FinTRec: Transformer Based Unified Contextual Ads Targeting and Personalization for Financial Applications

Dwipam Katariya
Capital One, AI Foundations
McLean, USA

Snehita Varma
Capital One, AI Foundations
San Francisco, USA

Akshat Shreemali
Capital One, AI Foundations
New York, USA

Benjamin Wu*
Capital One, AI Foundations
San Jose, USA

Kalanand Mishra
Capital One, AI Foundations
San Jose, USA

Pranab Mohanty
Capital One, AI Foundations
San Jose, USA

ABSTRACT

Transformer-based architectures are widely adopted in sequential recommendation systems, yet their application in Financial Services (FS) presents distinct practical and modeling challenges for real-time recommendation. These include:a) long-range user interactions (implicit and explicit) spanning both digital and physical channels generating temporally heterogeneous context, b) the presence of multiple interrelated products require coordinated models to support varied ad placements and personalized feeds, while balancing competing business goals. We propose FinTRec, a transformer-based framework that addresses these challenges and its operational objectives in FS. While tree-based models have traditionally been preferred in FS due to their explainability and alignment with regulatory requirements, our study demonstrate that FinTRec offers a viable and effective shift toward transformer-based architectures. Through historic simulation and live A/B test correlations, we show FinTRec consistently outperforms the production-grade tree-based baseline. The unified architecture, when fine-tuned for product adaptation, enables cross-product signal sharing, reduces training cost and technical debt, while improving offline performance across all products. To our knowledge, this is the first comprehensive study of unified sequential recommendation modeling in FS that addresses both technical and business considerations. [1]

CCS CONCEPTS

• **Computing methodologies** → **Machine learning**; *Knowledge representation and reasoning*; • **Information systems** → **Recommender systems**; **Temporal data**.

KEYWORDS

personalization, ads targeting, user sequence modeling, financial services, transformers, joint modeling, recommender systems

*Contributed to the study when at Capital One. Now at NVIDIA

Correspondence to: Dwipam Katariya <dwipam.katariya@capitalone.com>
[1]The modeling referenced in this research is not reflective of actual, granular activities or tasks unique to any specific organization and rather are common across the financial services industry

ACM Reference Format:
Dwipam Katariya, Snehita Varma, Akshat Shreemali, Benjamin Wu, Kalanand Mishra, and Pranab Mohanty. 2025. FinTRec: Transformer Based Unified Contextual Ads Targeting and Personalization for Financial Applications. In *the 19th ACM Conference on Recommender Systems (RecSys '25), September 22–26, 2025, Prague, Czech Republic.* ACM, New York, NY, USA, 10 pages.

1 INTRODUCTION

Financial Service (FS) enterprises maintain a rich history of user's multichannel interactions, with millions of users generating billions of interactions daily across several digital products and physical channels. Users transact with credit and debit cards, browse digital channels, talk to bank associates to accomplish tasks that may or may not directly signal product interests and even get exposed to the related ads on external search engine before an actual purchase, as illustrated in Figure 1. (See Appendix B for more details on customer journey). Sequential recommendation systems [4–7, 10, 15, 19, 30, 34, 37, 46] have used explicit and implicit signals [11, 32, 35, 43, 48, 52, 53] such as clicks, dislikes/likes, and others, combining longitudinal and heterogeneous contextual data. (See Appendix A for more on related work). However, prior work has not explored other dynamic and non-digital signals such as transaction and payments history, despite their dominance in FS. Representing such temporal multi-channel, multi-product behavior is important for accurate real-time personalization, specifically at scale under low-latency constraints for high throughput applications. This requires a principled design of sequential representations and optimization of architectural and embedding strategies [33, 50]. FS also has a varied content mix with various objectives, as shown in Table 1. Hence, optimizing solely for Click-Through-Rate (CTR) risks adverse selection, promoting low-utility, clickbait items. Similarly, optimizing solely for Conversion-Rate (CVR) may degrade user experience. Furthermore, a mix of content is served at different touchpoints by interrelated products such as feed-style ranking and placement-style ads targeting. Since feed-style personalization may globally rank locally personalized ads content, it demands cross-product awareness. This proliferates product-specific models which are effective in isolation but result in increased maintenance cost and technical debt [2]. Moreover, such siloed models fail to leverage valuable cross-product interactions, leading to suboptimal performance [41]. Recent research [36, 38, 41, 42, 45] has explored the integration of search and recommendation systems, primarily within the domains of e-commerce, travel, and digital media. However, this work does not sufficiently address the FS sector,

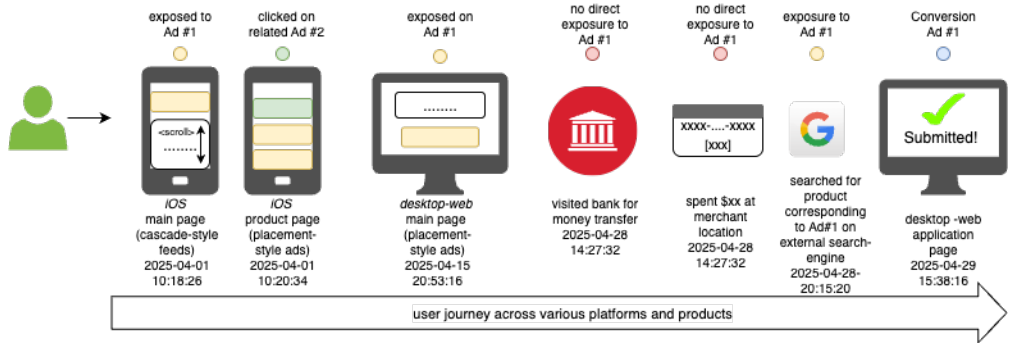


Figure 1: Illustrative example of user's journey across channels, products and pages

where large enterprises operate multiple interrelated products beyond search and recommendation systems. Moreover, FS still relies on tree-based approaches for all Machine Learning (ML) modeling tasks [3, 22, 39], primarily due to regulatory alignment (e.g., explaining loan application denials). This approach necessitates extensive feature engineering by domain experts, a process that is both time-consuming and complex. The evolving nature of these feature sets further complicates the management of large-scale ML systems, making it difficult to maintain and scale models efficiently. Hence, we present FinTRec, a unified transformer-based framework for personalized advertising and servicing needs for FS that: a) considers the user's raw sequential and heterogeneous context eliminating the need for feature engineering, b) balances both business value and user experience with a multi-objective function, c) addresses infrastructure complexity and cost in order to enable real-time inference by relying on a proprietary foundational model (FM) to capture the user's years worth of transactions, payments, statements and other context and, d) when fine-tuned (FT) for product adaptation, enables improved local optimization through cross-product awareness, reduces technical debt and deployment complexity.

2 TASK DEFINITION

Click-Through-Rate as *Next item Prediction*: Let $i \in \mathcal{I}_{\mathcal{T}}$ represent the candidate item shown to a user $u \in \mathcal{U}$ at time $t \in \mathcal{T}$. Hence, $\hat{p}_{\text{CTR}}(i,u)_t = \mathbb{P}(\text{click}_i = 1 \mid I_{t-1}, u_t)$.

