Improving LLMs for Recommendation with Out-Of-Vocabulary Tokens

Ting-Ji Huang [1 2] **Jia-Qi Yang** [1 2] **Chunxu Shen** [3] **Kai-Qi Liu** [4] **De-Chuan Zhan** [1 2] **Han-Jia Ye** [1 2]

Abstract

Characterizing users and items through grouped vector representations is crucial for various tasks in recommender systems. Recent approaches attempt to apply Large Language Models (LLMs) in recommendation through an instruction-based format, where real items are represented with grouped words formed from in-vocabulary tokens. However, these tokens do not effectively represent items since their meanings are derived from pre-training on natural language tasks, which limits the model's ability to learn meaningful user–item relationships. In this paper, we explore how to effectively characterize users and items in LLM-based recommender systems from the tokenizing strategy view. We demonstrate the necessity of using out-of-vocabulary (OOV) tokens for the characterization of items and users, and propose a well-constructed way of these OOV tokens. By clustering the learned representations from historical user-item interactions, we make the representations of user/item combinations share the same OOV tokens if they have similar properties. This construction allows us to capture user/item relationships well (*memorization*) and preserve the diversity of descriptions of users and items (*diversity*). Furthermore, integrating these OOV tokens into the LLM's vocabulary allows for better distinction between users and items and enhanced capture of user-item relationships during fine-tuning on downstream tasks. Our proposed framework outperforms existing state-of-the-art methods across various downstream recommendation tasks. Code is available at https://github.com/Tingji2419/META-ID.
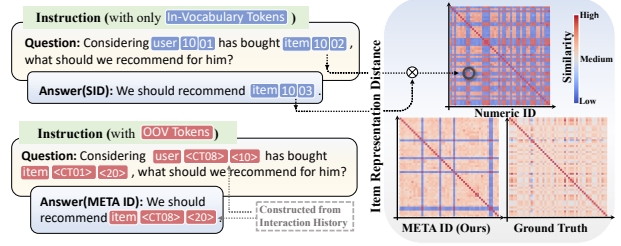
Figure 1: Left: Comparison of tokenizing strategies using In-Vocabulary (IV) and Out-of-Vocabulary (OOV) tokens to characterize items. The IV scenario leverages numeric identifiers (Numeric ID) for items, while the OOV scenario constructs OOV tokens from user-item interaction history (META ID). Right: Similarity heatmaps illustrate the alignment of Numeric ID, META ID, and ground truth, demonstrating the effectiveness of META ID in capturing relationship patterns. Ground truth is calculated from Equation 4.

1. Introduction

Modern recommender systems (RS) play a crucial role in various applications like video recommendation (Davidson et al., 2010), e-commerce (Chen et al., 2019), and social networking (Fan et al., 2019). The recent advent of large language models (LLMs) offers a new direction of exploration in this realm. Models such as T5 (Raffel et al., 2020a) and LLaMA (Touvron et al., 2023), training on massive natural language data, achieve impressive language understanding capabilities in text generation and conversation tasks (Zhao et al., 2023). Their success drives explorations to use pre-trained LLMs as model backbones for handling various recommendation tasks like sequential recommendation, rating, and explanation task (Bi et al., 2022; Wu et al., 2021). In such frameworks, all input features including ids, attributes, and contextual signals, are described as textual queries, and the LLM is expected to generate task-specific labels as responses (Cui et al., 2022; Geng et al., 2022; Zhai et al., 2024; Hong et al., 2025). After being trained in this instruction-based format, the model can be prompted to provide recommendations for a user or generate explanations for those recommendations.

How to characterize users and items so that LLMs can learn their personalized information during fine-tuning? Given that tokens are the fundamental units of input

---
[1]School of Artificial Intelligence, Nanjing University [2]National Key Laboratory for Novel Software Technology, Nanjing University [3]WeChat Technical Architecture Department, Tencent Inc. [4]Software Institute, Nanjing University. Correspondence to: Han-Jia Ye <yehj@lamda.nju.edu.cn>.

for LLMs, where each represents the smallest semantic element the model can process, it is natural to allocate specific tokens to represent users and items, allowing the model to learn their characteristics through token-level associations. While early works (Cui et al., 2022) directly represent a real item by text tokens (*e.g.*, "blue", "casual", "T-shirt"), LLMs might answer text that does not correspond to a real existing item (Hua et al., 2023; Ji et al., 2023). Recent advancements explore representing users/items with *Identifiers* (*IDs*) (Geng et al., 2022; Wang et al., 2022; Gao et al., 2023; Bao et al., 2023). As illustrated in Figure 1, a real item (Item No.2024) can be indexed with a compound word combined with several in-vocabulary tokens ("item", "20", "24"), and we call such a token combination as an ID. Each ID uniquely identifies a real item.

Characterizing users and items with in-vocabulary tokens is straightforward but also introduces the *mapping problem*, where it lacks diversity to align limited in-vocabulary tokens to billions of items in recommender systems, and token combinations are prone to lead to language conflicts (Bao et al., 2023; Hua et al., 2023). As shown in Figure 1, using numeric tokens to represent items leads to similar representations of distinctive items. One solution is to create brand new tokens (OOV tokens) specifically allocated for IDs to allow distinction, as in the previous work on Collaborative ID (CID) (Hua et al., 2023). However, we need to train these new token embeddings from scratch, which often requires a large amount of data, and our experiments also indicate that it is hard for LLM to capture the collaborative correlations of users/items during fine-tuning (see Section 4). Therefore, we should utilize redefined OOV tokens capable of capturing user/item correlations (*memorization*) while also distinguishing different items (*diversity*), two dimensions that are positively correlated for the performance of recommendation tasks.

