Bridging Textual-Collaborative Gap through Semantic Codes for Sequential Recommendation

Enze Liu
Gaoling School of Artificial Intelligence
Renmin University of China
Beijing, China
enzeliu@ruc.edu.cn

Bowen Zheng
Gaoling School of Artificial Intelligence
Renmin University of China
Beijing, China
bwzheng0324@ruc.edu.cn

Wayne Xin Zhao✉
Gaoling School of Artificial Intelligence
Renmin University of China
Beijing, China
batmanfly@gmail.com

Ji-Rong Wen
Gaoling School of Artificial Intelligence
Renmin University of China
Beijing, China
jrwen@ruc.edu.cn

Abstract

In recent years, substantial research efforts have been devoted to enhancing sequential recommender systems by integrating abundant side information with ID-based collaborative information. This study specifically focuses on leveraging the textual metadata (e.g., titles and brands) associated with items. While existing methods have achieved notable success by combining text and ID representations, they often struggle to strike a balance between textual information embedded in text representations and collaborative information from sequential patterns of user behavior. In light of this, we propose **CCFRec**, a novel Code-based textual and Collaborative semantic Fusion method for sequential Recommendation. The key idea behind our approach is to bridge the gap between textual and collaborative information using semantic codes. Specifically, we generate fine-grained semantic codes from multi-view text embeddings through vector quantization techniques. Subsequently, we develop a code-guided semantic-fusion module based on the cross-attention mechanism to flexibly extract and integrate relevant information from text representations. In order to further enhance the fusion of textual and collaborative semantics, we introduce an optimization strategy that employs code masking with two specific objectives: masked code modeling and masked sequence alignment. The merit of these objectives lies in leveraging mask prediction tasks and augmented item representations to capture code correlations within individual items and enhance the sequence modeling of the recommendation backbone. Extensive experiments conducted on four public datasets demonstrate the superiority of CCFRec, showing significant improvements over various sequential recommendation models. Our code is available at https://github.com/RUCAIBox/CCFRec.

CCS Concepts

• **Information systems** → **Recommendation systems**.

Keywords

Sequential Recommendation; Textual-Collaborative Representation Fusion; Semantic Codes

ACM Reference Format:
Enze Liu, Bowen Zheng, Wayne Xin Zhao✉, and Ji-Rong Wen. 2025. Bridging Textual-Collaborative Gap through Semantic Codes for Sequential Recommendation. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2 (KDD '25), August 3–7, 2025, Toronto, ON, Canada.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3711896.3736869

1 Introduction

Sequential recommender systems play a critical role in digital platforms, including video streaming and e-commerce applications. These systems aim to predict the next item by analyzing sequential patterns in the user's historical interactions. Existing methods utilize diverse neural architectures, including convolutional neural networks (CNNs) [30, 31], recurrent neural networks (RNNs) [8, 29], and Transformer models [16, 28, 45], to effectively model sequential user behaviors. Conventional recommendation models predominantly employ a unique ID for item representation, capturing collaborative information in user interaction data. While achieving remarkable success, these ID-based models neglect the rich textual metadata available in modern platforms, which is beneficial for modeling user preferences. In response to this issue, several recent studies [17, 19, 21, 33, 35] have increasingly focused on harnessing textual information to enhance the sequential recommendation framework.

Early approaches [11, 39] achieve moderate success by introducing text representations encoded by pre-trained language models (PLMs) in a straightforward way (*e.g.,* sum and concat), but these methods overlook the semantic gap between ID embeddings and

✉ Corresponding author.

KDD '25, Toronto, ON, Canada

ACM ISBN 979-8-4007-1454-2/2025/08
https://doi.org/10.1145/3711896.3736869

text representations. To address this limitation, some studies adopt various techniques, including contrastive learning [43], attention mechanisms [17, 35] and graph-based aggregation [13], to better fuse ID embeddings and text representations. Furthermore, recent studies propose mapping text representations into semantic codes for sequential recommendation [9, 13, 23] or generative recommendation [12, 25, 42]. These codes are typically derived through clustering or vector quantization based on text representations [15, 25]. A key merit of this approach is that the shared codes between different items reflect their semantic similarities. Despite their notable effectiveness, these methods struggle to achieve an appropriate trade-off between the textual semantics embedded in text representations and the collaborative semantics implied by sequential patterns. Integrating item ID embeddings with text representations may result in the model overemphasizing textual features while neglecting the collaborative relationships between items [9]. Although code-based methods can alleviate the over-reliance on text, they suffer from textual semantic loss in a certain extent.

Considering these issues, our idea is to employ semantic codes as a bridge to achieve a better balance between textual and collaborative information. This approach is motivated by two key characteristics of semantic codes. First, they can be regarded as a specialized form of item IDs, associated with learnable embeddings that can flexibly capture collaborative semantics by modeling interaction sequences. Second, the multiple codes for each item imply semantics of different granularities or levels, injecting prior knowledge of semantic similarity into item representations, which can serve as guidance for refined textual-collaborative semantics integration. To develop our approach, we focus on two aspects: (1) leveraging semantic codes as guidance to flexibly extract useful information from text representations, alleviating the over-reliance on textual information, and (2) learning informative item representations to enhance sequential recommendation.

To this end, we propose **CCFRec**, an innovative Code-based textual and Collaborative semantic Fusion approach for sequential Recommendation. Unlike previous methods that directly utilize text embeddings as item representations, our approach develops a code-guided fusion module to derive semantically enriched item representations. In this module, semantic codes serve as queries to flexibly extract relevant information from text representations, thereby mitigating the excessive dependence on item text while enabling a more comprehensive learning of code-based collaborative information. To obtain semantic codes with different granularity information, we encode various item attributes (*e.g.,* title, brand, categories) separately as multi-view text representations and quantize them into a tuple of codes through vector quantization. Then, we implement the semantic fusion module as a multi-layer cross-attention network to incorporate code embeddings and multi-view text representations. In order to further enhance semantic fusion and sequence modeling, we propose two optimization objectives through item code masking, namely *masked code modeling* and *masked sequence alignment*. For masked code modeling, the semantic fusion module recovers the masked code from multi-view text representations, which has two advantages. Firstly, it can capture the code relationships within individual items. Secondly, it promotes the integration between code embeddings and text representations. For masked sequence alignment, we align the user preference representations derived from interaction sequences involving masked codes with the original one, aiming to enhance the robustness of user preference modeling.

In summary, the main contributions of this paper are as follows:

- We propose **CCFRec**, a novel code-based textual and collaborative semantic fusion method for sequential recommendation, which bridges the textual-collaborative gap through semantic codes.
- We design an optimization method via code masking that further enhances semantic fusion and sequence modeling through masked code modeling and masked sequence alignment.
- Extensive experiments on four public recommendation benchmarks demonstrate the effectiveness of our proposed framework CCFRec.

2 Methodology

2.1 Problem Statement

In our scenario, we focus on sequential recommendation tasks. Let $\mathcal{U} = \{u_1, \ldots, u_{|\mathcal{U}|}\}$ denote a set of users and $\mathcal{V} = \{v_1, \ldots, v_{|\mathcal{V}|}\}$ denote a set of items. Each item v is associated with a unique item ID and a series of text attributes $\mathcal{A} = \{a_1, \ldots, a_m\}$ (*e.g.,* title, brand, category, *etc.*) where m represents the attribute number. Here, each item attribute a_i corresponds to a descriptive text. Additionally, a_i is mapped to a tuple of semantic codes, denoted by $C_i = (c_1^i, \ldots, c_k^i)$, where semantic codes under the same view (*e.g.,* title or brand) share a common code space. Given a user interaction sequence $s = \{v_1, v_2, \ldots, v_t\}$ in chronological order, the objective of sequential recommendation is to predict the next item that the user is most likely to interact with, which is formulated as $\max P(v_{t+1}|\{v_1, \ldots, v_t\})$.

2.2 Text Embeddings and Semantic Codes

In this part, we describe the process of deriving two distinct representations for items. This involves (a) constructing multi-view text embeddings by encoding each item attribute individually, and (b) mapping the text embeddings to semantic codes through multi-level vector quantization methods.

2.2.1 Multi-View Text Embeddings. Text embeddings are widely acknowledged for encapsulating rich textual information. To effectively harness item content, previous studies [13, 14, 21, 36] typically concatenate all associated text and encode it into a single embedding using pre-trained language models (PLMs). However, it's difficult to leverage fine-grained item information by jointly encoding all content. Moreover, due to the limited input length of PLMs, the item text often needs to be truncated, leading to potential incomplete semantics. Therefore, to fully exploit attribute-wise semantic information, we propose encoding each item attribute individually to obtain multi-view text embeddings. Formally, given text attributes $\mathcal{A} = \{a_1, \ldots, a_m\}$ of item v, the text embedding of attribute a_i is formulated as

$$\boldsymbol{z}_i^t = \text{PLM}(a_i),\ i \in \{1, \ldots, m\},$$

where $\boldsymbol{z}_i^t$ denotes the mean pooling representation of the output.

