Cross-Scale Collaboration between LLMs and Lightweight Sequential Recommenders with Domain-Specific Latent Reasoning

Yipeng Zhang[1], Xin Wang[12*], Hong Chen[1], Junwei Pan[3], Qian Li[3], Jun Zhang[3], Jie Jiang[3], Hong Mei[4], Wenwu Zhu[12*]

[1]Department of Computer Science and Technology, Tsinghua University
[2] Beijing National Research Center for Information Science and Technology, Tsinghua University
[3]Tencent Inc.
[4]MoE Key Lab of High Confidence Software Technologies, Peking University
{zhang-yp22, h-chen20}@mails.tsinghua.edu.cn, {xin_wang, wwzhu}@tsinghua.edu.cn,
{jonaspan, kathieqli, neoxzhang, zeus}@tencent.com, meih@pku.edu.cn

Abstract

Sequential recommendation aims to predict the next item based on historical interactions. To further enhance the reasoning capability in sequential recommendation, LLMs are employed to predict the next item or generate semantic IDs for item representation, given LLMs' extensive domain knowledge and reasoning ability. However, existing LLM-based methods suffer from two limitations. (i) The scarcity of recommendation data with reasoning paths makes it challenging to design suitable chain-of-thought prompting templates, and the full potential of LLMs' reasoning abilities remains underutilized. (ii) Upon obtaining semantic IDs, the LLMs and their representations are excluded from the subsequent recommendation model training, preventing downstream models from fully utilizing the rich semantic information encoded within these IDs. To address these issues, we propose a novel CoderRec framework, which is capable of fully exploiting the information encoded in semantic IDs to guide the recommendation process. Specifically, to address the problem of scarcity in reasoning path-augmented data, we introduce latent reasoning into sequential recommendation and treat the representation captured by the downstream model as domain-specific latent thought, enabling implicit logical inference without requiring explicit CoT annotations. To ensure that the downstream recommendation models are able to deeply leverage the semantic information within IDs, we propose a novel cross-scale model collaboration strategy, which employs cross-scale IDs and a two-phase approach to align LLM-derived semantics with recommendation objectives. Extensive experiments have shown the effectiveness of our proposed CoderRec framework. Codes will be available at https://github.com/defineZYP/CoderRec.

Introduction

Sequential recommendation aims to predict the next item users will interact with based on their historical behaviors, playing a crucial role in our daily lives (Wu et al. 2024; Boka, Niu, and Neupane 2024). To further enhance the capability of recommendation systems, various meta-information, such as item descriptions and user feedback, has been increasingly integrated (Zhou et al. 2020; Xie, Zhou, and Kim 2022; Chen et al. 2018; Kazi et al. 2020), which in turn poses new challenges to the representation ability of recommendation models due to the need for extensive domain knowledge.

With the rapid development of large language models (LLMs) in recent years, their powerful reasoning as well as text understanding abilities have attracted significant attention (Brown et al. 2020; Touvron et al. 2023; Bai et al. 2023). Leveraging their extensive domain knowledge and reasoning ability, some pioneering works introduce LLMs into recommendation tasks (Bao et al. 2023; Rajput et al. 2023; Liu et al. 2024; Zheng et al. 2024; Hong et al. 2025). Unlike traditional approaches that use numeric IDs for items, LLM-based methods typically represent items as text prompts with specific templates designed by human experts or enhanced by prompting strategies such as chain-of-thought (CoT) (Wei et al. 2022). Subsequently, the LLM processes item sequences and employs reasoning to generate more accurate and representative meta-information, and gets representations with extensive semantic information. To better adapt to the recommendation task, some approaches (Rajput et al. 2023; Liu et al. 2025) utilize vector quantization (Linde, Buzo, and Gray 2003), such as Residual Quantized Variational Autoencoder (RQVAE) (Zeghidour et al. 2021), to compress the text representations of items into several semantic IDs, enabling downstream models to operate more efficiently and accurately.

However, existing approaches suffer from two limitations. i) Unlike domains such as mathematical reasoning, which have been widely used to evaluate the reasoning abilities of LLMs, annotated data containing logical reasoning paths is rarely available for recommendation tasks. The rationale behind item recommendations is often subjective, flexible, and highly contextual, making it difficult to manually design universal reasoning templates or construct effective CoT prompts for different categories of items. ii) The downstream recommendation models cannot access the original

*Corresponding authors

representations extracted by the LLM. Besides, neither the LLM nor RQVAE participates in the training process of downstream tasks. As a result, the recommendation model is unable to fully utilize the complete information embedded in the semantic IDs, leading to sub-optimal results. Thus, existing LLM-based sequential recommendation methods fail to fully utilize the powerful capabilities of LLMs, and there are two critical challenges to overcome these problems. How to further explore the reasoning capabilities of LLMs in the absence of explicit reasoning data, and how to jointly leverage the broad knowledge of semantic IDs and domain-specific information of recommendation models.

To address these limitations, we propose **C**r**O**ss-Scale Model Collaboration for Sequential Recommendation with **D**omain-Specific Lat**E**nt **R**easoning (CoderRec) method, which is able to effectively leverage the information contained in the semantic IDs to guide the recommendation task. To better leverage the reasoning capabilities of LLMs in sequential recommendation scenarios where there is a lack of data with specific reasoning chains, we introduce latent reasoning into the recommendation system. Latent reasoning aims to activate the reasoning capabilities of LLMs in the latent space and can be seen as a kind of test-time scaling (Ruan et al. 2025; Saunshi et al. 2025; Zelikman et al. 2025; Tang et al. 2025), thereby not relying entirely on data with human-interpretable reasoning chains. Since recommendation tasks are not purely text-based, our goal is to improve item recommendations rather than overly emphasize human-readable reasoning processes. Therefore, we propose Domain-Specific Latent Reasoning for Sequential Recommendation, which enables LLMs to incorporate domain-specific knowledge from small-scale recommendation models into the latent reasoning process and makes the latent space suitable for recommendation tasks. To capture the complex information contained in the semantic IDs and extract latent thoughts related to recommendation, we propose the Cross-Scale Model Collaboration approach. The approach treats RQVAE as a bridge for the joint utilization of information between large-scale LLMs and small-scale downstream models, guiding the recommendation process. Specifically, for downstream recommendation tasks, we design an innovative embedding method to simultaneously handle the rich information brought by both item IDs and semantic IDs. Additionally, we introduce a two-stage training method and propose a cross-scale representation reconstruction auxiliary loss to help the downstream recommendation model better understand the diverse knowledge embedded in the semantic IDs generated by the LLM.