In this paper, we present META ID (META-path-guided IDentifier), a framework for characterizing users/items using out-of-vocabulary (OOV) tokens for LLM-based recommendations. Initially, we generate meta-paths to represent user-item interactions and then obtain user and item representations from a skip-gram model trained on these meta-paths. A meta-path is a sequence that represents interactions between users and items in a graph structure. By clustering these meta-path-based representations, we create hierarchical groups that serve as OOV tokens for constructing user and item IDs. This approach extends beyond previous research, which has predominantly focused on item IDs (Geng et al., 2022; Rajput et al., 2023), by making the representations of users/item IDs share the same OOV tokens if they have similar properties. Finally, integrating these OOV tokens into the LLM's vocabulary allows for better diversity and enhanced memorization during fine-tuning on downstream recommendation tasks. Additionally, we

align the token embedding layer of LLMs with a linear transformation layer to enrich the OOV token representations as an augmentation. Our contributions are as follows:

- We introduce memorization and diversity scores to evaluate ID representations in LLM-based recommender systems, focusing on capturing user/item correlations and ensuring diversity.
- We develop META ID, which uses out-of-vocabulary tokens to characterize users/items, enhancing the memorization and diversity of their representations for LLMs.
- The experiments show that META ID improves memorization and diversity score, which leads to improvements in various recommendation tasks.

2. Related Work

Characterizing Items by IDs. Modern recommendation models usually use unique IDs to represent users and items, which are subsequently converted to embedding vectors as learnable parameters (Fan et al., 2023). Common approaches include matrix factorization (Koren et al., 2009a; Rendle et al., 2009), two-tower models (Wang et al., 2021) and deep neural networks (Sun et al., 2019; Kang and McAuley, 2018; Zhou et al., 2020; Zhang et al., 2023b; Xie et al., 2022; Wang et al., 2023), which make predictions by examining historical user-item interactions to identify behavioral patterns and enable collaborative recommendations (Koren et al., 2022). To avoid large-scale vocabularies, one recent study proposes using hierarchical sequential transduction units to handle non-stationary vocabulary (Zhai et al., 2024). While some recent approaches adopt the concept of ID to represent the token combinations that characterize items in LLMs (Geng et al., 2022; Hua et al., 2023). We follow previous studies and also call these token combinations as ID. This paper aims to combine LLM with ID more efficiently by proposing a new ID construction method.

Instruction Tuning for Recommendation. The integration of Large Language Models (LLMs) into diverse tasks has seen significant growth recently. Recent works fine-tuned pretrained language models on large-scale NLP datasets verbalized via human-readable prompts (Sanh et al., 2022; Wei et al., 2022). These instruction tuning methods design prompts containing detailed task descriptions and adhere more to the natural language format. Driven by their exceptional natural language processing capabilities, researchers aim to transfer their linguistic ability to enhance recommender systems (Li et al., 2021; Zhang et al., 2023c; Hong et al., 2025). These LLMs process user interactions as sequences of tokens and, through fine-tuning, predict users' future interests based on past activities (Zhang et al., 2021; Chen, 2023). Moreover, some studies reframe tasks like retrieval, rating, and explanation generation as language comprehension tasks (Cui et al., 2022; Geng et al., 2022;

Li et al., 2023b), allowing LLMs to function as multi-task recommenders, producing recommendations and explanations with a unified architecture. In this paper, we apply LLMs based on instruction tuning of multi-task scenarios.

ID Construction with Tokens. Recent studies explore using in-vocabulary tokens to characterize items in LLMs, where users and items are represented by token combinations called IDs. Early efforts such as P5 (Geng et al., 2022) convert user-item interactions into natural language formats using numeric IDs constructed of in-vocabulary tokens of the T5 model (Raffel et al., 2020a). Further, sequential ID (SID) and collaborative ID (CID) are developed to enhance item information sharing (Hua et al., 2023). Some recent study explores the traditional ID embeddings with LLMs (Rajput et al., 2023; Li et al., 2023a; Zhang et al., 2023a). Despite these advancements in ID construction, challenges remain in the lack of ID construction criteria and the focus on item IDs. In contrast, our META ID approach assigns memorization and diversity scores for evaluation and constructs both user and item IDs, which improves the characterization of users and items.

3. Preliminary

We describe recommendation tasks (*e.g.*, sequential recommendation) under the instruction tuning setting, in which all data such as user-item interactions are converted to natural language sequences in a question&answer format (*e.g.*, Input: *Please recommend an item for user 2024 consider he has purchased item 2023*; Output: *item 2024*). More examples are presented in Appendix C.

3.1. Instruction Tuning for Recommendation

Given a user set $\mathbf{U}$, an item set $\mathbf{I}$ and a token set $\mathbf{T}$ from a LLM's vocabulary, we formulate a recommendation task as a natural language instruction that pairs a sequence of n input tokens $\mathbf{x} = (x_1, ..., x_n)$ with a sequence of m corresponding label tokens $\mathbf{y} = (y_1, ..., y_m)$. The goal is to train and LLM $\mathcal{M}_{\boldsymbol{\theta}}$ to generate $\mathbf{y}$ given $\mathbf{x}$. For simplification, each token in the input or output sequence is denoted as x_i or y_j, $x_i, y_j \in \mathbf{T}$, respectively. A user $u \in \mathbf{U}$ is represented by a set of pre-assigned tokens $\boldsymbol{x}_u = (x_{u_1}, x_{u_2}, ...)$. Similarly, an item $i \in \mathbf{I}$ is represented by a sequence of tokens $\boldsymbol{x}_i = (x_{i_1}, x_{i_2}, ...)$.

