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

Grupo 20
Paula Grune, Clemente Sánchez, Pablo Soto, Ignacio Vial

Definición y motivación del estudio

Transformers

- Es una arquitectura de red neuronal.
- Introducida por "Attention is All You Need" (2017), Vaswani et al.

- Típicamente recibe una secuencia de tokens y genera una nueva secuencia a partir de la entrada.

- Utiliza un mecanismo llamado self-attention para enfocarse en diferentes partes de la secuencia lo cual permite “capturar” relaciones (entre palabras por ejemplo) en cualquier lugar de la secuencia.

Recomendación Secuencial

- Enfoque en Sistemas Recomendadores.

- Tiene en cuenta el orden temporal además de los ítems con los que ha interactuado un usuario.
- Constituyen una cadena que representan las interacciones de un usuario.

DLRMs

- Deep Learning Recommendation Models
- Constituyen parte importante del **estado del arte** para el problema de recomendación.

- Se caracterizan por el uso de ***features* heterogéneas**:
  - ***Numerical features***: *counters*, *ratios*, *embeddings*.
  - ***Categorical features***: *user ids*, lenguaje del usuario, ciudades donde ha interactuado el usuario, etc.
- Dado que existen nuevos productos y contenido siendo añadido todo el tiempo, el espacio de *features* es **extremadamente alto**.
- Para manejar el gran espacio de *features* se emplean **distintas técnicas** como utilizar distintas redes para combinarlas o transformar estas a representaciones intermedias.

Contexto

Sistemas de recomendación a gran escala

Se caracterizan por tener una cardinalidad alta, *features* heterogéneas y manejan muchísimas acciones de usuarios diariamente *(tens of billions)*

Soluciones actuales tienen escalabilidad limitada

Deep Learning Recommendation Models (DLRMs) constituyen parte importante del estado del arte para el problema de recomendación

Éxito de *Transformers*

Como solución para abordar problemas de lenguaje y visión por computador

Motivación

Diseñar un nuevo paradigma de recomendación.

Desafíos observados

- Los autores presentan estos tres puntos al buscar alternativas capaces de escalar a grandes cantidades de usuarios.

- **Falta de estructura explícita en las *features*.**
- **Los sistemas de recomendación usan vocabularios que están en constante cambio.**
- **El costo computacional representa el principal cuello de botella para habilitar *large-scale sequential models*.**

Perspectivas Clave

- Se reformula el problema de recomendación como un problema de transducción secuencial (**Generative Recommenders**)

- Se reformulan las tareas fundamentales de ordenamiento (**ranking**) y recuperación (**retrieval**).

- El enfoque descrito permite aprovechar de manera sistemática las redundancias presentes en las características, el entrenamiento y la inferencia para mejorar la eficiencia del sistema.

Trabajos relacionados

Modelos secuenciales y aplicaciones industriales

Primeros esfuerzos para capturar **naturaleza secuencial** de las interacciones y las recomendaciones a escala industrial.

Eficiencia computacional

Se exploran formas de hacer que los modelos sean más **escalables y eficientes** en término de tiempo y memoria.

Modelos LLMs aplicados a recomendación

LLMs preentrandos pueden adaptarse a tareas de recomendación mediante aprendizaje en contexto.

Modelos secuenciales y aplicaciones industriales

- Bases conceptuales para modelar las interacciones de **usuarios como secuencias**
- Buscan mejorar la eficiencia y relevancia de las recomendaciones a **escala industrial**

- **Session-based recommendations with recurrent neural networks** (Hidasi et al., 2016): Aborda recomendaciones **basadas en secuencias**, pero carece de capacidad para manejar datos heterogéneos.
- **Deep interest network for click-through rate prediction** (Zhou et al., 2018): Aplicaciones a **escala industrial** de los enfoques secuenciales basados en la atención por pares.
- **Behavior sequence transformer for e-commerce recommendation in alibaba** (Chen et al., 2019): Precedente para integrar **Transformers** en sistemas de recomendación industriales.
- **Twin: Two-stage interest network for lifelong user behavior modeling in ctr prediction at kuaishou** (Chang et al., 2023): **Eficiencia y escalabilidad.**

Eficiencia computacional

Se resuelven desafíos computacionales permitiendo que los modelos propuestos en el paper manejen **secuencias más largas y complejas.**

- **Generating long sequences with sparse transformers** (Child et al., 2019): Introduce **sparse transformers** que reducen la complejidad computacional.
- **Transformers are rnns: Fast autoregressive transformers with linear attention** (Katharopoulos et al., 2020): Reformula los **self-attentions** para reducir complejidad
- **Self-attention does not need o(n^2) memory** (Rabe & Staats, 2021): Algoritmos que requieren **menos memoria** para procesar secuencias largas.
- **Transformer quality in linear time** (Hua et al., 2022): **Mejorar la eficiencia** sin sacrificar calidad.

Modelos LLMs aplicados a recomendación

Aunque los LLMs muestran potencial con pocos datos, sus limitaciones en escalas grandes refuerzan la necesidad de un marco generativo específicamente diseñado como el propuesto en el paper.

- **Zero-shot recommendation as language modeling** (Sileo et al., 2022): Utiliza LLMs para construir recomendaciones, utilizando un **enfoque generativo**.
- **Large language models are zero-shot rankers for recommender systems** (Hou et al., 2024): Evalúa cómo los LLMs pueden realizar **tareas de ranking** en recomendaciones.