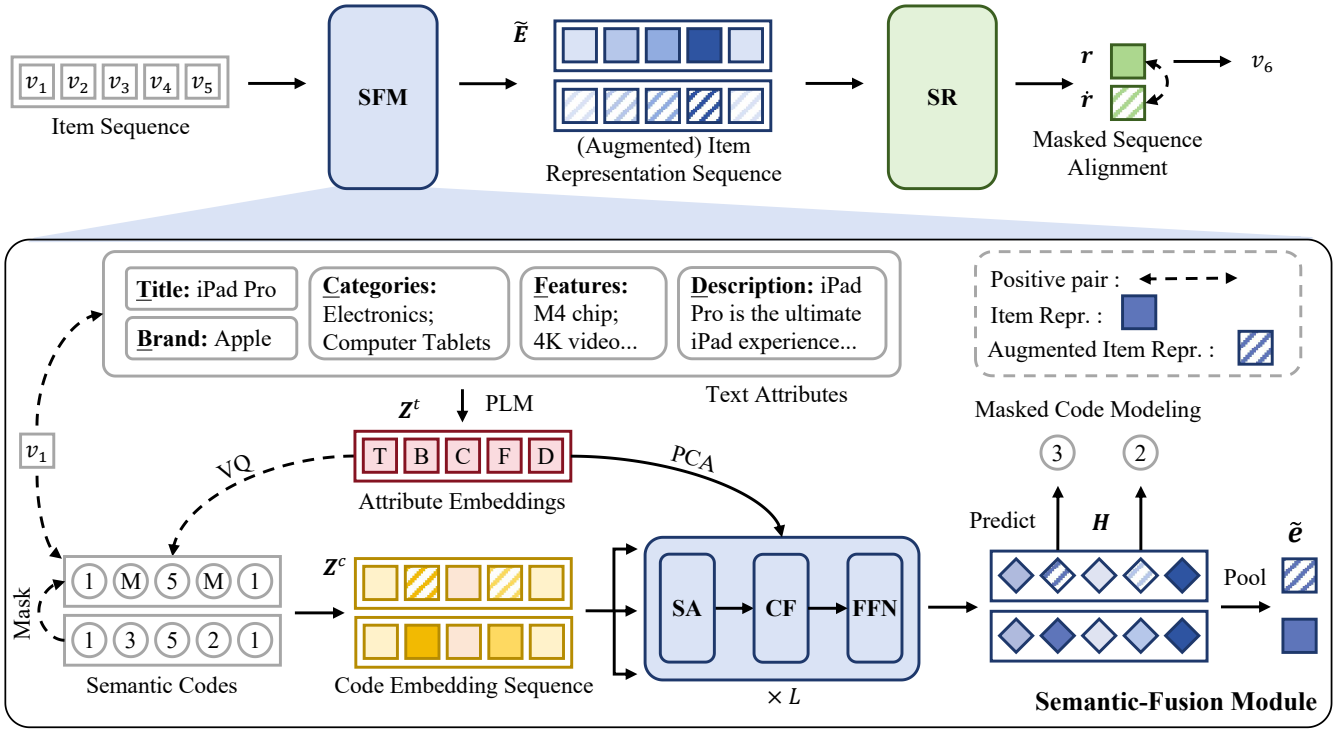

Figure 1: Overall framework of CCFRec, which comprises two key components, *i.e.,* the Semantic-Fusion Module (SFM) and the Sequential Recommendation backbone (SR). SFM stacks Self-Attention (SA) layers, Cross-Fusion (CF) layers, and Feed-Forward Networks (FFN) to enable effective learning of semantic-fused item representations.

2.2.2 Vector Quantization for Semantic Codes. In our framework, semantic codes serve a dual purpose. First, they are linked to learnable embeddings, enabling the flexible capture of collaborative information. Second, they act as bridges to facilitate the fusion of textual and collaborative semantics. Unlike unique IDs, semantic codes allow semantically similar items to share representations, thereby capturing inherent relationships among items. Moreover, the multi-level structure of these codes reflects textual semantics from different aspects, providing guidance for fine-grained semantic fusion. These characteristics make semantic codes particularly effective for item representation, as evidenced by prior research [9, 23, 25]. In our approach, we utilize vector quantization techniques, such as product quantization (PQ) and residual quantization (RQ), to map item attribute embeddings into multi-level discrete codes. Specifically, for an attribute embedding $\boldsymbol{z}_i^t$ of item v, the vector quantization process generates a tuple of discrete codes $C_i = (c_1^i, \ldots, c_k^i)$, where k denotes the number of quantized sub-vectors, which is a pre-defined hyper-parameter. Given that each item has m distinct attributes, this process produces a total of $n_c = m \times k$ semantic codes, represented as $(c_1^1, c_2^1, \ldots, c_{k-1}^m, c_k^m)$.

2.3 Code-Guided Semantic Fusion

To effectively integrate textual and ID-based collaborative semantics, prior research [5, 11, 13, 19, 36, 39, 43] has typically combined text representations with ID embeddings or utilized the discrete codes derived from text embeddings to enhance sequential recommenders. Despite their effectiveness, these methods often struggle to achieve an appropriate trade-off between the textual information encoded in text embeddings and the collaborative information implied by sequential patterns. To address this issue, we propose to employ semantic codes as bridges to extract and integrate relevant information from text representations. A key challenge in this approach lies in fully unleashing the potential of semantic codes to serve as effective bridges. To tackle it, we introduce a code-guided semantic-fusion module that integrates text representations with multi-level semantic code embeddings using cross-attention mechanisms, where semantic codes are considered as the query and text representations are regarded as the key and value. This design enables the semantic codes to flexibly extract useful information from text representations, mitigating the over-reliance on item text. Subsequently, a Transformer-based recommendation backbone leverages these fused item representations to model sequential user behavior and makes final recommendations.

2.3.1 Semantic Code Embeddings. Our approach leverages two types of embeddings, *i.e.,* attribute embeddings and code embeddings, to capture textual and collaborative information, respectively. Each item v is associated with n_c semantic codes denoted as $(c_1, \ldots, c_{n_c})$ (for simplicity, we do not distinguish the attributes). To learn the collaborative correlations among items, we set up n_c code embedding tables for lookup. For each position l, all items share a

common code embedding table $\boldsymbol{E}^{(l)} \in \mathbb{R}^{C\times d}$, where d denotes the dimension of the item representations. Using these tables, we look up the code embeddings for item v as $\{\boldsymbol{z}_i^c\}_{i=1}^{n_c}$ where $\boldsymbol{z}_i^c \in \boldsymbol{E}^{(i)}$. In addition to code embeddings, each item v is also associated with m attribute embeddings, denoted as $\{\boldsymbol{z}_i^t\}_{i=1}^{m}$. Here, $\boldsymbol{z}_i^t \in \mathbb{R}^{d_e}$ represents the embedding of the i-th attribute, where d_e is typically larger than d. To ensure compatibility between two types of embeddings, we apply Principal Component Analysis (PCA) to reduce the dimension of $\boldsymbol{z}_i^t$ to d.

2.3.2 Semantic-Fusion Module. To obtain semantic-fused item representations, we propose a semantic-fusion module composed of multiple stacked blocks. Each block consists of a self-attention layer, a cross-fusion layer and a multilayer perceptron network. Each item is associated with its attribute representations $\boldsymbol{Z}^t = [\boldsymbol{z}_1^t, \ldots, \boldsymbol{z}_m^t] \in \mathbb{R}^{m\times d}$ and code embeddings $\boldsymbol{Z}^c = [\boldsymbol{z}_1^c, \ldots, \boldsymbol{z}_{n_c}^c] \in \mathbb{R}^{n_c\times d}$. The code embeddings are first fed into the bi-directional self-attention layer to obtain mixed code representations. Then a cross-fusion layer based on the multi-head attention mechanism (denoted by MHA(Q, K, V)) is employed to achieve the semantic fusion of the text and code sequences. The whole process can be formally written as:

$$\tilde{\boldsymbol{H}}^l = \text{MHA}\left(\boldsymbol{H}^{l-1}, \boldsymbol{H}^{l-1}, \boldsymbol{H}^{l-1}\right), \tag{1}$$

$$\boldsymbol{H}^l = \text{FFN}\left(\text{MHA}\left(\tilde{\boldsymbol{H}}^l, \boldsymbol{Z}^t, \boldsymbol{Z}^t\right)\right),\ l \in \{1, \ldots, L\}, \tag{2}$$

where $\boldsymbol{H}^0 = \boldsymbol{Z}^c$ and L is the number of layers in the semantic-fusion module and FFN$(\cdot)$ denotes the multilayer perceptron network. The ordering of elements within these two sequences holds no intrinsic meaning. Therefore, we omit position embeddings to neglect sequential relationships. The final semantically fused sequence representation, $\boldsymbol{H} \in \mathbb{R}^{n_c\times d}$, is aggregated to generate the semantic-fused item representation denoted as $\boldsymbol{e} = \text{Pool}(\boldsymbol{H})$, where Pool$(\cdot)$ denotes mean pooling. To enhance semantic code embedding learning, the pooled semantic code representation is integrated, yielding the final item representation $\tilde{\boldsymbol{e}} = \boldsymbol{e} + \text{Pool}(\boldsymbol{Z}^c)$.

