SHOPPER: Semantic, Historical, Order-aware, and Product-conditioned Pooling for E-commerce Recommendation

Amit Jaspal
Meta Platforms, Inc.
Menlo Park, CA, USA
ajaspal@meta.com

Ashkan Sadeghi
Meta Platforms, Inc.
Menlo Park, CA, USA
ashkan@meta.com

Kevin Huang
Meta Platforms, Inc.
Menlo Park, CA, USA
kevh@meta.com

Nicolas Bievre
Meta Platforms, Inc.
Menlo Park, CA, USA
nbievre@meta.com

Behnaz Poursartip
Meta Platforms, Inc.
Menlo Park, CA, USA
behnazps@meta.com

Nikko Mizutani
Meta Platforms, Inc.
Menlo Park, CA, USA
nikkokun@meta.com

ABSTRACT

Modern large-scale e-commerce recommendation systems rely heavily on sequence modeling to infer user intent from historical actions. However, traditional models typically treat user journeys as flat lists of item identifiers and summarize user history early, without prior conditioning on the target item. This simplification discards rich contextual cues and limits cross-catalog product discovery, resulting in diluted, suboptimal product recommendations. We introduce **SHOPPER** (**S**emantic, **H**istorical, **O**rder-aware, and **P**roduct-conditioned **P**ooling for **E**-commerce **R**ecommendation) which addresses these bottlenecks through: (a) Semantic history, leveraging an upstream foundation model to extract rich multimodal event embedding of users shopping intent; (b) Order-aware local intent modeling, which enriches historical events with dense temporal metadata (e.g., dwell time, timestamp deltas) and applies a 1D convolutional layer to capture short-term, localized shopping intent bursts; and (c) Product-conditioned pooling, with serving-aware factorization, which splits computation into a heavy request-level phase (event enrichment) and a per-candidate phase (conditioning gate + cross-attention), enabling deep target-aware summarization — effectively isolating candidate-specific signals from noisy browsing histories.

Offline evaluation shows substantial NE reductions, and a live A/B test confirms +0.4% lift in top-line business metric while remaining deployable within strict latency constraints.

1 Introduction

Modern e-commerce recommendation systems must infer user intent from long, heterogeneous shopping journeys spanning browses, detail-page visits, add-to-cart actions, and purchases. The same interaction can carry different meaning depending on where it occurred, what action was taken, how long the user engaged, and how recently it happened. Existing model architectures for sequence modeling typically suffer from three bottlenecks:

- **Contextual and Temporal Stripping**: Sequential models process history as a flat list of product identifiers, stripping away contextual (e.g. page type, action type) and temporal metadata (e.g. temporal delta between clicks, user dwell time) which are potent indicators of shopping intent.
- **Semantic Sparsity**: The sheer scale and velocity of modern e-commerce catalogs introduce severe data sparsity. Purely ID-based sequence models act as effective memorization engines for retargeting familiar inventory, but they fundamentally fail to bridge disjoint ID spaces. Consequently, these models struggle to generalize user intent for broad prospecting campaigns or recognize cross-advertiser substitute and complementary relationships outside of highly engaged, previously clicked inventory.
- **Target-Agnostic Early Summarization**: Many architectures compress user's history into a single dense vector before it interacts with the target item, blurring diverse interests into an indistinguishable average. This design is not accidental — target-aware alternatives require per-candidate sequence computation, which is prohibitively expensive when scoring hundreds of candidates under strict latency budgets. The result is a forced trade-off between ranking quality and serving cost that many prior architectures accept.

We propose SHOPPER, a novel, industrial-scale sequence modeling framework for e-commerce recommendations. SHOPPER fundamentally restructures how user sequences are defined, contextualized, and evaluated against target items. Our main contributions are summarized as follows:

- **Order-aware Local Intent Modeling**: Moving deliberately beyond simple item IDs, we explicitly enrich historical journey events with dense temporal and contextual metadata (event types, timestamp deltas, dwell times, event position). Recognizing that acute shopping intent manifests in short-term temporal clusters, we apply a lightweight 1D convolution-based

pooling mechanism over the sequence axis to capture localized, multi-action intent bursts.

