FROM SEQUENCES TO TABLES: SEQUENTIAL RECOMMENDATION WITH TABULAR FOUNDATION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Sequential recommendation aims to predict users' next interactions from their historical behavior. Recent recommendation-specific prior-fitted networks (PFNs), pretrained on synthetic interaction sequences, have achieved strong empirical performance through in-context learning. Meanwhile, the PFN paradigm has also achieved strong results in tabular prediction, inspiring a range of tabular foundation models (TFMs). We investigate whether the general predictive capabilities of these TFMs can support sequential recommendation without recommendation-specific pretraining. To this end, we propose a reusable interface that reformulates next-item prediction as tabular candidate scoring. Our *content-aware feature construction* represents each history–candidate pair using item semantics and explicit short- and long-term relevance signals. These representations can be supplied to different TFM backbones, which condition on labeled historical examples to score and rank candidates while keeping their pretrained parameters fixed. Experiments on five benchmark datasets demonstrate strong recommendation performance, while ablation studies validate the effectiveness of our content-aware feature construction.

1 INTRODUCTION

Sequential recommendation aims to predict a user's next interaction from their interaction history (Kang & McAuley, 2018). It is widely used in online services, including e-commerce (Lv et al., 2019; Chen et al., 2019), video streaming (Dong et al., 2021; Tang et al., 2019), and short-video platforms such as TikTok (Pan et al., 2023; Li et al., 2025), to help users discover relevant products and content through personalized recommendations.

Representative work in this area includes recurrent models such as GRU4Rec (Hidasi et al., 2015) and self-attention models such as SASRec (Kang & McAuley, 2018). These methods typically require training a separate model for each dataset. To enable a single model to be reused across datasets, recent work has explored prior-fitted networks (PFNs) for sequential recommendation. Recommendation-specific PFNs, including SRPFN (Kang et al., 2026) and RecPFN (Tan et al., 2026), are pretrained on synthetic interaction sequences to learn how to predict from contextual examples. At inference time, they achieve strong next-item prediction performance by conditioning on examples from the target dataset, without updating their pretrained parameters.

SRPFN and RecPFN demonstrate the promise of reusable pretrained recommenders, but their pretraining remains specific to recommendation. The same PFN paradigm has also achieved strong results in general tabular prediction. Tabular foundation models (TFMs) make predictions from labeled examples supplied as context, without updating their pretrained parameters. Recent TFMs, including TabPFN (Hollmann et al., 2025), TabFM (Kong & Das, 2026), and LimiX (Zhang et al., 2026), have outperformed established methods such as XGBoost and LightGBM on diverse tabular benchmarks. These advances raise a natural question: can a reusable interface bring the capabilities of general TFMs to sequential recommendation without recommendation-specific pretraining?

Existing recommendation methods provide a foundation for building this interface. Some feature-based recommenders use interaction histories and structured metadata for tabular candidate scoring (Bauer & Jannach, 2024), but these features omit textual information that can reveal additional semantic relationships between items. Meanwhile, text-aware approaches exploit item semantics through predictors trained for recommendation (Hou et al., 2022; Li et al., 2023; Sugiura et al.,

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

2024). However, these approaches still require recommendation-specific predictor training and do not benefit from the strong tabular prediction capabilities of general frozen TFMs.

We therefore propose a reusable tabular interface that combines item semantics with interaction histories for prediction by general frozen TFMs. Specifically, our *content-aware feature construction* converts each history–candidate pair into a tabular row containing content-derived semantic IDs, structured item attributes, candidate–history similarities, and user context. The semantic IDs and attributes describe the candidate and previously consumed items, while the similarity features explicitly express their semantic relationships. We retain similarities to individual recent items as separate features, each corresponding to a fixed history position, so that their predictive relevance can be inferred from contextual examples. A full-history semantic affinity complements these features by representing the candidate's alignment with broader preferences beyond the recent window. By separating feature construction from the prediction backbone, this interface allows the same recommendation representation to be used with different TFMs.

We evaluate the proposed interface on five real-world datasets using three TFM backbones. Our TabPFN-v3.5 variant achieves the best results on Beauty, Toys, and Sports, while our TabFM variant performs best on LastFM. Ablation studies further show that item semantics and both short- and long-term history signals contribute to recommendation performance, supporting the effectiveness of our interface design.

In summary, our contributions are listed as follows:

- We introduce a new perspective on sequential recommendation: reusing general frozen TFMs as prediction backbones through in-context learning, without recommendation-specific pretraining.
- We develop a content-aware feature construction procedure for history–candidate pairs that combines item semantics, structured attributes, and user context with explicit short- and long-term relevance signals, making these relationships directly available for prediction.
- We evaluate our method on five real-world datasets using three TFM backbones, achieving leading recommendation performance on multiple datasets. Ablation studies further validate the effectiveness of our proposed method.

2 Related Works

Sequential Recommendation. Sequential recommendation predicts a user's next interaction from their ordered interaction history. Major approaches include Markov-chain models (Rendle et al., 2010), recurrent networks (Hidasi et al., 2015), convolutional networks (Yuan et al., 2019), graph-based models (Dong et al., 2021), and attention-based models (Kang & McAuley, 2018; Zhou et al., 2020; Liu et al., 2021; Du et al., 2023). More recently, recommendation-specific PFNs have enabled in-context prediction through pretraining on synthetic interactions (Kang et al., 2026; Tan et al., 2026). For broader reviews of sequential recommendation, we refer readers to existing surveys (Boka et al., 2024; Pan et al., 2026).

Our work is closely related to feature construction and semantic representation learning for recommendation. Feature-based approaches represent user and item attributes, interaction statistics, and history–candidate relationships as tabular inputs for tree-based candidate scoring (Lian et al., 2017; Schifferer et al., 2022; Bauer & Jannach, 2024). Text-based approaches learn semantic item representations (Hou et al., 2022; Li et al., 2023; Chen et al., 2024), with recent methods further adapting these representations and incorporating collaborative information (He et al., 2025; Cui et al., 2026; Wei et al., 2026). Content-derived semantic IDs provide discrete item representations for generative recommendation (Rajput et al., 2023; Wang et al., 2024; Fu et al., 2026). Connecting text representations with tabular prediction, existing work also incorporates text embeddings and candidate–history similarities into tabular rankers (Sugiura et al., 2024). Our work builds on these directions through content-aware feature construction, making item information and explicit short- and long-term relevance signals available to general frozen TFMs for in-context candidate scoring.

Tabular Foundation Models. Tabular foundation models aim to provide general-purpose predictors that can transfer across heterogeneous tabular datasets without being trained from scratch for

each new task. A representative line of work is TabPFN (Hollmann et al., 2025), which pretrains Transformer-based models over a large distribution of synthetic tabular tasks and performs prediction on new datasets through a single forward pass. Subsequent studies have extended this paradigm toward larger datasets, richer feature spaces, and more diverse prediction settings (**?**Qu et al., 2026; Kong & Das, 2026; Cho et al., 2026; Zhang et al., 2026; Tao et al., 2026; Team et al., 2026; Eo et al., 2026), establishing tabular foundation models as an alternative to conventional dataset-specific methods such as gradient-boosted trees and neural networks.

3 PRELIMINARIES

3.1 PROBLEM FORMULATION

Let $\mathcal{U}$ and $\mathcal{I}$ denote the user and item sets. Each item $i \in \mathcal{I}$ is associated with textual information $\mathbf{z}_i^{\text{text}}$, such as its title or description, and structured attributes $\mathbf{z}_i^{\text{attr}}$, consisting of numerical or categorical features. For each user $u \in \mathcal{U}$, let $T_u \in \mathbb{N}_+$ denote the number of interactions. The user's chronologically ordered interaction sequence is $\mathcal{S}_u = (i_{u,1}, \ldots, i_{u,T_u})$, where $i_{u,t} \in \mathcal{I}$ for all $t \in \{1, \ldots, T_u\}$.