Conversion-Rate: Traditionally, conversion rate (CVR) is defined as the probability of conversion given a user clicks. However, in FS platforms, exposure alone, without a click, can influence conversions. Additionally, users may convert without any platform exposure, arriving via external sources like search engines or direct links. Hence, $p\text{CVR}(i,u)_t = \mathbb{P}(\text{convert} = 1 \mid i, u_t)$.

Final Ranking: CTR reflects short-term engagement, while CVR captures long-term value. Optimizing solely for CTR can promote low-quality or clickbait content with limited business impact. In Platform Generated Content (PGC), where content supply is platform-driven, ranking must adapt quickly to shifting business priorities influenced by economic and policy changes. To support this, we introduce an urgency signal $us(i) \in \mathbb{R}_{\geq 0}$, quantifying the strategic importance of promoting item(i) at a given time(t) set at the campaign owner level. The final Ranking Score(RS) is computed as:

$$\text{RS}(i,u)_t = \lambda_{us} \cdot us(i)_t + \lambda_{ctr} \cdot p\text{CTR}(i,u)_t + \lambda_{cvr} \cdot p\text{CVR}(i,u)_t \cdot v(i)_t \quad (1)$$

where: $p\text{CTR}(i,u)_t$: predicted probability of a click after calibrating $\hat{p}_{\text{CTR}}(i,u)_t$, $p\text{CVR}(i,u)_t$: predicted probability of conversion, $v(i)$: present value(PV) per conversion in dollars, $us(i)$: urgency score, $\lambda_{us} \in \mathbb{R}_{\geq 0}$: stakeholder-controlled urgency weighting coefficient, $\lambda_{ctr} \in \mathbb{R}_{\geq 0}$: CTR weighting coefficient, $\lambda_{cvr} \in \mathbb{R}_{\geq 0}$: CVR weighting coefficient. This formulation enables flexible balancing of long-term value optimization and short-term business priorities, while preserving ranking interoperability. The urgency term acts as a soft override, allowing stakeholders to elevate the prominence of specific item categories without entirely bypassing user-preference signals (e.g., fed policy change affecting savings account APY). Since, PV is realized only when a conversion occurs, $p\text{CVR}(i,u)_t \cdot v(i)_t$ estimates the expected value per conversion. It should be noted that other signals do not have a monetary value associated with them. [2]

3 METHODOLOGY

3.1 Data Pre-processing

Clickstream activities are stored at the most granular level of web and mobile activity, along with the timestamp. Individual transactions - such as payments, ATM usage, call center events - are also stored in a tabular format at the interaction level and serve as valuable context signals for online ad targeting and personalization [8, 20]. A notable property of this data is its composition of two distinct signal types: dynamic and static [20, 47]. Dynamic signals, such as transactions, payments and money transfers, are characterized by their frequent changes over time. In contrast, Static signals, like a user's product ownership, maintain a persistent state over a longer temporal horizon. Table 2 illustrates sample data sources used for this study. A fundamental requirement for an accurate real-time recommendation is low latency feature fetching and transformation. This is particularly challenging when leveraging an extensive history of user interactions, including offline touchpoints such as transactions and payments. Given the impracticality of performing these transformations in real-time, due to both the

[2]For brevity, $\hat{p}_{\text{CTR}}(i,u)_t$ will be referred as pCTR and $p\text{CVR}(i,u)_t$ will be referred as pCVR

Table 1: Content mix served on FS digital products

Content Type	Illustrative Examples	Feedback
Marketing	open new accounts - credit card, loan, bank etc..	Convert
Servicing	enroll in paperless statement, check savings balance etc..	Click
Third-party	4x miles on Expedia, 5x cashback on Etsy, etc..	Click / Convert

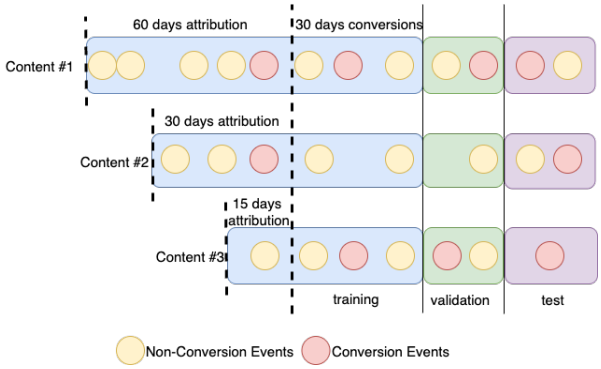


Figure 2: Yellow circle represent non−conversion events including impressions. Red circles represent conversions. For each conversion, item #1 queries data back to 60 days, while item #2 and item #3 queries data back to 30 and 15 days respectively. Then the data is temporally split for training/validation/test

volume and heterogeneous dimensionality of the data, we rely on our proprietary FM to aggregate dynamic heterogeneous context to point-in-time representation.

Our legacy Random Forest (RF) model [3, 22, 39] processes expert-engineered features derived from the dynamic and static context and clickstream data (e.g., total web/mobile logins and product page visits in 7/14/30 days, avg. conversion rates in 7/14/30 days since fraud reported, utlization rate etc.). A critical limitation of the RF approach is its reliance on flattening user interaction histories into time-aligned snapshots per impression. This simplification overlooks fine-grained temporal dependencies and evolving user preference signals. Another drawback is the model's difficulty in handling high dimensional embeddings - such as a 768-dimensional FM embeddings - which can lead to unstable feature sampling and degraded performance. To address this, we applied PCA to reduce the embeddings to 32 dimensions before training. In contrast, FinTRec directly leverages raw user interaction histories and 768-dim FM embeddings. It segments these histories into sequences, each culminating in a conversion/no-conversion and/or click/no-click event, thereby rigorously preserving chronological order and temporal dependencies across sessions, channels and products. Due to content-specific regulatory requirements such as 30 days marketing-opt out notices, and varied conversion time window (e.g., enrolling in credit monitoring is a quicker conversion time frame compared to enrolling in a savings account) requires careful content-specific attribution process. Our data processing pipeline allows for flexible conversion attribution as illustrated in Figure 2 and filters user sequences based on marketing opt outs. *Feature transformation*: Continuous context features undergo standardization. Categorical features like product enrollments, due to their multi-valued nature, are multi-hot encoded. Other categorical attributes (e.g., channel type, intra-page structured elements, layout, placements, device type) are tokenized as illustrated in Figure 3. FM embeddings are incorporated without further transformation. This process yields three fixed-dimensional context feature vectors - static context vector (F_s), dynamic context vector (F_d) , FM embedding vector (F_{fm}) per impression. For each user u, their tokenized interaction history is represented as a time-ordered sequence of items $S_u = (i_{t-l}, \ldots, i_{t-0})$, where l is the sequence length. Each item $i_t \in \mathcal{I}$ in S_u is associated with its context vector $\{F_s, F_d, F_{fm}\} \subseteq F_c$.

