ULTRA-HSTU: Bending the Scaling Law Curve in Large-Scale Recommendation Systems (Meta, 2026)

Challenges

- **Scaling inefficiency of prior long-sequence recommenders.** The paper targets the gap between model quality gains and the rapid growth of training/inference FLOPs in long-sequence settings. It emphasizes that scaling must improve both *quality* and *efficiency* for industrial deployment.
- **Ultra-long user histories.** Industrial recommenders process sequences with thousands to tens of thousands of events; the paper evaluates sequences of length 3,072 to 16,384 and highlights the need to preserve signal without exploding compute.
- **Model–system co-design.** Achieving practical speedups requires architectural changes plus system-level optimizations (e.g., attention kernels, precision formats), not just algorithmic tweaks.

Key Initiatives (Core Contributions)

- **ULTRA-HSTU architecture and scaling design.** The paper proposes ULTRA-HSTU to improve scaling efficiency, reporting large gains in training and inference scaling slopes vs vanilla HSTU.
- **Input-sequence optimization + semi-local attention.** It introduces sequence design and attention patterns that significantly reduce compute while maintaining accuracy.
- **Model–system co-design.** The paper integrates mixed-precision/FP8 pipelines and optimized attention kernels to realize end-to-end efficiency gains.

Methodology

Model Overview (Figure 2) Figure 1 shows the model-design overview from the paper, covering the general recommender design and the input-sequence optimization pipeline used by ULTRA-HSTU.

Training Setup Industrial evaluation uses a large-scale internal dataset with **over 6 billion samples**. Sequences range from **3,072 to 16,384 events**. The paper uses a chronological split: **85% training, 15% evaluation**. The metric is NE (lower is better), with C-NE and E-NE for consumption/engagement tasks.

Experiments (with key numbers)

Industrial Dataset Benchmark (Table 1)

- ULTRA-HSTU is the baseline (0% / 0% by definition). Competing models show positive deltas (worse):
- **HSTU:** ΔC-NE = **+0.43%**, ΔE-NE = **+0.04%**
- **STCA:** ΔC-NE = **+0.94%**, ΔE-NE = **+0.74%**
- **Transformer:** ΔC-NE = **+0.57%**, ΔE-NE = **+0.59%**
- **DIN:** ΔC-NE = **+1.41%**, ΔE-NE = **+1.91%**
- **SASRec:** ΔC-NE = **+1.12%**, ΔE-NE = **+1.28%**

Scaling on Industrial Dataset (Table 2) ULTRA-HSTU improves C-NE while lowering FLOPs. Example entries:

- **ULTRA-HSTU @ 8,192 length, 18 layers:** ΔC-NE **-0.58%** with **Training TFLOP 0.414** ($\downarrow$43.7%) and **Inference TFLOP 0.337** ($\downarrow$87.8%).
- **ULTRA-HSTU @ 16,384 length, 18 layers:** ΔC-NE **-0.78%** with **Training TFLOP 0.639** ($\downarrow$59.7%) and **Inference TFLOP 0.436** ($\downarrow$90.7%).

Open-source Benchmark (Table 3) On KuaiRand, ULTRA-HSTU achieves the best NE with lower compute:

- **ULTRA-HSTU:** Training TFLOP **504.41**, Inference TFLOP **166.41**, NE **0.8626** (best among baselines shown).
- Compared to HSTU (NE 0.8676) and Transformer (NE 0.8688), ULTRA-HSTU is both more accurate and more efficient.

References (selected)

- Zhai et al. (2024). HSTU.
- Guan et al. (2025). STCA.
- Zhou et al. (2018). DIN.
- Kang & McAuley (2018). SASRec.

References (Big-company, as cited in the paper)

- **Meta**: Zhai et al. (2024). HSTU.
- **ByteDance (Douyin)**: Guan et al. (2025). End-to-end 10k-sequence modeling at billion scale on Douyin.
- **Meituan**: Han et al. (2025). MTGR: Industrial-scale generative recommendation framework in Meituan.
- **Kuaishou**: Si et al. (2024). Twin v2: Scaling ultra-long user behavior sequence modeling at Kuaishou.
- **Alibaba**: Chen et al. (2019). Behavior sequence transformer for e-commerce recommendation in Alibaba.
- **Not cited in this paper**: Google, Microsoft, Pinterest.

