Collaborative Large Language Model for Recommender Systems

Yaochen Zhu*
University of Virginia
uqp4qh@virginia.edu

Liang Wu
LinkedIn Inc.
liawu@linkedin.com

Qi Guo
LinkedIn Inc.
qguo@linkedin.com

Liangjie Hong
LinkedIn Inc.
liahong@linkedin.com

Jundong Li
University of Virginia
jundong@virginia.edu

ABSTRACT

Recently, there has been growing interest in developing the next-generation recommender systems (RSs) based on pretrained large language models (LLMs). However, the semantic gap between natural language and recommendation tasks is still not well addressed, leading to multiple issues such as spuriously correlated user/item descriptors, ineffective language modeling on user/item data, inefficient recommendations via auto-regression, etc. In this paper, we propose **CLLM4Rec**, the first generative RS that tightly integrates the LLM paradigm and ID paradigm of RSs, aiming to address the above challenges simultaneously. We first extend the vocabulary of pretrained LLMs with user/item ID tokens to faithfully model user/item collaborative and content semantics. Accordingly, a novel *soft+hard prompting* strategy is proposed to effectively learn user/item collaborative/content token embeddings via language modeling on RS-specific corpora, where each document is split into a prompt consisting of heterogeneous *soft* (user/item) tokens and *hard* (vocab) tokens and a main text consisting of homogeneous item tokens or vocab tokens to facilitate stable and effective language modeling. In addition, a novel mutual regularization strategy is introduced to encourage CLLM4Rec to capture recommendation-related information from noisy user/item content. Finally, we propose a novel recommendation-oriented finetuning strategy for CLLM4Rec, where an item prediction head with multinomial likelihood is added to the pretrained CLLM4Rec backbone to predict hold-out items based on soft+hard prompts established from *masked* user-item interaction history, where recommendations of multiple items can be generated efficiently without hallucination[1].

CCS CONCEPTS

• **Information systems** → **Recommender systems**.

KEYWORDS

Recommender systems; large language models (LLM)

[*] Work done when Yaochen Zhu was an applied research intern at LinkedIn.
[1] Codes are released at https://github.com/yaochenzhu/llm4rec.

WWW '24, May 13–17, 2024, Singapore, Singapore
© 2024 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 979-8-4007-0171-9/24/05…$15.00
https://doi.org/10.1145/3589334.3645347

ACM Reference Format:
Yaochen Zhu, Liang Wu, Qi Guo, Liangjie Hong, and Jundong Li. 2024. Collaborative Large Language Model for Recommender Systems. In *Proceedings of the ACM Web Conference 2024 (WWW '24), May 13–17, 2024, Singapore, Singapore.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3589334.3645347

1 INTRODUCTION

With content growing exponentially on the Web, recommender systems (RS) have become essential components for online service platforms [1]. Nevertheless, RS has long been dominated by the ID-based paradigm, where users/items are represented by unique, continuous ID embeddings denoting their semantic similarity [2]. Exemplar ID-based RSs include matrix factorization-based methods (such as PMF [3]) and two-tower models [4], where user/item ID embeddings are either randomly initialized and learned from their historical interactions (i.e., collaborative filtering [5]), or established based on user/item features (i.e., content-based methods [6, 7]).

Recently, large language models (LLM) have become a heated topic for both academia and industry [8]. Large transformer networks pretrained on large-scale corpora, such as GPT [9], T5 [10], and LLaMA [11], have demonstrated **emergent ability** [12], showcasing unprecedented understandings of knowledge and patterns in natural language [8, 13]. Consequently, it is promising to develop the next generation of RS based on pretrained LLMs [14], fully utilizing their encoded knowledge, logical reasoning ability, and generative AI power to understand and reason with user/item semantics and make more accurate recommendations accordingly, especially when users and items are associated with large amounts of textual features, such as biographies, descriptions, content, reviews, and explanations, in modern online service platforms [15, 16].

Several preliminary studies have been conducted to explore the adaptation of LLMs for RSs [17–20]. Typically, these methods can be summarized into two steps: ***(i)*** First, instead of representing users/items with continuous ID embeddings, relevant information necessary for reasoning with user interests and generating recommendations, e.g., interacted items, user/item features, and candidate items, is converted into a discrete *natural language-based prompt*. ***(ii)*** Then, the prompt is used to query the LLM, where information relevant to recommendations is retrieved from the *textual output* of the LLM to generate recommendations (see Fig. 1 for an intuitive example). The above procedure can be performed in a zero-shot manner [21–24], where the recommendation decisions are obtained directly from the pretrained LLM (e.g., we input all relevant information regarding a user and an item into the chatbox of ChatGPT and ask if the user will interact with the item), or if the groundtruths are available, the pretrained LLMs can also be finetuned on both

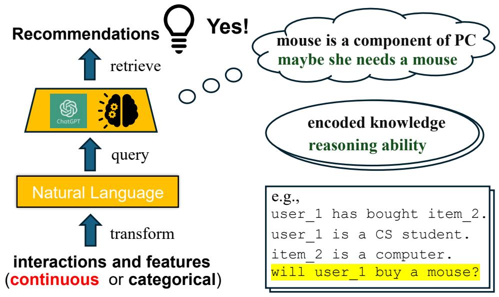


Figure 1: Prospectives of developing the next generation of recommender systems based on pretrained LLMs.

interaction and feature data, such that RS-specific knowledge can be incorporated for more accurate recommendations [18, 25–27].

Although impressive progress has been achieved, fundamental dichotomies between NLP and recommendation still remain to be addressed. One main challenge is the gap between natural language and user/item semantics. Generally, there are two strategies to represent users/items in an LLM-based RS. Pseudo-ID-based methods use an ID-like word (e.g., "user_i" or "item_j") to represent the i-th user or j-th item [18]. However, when tokenized, the ID word may be broken down into atomic tokens, e.g., "user_4332" into ["user", "_", "43", "32"], where spurious correlations can be introduced for irrelevant users/items (e.g., "user_4332" with "user_43" and "user_32"). In contrast, description-based methods use semantically meaningful tokens to index users/items, such as item titles [17, 22] or a small amount of newly-introduced tokens assigned to different users/items based on content similarity [28]. However, description-based methods introduce a strong inductive bias on user-item semantic similarity, which may not faithfully capture the true semantics. Introducing true user/item ID tokens, unfortunately, is generally considered infeasible for LLMs, as directly conducting language modeling (LM) on sequences with heterogeneous tokens can be ineffective and unstable, especially when the vocabulary of most LLMs can be diluted (e.g., ∼ 50k for GPT, and ∼ 30k for T5) by a large number of randomly initialized user/item embeddings.

Even if user/item ID token embeddings can be effectively learned via LM, more challenges exist that hinder effective and efficient recommendations with LLMs. First, since the interaction order usually does not matter for direct recommendations while human language naturally has an order, spurious temporal correlation can be introduced for items placed in different positions when transforming user historical interactions into a textual sentence. In addition, for content modeling, since pretrained LLMs are not recommendation-oriented, they can easily capture noise in user/item textual features irrelevant to the recommendation purpose. Furthermore, since LLMs generate the next token in an autoregressive manner, making multiple recommendations via LLM-based RSs can be inefficient compared with ID-based methods. Finally, for both pseudo-ID-based and description-based indexing methods, item candidates usually need to be explicitly provided in the prompt to avoid hallucination [18]. These issues hinder the practical applications of LLM-based RSs where candidate pools are large and low latency matters.

To address the above challenges, we present **CLLM4Rec**, the first generative RS that tightly combines the ID paradigm of RS with the LLM-based paradigm. We first extend the vocabulary of pretrained LLMs with user/item ID tokens to faithfully model the user/item collaborative/content semantics, where the token embeddings are learned in two stages. The *pretraining stage* consists of mutually regularized collaborative or content LLMs that learn user/item token embeddings via language modeling on RS-specific corpora established from user/item interactions and textual features. Specifically, a novel "soft+hard" prompting strategy is proposed for effective language modeling on documents with heterogeneous tokens, where each document is decomposed into a prompt consisting of *soft* [29] (user/item) and *hard* (vocab) tokens and a main text consisting of homogeneous item tokens (for collaborative modeling) or vocab tokens (for content modeling), respectively. Through this strategy, the prediction heads for the two LLMs can focus exclusively on collaborative and content information, such that the stability and effectiveness of language modeling can be substantially enhanced. In addition, a stochastic item reordering strategy is proposed for the collaborative LLM to ignore the order of item tokens without negative influence on the vocab tokens. Finally, we propose a novel recommendation-oriented *finetuning strategy* for CLLM4Rec, where an item prediction head with multinomial likelihood is added to the pretrained collaborative LLM backbone to predict hold-out items based on soft+hard prompts established from masked user interaction history, where recommendations of multiple items can be efficiently generated without hallucination. The contribution of this paper can be concretely summarized as:

- We present CLLM4Rec, the first generative RS that tightly couples the ID paradigm and LLM paradigm, where user/item ID token embeddings aligned to the LLM vocab space are introduced to well capture the intrinsic user interests and item properties.
- A novel soft+hard prompting strategy is proposed to effectively pretrain CLLM4Rec on heterogeneous tokens describing historical interactions and user/item features in a mutually regularized manner, where collaborative and content information can be effectively learned by the user/item token embeddings.
- A recommendation-oriented finetuning strategy is proposed that predicts hold-out items based on soft+hard prompts established from masked interactions via an item prediction head with multinomial likelihood, where recommendations for multiple items can be generated efficiently without hallucination.