Métodos propuestos

Recomendación como tarea de transducción secuencial

1. Se unifican las features heterogéneas en DLRMs

2. Se reformula ranking y retrieval como tarea de transducción secuencial

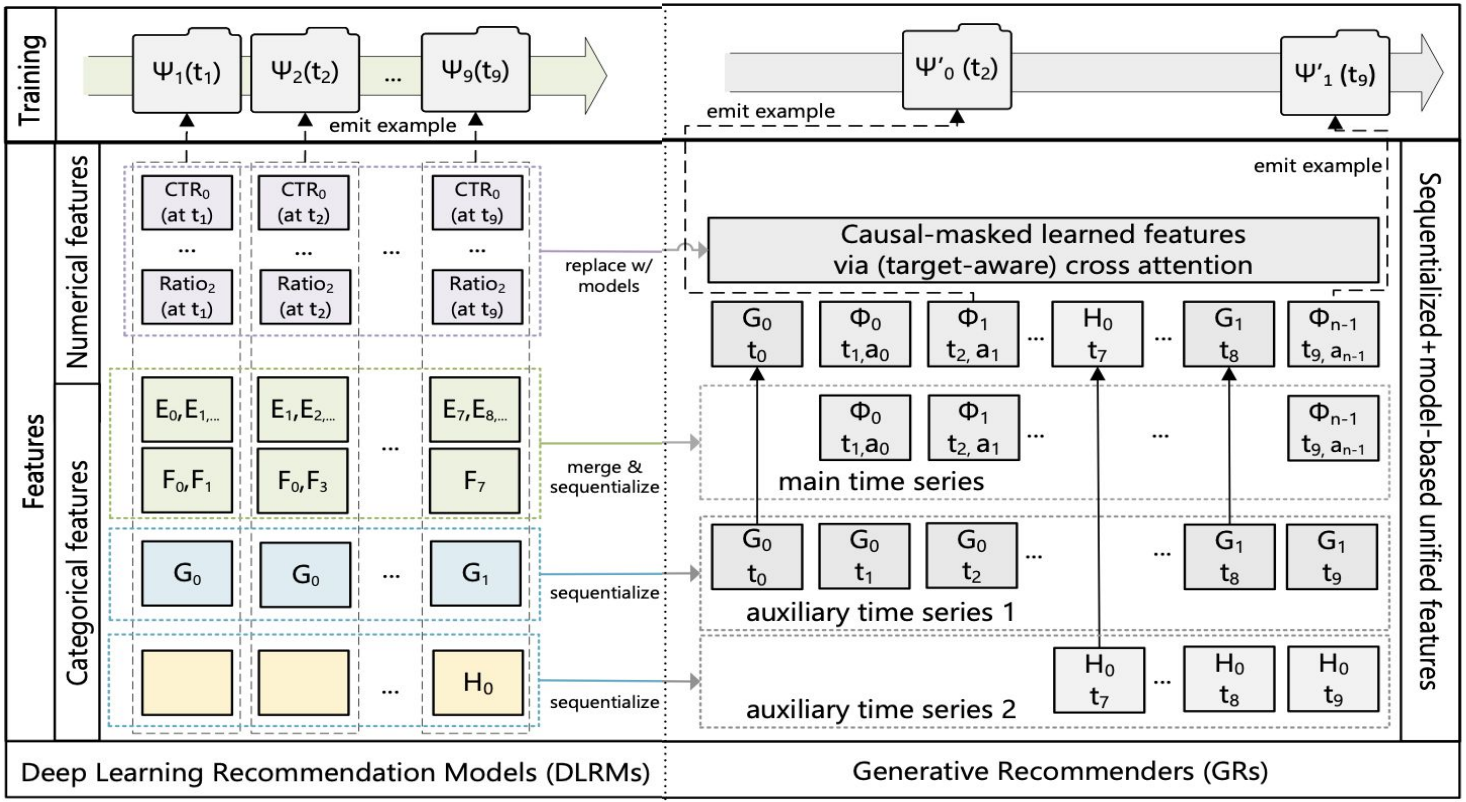
Unificar features heterogéneas en DLRMs

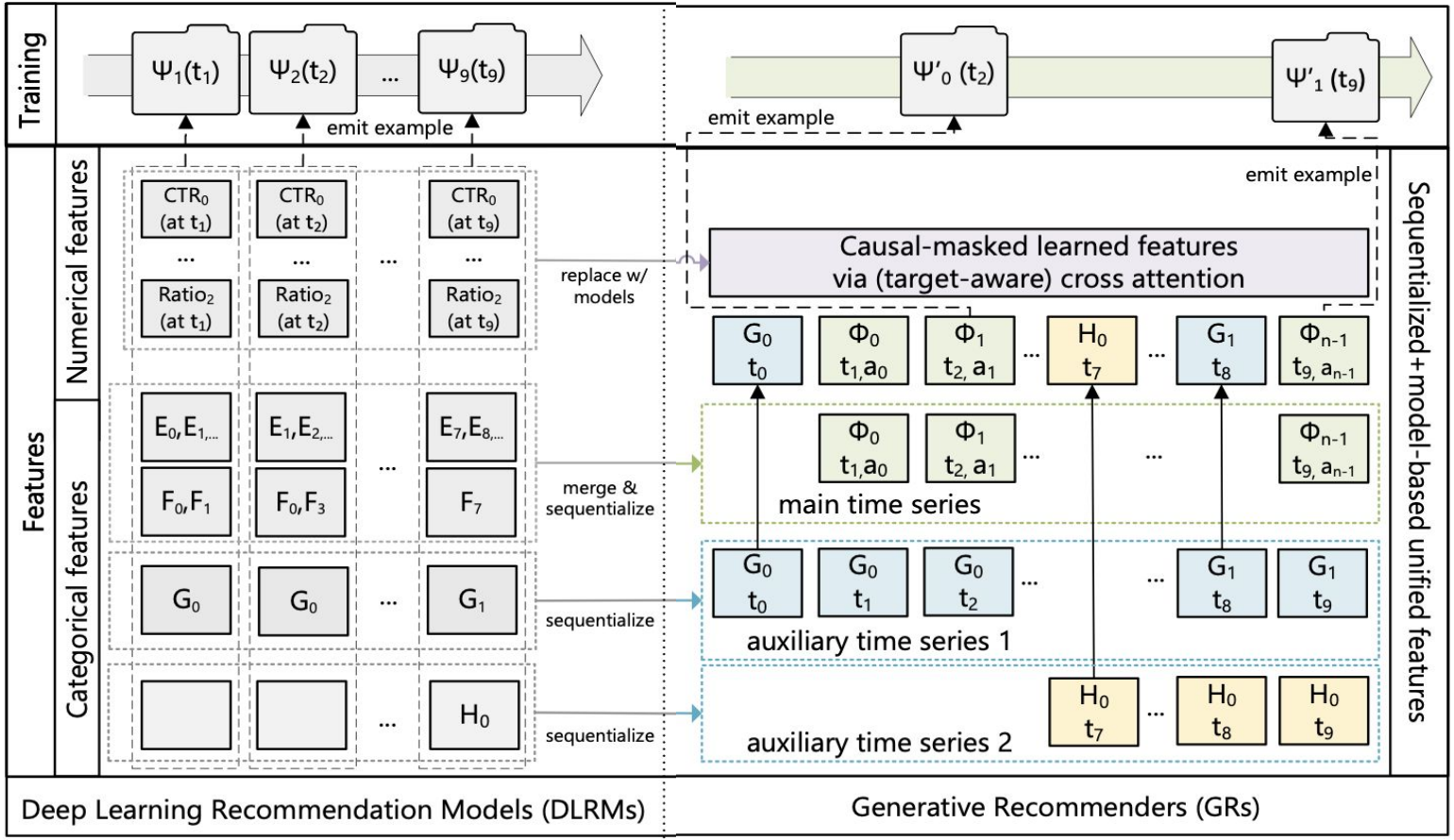
Para features categóricas:

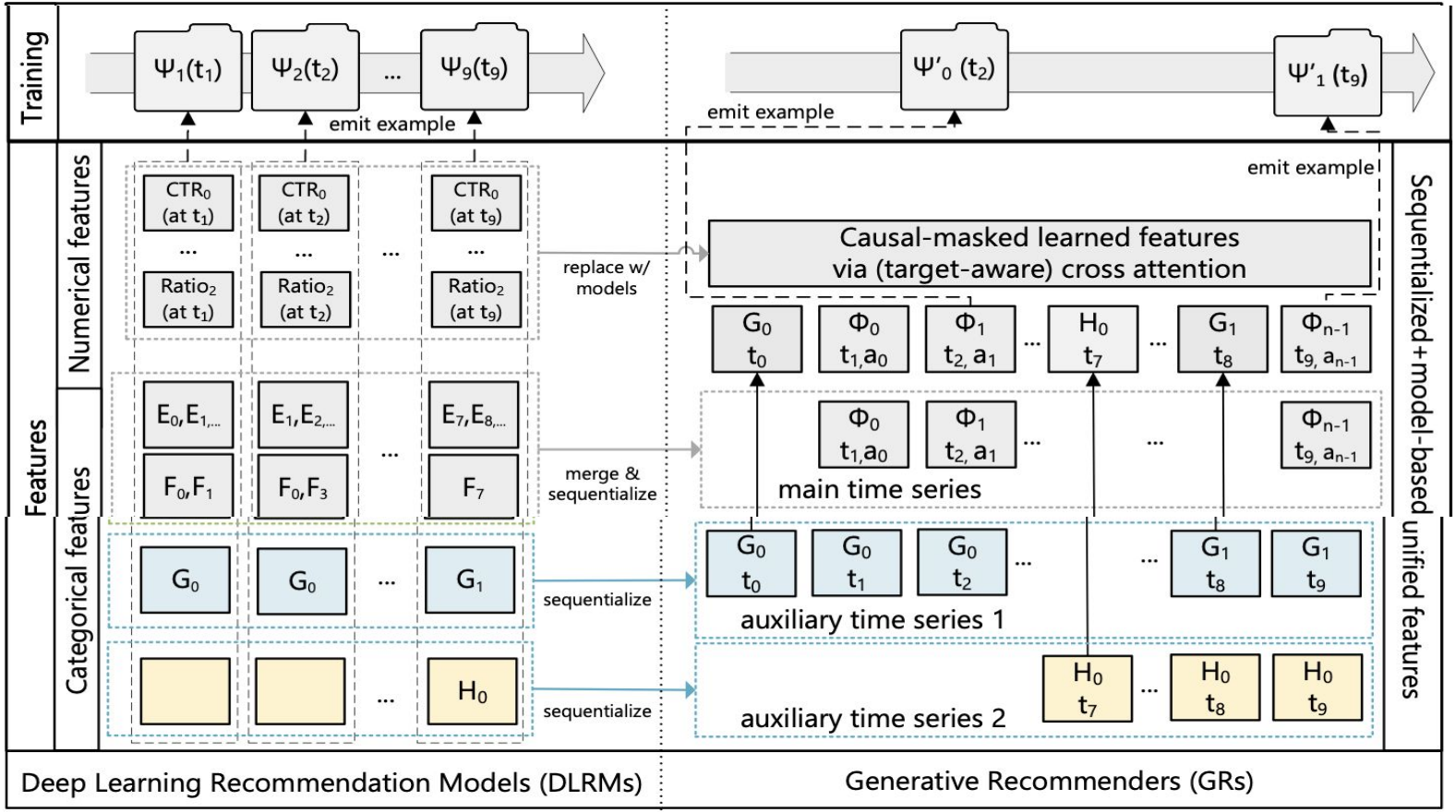
- Se pueden separar en dos tipos:
  a. *Features* de **interacción** del usuario con el ítem.
  b. Features de **información adicional** que no cambian significativamente con el tiempo (Ejemplo: variables demográficas)