- **Semantic History Encoding via Foundation Models**: To bridge disjoint catalog spaces, SHOPPER enriches each historical event with dense semantic representations from an upstream foundation model trained on multimodal product metadata and engagement logs, establishing a shared space for cross-advertiser generalization in prospecting campaigns
- **Product-conditioned Pooling with Serving-aware Factorization**: We build a serving-efficient extension of InterFormer [1]-based early fusion model. SHOPPER computes enriched, order-aware user-history representations once at the request level. Candidate-specific interaction is then handled by a product-conditioned pooling module. This factorization preserves target-aware history selection while avoiding repeated full-sequence computation across hundreds of candidates in low latency, high QPS serving environment

SHOPPER's key innovation lies in the factorization: by performing expensive enrichment and local intent modeling once per request while deferring target product conditioning to the per-candidate path, SHOPPER resolves the quality-vs-latency trade-off that forces prior systems into target-agnostic summarization. This lets the model answer not only what the user did, but which parts of that behavior matter for the product currently being ranked.

2 Related Works

2.1 Sequential Recommendation

While modern sequential recommendation heavily utilizes attention-based architectures [2], most still represent journeys as plain item-ID sequences. This discards temporal signals that carry intent, as identical clicks can imply different intent depending on time gap and engagement depth. Prior work has therefore begun to incorporate richer temporal structure. TiSASRec [3], for example, models relative time intervals between events, while dwell-time-aware variants of GRU-based recommenders [4] show that time spent on an item page carries useful preference signal. Convolutional sequence models such as Caser [5] further highlight the local pattern extraction by capturing short-term, contiguous interaction signals. Our work builds on this line by combining temporal enrichment with localized pooling over shopping events, rather than treating the journey as a plain item-ID sequence.

2.2 Semantic Representation Learning

To alleviate sparsity for long-tail or newly listed items, foundation-model-based approaches [6, 7] extract rich, transferable item representations from textual, visual, and metadata features. Our work follows this general direction but focuses specifically on how those semantic embeddings should be injected into journey-level sequence modeling rather than used only as standalone item features. In SHOPPER, semantic representations are attached to historical events and participate directly in the product-conditioned pooling pipeline.

2.3 Target-aware User Modeling

Many recommenders summarize user history before the model sees the candidate item. This target-agnostic summarization is computationally convenient, but it can blur multiple interests into a single fixed vector and suppress the specific historical evidence that matters for a given candidate. Target-aware models were introduced to address exactly this issue. DIN [8] showed that historical behaviors should be weighted relative to the target item, and later models such as TAGNN [9] extended this principle to richer interaction structures. CARCA [10] applies cross-attention between the full profile and the target item, while recent industrial retrieval work such as GRank [11] pushes target-aware conditioning closer to large-scale retrieval systems. The shared intuition across these methods is that user history should be interpreted relative to what is currently being scored. SHOPPER builds on this insight but differs from prior work by conditioning a contextually and semantically enriched sequence on the target item before final summarization. In other words, the target does not simply reweight a precomputed user representation; it actively shapes which contextual and semantic parts of the history are extracted by the pooling module.

3 Methodology

3.1 Problem Formulation

We treat impression-level recommendation in a large-scale e-commerce system as a binary classification task. For a given request, the system observes a user u, a target candidate item v, and request-level context x. Let the recent shopping journey of user u be defined as a sequence of chronological historical events up to a maximum length T:

$$S_u = [e_1, e_2, \dots, e_T] \quad (1)$$

The objective is to learn a scoring function parameterized by θ to estimate the probability that user takes a target action i.e. click on candidate item

$$\hat{y}_{u,v} = f_\theta(u, S_u, v, x) \quad (2)$$

$\hat{y}_{u,v} \in (0,1)$ is the predicted probability of the target action and f_θ is a learnable model parameterized by θ.

3.2 Enriched Event Representation

SHOPPER defines each event as a multi-dimensional interaction tuple:

$$e_t = (i_t, a_t, p_t, d_t, \Delta_t, m_t) \quad (3)$$

Where:

- i_t : Interacted item ID.
- a_t: Action or event type (e.g., click, add-to-cart).
- p_t: Page or surface type where the event occurred
- d_t: Dwell time or engagement duration.
- Δ_t: Time gap relative to current timestamp
- m_t: Optional structured metadata

To move beyond ID-centric modeling, the first stage of SHOPPER maps each raw historical event into a dense vector by fusing contextual temporal event embeddings with upstream foundation model guided semantic representations.