2 RELATED WORK

2.1 Large Language Model (LLM) Basics

Large transformer networks [30] trained on large corpora, i.e., large language models (LLMs), have demonstrated unprecedented understandings of natural language and logical reasoning ability [8]. According to the part of transformer utilized for language modeling, existing LLMs can be categorized into three classes: ***(i)*** encoder-only LLMs, such as BERT [31], ***(ii)*** encoder-decoder-based LLMs, such as T5 [10], and ***(iii)*** decoder-only LLMs, such as GPT, LLaMA [9, 11]. We focus on LLMs with decoders due to their superior generative abilities compared with the encoder-only models [32]. The training of LLMs is mainly based on two stages. In the pretraining stage, LLMs are trained on large corpora via language modeling (i.e., next/masked token prediction), where knowledge can be effectively encoded in the transformer network weights facilitated by the stacked self-attention modules. Then, during the finetuning stage, exemplar prompt-output pairs or human feedback on multiple generated answers are provided to the LLMs such that they can conduct logical reasoning and generate answers according to prompt based on the encoded knowledge from the pretrained stage.

2.2 LLM in Recommender Systems

Recently, LLM-based RSs have shown potential to address the long-standing issues of ID-based RSs, such as shallow understanding of user/item textual features [33], poor generalization [34], etc. Hou et al. [22] demonstrated that existing LLMs can be viewed as zero-shot rankers, which can sort the relevance of movies based on user historical interactions and movie descriptions. Recently, more efforts have been devoted to the finetuning of LLMs to obtain recommendation-oriented models. An exemplar work is P5 [18], which finetunes T5 on corpora established from both interactions and user/item features, where items are presented by pseudo-IDs. Afterward, M6 [17] was proposed to combine text infilling and auto-regression tasks in the pretraining stage, where pseudo IDs are replaced by textual descriptions. Recently, TALLRec [35] was proposed where items are represented by both pseudo-ID and textual descriptions. However, pseudo-ID-based item representations can introduce spurious correlations between irrelevant items. To address this issue, Hua et al. [28] proposed to introduce a small number of new tokens to describe the items, which are determined by their content and collaborative similarity. However, indexing items with shared tokens can still introduce bias. In addition, candidate items need to be explicitly provided in the prompt, and recommendations are generated via inefficient auto-regression. In summary, the dichotomy between NLP and RS is still not well-addressed.

3 METHODOLOGY

3.1 Problem Formulation

In this paper, we focus on recommendations with implicit feedback [36]. Consider a system of I users and J items. We use a binary rating vector $\mathbf{r}_i \in \{0,1\}^J$ to denote whether user i has interacted with the J items. In addition, we use $\mathbf{x}^u_i$, $\mathbf{x}^v_j$ to denote the textual features associated with user i and item j, such as user biography and item content, etc. $\mathbf{x}^{uv}_{ij}$ denotes the textual features associated with both user i and item j, such as user i's review for item j, etc. Hereafter, we take a sequential view of $\mathbf{x}^{\{u,v,uv\}}_{\{i,j,ij\}}$, where $\mathbf{x}^{\{u,v,uv\}}_{\{i,j,ij\},k}$ is a size N one-hot vector denoting the k-th token in the textual sequence. In addition, we have a pretrained large language model (LLM), of which we take a probabilistic view and denote it as $p_{llm}(\mathbf{x}_{k+1}|\mathbf{x}_{1:k})$. p_{llm} transforms $\mathbf{x}_{1:k}$ into a latent sequence $\mathbf{h}^{(L)}_{1:k} \in \mathbb{R}^{k \times K_h}$ via L stacked self-attention modules $llm(\mathbf{x}_{1:k})$ and maps $\mathbf{h}^{(L)}_k$ to the probability space of the next token $\mathbf{x}_{k+1}$. Since the LLM is pretrained on large corpora and finetuned on exemplar prompt-answer pairs, the generation of $\mathbf{x}_{k+1}$ is based on logical reasoning with the context information in $\mathbf{x}_{1:k}$ according to its pretrained knowledge.

Our aim is to design a new generative RS that tightly couples LLMs with the recommendation task by introducing user/item ID tokens (and token embeddings), such that user/item semantics (e.g., users' interests in item) can be accurately modeled for effective and efficient recommendations, and the encoded knowledge and reasoning ability of pretrained LLMs can be fully utilized simultaneously.

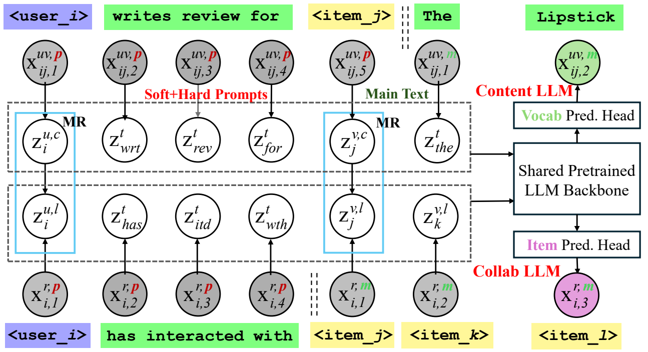


Figure 2: The overview of the proposed CLLM4Rec in the mutually-regularized pretraining stage. Mutual regularization for item_k is omitted for simplicity.

3.2 Extension of User/Item Tokens

3.2.1 ***Vocab Expansion.*** To tightly couple the pretrained LLM with the recommendation task, we first expand the vocabulary of the LLM by adding user/item ID tokens to describe the intrinsic user/item semantics, such that the semantic gap between RS and natural language can be well bridged. We use bracket notations "**<user_*i*>**" and "**<item_*j*>**" to denote the newly-introduced token for the i-th user and the j-th item, which has token ID $N+i$ and $N+I+j$, and will not be broken down into atomic tokens.

3.2.2 ***Token Embeddings.*** For LLMs to understand the newly introduced user/item tokens, they must first be transformed into dense embeddings. Accordingly, we use $\mathbf{z}^t_k \in \mathbb{R}^K$ to represent the pretrained embedding of the k-th vocab token. In addition, for the newly introduced user/item tokens, we introduce *two types of token embeddings* that are aligned with the vocab space to faithfully represent the user/item collaborative and content semantics. Specifically, we first sample user/item collaborative token embeddings from the same K-dimensional latent space as follows:

$$\mathbf{z}^{l,u}_i, \mathbf{z}^{l,v}_j \sim \mathcal{N}\left(\mathbf{0}, \lambda_l^{-1} \cdot \mathbf{I}_K\right), \tag{1}$$

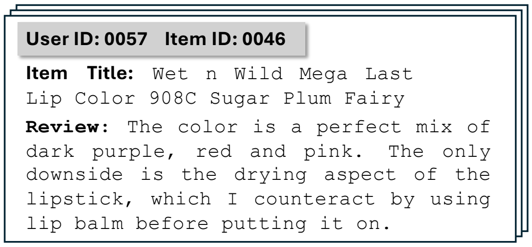
where λ_l is the prior precision for $\mathbf{z}^{l,u}_i, \mathbf{z}^{l,v}_j$. Importantly, to align the content semantics with the collaborative semantics for recommendation-oriented content modeling, we sample user/item content token embeddings from the following conditional prior:

$$\mathbf{z}^{c,u}_i \sim \mathcal{N}\left(\mathbf{z}^{l,u}_i, \lambda_c^{-1} \cdot \mathbf{I}_K\right), \mathbf{z}^{c,v}_j \sim \mathcal{N}\left(\mathbf{z}^{l,v}_j, \lambda_c^{-1} \cdot \mathbf{I}_K\right), \tag{2}$$

where λ_c is the precision for the conditional prior of $\mathbf{z}^{c,u}_i, \mathbf{z}^{c,v}_j$. The horizontally-stacked matrices of vocab/collaborative/content token embeddings are denoted as $\mathbf{Z}^t$, $\mathbf{Z}^{l,\{u,v\}}$, and $\mathbf{Z}^{c,\{u,v\}}$, respectively.

3.2.3 ***CLLM4Rec Base Model.*** With user/item tokens and the corresponding token embeddings introduced in the previous subsections, we are ready to introduce the CLLM4Rec base model with expanded vocabulary. The CLLM4Rec base model is denoted with

$$\mathbf{h}^{(L)}_{\{l,c\},1:k} = \hat{llm}_{\{l,c\}}(\mathbf{x}_{1:k}), \tag{3}$$



User ID: 0057 Item ID: 0046

Item Title: Wet n Wild Mega Last Lip Color 908C Sugar Plum Fairy

Review: The color is a perfect mix of dark purple, red and pink. The only downside is the drying aspect of the lipstick, which I counteract by using lip balm before putting it on.