The LLM $\mathcal{M}_{\boldsymbol{\theta}}$ employs a token embedding layer $\boldsymbol{E}(\cdot)$ with parameters $\boldsymbol{\theta_E} \in \boldsymbol{\theta}$, functioning as a lookup table that transforms each input token into a token embedding. It subsequently predicts the probability distribution of label tokens by forward propagation. The training objective is to minimize the negative log-likelihood of the label tokens conditioned on the input sequence and previously generated

tokens, formulated as:

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \mathcal{L}_{\boldsymbol{\theta}} = -\sum_{l=1}^{|\mathbf{y}|} \log P_{\boldsymbol{\theta}}\big(y_l \mid y_{<l}, \boldsymbol{E}(\mathbf{x})\big), \quad (1)$$

where $\boldsymbol{\theta}^*$ represents the optimal set of parameters that we aim to learn. This supervised learning approach helps the model to internalize personalized information about users and items through the token embeddings of their respective IDs, $\boldsymbol{x}_u$ and $\boldsymbol{x}_i$.

3.2. Represent Users/Items with In-Vocabulary Tokens

The token embedding layer $\boldsymbol{E}$ in LLMs transforms input tokens into corresponding token embeddings. Since users and items are also represented by IDs constructed of tokens, the corresponding token embeddings of ID $\boldsymbol{x}_u$ and $\boldsymbol{x}_i$ are the key to assess how LLMs capture and represent different items. We define the *ID representation* $\mathbf{e}_i$ for an item ID $\boldsymbol{x}_i$ as the average embeddings of its token combinations:

$$\mathbf{e}_i = \frac{1}{|\boldsymbol{x}_i|} \sum_{x_{i_t} \in \boldsymbol{x}_i} \boldsymbol{E}(x_{i_t}), \quad (2)$$

where $\boldsymbol{E}(x_{i_t})$ denotes the embedding vector of a token x_{i_t}. For example, an item identified by "item 2024" is represented by the average embedding vectors of three tokens ("item", "20", "24").

4. Metric for Representation Evaluation

As mentioned above, using numeric IDs to represent items leads to similar representations of distinctive items. For an intuitive explanation, we first visualize the cosine similarity matrix between items using heatmaps in Figure 1, where Random ID (RID) means assigning each item with a random number. It shows that RID and SID result in a large number of similar items due to semantic conflicts, and META ID constructed of OOV tokens shows distinguished similarity closer to the ground truth. We further plot the ID representations using T-SNE visualization in Figure 2b. It is clear that RID and SID (using in-vocabulary tokens) shrink in a small place relative to CID (using OOV tokens), reflecting in-vocabulary tokens lacking expressive power for distinctive users and items. To quantify this, here we introduce two metrics to assess the *memorization* and *diversity* of ID representations.

Diversity Score (DS) is a metric designed to quantify the diversity of ID representations within LLMs. Items represented by ID representations should be easily distinguishable for the model. For a pair of items i and j, their differentiation can be calculated using the Kullback-Leibler (KL) divergence on ID representations $\mathbf{e}_i \in \mathbf{E}$ and $\mathbf{e}_j \in \mathbf{E}$ from Equation 2, which reflects how distinct they are in the

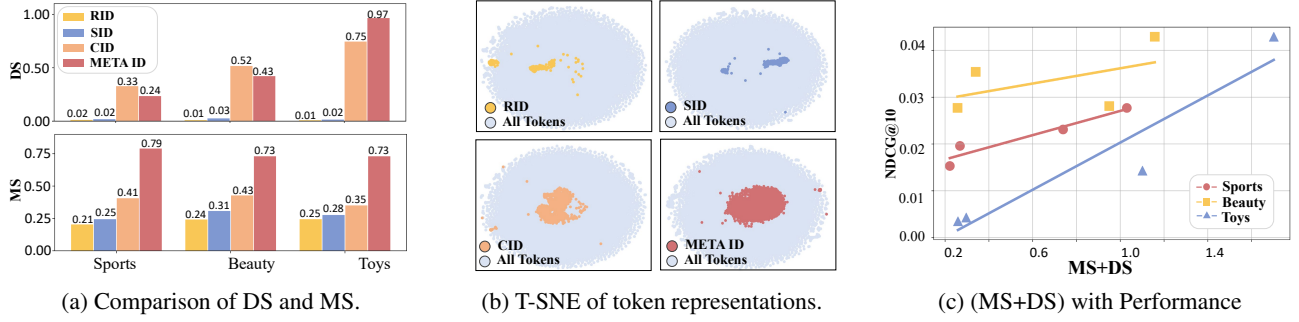(a) Comparison of DS and MS.   (b) T-SNE of token representations.   (c) (MS+DS) with Performance

Figure 2: (a) Comparison of Diversity Score (DS) and Memorization Score (MS) of ID representations from RID, SID, CID and META ID; (b) T-SNE visualizations of ID representations on Sports dataset; (c) Correlation between (DS + MS) with NDCG@10 on sequential recommendation task.

model's embedding space:

$$\text{DS}(\mathbf{E}) = \frac{1}{2N} \sum_{i_i, j \in \mathbf{I}}^{N} \left[D_{KL}(\mathbf{e}_i || \mathbf{e}_j) + D_{KL}(\mathbf{e}_j || \mathbf{e}_i) \right]. \quad (3)$$

Here, the $\mathbf{E} = [\mathbf{e}_1^\top, \ldots, \mathbf{e}_{|\mathbf{i}|}^\top]$ means the embedding matrix of all items $i \in \mathbf{I}$. N represents the number of randomly selected item pairs, where we apply a sampling strategy to reduce the computational complexity, and we give a convergence analysis in Figure 5.