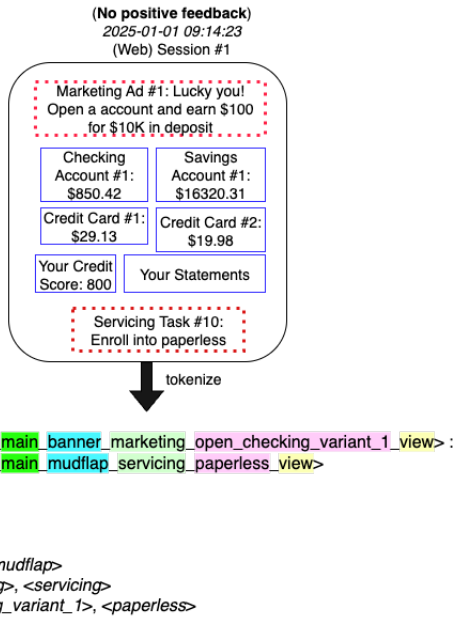


Figure 3: Illustrative example of intra-page tokenization. Tokens are grouped and arranged by their timestamps across pages

3.2 Model Architecture

A decoder-only architecture is used for pCTR (Figure 5b), whereas an encoder-only architecture is used for pCVR (Figure 5a). Encoder architectures are widely adopted for modeling conversion behavior, while decoder architectures are commonly used for next click

Table 2: Sample data sources used for this study

Data Source	Description	Type
Clickstream	Impressions and clicks across pages, users, platforms and products. Multiple impressions and clicks are grouped together per user session spanning from login to logout/idle > 30 mins	Dynamic
Heterogeneous Context	768-dim batched sequential representation of last 3 years transactions, payments, statements etc. with proprietary encoder-only FM	Dynamic
Product Enrollment	Products that user own	Static
Tenures	User relationship with all the products (in days)	Static
.....	.....	

prediction, as the sequential dependency of user clicks and immediate feedback naturally aligns with the autoregressive properties of decoders. Although conversion modeling involves long-term attribution and delayed feedback, there is no theoretical limitation that precludes the use of decoders for conversion or encoders for click prediction. In practice, however, the prevailing choice is often influenced by practical considerations, such as data storage schemas and serving infrastructure, which tend to favor encoders for conversion modeling and decoders for click prediction.

Sequence Processing: For both pCTR and pCVR models, the pipeline first aggregates F_d since user's most recent digital session into a dynamic context vector $F_{d(t)}$. S_u tokens are mapped to dense vectors via an input embedding layer and fused with the resulting item token embeddings at (t) with $F_{d(t)}$ to form a single, combined embedding vector. Given that F_s is time-invariant and F_{fm} are inherently sequential, the pCVR model utilizes F_s and F_{fm} corresponding to the latest interaction $(t-0)$. In contrast, the pCTR model retains F_s and and F_{fm} for all the time points within the sequence. **Order and Temporal Encoding**: To preserve the intra-session interaction order, we apply positional encodings p_t as proposed by [40]. To capture temporal dynamics, we construct a multivariate timestamp e_t representation by decomposing timestamp into dayofweek, weekofmonth, hourofday etc. We then compute an element-wise dot product between the p_t and e_t, and add the result to the input sequence embeddings resulting in $\hat{S_u}$. **Causal Decoder Block**: For pCTR task, We utilize a causal decoder-only architecture, where self-attention is masked based on timestamps rather than token positions, in contrast to the default formulation in [40] resulting in: $\mathbf{h}_t = \text{Decoder}(\hat{S_u})$. This forces strict time ordered look-back constraint. **Bi-Directional Encoder Block**: For pCVR, We adopt a bi-directional encoder block [40] resulting in $\mathbf{h}_u = \text{Encoder}(\hat{S_u})$. We then, pool the representations from $h_{(t-l)}$ to $h_{(t-0)}$ to capture a global summary of the interaction sequence: **Static and FM Context Fusion**: The decoder's output $\mathbf{h}_t$ is concatenated with $[f_{fm(t)}, f_{s(t)}]$ and $[f_{fm(0)}, f_{s(0)}]$ for encoder's output $\mathbf{h}_u$. This combined representation is passed through a multi-layer perceptron (MLP) followed by Sigmoid (in case of pCTR) and Softmax (in case of pCVR). **Loss Functions**: *pCVR*: For training the Random Forest model with down-sampled negatives, we use the standard Binary Classification loss followed by item-specific calibration. Transformers is trained with BCE and does not require calibration. *pCTR*: We adopt a modified Next Item Loss (as shown in eq.2) that includes only clicked items. Since certain item must be exposed regardless of clicks, the model is not optimized for predicting the immediate next item, but instead focuses on learning from positively engaged signals.

$$\text{Next Item Loss} = -\frac{1}{\sum_{u\in\mathcal{U}}|\mathcal{P}_u|}\sum_{u\in\mathcal{U}}\sum_{t\in\mathcal{P}_u} log\left(\frac{\exp(\hat{y}_{u,t}[y_{u,t+1}])}{\sum_{j=1}^{K}\exp(\hat{y}_{u,t}[j])}\right) \tag{2}$$

where, $\mathcal{U}$: set of users, $\mathcal{P}_u$: time steps t for user u such that a click occurred at t+1, $y_{u,t+1}$: the ground-truth clicked item at time t+1, $\hat{y}_{u,t} \in \mathbb{R}^K$: the logits (predicted scores) over the item vocabulary at time t

3.3 Product Adaptation FT

First, we collapse customer interactions across all the existing products and align them on a single flat timeline [20], followed by pre-training a FinTRec-decoder-only model. We then adopt Low-Rank Adaptation (LoRA) [17] for new product adaptation. During fine-tuning, only the low-rank product-specific matrices are updated. To incorporate product-specific semantics, we extend the pre-trained token embedding matrix with new learnable embeddings. Let $E \in \mathbb{R}^{V\times d}$ denote the original token embeddings for a vocabulary of size V and hidden size d. We introduce $E_{\text{new}} \in \mathbb{R}^{V_{\text{new}}\times d}$, representing embeddings for V_{new} new product(s). The combined embedding matrix becomes:

$$E' = \begin{bmatrix} E \\ E_{\text{new}} \end{bmatrix} \in \mathbb{R}^{(V+V_{\text{new}})\times d}$$

In our setup, E remains frozen during fine-tuning, while E_{new} is optimized jointly with the LoRA parameters. Formally, let θ_{LoRA} represent all LoRA parameters across selected transformer layers, and E_{new} the trainable token embeddings. The model is fine-tuned by minimizing a task-specific loss $\mathcal{L}$ over labeled training data (S_u, F_c, y):

$$\min_{\theta_{\text{LoRA}}, E_{\text{new}}} \mathcal{L}\left(f_{\text{base}}(S_u; F_c; \theta_{\text{frozen}}) + f_{\text{LoRA}}(S_u; F_c; \theta_{\text{LoRA}}, E_{\text{new}}), y\right)$$

Here, f_{base} is the frozen backbone model, and f_{LoRA} represents the learned low-rank adaptations and new token embeddings. This formulation ensures that only a small subset of model parameters along with new tokens are updated, enabling scalable and efficient adaptation to new products.

Linear Probing (LP) enables lightweight fine-tuning by updating only the final dense layer, while keeping all other model parameters frozen. **Full Fine-Tuning (F-FT)** All the model parameters of the

base transformer layers are updated.
Token embedding extension is also applied to both LP-FT and F-FT. Since different products have different output dimensions, a new output adapter is added by replacing the base model head. Output adapter is replaced for all three FT strategies. For F-FT and LoRA-FT, we also unfreeze parameters in the dense layer that correspond to F_s and F_{fm}. The goal of LP-FT was to set a lower bound on the maximum performance gain we could achieve with minimal updates; hence, the dense layers corresponding to F_s and F_{ffm} were kept frozen.