We hold out the final interaction i_{u,T_u} of each user for testing. The training data consist of the preceding interactions $(i_{u,1}, \ldots, i_{u,T_u-1})$ of all users. The task is to predict the held-out item i_{u,T_u} for each user $u \in \mathcal{U}$ using these training data and the item features.

3.2 TABULAR FOUNDATION MODELS

Tabular foundation models (TFMs) are pretrained across diverse tabular prediction tasks to learn predictive patterns that transfer to new datasets. We focus on TFMs that support in-context learning, which make predictions for a new task by conditioning on labeled examples without updating their pretrained parameters.

Consider a tabular prediction task with N_{train} labeled training samples and N_{test} unlabeled test samples, denoted by $\mathcal{D}_{\text{train}} = \{(\mathbf{x}_n, y_n)\}_{n=1}^{N_{\text{train}}}$ and $\mathcal{D}_{\text{test}} = \{\mathbf{x}_n^{\text{test}}\}_{n=1}^{N_{\text{test}}}$, respectively. The objective is to predict the unknown targets of the test samples. The pretrained TFM receives the labeled training rows together with the unlabeled test rows. The training rows serve as context from which the model infers relationships between features and targets, while the test rows specify the samples to be predicted. We write this prediction process as

$$\widehat{\mathbf{y}}_{\text{test}} = f_\theta(\mathcal{D}_{\text{train}}, \mathcal{D}_{\text{test}}),$$

where $\widehat{\mathbf{y}}_{\text{test}}$ is the predictions for the test samples and θ denotes the pretrained model parameters. Test targets are not supplied to the model, and θ remains fixed throughout inference. For classification tasks, the model provides class probabilities, which can also be used as prediction scores.

4 METHOD

Our framework reformulates sequential recommendation as a tabular prediction task. In Section 4.1, our *content-aware feature construction* represents history–candidate pairs using semantic IDs, structured attributes, and short- and long-term interaction features, yielding labeled training rows and unlabeled test rows. In Section 4.2, we use the labeled training rows as context for a pretrained TFM to score test candidates through in-context learning, without updating its parameters. We then rank the candidates for each user by their predicted scores to obtain next-item recommendations. The overall framework is illustrated in Figure 1.

4.1 CONTENT-AWARE FEATURE CONSTRUCTION

To enable sequential recommendation to benefit from advances in general tabular foundation models, we reformulate next-item prediction as a tabular prediction task.

Specifically, for each user $u \in \mathcal{U}$ and each position $t \in \{2, \ldots, T_u - 1\}$, we construct a prediction task using the observed prefix $(i_{u,1}, \ldots, i_{u,t-1})$ to predict $i_{u,t}$. We pair this prefix with the observed

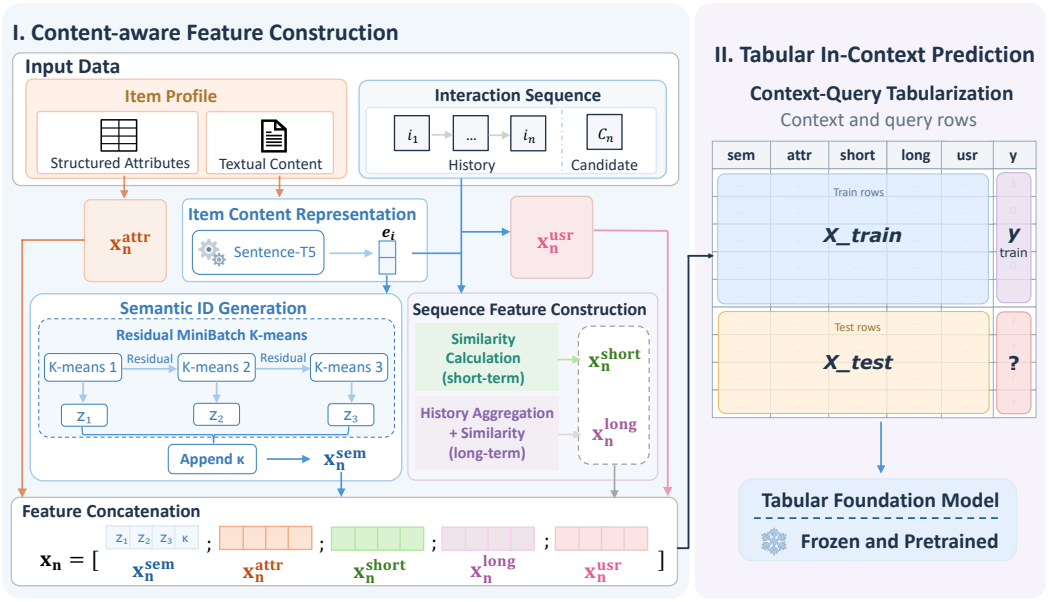

Figure 1: Overview of the proposed framework for sequential recommendation. The left part performs content-aware feature construction, converting each history–candidate pair into a tabular row. The right part performs tabular in-context prediction, using labeled rows as context for a frozen TFM to score and rank test candidates.

next item $i_{u,t}$ to form one positive sample and with K sampled negative items to form K negative samples. Each negative item differs from the prediction target. Thus, each prediction task produces $K+1$ history–candidate pairs.

Collecting these history–candidate pairs across all users and training positions yields a tabular dataset of N_{train} labeled samples, $\mathcal{D}_{\text{train}} = \{(\mathbf{x}_n, y_n)\}_{n=1}^{N_{\text{train}}}$. Here, $\mathbf{x}_n$ represents a history–candidate pair, and y_n equals 1 for a positive sample and 0 for a negative sample. Each feature vector consists of five complementary components:

$$\mathbf{x}_n = [\mathbf{x}_n^{\text{sem}}; \mathbf{x}_n^{\text{attr}}; \mathbf{x}_n^{\text{short}}; \mathbf{x}_n^{\text{long}}; \mathbf{x}_n^{\text{usr}}],$$

where semicolons denote concatenation. The components $\mathbf{x}_n^{\text{sem}}$ and $\mathbf{x}_n^{\text{attr}}$ contain the semantic IDs and structured attributes, respectively, of the candidate item and recent historical items. The components $\mathbf{x}_n^{\text{short}}$ and $\mathbf{x}_n^{\text{long}}$ capture the candidate's relationship with recent interactions and its alignment with long-term user preferences, respectively. Finally, $\mathbf{x}_n^{\text{usr}}$ records the user ID and the index of the corresponding interaction prefix for that user, providing information about user identity and progression through the interaction sequence.

We first describe how item text is encoded into content vectors and semantic IDs. The content vectors serve as intermediate representations and are not included directly in $\mathbf{x}_n$. We then explain how these representations, structured attributes, and interaction histories are used to construct the four tabular feature components.

4.1.1 Item Text Encoding

Item Content Representation. For each item $i \in \mathcal{I}$, we encode its textual information $\mathbf{z}_i^{\text{text}}$ using Sentence-T5 (Ni et al., 2022) and apply ℓ_2 normalization to obtain the content vector $\mathbf{e}_i \in \mathbb{R}^d$, where d is the dimension of the T5 representation. These content vectors serve as intermediate representations for constructing semantic IDs and computing the content similarities used in subsequent tabular features.

Item Semantic ID Construction. Inspired by the semantic ID construction in TIGER (Rajput et al., 2023), we convert each content vector $\mathbf{e}_i$ into a short sequence of integer indices through three

successive rounds of mini-batch K-means. This procedure constructs discrete item representations without training a neural encoder or decoder.