Figure 3: Exemplar review data from the Amazon Beauty dataset [15], where prior knowledge of natural language can help understand item property and user interests.

which maps the token sequence $\mathbf{x}_{1:k}$ into the hidden space $\mathbb{R}^{k \times K_h}$ through L stacked self-attention modules (the superscript (L) will be omitted if no ambiguity exists); here, $\mathbf{x}_k$ is a size $N+I+J$ one-hot vector denoting the token of either a vocab, a user, or an item. In addition, the subscript in $\hat{llm}_{\{l,c\}}$ denotes which embedding matrix is used to encode the user/item tokens (where l stands for matrix $\mathbf{Z}^{l,\{u,v\}}$ and c stands for matrix $\mathbf{Z}^{c,\{u,v\}}$). For the CLLM4Rec base model $\hat{llm}_{\{l,c\}}$, only the user/item token embeddings are trainable, whereas the vocab embeddings $\mathbf{Z}^t$ as well as the other parts of the backbone LLM are fixed to preserve the pretrained knowledge.

3.3 Mutually-Regularized Pretraining

With CLLM4Rec base model introduced in the previous section, we discuss the mutually-regularized pretraining strategy for CLLM4Rec. The aim is to learn user/item collaborative/content token embeddings based on language modeling on the corpora established from user-item interactions and user/item textual features, where the encoded knowledge and logical reasoning ability of the LLM can be fully utilized. The overall process can be referred to in Fig. 2.

3.3.1 ***Recommendation-Specific Corpora.*** Generally, we can transform the interactions $\mathbf{r}_i$ and user/item content features $\mathbf{x}^u_i$, $\mathbf{x}^v_j$, $\mathbf{x}^{uv}_{ij}$ into documents of user/item/vocab token sequences as follows:

Raw Corpora Transformed from Recommendation Data

(a) *Historical Interactions* $\mathbf{r}_i$:

<user_i> has interacted with <item_j> <item_k> ...

(b) *User/Item Textual Features* $\mathbf{x}^u_i$, $\mathbf{x}^v_j$, $\mathbf{x}^{uv}_{ij}$:

The biography of <user_i> is: Main biography.
The content of <item_j> is: Main contents.
<user_i> writes the review for <item_j>: Main reviews.

where an example based on the Amazon Beauty dataset [15] can be referred to in Fig. 3. However, directly conducting language modeling on the raw corpora is clearly infeasible, as each document is composed of heterogeneous vocab, user, and item tokens, where the number of meaningful vocab tokens (e.g., ~ 50k for GPT, and ~ 30k for T5) can be diluted by the large number of newly introduced user/item tokens with randomly initialized embeddings.

3.3.2 ***Soft+Hard Prompting.*** To address the above challenge, we propose a novel soft+hard prompting strategy to facilitate language modeling on RS-specific corpora with heterogeneous user/item/vocab tokens. The strategy is based on a key observation that documents transformed from both user-item interactions $\mathbf{r}_i$ and user/item textual features $\mathbf{x}^u_i$, $\mathbf{x}^v_j$, $\mathbf{x}^{uv}_{ij}$ can be broken down into two parts: A *heterogeneous* part composed of soft (user/item) and hard (vocab) tokens providing context information regarding the gist of the document, and a main text part with *homogeneous* item/vocab tokens fulfilling the pretexts in detail. Therefore, we can view the first part as a soft+hard prompt and conduct language modeling only on the second part. This encourages the model to focus exclusively on collaborative and content information, such that the effectiveness and stability of language modeling can be substantially enhanced.

For collaborative modeling, document $\mathbf{x}^r_i$ transformed from the historical interactions of user i can be broken down into the soft+hard prompt $\mathbf{x}^{r,p}_i$ and homogeneous item token sequence $\mathbf{x}^{r,m}_i$ as follows:

(a) *Historical Interactions* $\mathbf{r}_i$:

$\underbrace{\texttt{<user\_i> has interacted with}}_{\text{soft+hard prompt } \mathbf{x}^{r,p}_i}\ \underbrace{\texttt{<item\_j> <item\_k> ...}}_{\text{item token seq. } \mathbf{x}^{r,m}_i}$.

Accordingly, we introduce the **collaborative LLM** by adding an item prediction head $f_l : \mathbb{R}^{K_h} \rightarrow \mathbb{P}(J)$ to the CLLM4Rec base model $\hat{llm}_l$, which maps the final-layer last-step hidden representation $\mathbf{h}_{l,-1}$ calculated via $\hat{llm}_l$ to the item probability space $\mathbb{P}(J)$ to predict the next item token. The weights of f_l are tied with the item collaborative token embeddings $\mathbf{Z}^{l,v}$ as $f_l(\mathbf{h}_{l,-1}) = \text{softmax}(\mathbf{Z}^{l,v} \cdot \mathbf{h}_{l,-1})$. The generative process of the collaborative LLM can be denoted as:

$$\mathbf{x}^{r,m}_{i,k+1} \sim p^{f_l}_{\hat{llm}_l}\left(\mathbf{x}^{r,m}_{i,k+1} | \mathbf{x}^{r,m}_{i,1:k}, \mathbf{x}^{r,p}_i\right), \tag{4}$$

where the prompt $\mathbf{x}^{r,p}_i$ serves as a context to generate the next item token based on the previous item tokens. Since the generation of $\mathbf{x}^{r,m}_{i,k+1}$ requires attending to previous tokens, when maximizing the likelihood, the collaborative LLM pushes the token embeddings of user i, i.e., $\mathbf{z}^{l,u}_i$, and the token embeddings of the interacted items, i.e., $\mathbf{z}^{l,v}_j$, $\mathbf{z}^{l,v}_k$, $\cdots$, to be close to each other, where user/item collaborative semantics in recommendation can be accurately captured.

Similarly, for the document transformed from the user/item content $\mathbf{x}^{uv}_{ij}$, it can also naturally be split into a soft+hard prompt $\mathbf{x}^{uv,p}_{ij}$ and the main text $\mathbf{x}^{uv,m}_{ij}$ of homogeneous vocab token sequence as:

(b) *User/Item Textual Features* $\mathbf{x}^{uv}_{ij}$:

$\underbrace{\texttt{<user\_i> writes the review for <item\_j>}}_{\text{soft+hard prompt } \mathbf{x}^{uv,p}_{ij}}:\ \underbrace{\texttt{Main reviews.}}_{\text{vocab seq. } \mathbf{x}^{uv,m}_{ij}}$

Accordingly, we introduce the **content LLM** by adding a vocab prediction head $f_c : \mathbb{R}^{K_h} \rightarrow \mathbb{P}(N)$ to the CLLM4Rec base model $\hat{llm}_c$, which maps the final-layer last-step hidden representation $\mathbf{h}_{c,-1}$ calculated via $\hat{llm}_c$ (which shares the same pretrained LLM with $\hat{llm}_l$ but uses $\mathbf{Z}^{c,\{u,v\}}$ to decode the user/item tokens) to the vocab probability space. Similarly, the weights of f_c are tied with the vocab embeddings $\mathbf{Z}^t$ as $f_c(\mathbf{h}_{c,-1}) = \text{softmax}(\mathbf{Z}^t \cdot \mathbf{h}_{c,-1})$. The generative process of the content LLM can be denoted as follows:

$$\mathbf{x}^{uv,m}_{ij,k+1} \sim p^{f_c}_{\hat{llm}_c}\left(\mathbf{x}^{uv,m}_{ij,k+1} | \mathbf{x}^{uv,m}_{ij,1:k}, \mathbf{x}^{uv,p}_{ij}\right), \tag{5}$$

which generates the next vocab token $\mathbf{x}_{ij,k+1}^{uv,m}$ based on the previously generated vocab tokens $\mathbf{x}_{ij,1:k}^{uv,m}$ with prompt $\mathbf{x}_{ij}^{uv,p}$ as the context. When maximizing the likelihood, the content information in $\mathbf{x}_{ij}^{uv,m}$ can be encoded in the content token embeddings of user i and item j, i.e., $\mathbf{z}_i^{c,u}$, $\mathbf{z}_j^{c,v}$, where the pretrained knowledge of the LLM can be fully utilized. For example, for the review shown in Fig. 3, the pretrained LLM will know that **<item_46>** is a lipstick with dark purple, red, and pink colors and can have side effects of drying lips, and reasons that **<user_57>** likes the colors but hates the side effects, which can be alleviated by applying lip balm.

Discussion. Generally, since the "hard" (i.e., the vocab) part of the prompts $\mathbf{x}_i^{r,p}$ and $\mathbf{x}_{ij}^{uv,p}$ is what the pretrained LLM could understand, it is designed to trigger the reasoning ability of the pretrained LLM based on its encoded knowledge. For example, the relational phrase **"has interacted with"** in the prompt $\mathbf{x}_i^{r,p}$ guides the collaborative LLM to understand that the newly-introduced token **<user_*i*>** is a *user subject* and the tokens in the prompt $\mathbf{x}_i^{r,m}$ are the *objects* of interacted item sequences. Meanwhile, the contexts **"write the review for"** in $\mathbf{x}_{ij}^{uv,p}$ direct the content LLM to better understand the nature of main texts in $\mathbf{x}_{ij}^{uv,m}$, i.e., **<user_*i*>**'s judgment on **<item_*j*>** based on the personal using experience. The specific formulation of the prompt can be flexible, as Geng et al. [18] have demonstrated that variations in the expression of the prompt make less difference as long as the meaning is the same and the prompt is consistent across the training and testing phases.