Our contributions can be summarized as follows:

- To the best of our knowledge, we are the first to introduce LLM-based latent reasoning into downstream sequential recommendation models with semantic IDs.
- We design a cross-scale model collaboration framework, which improves the model architecture for the downstream recommendation task.
- We propose a novel cross-scale representation reconstruction auxiliary loss to help the downstream recommendation model better understand the rich domain knowledge contained in semantic IDs.
- Extensive experiments are conducted to demonstrate the superiority of our proposed CoderRec method over existing baselines.

Related Works

Sequential Recommendation. Sequential recommendation aims to predict the next item that users want to interact with based on their historical interactions with item sequences, playing a significant role in daily life (Wang et al. 2019, 2023; Boka, Niu, and Neupane 2024; Wu et al. 2023, 2024). The earliest works predicted items using collaborative filtering (Liu, Lai, and Lee 2009) or Markov chain-based matrix factorization methods (He and McAuley 2016a). With the development of deep learning, many studies have attempted to model sequential information using various neural network architectures, such as RNNs (Cui et al. 2018; Donkers, Loepp, and Ziegler 2017), CNNs (Tang and Wang 2018), GNNs (Zhang et al. 2022, 2024; Ge et al. 2025), and Transformers (Kang and McAuley 2018; Sun et al. 2019; Fan et al. 2022; Tang et al. 2025; Zhang et al. 2023). SASRec (Kang and McAuley 2018) is one of the classic algorithms in this area, introducing self-attention into sequential recommendation to analyze the similarity between items, assisting the model to learn latent transition patterns for recommendation. STOSA (Fan et al. 2022) enhances stability in sequential recommendation by modeling item representations as distributions rather than simple vectors, accounting for the inherent uncertainty in item features. In addition to directly using sequence information and item IDs, some works have attempted to incorporate more meta-information to improve the representation of items or users, such as product descriptions, attributes, and user feedback during interactions (Zhou et al. 2020; Xie, Zhou, and Kim 2022; Kazi et al. 2020). However, not all forms of meta-information are effective. On the one hand, not all meta-information has an explicit correlation with item sequences, often requiring reasoning to uncover these relationships. On the other hand, incorporating meta-information means the requirements to deal with a more diverse set of domain knowledge, which exceeds the capacity of traditional recommendation models.

LLM-based Recommendation. With the development of large language models, their powerful reasoning abilities and extensive domain knowledge have attracted significant attention. Several studies have attempted to serve LLMs as recommenders (Bao et al. 2023; Zhang et al. 2025; Bao et al. 2025; Lin et al. 2024; Zheng et al. 2024; Hong et al. 2025; Wang et al. 2025; Wu et al. 2025), which design prompts or fine-tune LLMs to obtain domain-specific models suitable for recommendation tasks. CoLLM (Zhang et al. 2025) improves the recommendation performance by extracting and translating collaborative information into the textual space of LLMs. LC-Rec (Zheng et al. 2024) introduces a learnable item ID and integrates language and collaborative semantics through various tasks to enhance recommender systems. EAGER-LLM (Hong et al. 2025) takes a further step and incorporates both endogenous and exogenous behavioral and semantic information to assist LLMs recommendation tasks.

Other methods treat LLMs as embedders, utilizing their domain knowledge to capture item representations, which are then passed to downstream recommendation models for further processing (Liu et al. 2024; Rajput et al. 2023; Liu et al. 2025). LLM-ESR (Liu et al. 2024) leverages LLMs to capture embeddings and inputs them into downstream sequential recommendation models, employing dual-view modeling to utilize the rich semantic information. TIGER (Rajput et al. 2023) compresses the representations captured by LLMs through RQVAE, obtaining semantic IDs which are then used for training downstream tasks. However, after extracting semantic IDs, these methods do not involve LLMs or RQVAE in the recommendation training process. As a result, the downstream tasks fail to fully capture the complex semantic information, leading to suboptimal results.

LLM-based Reasoning As parameter scaling surpasses a certain threshold, the reasoning abilities of LLMs exhibit significant improvements and deeper potential (Brown et al. 2020; Touvron et al. 2023; Bai et al. 2023). Consequently, several studies (Wei et al. 2022; Yao et al. 2023; Li 2024; Zelikman et al. 2022) have attempted to further enhance the reasoning capacity of models through various techniques, such as test-time scaling, to make LLM outputs more logical and accurate. Chain-of-Thought (CoT) (Wei et al. 2022) is one of the pioneering works. By carefully designing prompts and meticulously detailing the expected reasoning steps, CoT guides LLMs through the reasoning process, achieving remarkable success. Following this, approaches such as ToT (Yao et al. 2023) and GoT (Li 2024) have introduced further refinements. However, not all tasks come with abundant data that contains detailed reasoning chains. The STaR (Zelikman et al. 2022) method addresses this issue by iteratively training LLMs, allowing the models to select and augment data autonomously, significantly improving the performance of LLMs. Nevertheless, STaR still requires an initial dataset with reasoning chains and corresponding reasoning templates first to train the LLM. Latent Reasoning (Ruan et al. 2025; Saunshi et al. 2025; Geiping et al. 2025; Zelikman et al. 2025), a novel approach for test-time scaling, enhances LLM performance on reasoning tasks without the need for any pre-existing dataset containing reasoning chains. The Looped Transformer (Saunshi et al. 2025) provides both theoretical and empirical evidence that, under certain conditions, latent reasoning can achieve the same effectiveness as the meticulously designed CoT. Quiet-STaR (Zelikman et al. 2025) assumes that a latent thought exists between each pair of adjacent tokens and dynamically adjusts the influence of this latent thought on the final output using parallel generation and learnable weights.

CoderRec

In this section, we present a detailed description of the proposed CoderRec method, which enhances the performance of the sequential recommendation through latent reasoning. The framework of CoderRec is illustrated in Figure 1. We first introduce the necessary background and define the notations in the Preliminary. Next, we will describe the Cross-Scale Model Collaboration approach, which captures domain-specific latent thoughts and facilitates downstream recommendation by jointly leveraging information from large-scale LLMs and small-scale recommendation models. Following this, we elaborate on the Domain-Specific Latent Reasoning mechanism and explain how the previously extracted latent thoughts are utilized to stimulate the reasoning ability of the LLM and guide the downstream tasks.

