Actions Speak Louder than Words: Trillion-Parameter Sequential Transducers for Generative Recommendations (2024)

Challenges

- **DLRMs do not scale with compute.** The abstract notes that despite huge data and features, "most Deep Learning Recommendation Models (DLRMs) in industry fail to scale with compute." This motivates a new formulation.
- **High-cardinality, heterogeneous features at industrial scale.** The paper targets systems that must handle "tens of billions of user actions" with highly heterogeneous features.
- **Need a unified formulation for ranking and retrieval.** The authors argue for reformulating recommendation as a sequential transduction problem within a generative modeling framework (GRs) to enable scaling laws similar to LLMs.

Key Initiatives

- **Generative Recommenders (GRs).** The work "reformulate[s] recommendation problems as sequential transduction tasks" under a generative modeling framework.
- **HSTU encoder.** Introduces a new encoder design, *Hierarchical Sequential Transduction Unit (HSTU)*, a stack of identical layers that replaces heterogeneous DLRM modules with a single modular block.
- **Efficiency + scaling.** HSTU is reported as **5.3x–15.2x faster** than state-of-the-art Transformers on length-8192 sequences, enabling training/inference at scale.
- **Production impact.** The paper states that GRs/HSTU "have led to **12.4%** metric improvements in production" and allow models "285x more complex" with less inference compute.
- **M-FALCON inference algorithm.** New inference algorithm to reduce target-aware cross-attention cost and improve serving throughput (Figure 11/12).

Methodology

- **Sequential transduction formulation.** Ranking/retrieval are cast as sequence-to-sequence prediction of interleaved item/action tokens.
- **HSTU layer design.** Each layer contains: (1) Pointwise Projection, (2) Spatial Aggregation, (3) Pointwise Transformation.
- **Core equations.** The paper defines:
  - $U(X), V(X), Q(X), K(X) = \text{Split}(\phi_1(f_1(X)))$ (Eq. 1)
  - $A(X)V(X) = \phi_2(Q(X)K(X)^T + rab_{p,t})V(X)$ (Eq. 2)
  - $Y(X) = f_2(\text{Norm}(A(X)V(X)) \odot U(X))$ (Eq. 3)
- **Pointwise aggregated attention.** Replaces softmax normalization with a pointwise aggregated mechanism to better capture engagement intensity and handle non-stationary vocabularies.
- **Model figure.** The key model comparison is shown in **Figure 3** (DLRMs vs GRs, simplified HSTU block).

Experiments

- **Public dataset gains (Table 4).** HSTU and HSTU-large improve HR@10 / NDCG@10 across ML-1M, ML-20M, Books. Example results from Table 4:
  - **ML-1M:** HSTU HR@10 **0.3097** (+8.6%), NDCG@10 **0.1720** (+7.3%); HSTU-large HR@10 **0.3294** (+15.5%), NDCG@10 **0.1893** (+18.1%).
  - **ML-20M:** HSTU HR@10 **0.3252** (+11.9%), NDCG@10 **0.1878** (+15.9%); HSTU-large HR@10 **0.3567** (+22.8%), NDCG@10 **0.2106** (+30.0%).
  - **Books:** HSTU HR@10 **0.0404** (+38.4%), NDCG@10 **0.0219** (+40.6%); HSTU-large HR@10 **0.0469** (+60.6%), NDCG@10 **0.0257** (+65.8%).
- **Ablations (Table 5).** Comparing HSTU variants (with/without $rab_{p,t}$, softmax) shows clear quality drops in ablations, supporting the design choices.
- **Efficiency/scaling.** HSTU is up to **15.2x** more efficient and supports longer sequences (Figure 5/6/7), enabling scaling behavior similar to LLMs.
- **Production impact.** The paper claims GRs/HSTU deliver **12.4%** production metric improvements.

References (selected)

- Mudigere et al. (2022). DLRM at scale.
- Raffel et al. (2020). T5 / relative attention bias.
- Dao (2023). FlashAttention-2.
- Kang & McAuley (2018). SASRec.
- Zhou et al. (2018). DIN.

tion models over long sequences in a scalable manner, we move from traditional impression-level training to *generative training*, reducing the computational complexity by an $O(N)$ factor, as shown at the top of Figure 2. By doing so, encoder costs are amortized across multiple targets. More specifically, when we sample the i-th user at rate $s_u(n_i)$, the total training cost now scales as $\sum_i s_u(n_i) n_i (n_i^2 d + n_i d^2)$, which is reduced to $O(N^2 d + N d^2)$ by setting $s_u(n_i)$ to $1/n_i$. One way to implement this sampling in industrial-scale systems is to emit training examples at the end of a user's request or session, resulting in $\hat{s_u}(n_i) \propto 1/n_i$.