2.3.3 Recommendation Backbone. . We adopt a widely used Transformer architecture as our recommendation backbone. Given an interaction sequence $s = \{v_1, v_2, \ldots, v_n\}$, we can obtain the semantic-fused item representation matrix $\tilde{\boldsymbol{E}} = \{\tilde{\boldsymbol{e}}_1; \ldots; \tilde{\boldsymbol{e}}_n\} \in \mathbb{R}^{n\times d}$ where [;] denotes the concatenation operation. The item representations $\tilde{\boldsymbol{e}}_i$ and absolute position embeddings $\boldsymbol{p}_i$ are summed up as the input to the backbone for sequential preference modeling. Subsequently, the hidden states are updated through a multi-head attention (MHA) mechanism followed by a multilayer perceptron network (FFN). This process is formally defined as:

$$\hat{\boldsymbol{h}}_i^0 = \tilde{\boldsymbol{e}}_i + \boldsymbol{p}_i,\ i \in \{1, \ldots, n\}, \tag{3}$$

$$\hat{\boldsymbol{H}}^l = \text{FFN}\left(\text{MHA}\left(\hat{\boldsymbol{H}}^{l-1}, \hat{\boldsymbol{H}}^{l-1}, \hat{\boldsymbol{H}}^{l-1}\right)\right),\ l \in \{1, \ldots, L_r\}, \tag{4}$$

where $\hat{\boldsymbol{H}}^l = \{\hat{\boldsymbol{h}}_1^l; \ldots; \hat{\boldsymbol{h}}_n^l\} \in \mathbb{R}^{n\times d}$ denotes the hidden state sequence in the l-layer and L_r denotes the total number of layers in the recommendation backbone. The final hidden state corresponding to the last position is utilized as the representation of user preferences, denoted by $\boldsymbol{r} = \hat{\boldsymbol{H}}[-1]$. Following prior studies [36, 43, 44], we employ the cross-entropy loss as the training objective, which is written as

$$\mathcal{L}_{\text{CE}} = -\log \frac{\exp(g(\boldsymbol{r}, \tilde{\boldsymbol{e}}_{n+1})/\tau)}{\sum_{j\in\mathcal{B}} \exp(g(\boldsymbol{r}, \tilde{\boldsymbol{e}}_j)/\tau)}, \tag{5}$$

where $\mathcal{B}$ is the set of sampled negative items, $g(\cdot,\cdot)$ denotes the cosine similarity measure and τ is a temperature coefficient.

2.4 Enhanced Representation Learning via Code Masking

Compared with existing semantic fusion methods [13, 36, 39], optimizing our framework is more challenging due to its two hierarchically arranged components. A single recommendation optimization objective is insufficient to ensure the comprehensive learning of the entire framework, particularly for the code representations. To address this issue, we propose two optimization objectives based on code masking to further enhance the representation learning in CCFRec. Specifically, we introduce the Masked Code Modeling (MCM) task, which promotes the semantic fusion between text and codes to refine item representations. Additionally, we incorporate the Masked Sequence Alignment (MSA) task to improve the sequence modeling capabilities of the recommendation backbone.

Masked Code Modeling. To foster the semantic fusion between text representations and code embeddings, we present the Masked Code Modeling (MCM) task, inspired by the success of Masked language modeling (MLM) in natural language processing. MLM is widely used to predict randomly masked word tokens in sequences by conditioning on the unmasked tokens. In MCM, at each training step, a random ρ percent of the codes in the semantic code sequence is masked following the same strategy described in BERT [4]. The semantic-fusion module is then tasked with recovering the masked codes based on attribute embeddings and the remaining unmasked codes. Formally, given a semantic code sequence $s^c = \{c_1, \ldots, c_{n_c}\}$ of item v, we randomly mask a ρ ratio of the codes and obtain the masked sequence $\dot{s}^c = \{c_1, \ldots, [MASK], \ldots, c_{n_c}\}$ and let $\mathcal{M}^c$ represent the set of mask codes. Then, the code sequence representation produced by the semantic-fusion module is denoted as $\{\dot{\boldsymbol{h}}_i\}_{i=1}^{n}$ where n represents the sequence length. The MCM loss is formulated as

$$\mathcal{L}_{\text{MCM}} = \frac{1}{|\mathcal{M}^c|} \sum_{x\in\mathcal{M}^c} -\log \frac{\exp(g(\dot{\boldsymbol{h}}_x, \boldsymbol{z}_x^c)/\tau)}{\sum_{j\in C} \exp(g(\dot{\boldsymbol{h}}_x, \boldsymbol{z}_j^c)/\tau)}, \tag{6}$$

where C denotes the set of all semantic codes. This approach offers two main advantages: first, it strengthens the correlations among semantic codes; second, it improves the alignment between text and code embeddings through text-dependent masked code reconstruction, thereby facilitating more effective textual-collaborative semantic fusion.

Masked Sequence Alignment. MCM is primarily used to optimize the SFM. In order to enhance the sequence representation learning of the recommendation backbone, we introduce the Masked Sequence Alignment (MSA) task, using the contrastive learning technique that has been widely validated in sequential recommendation tasks [3, 24, 34, 38]. In MSA, code masking serves as an effective data augmentation strategy to generate augmented item representations. This approach increases the diversity of inputs to the recommendation backbone, thereby refining its ability to

model user sequences. Specifically, we first generate augmented item representations by feeding the masked code sequence and associated text representations into the SFM. These augmented item representations are then passed to the recommendation backbone to produce augmented sequence representations. Subsequently, the two sequence representations are aligned using the InfoNCE loss. The representations derived from the same sequence are considered as positive samples, while those from different sequences within the same batch are treated as negative samples. The objective function for *masked sequence alignment* is formulated as follows:

$$\mathcal{L}_{\text{MSA}} = \frac{1}{2}\left(\text{InfoNCE}(\dot{\boldsymbol{r}}, \boldsymbol{r}, \mathcal{B}_{\boldsymbol{r}}) + \text{InfoNCE}(\boldsymbol{r}, \dot{\boldsymbol{r}}, \mathcal{B}_{\dot{\boldsymbol{r}}})\right), \tag{7}$$

where $\mathcal{B}_{\boldsymbol{r}}$ and $\mathcal{B}_{\dot{\boldsymbol{r}}}$ denote the batch of original sequence representations and augmented sequence representations generated by the recommendation backbone, respectively. $\text{InfoNCE}(\cdot,\cdot,\cdot)$ denotes the InfoNCE loss, which can be written as:

$$\text{InfoNCE}(\boldsymbol{x}, \boldsymbol{y}^+, \mathcal{R}_{\boldsymbol{y}}) = -\log \frac{\exp(g(\boldsymbol{x}, \boldsymbol{y}^+)/\tau)}{\sum_{\boldsymbol{y}\in\mathcal{R}_{\boldsymbol{y}}} \exp(g(\boldsymbol{x}, \boldsymbol{y})/\tau)}, \tag{8}$$

where $\boldsymbol{x}$ and $\boldsymbol{y}^+$ represent a pair of positive instances and $\mathcal{R}_{\boldsymbol{y}}$ denotes a set of samples including both positive and negative instances.

Finally, the overall optimization objectives can be denoted as follows:

$$\mathcal{L} = \mathcal{L}_{\text{CE}} + \alpha\mathcal{L}_{\text{MCM}} + \beta\mathcal{L}_{\text{MSA}}, \tag{9}$$

where α and β are hyper-parameters that control the trade-off between the respective training objectives.

2.5 Discussion

To highlight the novelty and contributions of our approach, we compare CCFRec with existing text-enhanced methods, as shown in Table 1. Firstly, previous methods typically adopt text representations and ID embeddings to capture textual and collaborative information, respectively. As a result, these two types of representations remain isolated and lack information sharing prior to their interaction (*e.g.,* sum or attention mechanisms). In contrast, ***we establish a connection between them by adopting semantic codes to capture collaborative information***. The connection between semantic codes and text representations allows codes to serve as bridges connecting the two types of information. Secondly, we adopt the ***code-guided cross attention*** mechanism for semantic fusion by considering semantic codes as queries to extract useful information from text representations, enabling flexible textual semantic integration. Thirdly, we introduce the masked code modeling task to capture the ***code relationships within individual items*** and improve the fusion between text representations and code embeddings.

3 Time Complexity

Training. As outlined in Section 2, CCFRec consists of two key components, the semantic-fusion module (SFM) and the sequential recommendation backbone (SR). For the SFM, the computational complexities of the self-attention layer, the cross-fusion layer, and the feed-forward network when processing a single item are $O(M^2d)$, $O(MKd)$ and $O(Md^2)$, respectively. Here, d denotes the model dimension, while M and K represent the length of the semantic code sequence and item attribute sequence. Consequently, the overall time complexity for deriving a semantic-fused item representation is $O(M^2d + MKd + Md^2)$. For the SR, which utilizes a SASRec-like architecture, the time consumption of user behavior modeling is $O(N^2d + Nd^2)$, where N is the length of the user interactions. Thus, the total training complexity of CCFRec is $O(N^2d + Nd^2 + NM^2d + NMKd + NMd^2)$. For brevity, the computation of training objectives is omitted. Notably, in practical scenarios, the code length M and attribute number K are typically much smaller than interaction sequence length N (*i.e.,* $M, K \ll N$). This ensures that the additional computational overhead introduced by CCFRec remains acceptable for real-world applications.

Table 1: Comparison of different methods. 'Flexible' denotes the flexible integration of textual information. 'CRM' denotes the code relation modeling within individual items.