Memorization Score (MS) quantifies the relationships captured by ID representations through measuring the similarity between items and users. We use the adjusted cosine similarity formula (Sarwar et al., 2001), to provides ground truth relational values between users and items, which are given by:

$$\text{sim}(i,j) = \frac{\sum_{u \in U} (R_{u,i} - \bar{R}_u) \cdot (R_{u,j} - \bar{R}_u)}{\sqrt{\sum_{u \in U} (R_{u,i} - \bar{R}_u)^2 \cdot \sum_{u \in U} (R_{u,j} - \bar{R}_u)^2}}, \quad (4)$$

where $R_{u,i}$ and $R_{u,j}$ denote user u's ratings for items i and j, and $\bar{R}_u$ is user u's average ratings. To assess the relationship captured by learned ID representations, we employ Mean Square Error (MSE) to calculate the similarity bias on their cosine similarity with their corresponding ground truth relation values, which forms the basis of the MS:

$$\text{MS}(\mathbf{E}) = \frac{2}{|\mathbf{I}|(|\mathbf{I}| - 1)} \sum_{i, j \in N, i < j} \left(\frac{\mathbf{e}_i^\top \mathbf{e}_j}{|\mathbf{e}_i||\mathbf{e}_j|} - \text{sim}(i,j) \right)^2. \quad (5)$$

Our quantitative assessment, based on the metrics we introduced, reveals intriguing insights into the quality of these ID representations. Figure 2a illustrates that IDs constructed from in-vocabulary tokens (RID and SID) perform poorly in terms of both diversity and memorization. Although the use of OOV tokens (CID) enhances diversity, its memorization

score is unsatisfactory. These findings highlight the need to further develop more effective forms of OOV tokens for ID construction, aiming to improve LLMs' ability to capture user/item correlations and distinguish items.

5. META ID

We now introduce the META ID framework to enhance LLMs in recommendation tasks. META ID involves creating out-of-vocabulary (OOV) tokens for constructing user and item IDs, which provide a rich, expressive space and encapsulate comprehensive, collaborative information. As illustrated in Figure 3, our process begins with sampling meta-paths from user-item interaction history. We then apply a skip-gram model to learn user and item representations from these meta-path sequences. This process ensures that users' and items' features are projected into a shared space to capture their interaction relationship better. This is followed by K-Means clustering to group similar users or items, after which unique OOV tokens are assigned to each cluster to construct the META ID.

5.1. Meta-path-based Embedding

We frame user-item interactions within a graph embedding learning paradigm (Dong et al., 2017), constructing an interaction graph composed of user nodes U and item nodes I, linked by interaction history with ratings R. The core of this embedding learning strategy involves meta-paths (Dong et al., 2017) — a sequence of connections that reflect composite relationships within the graph. Our primary meta-path, denoted as *U-I-U*, materializes when users consistently rate an item with the same score r, forming a path:

$$p = \{U_1 \xrightarrow{r} I_2 \xrightarrow{r} U_3 \xrightarrow{r} I_4 ... \xrightarrow{r} U_k\}_{r=1}^{|R|}. \quad (6)$$

We employ a skip-gram model to capture the interaction dynamics represented by these meta-paths. This model

4

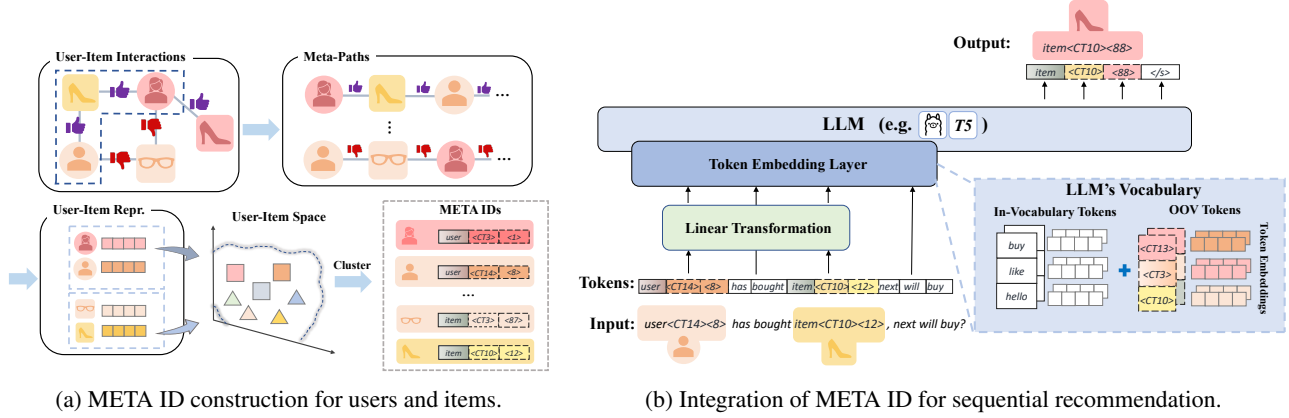(a) META ID construction for users and items.   (b) Integration of META ID for sequential recommendation.