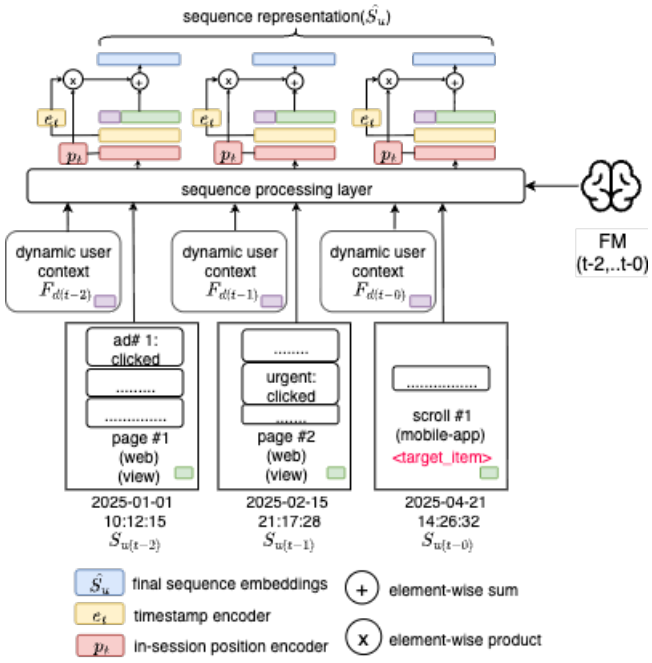
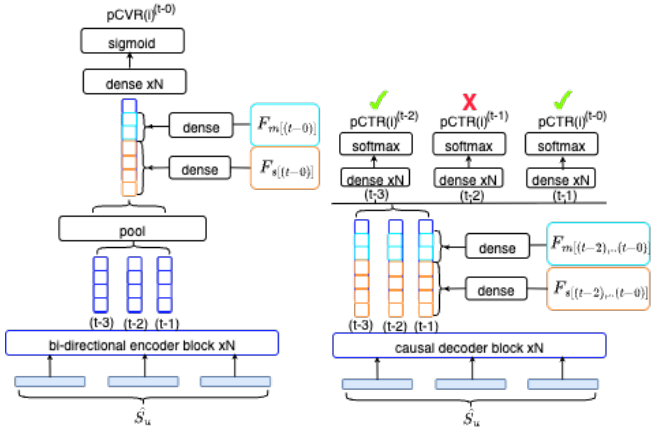


Figure 4: Dynamic context (F_d) and Clickstream sequence (S_u) are processed to form sequential user representations ($\hat{S}_u$). Static (F_s) and FM embeddings (F_{fm}) are time-aligned with $\hat{S}_u$

3.4 Experimental Setup

We use various data sources from the internal data ecosystem as mentioned in Table 1. For model training, we apply a temporal split of 90 days for training, 7 days for validation, and 7 days for testing on clickstream data. We focus our study on 30M+ user's sequences, 200+ items, 1BN+ interactions, and 10+ data sources across mobile and web channels.
Metrics: We adopt widely used evaluation metrics: log loss (for *pCVR*) and ranking recall@topk (for *pCTR*). *Log Loss*: quantifies the uncertainty in predictions required for trustworthy monetary value assessments. *Ranking Recall*: 99% of sessions display a maximum of five items (across products) at any given time. An effective personalization system aims to elevate relevant content to the highest possible rank within this limited display. Ranking Recall directly quantifies this performance, measuring how effectively relevant items are surfaced in such constrained visual interfaces.



(a) pCVR encoder-only architecture **(b) pCTR decoder-only architecture**

Figure 5: The pCVR model employs an encoder-only architecture (Figure 5a), while the pCTR model utilizes a causal decoder-only architecture (Figure 5b). Only positive samples are used for training pCTR, hence (t-2) is skipped from final loss calculation

Recall@topk is given as:

$$\text{Recall@top-k} = \frac{1}{\sum_{u \in \mathcal{U}} |\mathcal{P}_u|} \sum_{u \in \mathcal{U}} \sum_{t \in \mathcal{P}_u} \mathbb{1}\{I_{u,t+1} \in \hat{R}^n_{u,t}\} \quad (3)$$

as such, $\hat{R}^n_{u,t}$ is the top-k recommended items at time t for user u.
Product Adaptation: The effectiveness of our product adaptation approach was evaluated using a leave-product-out validation scheme. Specifically, product-specific interactions were held out during the pre-training phase before being subjected to the dedicated product adaptation strategy outlined in Section 3.3. We experiment our framework with both feed-style and placement-style products. We further tightly align our training with in-production inference in order to compute model performance metrics. For example, the PGC Marketing and Servicing model is inferred at the page level, while mobile homepage is inferred at each login.

Training Specifications: The FinTRec model was implemented using the PyTorch framework. We utilized NVIDIA 8 A10G GPUs and 512 GB of RAM. We adopted Distributed Data Processing (ddp) from PyTorch Lightning for parallelization. Each experiment ran for 50 epochs using the AdamW optimizer with a learning rate $1x10^-4$, a weight decay of $1x10^-5$ and a batch size of 64 per epoch. Model with the lowest validation log loss was selected. We did not employ early stopping. We utilize 10 multi-head attention blocks with hidden dimensions of 256 and input/output embedding dimension of 512.

4 OFFLINE RESULTS

Table 3 reports FinTRec's performance on the pCVR model for PGC marketing content. FinTRec (0.0439) substantially outperforms the production-grade RF baseline (0.0984) and RF + FM embeddings

Table 3: Experimental Results of FinTRec for pCVR task on PGC marketing content. Best results are boldfaced and highlighted

Model	Log Loss
RF + Feature Set	0.0984
RF + Feature Set (In Prod) + FM Embeddings	0.0938
FinTRec	**0.0439**
w/o time embeddings	0.0481
w/o FM embeddings	0.0605
context window=1	0.1135
context window=5	0.0844
context window=50	0.0478
context window=120	**0.0439**

Table 4: Experimental Results of FinTRec when FT for product adaptation with pCTR objective. Recall@1,5 shows relative lift in (%) over the product-specific baseline. Best results are boldfaced and highlighted. Second best are underlined.

Product	Model	Recall@1	Recall@5
PGC Servicing (placement-style)	(Product-Specific)	0.00	0.00
	FinTRec	-5.24	+2.90
	F-FT	**+26.85**	+20.09
	LP-FT	+11.41	+3.62
	LoRA-FT	+24.21	**+20.11**
Mobile Homepage (feeds-style)	(Product-Specific)	0.00	0.00
	FinTRec	-8.64	-16.34
	F-FT	+13.85	+5.15
	LP-FT	+5.36	+1.16
	LoRA-FT	**+14.11**	**+5.16**
Third-Party Marketing (placement-style)	(Product-Specific)	0.00	0.00
	FinTRec	-32.44	-24.14
	F-FT	**+25.63**	**+17.28**
	LP-FT	+8.17	+3.52
	LoRA-FT	+23.11	+15.75

LoRA-FT and LP-FT, required <5% and <1% of the pretrained model's parameters and <10% and <5% of its training time

(0.0938). Removing temporal encodings (0.0481) or FM embeddings (0.0605) degrades performance, demonstrating the importance of temporal dynamics and financial context. Long-range dependencies are prominent in FS, as demonstrated by the increased accuracy with a larger context window, a factor often overlooked or subdued by expert-engineered features. ***Product Adaptation***: Table 4 presents pCTR model gains across PGC Servicing, Mobile Homepage, and Third-Party Marketing. The Product-Specific model is a model developed in isolation based on the current framework, and incorporates both static and dynamic signals, as well as FM Embeddings. F-FT achieves the highest lift (up to +26.85% Recall@1). LoRA-FT reaches comparable performance (+24.21% Recall@1) with <5% parameter updates, and LP-FT offers smaller gains with <1% parameter fine-tuning, underscoring the importance of full temporal adaptation. LoRA-FT strikes a balance between model performance and training cost by improving code base standardization. FinTRec without fine-tuning under performs in most cases, due to: a) differences in pCTR across products , specifically negatively affecting Third-Party more than PGC Servicing, where pCTR for PGC Servicing is marginally higher and b) a lack of product-specific temporal alignment.