Methods	Input	Fusion Type	Flexible	CRM
UniSRec [11]	Text & ID	Vector addition	✗	✗
VQRec [9]	Code	Vector quantization	✓	✗
TedRec [36]	Text & ID	Contextual convolution	✗	✗
CCFRec (ours)	Text & Code	Code-guided attention	✓	✓

Deployment. Our proposed framework can be efficiently deployed in a lightweight manner by eliminating the SFM module. During deployment, as the learned representations of text attributes and semantic codes remain consistent for each item, we can precompute and store their semantic-fused representations by processing the attribute and code sequences through the SFM in advance. Consequently, the inference complexity is maintained at $O(N^2d + Nd^2)$, which is identical to that of the mainstream sequential recommendation models such as SASRec.

4 Experiments

In this section, we empirically demonstrate the effectiveness of our proposed framework CCFRec through extensive experiments and rigorous analysis.

4.1 Experiment Setup

4.1.1 Dataset. We conduct experiments on four subsets of the latest Amazon 2023 review data [10], *i.e.,* "*Musical Instruments*", "*Video Games*", "*Baby Products*" and "*Industrial Scientific*". To ensure a robust evaluation, we follow the same preprocessing steps (*i.e.,* 5-core filtering) as described in prior studies [36, 43, 44] to remove inactive users and unpopular items with less than five interactions. Next, we organize user behavior sequences in chronological order, limiting the maximum sequence length to 20 items. We categorize item attributes into five fields: *title*, *brand*, *categories*, *features*, and *description*, to capture fine-grained and rich semantic information. Each field's content is truncated to a maximum of 512 tokens per item. The detailed statistics of the preprocessed datasets are presented in Table 2.

4.1.2 Baseline Models. We compare the proposed CCFRec with various sequential recommendation (SR) baselines, including the

Table 2: Statistics of the preprocessed datasets. Avg.*len* denotes the average length of the historical interaction sequences.

Dataset	#Users	#Items	#Interactions	Sparsity	Avg.*len*
Instrument	57,439	24,587	511,836	99.964%	8.91
Scientific	50,985	25,848	412,947	99.969%	8.10
Game	94,762	25,612	814,586	99.966%	8.60
Baby	150,777	36,013	1,241,083	99.977%	8.23

following three categories: (1) *Traditional SR models*: **GRU4Rec** [8] adopts GRUs to capture sequential patterns in user interactions. **BERT4Rec** [28] applies the bidirectional self-attention mechanism and mask prediction task for sequential recommendation. **SASRec** [16] utilizes a unidirectional self-attentive model to predict the next item of interest. **FMLP-Rec** [44] introduces an all-MLP model with learnable filters to reduce the noise in sequence modeling. (2) *CL-based SR models*: **S^3-Rec** [43] employs the correlation between items and features as self-supervised signals to empower user behavior modeling. **DuoRec** [24] proposes a model-level data augmentation approach based on Dropout to resolve the representation degeneration. **MAERec** [38] leverages a graph-masked autoencoder to adaptively enhance the sequential recommender. (3) *Text-enhanced SR models*: **FDSA** [39] captures the item-level and feature-level correlations through a dual-stream self-attentive network. **UniSRec** [11] learns transferable textual item representations through a MoE-enhanced adaptor. We implement two variants of it: (i) $\text{UniSRec}_{\text{T}}$ with only text representations, and (ii) $\text{UniSRec}_{\text{ID+T}}$ with both text and ID representations. **VQRec** [9] introduces vector-quantized item representations for sequential recommendation. **MMSR** [13] proposes a graph-based method to integrate multimodal information in an adaptive order for enhanced sequential recommenders. **TedRec** [36] achieves sequence-level text-ID semantic fusion in the frequency domain, improving sequential recommendation performance.

4.1.3 Evaluation Settings. To evaluate the performance of sequential recommendation task, we adopt top-K Recall and top-K Normalized Discounted Cumulative Gain (NDCG) as the evaluation metrics, where K is set to 5 and 10. In line with prior studies [25, 43], we employ the *leave-one-out* strategy for dataset splitting. For each user interaction sequence, the last interacted item is used as testing data, the second-last item is used as validation data, and all remaining items are used for training. We compute the ranking results across the entire item set to guarantee rigorous evaluations.

4.1.4 Implementation Details. For CCFRec, we leverage Sentence-T5 [22] to encode the text attributes associated with each item as its text embeddings. Both the semantic-fusion module and the sequential recommender are configured with 2 layers and 2 attention heads, with a hidden dimension of 512. The embedding size is fixed at 128, while each attribute vector is quantized into 4 sub-vectors (*i.e.,* $k = 4$) with a codebook size of 256. The temperature coefficient τ and mask ratio ρ are set to 0.07 and 0.5, respectively. We adopt the Adam optimizer for model training, with the learning rate tuned in {0.003, 0.001, 0.0005}. The dropout probability is tuned within {0.1, 0.2, 0.3, 0.4, 0.5} and the loss coefficients α and β are searched in {0.2, 0.4, 0.6, 0.8} for optimal performance. To avoid overfitting, we utilize an early-stopping strategy, halting training when NDCG@10 in the validation set shows no improvement over 10 consecutive epochs. We implement the majority of baseline models using the open-source recommendation system library RecBole [37, 40, 41]. For CL-based SR models, namely DuoRec and MAERec, we utilize the self-supervised learning framework SSLRec [26]. For fair comparisons, we fix the embedding size of all models to 128 and perform the hyperparameter grid search for optimal results. Furthermore, all Transformer-based baseline models adopt the same recommendation backbone architecture as our CCFRec for fairness.

4.2 Overall Performance

We compare CCFRec with various baseline approaches on four public benchmarks and present the overall results in Table 3. From the results, we have the following findings:

For traditional SR models, FMLP-Rec demonstrates superior performance than SASRec by replacing the self-attention layer with filter-enhanced MLPs. CL-based SR models (*i.e.,* S^3Rec, DuoRec, MAERec) generally outperform traditional SR models (*i.e.,* GRU4Rec, BERT4Rec, SASRec, FMLP-Rec), highlighting the effectiveness of contrastive learning techniques in enhancing sequence representation learning. Notably, the pure ID-based MAERec exhibits better performance than most text-enhanced SR models (*i.e.,* FDSA, UniSRec, VQRec, MMSR) in the Scientific and Game datasets, underscoring the critical role of collaborative information in modeling user interactions.

Regarding text-enhanced SR models, they typically outperform other SR models by leveraging auxiliary textual information from item contents. $\text{UniSRec}_{\text{T}}$ performs better than $\text{UniSRec}_{\text{ID+T}}$ indicating that the fusion of textual information and ID-based collaborative information remains suboptimal. VQRec achieves outstanding results in the Baby, demonstrating the effectiveness of vector-quantized item representations. TedRec excels by achieving sequence-level semantic fusion between text and IDs in the frequency domain, showing the best performance across most datasets.

Finally, our proposed CCFRec(PQ) achieves the best or second-best performance across all datasets, significantly outperforming the best baseline models in most cases. CCFRec(RQ) also achieves competitive results but performs slightly worse than CCFRec(PQ), which may be attributed to the fact that PQ-based semantic codes preserving semantics of different levels are more suitable for our framework than RQ-based codes implying semantics of different granularities. Different from existing text-enhanced methods, we propose leveraging semantic codes to bridge the gap between textual and collaborative information. With the specially designed semantic-fusion module and optimization objectives, including masked code modeling and masked sequence alignment, we achieve effective textual-collaborative semantic fusion for sequential recommendations, significantly improving the model performance.

4.3 Ablation Study

To evaluate the contribution of each proposed component, we conduct ablation studies on the Instrument, Scientific, and Baby datasets. Table 4 presents the comparative results of the following six variants: (1) $\underline{w/o\ \mathcal{L}_{\text{MCM}}}$ removes the masked code modeling

Table 3: Performance comparisons among different methods. The best and second-best results are highlighted in bold and underlined font, respectively. "N@K" and "R@K" represent abbreviations of "NDCG@K" and "Recall@K", respectively. "PQ" and "RQ" are short for "Product Quantization" and "Residual Quantization". "Improv." denotes the relative improvement ratios of CCFRec over the top-performing baselines.