3.3.3 *Mutually-Regularization*

Since pretrained LLMs are not recommendation-oriented, naively optimizing Eq. (5) unavoidably captures noisy information from content features irrelevant to recommendations. In addition, since user/item interactions are sparse, the collaborative LLM can easily overfit on the observed interactions when optimizing Eq. (4). To address these issues, we propose a mutually regularized pretraining strategy for CLLM4Rec, where collaborative LLM can guide content LLM to capture recommendation-related information from user/item content, and content LLM can in turn introduce side information to support collaborative filtering.

The mutual regularization naturally comes with the aligned generative process of CLLM4Rec defined in Eqs. (1), (2). Specifically, for user i, if we denote the stacked item token embeddings as $\mathbf{Z}_i^{c,v}$, $\mathbf{Z}_i^{l,v}$, which contains item j and other items interacted by the user i, the generation process of CLLM4Rec associated with $\mathbf{x}_i^{r}$ and $\mathbf{x}_{ij}^{uv}$ can be defined as the joint distribution as follows:

$$p\left(\mathbf{x}_i^{r,m}, \mathbf{x}_{ij}^{uv,m}, \mathbf{z}_i^{l,u}, \mathbf{Z}_i^{l,v}, \mathbf{z}_i^{c,u}, \mathbf{Z}_i^{c,v} \middle| \mathbf{x}_i^{r,p}, \mathbf{x}_{ij}^{uv,p}\right) =$$
$$\underbrace{\Pi_k p_{\hat{llm}_l}^{f_l}\left(\mathbf{x}_{i,k}^{r,m} \middle| \mathbf{x}_{i,1:k-1}^{r,m}, \mathbf{x}_i^{r,p}\right)}_{\text{LM for collab. LLM}} \cdot \underbrace{\Pi_k p_{\hat{llm}_c}^{f_c}\left(\mathbf{x}_{ij,k}^{uv,m} \middle| \mathbf{x}_{ij,1:k-1}^{uv,m}, \mathbf{x}_{ij}^{uv,p}\right)}_{\text{LM for content LLM}} \cdot$$
$$\underbrace{p\left(\mathbf{z}_i^{c,u} \middle| \mathbf{z}_i^{l,u}\right) \cdot \Pi_k p\left(\mathbf{z}_{ik}^{c,v} \middle| \mathbf{z}_{ik}^{l,v}\right)}_{\text{mutual regularization}} \cdot \underbrace{p\left(\mathbf{z}_i^{l,u}\right) \cdot \Pi_k p\left(\mathbf{z}_{ik}^{l,v}\right)}_{\text{prior}}. \tag{6}$$

A scrutiny of Eq. (6) reveals that the joint distribution can be decomposed into three parts: ***(i)*** the language modeling of the collaborative and content LLMs that learn user/item token embeddings as Eqs. (4) and (5); ***(ii)*** the mutual regularization that connects the user/item token embeddings of the two LLMs (i.e., according to Eqs. (1), (2), $p\left(\mathbf{z}_i^{c,u} \middle| \mathbf{z}_i^{l,u}\right)$ and $p\left(\mathbf{z}_{ik}^{c,v} \middle| \mathbf{z}_{ik}^{l,v}\right)$ are conditional Gaussian, which will introduce MSE regularization between $\mathbf{z}_i^{c,u}$, $\mathbf{z}_i^{l,u}$, and $\mathbf{z}_{ik}^{c,v}$, $\mathbf{z}_{ik}^{l,v}$ when log-likelihood is maximized); ***(iii)*** the prior of $\mathbf{z}_i^{l,u}$ and $\mathbf{z}_{ik}^{l,v}$, which will be ignored due to the existence of mutual regularization (i.e., setting the precision λ_l in the prior in Eq. (1) as zero).

We use Maximum a Posteriori (MAP) [37] to estimate the user/item token embeddings $\mathbf{z}_i^{l,u}, \mathbf{Z}_i^{l,v}, \mathbf{z}_i^{c,u}, \mathbf{Z}_i^{c,v}$, where the objective is proportional to the logarithm of the joint distribution defined in Eq. (6). Here, we take alternate steps to optimize the MAP objective. If we denote the trainable parameters associated with the item token prediction head f_l and vocab token prediction head f_c as $\boldsymbol{\theta}$ (which are tied with the corresponding token embeddings), the objective for the collaborative LLM (L-step) and content LLM (C-step) with mutual regularization can be derived as follows:

L-step. In the L-step, we fix user/item content embeddings $\mathbf{z}_i^{c,u}, \mathbf{Z}_i^{c,v}$ as $\hat{\mathbf{z}}_i^{c,u}, \hat{\mathbf{Z}}_i^{c,v}$ in Eq. (6), and use them to constrain the user/item collaborative embeddings along with the language modeling of collaborative LLM, leading to the following composite objective:

$$\mathcal{L}_{\text{l\_step}}^{\text{MAP}}\left(\mathbf{z}_i^{l,u}, \mathbf{Z}_i^{l,v}; \boldsymbol{\theta}\right) = \sum_k \underbrace{-\ln p_{\hat{llm}_l}^{f_l}\left(\mathbf{x}_{i,k}^{r,m} \middle| \mathbf{x}_{i,1:k-1}^{r,m}, \mathbf{x}_i^{r,p}\right)}_{\text{LM loss for collab. LLM}}$$
$$\underbrace{+\frac{\lambda_c}{2}\left\|\mathbf{z}_i^{l,u} - \hat{\mathbf{z}}_i^{c,u}\right\|_2^2 + \sum_k \frac{\lambda_c}{2} \cdot \left\|\mathbf{z}_{ik}^{l,v} - \hat{\mathbf{z}}_{ik}^{c,v}\right\|_2^2}_{\text{MR loss with content LLM}} \underbrace{+\frac{\lambda_l}{2}\left\|\mathbf{z}_i^{l,u}\right\|_2^2 + \frac{\lambda_l}{2}\left\|\mathbf{z}_j^{l,v}\right\|_2^2}_{\text{Prior loss}} + C_l, \tag{7}$$

where C_l is the constant irrelevant for optimization. The ***LM loss*** captures the collaborative similarity between token embeddings of user i and the interacted items, where side information can be introduced via the ***MR loss*** to support collaborative filtering.

C-step. After one-step optimization of the L-step, we fix the user/item collaborative token embeddings $\mathbf{z}_i^{l,u}$, $\mathbf{z}_j^{l,v}$ as $\hat{\mathbf{z}}_i^{l,u}$, $\hat{\mathbf{z}}_j^{l,v}$ in Eq. (6), leading to the following composite objective for the content LLM:

$$\mathcal{L}_{\text{c\_step}}^{\text{MAP}}\left(\mathbf{z}_i^{c,u}, \mathbf{z}_j^{c,v}; \boldsymbol{\theta}\right) = \sum_k \underbrace{-\ln p_{\hat{llm}_c}^{f_c}\left(\mathbf{x}_{ij,k}^{uv,m} \middle| \mathbf{x}_{ij,1:k-1}^{uv,m}, \mathbf{x}_{ij}^{uv,p}\right)}_{\text{LM loss for content LLM}}$$
$$\underbrace{+\frac{\lambda_c}{2}\left\|\mathbf{z}_i^{c,u} - \hat{\mathbf{z}}_i^{l,u}\right\|_2^2 + \frac{\lambda_c}{2} \cdot \left\|\mathbf{z}_j^{c,v} - \hat{\mathbf{z}}_j^{l,v}\right\|_2^2}_{\text{MR loss with collab. LLM}} + C_c, \tag{8}$$

where ***MR loss*** encourages the content LLM to capture recommendation-oriented information from user/item textual features. In Eqs. (7) and (8), λ_c controls the strength of mutual regularization, which will be thoroughly discussed in the empirical study.

3.3.4 *Stochastic Item Reordering*

Another issue that hinders effective collaborative filtering via Eq. (7) is the order of item tokens when transforming the historical interactions $\mathbf{r}_i$ into a token sequence $\mathbf{x}_i^{r,m}$. Item order usually does not matter for direct recommendations as users' long-term interests can be viewed as fixed (even if it matters, the positional embeddings denoting the order of natural language may not capture the semantics of the order of interactions). To address this issue, we propose a stochastic item reordering strategy to randomly permute the item tokens in $\mathbf{x}_i^{r,m}$, with soft+hard prompt $\mathbf{x}_i^{r,p}$ fixed when optimizing the collaborative LLM as Eq. (7). Through this strategy, the order of items can be ignored without negative influence on the vocab tokens in $\mathbf{x}_i^{r,p}$.