Figure 3: Illustration of the proposed framework: (a) The first step involves sampling meta-paths to train a skip-gram model to learn the representation of users and items. Following this, users and items are represented by META IDs constructed of OOV token combinations through clustering their representations. (b) A large language model (LLM) integrated with META ID is utilized for sequential recommendation. The LLM first encodes the input sequence into tokens and then lookup their embeddings through the token embedding layer. Here, the OOV tokens undergo an extra transformation layer before the token embedding layer of LLMs as a representation augmentation.

is trained on sequences generated from meta-path-based random walks (Dong et al., 2017), producing representations $\mathbf{W_U} = [\mathbf{w}_{u_1}^\top, ..., \mathbf{w}_{u_{|\mathbf{U}|}}^\top] \in \mathbb{R}^{d \times |\mathbf{U}|}$ and $\mathbf{W_I} = [\mathbf{w}_{i_1}^\top, ..., \mathbf{w}_{i_{|\mathbf{I}|}}^\top] \in \mathbb{R}^{d \times |\mathbf{I}|}$, denoted the representations for $|\mathbf{U}|$ users and $|\mathbf{I}|$ items in a d-dimension space respectively.

5.2. OOV Token Generation

To balance diversity and memorization in large-scale recommendation systems, we adopt a cluster-based approach. Representations derived from user-item interactions are organized into a shared embedding space and segmented into G clusters. Each cluster center, μ_g, is defined by the average of representations within that cluster, effectively capturing the collective characteristics of its members. These centroids then categorize each user and item, providing a refined foundation for constructing granular IDs. The generation process is a two-step procedure:

1). Assign coarse-grained tokens based on centroids. We first cluster the learned representations $\mathbf{W}$ into G clusters. For each cluster g, let $\mathbb{I}_g = \{i \mid g_i = g\}$ denote the index set of representations assigned to it. The centroid $\boldsymbol{\mu}_g$ is then computed as:

$$\boldsymbol{\mu}_g = \left\{ \frac{1}{|\mathbb{I}_g|} \sum_{i \in \mathbb{I}_g} \mathbf{w}_i \right\}_{g=1}^G. \tag{7}$$

As a coarse-grained distinction between users and items, we assign an OOV token "$\langle CT_i \rangle$" to each centroid.

2). Assign fine-grained tokens based on distance. Within each cluster, assign a token based on the distance to the

centroid. In detail, we assign fine-grained tokens "$\langle CI_i \rangle$" in ascending order according to their distance from the cluster centroid, and use it as a fine-grained token.

The resulting identifiers, META ID, combine a coarse-grained token and a fine-grained one that uniquely identifies each user or item within that cluster. For example, an item might be represented as "$\langle Item \rangle \langle CT_i \rangle \langle CI_i \rangle$" labeled with three tokens, "$\langle Item \rangle$", "$\langle CT_i \rangle$", "$\langle CI_i \rangle$", where "$\langle Item \rangle$" denotes it as an item, "$\langle CT_i \rangle$" is its coarse-grained token and "$\langle CI_i \rangle$" is its fine-grained token.

This clustering and labeling process effectively compresses the vocabulary needed for ID representation while preserving the rich information necessary for recommendation tasks. In our implementation, we apply K-Means clustering (Arthur and Vassilvitskii, 2007), utilizing cosine similarity to measure the affinity of representations to cluster centers. This method simplifies the complexity of representations space management and proves to be robust in our experimental validations of Section 6.4.

5.3. Integration of META ID with LLMs

The integration of OOV tokens of META ID, denoted as $\mathbf{x}_{\text{OOV}}$, with LLMs involves expanding the vocabularies. This is achieved by extending the token embedding layer's parameters from $\boldsymbol{\theta_E} \in \mathbb{R}^{|\mathbf{T}| \times d}$ to $\boldsymbol{\theta_{E'}} \in \mathbb{R}^{(|\mathbf{T}|+n) \times d}$, where $|\mathbf{T}|$ is the number of in-vocabulary tokens, n is the number of OOV tokens, and d is the dimension of the token embeddings.

A good initialization helps token embedding learning, here we give a representation augmentation approach different

Table 1: Performance comparison of different methods on sequential recommendation task. META ID (T) and META ID (L) refer to the use of T5 and LLaMA2-7b as the backbone. The best and second-best performance methods are denoted in bold and underlined fonts respectively.