3. A High Performance Self-Attention Encoder for Generative Recommendations

To scale up GRs for industrial-scale recommendation systems with large, non-stationary vocabularies, we next introduce a new encoder design, *Hierarchical Sequential Transduction Unit* (HSTU). HSTU consists of a stack of identical layers connected by residual connections (He et al., 2015). Each layer contains three sub-layers: Pointwise Projection (Equation 1), Spatial Aggregation (Equation 2), and Pointwise Transformation (Equation 3):

$$U(X), V(X), Q(X), K(X) = \text{Split}(\phi_1(f_1(X))) \quad (1)$$

$$A(X)V(X) = \phi_2\left(Q(X)K(X)^T + \text{rab}^{p,t}\right)V(X) \quad (2)$$

$$Y(X) = f_2\left(\text{Norm}\left(A(X)V(X)\right) \odot U(X)\right) \quad (3)$$

where $f_i(X)$ denotes an MLP; we use one linear layer, $f_i(X) = W_i(X) + b_i$ for f_1 and f_2 to reduce compute complexity and further batches computations for queries $Q(X)$, keys $K(X)$, values $V(X)$, and gating weights $U(X)$ with a fused kernel; ϕ_1 and ϕ_2 denote nonlinearity, for both of which we use SiLU (Elfwing et al., 2017); Norm is layer norm; and $\text{rab}^{p,t}$ denotes relative attention bias (Raffel et al., 2020) that incorporates positional (p) and temporal (t) information. Full notations used can be found in Table 9.

HSTU encoder design allows for the replacement of heterogeneous modules in DLRMs with a single modular block. We observe that there are, effectively, three main stages in DLRMs: *Feature Extraction*, *Feature Interactions*, and *Transformations of Representations*. *Feature Extractions* retrieves the pooled embedding representations of categorical features. Their most advanced versions can be generalized as pairwise attention and target-aware pooling (Zhou et al., 2018), which is captured with HSTU layers.

Feature Interaction is the most critical part of DLRMs. Common approaches used include factorization machines and their neural network variants (Rendle, 2010; Guo et al., 2017; Xiao et al., 2017), higher order feature interactions (Wang et al., 2021), etc. HSTU replaces feature interactions by enabling attention pooled features to directly "interact" with other features via $\text{Norm}\left(A(X)V(X)\right) \odot U(X)$.

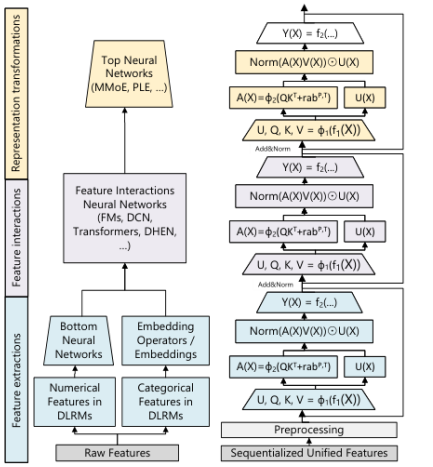

Figure 3. Comparison of key model components: DLRMs vs GRs. The complete DLRM setup (Mudigere et al., 2022) is shown on the left side and a simplified HSTU is shown on the right.

This design is motivated by the difficulty of approximating dot products with learned MLPs (Rendle et al., 2020; Zhai et al., 2023a). Given SiLU is applied to $U(X)$, $\text{Norm}\left(A(X)V(X)\right) \odot U(X)$ can also be interpreted as a variant of SwiGLU (Shazeer, 2020).

Transformations of Representations is commonly done with Mixture of Experts (MoEs) and routing to handle diverse, heterogeneous populations. A key idea is to perform conditional computations by specializing sub-networks for different users (Ma et al., 2018; Tang et al., 2020). Element-wise dot products in HSTU can virtually perform gating operations used in MoEs up to a normalization factor.

3.1. Pointwise aggregated attention

HSTU adopts a new pointwise aggregated (normalized) attention mechanism (in contrast, softmax attention computes normalization factor over the entire sequence). This is motivated by two factors. First, the number of prior data points related to target serves as a strong feature indicating the intensity of user preferences, which is hard to capture after softmax normalization. This is critical as we need to predict both the intensity of engagements, e.g., time spent on a given item, and the relative ordering of the items, e.g., predicting an ordering to maximize AUC. Second, while softmax activation is robust to noise by construction, it is less suited for non-stationary vocabularies in streaming settings.

The proposed pointwise aggregated attention mechanism is depicted in Equation (2). Importantly, layer norm is needed after pointwise pooling to stabilize training. One way to

Figure 1: Model comparison: DLRMs vs GRs (from HSTU paper, Figure 3).