3.4 Recommendation-Oriented Finetuning

3.4.1 ***Pretraining v.s. Finetuning.*** The pretraining of CLLM4Rec aims to learn the user/item token embeddings based on the large corpora established from user-item interactions $\mathbf{r}_i$ and user/item textual features $\mathbf{x}_i^u, \mathbf{x}_j^v, \mathbf{x}_{ij}^{uv}$ via language modeling, such that prompts with heterogeneous user/item/vocab tokens can be properly understood by CLLM4Rec. However, for now, the pretrained CLLM4Rec can only complete item/vocab token sequences based on prompts, rather than making recommendations, and therefore the gap between NLP and RS is still not completely eliminated. In addition, naively treating the collaborative LLM as a recommendation model can lead to huge computational costs as the recommended items are sequentially generated via auto-regression. Therefore, we propose a novel recommendation-oriented finetuning strategy for CLLM4Rec, which aims to further finetune the pretrained collaborative LLM and tailor it for more efficient recommendations.

3.4.2 ***Masked Prompting with Multinomial Prediction Head.*** To achieve this purpose, we first design a masked prompting strategy to generate recommendation-oriented prompts and targets for CLLM4Rec finetuning. Specifically, for each user, we randomly mask the interacted items $\mathbf{r}_i$ by $100 \times p_m\%$, where the remaining items are denoted as $\mathbf{r}_i^{masked}$. We then use $\mathbf{r}_i^{masked}$ to generate a recommendation-oriented prompt $\mathbf{x}_i^{rec,p}$ as the input. All hold-out items, which we denote with a multi-hot vector $\mathbf{r}_i^{hold}$, are treated as the target. The prompt $\mathbf{x}_i^{rec,p}$ based on $\mathbf{r}_i^{masked}$ is designed as:

(c) *Masked Prompts & Target for Finetuning*

(prompt) <user_*i*> has interacted with <item_*j*'> ☒

☒ <item_*k*'> the user will interact with:

(target) $\mathbf{r}_i^{hold} \in \{0,1\}^J$ ☒ (masked items)

which triggers the reasoning ability of the pretrained LLM by using relational phrase **"has interacted with"** to describe the historical interactions, and using the phrase **"the user will interact with"** to guide the prediction of the target hold-out items $\mathbf{r}_i^{hold}$.

We name CLLM4Rec in the finetuning stage as **RecLLM**, which inherits the CLLM4Rec base model $\hat{llm}_l$ from the collaborative LLM in the pretraining stage and introduces a new item prediction head with multinomial likelihood, i.e., f_{rec}, whose weights are also tied with the item token embeddings $\mathbf{Z}^{l,v}$. The generation of the hold-out items $\mathbf{r}_i^{hold}$ via the RecLLM can be formulated as follows:

$$\mathbf{r}_i^{hold} \sim multi\left(f_{rec}\left(\mathbf{h}^{rec}_{l,i,-1}\right), N_i^{hold}\right), \text{ where } \mathbf{h}^{rec}_{l,i} = \hat{llm}_l\left(\mathbf{x}_i^{rec,p}\right), \tag{9}$$

where $multi$ denotes the multinomial distribution, and N_i^{hold} is the number of hold-out items for user i. When finetuning the RecLLM according to Eq. (9), $\mathbf{h}^{rec}_{l,i,-1}$, which can be viewed as the latent variable summarizing the historical interaction of user i, is encouraged to be similar to the collaborative embeddings of all the interacted items. In addition, we keep it regularized with the content LLM in a similar manner as Eq. (7)[2], and use the stochastic item reordering strategy to generate the prompt $\mathbf{x}_i^{rec,p}$. Through the proposed recommendation-oriented finetuning strategy, CLLM4Rec can efficiently generate recommendations in a single forward-propagation step while fully utilizing the encoded knowledge of the pretrained LLM backbone and the user/item token embeddings learned via mutually-regularized pretraining, where all J items serve as the candidates. In addition, since the target $\mathbf{r}_i^{hold}$ is constrained to be in the item probability space, hallucinated items can be avoided.

3.5 Predictions with CLLM4Rec

After the pretraining and finetuning of CLLM4Rec, to make recommendations for user i, we can convert the *whole* historical interactions of the user, i.e., $\mathbf{r}_i$, into the recommendation-oriented prompt $\hat{\mathbf{x}}_i^{rec,p}$ as described in Section 3.4.2 (with no masked items) and input it into the RecLLM model. Then, the multinomial probability $\hat{\mathbf{r}}_i$ **over all** J **items** can be obtained through one forward propagation via $\hat{\mathbf{r}}_i = multi\left(f_{rec}\left(\hat{\mathbf{h}}^{rec}_{i,-1}\right)\right), \hat{\mathbf{h}}^{rec}_i = \hat{llm}_l\left(\hat{\mathbf{x}}_i^{rec,p}\right)$, where uninteracted items with top-M scores in $\hat{\mathbf{r}}_i$ can be selected as recommendations.

4 EMPIRICAL STUDY

In this section, we present and analyze the experiments on four public datasets and the LinkedIn job recommendation dataset, aiming to answer the following three research questions:

- **RQ1.** How does CLLM4Rec, the first RS that tightly couples the ID-based paradigm with the LLM-based paradigm, perform compared to state-of-the-art ID-based and LLM-based RSs?
- **RQ2.** How does the **pretraining stage** of CLLM4Rec (including the mutual regularization trick and the stochastic item reorder strategy) influence the performance of CLLM4Rec?
- **RQ3.** How does the **finetuning stage** of CLLM4Rec with masked prompting and multinomial item prediction head influence the efficiency and effectiveness of recommendations?

Due to space limitation, we only discuss CLLM4Rec with GPT-2 [9] backbone in this section, which has 768-dimensional token embeddings and token size 50,257. Experiments with more LLM backbones are thoroughly discussed in Appendix B.

4.1 Experimental Setup

4.1.1 ***Datasets.*** The four public datasets we include for experiments are Amazon (AM)-Beauty dataset, AM-Toys dataset, AM-Sports dataset [15] and Yelp dataset [38]. In preprocessing, we binarize the interactions by keeping only ratings > 3 and treat them as implicit feedback [39]. In addition, we filter the datasets such that they keep the 5-core property after binarization. For each user, we randomly select 80% of interactions for training, 10% for validation, and 10% for testing, where at least one item is selected in the validation and the test set. The reviews users provide to the items are collected as the textual feature $\mathbf{x}_{ij}^{uv}$. The **real-world experiments** are based on a job recommendation dataset collected at LinkedIn, where users' clicks on the job Ads are logged as the implicit feedback, and users' self-provided biography $\mathbf{x}_i^u$ and the job descriptions

[2] The objective of the RecLLM is formulated in Eq. (10) in Appendix A.2.

$\mathbf{x}_j^v$ are collected as the textual features, respectively. The statistics of the dataset are summarized in Table 3 in the Appendix.

4.2 Comparison with Baselines

4.2.1 ***Baselines.*** To demonstrate the multifaceted superiority of the proposed CLLM4Rec, we include the following ID-based and (L)LM-based RSs as baselines for comparisons:

(i) **ID-based Baselines.**

- **Multi-Vae** [39] is an ID-based collaborative filtering baseline that recommends new items by reconstructing the ratings $\mathbf{r}_i$ via a variational auto-encoder (VAE) with multinomial likelihood.
- **Md-Cvae** [40] is a hybrid RS that extends the Multi-VAE by introducing a dual feature VAE on textual features to regularize the reconstruction of $\mathbf{r}_i$ in Multi-VAE.

(ii) **LM-based Baselines**[3].

- **Bert4Rec** [41] uses masked language modeling (MLM) proposed in BERT [31] to learn user/item embeddings for recommendation via bidirectional self-attention.
- **S^{3}Rec** [38] extends BERT4Rec by augmenting the MLM with auxiliary tasks such as item attribute prediction, where content features can be fused for self-supervised learning.

(iii) **LLM-based Baselines.**

- **Llm-Scratch** has the same structure as CLLM4Rec, but it trains the whole model from scratch instead of loading and fixing the weights of the pretrained LLM backbone.
- **Llm-CF** eliminates the content LLM from CLLM4Rec and the mutually-regularized pretraining step and uses only the collaborative LLM and RecLLM for recommendations.
- **Llm-FtALL** has the same structure as CLLM4Rec, but it finetunes the whole network, including the vocab embeddings as well as other parts of the pretrained LLM, instead of training only the newly-introduced user/item token embeddings.
- **Llm-FixOrd** has the same structure as CLLM4Rec, but it removes the stochastic item reordering strategy for both the collaborative LLM in pretraining and the RecLLM in finetuning.
- **Llm-PreRec** discards finetuning and ranks the categorical probability from the next item token prediction head of the collaborative LLM in the pretraining stage to make recommendations.

4.2.2 ***Qualitative Analysis.*** For other existing LLM-based RSs (i.e., both pseudo-ID-based and description-based methods introduced in Section 2.2), they represent users/items with multiple tokens and formulate direct recommendation as a next token generation problem. Since the generated tokens could be irrelevant to the recommendation purpose, candidate items usually need to be explicitly provided in the prompt to avoid hallucination (e.g., P5 [18] provides 100 candidate items where one is positive, and TALLRec [35] outputs yes/no decision based on user/item descriptions in the prompts, etc.). In contrast, CLLM4Rec can simultaneously generate multiple recommendations from the entire item candidate pool. Therefore, these methods cannot directly work in our setting, and the comparisons are mainly based on qualitative analysis.