We initialize the residual vector as $\mathbf{r}_i^{(0)} = \mathbf{e}_i \in \mathbb{R}^d$. At each round $\ell \in \{1, 2, 3\}$, we apply mini-batch K-means to the current residual vectors $\{\mathbf{r}_i^{(\ell-1)} : i \in \mathcal{I}\}$, obtaining B cluster centers $\boldsymbol{\mu}_{\ell,b} \in \mathbb{R}^d$, $b \in \{1, \ldots, B\}$. For each item, we record the index $z_{i,\ell}$ of its nearest cluster center and subtract that center from the current residual:

$$z_{i,\ell} = \operatorname*{arg\,min}_{b \in \{1,\ldots,B\}} \|\mathbf{r}_i^{(\ell-1)} - \boldsymbol{\mu}_{\ell,b}\|_2^2,$$
$$\mathbf{r}_i^{(\ell)} = \mathbf{r}_i^{(\ell-1)} - \boldsymbol{\mu}_{\ell,z_{i,\ell}}.$$

The resulting residual is passed to the next clustering round. Thus, the first round clusters the original content vectors, while subsequent rounds cluster the differences left by the preceding approximations. The cluster centers are fitted solely on item content, without interaction sequences or recommendation labels, and remain fixed during recommendation.

Different items may receive the same three cluster indices $(z_{i,1}, z_{i,2}, z_{i,3})$, so this tuple does not necessarily identify an item uniquely. To distinguish such items, we append an additional integer index κ_i. Specifically, within each group of items sharing the same tuple, we sort the items by their internal item IDs and assign distinct indices in that order. The final semantic ID is

$$\mathbf{z}_i^{\text{sem}} = (z_{i,1}, z_{i,2}, z_{i,3}, \kappa_i) \in \mathbb{Z}^4.$$

The first three components describe the item's cluster assignments, while the fourth distinguishes items with identical assignments.

4.1.2 Feature Construction

Recall that each sample $(\mathbf{x}_n, y_n)$ is constructed for a user u at a prediction position t. The feature vector $\mathbf{x}_n$ encodes the observed prefix $(i_{u,1}, \ldots, i_{u,t-1})$ together with a candidate item $c_n \in \mathcal{I}$. The label y_n equals 1 when $c_n = i_{u,t}$ and 0 for a sampled negative item. For this sample, let $L_n = t - 1$ denote the history length and $h_{n,j} = i_{u,t-j}$ denote the j-th most recent historical item, for $j \in \{1, \ldots, L_n\}$.

We use a recent-history window of $J = 6$ interactions for the semantic ID, structured attribute, and short-term interaction features. Whenever fewer than J historical interactions are available, the remaining feature positions are filled with reserved padding values 0. The long-term preference features use the full observed history.

Semantic ID Features. We construct $\mathbf{x}_n^{\text{sem}}$ by concatenating the semantic ID of the candidate item c_n with those of the J most recent historical items in recency order:

$$\mathbf{x}_n^{\text{sem}} = [\mathbf{z}_{c_n}^{\text{sem}}; \mathbf{z}_{h_{n,1}}^{\text{sem}}; \ldots; \mathbf{z}_{h_{n,J}}^{\text{sem}}].$$

Each semantic ID contains four integer components, giving $4 \times (J + 1)$ feature columns.

Structured Attribute Features. We concatenate the structured attributes of the candidate item and the J most recent historical items:

$$\mathbf{x}_n^{\text{attr}} = [\mathbf{z}_{c_n}^{\text{attr}}; \mathbf{z}_{h_{n,1}}^{\text{attr}}; \ldots; \mathbf{z}_{h_{n,J}}^{\text{attr}}].$$

These features provide numerical or categorical information about both the candidate and the user's recent interactions.

Short-Term Interaction Features. We measure the candidate's content similarity to the J most recent historical items using the normalized content vectors:

$$\mathbf{x}_n^{\text{short}} = [\mathbf{e}_{h_{n,1}}^\top \mathbf{e}_{c_n}; \ldots; \mathbf{e}_{h_{n,J}}^\top \mathbf{e}_{c_n}].$$

For unavailable historical positions, we use zero vectors in this expression, yielding zero-padded features. Each remaining entry is a cosine similarity because the item content vectors are ℓ_2-normalized. This feature block captures the candidate's relationship with recent interactions in recency order.

Long-Term Preference Features. To represent preferences across the full observed history, let $\mathcal{V}_n = \{h_{n,j} : j = 1, \ldots, L_n\}$ be the set of distinct historical items, and let $m_n = |\mathcal{V}_n|$. We average their content vectors to obtain a user preference representation and construct

$$\bar{\mathbf{e}}_n = \frac{1}{m_n} \sum_{i \in \mathcal{V}_n} \mathbf{e}_i,$$

$$\mathbf{x}_n^{\text{long}} = [\bar{\mathbf{e}}_n^\top \mathbf{e}_{c_n}; \log(1 + m_n)].$$

The first component measures the candidate's average content similarity to the distinct items in the user's observed history. The second records the number of distinct historical items on a logarithmic scale.

User Features. For each sample n, let u_n denote the user identifier and p_n the position of the target interaction within that user's retained interaction sequence, with $p_n = 1$ corresponding to the first interaction. We construct

$$\mathbf{x}_n^{\text{usr}} = [u_n; p_n].$$

These features encode user identity and the position at which the prediction is made. All candidates paired with the same observed history share these two feature values

4.2 Next-Item Prediction via Tabular Foundation Models

At test time, we pair each user's observed prefix $(i_{u,1}, \ldots, i_{u,T_u-1})$ with each candidate item. Each history–candidate pair is converted into a feature vector using the construction in Section 4.1.2, with prediction position $t = T_u$. Collecting these feature vectors across all users and their candidates yields the unlabeled test dataset $\mathcal{D}_{\text{test}} = \{\mathbf{x}_n^{\text{test}}\}_{n=1}^{N_{\text{test}}}$, where N_{test} is the total number of test history–candidate pairs.

Following the in-context learning setup in Section 3.2, we provide the pretrained TFM with $\mathcal{D}_{\text{train}}$ as labeled context and $\mathcal{D}_{\text{test}}$ as unlabeled query rows. The model keeps its pretrained parameters fixed and assigns each test row a positive-class probability, which serves as the score for the corresponding candidate. For each user, we rank the candidates in descending order of these scores and select the highest-scoring candidate as the predicted next item.

5 Experiments

In this section, we evaluate pretrained TFMs for next-item prediction through our proposed tabular interface. We first describe the experimental setup in Section 5.1 and compare overall performance in Section 5.2. We then examine our content-aware feature construction through ablation studies in Section 5.3 and comparisons with alternative feature representations in Section 5.4. Finally, we analyze performance across item popularity levels in Section 5.5.

5.1 Experimental Setup

Datasets. We evaluate on five real-world sequential recommendation datasets drawn from three sources: **Beauty, Sports, and Toys** are derived from the Amazon Reviews 2014 data[1] and represent e-commerce product recommendation in the "Beauty", "Sports and Outdoors", and "Toys and Games" categories, respectively. **LastFM**[2] is a music recommendation dataset containing user–artist interactions, with longer sequences and denser interactions than the other datasets. **Yelp**[3] is a local-business recommendation dataset covering user interactions with restaurants and other businesses.

Following common preprocessing protocols in sequential recommendation (Kang & McAuley, 2018; Zhou et al., 2020), we iteratively apply 5-core filtering to retain users and items with at least five interactions. We treat all observed interactions as implicit feedback and order each user's interaction sequence chronologically. The statistics of the processed datasets and the dataset-specific attributes and textual inputs are provided in Section A.1.