5 VISIT ABLATION

Understanding the contribution of individual user touch points and their interaction effects within a model is critical for multiple reasons. Beyond its utility for budget attribution, this interpretability is essential for regulatory compliance, such as with Fair Lending laws, the Fair Housing Act, and Model Risk Management. It also helps pave the way for consumer-facing explainability, as is increasingly mandated by regulatory bodies like the Consumer Financial Protection Bureau (CFPB) and the European Union's General Data Protection Regulation (GDPR) via the "right to explanation". Unlike aggregate feature importance, visit-level attribution makes such explanations possible, strengthens consumer trust and provides temporal granularity, enabling compliance teams and regulators to trace why a particular sequence of behaviors led to a specific recommendation. Hence, we employed multiple attention and gradient based explainability frameworks like attention weights [51] and GRAD-SAM [1]. Attention weights provide direct interpretability as the Transformer architecture learns how much weight to assign to each visit. Specifically we extract the attention matrix and average across heads and layers to obtain a scalar weight per visit, then normalize the scores to create a visit importance distribution. While Grad-SAM combines both the self attention weights and their gradients, producing a single relevance map. Just a few touch points can possess predictive power comparable to that of the entire interaction sequence, as shown in Table 6. This finding highlights the efficacy of our explainable insight in an event of regulatory findings. Applicability and differences from various other post-hoc gradient and SHAP based methods like Integrated Gradients [31] and Gradient-SHAP [21] are yet to be studied on our data.

6 INFERENCE AND SERVING

Real-time model deployment involves multiple execution stages to deliver final recommendations. We maintain a strict end-to-end latency requirement of 120ms at the 99th percentile for under 1500 api requests per second, irrespective of the number of model calls. Our solution is deployed on internally developed real-time serving platforms that handle both feature and model serving. The framework utilizes a combination of FM Embeddings and other contextual features computed and batched nightly, as well as features generated since the last batch job. At the time of a request, these features are retrieved and combined with a set of content determined to be eligible for the user, which is then provided to the model to generate the final ranking.

7 ONLINE A/B EXPERIMENTS

We conduct offline simulations to assess a model's potential impact on downstream business outcomes before it is deployed to production. Our simulation methodology, inspired by established

Table 5: Log Loss vs A/B test performance. Log Loss and PV shows relative lift in %. Act. PV shows actual realized PV in production. Est. PV when simulated offline pre-deployment is calculated based on final RS. Feature set corresponds to our proprietary user context features set as illustrated in Table 2. Our simulation do not guarantee the magnitude of log loss improvement resulting in equal magnitude of PV improvement.

Control	Test	Log Loss(%)	Act. PV(%) (Est. PV(%))
RF	RF + Feature Set	-7.00	+3.75 (<+6.08)
RF + Features Set	RF + Features Set + FM Embeddings	-4.67	+10.00 (<+37.50)
RF + Features Set	FinTRec + Features Set + FM Embeddings	-55.38	TBD (<+41.50)

Table 6: Visit quantification for explainability. AUROC is obtained from a model trained with important touch points. Relative AUROC in % is reported compared to the best performing model from the Table 3. (%) is rounded to the nearest integer.

Visit Importance	AUROC(%)
Most Important	-4.00%
Second Most Important	-5.00%
First Two Important Visit	-2.00%

Visit importance identified with the Attention Tracing methods described in [1, 51] and averaged across both the methods

approaches from Netflix [29] and Yahoo [24], systematically explores the search space of weighting coefficients ($\lambda_{us}, \lambda_{ctr}, \lambda_{cvr}$) to determine an optimal balance for the ranking score defined in Eq. 1. This process generates a sensitivity curve (Figure 6) that illustrates the trade-off between estimated Clicks and PV. The final coefficients are selected by product owners to align with current business objectives, typically by setting the values to match historically observed CTR and then noting the resulting increase in PV. Table 5 compares log loss reductions(%) with both actual and estimated total PV lifts from live A/B tests(%) and offline simulations. Using the latest dataset, we compare the top-ranked items from the current and new models using our final ranking objective, and estimate the lift in total PV. This evaluation shows that lower log loss generally correlates with higher user engagement, though the relationship is not strictly proportional. Adding the relevant feature set to the RF, the baseline yields a -7.00% log loss reduction and a +3.75% realized PV gain with offline estimate showing up to +6.08% PV gain. Further incorporating FM embeddings improves PV by +10.00%, with simulations estimating up to +37.50% lift. The largest log loss reduction (-55.38%) is achieved with FinTRec, with a projected simulated PV gain up to +41.50%. While actual PV impact is pending, we conclude the offline results support FinTRec deployment. Furthermore, we also monitor other guardrail metrics like Fraud Rate (eg: % Frauds reported in 30 days), 2hr. Call Rate post a user session, Credit Card On-Time Payments which may affect regulatory findings. In addition to guardrail metrics, tracking stakeholder-defined metrics is also essential to account for variations in PV, urgency scores and external factors like interest rates. For instance, business can decide thresholds for impression rates on specific content categories. This allows to detect unintended - a) model consequences early b) cannibalization due to content relatedness.

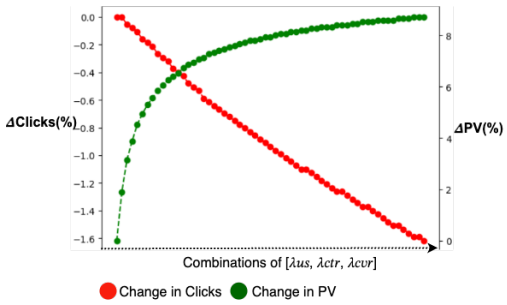


Figure 6: Change in expected clicks and PV based on offline simulation runs.

8 CONCLUSION

We introduces FinTRec and show superior performance over traditional and widely used tree-based baseline. We further show the effectiveness of incorporating time, long-range sequences, and implicit feedback via FM embeddings from years of transactional and other offline context. FM Embeddings enable real-time inference by eliminating the need for costly history retrieval. Fine-tuning across products further enhances performance through cross-product knowledge sharing, reduces infrastructure overhead, and improves codebase standardization. Historical A/B tests, simulations and explainability study confirms FinTRec's production readiness. **Limitations and Future Work:** a) pCTR and pCVR models are developed in separate codebases. Hence, future work will explore scalable architectures and unified frameworks for click and conversion models to further mitigate technical debt, maintenance, and training costs b) Our current FM embeddings are nightly batched and therefore stale, necessitating solutions for same day interaction awareness without incurring additional latency c) Although we applied explainability methods for visit importance in budget allocation and initial regulatory analysis, use-case-specific frameworks for finer alignment are yet to be fully explored. *Broader Impact Statement:* While studied in context of Financial Services and Banking, this work is also applicable to diverse industries such as e-commerce, retail, travel, and media. We invite other researchers to expand our study further.