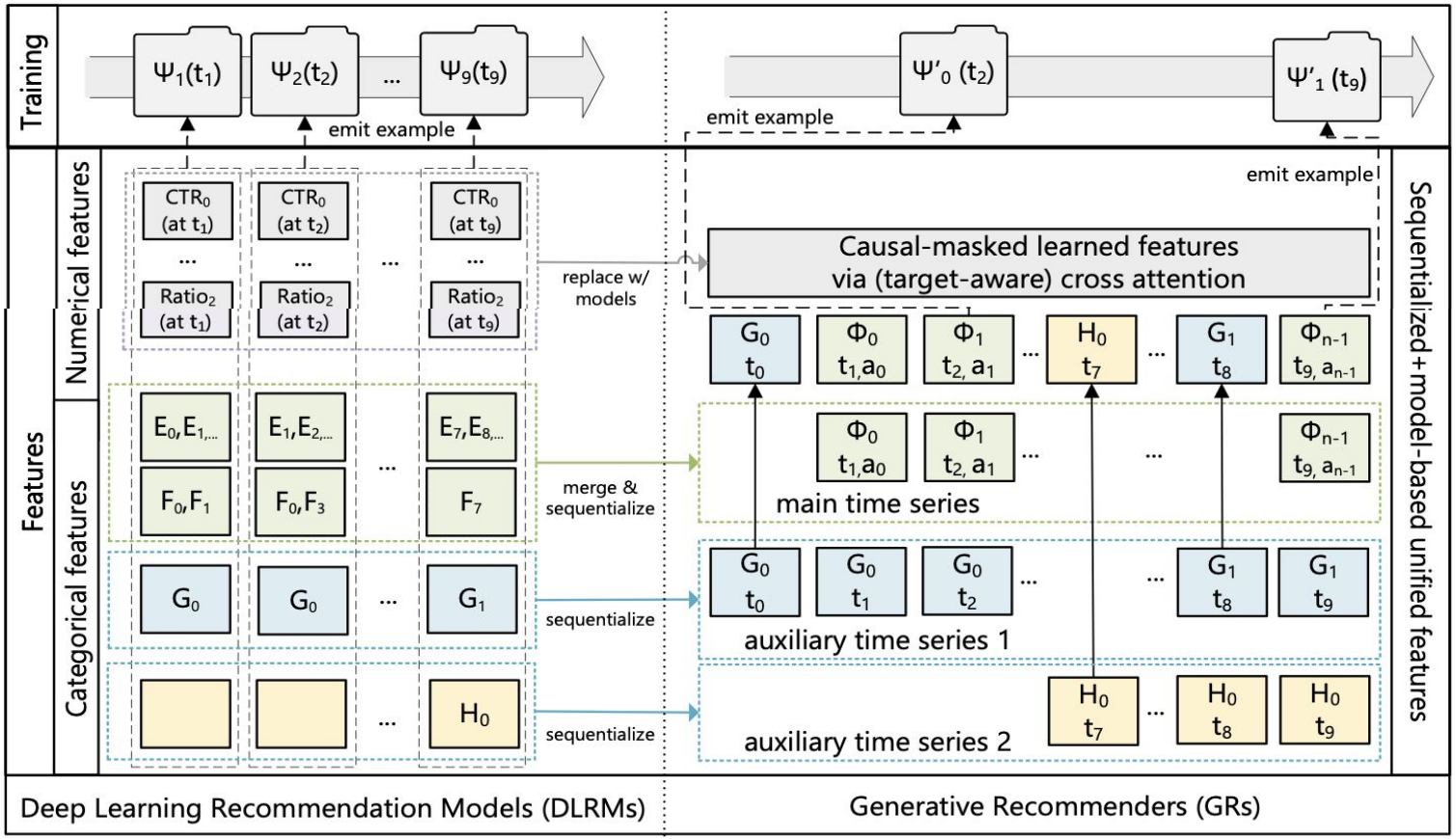
Para features numéricas:

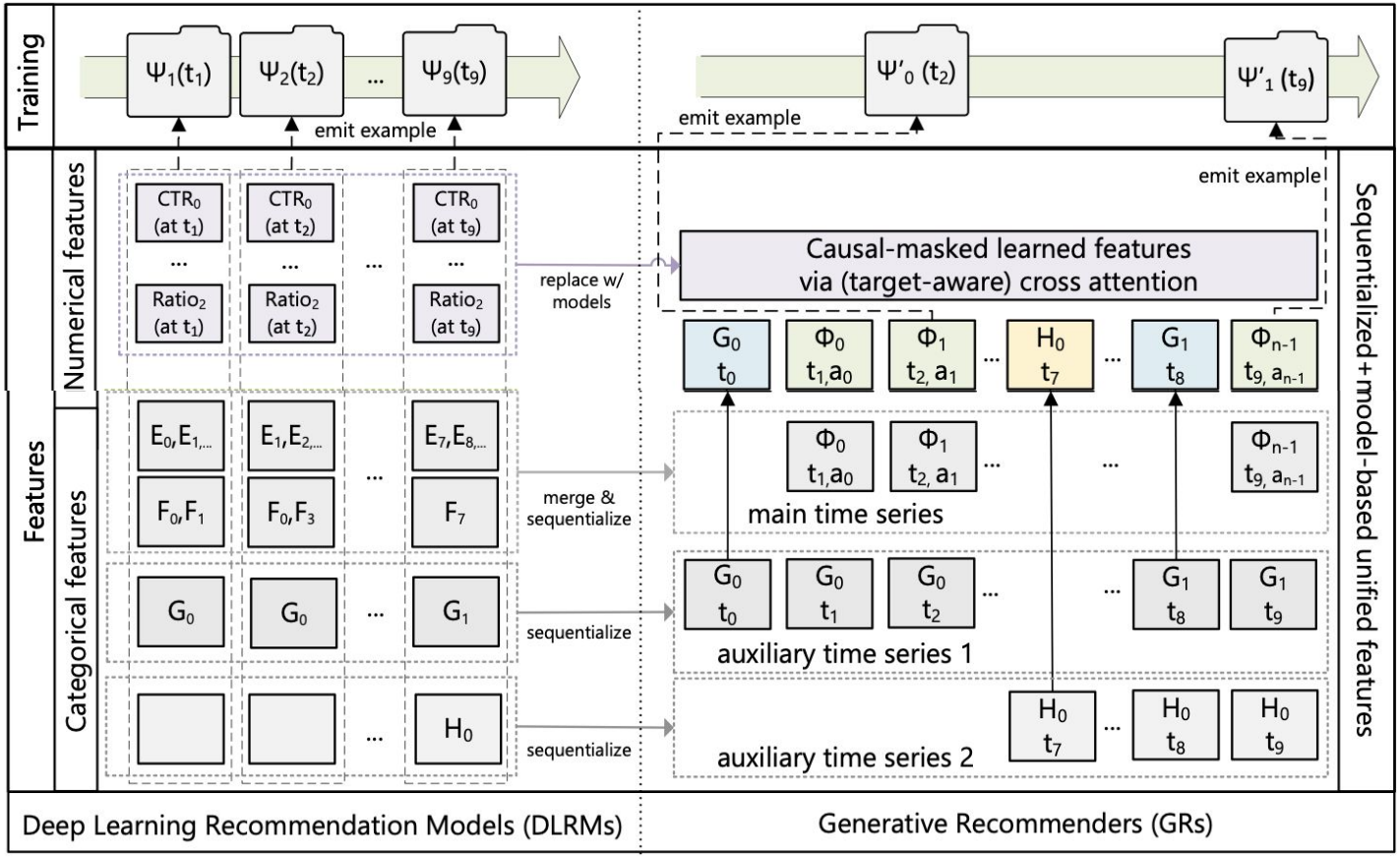
- Variables numéricas dependen de variables categóricas (Ejemplo: Click-Through Rate)
- Autores argumentan que se pueden remover estas features. Dado que las variables categóricas sobre las que se calculan estos valores agregados ya fueron agregadas a la secuencia.