Methods	Instrument				Scientific				Game				Baby			
	R@5	R@10	N@5	N@10	R@5	R@10	N@5	N@10	R@5	R@10	N@5	N@10	R@5	R@10	N@5	N@10
GRU4Rec	0.0339	0.0540	0.0216	0.0281	0.0230	0.0374	0.0148	0.0194	0.0530	0.0820	0.0350	0.0443	0.0219	0.0354	0.0140	0.0184
BERT4Rec	0.0307	0.0485	0.0195	0.0252	0.0186	0.0296	0.0119	0.0155	0.0460	0.0735	0.0298	0.0386	0.0181	0.0300	0.0116	0.0154
SASRec	0.0333	0.0523	0.0213	0.0274	0.0259	0.0412	0.0150	0.0199	0.0535	0.0847	0.0331	0.0438	0.0221	0.0362	0.0135	0.0180
FMLP-Rec	0.0339	0.0536	0.0218	0.0282	0.0269	0.0422	0.0155	0.0204	0.0528	0.0857	0.0338	0.0444	0.0228	0.0367	0.0146	0.0190
S^3Rec	0.0317	0.0496	0.0199	0.0257	0.0263	0.0418	0.0171	0.0219	0.0485	0.0769	0.0315	0.0406	0.0216	0.0338	0.0126	0.0168
DuoRec	0.0373	0.0573	0.0246	0.0310	0.0245	0.0379	0.0166	0.0209	0.0559	0.0844	0.0378	0.0469	0.0211	0.0346	0.0139	0.0182
MAERec	0.0369	0.0577	0.0243	0.0310	0.0303	0.0452	0.0205	0.0253	0.0618	0.0936	0.0411	0.0513	0.0224	0.0370	0.0146	0.0193
FDSA	0.0369	0.0576	0.0240	0.0307	0.0273	0.0416	0.0183	0.0229	0.0544	0.0852	0.0361	0.0460	0.0243	0.0387	0.0158	0.0205
$\text{UniSRec}_{\text{T}}$	0.0376	0.0589	0.0244	0.0312	0.0296	0.0469	0.0191	0.0246	0.0587	0.0925	0.0372	0.0480	0.0239	0.0380	0.0153	0.0198
$\text{UniSRec}_{\text{ID+T}}$	0.0370	0.0598	0.0234	0.0308	0.0286	0.0457	0.0157	0.0214	0.0563	0.0921	0.0347	0.0459	0.0229	0.0386	0.0140	0.0190
VQRec	0.0379	0.0602	0.0227	0.0298	0.0293	0.0461	0.0170	0.0224	0.0581	0.0926	0.0355	0.0466	0.0260	0.0409	0.0166	0.0214
MMSR	0.0360	0.0569	0.0231	0.0300	0.0264	0.0427	0.0170	0.0208	0.0558	0.0881	0.0372	0.0461	0.0232	0.0389	0.0142	0.0185
TedRec	0.0374	0.0570	0.0249	0.0313	0.0282	0.0415	0.0195	0.0238	0.0624	0.0956	**0.0419**	0.0526	0.0246	0.0380	0.0165	0.0208
CCFRec(RQ)	0.0426	0.0670	0.0273	0.0351	0.0348	0.0540	0.0214	0.0276	0.0642	0.1023	0.0400	0.0523	0.0285	0.0443	0.0182	0.0232
CCFRec(PQ)	**0.0432**	**0.0682**	**0.0281**	**0.0361**	**0.0364**	**0.0555**	**0.0224**	**0.0285**	**0.0658**	**0.1042**	0.0413	**0.0536**	**0.0286**	**0.0455**	**0.0184**	**0.0238**
Improv.	+14.0%	+13.3%	+12.9%	+15.3%	+20.1%	+18.3%	+9.3%	+12.7%	+5.5%	+9.0%	–	+1.9%	+10.0%	+11.3%	+10.8%	+11.2%

Table 4: Ablation study of our proposed method on three datasets. The best and second-best results are denoted in bold and underlined fonts, respectively.

Variants	Instrument				Scientific				Baby			
	R@5	R@10	N@5	N@10	R@5	R@10	N@5	N@10	R@5	R@10	N@5	N@10
(0) CCFRec	**0.0432**	**0.0682**	**0.0281**	**0.0361**	**0.0364**	**0.0555**	**0.0224**	**0.0285**	**0.0286**	**0.0455**	**0.0184**	**0.0238**
(1) *w/o* $\mathcal{L}_{\text{MCM}}$	0.0411	0.0668	0.0260	0.0343	0.0348	0.0548	0.0214	0.0279	0.0275	0.0444	0.0176	0.0230
(2) *w/o* $\mathcal{L}_{\text{MSA}}$	0.0414	0.0664	0.0262	0.0342	0.0356	0.0548	0.0215	0.0277	0.0275	0.0439	0.0176	0.0229
(3) *w/o* Text Emb	0.0416	0.0649	0.0268	0.0343	0.0343	0.0538	0.0212	0.0274	0.0281	0.0444	0.0183	0.0236
(4) Random Code	0.0406	0.0641	0.0256	0.0331	0.0317	0.0501	0.0191	0.0250	0.0270	0.0427	0.0171	0.0222
(5) Global Emb	0.0422	0.0671	0.0273	0.0352	0.0330	0.0533	0.0203	0.0268	0.0278	0.0438	0.0178	0.0229
(6) *w/o* CA	0.0408	0.0640	0.0251	0.0326	0.0319	0.0509	0.0185	0.0246	0.0262	0.0427	0.0167	0.0220

(MCM) defined in Eq. (6). (2) *w/o* $\mathcal{L}_{\text{MSA}}$ without the masked sequence alignment (MSA) in Eq. (7). (3) *w/o* Text Emb substitutes text embeddings with semantic codes as cross-fusion layer input. (4) Random Code replaces the semantic codes derived from text embeddings with random codes. (5) Global Emb encodes all text attributes into a single global embedding, replacing the individual attribute embeddings (6) *w/o* CA replaces the SFM with mean pooling of all associated text representations and code embeddings to generate the final item representation.

The experimental results reveal that removing any component from CCFRec consistently degrades model performance. This performance decline underscores the critical role of both proposed optimization objectives (i.e., MCM and MSA) in improving the representation learning of the overall framework. Furthermore, it demonstrates the significance of textual and collaborative information, as captured by text representations and code embeddings, respectively.

Table 5: Ablation studies of item attributes on Instrument and Scientific datasets.

Title	Brand	Feat.	Cate.	Desc.	Instrument		Scientific	
					R@10	N@10	R@10	N@10
✓					0.0610	0.0319	0.0520	0.0262
✓	✓				0.0628	0.0329	0.0531	0.0267
✓	✓	✓			0.0653	0.0344	0.0545	0.0278
✓	✓	✓	✓		0.0667	0.0355	0.0552	0.0283
✓	✓	✓	✓	✓	**0.0682**	**0.0361**	**0.0555**	**0.0285**

4.4 Further Analysis

4.4.1 Impact of Item Attributes. To investigate how each item attribute affects the final performance, we conduct ablation studies on the Instrument and Scientific datasets by sequentially adding individual attributes. The results are reported in Table 5, from which we can find that text contents have a significant impact on the final performance. In general, richer attribute contents enable CCFRec

Table 6: Performance analysis of code number in CCFRec. k, m and C denote the number of sub-vectors in vector quantization, the number of attributes and the codebook size of each sub-vector, respectively.

Number	Instrument				Scientific			
($k \times m \times C$)	R@5	R@10	N@5	N@10	R@5	R@10	N@5	N@10
$2 \times 5 \times 256$	0.0400	0.0649	0.0256	0.0337	0.0338	0.0542	0.0208	0.0274
$4 \times 5 \times 256$	0.0432	0.0682	0.0281	0.0361	0.0364	0.0555	0.0224	0.0285
$8 \times 5 \times 256$	**0.0445**	**0.0696**	**0.0285**	**0.0368**	**0.0375**	**0.0564**	**0.0232**	**0.0288**

to extract more textual information beneficial for recommendation, allowing more refined item representations and user preference modeling.

4.4.2 Impact of Code Number. We analyze the impact of code number on the final recommendation performance of CCFRec by changing the number of sub-vectors in the vector quantization from 2 to 8. The results in Table 6 indicate that CCFRec consistently benefits from increasing the code number. The reason is that a larger code number allows CCFRec to capture textual semantics and model the item correlations more effectively in a broader representation space. However, increasing the number of codes also raises computational resource requirements. Therefore, it is essential to strike a balance between model performance and computational efficiency.

4.4.3 Impact of Code Type. To examine the impact of code types, we replace the original text-based semantic codes with three alternatives: random codes, codes derived from SASRec embeddings and mixed codes that combine equal proportions of text-based and SASRec-based codes. To ensure a fair comparison, the number of semantic codes in each case is kept consistent. As shown in Figure 2, the use of SASRec-based semantic codes leads to significant performance degradation. This is primarily because the collaborative information captured by SASRec-based codes is redundant, as CCFRec is already capable of learning it. Our results indicate that models perform better when using more text semantic codes (Text > Mix > SASRec), underscoring the importance of textual semantic codes. Interestingly, random codes outperform SASRec-based codes, likely due to their stronger regularization effect.

4.4.4 Impact of Loss Coefficients. α and β are key hyperparameters in CCFRec, controlling the importance of the masked code modeling (MCM) and masked sequence alignment (MSA) objectives. We evaluate their impact by varying their values within {0.2, 0.4, 0.6, 0.8} on the Instrument and Scientific datasets. The results for Recall@10 and NDCG@10 are shown in Figure 3. On the Instrument dataset, CCFRec's performance initially improves but declines as α increases from 0.2 to 0.8, as moderate α values enhance semantic fusion while excessive values disrupt sequential pattern learning. On the Scientific dataset, the impact of α varies with β, indicating MCM's influence is less predictable. For β, its effect depends on α, with the optimal value shifting based on α, highlighting the need to balance both hyperparameters for optimal performance.

4.4.5 Compatibility with ID-based Representations. In previous sections, we excluded unique item IDs and relied solely on text embeddings and their associated semantic codes to model items and user

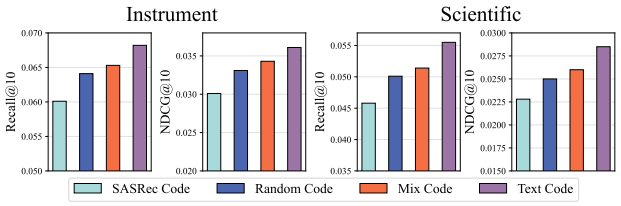

Figure 2: Performance comparison of different semantic codes. 'Mix Code' refers to a combination of half text-based codes and half SASRec-based codes.