9 AUTHOR BIO

Dwipam Katariya(Mclean, VA), Snehita Varma(San Francisco, CA) and Akshat Shreemali(New York, NY) are Data Science Manager at Capital One. Kalanand Mishra(San Jose, CA) is Director of Data Science at Capital One. Pranab Mohanty(Seattle, WA) is Sr. Director of Applied Research at Capital One. They all are part of recommendation and personalization team within the AI Foundations org. Benjamin Wu worked as Data Science Manager at Capital One, before joining NVIDIA as Sr. Solutions Architect.

ACKNOWLEDGMENTS

We thank Aditya Sasanur and Neil Kumar for their contribution to ML experiments. We thank Jesse Zymet for a thought provoking discussion and insights for highlighting value of the implicit feedback data. We are thankful to our collaborators from Foundation Models team. We thank Giri Iyengar for the support and sponsoring the study.

REFERENCES

[1] Barkan, O., Hauon, E., Caciularu, A., Katz, O., Malkiel, I., Armstrong, O., and Koenigstein, N. Grad-sam: Explaining transformers via gradient self-attention maps. *Proceedings of the 30th ACM Int'l Conf. on Information and Knowledge Management* (2021), 6.

[2] Bhattacharya, M., Ostuni, V., and Lamkhede, S. Joint modeling of search and recommendations via an unified contextual recommender (unicorn). In *Proceedings of the 18th ACM Conference on Recommender Systems* (2024), pp. 793–795.

[3] Borisov, V., Broelemann, K., Kasneci, E., and Kasneci, G. Deeptlf: robust deep neural networks for heterogeneous tabular data. *International Journal of Data Science and Analytics 16*, 1 (2023), 85–100.

[4] Cao, P., and Lio, P. Genrec: Generative personalized sequential recommendation. *arXiv preprint arXiv:2407.21191* (2024).

[5] Chang, J., Gao, C., Zheng, Y., Hui, Y., Niu, Y., Song, Y., Jin, D., and Li, Y. Sequential recommendation with graph neural networks. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval* (2021), pp. 378–387.

[6] 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* (2019), pp. 1–4.

[7] Chen, Y.-C., Chen, Y.-L., and Hsu, C.-H. G-transrec: A transformer-based next-item recommendation with time prediction. *IEEE Transactions on Computational Social Systems* (2024).

[8] Chitsazan, N., Sharpe, S., Katariya, D., Cheng, Q., and Rajasethupathy, K. Dynamic customer embeddings for financial service applications. *arXiv preprint arXiv:2106.11880* (2021).

[9] Covington, P., Adams, J., and Sargin, E. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems* (New York, NY, USA, 2016), RecSys '16, Association for Computing Machinery, p. 191–198.

[10] de Souza Pereira Moreira, G., Rabhi, S., Lee, J. M., Ak, R., and Oldridge, E. Transformers4rec: Bridging the gap between nlp and sequential/session-based recommendation. In *Proceedings of the 15th ACM conference on recommender systems* (2021), pp. 143–153.

[11] Fan, Z., Ou, D., Gu, Y., Fu, B., Li, X., Bao, W., Dai, X.-Y., Zeng, X., Zhuang, T., and Liu, Q. Modeling users' contextualized page-wise feedback for click-through rate prediction in e-commerce search. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining* (Feb. 2022), WSDM '22, ACM, p. 262–270.

[12] Garuti, F., Luetto, S., Sangineto, E., and Cucchiara, R. Large-scale transformer models for transactional data. In *CEUR WORKSHOP PROCEEDINGS* (2024), pp. 242 –247.

[13] Gorishniy, Y., Rubachev, I., Khrulkov, V., and Babenko, A. Revisiting deep learning models for tabular data. *Advances in neural information processing systems 34* (2021), 18932–18943.

[14] Grbovic, M., and Cheng, H. Real-time personalization using embeddings for search ranking at airbnb. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining* (2018), pp. 311–320.

[15] Hansen, C., Hansen, C., Maystre, L., Mehrotra, R., Brost, B., Tomasi, F., and Lalmas, M. Contextual and sequential user embeddings for large-scale music recommendation. In *Proceedings of the 14th ACM Conference on Recommender Systems* (2020), pp. 53–62.

[16] Hidasi, B., and Karatzoglou, A. Recurrent neural networks with top-k gains for session-based recommendations. In *Proceedings of the 27th ACM international conference on information and knowledge management* (2018), pp. 843–852.

[17] Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W., et al. Lora: Low-rank adaptation of large language models. *ICLR 1*, 2 (2022), 3.

[18] Huang, X., Khetan, A., Cvitkovic, M., and Karnin, Z. Tabtransformer: Tabular data modeling using contextual embeddings. *arXiv preprint arXiv:2012.06678* (2020).

[19] Kang, W.-C., and McAuley, J. Self-attentive sequential recommendation. In *2018 IEEE international conference on data mining (ICDM)* (2018), IEEE, pp. 197–206.

[20] Katariya, D., Origgi, J. M., Wang, Y., and Caputo, T. Timesync: Temporal intent modelling with synchronized context encodings for financial service applications. *arXiv preprint arXiv:2410.12825* (2024).

[21] Kokhlikyan, N., Miglani, V., Martin, M., Wang, E., Alsallakh, B., Reynolds, J., Melnikov, A., Kliushkina, N., Araya, C., Yan, S., and Reblitz-Richardson, O. Captum: A unified and generic model interpretability library for pytorch.

[22] Kotios, D., Makridis, G., Fatouros, G., and Kyriazis, D. Deep learning enhancing banking services: a hybrid transaction classification and cash flow prediction approach. *Journal of big Data 9*, 1 (2022), 100.

[23] Li, C., Liu, Z., Wu, M., Xu, Y., Zhao, H., Huang, P., Kang, G., Chen, Q., Li, W., and Lee, D. L. Multi-interest network with dynamic routing for recommendation at tmall. In *Proceedings of the 28th ACM international conference on information and knowledge management* (2019), pp. 2615–2623.

[24] Li, L., Chu, W., Langford, J., and Wang, X. Unbiased offline evaluation of contextual-bandit-based news article recommendation algorithms. In *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining* (New York, NY, USA, 2011), WSDM '11, Association for Computing Machinery, p. 297–306.

[25] Li, Z., Yang, C., Chen, Y., Wang, X., Chen, H., Xu, G., Yao, L., and Sheng, M. Graph and sequential neural networks in session-based recommendation: A survey. *ACM Computing Surveys 57*, 2 (2024), 1–37.

[26] Liu, C., Cao, J., Huang, R., Zheng, K., Luo, Q., Gai, K., and Zhou, G. Kuaiformer: Transformer-based retrieval at kuaishou. *arXiv preprint arXiv:2411.10057* (2024).