Preliminary

In this subsection, we first introduce the sequential recommendation task and define the notations used, followed by an overview of TIGER (Rajput et al. 2023).

Sequential Recommendation. Suppose we have a set of items $\mathcal{V} = \{v_1, v_2, ..., v_{|\mathcal{V}|}\}$ and a set of users $\mathcal{U} = \{u_1, u_2, ..., u_{|\mathcal{U}|}\}$. For each user $u \in \mathcal{U}$, we can organize the interaction behaviors of that user into a sequence in chronological order as $\mathcal{I}^u = [v_1^u, v_2^u, ..., v_M^u]$, where $v_i^u \in \mathcal{V}$ is the item involved in the i-th interaction of user u, M is the maximum sequence length. The purpose of sequential recommendation is to predict the item v_t with which a certain user u is most likely to interact next based on previous interactions $|\mathcal{I}|$.

TIGER. TIGER introduces language models into sequential recommendation. For each item $v \in \mathcal{V}$, it collects textual information such as the item's title and brand, denoted as p, and encodes it into the textual representation space as $h_{LM} = LM(p)$. This representation is then utilized in the downstream recommendation task. However, the dimensionality D of the textual space is extremely larger than the dimensionality d typically required in recommendation scenarios. Besides, items often share similar attributes. For example, some items may have the same brands. It is unnecessary for all item representations to be entirely independent. Thus TIGER employs RQVAE to compress the representations $\{h_{LM}^i\}_{i=1}^{|\mathcal{V}|}$ and extract semantic IDs.

RQVAE is a kind of vector quantization. It contains a DNN-based encoder $\mathcal{E}(\cdot)$ and decoder $\mathcal{D}(\cdot)$. Besides, there are K vector quantization layers between $\mathcal{E}$ and $\mathcal{D}$, each layer k maintains a codebook $C_k = \{c_k^1, c_k^2, ..., c_k^N\}$, represents the residual error of the ground truth value and the predicted code of the previous layer. Thus, for each item v, TIGER obtains a K-dimensional semantic ID $[s_1, s_2, ..., s_K]$ as follows:

$$s_k = arg\min_n ||r_{k-1} - c_k^n||, \tag{1}$$

$$r_k = r_{k-1} - c_k^{s_k}, \tag{2}$$

$$r_0 = \mathcal{E}(h_{LM}). \tag{3}$$

To prevent codebook collisions, where different items are assigned the same semantic IDs, TIGER introduces an additional dimension to the semantic ID, denoted as s_{K+1}, which indicates that the item is the s_{K+1}-th item associated with the given ID $[s_1, s_2, ..., s_K]$. Once the semantic IDs of items are constructed, TIGER uses them as a replacement for the original item indexes in the downstream sequential recommendation task and trains an embedder for each dimensionality of the semantic IDs.

It can be observed that the downstream task does not directly receive any information produced by the language

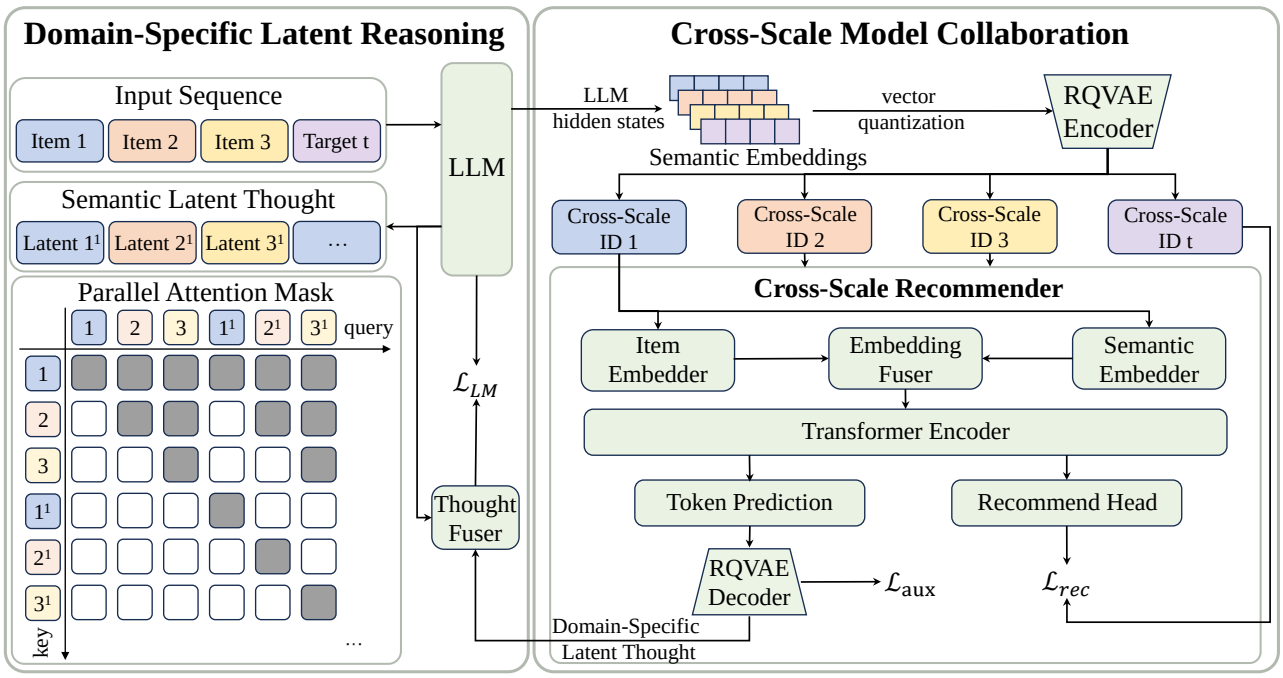

Figure 1: Framework of CoderRec. The proposed method consists of two main components. To further leverage the reasoning capabilities and domain knowledge of LLMs, we introduce Latent Reasoning into sequential recommendation. To capture the domain-specific latent thought of the downstream recommendation tasks, we propose a Cross-Scale Model Collaboration mechanism. This mechanism enhances the performance of the recommendation model by incorporating information produced by models at different scales and employing multiple task heads for joint optimization.

model. During the information compression process with RQVAE, potential information loss may prevents the recommendation model from fully leveraging the rich information contained in the semantic IDs.