Training
Ψ1(t1)
Ψ2(t2)
...
Ψ9(t9)
emit example
Ψ'0 (t2)
Ψ'1 (t9)
emit example
emit example
Features
Numerical features
Categorical features
CTR0 (at t1)
CTR0 (at t2)
CTR0 (at t9)
Ratio2 (at t1)
Ratio2 (at t2)
Ratio2 (at t9)
E0,E1,...
E1,E2,...
E7,E8,...
F0,F1
F0,F3
F7
G0
G1
H0
replace w/ models
merge & sequentialize
sequentialize
sequentialize
Causal-masked learned features via (target-aware) cross attention
G0 t0
Φ0 t1,a0
Φ1 t2, a1
H0 t7
G1 t8
Φn-1 t9, an-1
main time series
auxiliary time series 1
auxiliary time series 2
G0 t1
G0 t2
G1 t9
H0 t8
H0 t9
Sequentialized+model-based unified features
Deep Learning Recommendation Models (DLRMs)
Generative Recommenders (GRs)


Training
Features
Numerical features
Categorical features
Ψ1(t1)
Ψ2(t2)
...
Ψ9(t9)
emit example
Ψ'0 (t2)
Ψ'1 (t9)
emit example
emit example
CTR0 (at t1)
CTR0 (at t2)
CTR0 (at t9)
Ratio2 (at t1)
Ratio2 (at t2)
Ratio2 (at t9)
E0,E1,...
E1,E2,...
E7,E8,...
F0,F1
F0,F3
F7
G0
G0
G1
H0
replace w/ models
merge & sequentialize
sequentialize
sequentialize
Causal-masked learned features via (target-aware) cross attention
G0 t0
Φ0 t1,a0
Φ1 t2, a1
H0 t7
G1 t8
Φn-1 t9, an-1
main time series
auxiliary time series 1
auxiliary time series 2
G0 t1
G0 t2
G1 t9
H0 t8
H0 t9
Sequentialized+model-based unified features
Deep Learning Recommendation Models (DLRMs)
Generative Recommenders (GRs)


Training
Ψ1(t1)
Ψ2(t2)
...
Ψ9(t9)
emit example
Ψ'0 (t2)
Ψ'1 (t9)
emit example
emit example
Features
Numerical features
Categorical features
CTR0 (at t1)
CTR0 (at t2)
CTR0 (at t9)
Ratio2 (at t1)
Ratio2 (at t2)
Ratio2 (at t9)
E0,E1,...
E1,E2,...
E7,E8,...
F0,F1
F0,F3
F7
G0
G1
H0
replace w/ models
merge & sequentialize
sequentialize
sequentialize
Causal-masked learned features via (target-aware) cross attention
G0 t0
Φ0 t1,a0
Φ1 t2, a1
H0 t7
G1 t8
Φn-1 t9, an-1
main time series
auxiliary time series 1
auxiliary time series 2
G0 t1
G0 t2
G1 t9
H0 t8
H0 t9
Sequentialized+model-based unified features
Deep Learning Recommendation Models (DLRMs)
Generative Recommenders (GRs)


Training
Ψ1(t1)
Ψ2(t2)
...
Ψ9(t9)
emit example
Ψ'0 (t2)
Ψ'1 (t9)
emit example
emit example
Features
Numerical features
Categorical features
CTR0 (at t1)
CTR0 (at t2)
CTR0 (at t9)
Ratio2 (at t1)
Ratio2 (at t2)
Ratio2 (at t9)
E0,E1,...
E1,E2,...
E7,E8,...
F0,F1
F0,F3
F7
G0
G1
H0
replace w/ models
merge & sequentialize
sequentialize
sequentialize
Causal-masked learned features via (target-aware) cross attention
G0 t0
Φ0 t1,a0
Φ1 t2, a1
H0 t7
G1 t8
Φn-1 t9, an-1
main time series
auxiliary time series 1
auxiliary time series 2
G0 t1
G0 t2
G1 t9
H0 t8
H0 t9
Sequentialized+model-based unified features
Deep Learning Recommendation Models (DLRMs)
Generative Recommenders (GRs)


Training
Ψ1(t1)
Ψ2(t2)
...
Ψ9(t9)
emit example
Ψ'0 (t2)
Ψ'1 (t9)
emit example
emit example
Features
Numerical features
Categorical features
CTR0 (at t1)
CTR0 (at t2)
CTR0 (at t9)
Ratio2 (at t1)
Ratio2 (at t2)
Ratio2 (at t9)
replace w/ models
Causal-masked learned features via (target-aware) cross attention
G0 t0
Φ0 t1,a0
Φ1 t2, a1
H0 t7
G1 t8
Φn-1 t9, an-1
E0,E1,...
E1,E2,...
E7,E8,...
F0,F1
F0,F3
F7
merge & sequentialize
main time series
G0
G1
sequentialize
auxiliary time series 1
H0
auxiliary time series 2
Sequentialized+model-based unified features
Deep Learning Recommendation Models (DLRMs)
Generative Recommenders (GRs)

Reformulación ranking y retrieval

Secuencia de entrada: $x_0, x_1, \ldots, x_{n-1}\ (x_i \in \mathbb{X})$
Tokens que representan contenido, características históricas o interacciones de usuario, ordenados cronológicamente.