[1] https://cseweb.ucsd.edu/~jmcauley/datasets/amazon/links.html

[2] https://grouplens.org/datasets/hetrec-2011/

[3] https://business.yelp.com/data/resources/open-dataset/

Table 1: Overall performance on five datasets. XGB, LGBM, TabICL, TabFM, and TabPFN denote XGBoost-SR, LGBM-SR, TabICL-v2-SR, TabFM-SR, and TabPFN-v3.5-SR, respectively. XGBoost-SR and LGBM-SR use the same feature construction and labeled training rows as the TFMs. Bold and underlined values indicate the best and second-best results in each row, respectively. Among all the baselines that do not use our constructed content-aware features, † marks the best result in each row. For the three TFMs, red upward arrows (↑) indicate scores above the dagger-marked baseline, while green downward arrows (↓) indicate scores at or below it.

		Recommendation baselines							Tree-based models		TFMs		
Dataset	Metric	ItemKNN	GRU4Rec	SASRec	CoSeRec	RecFormer	UniSRec	SRPFN	XGB	LGBM	TabICL	TabFM	TabPFN
Beauty	HR@5	0.2206	0.3572	0.3753	0.3861	0.4092	0.4557†	0.4414	0.4805	0.4781	0.4714↑	**0.5125**↑	0.5110↑
	HR@10	0.2734	0.4526	0.4659	0.4827	0.5108	0.5722†	0.5408	0.5936	0.5915	0.5807↑	**0.6193**↑	0.6181↑
	NDCG@5	0.1734	0.2761	0.2856	0.2921	0.3134	0.3351	0.3457†	0.3638	0.3624	0.3622↑	**0.3969**↑	0.3956↑
	NDCG@10	0.1904	0.3068	0.3147	0.3234	0.3462	0.3729	0.3778†	0.4003	0.3991	0.3975↑	**0.4313**↑	0.4302↑
Toys	HR@5	0.2012	0.3637	0.3631	0.3677	0.3888	0.4522†	0.4319	0.4897	0.4865	0.4796↑	0.5197↑	**0.5198**↑
	HR@10	0.2530	0.4638	0.4597	0.4674	0.4810	0.5683†	0.5258	0.6016	0.5990	0.5890↑	**0.6288**↑	0.6261↑
	NDCG@5	0.1620	0.2765	0.2724	0.2825	0.3077	0.3414†	0.3405	0.3815	0.3803	0.3778↑	**0.4140**↑	0.4124↑
	NDCG@10	0.1786	0.3088	0.3036	0.3146	0.3375	0.3790†	0.3708	0.4176	0.4166	0.4131↑	**0.4493**↑	0.4468↑
Sports	HR@5	0.1709	0.3504	0.3429	0.3634	0.3935	0.4386†	0.4254	0.4469	0.4460	0.4282↓	0.4798↑	**0.4819**↑
	HR@10	0.2217	0.4738	0.4617	0.4770	0.5296	0.5758†	0.5484	0.5830	0.5802	0.5576↓	0.6129↑	**0.6141**↑
	NDCG@5	0.1284	0.2530	0.2455	0.2667	0.2835	0.3189	0.3192†	0.3272	0.3278	0.3163↓	0.3583↑	**0.3622**↑
	NDCG@10	0.1447	0.2929	0.2838	0.3033	0.3273	0.3632†	0.3589	0.3711	0.3712	0.3581↓	0.4013↑	**0.4049**↑
LastFM	HR@5	0.2697	0.2706	0.3064	0.2083	0.3523	0.4018†	0.3284	0.3716	0.3780	0.3927↓	**0.4165**↑	0.4073↑
	HR@10	0.3716	0.3991	0.4468	0.3165	0.5073	**0.5706**†	0.4716	0.5303	0.5349	0.5486↓	0.5587↓	0.5422↓
	NDCG@5	0.1854	0.1868	0.2096	0.1440	0.2442	0.2757†	0.2303	0.2529	0.2632	0.2786↑	**0.2961**↑	0.2930↑
	NDCG@10	0.2181	0.2286	0.2546	0.1790	0.2941	0.3305†	0.2766	0.3042	0.3140	0.3286↓	**0.3417**↑	0.3368↑
Yelp	HR@5	0.2703	0.5663	0.5815	0.5967	0.6309	**0.6661**†	0.6301	0.5838	0.5910	0.5581↓	0.5864↓	0.6044↓
	HR@10	0.3265	0.7391	0.7535	0.7577	0.7909	**0.8247**†	0.7889	0.7564	0.7657	0.7265↓	0.7521↓	0.7723↓
	NDCG@5	0.2025	0.4021	0.4122	0.4384	0.4575	**0.4906**†	0.4632	0.4197	0.4265	0.4027↓	0.4279↓	0.4445↓
	NDCG@10	0.2207	0.4581	0.4679	0.4907	0.5095	**0.5422**†	0.5148	0.4757	0.4831	0.4572↓	0.4816↓	0.4989↓

Evaluation Protocol and Metrics. For evaluation, each user's most recent interaction serves as the test target under the standard leave-one-out protocol (Kang & McAuley, 2018; Du et al., 2023). For all methods, the second-to-last interaction is used for validation, while all preceding interactions are used for training.

We further follow the commonly adopted sampled-ranking setting (Kang & McAuley, 2018; Kang et al., 2026), where each test target is ranked against 99 randomly sampled negative items. To ensure a consistent comparison, all methods are evaluated using exactly the same test targets and negative candidates. We report Hit Ratio (HR) and Normalized Discounted Cumulative Gain (NDCG) at $k \in \{5, 10\}$. HR@k measures whether the ground-truth item appears in the top-k ranked candidates, while NDCG@k accounts for its ranking position. All metrics are averaged over test users.

Baselines. We implement our framework with three TFMs: TabPFN-v3.5 (Jäger et al., 2026), TabICL-v2 (Qu et al., 2026), and TabFM (Kong & Das, 2026), yielding **TabPFN-v3.5-SR**, **TabICL-v2-SR**, and **TabFM-SR**, respectively. We compare with the statistical baseline **ItemKNN** (Deshpande & Karypis, 2004) and the sequential recommendation models **GRU4Rec** (Hidasi et al., 2015), **SASRec** (Kang & McAuley, 2018), **CoSeRec** (Liu et al., 2021), **UniSRec** (Hou et al., 2022), **RecFormer** (Li et al., 2023), and **SRPFN** (Kang et al., 2026). We also apply the same content-aware feature construction to the tree-based models XGBoost (Chen & Guestrin, 2016) and LightGBM (Ke et al., 2017), yielding **XGBoost-SR** and **LGBM-SR**. The suffix **-SR** denotes the application to sequential recommendation through our tabular representation. Further details on model versions, configurations, and implementations are provided in Section A.2.

5.2 Overall Performance

Table 1 summarizes the results of all methods. TabPFN-v3.5-SR and TabFM-SR outperform the strongest recommendation baseline in most evaluated settings, including all metrics on Beauty, Toys, and Sports. TabFM-SR achieves the best results on Beauty and most metrics on Toys and LastFM, while TabPFN-v3.5-SR leads on Sports. TabICL-v2-SR delivers less consistent gains and generally trails the other two TFM variants. Their advantage over TabICL-v2-SR broadly agrees with their backbones' stronger performance on TabArena (Erickson et al., 2025), suggesting that the predictive capabilities of the backbone matter for recommendation as well. These results indicate that our in-

Table 2: Ablation study of TabPFN-v3.5-SR on Beauty, Sports, and LastFM. $\Delta\%$ denotes the relative change in NDCG@10 compared with the full model. The lower two rows remove multiple input groups jointly. The highest NDCG@10 for each dataset is shown in bold.