3.2.1 Contextual Temporal Encoding: Each component of the event tuple is embedded separately and projected into a shared event space via a Multi-Layer Perceptron (MLP). This ensures the model captures the behavioral nuance of the interaction (e.g., distinguishing a quick browse from a long dwell):

$$h_t^{ctx} = \text{MLP}([E(i_t);\ E(a_t);\ E(p_t);\ E(d_t);\ E(\Delta_t);\ E(m_t)]) \quad (4)$$

(Where $E(.)$ denotes embedding lookup and numerical transformation, and [.;.] denotes concatenation).

3.2.2 Semantic Enrichment: To bridge disjoint advertiser catalogs and improve semantic generalization across broad prospecting campaigns, SHOPPER incorporates dense product semantics produced by an upstream foundation model trained over rich semantic features, including product text, structured attributes, taxonomy, and optionally image-derived signals. Drawing on recent advances in foundation model transfer frameworks such as LFM4Ads [12], we avoid flat feature extraction by comprehensively transferring dedicated event representations. The foundation model is trained on cross surface catalog engagement logs and its params are frozen during SHOPPER training. For each historical event e_t, we obtain a semantic representation:

$$z_t^{sem} = f_{FM}(e_t) \in R^{\{d\}} \quad (5)$$

where f_{FM} denotes the upstream foundation encoder. We adapt a gated fusion mechanism that allows the model to dynamically balance contextual-temporal and semantic signals per event. A learned gate determines how much semantic information to inject:

$$g_t = \sigma\left(W_g[\, h_t^{ctx}\, ;\, z_t^{sem}] + b_g\right) \quad (6a)$$

$$h_t = g_t \odot h_t^{ctx} + (1 - g_t) \odot z_t^{sem} \quad (6b)$$

where σ denotes the sigmoid function, $W_g \in R^{\{d*2d\}}$ and $b_g \in R^{\{d\}}$ are learnable gating parameters, [.;.] denotes concatenation. The gate g_t learns to suppress semantic signal when the contextual-temporal representation is already informative and to amplify it when ID-based representations are sparse e.g. for prospecting campaigns with unseen products.

3.3 Sequential Intent Modeling

SHOPPER factors sequential intent modeling into two computation phases (Figure 1). The request-level phase computes enriched, order-aware history representations once per user request, caches them. The per-candidate phase applies lightweight product-conditioned pooling over the cached representations for each candidate item. This factorization is what makes deep target-aware sequence modeling practical at production

3.3.1 Order Aware Local Intent Modeling: To avoid quadratic self-attention cost in high-QPS serving, we apply 1D convolutions with complexity O (N * K) over enriched history tokens, $H_u = [h_1, h_2, \ldots h_T]$ to capture short-term shopping patterns such as rapid product comparisons. With kernel size K=5 (selected via grid search over $K \in \{3, 5, 7, 9\}$), the convolutional operation is defined as:

$$C_t = \text{ReLU}\left(\sum_{j=-\lfloor \frac{K}{2} \rfloor}^{j=\lceil \frac{K}{2} \rceil} b + \left(W_j * h_{t+j}\right)\right) \quad (7)$$

where W_j denotes learnable weight matrix for position j within the sliding kernel window, b is the shared bias vector and h_{t+j} is the enriched history token treated as zero vector when $(t + j) < 1$ or $(t + j) > T$. We use learnable latent seed vector $s \in R^{\{d\}}$ to summarize the convoluted sequence C as follows:

$$z_t = \frac{s * C_t}{\sqrt{d}} \quad (8)$$

$$\alpha_t = \frac{\exp(z_t)}{\sum_{i=1}^{T} \exp(z_t)} \quad (9)$$

$$r_u = \sum_{i=1}^{T} (\alpha_t * C_t) \in R^{\{d\}} \quad (10)$$

The global intent representation vector r_u is the attention weighted sum of the local intent vectors. No te that this request-level computation is executed once and reused across all candidates.