Cross-Scale Model Collaboration

To fully exploit the extensive domain knowledge embedded in the semantic IDs derived from LLMs, we propose a Cross-Scale Model Collaboration method. We modify the architecture of the downstream recommendation model and design a novel auxiliary loss to guide the training process.

In contrast to previous work, we do not discard the original item ID information. Instead, we jointly model both the item ID v and the semantic IDs $[s_1^v, s_2^v, ..., s_{K+1}^v]$ as Cross-Scale IDs to construct item representations. Although semantic IDs have encoded diverse knowledge, the presence of potential codebook collisions necessitates the addition of an extra semantic ID dimension as a disambiguation tag. This means that even completely different items are likely to share at least $1/(K+1)$ same information. In practice, only $K = 3$ to 4 RQVAE layers are typically sufficient to cover the entire item set, which makes the problem more significant. As a result, it reduces the downstream models' ability to utilize the semantic information fully and hinders their capacity to distinguish and recommend diverse items effectively.

For the sequence $\mathcal{I} = \{(v_i, [s_1^{v_i}, ..., s_{K+1}^{v_i}])\}_{i=1}^M$, we first embed the item IDs using the traditional approach as follows:

$$e_{ID} = \text{Embedder}_{ID}(\{v_i\}_{i=1}^M) + e_{pos}, \tag{4}$$

where Embedder_{ID} is the embedder and e_{pos} is the position embedding. For the semantic IDs, we maintain an embedder of size $(K+1) \times N$, where N denotes the codebook size. For each k, the corresponding representation is given by:

$$e_k = \text{Embedder}_{SEM}(\{s_k^{v_i}\}_{i=1}^M + N \times (k-1)). \tag{5}$$

Subsequently, we concatenate the representations and apply a linear Embedding Fuser to obtain the final representation as:

$$e_u = \text{Fuser}([e_{ID}, e_1, e_2, ..., e_{K+1}]), \tag{6}$$

where $e \in \mathbb{R}^{M \times d}$, d is the dimension of the hidden states. Subsequently, we learn the representations using a transformer-based architecture and get $h_u = \text{Transformer}(e_u) \in \mathbb{R}^{M \times d}$. The representations are expected to satisfy the following two properties: (i) they are capable of accurately predicting the next item in the sequence, and (ii) they could effectively capture the domain knowledge encoded in the semantic IDs by the LLM. To achieve this, we design two task heads, a recommendation head and a token prediction head.

For the recommendation head, we first calculate the predicted scores for items $R_u = h_u E_{ID}^T$, where $E_{ID} \in \mathbb{R}^{|\mathcal{V}| \times d}$ is the ID embedding matrix. Thus, the element $r_{i,v}^u$ of R_u represents the predicted score of item v at the i-th position

of the sequence. We apply the cross-entropy loss to the recommendation head as:

$$\mathcal{L}_{rec} = -\mathbb{E}_{u\in\mathcal{U},i}[\log \frac{\exp(r^u_{i,v_i})}{\sum_{v\in\mathcal{V}}\exp(r^u_{i,v})}]. \tag{7}$$

For the token prediction head, the purpose is to generate the semantic IDs of both the current sequence and the next item, and then reconstruct them using the RQVAE decoder. This enables the model to improve its capability to recover the semantic information in the original text space of LLM. We utilize an MLP as the token prediction head, and the process can be formulated as:

$$[\hat{s}^{v_i}_1, \hat{s}^{v_i}_2, ..., \hat{s}^{v_i}_{(K+1)}] = \text{MLP}(h_{u,i}), \tag{8}$$

$$\hat{h}^{v_i}_{LM} = \mathcal{D}([\hat{s}^{v_i}_1, \hat{s}^{v_i}_2, ..., \hat{s}^{v_i}_{(K+1)}]), \tag{9}$$

where $h_{u,i}$ is the i-th row of h_u. To learn the token prediction head, we propose the cross-scale representation reconstruction auxiliary loss as:

$$\mathcal{L}_{aux} = \sum_{u\in\mathcal{U},i}[||\hat{h}^{v_i}_{LM} - h^{v_i}_{LM}||^2_2]. \tag{10}$$

Domain-Specific Latent Reasoning

At this point, we are able to effectively leverage the complex information encoded in the semantic IDs. However, to fully exploit the reasoning capabilities of LLMs while lacking data with reasoning chains and enhance the relevance between the information contained in semantic IDs and the downstream task, further investigation is still required. Inspired by Quiet-STaR (Zelikman et al. 2025), we introduce Domain-Specific Latent Reasoning to the downstream task, utilizing feedback signals from the recommendation model as domain-specific latent thoughts to guide and enhance the recommendation process.

For an item sequence $\mathcal{I} = \{v_1, v_2, ..., v_M\}$, we treat each item as a token and assume that the transitions between items are influenced by a latent thought l^m, which serves as the underlying reason behind a user's decision to purchase subsequent items, i.e., $v_1 \wedge v_2 \wedge ... \wedge v_m \wedge l^m \rightarrow v^{m+1}$. In traditional LLM training and inference, even we have to calculate the representation $h_{LM} \in \mathbb{R}^{M\times D}$, only the last row of the representation $h^M_{LM} \in \mathbb{R}^D$ is used to predict the next token v_{M+1}, while the other rows are discarded since the previous M tokens are already determined. This results in substantial computational waste. However, these representations, reflecting the model's predictions under earlier token conditions $v_m, m < M$, can be viewed as latent thoughts contributing to the prediction of the next token v_{m+1}.

