Experiments (key numbers)

- **Dataset scale (Table 1).** 29.1B impressions, 27.9M users, 10.2M items.
- **Offline effectiveness vs. strong baselines (Table 2).** Compared to a production baseline DCNv2+DIN, OneTransL (default) reports improvements of **+1.53%/+2.79%** CTR AUC/UAUC and **+1.14%/+3.23%** CVR AUC/UAUC (relative).

- **Systems efficiency ablations (Table 4).** Adding pyramid stack and KV caching yields large training/runtime and inference/latency reductions; mixed precision with recomputation reports a **-69.1%** p99 latency change (relative) vs. the unoptimized OneTransS pipeline.
- **Online A/B tests (Table 6).** In Feeds: **+5.685% GMV/u** (and +7.737% click/u). In Mall: **+3.670% GMV/u**. The abstract highlights a **5.68% lift in per-user GMV** online.

References (selected)

- Chai et al. (2025). LONGER: Scaling Up Long Sequence Modeling in Industrial Recommenders.
- Wang et al. (2021). DCNv2: Improved Deep & Cross Network.
- Zhou et al. (2018). DIN: Deep Interest Network.
- Kang & McAuley (2018). SASRec.
- Dao et al. (2023). FlashAttention-2.



(a) OneTrans Framework (b) OneTrans Block (c) Mix Parameterization

Figure 2: System Architecture. (a) OneTrans overview. Sequential (S, blue) and non-sequential (NS, orange) features are tokenized separately. After inserting [SEP] between user behavior sequences, the unified token sequence is fed into stacked *OneTrans Pyramid Blocks* that progressively shrink the token length until it matches the number of NS tokens. (b) OneTrans Block: a causal pre-norm Transformer Block with RMSNorm, *Mixed Causal Attention* and *Mixed FFN*. (c) "Mixed" = mixed parameterization: S tokens share one set of QKV/FFN weights, while each NS token receives its own token-specific QKV/FFN.

interaction paradigm, which pushes interactions to a separate stage and blocks unified optimization with user sequence modeling [30].

To date, progress in RecSys has largely advanced along two independent tracks: sequence modeling and feature interaction. InterFormer [30] attempts to bridge this gap through a summary-based bidirectional cross architecture that enables mutual signal exchange between the two components. However, it still maintains them as separate modules, and the cross architecture introduces both architectural complexity and fragmented execution. Without a unified backbone for joint modeling and optimization, scaling the system as an integrated whole remains challenging.

Recent work on Generative Recommenders (GRs) frames recommendation as sequential transduction and proposes efficient long-context backbones such as HSTU [31]. This line is complementary to DLRMs that rely on rich non-sequential (NS) features.

3 Methodology

Before detailing our method, we briefly describe the task setting. In a cascaded industrial RecSys, each time the recall stage returns a candidate set (typically hundreds of candidate items) for a user u. The ranking model then predicts a score to each candidate item i:

$$\hat{y}_{u,i} = f\left(i \,\middle|\, \mathcal{NS}, \mathcal{S}; \Theta\right) \tag{1}$$

where $\mathcal{NS}$ is a set of non-sequential features derived from the user, the candidate item, and the context; $\mathcal{S}$ is a set of historical behavior sequences from the user; and Θ are trainable parameters. Common task predictions include the click-through rate (CTR) and the post-click conversion rate (CVR).

$$\begin{aligned} \mathrm{CTR}_{u,i} &= P\left(\mathrm{click} = 1 \,\middle|\, \mathcal{NS}, \mathcal{S}; \Theta\right), \\ \mathrm{CVR}_{u,i} &= P\left(\mathrm{conv} = 1 \,\middle|\, \mathrm{click} = 1, \mathcal{NS}, \mathcal{S}; \Theta\right). \end{aligned} \tag{2}$$

3.1 OneTrans Framework Overview

As illustrated in Fig. 2(a), OneTrans employs a *unified tokenizer* that maps sequential features $\mathcal{S}$ to S-tokens, and non-sequential features $\mathcal{NS}$ to NS-tokens. A *pyramid-stacked Transformer* then consumes the unified token sequence jointly within a single computation graph. We denote the initial token sequence as

$$\mathbf{X}^{(0)} = \left[\text{S-tokens}\,;\,\text{NS-tokens}\right] \in \mathbb{R}^{(L_{\mathrm{S}}+L_{\mathrm{NS}})\times d}. \tag{3}$$

This token sequence is constructed by concatenating L_{S} number of S-tokens and L_{NS} number of NS-tokens, with all tokens having dimensionality d. Note that, the S-tokens contain learnable [SEP] tokens inserted to delimit boundaries between different kind of user-behavior sequences. As shown in Fig. 2(b), each OneTrans block progressively refines the token states through:

$$\mathbf{Z}^{(n)} = \mathrm{MixedMHA}\Big(\mathrm{Norm}\big(\mathbf{X}^{(n-1)}\big)\Big) + \mathbf{X}^{(n-1)}, \tag{4}$$

$$\mathbf{X}^{(n)} = \mathrm{MixedFFN}\Big(\mathrm{Norm}\big(\mathbf{Z}^{(n)}\big)\Big) + \mathbf{Z}^{(n)}. \tag{5}$$

Here, MixedMHA (Mixed Multi-Head Attention) and MixedFFN (Mixed Feed-Forward Network) adopt a mixed parameterization strategy (see Fig. 2(c)) sharing weights across sequential tokens, while assigning separate parameters to non-sequential tokens in both the attention and feed-forward layers.

A unified causal mask enforces autoregressive constraints, restricting each position to attend only to preceding tokens. Specifically, NS-tokens are permitted to attend over the entire history of S-tokens, thereby enabling comprehensive cross-token interaction. By stacking such blocks with pyramid-style tail truncation applied to S-tokens, the model progressively distills compact high-order

Figure 1: System architecture (from OneTrans paper, Figure 2): unified tokenizer + pyramid stack + mixed parameterization.