	Beauty		Sports		LastFM	
	NDCG@10	$\Delta\%$	NDCG@10	$\Delta\%$	NDCG@10	$\Delta\%$
Full model	**0.4302**	-	**0.4049**	-	**0.3368**	-
w/o SID ($\mathbf{x}_n^{\text{sem}}$)	0.4275	-0.63	0.3992	-1.41	0.3361	-0.18
w/o Short ($\mathbf{x}_n^{\text{short}}$)	0.4062	-5.58	0.3840	-5.17	0.3195	-5.12
w/o Long ($\mathbf{x}_n^{\text{long}}$)	0.4265	-0.85	0.3973	-1.88	0.3177	-5.65
w/o Attributes ($\mathbf{x}_n^{\text{attr}}$)	0.4221	-1.88	0.3930	-2.94	0.3296	-2.14
w/o SID + Short	0.4039	-6.11	0.3808	-5.95	0.3220	-4.38
w/o SID + Short + Long	0.3316	-22.91	0.3135	-22.57	0.1964	-41.69

terface can translate the capabilities of strong general TFMs into recommendation performance that surpasses dedicated recommendation models in most evaluated settings, without recommendation-specific pretraining or target-domain parameter updates.

On Yelp, UniSRec achieves higher scores than TabPFN-v3.5-SR. One possible factor is that the limited and heterogeneous item content may make it difficult to construct informative semantic IDs. In the dataset, more than 50% of businesses share the same second-level codeword, suggesting limited semantic differentiation at this level. This concentration may weaken the representation of fine-grained preferences over businesses and partly contribute to the performance gap.

Using the same content-aware feature construction and labeled training rows as our TFM variants, XGBoost-SR and LGBM-SR also perform competitively on the three Amazon datasets. However, TabPFN-v3.5-SR consistently outperforms both tree-based models across all five datasets, suggesting an additional predictive benefit from using a pretrained TFM with the same tabular inputs. We examine the contributions of individual feature groups in the subsequent ablation study.

5.3 Ablation Study

We conduct ablation studies to assess the contributions of item representations and candidate–history relevance signals in our content-aware feature construction. Using TabPFN-v3.5-SR, we evaluate six variants on Beauty, Sports, and LastFM. These variants remove semantic ID features $\mathbf{x}_n^{\text{sem}}$ (SID), structured attribute features $\mathbf{x}_n^{\text{attr}}$ (Attributes), short-term interaction features $\mathbf{x}_n^{\text{short}}$ (Short), or long-term preference features $\mathbf{x}_n^{\text{long}}$ (Long), individually or in combination. The results are shown in Table 2.

Among the single-group ablations, removing short-term interaction features causes the largest relative decrease in NDCG@10 on Beauty and Sports, at 5.58% and 5.16%, respectively. On LastFM, removing long-term preference features has the largest effect, reducing NDCG@10 by 5.67%. This difference is consistent with the sequence statistics: LastFM has an average sequence length of 48.2, compared with 8.9 for Beauty and 8.3 for Sports. Longer sequences may increase the value of summarizing preferences beyond recent interactions.

Jointly removing semantic ID, short-term interaction, and long-term preference features reduces NDCG@10 by 22.92%, 22.57%, and 41.69% on Beauty, Sports, and LastFM, respectively. Removing long-term preference features also causes a larger decrease when semantic ID and short-term interaction features are already absent. These patterns suggest that different feature groups provide partially complementary information, allowing the model to tolerate the absence of a single group to some extent. Detailed ablation results for individual datasets are provided in Appendix A.3.

5.4 Comparison of Feature Representations

We compare our content-aware feature construction in Section 4.1 with alternative feature construction pipelines (Schifferer et al., 2022; Sugiura et al., 2024; Bauer & Jannach, 2024) adapted from previous recommendation methods. All configurations are evaluated using the same data partitions,

Table 3: Comparison of TabPFN-v3.5 with different feature-engineering methods on Beauty, Sports, and LastFM. The best result in each column is shown in bold.

Feature set	Beauty		Sports		LastFM	
	HR@10	NDCG@10	HR@10	NDCG@10	HR@10	NDCG@10
Schifferer et al. (2022)	0.5113	0.3550	0.5130	0.3299	0.5275	0.3288
Sugiura et al. (2024)	0.4992	0.3298	0.5000	0.3065	0.4881	0.2857
Bauer & Jannach (2024)	0.5161	0.3590	0.5324	0.3368	0.5349	0.3311
Ours	**0.6181**	**0.4302**	**0.6141**	**0.4049**	**0.5422**	**0.3368**

Table 4: NDCG@10 across item popularity groups on five datasets. Head, Mid, and Tail denote the most frequent 20%, middle 60%, and least frequent 20% of items, respectively.

Group	Method	Beauty	Toys	Sports	LastFM	Yelp
Head	SRPFN	0.5669	0.5180	0.5624	0.3883	0.6649
	UniSRec	0.5500	0.5058	0.5615	0.4272	**0.6927**
	TabPFN-v3.5-SR	**0.5826**	**0.5620**	**0.5680**	**0.4325**	0.6362
Mid	SRPFN	0.2469	0.2843	0.2241	0.2173	0.4005
	UniSRec	0.2602	0.3209	0.2469	**0.2697**	**0.4359**
	TabPFN-v3.5-SR	**0.3174**	**0.3780**	**0.2859**	0.2692	0.3893
Tail	SRPFN	0.1661	0.2444	0.1308	0.1491	0.2726
	UniSRec	0.1549	0.2417	0.1123	0.2393	0.2742
	TabPFN-v3.5-SR	**0.2748**	**0.3497**	**0.2426**	**0.2614**	**0.2928**

candidate sets, and frozen TabPFN-v3.5 backbone. Table 3 reports the results on Beauty, Sports, and LastFM, where Ours denotes our content-aware feature construction.

Our representation achieves the highest HR@10 and NDCG@10 on Beauty and Sports. On LastFM, it achieves the highest HR@10, while HySAR yields a slightly higher NDCG@10. These results support the effectiveness of our content-aware feature construction for sequential recommendation with a frozen TFM.

5.5 Popularity Bias Analysis

To examine performance across item popularity levels, we count item interactions after excluding each user's test interaction. Following SRPFN (Kang et al., 2026), we rank items by decreasing frequency and divide them into Head, Mid, and Tail groups containing 20%, 60%, and 20% of the items, respectively. Each test query is assigned to the group containing its ground-truth item. Table 4 reports NDCG@10 for SRPFN, UniSRec, and TabPFN-v3.5-SR within each group.

TabPFN-v3.5-SR achieves the highest Tail NDCG@10 among the compared methods on all five datasets. On Beauty, Toys, and Sports, its performance advantage over the stronger baseline is more pronounced in the Tail group than in the Head group. On Yelp, it outperforms both baselines on Tail items, although UniSRec performs better in the Head and Mid groups. These results support the effectiveness of our interface for recommending less frequently interacted items.

6 Conclusions

We presented a reusable tabular interface that enables general frozen TFMs to perform sequential recommendation without recommendation-specific pretraining. Our content-aware feature construction combines item semantics, structured attributes, and user context with explicit short- and long-term relevance signals. Experiments on five datasets using three TFM backbones demonstrate strong performance, while ablation studies support the effectiveness of our representation design.

AI USE STATEMENT

In this work, we used generative AI tools to assist with reviewing experimental code pipelines for implementation consistency and potential issues, and with scheduling and orchestrating parallel experiments across multiple GPUs to improve execution efficiency. We also used generative AI tools during manuscript preparation to check formatting, spelling, and the consistency of mathematical notation. All AI-assisted outputs were manually reviewed and verified by the authors. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REFERENCES

Josef Bauer and Dietmar Jannach. Hybrid session-aware recommendation with feature-based models: J. bauer, d. jannach. *User Modeling and User-Adapted Interaction*, 34(3):691–728, 2024.