The basic pipeline is shown in Figure 2. We introduce a special token $< think >$ to explicitly denote the intention to perform latent reasoning. The approach recurrently extends and feeds the sequence into the LLM to obtain more accurate latent thoughts. Following the design of Quiet-STaR, we adopt a parallel generation strategy to efficiently produce latent thoughts corresponding to different tokens. The core idea lies in constructing a carefully designed parallel attention mask, which ensures that each latent token l^m_j only has the attention score with the previous token in the same latent

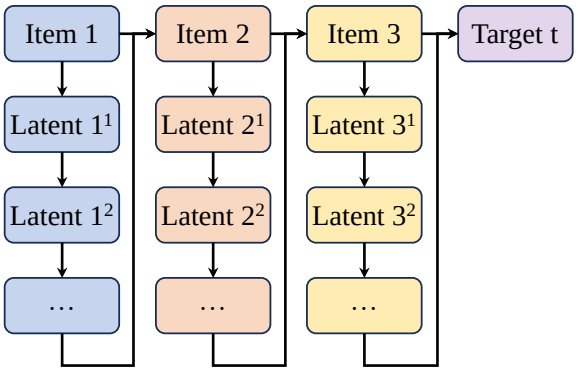

Figure 2: Basic pipeline of latent reasoning. For each token i, latent reasoning utilizes the representation at the corresponding position to predict a sequence of latent thought tokens $\{i^1, i^2, ...\}$, which is then used to assist the prediction of the subsequent token $i + 1$.

thought $l^m_{j'}, j' < j$, and the tokens $v_{m'}, m' < m$. To obtain high-quality latent thoughts within a limited number of steps and prevent the LLM's reasoning from becoming overly divergent, we incorporate feedback representations from the downstream recommendation tasks $\hat{h}_{LM}$ as domain-specific latent thoughts to constrain and guide the associative direction of the LLM.

Additionally, we introduce a Thought Fuser to learn and integrate the raw outputs of the LLM h, the Latent Thoughts l, and the Domain-Specific Latent Thoughts $\hat{h}_{LM}$:

$$[w_1, w_2] = \text{MLP}([h, l, h_{LM}]), \tag{11}$$

$$\hat{h} = (1 - w_1 - w_2)h + w_1 l + w_2 h_{LM}. \tag{12}$$

To ensure training stability in the early stages, we initialize the Thought Fuser with zeros, which guarantees that $\hat{h} = h$ at the beginning. Meanwhile, we apply a loss function similar to $\mathcal{L}_{rec}$ to guide the LLM's training and evolution toward the domain-specific direction:

$$r^u_{i,v} = \hat{h}{h^v_{LM}}^T, \tag{13}$$

$$\mathcal{L}_{LM} = -\mathbb{E}_{u\in\mathcal{U},i}[\log \frac{\exp(r^u_{i,v_i})}{\sum_{v\in\mathcal{V}}\exp(r^u_{i,v})}]. \tag{14}$$

Training Strategy

Due to the significant gap between the representation spaces of the recommendation model and the LLM, direct end-to-end training is not reasonable. Therefore, we first perform a warm-up stage on the downstream recommendation task, where the training objective is $\mathcal{L}_{warm} = \mathcal{L}_{rec} + \lambda_1\mathcal{L}_{aux}$. Subsequently, we jointly train the recommendation model and the LLM, with the training objective as $\mathcal{L} = \mathcal{L}_{rec} + \lambda_1\mathcal{L}_{aux} + \lambda_2\mathcal{L}_{LM}$. λ_1 and λ_2 are hyperparameters.

Experiments

We have conducted extensive experiments to show the performance of the CoderRec method. We first give a brief in-

troduction to the experimental setup, followed by the quantitative results of the proposed method. Ablation studies further demonstrate the effectiveness of each proposed module.

Experiments Settings

Datasets. To ensure a fair comparison, we follow the dataset selection and pre-processing protocols of previous works (Rajput et al. 2023; Zheng et al. 2024; Hong et al. 2025). Specifically, we conduct experiments on the Beauty, Sports, and Instruments of the Amazon dataset (He and McAuley 2016b; McAuley et al. 2015; Ni, Li, and McAuley 2019) under the 5-core setting, with detailed statistics provided in Table 2. We evaluate the performance using Recall@K and NDCG@K, where $K \in \{5, 10\}$.

Hyperparameters. We employ the AdamW (Loshchilov and Hutter 2017) optimizer to optimize the model. For the optimizer's hyperparameters, we perform a grid search over the learning rate $lr \in \{0.01, 0.001, 0.0001\}$, weight decay $\lambda_w \in \{0.01, 0.001, 0.0001\}$, and weights of auxiliary losses $\lambda_1, \lambda_2 \in \{0.1, 0.01, 0.001, 0.0001\}$. For the hyperparameters of the model architectures of the sequential recommendation tasks, we also perform a grid search over the number of layers $n_l \in \{1, 2, 3\}$, number of attention heads $n_h \in \{1, 2, 4\}$, dropout rate $p \in \{0.3, 0.4, 0.5, 0.6, 0.7\}$, and hidden size $d \in \{32, 64, 128\}$. For all of the datasets, we set the max length of the sequences as $M = 20$.

Baselines. Since our method is highly related to sequential recommendation, LLMs, and reasoning, we select the following baselines for comprehensive comparison:

- SASRec (Kang and McAuley 2018) introduces self-attention to model item sequences and improves the performance of the sequential recommendation task.
- Bert4Rec (Sun et al. 2019) utilizes a bidirectional transformer architecture to predict the next item.
- Sˆ3-Rec (Zhou et al. 2020) utilizes mutual information maximization to train the recommendation model and learn the relation between items.
- ReaRec (Tang et al. 2025) introduces latent reasoning to the transformer-based sequential recommender and utilizes a recurrent manner to learn the reasoning thoughts.
- LLM-ESR (Liu et al. 2024) proposes a dual-view modeling approach to assist the LLM-based sequential recommendation task.
- TIGER (Rajput et al. 2023) captures the semantic IDs from language models with RQVAE, which are then employed to learn the relevance between items.
- LC-Rec (Zheng et al. 2024) adopts a semantic ID-based approach for item modeling. In addition, it leverages the reasoning capabilities of LLMs by designing specific prompts and predicting user preferences during training, and employs a multi-task learning framework to assist the LLM in making recommendations.
- EAGER-LLM (Hong et al. 2025) proposes a decoder-only framework that integrates with both endogenous and exogenous behavioral information. Besides, it proposes an annealing adapter to balance the recommendation performance and the comprehension capabilities of LLMs.

Main Results

We report the recommendation performances of different methods in Table 1.

Compared to traditional recommendation methods such as SASRec and Bert4Rec, language model-based approaches like TIGER and LC-Rec significantly enhance recommendation performance. This improvement stems from the fact that the items are diverse and different, potentially associated with various functionalities, tags, and brand-related semantic information. Trained on large-scale textual data, LLMs possess extensive domain knowledge and have a strong ability to understand such semantic information. It is beneficial for the recommendation tasks. By employing Cross-Scale Model Collaboration, our method effectively leverages the semantic representations encoded in LLMs, thus achieving superior performance than other baselines.