[27] Luetto, S., Garuti, F., Sangineto, E., Forni, L., and Cucchiara, R. One transformer for all time series: Representing and training with time-dependent heterogeneous tabular data. *arXiv preprint arXiv:2302.06375* (2023).

[28] Lv, F., Li, M., Guo, T., Yu, C., Sun, F., Jin, T., and Ng, W. Xdm: Improving sequential deep matching with unclicked user behaviors for recommender system. In *International Conference on Database Systems for Advanced Applications* (2022), Springer, pp. 364–376.

[29] McInerney, J., Brost, B., Chandar, P., Mehrotra, R., and Carterette, B. Counterfactual evaluation of slate recommendations with sequential reward interactions. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (New York, NY, USA, 2020), KDD '20, Association for Computing Machinery, p. 1779–1788.

[30] Moreira, G. d. S. P., Rabhi, S., Ak, R., Kabir, M. Y., and Oldridge, E. Transformers with multi-modal features and post-fusion context for e-commerce session-based recommendation. *arXiv preprint arXiv:2107.05124* (2021).

[31] Mukund Sundarajan, Ankur Taly, Q. Y. Axiomatic attribution for deep networks. *ICML* (2017).

[32] Ouyang, W., Zhang, X., Li, L., Zou, H., Xing, X., Liu, Z., and Du, Y. Deep spatio-temporal neural networks for click-through rate prediction. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (2019), pp. 2078–2086.

[33] Pancha, N., Zhai, A., Leskovec, J., and Rosenberg, C. Pinnerformer: Sequence modeling for user representation at pinterest. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining* (2022), pp. 3702–3712.

[34] Pi, Q., Bian, W., Zhou, G., Zhu, X., and Gai, K. Practice on long sequential user behavior modeling for click-through rate prediction. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining* (2019), pp. 2671–2679.

[35] Ren, K., Fang, Y., Zhang, W., Liu, S., Li, J., Zhang, Y., Yu, Y., and Wang, J. Learning multi-touch conversion attribution with dual-attention mechanisms for online advertising. In *Proceedings of the 27th acm international conference on information and knowledge management* (2018), pp. 1433–1442.

[36] Shi, T., Xu, J., Zhang, X., Zang, X., Zheng, K., Song, Y., and Yu, E. Unified generative search and recommendation. *arXiv preprint arXiv:2504.05730* (2025).

[37] Sun, F., Liu, J., Wu, J., Pei, C., Lin, X., Ou, W., and Jiang, P. Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM international conference on information and knowledge management* (2019), pp. 1441–1450.

[38] Sun, Z., Si, Z., Zang, X., Leng, D., Niu, Y., Song, Y., Zhang, X., and Xu, J. Kuaisar: A unified search and recommendation dataset. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management* (2023), pp. 5407–5411.

[39] Taha, A. A., and Malebary, S. J. An intelligent approach to credit card fraud detection using an optimized light gradient boosting machine. *IEEE access 8* (2020), 25579–25587.
[40] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. Attention is all you need, 2023.
[41] Wang, J., Zhang, Y., and Chen, T. Unified recommendation and search in e-commerce. In *Information Retrieval Technology: 8th Asia Information Retrieval Societies Conference, AIRS 2012, Tianjin, China, December 17-19, 2012. Proceedings 8* (2012), Springer, pp. 296–305.
[42] Xie, J., Liu, S., Cong, G., and Chen, Z. Unifiedssr: A unified framework of sequential search and recommendation. In *Proceedings of the ACM Web Conference 2024* (2024), pp. 3410–3419.
[43] Xie, R., Ling, C., Wang, Y., Wang, R., Xia, F., and Lin, L. Deep feedback network for recommendation. In *Proceedings of the twenty-ninth international conference on international joint conferences on artificial intelligence* (2021), pp. 2519–2525.
[44] Yang, J., Yi, X., Cheng, D. Z., Hong, L., Li, Y., Wang, S., Xu, T., and Chi, E. H. Mixed negative sampling for learning two-tower neural networks in recommendations.
[45] Yao, J., Dou, Z., Xie, R., Lu, Y., Wang, Z., and Wen, J.-R. User: A unified information search and recommendation model based on integrated behavior sequence. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management* (2021), pp. 2373–2382.
[46] Yuan, W., Wang, H., Yu, X., Liu, N., and Li, Z. Attention-based context-aware sequential recommendation model. *Information Sciences 510* (2020), 122–134.
[47] Zhang, D., Wang, L., Dai, X., Jain, S., Wang, J., Fan, Y., Yeh, C.-C. M., Zheng, Y., Zhuang, Z., and Zhang, W. Fata-trans: Field and time-aware transformer for sequential tabular data. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management* (2023), pp. 3247–3256.
[48] Zhang, H., Meng, C., Guo, W., Guo, H., Zhu, J., Zhao, G., Tang, R., and Li, X. Time-aligned exposure-enhanced model for click-through rate prediction. *arXiv preprint arXiv:2308.09966* (2023).
[49] Zhang, H., Wang, S., Zhang, K., Tang, Z., Jiang, Y., Xiao, Y., Yan, W., and Yang, W.-Y. Towards personalized and semantic retrieval: An end-to-end solution for e-commerce search via embedding learning. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval* (2020), pp. 2407–2416.
[50] Zhang, W., Li, D., Liang, C., Zhou, F., Zhang, Z., Wang, X., Li, R., Zhou, Y., Huang, Y., Liang, D., et al. Scaling user modeling: Large-scale online user representations for ads personalization in meta. In *Companion Proceedings of the ACM Web Conference 2024* (2024), pp. 47–55.
[51] Zhengxuan Wu, Thanh-Son Nguyen, Y. C. O. Structured self attention weights encode semantics in sentiment analysis. *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP* (2020), 255–264.
[52] Zhou, G., Mou, N., Fan, Y., Pi, Q., Bian, W., Zhou, C., Zhu, X., and Gai, K. Deep interest evolution network for click-through rate prediction. In *Proceedings of the AAAI conference on artificial intelligence* (2019), vol. 33, pp. 5941–5948.
[53] 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. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining* (2018), pp. 1059–1068.