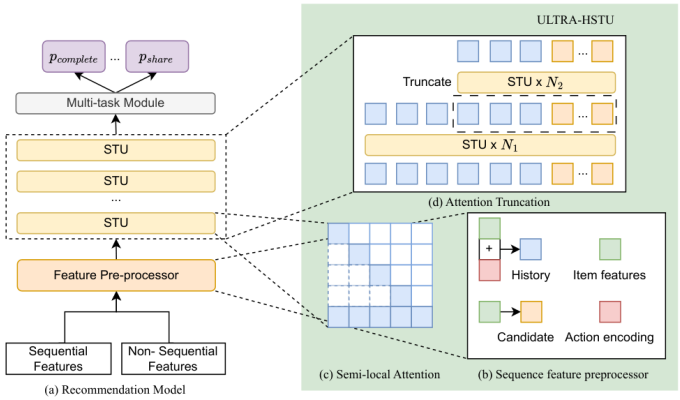

Figure 2 Model design overview: (a) General recommendation model design. (b) Input sequence optimizations with action-aware designs (c) Semi-local attention mask with linear complexity (d) Attention truncation for dynamic topological designs.

3 Background

As illustrated in Figure 2 (a), a typical recommender system takes input features and trains on multi-task classification problems. Formally, it learns a multi-task model $\mathcal{M}$ to output a probability $\hat{y}_k = \mathcal{M}(X, x_j) \in [0, 1]$ for a candidate x_j across different prediction tasks y_k (e.g., like, video completion, comment), and rank the candidates based on predicted scores. Here X is the input features of user. We optimize the model by minimizing the cross-entropy loss between predictions $\hat{y}_k$ and ground truth labels y_k collected from logs. Throughout the paper, we use L to denote the general sequence length, X to denote the input features. Most generative ranking paradigm models the input X as a sequence of embeddings (see context below), and leverage attention layers to learn probabilities from those sequential embeddings.

Input As shown in Figure 2 (a), a recommender uses a feature preprocessor to transfer different input features into a sequence of embeddings, including: **user interaction history (UIH) sequence** records a sequence of items a specific user interacted with and the corresponding actions (e.g., like, comment, video completion, etc.) and context (e.g., timestamp). Raw item IDs (and its multi-modal representation), action types are represented by d-dimensional learnable embeddings via embedding table lookups. For user i, we denote its UIH as $X_i = \{I_i, A_i\}$, where the item embeddings in UIH are $I_i = \{I_{i,j}\}_{j=1}^{L_i} \in R^{L_i,d}$, and action embeddings are $A_i = \{a_{i,j}\}_{j=1}^{L_i} \in R^{L_i,d}$ and L_i is the total length of user i's UIH. **Non-sequential features** include user-side features, such as country, user language, etc., and item-side features, such as sparse (e.g., raw ID of item) and dense (e.g., click-through rate for this item) features. Those user-side features could be summarized into context-embeddings and put at the beginning of sequential UIH (Zhai et al., 2024). Those item-side features could be summarized as item-embeddings and inserted into sequences as target-side embeddings (Zhang et al., 2025).

Model Given a sequence of embeddings, modern recommender leverages transformer-style models. One typical arch is Hierarchical Sequential Transduction Units (HSTU) (Zhai et al., 2024), which shows significant wins on top of vanilla transformers in recommender systems by the following modifications:

$$\text{normaliation:} \quad X = \text{Norm}(Z) \tag{1}$$
$$\text{pre-attention:} \quad U, Q, K, V = \phi_1(f_1(X)) \tag{2}$$
$$\text{attention:} \quad A = \left(\phi_2(QK^T) \odot M\right)V \tag{3}$$
$$\text{post-attention:} \quad Y = f_2\left(\text{Norm}\left(A\right) \odot U\right) \tag{4}$$
$$\text{residual connection:} \quad Z = Y + Z \tag{5}$$

Figure 1: Model design overview (from ULTRA-HSTU paper, Figure 2).