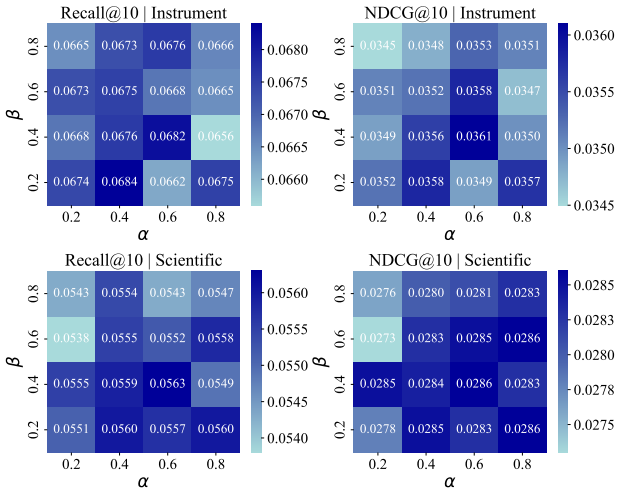

Figure 3: Performance comparison of different loss coefficients on Instrument and Scientific.

preferences. However, our proposed model, CCFRec, is also compatible with ID-based representations. Below, we present a parameter-efficient approach for integrating unique item ID representations into our framework. To begin with, we pre-train CCFRec using the original pipeline. Next, we introduce a raw item ID embedding table while freezing all other components of CCFRec. The unique ID embeddings are then summed up with the semantic-fused item representations and fed as input to the sequential recommender. The model is then fine-tuned according to the recommendation objective $\mathcal{L}_{\text{CE}}$, as defined in Eq. (5), until convergence. This fine-tuning process is highly efficient since only the item embeddings are trainable. The experimental results are summarized in Table 7. The incorporation of unique item IDs further enhances the performance of CCFRec. This improvement can be attributed to the incremental collaborative information provided by unique IDs, which enable CCFRec to better distinguish between items. Unlike semantic codes, unique IDs offer a more flexible representation space, allowing for finer-grained differentiation.

4.4.6 Analysis of the balance between collaborative and textual signals. To validate our model's ability to bridge the textual and collaborative gap, we visualize its top-2000 similar items via t-SNE and use blue and red colors to represent the collaboratively similar and textually similar items, respectively. We present the examples of

Table 7: Performance of CCFRec empowered with ID-based representations on four datasets.

Methods	Instrument		Scientific		Game		Baby	
	R@10	N@10	R@10	N@10	R@10	N@10	R@10	N@10
CCFRec	0.0682	0.0361	0.0555	0.0285	0.1042	0.0536	0.0455	0.0238
+ Item ID	**0.0706**	**0.0366**	**0.0562**	**0.0288**	**0.1063**	**0.0540**	**0.0471**	**0.0246**

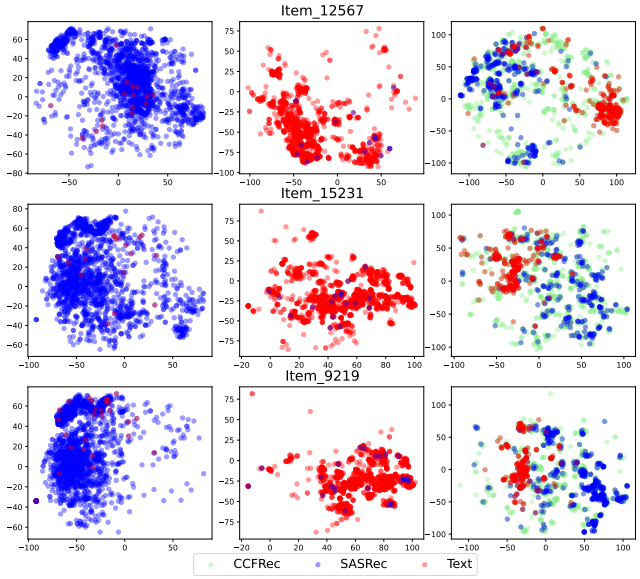

Figure 4: Visualization of similar items with different signals on Instrument. The left, middle, and right panels display t-SNE visualizations of the top-2000 similar items for a specific item derived from the SASRec embeddings, text embeddings, and CCFRec's fused representations, respectively.

three items in Figure 4. The results reveal that CCFRec achieves a balanced distribution of textually and collaboratively similar items, confirming its ability to harmonize both information effectively.

5 Related Work

5.1 Sequential Recommendation

Sequential recommendation focuses on modeling user preferences from historical behavior sequences to predict the next item of interest. Early studies [7, 27] treated user behavior as Markov chains, emphasizing item transition relationships. Inspired by the success of deep learning in sequence modeling, researchers applied various neural network architectures for sequential recommendation, including CNNs [30], RNNs [8, 29], and GNNs [1, 32]. Recently, Transformer-based methods [6, 16, 28] have shown significant success in sequential behavior modeling. Moreover, recent studies [2, 20, 31, 45] have proposed to improve the Transformer architecture to further enhance sequential recommender performance. For example, FMLP-Rec [44] replaces the self-attention layer with filter-enhanced MLPs to reduce noise during user preference modeling. However, these methods rely solely on IDs to capture correlations among items, which leads to issues such as the cold-start problem.

5.2 Text-Enhanced Recommendation

With the rapid development of various application platforms, a growing amount of item content information has become available. This has motivated researchers to explore its potential for improving the performance of sequential recommender systems. We refer to this approach as text-enhanced recommendation. Previous studies [5, 17, 21, 33, 35, 36] have typically utilized text embeddings encoded by pre-trained language models (PLMs) to enhance item representations and improve behavior modeling. For instance, FDSA [39] employs a dual-stream Transformer-based network to independently model item IDs and feature sequences. UniSRec [11] uses an MoE-enhanced adaptor to extract universal information from text embeddings through multi-domain pretraining. Another line of research [14, 18, 23, 25] utilizes the discrete codes derived from the text embeddings in sequential or generative recommendation. For instance, VQRec [9] applies vector quantization to transform text embeddings into discrete item IDs, leveraging these quantized representations to model user preferences. MMSR [13] uses clustering to decompose original modality features into discrete semantic codes, treating these indices as nodes to build a dense item relation graph. Unlike the above methods, in CCFRec, we leverage discrete semantic codes as bridges to achieve effective textual-collaborative semantic fusion.

6 Conclusion

In this work, we proposed CCFRec to achieve effective textual and collaborative semantic fusion for sequential recommendation. The core idea of our approach is to bridge the textual-collaborative semantic gap through semantic codes. For this purpose, we construct multi-view text embeddings and corresponding semantic codes derived from vector quantization for fine-grained semantic utilization. Next, we design a code-guided semantic-fusion module to enable flexible textual information extraction and integration. Furthermore, we devise an optimization method via code masking, comprising masked code modeling and masked sequence alignment, to model code correlations within individual items and enhance sequence modeling. Extensive experiments and in-depth analysis on four latest benchmarks demonstrate the effectiveness of our proposed CCFRec. In future work, we will leverage our method in other recommendation tasks (*e.g.,* multi-modal recommendation and multi-domain recommendation).

Acknowledgments

This work was partially supported by National Natural Science Foundation of China under Grant No. 92470205 and 62222215, Beijing Municipal Science and Technology Project under Grant No. Z231100010323009, and Beijing Natural Science Foundation under Grant No. L233008. Xin Zhao is the corresponding author.

References

[1] Jianxin Chang, Chen Gao, Yu Zheng, Yiqun Hui, Yanan Niu, Yang Song, Depeng Jin, and Yong Li. 2021. Sequential Recommendation with Graph Neural Networks. In *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021*, Fernando Diaz, Chirag Shah, Torsten Suel, Pablo Castells, Rosie Jones, and Tetsuya Sakai (Eds.). ACM, 378–387. doi:10.1145/3404835.3462968

[2] Huiyuan Chen, Yusan Lin, Menghai Pan, Lan Wang, Chin-Chia Michael Yeh, Xiaoting Li, Yan Zheng, Fei Wang, and Hao Yang. 2022. Denoising Self-Attentive Sequential Recommendation. In *RecSys '22: Sixteenth ACM Conference on Recommender Systems, Seattle, WA, USA, September 18 - 23, 2022*, Jennifer Golbeck, F. Maxwell Harper, Vanessa Murdock, Michael D. Ekstrand, Bracha Shapira, Justin Basilico, Keld T. Lundgaard, and Even Oldridge (Eds.). ACM, 92–101. doi:10.1145/3523227.3546788

[3] Yongjun Chen, Zhiwei Liu, Jia Li, Julian J. McAuley, and Caiming Xiong. 2022. Intent Contrastive Learning for Sequential Recommendation. In *WWW '22: The ACM Web Conference 2022, Virtual Event, Lyon, France, April 25 - 29, 2022*, Frédérique Laforest, Raphaël Troncy, Elena Simperl, Deepak Agarwal, Aristides Gionis, Ivan Herman, and Lionel Médini (Eds.). ACM, 2172–2182. doi:10.1145/3485447.3512090

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, Jill Burstein, Christy Doran, and Thamar Solorio (Eds.). Association for Computational Linguistics, 4171–4186. doi:10.18653/V1/N19-1423