Secuencia de salida: $y_0, y_1, \cdots, y_{n-1}\ (y_i \in \mathbb{X} \cup \{\varnothing\})$,
Tokens que representan respuestas del usuario o predicciones realizadas por el sistema:

- $y_i = \varnothing$: Salida indefinida para tokens no interactivos.

$\Phi_i \in \mathbb{X}_c\ (\mathbb{X}_c \subseteq \mathbb{X})$ denota un contenido (imagen o video), son no estacionarios. El usuario puede responder a Φ_i con una acción a_i

Podemos definir ranking y *retrieval* como tareas de transducción secuencial

Retrieval

Queremos aprender la distribución

$$p(\Phi_{i+1}|u_i)$$

Para esto maximizamos la recompensa esperada:

$$\arg\max_{\Phi \in \mathbb{X}_c} p(\Phi|u_i)$$

De esta forma manejamos conjuntos no estacionarios y características ambiguas.

Ranking

Formulación *target-aware:* se requiere modelar interacciones entre el contenido objetivo Φ_{i+1} y características históricas.

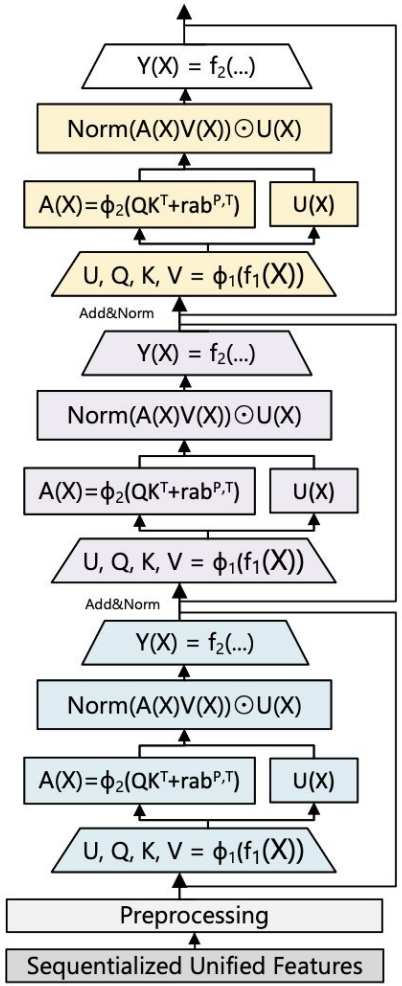
Para esto se intercalan los ítems y acciones formulando el ranking como

$$p(a_{i+1}|\Phi_0, a_0, \Phi_1, a_1, \ldots, \Phi_{i+1})$$

Se aplica una pequeña red neuronal que transforma las salidas Φ_{i+1}

De esta forma se integran interacciones históricas y objetivos futuros para un ranking más relevante y eficiente.

Hierarchical Sequential Transduction Unit (HSTU)



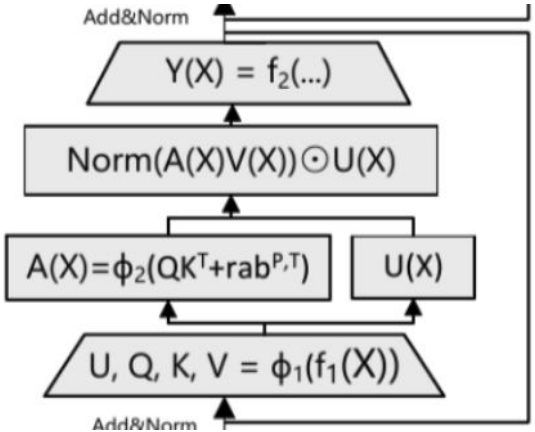
Hierarchical Sequential Transduction Unit (HSTU)

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

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

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

Hierarchical Sequential Transduction Unit (HSTU)



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

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

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

Elección de la arquitectura

Pointwise aggregated attention

“while softmax activation is robust to noise by construction, it is less suited for non-stationary vocabularies in streaming settings.”

El mecanismo de aggregated attention lo realiza como lo define en eq.2 donde existe un relative attention bias y Φ función no lineal SiLU.

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

Elección de la arquitectura

Pointwise aggregated attention

Architecture	HR@10	HR@50
Transformers	.0442	.2025
HSTU (-rab^{p,t}, Softmax)	.0617	.2496
HSTU (-rab^{p,t})	**.0893**	**.3170**

Table 2. Synthetic data in one-pass streaming settings.

HSTU mejora en Hit Rate@10 en más de 100% comparado con Transformers.

También, **reemplazando *HSTU's pointwise attention mechanism* con *softmax* también indica una reducción en el *hit rate***, verificando la importancia de *pointwise attention-like aggregation mechanisms.*

Elección de la arquitectura

Leveraging and algorithmically increasing sparsity

Cómputo Optimizado: Fusiona operaciones GEMM consecutivas y realiza atención completamente raggificada para mayor eficiencia en GPUs.

Self-attention en HSTU: Limitada por accesos a memoria, convirtiéndose en una operación dependiente de la memoria.

Mejoras de Rendimiento: Logra **ganancias de rendimiento de 2-5x**

Stochastic Length (SL): Aumenta el *sparsity* en las secuencias de historial del usuario, lo que permite:

- **Entrenamiento Eficiente**: Reduce la carga computacional.
- ***Cost-effective***: Importante debido a que los costos de entrenamiento suelen ser significativamente mayores que los de inferencia.

Resultado: Combina un diseño eficiente para hardware y *sparsity* para mecanismos de self-attention escalables y *cost-effective*.

Elección de la arquitectura

Leveraging and algorithmically increasing sparsity

Alpha (α)	Max Sequence Lengths			
	1,024	2,048	4,096	8,192
1.6	71.5%	76.1%	80.5%	84.4%
1.7	56.1%	63.6%	69.8%	75.6%
1.8	40.2%	45.3%	54.1%	66.4%
1.9	17.2%	21.0%	36.3%	64.1%
2.0	3.1%	6.6%	29.1%	64.1%

Table 3. Impact of *Stochastic Length* (SL) on sequence sparsity.

En la tabla de sparsity para diferentes longitudes de secuencia y valores de α, se muestran los resultados para una configuración a nivel industrial con un historial de usuario de 30 días.