Tesfaye Fenta Boka, Zhendong Niu, and Rama Bastola Neupane. A survey of sequential recommendation systems: Techniques, evaluation, and future directions. *Information Systems*, 125:102427, 2024.

Junyi Chen, Lu Chi, Bingyue Peng, and Zehuan Yuan. Hllm: Enhancing sequential recommendations via hierarchical large language models for item and user modeling. *arXiv preprint arXiv:2409.12740*, 2024.

Qiwei Chen, Huan Zhao, Wei Li, Pipei Huang, and Wenwu Ou. 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*, pp. 1–4, 2019.

Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pp. 785–794, 2016.

Minyong Cho, Minho Jeong, Dooho Lee, Jinmo Lee, and Jaemin Yoo. Causilo technical report. *arXiv preprint arXiv:2609.22866*, 2026.

Yu Cui, Feng Liu, Zhaoxiang Wang, Changwang Zhang, Jun Wang, Can Wang, and Jiawei Chen. Spectran: Spectral-aware transformer-based adapter for llm-enhanced sequential recommendation. *arXiv preprint arXiv:2601.21986*, 2026.

Mukund Deshpande and George Karypis. Item-based top-n recommendation algorithms. *ACM Transactions on Information Systems (TOIS)*, 22(1):143–177, 2004.

Xinzhou Dong, Beihong Jin, Wei Zhuo, Beibei Li, and Taofeng Xue. Improving sequential recommendation with attribute-augmented graph neural networks. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp. 373–385. Springer, 2021.

Xinyu Du, Huanhuan Yuan, Pengpeng Zhao, Jianfeng Qu, Fuzhen Zhuang, Guanfeng Liu, Yanchi Liu, and Victor S Sheng. Frequency enhanced hybrid attention network for sequential recommendation. In *Proceedings of the 46th international ACM SIGIR conference on research and development in information retrieval*, pp. 78–88, 2023.

Moonjung Eo, Min-Kook Suh, Hye-Seung Cho, Jiwon Kim, Seoyoon Kim, Sangjun Nam, and Soonyoung Lee. Exaone tabular 1.0: Technical report. *arXiv preprint arXiv:2608.25774*, 2026.

Nick Erickson, Lennart Purucker, Andrej Tschalzev, David Holzmüller, Prateek Desai, David Salinas, and Frank Hutter. Tabarena: A living benchmark for machine learning on tabular data. *Advances in Neural Information Processing Systems*, 38, 2025.

Junchen Fu, Xuri Ge, Alexandros Karatzoglou, Ioannis Arapakis, Suzan Verberne, Joemon M Jose, and Zhaochun Ren. Differentiable semantic id for generative recommendation. *arXiv preprint arXiv:2601.19711*, 2026.

Yingzhi He, Xiaohao Liu, An Zhang, Yunshan Ma, and Tat-Seng Chua. Llm2rec: Large language models are powerful embedding models for sequential recommendation. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pp. 896–907, 2025.

Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939*, 2015.

Noah Hollmann, Samuel Müller, Lennart Purucker, Arjun Krishnakumar, Max Körfer, Shi Bin Hoo, Robin Tibor Schirrmeister, and Frank Hutter. Accurate predictions on small data with a tabular foundation model. *Nature*, 637(8045):319–326, 2025.

Yupeng Hou, Shanlei Mu, Wayne Xin Zhao, Yaliang Li, Bolin Ding, and Ji-Rong Wen. Towards universal sequence representation learning for recommender systems. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*, pp. 585–593, 2022.

Benjamin Jäger, Nick Erickson, Léo Grinsztajn, Felix Birkel, Klemens Flöge, Oscar Key, Kürşat Kaya, Jonas Kübler, Adèle Frankel, Tobias Schröder, et al. Tabpfn-3.5: Technical report. *arXiv preprint arXiv:2609.17895*, 2026.

Wang-Cheng Kang and Julian McAuley. Self-attentive sequential recommendation. In *2018 IEEE international conference on data mining (ICDM)*, pp. 197–206. IEEE, 2018.

Woosung Kang, Jiwon Jeong, Jonghyeok Shin, Jeongwhan Choi, and Noseong Park. One sequential recommendation model pretrained from synthetic priors predicts multiple datasets. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pp. 2273–2284, 2026.

Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30, 2017.

Weihao Kong and Abhimanyu Das. Introducing tabfm: a zero-shot foundation model for tabular data. *Google Research Blog*, 2026.

Jiacheng Li, Ming Wang, Jin Li, Jinmiao Fu, Xin Shen, Jingbo Shang, and Julian McAuley. Text is all you need: Learning language representations for sequential recommendation. In *Proceedings of the 29th ACM SIGKDD conference on knowledge discovery and data mining*, pp. 1258–1267, 2023.

Wuchao Li, Rui Huang, Haijun Zhao, Chi Liu, Kai Zheng, Qi Liu, Na Mou, Guorui Zhou, Defu Lian, Yang Song, et al. Dimerec: a unified framework for enhanced sequential recommendation via generative diffusion models. In *Proceedings of the Eighteenth ACM International Conference on Web Search and Data Mining*, pp. 726–734, 2025.

Jianxun Lian, Fuzheng Zhang, Min Hou, Hongwei Wang, Xing Xie, and Guangzhong Sun. Practical lessons for job recommendations in the cold-start scenario. In *Proceedings of the recommender systems challenge 2017*, pp. 1–6. 2017.

Zhiwei Liu, Yongjun Chen, Jia Li, Philip S Yu, Julian McAuley, and Caiming Xiong. Contrastive self-supervised sequential recommendation with robust augmentation. *arXiv preprint arXiv:2108.06479*, 2021.

Fuyu Lv, Taiwei Jin, Changlong Yu, Fei Sun, Quan Lin, Keping Yang, and Wilfred Ng. Sdm: Sequential deep matching model for online large-scale recommender system. In *Proceedings of the 28th ACM international conference on information and knowledge management*, pp. 2635–2643, 2019.

Jianmo Ni, Gustavo Hernandez Abrego, Noah Constant, Ji Ma, Keith Hall, Daniel Cer, and Yinfei Yang. Sentence-t5: Scalable sentence encoders from pre-trained text-to-text models. In *Findings of the association for computational linguistics: ACL 2022*, pp. 1864–1874, 2022.

Li-Wei Pan, Wei-Ke Pan, Mei-Yan Wei, Hong-Zhi Yin, and Zhong Ming. A survey on sequential recommendation. *Frontiers of Computer Science*, 20(3):2003606, 2026.

Yunzhu Pan, Chen Gao, Jianxin Chang, Yanan Niu, Yang Song, Kun Gai, Depeng Jin, and Yong Li. Understanding and modeling passive-negative feedback for short-video sequential recommendation. In *Proceedings of the 17th ACM conference on recommender systems*, pp. 540–550, 2023.

Jingang Qu, David Holzmüller, Gaël Varoquaux, and Marine Le Morvan. TabICLv2: A better, faster, scalable, and open tabular foundation model. *arXiv preprint arXiv:2602.11139*, 2026.

Shashank Rajput, Nikhil Mehta, Anima Singh, Raghunandan Hulikal Keshavan, Trung Vu, Lukasz Heldt, Lichan Hong, Yi Tay, Vinh Tran, Jonah Samost, et al. Recommender systems with generative retrieval. *Advances in Neural Information Processing Systems*, 36:10299–10315, 2023.

Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. Factorizing personalized markov chains for next-basket recommendation. In *Proceedings of the 19th international conference on World wide web*, pp. 811–820, 2010.