Appropriate reasoning can enhance the performance of both large and small-scale models. On the one hand, ReaRec significantly outperforms traditional sequential recommendation models such as SASRec and Bert4Rec. Even without leveraging any enhancements from LLMs, ReaRec achieves performance comparable to advanced LLM-based methods like TIGER, LC-Rec, and EAGER-LLM on certain datasets. Meanwhile, LC-Rec and EAGER-LLM design various prompt templates to enable further reasoning over the original data. Thus, they can obtain more informative texts from LLM inference, such as users' purchasing preferences. By incorporating these enriched texts into a multi-task learning framework, they achieve considerably better results than methods like LLM-ESR and TIGER, which directly extract information from raw textual data. Our method introduces domain-specific latent reasoning, enabling the LLM to perform self-reflection and infer latent thoughts that guide token transitions. In addition, we constrain the training process by the domain-specific knowledge provided by the downstream task, ultimately leading to the best performance.

Furthermore, the experimental results of both ReaRec and our method suggest that latent reasoning, which can be viewed as a kind of test-time scaling, demonstrates consistent effectiveness for both large and small models. It helps unlock the inherent potential of the models, thus enhancing the performance in downstream tasks.

Ablation Studies

Proposed Modules We conduct ablation studies on the proposed modules using the Beauty dataset, and the results are presented in Table 3.

It can be observed that all the proposed modules contribute to performance improvements in the sequential recommendation task. Specifically, the performance drops significantly in the case of w/o item ID, indicating that the domain knowledge captured by large-scale LLMs solely through semantic IDs is insufficient to fully satisfy the requirements of the downstream task. Nevertheless, w/o item ID still outperforms the TIGER method, suggesting that leveraging reconstruction techniques and latent reasoning can enhance the model's ability to utilize the domain knowledge embedded in semantic IDs, thereby benefiting sequen-

Dataset	Metric	SASRec	Bert4Rec	Sˆ3-Rec	ReaRec	LLM-ESR	TIGER	LC-Rec	EAGER -LLM	Ours
Beauty	R@5	0.0387	0.0203	0.0387	0.0541	0.0315	0.0454	0.0482	<u>0.0548</u>	**0.0607**
	R@10	0.0605	0.0347	0.0647	<u>0.0743</u>	0.0445	0.0648	0.0681	**0.0830**	**0.0830**
	N@5	0.0249	0.0124	0.0244	0.0364	0.0224	0.0321	0.0327	<u>0.0369</u>	**0.0440**
	N@10	0.0318	0.0170	0.0327	0.0428	0.0267	0.0384	0.0409	<u>0.0459</u>	**0.0511**
Sports	R@5	0.0233	0.0115	0.0251	0.0369	0.0166	0.0264	0.0304	<u>0.0373</u>	**0.0403**
	R@10	0.0350	0.0191	0.0385	0.0524	0.0301	0.0400	0.0451	<u>0.0569</u>	**0.0595**
	N@5	0.0154	0.0075	0.0161	<u>0.0255</u>	0.0098	0.0181	0.0196	0.0251	**0.0280**
	N@10	0.0192	0.0099	0.0204	0.0305	0.0141	0.0225	0.0246	<u>0.0315</u>	**0.0342**
Instruments	R@5	0.0751	0.0671	0.0863	0.0986	0.0803	0.0863	0.0964	<u>0.0991</u>	**0.1174**
	R@10	0.0947	0.0822	0.1136	0.1142	0.1007	0.1064	0.1177	<u>0.1224</u>	**0.1355**
	N@5	0.0627	0.0560	0.0626	0.0850	0.0661	0.0738	0.0819	<u>0.0851</u>	**0.0954**
	N@10	0.0690	0.0608	0.0714	0.0901	0.0722	0.0803	0.0890	<u>0.0926</u>	**0.1013**

Table 1: The performance comparison between baselines and our proposed CoderRec method. R@K and N@K correspond to metrics Recall@K and NDCG@K. The best results are highlighted in bold, and the second-optimal results are underlined.

Dataset	Users	Items	Interactions	Sparsity
Beauty	22363	12101	198360	0.00073
Sports	35598	18357	296175	0.00045
Instruments	24772	9922	206153	0.00084

Table 2: Statistics of the datasets

Modules	R@5	R@10	N@5	N@10
w/o item ID	0.0543	0.0746	0.0387	0.0454
w/o $\mathcal{L}_{aux}$	0.0597	0.0812	0.0426	0.0495
w/o reasoning	0.0600	0.0806	0.0427	0.0494
full	0.0607	0.0830	0.0440	0.0511

Table 3: The effectiveness of each proposed module, where w/o item ID means performance without using item IDs, w/o $\mathcal{L}_{aux}$ means performance without cross-scale reconstruction auxiliary loss, and w/o reasoning means performance without domain-specific latent reasoning.

tial recommendation. The results of w/o $\mathcal{L}_{aux}$ and w/o reasoning also support the observation.

Auxiliary Weights We also analyze the impact of different auxiliary loss weights proposed in our method on the Beauty dataset, with the results summarized in Table 4.

It can be observed that all the auxiliary losses have a positive impact on the performance of the sequential recommendation. However, it is worth noting that when λ_2 becomes too large, the overall performance not only fails to improve but even degrades. This phenomenon can be attributed to two primary reasons. First, the LLM space contains abundant domain knowledge, some of which may be irrelevant to the recommendation task, thereby introducing a semantic gap between the LLM and the recommendation model. Therefore, when the weights are large, small-scale models may struggle to distinguish useful information, leading

Weights	R@5	R@10	N@5	N@10
$\lambda_1 = 0.1$	0.0607	**0.0830**	0.0440	**0.0511**
$\lambda_1 = 0.01$	**0.0608**	0.0825	**0.0441**	0.0507
$\lambda_1 = 0.001$	0.0598	0.0813	0.0435	0.0504
$\lambda_1 = 0.0001$	0.0599	0.0799	0.0437	0.0501
$\lambda_1 = 0.0$	0.0597	0.0812	0.0426	0.0495
$\lambda_2 = 0.1$	0.0591	0.0787	0.0430	0.0493
$\lambda_2 = 0.01$	**0.0611**	0.0816	**0.0442**	0.0508
$\lambda_2 = 0.001$	0.0607	**0.0830**	0.0440	**0.0511**
$\lambda_2 = 0.0001$	0.0602	0.0814	0.0437	0.0506
$\lambda_2 = 0.0$	0.0600	0.0806	0.0427	0.0494

Table 4: The effectiveness of different auxiliary weights.

to sub-optimal results. Second, from a numerical perspective, the representations from the LLM have significantly larger dimensions and value ranges compared to those in the recommendation space. Consequently, $\mathcal{L}_{LM}$ tends to dominate the gradients, potentially destabilizing the optimization process. That explains the necessity of the warmup phase, where we perform reconstruction to align the semantic spaces of large-scale and small-scale models via RQVAE, thereby enhancing both performance and robustness.