Methods	Sports				Beauty				Toys			
	H@5	N@5	H@10	N@10	H@5	N@5	H@10	N@10	H@5	N@5	H@10	N@10
Caser (Tang and Wang, 2018)	0.0116	0.0072	0.0194	0.0097	0.0205	0.0131	0.0347	0.0176	0.0166	0.0107	0.0270	0.0141
HGN (Ma et al., 2019)	0.0189	0.0120	0.0313	0.0159	0.0325	0.0206	0.0512	0.0266	0.0321	0.0221	0.0497	0.0277
GRU4Rec (Hidasi et al., 2016)	0.0129	0.0086	0.0204	0.0110	0.0164	0.0099	0.0283	0.0137	0.0097	0.0059	0.0176	0.0084
BERT4Rec (Sun et al., 2019)	0.0115	0.0075	0.0191	0.0099	0.0203	0.0124	0.0347	0.0170	0.0116	0.0071	0.0203	0.0099
FDSA (Hao et al., 2023)	0.0182	0.0122	0.0288	0.0156	0.0267	0.0163	0.0407	0.0208	0.0228	0.0140	0.0381	0.0189
SASRec (Kang and McAuley, 2018)	0.0233	0.0154	0.0350	0.0192	0.0387	0.0249	0.0605	0.0318	0.0463	0.0306	0.0675	0.0374
S^3-Rec (Zhou et al., 2020)	0.0251	0.0161	0.0385	0.0204	0.0387	0.0244	0.0647	0.0327	0.0443	0.0294	0.0700	0.0376
CL4SRec (Xie et al., 2022)	0.0219	0.0138	0.0358	0.0182	0.0330	0.0201	0.0546	0.0270	0.0427	0.0244	0.0617	0.0305
TIGER (Rajput et al., 2023)	0.0264	0.0181	0.0400	0.0225	0.0454	<u>0.0320</u>	0.0648	0.0384	<u>0.0521</u>	<u>0.0371</u>	<u>0.0712</u>	<u>0.0412</u>
RID (Hua et al., 2023)	0.0208	0.0122	0.0288	0.0153	0.0213	0.0178	0.0479	0.0277	0.0044	0.0029	0.0062	0.0035
SID (Geng et al., 2022)	0.0223	0.0173	0.0294	0.0196	0.0404	0.0299	0.0573	0.0354	0.0050	0.0031	0.0088	0.0043
CID (Hua et al., 2023)	0.0269	0.0196	0.0378	0.0231	0.0336	0.0227	0.0507	0.0281	0.0172	0.0109	0.0279	0.0143
META ID (T)	<u>0.0322</u>	<u>0.0223</u>	<u>0.0487</u>	<u>0.0277</u>	**0.0510**	**0.0351**	**0.0753**	**0.0429**	**0.0533**	**0.0372**	**0.0761**	**0.0441**
META ID (L)	**0.0392**	**0.0278**	**0.0561**	**0.0332**	<u>0.0458</u>	<u>0.0320</u>	<u>0.0678</u>	<u>0.0391</u>	0.0387	0.0264	0.0535	0.0312

from previous works using random initialization (Geng et al., 2022; Hua et al., 2023). As shown in Figure 3b, the OOV tokens undergo a linear layer $F(\cdot)$ initialized with the category embeddings μ_g from Equation 7. Finally, the training objective for integrating META ID into LLMs is reformulated as:

$$
\begin{aligned}
\boldsymbol{\theta}'^* &= \arg\min_{(\boldsymbol{\theta}', \boldsymbol{F})} \mathcal{L}_{\boldsymbol{\theta}'} \\
&= -\sum_{l=1}^{|\mathbf{y}|} \log P_{\theta'}\big(y_l \mid y_{<l}, \boldsymbol{E}'(\mathbf{x}), \boldsymbol{F}(\mathbf{x}_{\text{OOV}}) \big),
\end{aligned}
\tag{8}
$$

where $\boldsymbol{\theta}'$ now includes $\boldsymbol{F}$, aligning with our modified embedding layer $\boldsymbol{\theta}_{\boldsymbol{E}'}$ to optimize the model's performance with the OOV tokens $\mathbf{x}_{\text{OOV}}$. This training goal ensures the model can effectively distinguish between diverse users and items, improving its ability to capture user-item relationships. By enhancing memorization and diversity of OOV tokens, the model achieves better performance in recommendation tasks, leading to more accurate and personalized recommendations in our experiments.

6. Experiment

We evaluate META ID on five downstream recommendation tasks: sequential recommendation, direct recommendation, rating prediction, explanation generation, and review related tasks. We analyze the influence of critical components in META ID and assess the ID representations through visualization and our proposed metrics. Details of task descriptions and pre-processing are in Appendix C.

6.1. Evaluation on Sequential Recommendation

Setups. We evaluate our META ID framework on three public real-world datasets from the Amazon Product Reviews

dataset (Ni et al., 2019), focusing specifically on Sports, Beauty, and Toys. The datasets are processed following the methodology in P5 (Geng et al., 2022).

Baselines. We compare to a variety of established models (which are described briefly in Appendix C), spanning from CNN-based to LLM-based frameworks. Caser (Tang and Wang, 2018), HGN (Ma et al., 2019), GRU4Rec (Hidasi et al., 2016), BERT4Rec (Sun et al., 2019), FDSA (Hao et al., 2023), SASRec (Kang and McAuley, 2018), S^3-Rec (Zhou et al., 2020), CL4SRec (Xie et al., 2022) and TIGER (Rajput et al., 2023). Specifically, we provide P5 with its variations, equipped with different ID construction strategies like Sequential ID (SID), Random ID (RID), and Collaborative ID (CID) (Hua et al., 2023).

Evaluations. We apply widely accepted metrics, top-k Hit ratio (H@K) and Normalized discounted cumulative gain (N@K) with K = 5, 10 to evaluate the recommendation performance.