Benedikt Schifferer, Jiwei Liu, Sara Rabhi, Gilberto Titericz, Chris Deotte, Gabriel De Souza P. Moreira, Ronay Ak, and Kazuki Onodera. A diverse models ensemble for fashion session-based recommendation. In *Proceedings of the Recommender Systems Challenge 2022*, pp. 10–17. 2022.

Tetsuro Sugiura, Yosuke Yamagishi, and Yodai Kishimoto. Leveraging lightgbm ranker for efficient large-scale news recommendation systems. In *Proceedings of the Recommender Systems Challenge 2024*, pp. 27–31. 2024.

En Zhi Tan, Jia Xiang Lim, Bryan Lijie Chew, Tze Minh Ng, and Benjamin Yan Han Yap. Recpfn: Prior-fitted networks for in-context-based recommendations. In *Proceedings of the 49th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1731–1742, 2026.

Jiaxi Tang, Francois Belletti, Sagar Jain, Minmin Chen, Alex Beutel, Can Xu, and Ed H. Chi. Towards neural mixture recommender for long range dependent user sequences. In *The World Wide Web Conference*, pp. 1782–1793, 2019.

Yefan Tao, Xiyuan Zhang, Xinyi Liu, Boran Han, Danielle Maddix, Haoyang Fang, Zhen Han, Jiading Gai, Xuanqing Liu, Michael Bohlke-Schneider, et al. Mitra-v2 technical report. *arXiv preprint arXiv:2609.04540*, 2026.

TabLDM Team, Penghui Wang, Wei Liu, Hong Wang, Chengyue Huang, Yuxi Sun, Zirui Wang, Hongming Huang, Quan Wang, Chunxiao Liu, et al. Xiaomi-tabldm: A tabular foundation model technical report. *arXiv preprint arXiv:2609.03880*, 2026.

Wenjie Wang, Honghui Bao, Xinyu Lin, Jizhi Zhang, Yongqi Li, Fuli Feng, See-Kiong Ng, and Tat-Seng Chua. Learnable item tokenization for generative recommendation. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, pp. 2400–2409, 2024.

Zhifu Wei, Yizhou Dang, Guibing Guo, Chuang Zhao, and Zhu Sun. Fusion and alignment enhancement with large language models for tail-item sequential recommendation. *arXiv preprint arXiv:2604.03688*, 2026.

Fajie Yuan, Alexandros Karatzoglou, Ioannis Arapakis, Joemon M Jose, and Xiangnan He. A simple convolutional generative network for next item recommendation. In *Proceedings of the twelfth ACM international conference on web search and data mining*, pp. 582–590, 2019.

Xingxuan Zhang, Gang Ren, Hao Yuan, Hao Zou, Hongze Tan, Hui Wang, Jianhao Song, Jiansheng Li, Jiayao Zhang, Jinghan Zhang, et al. Limix-2: A contextual mechanism network towards general structured-data intelligence. *arXiv preprint arXiv:2609.17488*, 2026.

Kun Zhou, Hui Wang, Wayne Xin Zhao, Yutao Zhu, Sirui Wang, Fuzheng Zhang, Zhongyuan Wang, and Ji-Rong Wen. S3-rec: Self-supervised learning for sequential recommendation with mutual information maximization. In *Proceedings of the 29th ACM international conference on information & knowledge management*, pp. 1893–1902, 2020.

A Additional Experimental Details

A.1 Dataset Statistics

Table 5 summarizes the statistics of the processed datasets after iterative 5-core filtering, including the numbers of users, items, and interactions, together with their sequence statistics.

Table 6 summarizes the item attributes used for feature construction and the textual content provided to Sentence-T5 for item content representation. All fields are obtained directly from the original dataset metadata.

Table 5: Statistics of the datasets after preprocessing.

Dataset	Users	Items	Interactions	Avg. length	Sparsity
Beauty	22,363	12,101	198,502	8.9	99.93%
Sports	35,598	18,357	296,337	8.3	99.95%
Toys	19,412	11,924	167,597	8.6	99.93%
LastFM	1,090	3,646	52,551	48.2	98.68%
Yelp	30,431	20,033	316,354	10.4	99.95%

Table 6: Item attributes and Content input fields used for each dataset.

Dataset	Item Attributes	Content Input
Beauty	`main_category_id`, `log_price`, `log_sales_rank`	title, categories, description, salesRank
Toys	`category_level_2_id`, `log_price`, `log_sales_rank`	title, categories, description, salesRank
Sports	`category_level_2_id`, `log_price`, `log_sales_rank`	title, categories, description, salesRank
Yelp	`primary_category_id`, `city_id`, `stars`, `log_review_count`	name, categories, city, state, address
LastFM	`primary_tag_id`, `filtered_tag_count`	artist_name, top5_tags, primary_tag

A.2 Baseline Implementations

We summarize the main characteristics of the recommendation baselines, tree-based models, and tabular foundation models evaluated in our experiments. The public implementations used in our experiments are summarized in Table 7.

Table 7: Implementations used for the evaluated models.

Model	**Repository / Implementation**
ItemKNN	`https://github.com/RUCAIBox/RecBole`
GRU4Rec	`https://github.com/hidasib/GRU4Rec_PyTorch_Official`
SASRec	`https://github.com/kang205/SASRec`
CoSeRec	`https://github.com/YChen1993/CoSeRec`
UniSRec	`https://github.com/RUCAIBox/UniSRec`
RecFormer	`https://github.com/AaronHeee/RecFormer`
SRPFN	`https://github.com/wooooosung/SRPFN`
XGBoost	`https://github.com/dmlc/xgboost`
LightGBM	`https://github.com/lightgbm-org/LightGBM`
TabPFN-v3.5	`https://github.com/PriorLabs/TabPFN`
TabICL-v2	`https://github.com/soda-inria/tabicl`
TabFM	`https://github.com/google-research/tabfm`

All methods use the same test targets, candidate sets, and evaluation protocol. The tree-based baselines and tabular foundation models use candidate-row tables with a recent-history window of $J = 6$, where each row contains at most six recent items and one candidate. Each observed next-item event forms one positive row paired with one sampled negative row ($K = 1$). We use a frozen Sentence-T5-XXL (Ni et al., 2022) encoder to encode item text. Tabular foundation models treat the labeled

rows as context and test-candidate rows as queries, ranking candidates by their positive-class probabilities. The tree-based baselines use the same row representation generated by our content-aware feature construction.

The tabular foundation models use their released pretrained weights without target-domain parameter updates. Experiments are conducted primarily on two NVIDIA A800 GPUs with 80 GB of memory each.

Recommendation baselines.

- **ItemKNN** (Deshpande & Karypis, 2004) estimates candidate relevance through item–item similarities derived from observed user–item interactions.
- **GRU4Rec** (Hidasi et al., 2015) uses gated recurrent units to update a hidden representation as interactions arrive and predict the subsequent item.
- **SASRec** (Kang & McAuley, 2018) combines positional embeddings with causal self-attention to identify relevant preceding interactions for next-item prediction.
- **CoSeRec** (Liu et al., 2021) supplements next-item prediction with contrastive learning between sequence views constructed using correlations among items.
- **UniSRec** (Hou et al., 2022) transforms pretrained item-text embeddings through a mixture-of-experts adaptor and uses contrastive pretraining to learn sequence representations that transfer across recommendation domains.
- **RecFormer** (Li et al., 2023) converts item attributes into text sequences and uses a bidirectional Transformer to encode interaction histories for next-item retrieval.
- **SRPFN** (Kang et al., 2026) learns from synthetic interaction sequences during pretraining and conditions on target-domain item-transition examples to predict subsequent items without parameter updates.

Tree-based models. Both models use the same candidate-row representation and features as our framework.