3.3.2 Product Conditioned Pooling: To avoid blurring diverse user interests into a static, target-agnostic summary, SHOPPER introduces an early-fusion staging module. Rather than aggregating the history in isolation, the sequence is explicitly conditioned on the target candidate representation q_v. Inspired by personalized feature modulation techniques [1], the history tokens H_u first interact directly with the target candidate via a conditional gating network:

$$f_{cond}(q_v) = \sigma(W_2 * \text{ReLU}(W_1 * q_v + b_1) + b_2) \quad (11)$$

$$\widetilde{h_t} = f_{cond}(q_v) \odot h_t \quad (12)$$

This conditional interaction suppresses irrelevant history and amplifies signals relevant to the specific item being scored. Once conditioned, the model utilizes cross-attention to

summarize the sequence, using the target representation as the query:

$$s_u(v) = \text{CrossAttn}\left(q_v, \left[\widetilde{h_1}, \widetilde{h_2}, \ldots, \widetilde{h_T}\right]\right) \quad (13)$$

Because the enriched history tokens H_u and convolutional summary r_u are computed once at the request level (Section 3.3.1), only the conditioning gate (Eq. 11-12) and cross-attention (Eq. 13) execute per candidate — keeping the per-item cost to two lightweight operations over cached representations.

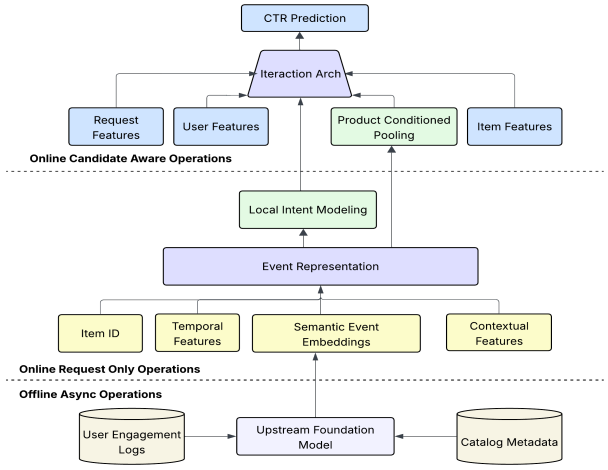

Figure 1: Overview of the SHOPPER architecture

3.4 Final Scoring and Optimization

The final prediction fuses five signal groups through a mix-net (DCN/DHEN-style feature interactions): global user and item features F_{uv}, the product-conditioned cross-attention summary $s_u(v)$, the localized convolutional summary r_u, the target candidate representation q_v, and the request context:

$$g_u(v) = \text{MixNet}([s_u(v); r_u; q_v; x; F_{uv}]) \quad (14)$$

The final predicted probability of the target action is calculated via a final MLP with a sigmoid activation:

$$\hat{y}_{u,v} = \sigma\left(\text{MLP}\left(g_u(v)\right)\right) \quad (15)$$

SHOPPER is trained in a ranking setting using logged impression-response pairs $\mathcal{D}$. We optimize the model using a standard binary cross-entropy (BCE) loss over the observed target action labels $y_{u,v} \in (0,1)$

$$L = -\sum_{(u,S_u,v,x,y_{u,v})\in\mathcal{D}} \left[y_{u,v}\log\hat{y}_{u,v} + \left(1-y_{u,v}\right)\log\left(1-\hat{y}_{u,v}\right)\right] \quad (16)$$

4 Offline Experiments

We conducted extensive offline experiments on a large-scale, proprietary e-commerce dataset to evaluate the proposed SHOPPER framework. The dataset consists of billions of logged impression-click pairs; to measure the ranking quality and calibration of our predictions, we use Normalized Entropy (NE) [13] as our primary evaluation metric, where a lower NE indicates superior performance. NE gain of 0.02% is considered significant for experiments on internal datasets

Our experimental study is designed to answer four key research questions:

- **(RQ-1) Overall Gains**: Does SHOPPER improve recommendations over strong sequential baselines?
- **(RQ-2) Contextual and Temporal Encoding Value**: Does enriching event with temporal metadata and contextual information resolve the stripping bottleneck discussed in Section 1?
- **(RQ-3) Semantic Enrichment Value:** Do semantic embeddings bridge the cross-catalog sparsity gap discussed in Section 1?
- **(RQ-4) SHOPPER's Gain Concentration:** Does conditioning on the target item before history summarization resolve the target-agnostic compression bottleneck identified in Section 1?