Implementation Details. For constructing META IDs, the clustering groups are limited to $|G|$=100 (200 for Toys). For LLM fine-tuning, we consider both encoder-decoder architecture T5-small (Raffel et al., 2020b) and decoder-only architecture LLaMA2-7b (Touvron et al., 2023). We fully fine-tune the T5 model and employ the LoRA (Hu et al., 2022) to fine-tune LLaMA2-7b. Vocabulary sizes of these models are shown in Table 10. For LLM inferencing, we use beam search to generate potential items evaluated under the all-item setting.

Results. Table 1 presents our findings for sequential recommendation task [1]. Our observations are as follows: 1)

---
[1]We show the standard error of the metrics in Table 12.

Table 2: Performance comparison of different methods on direct recommendation task.

Methods	Sports				Beauty				Toys			
	H@5	N@5	H@10	N@10	H@5	N@5	H@10	N@10	H@5	N@5	H@10	N@10
RID (Hua et al., 2023)	0.0030	0.0023	0.0042	0.0027	0.0203	0.0155	0.0276	0.0178	0.0046	0.0030	0.0063	0.0035
SID (Geng et al., 2022)	0.0211	0.0169	0.0267	0.0187	0.0296	0.0226	0.0405	0.0261	0.0025	0.0014	0.0041	0.0019
CID (Hua et al., 2023)	0.0250	0.0189	0.0342	0.0219	0.0216	0.0147	0.0340	0.0187	0.0076	0.0049	0.0014	0.0070
META ID	**0.0357**	**0.0256**	**0.0520**	**0.0308**	**0.0480**	**0.0336**	**0.0689**	**0.0403**	**0.0564**	**0.0391**	**0.0803**	**0.0468**

Table 3: Performance comparison of different methods on rating prediction, explanation generation, and review tasks.

Task Type	Metric	Sports				Beauty			
		RID	SID	CID	**META ID**	RID	SID	CID	**META ID**
Rating	RMSE	1.0382	1.0486	1.0383	**1.0327**	1.2829	1.3098	1.2819	**1.2818**
Explanation	BLEU-1	16.2567	16.5825	16.6121	**16.9005**	18.2299	18.3981	19.3499	**19.5106**
	BLEU-4	2.1782	2.1944	2.2332	**2.3481**	2.9027	2.8071	**3.0626**	3.0592
Review	BLEU-1	7.6140	7.7948	7.6586	**7.8819**	6.2282	6.5055	6.5854	**7.0500**
	BLEU-4	2.3228	1.2406	2.4109	**2.6546**	1.9891	1.2406	1.9718	**2.7485**

META ID demonstrates superior performance on all three datasets, underscoring its robustness. 2) IDs constructed of in-vocabulary tokens, RID and SID, underperform on Toys, suggesting limitations in their recommendation efficacy for LLMs. 3) CID shows marked improvements over RID and SID on Toys dataset, highlighting the benefits of incorporating OOV tokens with collaborative information. 4). While the LLaMA2-7b backbone is better in the Sports dataset, its performance in the Beauty and Toys dataset is not as good as T5, which could be linked to the distinct fine-tuning methodologies applied to these models.

6.2. Evaluation on Various Recommendation Tasks

Setups. To validate META ID's adaptability, we extend our evaluation to include direct recommendation, rating prediction, explanation generation, and review tasks, akin to P5 (Geng et al., 2022). For direct recommendation, the model is asked to recommend item for users directly without providing user's interaction history. For rating prediction, the model predicts a numerical rating between 1 and 5 based on user-item data. For explanation tasks, it generates textual justifications for a user's preference towards an item, while in review tasks, it summarizes lengthy reviews into concise titles.

Evaluations. For direct recommendation, we apply the same metrics as in Section 6.1. For rating prediction, we use MSE metric. For explanation and review tasks, we employ BLEU-1/4 metrics.

Results. For direct recommendation (Table 2), META ID exceeds other methods across datasets in all-item setting. This suggests that META ID effectively model the direct relationship between users and items. The results for the other

three tasks in Table 3, show that META ID significantly improves the BLEU scores Sports and Beauty compared to other methods. This result suggests that META ID can improve performance in text relevance tasks, including the interpretation of recommendations.

6.3. Evaluation of ID Representation

Visualization. The amount of numeric tokens available in LLMs is relatively limited, which complicates the establishment of unique one-to-one ID relationships, and two unrelated items might share the tokens as ID. For an intuitive explanation, we visualized the cosine similarity matrix between items using heatmaps in Figure 1, where we random sample 50 items from the Toys dataset and take their adjusted cosine similarity from Equation 4 as ground truth compared with RID, SID and META ID. RID and SID result in a large number of similar items due to semantic conflicts, while META ID shows distinguished similarity closer to ground truth. This suggests that using META ID allows LLMs to better capture relationships between users and items.

Quantitative Analysis. We quantitatively assess the quality of these ID representations by the proposed two metrics (Section 4): memorization score (MS) and diversity score (DS). Our results, shown in Figure 2a, indicate that constructing IDs of in-vocabulary tokens (RID and SID) perform poorly in the diversity dimension. For intuitive interpretation, we further employ t-SNE visualization to map ID representations and observe a tendency for these tokens to cluster narrowly in Figure 2b. META ID shows robust memorization and diversity across three datasets, reflecting its ability to capture correlations between users and items from historical data while ensuring items remain distinguishable.