- **XGBoost** (Chen & Guestrin, 2016) fits an additive ensemble of decision trees using a regularized objective and first- and second-order gradient information.
- **LightGBM** (Ke et al., 2017) combines histogram-based split search with leaf-wise tree growth to efficiently learn nonlinear relationships between tabular features and prediction targets.

Tabular foundation models. We use the following models as pretrained backbones, keeping their parameters fixed during recommendation.

- **TabPFN-v3.5** (Jäger et al., 2026) is pretrained on synthetic tabular tasks to infer predictions for query rows by conditioning on labeled examples.
- **TabICL-v2** (Qu et al., 2026) first embeds columns and aggregates their features into row representations, then performs in-context prediction across labeled and query rows.
- **TabFM** (Kong & Das, 2026) alternates attention across rows and columns before compressing each row into a vector for a subsequent in-context prediction module.

A.3 Ablation Study

We report detailed ablation results of TabPFN-v3.5-SR for each dataset.

Table 8: Ablation results of TabPFN-v3.5-SR on Beauty. $\Delta\%$ denotes the relative change from the full model. The best result for each metric is shown in bold.

Variant	HR@5	$\Delta\%$	HR@10	$\Delta\%$	NDCG@5	$\Delta\%$	NDCG@10	$\Delta\%$
Full model	**0.5110**	–	**0.6181**	–	**0.3956**	–	**0.4302**	–
w/o SID ($\mathbf{x}_n^{\text{sem}}$)	0.5073	-0.73	0.6161	-0.32	0.3923	-0.83	0.4275	-0.63
w/o Short ($\mathbf{x}_n^{\text{short}}$)	0.4828	-5.52	0.5979	-3.26	0.3690	-6.72	0.4062	-5.58
w/o Long ($\mathbf{x}_n^{\text{long}}$)	0.5043	-1.32	0.6123	-0.93	0.3916	-1.02	0.4265	-0.85
w/o Attributes ($\mathbf{x}_n^{\text{attr}}$)	0.4999	-2.18	0.6055	-2.03	0.3879	-1.95	0.4221	-1.88
w/o SID + Short	0.4810	-5.88	0.5973	-3.36	0.3664	-7.39	0.4039	-6.11
w/o SID + Short + Long	0.3962	-22.47	0.5129	-17.01	0.2940	-25.68	0.3316	-22.91

Table 9: Ablation results of TabPFN-v3.5-SR on Sports. $\Delta\%$ denotes the relative change from the full model. The best result for each metric is shown in bold.

Variant	HR@5	$\Delta\%$	HR@10	$\Delta\%$	NDCG@5	$\Delta\%$	NDCG@10	$\Delta\%$
Full model	**0.4819**	–	**0.6141**	–	**0.3622**	–	**0.4049**	–
w/o SID ($\mathbf{x}_n^{\text{sem}}$)	0.4773	-0.97	0.6085	-0.90	0.3569	-1.47	0.3992	-1.41
w/o Short ($\mathbf{x}_n^{\text{short}}$)	0.4630	-3.93	0.5979	-2.64	0.3404	-6.03	0.3840	-5.17
w/o Long ($\mathbf{x}_n^{\text{long}}$)	0.4729	-1.88	0.6043	-1.59	0.3548	-2.04	0.3973	-1.88
w/o Attributes ($\mathbf{x}_n^{\text{attr}}$)	0.4684	-2.80	0.5979	-2.64	0.3512	-3.06	0.3930	-2.94
w/o SID + Short	0.4600	-4.55	0.5962	-2.90	0.3368	-7.02	0.3808	-5.95
w/o SID + Short + Long	0.3770	-21.77	0.5147	-16.18	0.2693	-25.65	0.3135	-22.57

Table 10: Ablation results of TabPFN-v3.5-SR on LastFM. $\Delta\%$ denotes the relative change from the full model. The best result for each metric is shown in bold.

Variant	HR@5	$\Delta\%$	HR@10	$\Delta\%$	NDCG@5	$\Delta\%$	NDCG@10	$\Delta\%$
Full model	**0.4073**	–	0.5422	–	**0.2930**	–	**0.3368**	–
w/o SID ($\mathbf{x}_n^{\text{sem}}$)	0.4000	-1.80	**0.5514**	$+1.69$	0.2875	-1.88	0.3361	-0.18
w/o Short ($\mathbf{x}_n^{\text{short}}$)	0.3927	-3.60	0.5321	-1.86	0.2747	-6.24	0.3195	-5.12
w/o Long ($\mathbf{x}_n^{\text{long}}$)	0.3771	-7.43	0.5220	-3.72	0.2705	-7.68	0.3177	-5.65
w/o Attributes ($\mathbf{x}_n^{\text{attr}}$)	0.3908	-4.05	0.5376	-0.85	0.2823	-3.65	0.3296	-2.14
w/o SID + Short	0.3908	-4.05	0.5330	-1.69	0.2762	-5.73	0.3220	-4.38
w/o SID + Short + Long	0.2459	-39.64	0.3477	-35.87	0.1638	-44.11	0.1964	-41.69

Table 11: Ablation results of TabPFN-v3.5-SR on Toys. $\Delta\%$ denotes the relative change from the full model. The best result for each metric is shown in bold.

Variant	HR@5	$\Delta\%$	HR@10	$\Delta\%$	NDCG@5	$\Delta\%$	NDCG@10	$\Delta\%$
Full model	**0.5198**	–	**0.6261**	–	**0.4124**	–	**0.4468**	–
w/o SID ($\mathbf{x}_n^{\text{sem}}$)	0.5192	-0.12	0.6255	-0.08	0.4117	-0.18	0.4460	-0.18
w/o Short ($\mathbf{x}_n^{\text{short}}$)	0.4970	-4.39	0.6116	-2.31	0.3830	-7.13	0.4199	-6.01
w/o Long ($\mathbf{x}_n^{\text{long}}$)	0.5139	-1.14	0.6222	-0.61	0.4076	-1.17	0.4427	-0.92
w/o Attributes ($\mathbf{x}_n^{\text{attr}}$)	0.5037	-3.09	0.6113	-2.35	0.3995	-3.13	0.4343	-2.80
w/o SID + Short	0.4947	-4.82	0.6074	-2.99	0.3834	-7.03	0.4198	-6.05
w/o SID + Short + Long	0.4286	-17.54	0.5494	-12.24	0.3177	-22.96	0.3567	-20.16

Table 12: Ablation results of TabPFN-v3.5-SR on Yelp. $\Delta\%$ denotes the relative change from the full model. The best result for each metric is shown in bold.

Variant	HR@5	$\Delta\%$	HR@10	$\Delta\%$	NDCG@5	$\Delta\%$	NDCG@10	$\Delta\%$
Full model	0.6044	–	0.7723	–	0.4445	–	0.4989	–
w/o SID ($\mathbf{x}_n^{\text{sem}}$)	**0.6188**	$+2.38$	**0.7828**	$+1.36$	**0.4561**	$+2.60$	**0.5092**	$+2.05$
w/o Short ($\mathbf{x}_n^{\text{short}}$)	0.6044	$+0.00$	0.7715	-0.11	0.4387	-1.31	0.4928	-1.22
w/o Long ($\mathbf{x}_n^{\text{long}}$)	0.5923	-2.00	0.7590	-1.72	0.4313	-2.96	0.4853	-2.72
w/o Attributes ($\mathbf{x}_n^{\text{attr}}$)	0.5627	-6.90	0.7332	-5.06	0.4050	-8.88	0.4602	-7.77
w/o SID + Short	0.6117	$+1.20$	0.7754	$+0.40$	0.4462	$+0.38$	0.4993	$+0.08$
w/o SID + Short + Long	0.5689	-5.88	0.7471	-3.27	0.4011	-9.76	0.4587	-8.05