Conclusion

In this paper, we introduce latent reasoning into sequential recommendation and propose a novel CoderRec framework. Different from existing methods, CoderRec adopts Cross-Scale Model Collaboration that jointly leverages the complementary strengths of large-scale LLMs and small-scale models. Furthermore, we treat the representations learned by the downstream recommendation model as domain-specific latent thoughts, which are used to guide the training process of LLMs for improved recommendation performance. Extensive experiments demonstrate the superior effectiveness of our proposed CoderRec framework.

Acknowledgements

This work was supported by the National Key Research and Development Program of China No.2023YFF1205001, National Natural Science Foundation of China (No. 62222209), Beijing National Research Center for Information Science and Technology under Grant No. BNR2023TD03006, and Beijing Key Lab of Networked Multimedia.

References

Bai, J.; Bai, S.; Chu, Y.; Cui, Z.; Dang, K.; Deng, X.; Fan, Y.; Ge, W.; Han, Y.; Huang, F.; et al. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Bao, K.; Zhang, J.; Wang, W.; Zhang, Y.; Yang, Z.; Luo, Y.; Chen, C.; Feng, F.; and Tian, Q. 2025. A bi-step grounding paradigm for large language models in recommendation systems. *ACM Transactions on Recommender Systems*, 3(4): 1–27.

Bao, K.; Zhang, J.; Zhang, Y.; Wang, W.; Feng, F.; and He, X. 2023. Tallrec: An effective and efficient tuning framework to align large language model with recommendation. In *Proceedings of the 17th ACM conference on recommender systems*, 1007–1014.

Boka, T. F.; Niu, Z.; and Neupane, R. B. 2024. A survey of sequential recommendation systems: Techniques, evaluation, and future directions. *Information Systems*, 125: 102427.

Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901.

Chen, X.; Xu, H.; Zhang, Y.; Tang, J.; Cao, Y.; Qin, Z.; and Zha, H. 2018. Sequential recommendation with user memory networks. In *Proceedings of the eleventh ACM international conference on web search and data mining*, 108–116.

Cui, Q.; Wu, S.; Liu, Q.; Zhong, W.; and Wang, L. 2018. MV-RNN: A multi-view recurrent neural network for sequential recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 32(2): 317–331.

Donkers, T.; Loepp, B.; and Ziegler, J. 2017. Sequential user-based recurrent neural network recommendations. In *Proceedings of the eleventh ACM conference on recommender systems*, 152–160.

Fan, Z.; Liu, Z.; Wang, Y.; Wang, A.; Nazari, Z.; Zheng, L.; Peng, H.; and Yu, P. S. 2022. Sequential recommendation via stochastic self-attention. In *Proceedings of the ACM web conference 2022*, 2036–2047.

Ge, C.; Wang, X.; Zhang, Z.; Qin, Y.; Chen, H.; Wu, H.; Zhang, Y.; Yang, Y.; and Zhu, W. 2025. Behavior importance-aware graph neural architecture search for cross-domain recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 11708–11716.

Geiping, J.; McLeish, S.; Jain, N.; Kirchenbauer, J.; Singh, S.; Bartoldson, B. R.; Kailkhura, B.; Bhatele, A.; and Goldstein, T. 2025. Scaling up test-time compute with latent reasoning: A recurrent depth approach. *arXiv preprint arXiv:2502.05171*.

He, R.; and McAuley, J. 2016a. Fusing similarity models with markov chains for sparse sequential recommendation. In *2016 IEEE 16th international conference on data mining (ICDM)*, 191–200. IEEE.

He, R.; and McAuley, J. 2016b. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *proceedings of the 25th international conference on world wide web*, 507–517.

Hong, M.; Xia, Y.; Wang, Z.; Zhu, J.; Wang, Y.; Cai, S.; Yang, X.; Dai, Q.; Dong, Z.; Zhang, Z.; et al. 2025. EAGER-LLM: Enhancing Large Language Models as Recommenders through Exogenous Behavior-Semantic Integration. In *Proceedings of the ACM on Web Conference 2025*, 2754–2762.

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

Kazi, M.; Guo, W.; Gao, H.; and Long, B. 2020. Incorporating user feedback into sequence to sequence model training. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, 2557–2564.

Li, Y. 2024. Graph-of-thought: Utilizing large language models to solve complex and dynamic business problems. *arXiv preprint arXiv:2401.06801*.

Lin, X.; Wang, W.; Li, Y.; Yang, S.; Feng, F.; Wei, Y.; and Chua, T.-S. 2024. Data-efficient Fine-tuning for LLM-based Recommendation. In *Proceedings of the 47th international ACM SIGIR conference on research and development in information retrieval*, 365–374.

Linde, Y.; Buzo, A.; and Gray, R. 2003. An algorithm for vector quantizer design. *IEEE Transactions on communications*, 28(1): 84–95.

Liu, D.-R.; Lai, C.-H.; and Lee, W.-J. 2009. A hybrid of sequential rules and collaborative filtering for product recommendation. *Information Sciences*, 179(20): 3505–3519.

Liu, E.; Zheng, B.; Ling, C.; Hu, L.; Li, H.; and Zhao, W. X. 2025. Generative recommender with end-to-end learnable item tokenization. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 729–739.

Liu, Q.; Wu, X.; Wang, Y.; Zhang, Z.; Tian, F.; Zheng, Y.; and Zhao, X. 2024. Llm-esr: Large language models enhancement for long-tailed sequential recommendation. *Advances in Neural Information Processing Systems*, 37: 26701–26727.

Loshchilov, I.; and Hutter, F. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

McAuley, J.; Targett, C.; Shi, Q.; and Van Den Hengel, A. 2015. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval*, 43–52.