A RELATED WORK

Recent advancements in recommendation systems have shown the importance of using sequential machine learning methods for representing a user's Point-of-Interest (POI) based on their past interactions and automatically recommending interested items. Users provide explicit and implicit feedback from interactions such as clicks, conversions, likes, dislikes, comments, shares, and/or dwell time. However, utilizing all user historical information poses challenges in capturing evolving user preferences over time. This has led to the emergence of session-aware recommendation (SAR) [25] models that adapt dynamically to ongoing user behavior. SAR has witnessed widespread adoption within academia as well as the industry, growing over 2x YoY since FY 2019 [25]. For instance, SAR has been widely studied within the e-commerce, travel, and entertainment industries. Recently, transformers have shown superior performance for SAR over RNNs and other traditional Machine Learning Models [6, 19, 26, 33, 37]. Although a plethora of research addresses peculiarities of transformers for SAR [19, 37], these academic efforts often fail to address certain industrial challenges, which has limited their effectiveness in driving success in large-scale recommendation systems [26]. For example, AliBaba [6] introduced a behavior sequence transformer for e-commerce recommendation and demonstrated the superiority of sequential actions over an embedding and MLP paradigm—where raw features are embedded into low-dimensional vectors and passed into an MLP for final recommendation. However, while the model showed improvements, the study did not sufficiently address industrial challenges like scalability and inference latency for large-scale recommendation systems. Pinterest studied the effectiveness of deploying a model that leverages a user's historical actions within a large-scale recommendation system, focusing on an industrial solution capable of handling billions of pins (items) and 400M+ MAU. Their work highlighted the complexities of SAR, including using streaming infrastructure to capture the latest user activity and managing data to encode the user state in real-time [33]. Furthermore, the retrieval task—a precursor to the final recommendation—is responsible for selecting thousands of candidate items from billions of options. Applying Transformers directly to this large-scale operation in real-time is computationally prohibitive [26]. To address these challenges, a specialized architecture, KuaiFormer, was introduced by Kuaishou. Besides SAR studied in the e-commerce, travel, and entertainment domains, SAR has also been deployed in wealth management and, more recently, has started to gain adoption in retail banking. Several competitions on Kaggle held by Santander [16] highlight the growing need for recommending relevant retail banking products to users. Although these studies have been discussed in the realm of model architectures, datasets, evaluation, and bias detection, there are limited studies available on industrial applications of transformers in the Financial Services domain. And unlike digital-only platforms, users in FS also interact across various channels, for example, by logging into mobile and/or web applications, using their credit/debit cards, physical locations, etc., generating a rich history of user's sequential interaction [20]. TIMeSynC [20] addressed this through a data processing model architecture, yet failed to address the real-time aspect of data retrieval. Customer interactions across these channels are not only explicit but also leave trails of their implicit feedback. While several studies have focused on modeling positive feedback—such as clicks or conversions using sequential models—negative signals such as skips and exposure without engagement have been often overlooked [11]. DIN [53] made an early attempt to capture the diversity of user interests by designing an adaptive activation mechanism, allowing the model to selectively attend to relevant historical behaviors based on current context. However, DIN is limited in its ability to model the evolution of latent interests over time [52]. DIEN [52] extended DIN by adding a latent temporal interest extractor and an explicit user behavior extractor, followed by a GRU-based sequential layer to learn dynamic user representations. Despite these improvements, DIEN still does not explicitly account for negative feedback signals. DSTN [32] and DFN [43] highlighted the importance of modeling both a user's positive and negative (exposure, unclicked) feedback for CTR prediction tasks. Exposure data is abundant, and recent studies demonstrated the usefulness of using such data in behavioral sequential tasks. For instance, XDM [28] demonstrated the effectiveness of exposure data to help guide

positive explicit clicks. DFN [43], DSTN [32], DIEN [52], and XDM [28] all incorporated users' positive and exposure feedback independently, ignoring mutual interplay and contextual dependencies between them. Moreover, they also overlooked the influence of spatial page context on user behavior. RACP [11] addressed this gap by jointly modeling both the exposure and click sequences, capturing their interactions. RACP used a hierarchical attention architecture enabling the model to learn both intra-page context (how items on the same page influence each other) and inter-page interest shifts across sessions. Despite notable advancements, all these studies overlook the effect of sequence truncation in multi-feedback settings, undermining the cross-correlation between different feedback signals. TEM4CTR [48] employs a three-stage framework by first searching unclicked records close to clicked records, followed by an attention mechanism to extract relevant exposure information, and finally an interest extraction module to jointly extract a user's latent interest representations from both the clicked and unclicked sequences for the CTR prediction task. However, these studies do not address other implicit feedback behavior data generated from cross-channel behaviors that are usually observed and dominant in the FS domain. Recent studies have shown the importance of combining cross-channel behavior for real-time intelligent personalization. [9, 14, 15, 23, 34, 44, 49] studied pre-trained transfer learning approaches and demonstrated the efficacy of embedding data streams combined with the final model to be suitable for real-world systems. However, FS has several such data streams, and combining them for low latency inference is yet to be studied. This requires combining pre-trained embeddings with sequential data. Tab-Transformer [18] contextualized categorical features but lacked contextualization across other feature types [13]. Both FT-Transformer and Tab-Transformer studied the architectural changes on several datasets including bank marketing datasets; however, they lacked interaction signals from other heterogeneous data such as payments, transactions, etc., that are dominant in FS [13]. Due to various data sources and multi-modality, developing embeddings on individual data streams and using them in downstream models has been previously discussed [8, 12, 27, 33]. [8] discussed clickstream embeddings as an input for fraud, intent prediction, and tasks in FS. Similarly, [27] demonstrated the impact of payments and transaction embeddings on Fraud as a downstream task. Thus, these studies focused on single data stream representation and lacked combining representation from different data streams for accurate marketing and ads recommendation in FS. Furthermore, it's a common practice to perform feature engineering and convert raw sequences into tabular data [3, 22] for machine learning algorithms such as Gradient Boosted Trees, more specifically in the FS and banking domain [39] due to their higher explainability for closer regulatory alignment. However, this requires extensive feature engineering on longitudinal tabular data leading to long hours spent in developing accurate features. Traditionally, organizations have developed product-specific machine learning models to deliver personalized user experiences. While effective in isolation, this approach often results in fragmented infrastructure and substantial technical debt [2]. Moreover, such siloed models fail to leverage valuable cross-product interactions, leading to suboptimal performance [41]. Recent research [36, 38, 41, 42, 45] has explored unifying search and recommendation systems, primarily within the domains of e-commerce, travel, and digital media. However, this body of work does not sufficiently address the financial services (FS) sector, where enterprises often operate multiple interrelated products beyond just search and recommendation systems. In the FS industry, personalization efforts and advertising strategies have similarly relied on distinct, product-specific models, thereby overlooking potential gains from shared signals across products. Notably, naive attempts to merge signals from diverse products have revealed a fundamental trade-off: improvements in one task frequently degrade performance in others [41]. This challenge is compounded by the heterogeneity of sequential signals across products—each product may depend on distinct features, with only partial overlaps—resulting in redundant feature engineering and inefficient inference pipelines. Hence, in this study, we address all the gaps above and propose FinTRec tailored to unifying personalization and ads targeting in FS.

B CUSTOMER JOURNEY

Users interact with FS through multiple channels and products to complete their tasks. As they navigate digital platforms, they are exposed to personalized ads and service messages, delivered via mechanisms such as placement-based targeting or feed-style personalization. However, digital exposure is only part of the journey—users also engage through offline touch-points, including visits to physical locations or interactions with call center agents. An illustrative user journey Figure 1 shows seven touch-points over a month: four via digital platforms, one physical location, one transaction, and one external source (e.g., Google Search). High-value outcomes, such as product acquisitions, are not always attributable to digital ads alone but may result from this broader interaction landscape. It further gets complicated by FS's diverse and interrelated product offerings (Table 1), necessitating accurate capture of contextual information such as channel, page type, placement, and layout. Channels include email, SMS, mobile, and web applications. Ads can appear across hundreds of pages—e.g., the homepage, credit card pages, or other product-specific sections—each with multiple placements (e.g., welcome modules, account details, recent transactions, action buttons, or marketing messages). Interaction data is stored at a granular level across disparate databases, using various schema and formats. Each event is timestamped, enabling alignment along a user's timeline for sequential modeling tasks.