4.1 (RQ-1) Overall Gains

We benchmark SHOPPER against four sequential baselines, all using comparable DCN/DHEN-style mix nets. **PMA** [14]: attention-based pooling with learnable seed vectors; sequence-aware but not target-aware. **SASRec** [3]: self-attention over the event sequence; order-aware but not target-aware. **DIEN** [15]: sequence aware and applies target aware attention for history summarization without contextual or semantic enrichment. **InterFormer** [1]: early fusion interaction between history and target by applying product conditioning, cross attention and self-attention. To control for compute budget, we also evaluate a FLOP-equivalent InterFormer variant with a wider MLP that matches SHOPPER's total FLOPs.

As shown in Table 1, SHOPPER significantly outperforms all four baselines, achieving −0.31% NE reduction over PMA. The consistent improvement over both target-agnostic (PMA, SASRec) and target-aware (DIEN, InterFormer) baselines confirms that SHOPPER's gains stem from the combination of enriched event representations, order aware local intent modeling and product-conditioned pooling.

Table 1. SHOPPER Overall Offline Performance

Test Version	Seq Aware	Target Aware	ΔNE (%)
PMA	Yes	No	ref
SASRec	Yes	No	- 0.04
DIEN	Yes	Yes	- 0.09
InterFormer	Yes	Yes	-0.16
InterFormer FLOP-eq.)	Yes	Yes	- 0.19
SHOPPER	Yes	Yes	- 0.31%

4.2 (RQ-2) Contextual and Temporal Value

To answer RQ-2, we performed an ablation study isolating the incremental effects of (a) Contextual features (action types, page types), (b) Temporal features (timestamp deltas, dwell times), and (c) The 1D convolutional intent pooling.

Table 2 shows the addition of contextual and temporal features yielded significant reductions in Normalized Entropy (NE) compared to the ID-only baseline. These results confirm that the contextual and temporal-stripping bottleneck identified in Section 1 is a measurable source of prediction error: the baseline's flat-ID representation discards contextual intent signals worth −0.05% NE and temporal signal worth -0.02% NE respectively. The 1D convolutional layer recovers an additional −0.05% NE by capturing localized intent bursts that even per-event temporal features cannot represent, validating that short-term shopping intent manifests in multi-event temporal patterns.

Table 2. Ablation of Contextual and Temporal Modeling

Config	Component Added	Cumulative ΔNE (%)	Incremental ΔNE (%)
T1	Item Id only	ref	ref
T2	+ contextual	- 0.05	-0.05
T3	+ temporal	- 0.07	-0.02
T4	Full Encoder	- 0.12	-0.05

4.3 (RQ-3) Semantic Enrichment Value

We compare SHOPPER against a variant without semantic embeddings to isolate the value of upstream foundation model representations. As shown in Table 3, removing semantic embeddings degrades overall NE by +0.08%. Crucially, stratifying by campaign objective reveals a 2× larger degradation on prospecting campaigns (+0.15% NE), where users have no prior interaction with the target brand. This asymmetry confirms the semantic-sparsity bottleneck: ID-only models fail precisely where cross-catalog generalization is most needed and validates the incremental value of semantic foundation model embeddings to bridge disjoint catalog spaces

Table 3. Impact of Semantic Event Embeddings

	ΔNE (%)	
Config	Overall	Prospecting
SHOPPER	N/A	N/A
w/o semantic emb	+ 0.08%	+ 0.15%

4.4 (RQ-4) Product-Conditioned Pooling Value

To quantify the target-agnostic summarization bottleneck, we compare SHOPPER against a variant without the product-conditioned pooling module, stratified by user history length and intent diversity (Table 4). SHOPPER's gains increase monotonically with both history length (−0.12% → −0.23% → −0.35%) and intent diversity (−0.10% → −0.23% → −0.37%), confirming that early compression is most destructive for long, multi-intent journeys — precisely where product-conditioned pooling recovers the most signal.

Table 4. SHOPPER Gains by User History Length and Shopping Intent Diversity

	ΔNE (%)
By history length	
Short	- 0.12%
Medium	- 0.23%
Long	- 0.35%
By interest diversity	
≤1 category	- 0.10%
2-5 categories	- 0.23%
6-10 categories	- 0.37%

Table 5 synthesizes results across all four research questions, mapping each bottleneck from Section 1 to its architectural solution and supporting evidence.