[5] Jingtong Gao, Xiangyu Zhao, Muyang Li, Minghao Zhao, Runze Wu, Ruocheng Guo, Yiding Liu, and Dawei Yin. 2024. SMLP4Rec: An Efficient All-MLP Architecture for Sequential Recommendations. *ACM Trans. Inf. Syst.* 42, 3 (2024), 86:1–86:23. doi:10.1145/3637871

[6] Yongjing Hao, Tingting Zhang, Pengpeng Zhao, Yanchi Liu, Victor S. Sheng, Jiajie Xu, Guanfeng Liu, and Xiaofang Zhou. 2023. Feature-Level Deeper Self-Attention Network With Contrastive Learning for Sequential Recommendation. *IEEE Trans. Knowl. Data Eng.* 35, 10 (2023), 10112–10124. doi:10.1109/TKDE.2023.3250463

[7] Ruining He and Julian J. McAuley. 2016. Fusing Similarity Models with Markov Chains for Sparse Sequential Recommendation. In *IEEE 16th International Conference on Data Mining, ICDM 2016, December 12-15, 2016, Barcelona, Spain*, Francesco Bonchi, Josep Domingo-Ferrer, Ricardo Baeza-Yates, Zhi-Hua Zhou, and Xindong Wu (Eds.). IEEE Computer Society, 191–200. doi:10.1109/ICDM.2016.0030

[8] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2016. Session-based Recommendations with Recurrent Neural Networks. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, Yoshua Bengio and Yann LeCun (Eds.). http://arxiv.org/abs/1511.06939

[9] Yupeng Hou, Zhankui He, Julian J. McAuley, and Wayne Xin Zhao. 2023. Learning Vector-Quantized Item Representation for Transferable Sequential Recommenders. In *Proceedings of the ACM Web Conference 2023, WWW 2023, Austin, TX, USA, 30 April 2023 - 4 May 2023*, Ying Ding, Jie Tang, Juan F. Sequeda, Lora Aroyo, Carlos Castillo, and Geert-Jan Houben (Eds.). ACM, 1162–1171. doi:10.1145/3543507.3583434

[10] Yupeng Hou, Jiacheng Li, Zhankui He, An Yan, Xiusi Chen, and Julian J. McAuley. 2024. Bridging Language and Items for Retrieval and Recommendation. *CoRR* abs/2403.03952 (2024). doi:10.48550/ARXIV.2403.03952 arXiv:2403.03952

[11] Yupeng Hou, Shanlei Mu, Wayne Xin Zhao, Yaliang Li, Bolin Ding, and Ji-Rong Wen. 2022. Towards Universal Sequence Representation Learning for Recommender Systems. In *KDD '22: The 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, August 14 - 18, 2022*. ACM, 585–593. doi:10.1145/3534678.3539381

[12] Yupeng Hou, Jianmo Ni, Zhankui He, Noveen Sachdeva, Wang-Cheng Kang, Ed H. Chi, Julian J. McAuley, and Derek Zhiyuan Cheng. 2025. ActionPiece: Contextually Tokenizing Action Sequences for Generative Recommendation. *CoRR* abs/2502.13581 (2025). doi:10.48550/ARXIV.2502.13581 arXiv:2502.13581

[13] Hengchang Hu, Wei Guo, Yong Liu, and Min-Yen Kan. 2023. Adaptive Multi-Modalities Fusion in Sequential Recommendation Systems. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management, CIKM 2023, Birmingham, United Kingdom, October 21-25, 2023*, Ingo Frommholz, Frank Hopfgartner, Mark Lee, Michael Oakes, Mounia Lalmas, Min Zhang, and Rodrygo L. T. Santos (Eds.). ACM, 843–853. doi:10.1145/3583780.3614775

[14] Hengchang Hu, Qijiong Liu, Chuang Li, and Min-Yen Kan. 2024. Lightweight Modality Adaptation to Sequential Recommendation via Correlation Supervision. In *Advances in Information Retrieval - 46th European Conference on Information Retrieval, ECIR 2024, Glasgow, UK, March 24-28, 2024, Proceedings, Part I (Lecture Notes in Computer Science, Vol. 14608)*, Nazli Goharian, Nicola Tonellotto, Yulan He, Aldo Lipani, Graham McDonald, Craig Macdonald, and Iadh Ounis (Eds.). Springer, 123–139. doi:10.1007/978-3-031-56027-9_8

[15] Wenyue Hua, Shuyuan Xu, Yingqiang Ge, and Yongfeng Zhang. 2023. How to Index Item IDs for Recommendation Foundation Models. In *Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region, SIGIR-AP 2023, Beijing, China, November 26-28, 2023*, Qingyao Ai, Yiqin Liu, Alistair Moffat, Xuanjing Huang, Tetsuya Sakai, and Justin Zobel (Eds.). ACM, 195–204. doi:10.1145/3624918.3625339

[16] Wang-Cheng Kang and Julian J. McAuley. 2018. Self-Attentive Sequential Recommendation. In *IEEE International Conference on Data Mining, ICDM 2018, Singapore, November 17-20, 2018*. IEEE Computer Society, 197–206. doi:10.1109/ICDM.2018.00035

[17] Chang Liu, Xiaoguang Li, Guohao Cai, Zhenhua Dong, Hong Zhu, and Lifeng Shang. 2021. Noninvasive Self-attention for Side Information Fusion in Sequential Recommendation. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*. AAAI Press, 4249–4256. doi:10.1609/AAAI.V35I5.16549

[18] Enze Liu, Bowen Zheng, Cheng Ling, Lantao Hu, Han Li, and Wayne Xin Zhao. 2024. End-to-End Learnable Item Tokenization for Generative Recommendation. *CoRR* abs/2409.05546 (2024). doi:10.48550/ARXIV.2409.05546 arXiv:2409.05546

[19] Haibo Liu, Zhixiang Deng, Liang Wang, Jinjia Peng, and Shi Feng. 2023. Distribution-based Learnable Filters with Side Information for Sequential Recommendation. In *Proceedings of the 17th ACM Conference on Recommender Systems, RecSys 2023, Singapore, Singapore, September 18-22, 2023*, Jie Zhang, Li Chen, Shlomo Berkovsky, Min Zhang, Tommaso Di Noia, Justin Basilico, Luiz Pizzato, and Yang Song (Eds.). ACM, 78–88. doi:10.1145/3604915.3608782

[20] Mingrui Liu, Sixiao Zhang, and Cheng Long. 2024. Facet-Aware Multi-Head Mixture-of-Experts Model for Sequential Recommendation. *CoRR* abs/2411.01457 (2024). doi:10.48550/ARXIV.2411.01457 arXiv:2411.01457

[21] Qidong Liu, Xian Wu, Wanyu Wang, Yejing Wang, Yuanshao Zhu, Xiangyu Zhao, Feng Tian, and Yefeng Zheng. 2024. Large Language Model Empowered Embedding Generator for Sequential Recommendation. *CoRR* abs/2409.19925 (2024). doi:10.48550/ARXIV.2409.19925 arXiv:2409.19925

[22] Jianmo Ni, Gustavo Hernández Ábrego, Noah Constant, Ji Ma, Keith B. Hall, Daniel Cer, and Yinfei Yang. 2022. Sentence-T5: Scalable Sentence Encoders from Pre-trained Text-to-Text Models. In *Findings of the Association for Computational Linguistics: ACL 2022, Dublin, Ireland, May 22-27, 2022*. Association for Computational Linguistics, 1864–1874. doi:10.18653/V1/2022.FINDINGS-ACL.146

[23] Aleksandr V. Petrov and Craig Macdonald. 2024. RecJPQ: Training Large-Catalogue Sequential Recommenders. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining, WSDM 2024, Merida, Mexico, March 4-8, 2024*, Luz Angelica Caudillo-Mata, Silvio Lattanzi, Andrés Muñoz Medina, Leman Akoglu, Aristides Gionis, and Sergei Vassilvitskii (Eds.). ACM, 538–547. doi:10.1145/3616855.3635821

[24] Ruihong Qiu, Zi Huang, Hongzhi Yin, and Zijian Wang. 2022. Contrastive Learning for Representation Degeneration Problem in Sequential Recommendation. In *WSDM '22: The Fifteenth ACM International Conference on Web Search and Data Mining, Virtual Event / Tempe, AZ, USA, February 21 - 25, 2022*, K. Selcuk Candan, Huan Liu, Leman Akoglu, Xin Luna Dong, and Jiliang Tang (Eds.). ACM, 813–823. doi:10.1145/3488560.3498433

[25] Shashank Rajput, Nikhil Mehta, Anima Singh, Raghunandan Hulikal Keshavan, Trung Vu, Lukasz Heldt, Lichan Hong, Yi Tay, Vinh Q. Tran, Jonah Samost, Maciej Kula, Ed H. Chi, and Mahesh Sathiamoorthy. 2023. Recommender Systems with Generative Retrieval. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*. http://papers.nips.cc/paper_files/paper/2023/hash/20dcab0f14046a5c6b02b61da9f13229-Abstract-Conference.html

[26] Xubin Ren, Lianghao Xia, Yuhao Yang, Wei Wei, Tianle Wang, Xuheng Cai, and Chao Huang. 2024. SSLRec: A Self-Supervised Learning Framework for Recommendation. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining, WSDM 2024, Merida, Mexico, March 4-8, 2024*, Luz Angelica Caudillo-Mata, Silvio Lattanzi, Andrés Muñoz Medina, Leman Akoglu, Aristides Gionis, and Sergei Vassilvitskii (Eds.). ACM, 567–575. doi:10.1145/3616855.3635814