Table 1: Comparison between CLLM4Rec and various baselines with GPT-backbone on three Amazon Review datasets.

AM-Beauty	Recall@20	Recall@40	NDCG@100
Multi-VAE	0.1295	0.1720	0.0835
MD-CVAE	0.1472	0.2058	0.0976
BERT4Rec	0.1126	0.1677	0.0781
S^{3}Rec	0.1354	0.1789	0.0867
LLM-Scratch	0.0840	0.1265	0.0583
LLM-CF	0.1319	0.1841	0.0855
LLM-FtAll	0.1335	0.1988	0.0836
LLM-FixOrd	0.1524	0.2219	0.1072
LLM-PreRec	0.1547	0.2196	0.1051
CLLM4Rec	**0.1656**	**0.2323**	**0.1118**

AM-Toys	Recall@20	Recall@40	NDCG@100
Multi-VAE	0.1076	0.1558	0.0781
MD-CVAE	0.1291	0.1804	0.0844
BERT4Rec	0.0853	0.1375	0.0532
S^{3}Rec	0.1064	0.1524	0.0665
LLM-Scratch	0.0485	0.0771	0.0362
LLM-CF	0.1027	0.1434	0.0680
LLM-FtAll	0.1162	0.1542	0.0696
LLM-FixOrd	0.1342	0.1887	0.0889
LLM-PreRec	0.1308	0.1859	0.0874
CLLM4Rec	**0.1436**	**0.1933**	**0.0918**

AM-Sports	Recall@20	Recall@40	NDCG@100
Multi-VAE	0.0659	0.0975	0.0446
MD-CVAE	0.0714	0.1180	0.0514
BERT4Rec	0.0521	0.0701	0.0305
S^{3}Rec	0.0616	0.0813	0.0438
LLM-Scratch	0.0362	0.0538	0.0362
LLM-CF	0.0642	0.0966	0.0419
LLM-FtAll	0.0794	0.1002	0.0424
LLM-FixOrd	0.0901	0.1295	0.0592
LLM-PreRec	0.0839	0.1248	0.0561
CLLM4Rec	**0.0926**	**0.1351**	**0.0634**

4.2.3 ***Results on the Public Datasets.*** We first analyze the experimental results on four public datasets to provide preliminary answers for **RQs. 1, 2, 3**. From Tables 1 and 2, we can find that the ID-base method, Multi-VAE, remains a strong baseline for collaborative filtering (CF). LLM-CF, the CF backbone of CLLM4Rec, cannot beat Multi-VAE on both AM-Sports and Toys datasets, even if the "hard" part of the prompt triggers the reasoning ability of the pretrained LLM. However, when utilizing the large textual data, CLLM4Rec outperforms its ID-based counterpart, MD-CVAE (which tightly couples an item content VAE with Multi-VAE) by a large margin. This is because MD-CVAE uses shallow bag-of-word representations of textual features, for which pretrained LLMs in CLLM4Rec can provide deeper understanding via their pretrained knowledge. The importance of pretrained knowledge can also be shown by the LLM-Scratch model, which performs the worst among all included baselines. An interesting finding is that, LLM-FtAll, which finetunes the whole model including the pretrained LLM backbone, performs worse than CLLM4Rec, which optimizes only the newly

[3] Note that both BERT4Rec and S^{3}Rec are original designed for sequential recommendation. In this paper, we use similar recommendation-oriented finetuning as CLLM4Rec to adapt them to direct recommendation, where item sequences generated from masked interactions are used to predict all hold-out items with multinomial likelihood.

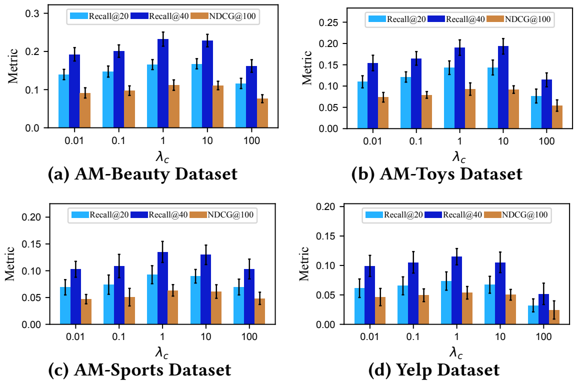
Table 2: Comparison between CLLM4Rec and various baselines on the Yelp dataset and the LinkedIn dataset.

Yelp	Recall@20	Recall@40	NDCG@100
Multi-VAE	0.0526	0.0842	0.0424
MD-CVAE	0.0664	0.1058	0.0497
BERT4Rec	0.0418	0.0724	0.0361
S^{3}Rec	0.0563	0.0893	0.0485
LLM-Scratch	0.0199	0.0325	0.0159
LLM-CF	0.0541	0.0860	0.0412
LLM-FtAll	0.0653	0.0989	0.0520
LLM-FixOrd	0.0694	0.1053	0.0524
LLM-PreRec	0.0639	0.1021	0.0498
CLLM4Rec	**0.0735**	**0.1149**	**0.0536**

LinkedIn	Recall@10	Recall@20	NDCG@10
Two-Tower	0.1186	0.2041	0.0979
M6-Retrieval	0.1279	0.2118	0.1020
CLLM4Rec-Emb	0.1302	0.2165	0.1034
CLLM4Rec	**0.1427**	**0.2398**	**0.1199**

introduced user/item token embeddings. The reason could be that, since the weights of the pretrained LLM are fully optimized, the recommendation-specific corpora are still not enough to adapt the pretrained LLM with good generalization ability for RS. Therefore, the cons of degenerating the pretrained knowledge outweigh the pros of introducing extra RS-specific knowledge. We can also find that LLM-PreRec, which uses the collaborative LLM in the pretraining stage to generate recommendations, is already a strong baseline. This demonstrates the effectiveness of the soft+hard prompting strategy to facilitate efficient and stable language modeling on recommendation-oriented corpora with heterogeneous tokens. Still, CLLM4Rec performs better than LLM-PreRec, which demonstrates the effectiveness of recommendation-oriented finetuning in adapting collaborative LLM for efficient recommendations.

4.2.4 ***Results on the LinkedIn Dataset.*** In the real-world experiment, we compare CLLM4Rec with the two-tower (TT) model utilized in LinkedIn for job recommendations. The TT model is implemented as a two-branch multi-layer perceptron (MLP), where the input user/item embeddings include embeddings extracted from a graph neural network (GNN) learned on the user-job bipartite graph, as well as features extracted from an internal BERT model. In addition, since the textual features are available for almost every user and item, we compare CLLM4Rec with the state-of-the-art LLM-based RS, M6-Retrieval [17], which takes the dimensional-reduced embeddings of user/item descriptions from M6 transformer for contrastive recommendations. The results are summarized in Table 2. For Table 2, we can find that CLLM4Rec outperforms the shallow TT model by a large margin. However, although the inference latency for CLLM4Rec is significantly improved compared with existing methods due to the introduction of recommendation-oriented finetuning, directly deploying CLLM4Rec online is still infeasible, as the inference budgets are higher compared to the TT model. Therefore, we design the CLLM4Rec-Emb baseline, which includes the user/item token embeddings $\mathbf{Z}^{l,u}$ and $\mathbf{Z}^{l,v}$ learned from CLLM4Rec (projected into 128 dimensions) as extra inputs for the TT model, which demonstrates a performance improvement than the original TT model and the M6-Retrieval model in our offline experiment. This demonstrates the potential application of CLLM4Rec in industrial applications where low latency matters.



(a) AM-Beauty Dataset

(b) AM-Toys Dataset

(c) AM-Sports Dataset

(d) Yelp Dataset

Figure 4: Sensitivity analysis w.r.t. λ_c, which controls the strength of mutual-regularization for CLLM4Rec.

4.3 Parameter Sensitivity Analysis

To further answer **RQs. 2** and **3**, we vary λ_c in Eqs. (7), (8), and (10) that controls the strength of mutual regularization and investigates how it influences the performance of CLLM4Rec. From Fig. 4, we can find that, when λ_c is small, the mutual regularization is weak, and the content LLM cannot provide enough user/item content side information to support the collaborative LLM and RecLLM. Therefore, the recommendation performance degenerates to a similar level as the LLM-CF. On the other hand, when λ_c is too large, the MR loss in Eqs. (7), (8) and (10) dominates, which hinders CLLM4Rec from learning useful user/item token embeddings via language modeling. Generally, for all four datasets, the performance of CLLM4Rec peaks at around $\lambda_c = 1$, which serves as a good start when applying the GPT-based CLLM4Rec to new datasets.

5 CONCLUSION

In this paper, we proposed CLLM4Rec, the first method that tightly couples the ID paradigm and the LLM paradigm of RS, which faithfully captures user/item semantics while fully utilizing encoded knowledge and logical reasoning ability of pretrained LLMs simultaneously. Specifically, with mutually regularized pretraining based on soft+hard prompting strategy, CLLM4Rec can effectively capture the user/item collaborative and content information via language modeling. Furthermore, with recommendation-oriented finetuning, the pretrained knowledge of CLLM4Rec can be fully utilized to efficiently generate recommendations. Extensive experiments show the multifaceted superiority of CLLM4Rec over the state-of-the-art.