5 Online Deployment

We conducted a two-week online A/B test on our e-commerce advertising platform, deploying SHOPPER on live ranking traffic against the current baseline. Integrating candidate-aware cross-attention into a high-throughput serving environment required three system optimizations: (i) foundation-model embeddings are pre-computed via a daily offline batch pipeline and served through lookup, decoupling the upstream encoder from the real-time request path; (ii) the enriched event representations and 1D convolutional module are executed once per user request and cached, eliminating redundant computation across the hundreds of candidates scored per impression; and (iii) the product-conditioned pooling and cross-attention are optimized for GPU-accelerated inference.

SHOPPER delivered a statistically significant 0.4% gain in the primary business metric, with strongest improvements for users with medium-to-long shopping histories — consistent with the offline finding (RQ-4) that product-conditioned pooling recovers the most signal where early compression is most destructive. The treatment introduced no p99 latency regression relative to the production baseline, confirming that the serving-aware factorization keeps the critical path lean.

Table 5. Mapping from identified bottlenecks to architectural components and experimental evidence

Bottleneck (Section 1)	Component (Section 3)	Evidence (Section 4)	Key Finding
Temporal stripping	Contextual, temporal features with 1D convolution encoder	RQ-2, Table 2	−0.12% NE from full encoder
Semantic sparsity	Foundation model embedding w/ gated fusion	RQ-3, Table 3	2× better on prospecting with semantic embeddings
Target-agnostic summarization	Product-conditioned cross attention	RQ-4, Table 4	Gains scale w/ sequence length and diversity

6 Conclusion and Limitations

SHOPPER addresses three representational bottlenecks in sequential e-commerce recommendation—contextual stripping, semantic sparsity, and target-agnostic summarization—by unifying order-aware local intent modeling, foundation-model based semantic enrichment, and product-conditioned cross-attention pooling in a single architecture. Offline experiments show substantial NE reductions over strong sequential baselines, with gains concentrated among long, multi-intent user shopping journeys. A live A/B test confirms a statistically significant 0.4% lift in the top-line business metric within production latency constraints.

Several operational limitations remain. Semantic representation quality depends on the upstream foundation model's update frequency, risking staleness on rapidly evolving catalogs. The fixed maximum sequence length discards older yet potentially informative history signals. Finally, offline pre-computation of semantic embeddings introduces cold-start gap for newly created items whose vectors are not yet available at serving time. Future work will explore incremental embedding updates and adaptive-length sequence modeling to mitigate these constraints.

REFERENCES

[1] Zeng, Z., Liu, X., Hang, M., Liu, X., Zhou, Q., Yang, C., Liu, Y., Ruan, Y., Chen, L., Chen, Y., Hao, Y., Xu, J., Nie, J., Liu, X., Zhang, B., Wen, W., Yuan, S., Yin, H., Zhang, X., et al. 2025. InterFormer: Effective Heterogeneous Interaction Learning for Click-Through Rate Prediction. In *CIKM 2025 - Proceedings of the 34th ACM International Conference on Information and Knowledge Management*. Association for Computing Machinery, 6225–6233. https://doi.org/10.1145/3746252.3761527

[2] Sun, F., Liu, J., Wu, J., Pei, C., Lin, X., Ou, W., and Jiang, P. 2019. BERT4Rec: Sequential Recommendation with Bidirectional Encoder Representations from Transformer. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management* (CIKM '19), November 3–7, 2019, Beijing, China. Association for Computing Machinery, New York, NY, USA, 1441–1450. https://doi.org/10.1145/3357384.3357895

[3] Li, J., Wang, Y., and McAuley, J. 2020. Time Interval Aware Self-Attention for Sequential Recommendation. In *Proceedings of the Thirteenth ACM International Conference on Web Search and Data Mining* (WSDM '20), February 3–7, 2020, Houston, TX, USA. Association for Computing Machinery, New York, NY, USA, 322–330. https://doi.org/10.1145/3336191.3371786

[4] Liu, Q., Zeng, Y., Mokhosi, R., and Zhang, H. 2018. STAMP: Short-Term Attention/Memory Priority Model for Session-based Recommendation. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD '18)*, August 19–23, 2018, London, United Kingdom. Association for Computing Machinery, New York, NY, USA, 1831–1839. https://doi.org/10.1145/3219819.3219950

[5] Tang, J. and Wang, K. 2018. Personalized Top-N Sequential Recommendation via Convolutional Sequence Embedding. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining (WSDM '18)*, February 5–9, 2018, Marina Del Rey, CA, USA. Association for Computing Machinery, New York, NY, USA, 565–573. https://doi.org/10.1145/3159652.3159656

[6] Liu, C., Hou, P., Zeng, A., and Yu, H. 2024. Transformer-Empowered Multi-Modal Item Embedding for Enhanced Image Search in E-commerce. *Proceedings of the AAAI Conference on Artificial Intelligence* 38, 21, 22770–22778. https://doi.org/10.1609/aaai.v38i21.30311.