Los ajustes que no provocan un retroceso significativo en la calidad del modelo están destacados en azul.

Elección de la arquitectura

Minimizing activation memory usage

“The use of large batch sizes is crucial for both training throughput (Mudigere et al., 2022) and model quality (Yang et al., 2020; Chen et al., 2020; Zhai et al., 2023a).”

La memoria de activación se convierte en una limitación de escalabilidad, a diferencia de los modelos de lenguaje grande dominados por el uso de memoria de parámetros.

Solución: HSTU

- **Número de linear layers outside attention** se ven disminuidas.
- **Fusión en el cómputo de funciones,**φ1(f1(·)) in Eq (1), layer norm, optional dropout, y output MLP in Eq (3).

Resultado: Escalabilidad, soporta modelos con **más de 2x deeper layers**, superando las limitaciones de memoria para el entrenamiento con *batches* grandes en sistemas de recomendación.

Elección de la arquitectura

Scaling up inference via cost-amortization

Los sistemas de recomendación deben procesar un gran número de candidatos al momento de recomendar. El ranking es crucial, ya que el costo del encoder es amortizable y existen algoritmos de retrieval eficientes.

Solución:
Proponemos **M-FALCON** (Microbatched-Fast Attention Leveraging Cacheable OperatioNs), un algoritmo que realiza inferencia de manera eficiente para m candidatos con tamaño de secuencia de input n.

Usando micro-batches es capaz de reducir la complejidad del cross-attention y utilizando un sistema de caché para cálculos de attention resultando que escala de mejor forma.

Resultado: Logramos un modelo de *cross-attention target-aware 285x more complex*, con una mejora de 1.5x a 3x, manteniendo un presupuesto de inferencia constante para configuraciones típicas de ranking.

Resultados

Ajustes Secuenciales

Datasets	Técnicas de Seteo	Métricas	Baseline
MovieLens	Full Shuffle	HitRate@k	SASRec
Amazon Reviews	Multi-epoch training	NDCG@k	

Resultados

Table 4. Evaluations of methods on public datasets in multi-pass, full-shuffle settings.

	Method	HR@10	HR@50	HR@200	NDCG@10	NDCG@200
ML-1M	SASRec (2023)	.2853	.5474	.7528	.1603	.2498
	HSTU	.3097 (+8.6%)	.5754 (+5.1%)	.7716 (+2.5%)	.1720 (+7.3%)	.2606 (+4.3%)
	HSTU-large	**.3294 (+15.5%)**	**.5935 (+8.4%)**	**.7839 (+4.1%)**	**.1893 (+18.1%)**	**.2771 (+10.9%)**
ML-20M	SASRec (2023)	.2906	.5499	.7655	.1621	.2521
	HSTU	.3252 (+11.9%)	.5885 (+7.0%)	.7943 (+3.8%)	.1878 (+15.9%)	.2774 (+10.0%)
	HSTU-large	**.3567 (+22.8%)**	**.6149 (+11.8%)**	**.8076 (+5.5%)**	**.2106 (+30.0%)**	**.2971 (+17.9%)**
Books	SASRec (2023)	.0292	.0729	.1400	.0156	.0350
	HSTU	.0404 (+38.4%)	.0943 (+29.5%)	.1710 (+22.1%)	.0219 (+40.6%)	.0450 (+28.6%)
	HSTU-large	**.0469 (+60.6%)**	**.1066 (+46.2%)**	**.1876 (+33.9%)**	**.0257 (+65.8%)**	**.0508 (+45.1%)**

Ajustes a Gran Escala

Ranking

Se utiliza Normalized Entropy como métrica

Se divide en E-tasks (Interaccion) y C-tasks (Consumo)

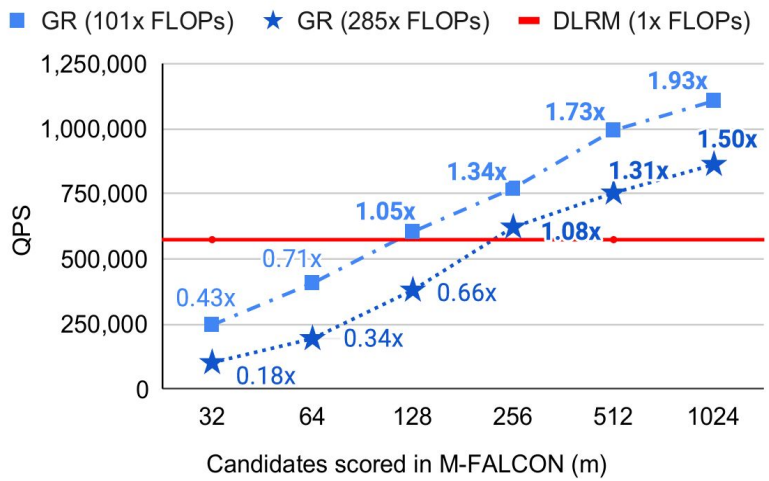
Retrival:
Log-perplexity

Architecture	Retrieval	Ranking (NE)	
	log pplx.	E-Task	C-Task
Transformers	4.069	NaN	NaN
HSTU (-rab^{p,t}, Softmax)	4.024	.5067	.7931
HSTU (-rab^{p,t})	4.021	.4980	.7860
Transformer++	4.015	.4945	.7822
HSTU (original rab)	4.029	.4941	.7817
HSTU	**3.978**	**.4937**	**.7805**

Evaluation of HSTU, ablated HSTU, and Transformers on industrial-scale data sets in one-pass streaming settings.

Eficiencia del Encoder con Stochastic Length

- **Stochastic Length (SL):** Técnica que ajusta la longitud de las secuencias.
- Reduce el costo de entrenamiento sin degradar métricas clave (>0.2%).
- **HSTU vs. Transformers:** HSTU es más eficiente en entrenamiento e inferencia.
- Beneficio: permite mantener la calidad del modelo mientras se optimizan recursos.



Comparison of inference throughput, in the most challenging ranking setup

GRs vs DLRMs

Escalabilidad

DLRMs saturan rendimiento al aumentar parámetros (>200 mil millones).

Recomendadores Generativos (GRs) alcanzan hasta **1.5 billones de parámetros**.