ACKNOWLEDGMENT

Yaochen Zhu and Jundong Li are supported in part by the National Science Foundation under grants (IIS-2006844, IIS-2144209, IIS-2223769, CNS2154962, and BCS-2228534), the Commonwealth Cyber Initiative Awards under grants (VV-1Q23-007, HV2Q23-003, and VV-1Q24-011), the JP Morgan Chase Faculty Research Award, and the Cisco Faculty Research Award.

REFERENCES

[1] Dietmar Jannach, Markus Zanker, Alexander Felfernig, and Gerhard Friedrich. *Recommender Systems: An Introduction.* Cambridge University Press, 2010.

[2] Zheng Yuan, Fajie Yuan, Yu Song, Youhua Li, Junchen Fu, Fei Yang, Yunzhu Pan, and Yongxin Ni. Where to go next for recommender systems? ID vs. modality-based recommender models revisited. *arXiv preprint arXiv:2303.13835*, 2023.

[3] Andriy Mnih and Russ R Salakhutdinov. Probabilistic matrix factorization. In *NeurIPS*, volume 20, 2007.

[4] Ledell Wu, Adam Fisch, Sumit Chopra, Keith Adams, Antoine Bordes, and Jason Weston. Starspace: Embed all the things! In *AAAI*, volume 32, 2018.

[5] Yehuda Koren, Steffen Rendle, and Robert Bell. Advances in collaborative filtering. *Recommender Systems Handbook*, pages 91–142, 2021.

[6] Pasquale Lops, Marco De Gemmis, and Giovanni Semeraro. Content-based recommender systems: State of the art and trends. *Recommender Systems Handbook*, pages 73–105, 2011.

[7] Yaochen Zhu, Jing Ma, Liang Wu, Qi Guo, Liangjie Hong, and Jundong Li. Path-specific counterfactual fairness for recommender systems. In *SIGKDD*, page 3638–3649, 2023.

[8] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 2023.

[9] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.

[10] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *JMLR*, 21(1):5485–5551, 2020.

[11] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. LlaMA: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[12] Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*, 2022.

[13] Song Wang, Yaochen Zhu, Haochen Liu, Zaiyi Zheng, Chen Chen, and Jundong Li. Knowledge editing for large language models: A survey, 2023.

[14] Wenqi Fan, Zihuai Zhao, Jiatong Li, Yunqing Liu, Xiaowei Mei, Yiqi Wang, Jiliang Tang, and Qing Li. Recommender systems in the era of large language models (LLMs). *arXiv preprint arXiv:2307.02046*, 2023.

[15] Julian McAuley and Alex Yang. Addressing complex and subjective product-related queries with customer reviews. In *WWW*, pages 625–635, 2016.

[16] Yaochen Zhu and Zhenzhong Chen. Variational bandwidth auto-encoder for hybrid recommender systems. *IEEE TKDE*, 35(5):5371–5385, 2022.

[17] Zeyu Cui, Jianxin Ma, Chang Zhou, Jingren Zhou, and Hongxia Yang. M6-rec: Generative pretrained language models are open-ended recommender systems. *arXiv preprint arXiv:2205.08084*, 2022.

[18] Shijie Geng, Shuchang Liu, Zuohui Fu, Yingqiang Ge, and Yongfeng Zhang. Recommendation as language processing (RLP): A unified pretrain, personalized prompt & predict paradigm (P5). In *ACM RecSys*, pages 299–315, 2022.

[19] Jiaxing Qu, Yuxuan Richard Xie, and Elif Ertekin. A language-based recommendation system for material discovery. In *ICML*, 2023.

[20] Lei Li, Yongfeng Zhang, and Li Chen. Personalized prompt learning for explainable recommendation. *ACM TIST*, 41(4):1–26, 2023.

[21] Yunfan Gao, Tao Sheng, Youlin Xiang, Yun Xiong, Haofen Wang, and Jiawei Zhang. Chat-Rec: Towards interactive and explainable llms-augmented recommender system. *arXiv preprint arXiv:2303.14524*, 2023.

[22] Yupeng Hou, Junjie Zhang, Zihan Lin, Hongyu Lu, Ruobing Xie, Julian McAuley, and Wayne Xin Zhao. Large language models are zero-shot rankers for recommender systems. *arXiv preprint arXiv:2305.08845*, 2023.

[23] Junjie Zhang, Ruobing Xie, Yupeng Hou, Wayne Xin Zhao, Leyu Lin, and Ji-Rong Wen. Recommendation as instruction following: A large language model empowered recommendation approach. *arXiv preprint arXiv:2305.07001*, 2023.

[24] Zhankui He, Zhouhang Xie, Rahul Jha, Harald Steck, Dawen Liang, Yesu Feng, Bodhisattwa Prasad Majumder, Nathan Kallus, and Julian McAuley. Large language models as zero-shot conversational recommenders. *arXiv preprint arXiv:2308.10053*, 2023.

[25] Fan Yang, Zheng Chen, Ziyan Jiang, Eunah Cho, Xiaojiang Huang, and Yanbin Lu. Palr: Personalization aware llms for recommendation. *arXiv e-prints*, pages arXiv–2305, 2023.

[26] Jianchao Ji, Zelong Li, Shuyuan Xu, Wenyue Hua, Yingqiang Ge, Juntao Tan, and Yongfeng Zhang. GenRec: Large language model for generative recommendation. *arXiv e-prints*, pages arXiv–2307, 2023.

[27] Zhixuan Chu, Hongyan Hao, Xin Ouyang, Simeng Wang, Yan Wang, Yue Shen, Jinjie Gu, Qing Cui, Longfei Li, Siqiao Xue, et al. Leveraging large language models for pre-trained recommender systems. *arXiv preprint arXiv:2308.10837*, 2023.

[28] Wenyue Hua, Shuyuan Xu, Yingqiang Ge, and Yongfeng Zhang. How to index item ids for recommendation foundation models. *arXiv preprint arXiv:2305.06569*, 2023.

[29] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*, 2021.

[30] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, volume 30, 2017.

[31] Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In *NAACL*, pages 4171–4186, 2019.

[32] Bonan Min, Hayley Ross, Elior Sulem, Amir Pouran Ben Veyseh, Thien Huu Nguyen, Oscar Sainz, Eneko Agirre, Ilana Heintz, and Dan Roth. Recent advances in natural language processing via large pre-trained language models: A survey. *ACM CSUR*, 56(2):1–40, 2023.

[33] Peng Liu, Lemei Zhang, and Jon Atle Gulla. Pre-train, prompt and recommendation: A comprehensive survey of language modelling paradigm adaptations in recommender systems. *arXiv preprint arXiv:2302.03735*, 2023.

[34] Jianghao Lin, Xinyi Dai, Yunjia Xi, Weiwen Liu, Bo Chen, Xiangyang Li, Chenxu Zhu, Huifeng Guo, Yong Yu, Ruiming Tang, et al. How can recommender systems benefit from large language models: A survey. *arXiv preprint arXiv:2306.05817*, 2023.

[35] Keqin Bao, Jizhi Zhang, Yang Zhang, Wenjie Wang, Fuli Feng, and Xiangnan He. TallRec: An effective and efficient tuning framework to align large language model with recommendation. *arXiv preprint arXiv:2305.00447*, 2023.

[36] Yifan Hu, Yehuda Koren, and Chris Volinsky. Collaborative filtering for implicit feedback datasets. In *ICDM*, pages 263–272, 2008.

[37] Kevin P Murphy. *Machine learning: A probabilistic perspective.* MIT press, 2012.

[38] Kun Zhou, Hui Wang, Wayne Xin Zhao, Yutao Zhu, Sirui Wang, Fuzheng Zhang, Zhongyuan Wang, and Ji-Rong Wen. S3-Rec: Self-supervised learning for sequential recommendation with mutual information maximization. In *CIKM*, pages 1893–1902, 2020.

[39] Dawen Liang, Rahul G Krishnan, Matthew D Hoffman, and Tony Jebara. Variational autoencoders for collaborative filtering. In *WWW*, pages 689–698, 2018.

[40] Yaochen Zhu and Zhenzhong Chen. Mutually-regularized dual collaborative variational auto-encoder for recommendation systems. In *WWW*, pages 2379–2387, 2022.

[41] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer. In *CIKM*, pages 1441–1450, 2019.

[42] Harald Steck. Embarrassingly shallow autoencoders for sparse data. In *WWW*, pages 3251–3257, 2019.

[43] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. In *UAI*, pages 452–461, 2009.

Appendix

Table 3: Statistics of the datasets. #Feat. stands for number of textual features (i.e., # reviews for AM/Yelp datasets, and #user biography+#job descriptions for the LinkedIn dataset.