[7] Czerwinska, U., Bircanoglu, C., and Chamoux, J. 2025. Benchmarking Image Embeddings for E-Commerce: Evaluating Off-the Shelf Foundation Models, Fine-Tuning Strategies and Practical Trade-offs. *arXiv preprint arXiv:2504.07567*. https://doi.org/10.48550/arXiv.2504.07567.

[8] Zhou, G., Song, C., Zhu, X., Fan, Y., Zhu, H., Ma, X., Yan, Y., Jin, J., Li, H., and Gai, K. 2018. Deep Interest Network for Click-Through Rate Prediction. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD '18)*. Association for Computing Machinery, New York, NY, USA, 1059–1068. https://doi.org/10.1145/3219819.3219823

[9] Yu, F., Zhu, Y., Liu, Q., Wu, S., Wang, L., and Tan, T. 2020. TAGNN: Target Attentive Graph Neural Networks for Session-based Recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '20)*. Association for Computing Machinery, New York, NY, USA, 1921–1924. https://doi.org/10.1145/3397271.3401319

[10] Rashed, A., Elsayed, S., and Schmidt-Thieme, L. 2022. CARCA: Context and Attribute-Aware Next-Item Recommendation via Cross-Attention. In *Proceedings of the 16th ACM Conference on Recommender Systems (RecSys '22)*. Association for Computing Machinery, New York, NY, USA, 71–80. https://doi.org/10.1145/3523227.3546777

[11] Sun, Y., Huang, S., Guan, Z., Luo, Q., Tang, R., Gai, K., and Zhou, G. 2026. GRank: Towards Target-Aware and Streamlined Industrial Retrieval with a Generate-Rank Framework. In *Proceedings of the ACM Web Conference 2026 (WWW '26)*. Association for Computing Machinery, New York, NY, USA, 7798–7808. https://doi.org/10.1145/3774904.3792810.

[12] Zhang, S., Quan, S., Wang, Z., Pan, J., Zhuang, T., Fu, B., Sun, Y., Lin, J., Chen, J., Li, X., Feng, Z., Hu, X., Deng, H., Lu, H., Wang, J., Dai, B., Chen, X., Hu, B., Huang, L., Wu, Y., Cai, Y., Zhou, Q., Tang, H., Yang, C., Yin, C., Jiang, T., Wang, L., Huang, S., Liu, D., Xiao, L., Gu, H., Xia, S.-T., and Jiang, J. 2025. Large Foundation Model for Ads Recommendation. *arXiv preprint arXiv:2508.14948*. https://doi.org/10.48550/arXiv.2508.14948

[13] He, X., Pan, J., Jin, O., Xu, T., Liu, B., Xu, T., Shi, Y., Atallah, A., Herbrich, R., Bowers, S., and Candela, J. Q. 2014. Practical Lessons from Predicting Clicks on Ads at Facebook. In *Proceedings of the Eighth International Workshop on Data Mining for Online Advertising (ADKDD '14)*. Association for Computing Machinery, New York, NY, USA, Article 5, 1–9. https://doi.org/10.1145/2648584.2648589

[14] Lee, J., Lee, Y., Kim, J., Kosiorek, A. R., Choi, S., and Teh, Y. W. 2019. Set Transformer: A Framework for Attention-based Permutation-Invariant Neural Networks. In *Proceedings of the 36th International Conference on Machine Learning (ICML '19)*. PMLR, 3744–3753. http://proceedings.mlr.press/v97/lee19d.html

[15] Zhou, G., Mou, N., Fan, Y., Pi, Q., Bian, W., Zhou, C., Zhu, X., and Gai, K. 2019. Deep Interest Evolution Network for Click-Through Rate Prediction. *Proceedings of the AAAI Conference on Artificial Intelligence* 33, 01, 5941–5948. https://doi.org/10.1609/aaai.v33i01.33015941