Eficiencia

GRs son más complejos, pero más eficientes en **queries per second (QPS)**.

Ventaja en entornos dinámicos y de gran escala.

Conclusiones y hallazgos principales

Conclusiones

- **Nuevo paradigma:** Introducción de **Generative Recommenders (GRs)**, que reformulan el **ranking** y la recuperación (***retrieval***) como tareas secuenciales, entrenadas de manera generativa.
- **Innovación técnica:** Implementación del **HSTU** Encoder, 5.3x-15.2x más rápido que Transformers en secuencias largas, optimizado con el algoritmo M-FALCON.
- **Mejores resultados: 12.4% de mejora en métricas** de producción y mejor escalabilidad en comparación con DLRMs tradicionales.
- **Simplificación de características:** Introducción de un **espacio de características unificado**, facilitando la creación de modelos fundacionales para recomendaciones, búsqueda y anuncios.
- **Configuración end-to-end:** Habilita una **formulación generativa** completamente integrada, mejorando la capacidad de los sistemas para asistir a los usuarios de forma más integral y holística.

Impactos:

- **Privacidad:** Menor dependencia de características heterogéneas, haciendo los sistemas más amigables con la privacidad.
- **Calidad de contenido:** Reducción de contenido perjudicial (e.g., clickbaits y noticias falsas), alineando los sistemas con los valores y metas de los usuarios.
- **Sustentabilidad:** Aplicaciones de modelos fundacionales que disminuyen la huella de carbono asociada con el desarrollo de sistemas de recomendación.

Referencias bibliográficas

- Chang, J., Zhang, C., Fu, Z., Zang, X., Guan, L., Lu, J., Hui, Y., Leng, D., Niu, Y., Song, Y., and Gai, K. Twin: Twostage interest network for lifelong user behavior modeling in ctr prediction at kuaishou, 2023.
- Chen, Q., Zhao, H., Li, W., Huang, P., and Ou, W. 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, DLP-KDD '19, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450367837. doi: 10.1145/3326937.3341261. URL https://doi.org/10.1145/3326937.3341261.
- Child, R., Gray, S., Radford, A., and Sutskever, I. Generating long sequences with sparse transformers. CoRR, abs/1904.10509, 2019. URL http://arxiv.org/abs/1904.10509.
- Hidasi, B., Karatzoglou, A., Baltrunas, L., and Tikk, D. Session-based recommendations with recurrent neural networks. In Bengio, Y. and LeCun, Y. (eds.), 4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings, 2016. URL http://arxiv.org/abs/1511.06939.
- Hou, Y., Zhang, J., Lin, Z., Lu, H., Xie, R., McAuley, J., and Zhao, W. X. Large language models are zero-shot rankers for recommender systems. In Advances in Information Retrieval - 46th European Conference on IR Research, ECIR 2024, 2024.
- Hua, W., Dai, Z., Liu, H., and Le, Q. V. Transformer quality in linear time. In Chaudhuri, K., Jegelka, S., Song, L., Szepesv´ ari, C., Niu, G., and Sabato, S. (eds.), International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA, volume 162 of Proceedings of Machine Learning Research, pp. 9099-9117. PMLR, 2022. URL https://proceedings.mlr.press/v162/hua22a.html.
- Katharopoulos, A., Vyas, A., Pappas, N., and Fleuret, F. Transformers are rnns: Fast autoregressive transformers with linear attention. In Proceedings of the 37th International Conference on Machine Learning, ICML'20. JMLR.org, 2020.

- Mudigere, D., Hao, Y., Huang, J., Jia, Z., Tulloch, A., Sridharan, S., Liu, X., Ozdal, M., Nie, J., Park, J., Luo, L., Yang, J. A., Gao, L., Ivchenko, D., Basant, A., Hu, Y., Yang, J., Ardestani, E. K., Wang, X., Komuravelli, R., Chu, C.-H., Yilmaz, S., Li, H., Qian, J., Feng, Z., Ma, Y., Yang, J., Wen, E., Li, H., Yang, L., Sun, C., Zhao, W., Melts, D., Dhulipala, K., Kishore, K., Graf, T., Eisenman, A., Matam, K. K., Gangidi, A., Chen, G. J., Krishnan, M., Nayak, A., Nair, K., Muthiah, B., khorashadi, M., Bhattacharya, P., Lapukhov, P., Naumov, M., Mathews, A., Qiao, L., Smelyanskiy, M., Jia, B., and Rao, V. Software-hardware co-design for fast and scalable training of deep learning recommendation models. In Proceedings of the 49th Annual International Symposium on Computer Architecture, ISCA '22, pp. 993–1011, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450386104. doi: 10.1145/3470496.3533727. URL https://doi.org/10.1145/3470496.3533727.
- Rabe, M. N. and Staats, C. Self-attention does not need o(n2) memory, 2021.
- Sileo, D., Vossen, W., and Raymaekers, R. Zero-shot recommendation as language modeling. In Hagen, M., Verberne, S., Macdonald, C., Seifert, C., Balog, K., Nørv˚ ag, K., and Setty, V. (eds.), Advances in Information Retrieval - 44th European Conference on IR Research, ECIR 2022, Stavanger, Norway, April 10-14, 2022, Proceedings, Part II, volume 13186 of Lecture Notes in Computer Science, pp. 223–230. Springer, 2022. doi: 10.1007/978-3-030-99739-7\26. URL https://doi.org/10.1007/978-3-030-99739-7_26.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. Attention is all you need. In Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS'17, pp. 6000–6010, 2017. ISBN 9781510860964.
- Yang, J., Yi, X., Zhiyuan Cheng, D., Hong, L., Li, Y., Xiaoming Wang, S., Xu, T., and Chi, E. H. Mixed negative sampling for learning two-tower neural networks in recommendations. In Companion Proceedings of the Web Conference 2020, WWW '20, pp. 441–447, 2020. ISBN 9781450370240.
- Zhou, G., Zhu, X., Song, C., Fan, Y., Zhu, H., Ma, X., Yan, Y., Jin, J., Li, H., and Gai, K. Deep interest network for click-through rate prediction. KDD '18, 2018.