Dataset	#Int.	#Users	#Items	Sparsity	#Feat.
AM-Beauty	94,148	10, 553	6, 086	99.85%	70,604
AM-Toys	95,420	11, 268	7, 309	99.88%	70,784
AM-Sports	185,718	22, 686	12, 301	99.93%	137,618
Yelp	292,017	28, 330	18, 775	99.94%	224,825
LinkedIn	90,173	22, 391	1, 071	99.62%	23,362

A TECHNICAL DETAILS

A.1 Implementation of Soft+Hard Prompting

To implement the soft+hard prompting strategy discussed in Section 3.3.2, for decoder-only LLMs such as GPT, we can generate only the "keys" and "values" for the heterogeneous tokens in the prompts $\mathbf{x}_i^{r,p}$, $\mathbf{x}_{ij}^{uv,p}$, and use the "query" of the last token as the start to generate the homogeneous tokens of the main texts $\mathbf{x}_i^{r,m}$, $\mathbf{x}_{ij}^{uv,m}$ for language modeling. For encoder-decoder-based LLMs such as T5, a natural thought is to input the prompts $\mathbf{x}_i^{r,p}$, $\mathbf{x}_{ij}^{uv,p}$ in the encoder, and use the decoder to generate the main texts $\mathbf{x}_i^{r,m}$, $\mathbf{x}_{ij}^{uv,m}$.

A.2 Mutually Regularized Objective for Recommendation-Oriented Finetuning

If we denote the multinomial probability obtained from the RecLLM prediction head f_{rec} as $\hat{\mathbf{r}}_i^{hold}$, and denote the stacked item collaborative token embeddings of items interacted by user i as $\mathbf{Z}_i^{l,v}$, the **rec-step** objective of the recommendation-oriented finetuning (regularized with the content LLM) can be formulated as:

$$\mathcal{L}^{\text{MAP}}_{\text{rec\_step}}\left(\mathbf{z}_i^{l,u}, \mathbf{Z}_i^{l,v}; \theta\right) = \underbrace{-\sum_k r_{ik}^{hold} \ln \hat{r}_{ik}^{hold}}_{\text{Multinomial NLL Loss}} \underbrace{+\frac{\lambda_l}{2}\left\|\mathbf{z}_i^{l,u}\right\|_2^2 + \sum_k \frac{\lambda_l}{2}\left\|\mathbf{z}_{ik}^{l,v}\right\|_2^2}_{\text{Prior loss}} \underbrace{+\frac{\lambda_c}{2}\left\|\mathbf{z}_i^{l,u} - \hat{\mathbf{z}}_i^{c,u}\right\|_2^2 + \sum_k \frac{\lambda_c}{2}\cdot\left\|\mathbf{z}_{ik}^{l,v} - \hat{\mathbf{z}}_{ik}^{c,v}\right\|_2^2}_{\text{MR loss with content LLM}} + C_{rec}, \tag{10}$$

where NLL stands for negative log-likelihood, and C_{rec} is the constant irrelevant for the optimization purpose. From the form of the multinomial NLL loss we can find that, when finetuning the RecLLM according to Eq. (10), the $\mathbf{h}_{l,i,-1}^{rec}$ output by the CLLM4Rec base model $\hat{llm}_l$, which can be viewed as the latent variable summarizing the historical interaction of user i, is encouraged to be similar to the collaborative embeddings of all the interacted items.

B EXPERIMENTS

B.1 Statistics of the Datasets

The statistics of the public datasets and the LinkedIn recommendation dataset in the main paper are summarized in Table 3.

Table 4: Comparison between CLLM4Rec with more backbones and more baselines on three Amazon Review datasets.

AM-Beauty	**Recall@20**	**Recall@40**	**NDCG@100**
Multi-VAE	0.1295	0.1720	0.0835
EASE	0.1325	0.1757	0.0904
BPR	0.1391	0.1803	0.0862
MD-CVAE	0.1472	0.2058	0.0976
BERT4Rec	0.1126	0.1677	0.0781
S^{3}Rec	0.1354	0.1789	0.0867
CLLM4Rec-T5	0.1538	0.2105	0.1052
CLLM4Rec-LLaMA	0.1614	0.2297	0.1103
CLLM4Rec-GPT2	**0.1656**	**0.2323**	**0.1118**

AM-Toys	**Recall@20**	**Recall@40**	**NDCG@100**
Multi-VAE	0.1076	0.1558	0.0781
EASE	0.1082	0.1561	0.0787
BPR	0.1124	0.1579	0.0824
MD-CVAE	0.1291	0.1804	0.0844
BERT4Rec	0.0853	0.1375	0.0532
S^{3}Rec	0.1064	0.1524	0.0665
CLLM4Rec-T5	0.1328	0.1840	0.0851
CLLM4Rec-LLaMA	0.1369	0.1877	0.0896
CLLM4Rec-GPT2	**0.1436**	**0.1933**	**0.0918**

AM-Sports	**Recall@20**	**Recall@40**	**NDCG@100**
Multi-VAE	0.0659	0.0975	0.0446
EASE	0.0694	0.1038	0.0501
BPR	0.0756	0.1057	0.0539
MD-CVAE	0.0714	0.1180	0.0514
BERT4Rec	0.0521	0.0701	0.0305
S^{3}Rec	0.0616	0.0813	0.0438
CLLM4Rec-T5	0.0845	0.1226	0.0589
CLLM4Rec-LLaMA	**0.0938**	**0.1369**	**0.0648**
CLLM4Rec-GPT2	0.0926	0.1351	0.0634

B.2 Implementation Details for the GPT-2-based CLLM4Rec

In this section, we introduce the implementation details for the GPT-2 based CLLM4Rec used in the main paper. During the training stage, we first optimize the content LLM as Eq. (5) via language modeling for 10 epochs to warm up the user/item content token embeddings. Then, in the mutually regularized pretraining stage, we alternately train the collaborative and content LLMs as specified in Eqs. (7) and (8) for 100 epochs. Finally, we conduct the recommendation-oriented finetuning for 150 epochs, where the RecLLM is monitored with metrics Recall@20, Recall@40, and NDCG@100 calculated on the validation set as with [39]. RecLLM with the best performance is logged and evaluated on the test set as the final results. The prior precision λ_c in Eqs. (7) and (8) is an important hyperparameter that controls the strength of mutual regularization. In the main paper, we first fix its value to the optimal one found by grid search and compare it with other baselines in Section 4.2, and then we discuss its influence in Section 4.3.

B.3 Additional Results

B.3.1 ***Implementation Details for More Backbones.*** In the appendix, we report the experiments of CLLM4Rec with two more LLM backbones to demonstrate the generalization ability of the proposed CLLM4Rec. The first backbone we consider is the T5-base model [10], which is an encoder-decoder-based LLM with 32,128 vocab tokens (the last 28 tokens are empty), and each token is associated with a 768-dimensional vocab embedding. Another backbone is the LLaMA-7B model [11], which has the same number of tokens as the T5 model (as both use the sentence-piece tokenizer), and each token is associated with a 4,096-dimensional token embedding. For the non-symmetric LLM models where the weights of the LM prediction head are not tied with the vocab token embeddings, we randomly initialize the weights for the item prediction head for collaborative LLM (see Eq. (4) for details) and learn them together with the item collaborative token embeddings in both the pretraining and finetuning stages. Model training generally follows similar steps as the model with GPT-2 backbone described in Section B.1, where we first warm up the content LLM as Eq. (5) for ten epochs. Then, we conduct the mutually-regularized pretraining as Eqs. (7), (8) for 100 epochs, and continue for the recommendation-oriented finetuning as specified by Eq. (10) for 150 more epochs.

B.3.2 ***More Baselines.*** In addition, we report the experiments of two more strong ID-based baselines, i.e., EASE [42] and BPR [43] in this Appendix. Specifically, EASE improves over the Multi-VAE by introducing a single-layer auto-encoder with constraints of no self-reconstruction, which shows better generalization ability due to its reduced variance and more suitable inductive bias faced with the sparse rating data. For the BPR model, we concatenate the user/item bag-of-word textual features with the user/item collaborative latent variables when optimizing the ranking-based objective.

B.3.3 ***Results & Analysis.*** The additional experimental results are summarized in Table 4. From Table 4 we can find that, although CLLM4Rec-T5 generally outperforms the ID-based baselines and shallow LM-based baselines, its performance is consistently worse than the CLLM4Rec-GPT2 model. The reason for the overall inferior performance of CLLM4Rec with T5 backbone can be two-fold. First, we note that the weights in T5 are initialized with unit variance, whereas weights in GPT-2 are initialized with a variance of 0.02. Therefore, weights in T5 have much larger numerical values, which leads to large update steps. Therefore, the training of CLLM4Rec-T5 is not as stable as CLLM4Rec-GPT2. In addition, in the finetuning stage of T5, the prompts are generally used to guide the macro behavior of the model. e.g., changing model behavior from question answering to machine generation via prompt "Translate English to French:". Therefore, another reason could be the mismatch between the original T5 prompts and the prompts intended to be used in CLLM4Rec. In addition, we can find that CLLM4Rec with the larger LLaMA-7B backbone cannot outperform CLLM4Rec-GPT2 on two smaller AM-Beauty and AM-Sports datasets, where the model can overfit on limited data. However, CLLM4Rec-LLaMA performs slightly better than CLLM4Rec-GPT2 on the comparatively large AM-Sports dataset, which demonstrates the generalization ability of CLLM4Rec with both larger models and larger data.