Ni, J.; Li, J.; and McAuley, J. 2019. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th in-*

ternational joint conference on natural language processing (EMNLP-IJCNLP), 188–197.

Rajput, S.; Mehta, N.; Singh, A.; Hulikal Keshavan, R.; Vu, T.; Heldt, L.; Hong, L.; Tay, Y.; Tran, V.; Samost, J.; et al. 2023. Recommender systems with generative retrieval. *Advances in Neural Information Processing Systems*, 36: 10299–10315.

Ruan, Y.; Band, N.; Maddison, C. J.; and Hashimoto, T. 2025. Reasoning to learn from latent thoughts. *arXiv preprint arXiv:2503.18866*.

Saunshi, N.; Dikkala, N.; Li, Z.; Kumar, S.; and J. Reddi, S. 2025. Reasoning with Latent Thoughts: On the Power of Looped Transformers. In Yue, Y.; Garg, A.; Peng, N.; Sha, F.; and Yu, R., eds., *International Conference on Representation Learning*, volume 2025, 14855–14881.

Sun, F.; Liu, J.; Wu, J.; Pei, C.; Lin, X.; Ou, W.; and Jiang, P. 2019. BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM international conference on information and knowledge management*, 1441–1450.

Tang, J.; Dai, S.; Shi, T.; Xu, J.; Chen, X.; Chen, W.; Jian, W.; and Jiang, Y. 2025. Think before recommend: Unleashing the latent reasoning power for sequential recommendation. *arXiv preprint arXiv:2503.22675*.

Tang, J.; and Wang, K. 2018. Personalized top-n sequential recommendation via convolutional sequence embedding. In *Proceedings of the eleventh ACM international conference on web search and data mining*, 565–573.

Touvron, H.; Lavril, T.; Izacard, G.; Martinet, X.; Lachaux, M.-A.; Lacroix, T.; Rozière, B.; Goyal, N.; Hambro, E.; Azhar, F.; et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Wang, S.; Hu, L.; Wang, Y.; Cao, L.; Sheng, Q. Z.; and Orgun, M. 2019. Sequential recommender systems: challenges, progress and prospects. *arXiv preprint arXiv:2001.04830*.

Wang, X.; Chen, H.; Pan, Z.; Zhou, Y.; Guan, C.; Sun, L.; and Zhu, W. 2025. Automated disentangled sequential recommendation with large language models. *ACM Transactions on Information Systems*, 43(2): 1–29.

Wang, X.; Pan, Z.; Zhou, Y.; Chen, H.; Ge, C.; and Zhu, W. 2023. Curriculum co-disentangled representation learning across multiple environments for social recommendation. In *International Conference on Machine Learning*, 36174–36192. PMLR.

Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Xia, F.; Chi, E.; Le, Q. V.; Zhou, D.; et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35: 24824–24837.

Wu, L.; Zheng, Z.; Qiu, Z.; Wang, H.; Gu, H.; Shen, T.; Qin, C.; Zhu, C.; Zhu, H.; Liu, Q.; et al. 2024. A survey on large language models for recommendation. *World Wide Web*, 27(5): 60.

Wu, Z.; Wang, X.; Chang, H.; Chen, H.; Sun, L.; and Zhu, W. 2025. Aligning Large Multimodal Model with Sequential Recommendation via Content-Behavior Guidance. In *Proceedings of the 2025 International Conference on Multimedia Retrieval*, 1507–1516.

Wu, Z.; Wang, X.; Chen, H.; Li, K.; Han, Y.; Sun, L.; and Zhu, W. 2023. Diff4rec: Sequential recommendation with curriculum-scheduled diffusion augmentation. In *Proceedings of the 31st ACM international conference on multimedia*, 9329–9335.

Xie, Y.; Zhou, P.; and Kim, S. 2022. Decoupled side information fusion for sequential recommendation. In *Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval*, 1611–1621.

Yao, S.; Yu, D.; Zhao, J.; Shafran, I.; Griffiths, T.; Cao, Y.; and Narasimhan, K. 2023. Tree of thoughts: Deliberate problem solving with large language models. *Advances in neural information processing systems*, 36: 11809–11822.

Zeghidour, N.; Luebs, A.; Omran, A.; Skoglund, J.; and Tagliasacchi, M. 2021. Soundstream: An end-to-end neural audio codec. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30: 495–507.

Zelikman, E.; Harik, G.; Shao, Y.; Jayasiri, V.; Haber, N.; and Goodman, N. D. 2025. Quiet-STaR: Language Models Can Teach Themselves to Think Before Speaking. In *First Conference on Language Modeling*.

Zelikman, E.; Wu, Y.; Mu, J.; and Goodman, N. 2022. Star: Bootstrapping reasoning with reasoning. *Advances in Neural Information Processing Systems*, 35: 15476–15488.

Zhang, M.; Wu, S.; Yu, X.; Liu, Q.; and Wang, L. 2022. Dynamic graph neural networks for sequential recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 35(5): 4741–4753.

Zhang, Y.; Feng, F.; Zhang, J.; Bao, K.; Wang, Q.; and He, X. 2025. Collm: Integrating collaborative embeddings into large language models for recommendation. *IEEE Transactions on Knowledge and Data Engineering*.

Zhang, Y.; Wang, X.; Chen, H.; and Zhu, W. 2023. Adaptive disentangled transformer for sequential recommendation. In *Proceedings of the 29th ACM SIGKDD conference on knowledge discovery and data mining*, 3434–3445.

Zhang, Z.; Wang, X.; Chen, H.; Li, H.; and Zhu, W. 2024. Disentangled Dynamic Graph Attention Network for Out-of-Distribution Sequential Recommendation. *ACM Transactions on Information Systems*, 43(1): 1–42.

Zheng, B.; Hou, Y.; Lu, H.; Chen, Y.; Zhao, W. X.; Chen, M.; and Wen, J.-R. 2024. Adapting large language models by integrating collaborative semantics for recommendation. In *2024 IEEE 40th International Conference on Data Engineering (ICDE)*, 1435–1448. IEEE.

Zhou, K.; Wang, H.; Zhao, W. X.; Zhu, Y.; Wang, S.; Zhang, F.; Wang, Z.; and Wen, J.-R. 2020. S3-rec: Self-supervised learning for sequential recommendation with mutual information maximization. In *Proceedings of the 29th ACM international conference on information & knowledge management*, 1893–1902.