[27] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. 2010. Factorizing personalized Markov chains for next-basket recommendation. In *Proceedings of the 19th International Conference on World Wide Web, WWW 2010, Raleigh, North Carolina, USA, April 26-30, 2010*, Michael Rappa, Paul Jones, Juliana Freire, and Soumen Chakrabarti (Eds.). ACM, 811–820. doi:10.1145/1772690.1772773

[28] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 2019. BERT4Rec: Sequential Recommendation with Bidirectional Encoder Representations from Transformer. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019, Beijing, China, November 3-7, 2019*, Wenwu Zhu, Dacheng Tao, Xueqi Cheng, Peng Cui, Elke A. Rundensteiner, David Carmel, Qi He, and Jeffrey Xu Yu (Eds.). ACM, 1441–1450. doi:10.1145/3357384.3357895

[29] Yong Kiam Tan, Xinxing Xu, and Yong Liu. 2016. Improved Recurrent Neural Networks for Session-based Recommendations. In *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems, DLRS@RecSys 2016, Boston, MA, USA, September 15, 2016*. ACM, 17–22. doi:10.1145/2988450.2988452

[30] Jiaxi Tang and Ke Wang. 2018. Personalized Top-N Sequential Recommendation via Convolutional Sequence Embedding. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, WSDM 2018, Marina Del Rey, CA, USA, February 5-9, 2018*, Yi Chang, Chengxiang Zhai, Yan Liu, and Yoelle Maarek (Eds.). ACM, 565–573. doi:10.1145/3159652.3159656

[31] Hao Wang, Jianxun Lian, Mingqi Wu, Haoxuan Li, Jiajun Fan, Wanyue Xu, Chaozhuo Li, and Xing Xie. 2023. ConvFormer: Revisiting Transformer for Sequential User Modeling. *CoRR* abs/2308.02925 (2023). doi:10.48550/ARXIV.2308.02925 arXiv:2308.02925

[32] Shu Wu, Yuyuan Tang, Yanqiao Zhu, Liang Wang, Xing Xie, and Tieniu Tan. 2019. Session-Based Recommendation with Graph Neural Networks. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*. AAAI Press, 346–353. doi:10.1609/AAAI.V33I01.3301346

[33] Yunjia Xi, Weiwen Liu, Jianghao Lin, Xiaoling Cai, Hong Zhu, Jieming Zhu, Bo Chen, Ruiming Tang, Weinan Zhang, and Yong Yu. 2024. Towards Open-World Recommendation with Knowledge Augmentation from Large Language Models. In *Proceedings of the 18th ACM Conference on Recommender Systems, RecSys 2024, Bari, Italy, October 14-18, 2024*, Tommaso Di Noia, Pasquale Lops, Thorsten Joachims, Katrien Verbert, Pablo Castells, Zhenhua Dong, and Ben London (Eds.). ACM, 12–22. doi:10.1145/3640457.3688104

[34] Xu Xie, Fei Sun, Zhaoyang Liu, Shiwen Wu, Jinyang Gao, Jiandong Zhang, Bolin Ding, and Bin Cui. 2022. Contrastive Learning for Sequential Recommendation. In *38th IEEE International Conference on Data Engineering, ICDE 2022, Kuala Lumpur, Malaysia, May 9-12, 2022*. IEEE, 1259–1273. doi:10.1109/ICDE53745.2022.00099

[35] Yueqi Xie, Peilin Zhou, and Sunghun Kim. 2022. Decoupled Side Information Fusion for Sequential Recommendation. In *SIGIR '22: The 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, Madrid, Spain, July 11 - 15, 2022*, Enrique Amigó, Pablo Castells, Julio Gonzalo, Ben Carterette, J. Shane Culpepper, and Gabriella Kazai (Eds.). ACM, 1611–1621. doi:10.1145/3477495.3531963

[36] Lanling Xu, Zhen Tian, Bingqian Li, Junjie Zhang, Daoyuan Wang, Hongyu Wang, Jinpeng Wang, Sheng Chen, and Wayne Xin Zhao. 2024. Sequence-level Semantic Representation Fusion for Recommender Systems. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management, CIKM 2024, Boise, ID, USA, October 21-25, 2024*, Edoardo Serra and Francesca Spezzano (Eds.). ACM, 5015–5022. doi:10.1145/3627673.3680037

[37] Lanling Xu, Zhen Tian, Gaowei Zhang, Junjie Zhang, Lei Wang, Bowen Zheng, Yifan Li, Jiakai Tang, Zeyu Zhang, Yupeng Hou, Xingyu Pan, Wayne Xin Zhao, Xu Chen, and Ji-Rong Wen. 2023. Towards a More User-Friendly and Easy-to-Use Benchmark Library for Recommender Systems. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2023, Taipei, Taiwan, July 23-27, 2023*. ACM, 2837–2847. doi:10.1145/3539618.3591889

[38] Yaowen Ye, Lianghao Xia, and Chao Huang. 2023. Graph Masked Autoencoder for Sequential Recommendation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2023, Taipei, Taiwan, July 23-27, 2023*, Hsin-Hsi Chen, Wei-Jou (Edward) Duh, Hen-Hsen Huang, Makoto P. Kato, Josiane Mothe, and Barbara Poblete (Eds.). ACM, 321–330. doi:10.1145/3539618.3591692

[39] Tingting Zhang, Pengpeng Zhao, Yanchi Liu, Victor S. Sheng, Jiajie Xu, Deqing Wang, Guanfeng Liu, and Xiaofang Zhou. 2019. Feature-level Deeper Self-Attention Network for Sequential Recommendation. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, Sarit Kraus (Ed.). ijcai.org, 4320–4326. doi:10.24963/IJCAI.2019/600

[40] Wayne Xin Zhao, Yupeng Hou, Xingyu Pan, Chen Yang, Zeyu Zhang, Zihan Lin, Jingsen Zhang, Shuqing Bian, Jiakai Tang, Wenqi Sun, Yushuo Chen, Lanling Xu, Gaowei Zhang, Zhen Tian, Changxin Tian, Shanlei Mu, Xinyan Fan, Xu Chen, and Ji-Rong Wen. 2022. RecBole 2.0: Towards a More Up-to-Date Recommendation Library. In *CIKM*. ACM, 4722–4726.

[41] Wayne Xin Zhao, Shanlei Mu, Yupeng Hou, Zihan Lin, Yushuo Chen, Xingyu Pan, Kaiyuan Li, Yujie Lu, Hui Wang, Changxin Tian, Yingqian Min, Zhichao Feng, Xinyan Fan, Xu Chen, Pengfei Wang, Wendi Ji, Yaliang Li, Xiaoling Wang, and Ji-Rong Wen. 2021. RecBole: Towards a Unified, Comprehensive and Efficient Framework for Recommendation Algorithms. In *CIKM '21: The 30th ACM International Conference on Information and Knowledge Management, Virtual Event, Queensland, Australia, November 1 - 5, 2021*. ACM, 4653–4664.

[42] Bowen Zheng, Yupeng Hou, Hongyu Lu, Yu Chen, Wayne Xin Zhao, Ming Chen, and Ji-Rong Wen. 2024. Adapting Large Language Models by Integrating Collaborative Semantics for Recommendation. In *40th IEEE International Conference on Data Engineering, ICDE 2024, Utrecht, The Netherlands, May 13-16, 2024*. IEEE, 1435–1448. doi:10.1109/ICDE60146.2024.00118

[43] Kun Zhou, Hui Wang, Wayne Xin Zhao, Yutao Zhu, Sirui Wang, Fuzheng Zhang, Zhongyuan Wang, and Ji-Rong Wen. 2020. S3-Rec: Self-Supervised Learning for Sequential Recommendation with Mutual Information Maximization. In *CIKM '20: The 29th ACM International Conference on Information and Knowledge Management, Virtual Event, Ireland, October 19-23, 2020*, Mathieu d'Aquin, Stefan Dietze, Claudia Hauff, Edward Curry, and Philippe Cudré-Mauroux (Eds.). ACM, 1893–1902. doi:10.1145/3340531.3411954

[44] Kun Zhou, Hui Yu, Wayne Xin Zhao, and Ji-Rong Wen. 2022. Filter-enhanced MLP is All You Need for Sequential Recommendation. In *WWW '22: The ACM Web Conference 2022, Virtual Event, Lyon, France, April 25 - 29, 2022*, Frédérique Laforest, Raphaël Troncy, Elena Simperl, Deepak Agarwal, Aristides Gionis, Ivan Herman, and Lionel Médini (Eds.). ACM, 2388–2399. doi:10.1145/3485447.3512111

[45] Peilin Zhou, Qichen Ye, Yueqi Xie, Jingqi Gao, Shoujin Wang, Jae Boum Kim, Chenyu You, and Sunghun Kim. 2023. Attention Calibration for Transformer-based Sequential Recommendation. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management, CIKM 2023, Birmingham, United Kingdom, October 21-25, 2023*, Ingo Frommholz, Frank Hopfgartner, Mark Lee, Michael Oakes, Mounia Lalmas, Min Zhang, and Rodrygo L. T. Santos (Eds.). ACM, 3595–3605. doi:10.1145/3583780.3614785