Table 4: Performance of different construciton methods for OOV tokens on sequential recommendation task.

| Methods	Sports				Beauty				Toys			
	H@5	N@5	H@10	N@10	H@5	N@5	H@10	N@10	H@5	N@5	H@10	N@10
DBSCAN (Ester et al., 1996)	0.0078	0.0043	0.0145	0.0065	0.0257	0.0168	0.0428	0.0223	0.0180	0.0109	0.0304	0.0149
Spectral (Kluger et al., 2003)	0.0199	0.0124	0.0336	0.0167	0.0360	0.0236	0.0588	0.0310	0.0295	0.0184	0.0514	0.0254
RQ-VAE (Zeghidour et al., 2022)	0.0122	0.0077	0.0171	0.0093	0.0368	0.0254	0.0536	0.0309	**0.0511**	0.0335	0.0667	0.0395
K-Means (Arthur and Vassilvitskii, 2007)	**0.0322**	**0.0223**	**0.0487**	**0.0277**	**0.0510**	**0.0351**	**0.0753**	**0.0429**	0.0503	**0.0352**	**0.0742**	**0.0429**
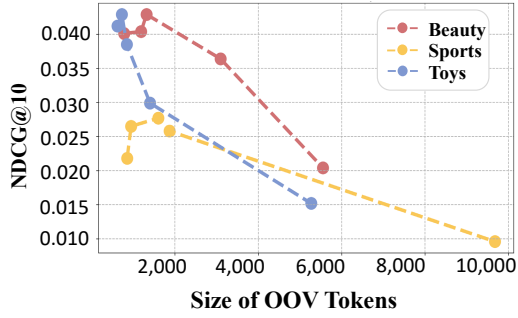
Figure 4: Performance comparison of different OOV token size within META ID on three datasets.

Table 5: Performance comparison of different embedding extraction strategies on sequential recommendation task.

Embedding Type	Beauty			
	H@5	N@5	H@10	N@10
MF-based	0.0220	0.0142	0.0344	0.0207
GCN-Based	0.0267	0.0198	0.0382	0.0212
Meta-Path-Based	**0.0322**	**0.0223**	**0.0487**	**0.0277**

Metrics Analysis. Furthermore, we conduct a correlation analysis to explore the relationship between MS/DS and sequential recommendation performance. We sum the MS and DS of different ID strategies and plot their performance on the sequential recommendation task. As shown in Figure 2c, the sum of the MS and DS positively correlate with NDCG@10, suggesting that memorization and diversity of IDs are two essential properties in recommendation tasks.

6.4. Ablation Studies

We analyze the properties of META ID following the evaluation in Section 6.1, including the impact of grouping methods, the size of OOV tokens, and different indexing ranges on the performance of META ID.

Token Grouping. We study the importance of different grouping methods for OOV token generation in our framework. In Table 4, we compare the performance of DBSCAN (Ester et al., 1996) and Spectral Clustering (Kluger et al.,

2003) against K-Means clustering (Arthur and Vassilvitskii, 2007). We also consider generating OOV tokens with RQ-VAE (Zeghidour et al., 2022) using meta-path-based embeddings. Our results show that simply applying K-Means outperforms other grouping methods in most cases.

Embedding Extraction. To assess the necessity of the meta-path approach, we perform an ablation study where the meta-path structure is removed, and the embeddings are learned purely based on user-item interactions without higher-order relationships with Matrix Factorization (MF) (Koren et al., 2009b). We also compare our approach with more recent graph-based models GCN (Kipf and Welling, 2017), which are designed to capture higher-order relationships in graph-structured data. As shown in Table 5, the meta-path-based embedding used in our approach outperforms than others. This is due to our training on user-item interaction sequences, which allows it to better capture temporal dynamics and patterns in user behavior.

OOV Token Size. Since varying cluster sizes G result in different numbers of OOV tokens, we also investigate the impact of different cluster sizes for META ID in Figure 4. We find that the granularity of token clusters plays a crucial role in recommendation performance. An excessive token scale can introduce noise, reducing the performance. Therefore, finding an optimal token size is vital to ensure that META ID effectively adapts to various datasets' nuances.

7. Conclusion

This study introduces META ID, a method enhancing Large Language Models (LLMs) for recommender systems using OOV tokens. Moving beyond constructing IDs with in-vocabulary tokens, META ID incorporates user-item interaction information to align LLMs more effectively with recommendation tasks. We learn representations from user-item interactions utilizing meta-paths sampling. By clustering these representations we generate OOV tokens to construct META ID. This approach guarantees tokens capturing correlations between users and items from historical data while ensuring distinctiveness among item. Our experiments across various real-world datasets demonstrate META ID's robust performance in diverse recommendation tasks, including sequential and direct recommendation, as well as complex tasks requiring detailed textual responses.

Essentially, META ID effectively combines the capabilities of LLMs with the nuanced requirements of recommendation scenarios, such as planning highly personalized content for users as virtual shopping assistants.

Acknowledgements

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

David Arthur and Sergei Vassilvitskii. 2007. k-means++: the advantages of careful seeding. In *SODA*.

Keqin Bao, Jizhi Zhang, Yang Zhang, Wenjie Wang, Fuli Feng, and Xiangnan He. 2023. TALLRec: An Effective and Efficient Tuning Framework to Align Large Language Model with Recommendation. In *RecSys*.

Qiwei Bi, Jian Li, Lifeng Shang, Xin Jiang, Qun Liu, and Hanfang Yang. 2022. MTRec: Multi-Task Learning over BERT for News Recommendation. In *ACL Findings*.

Qiwei Chen, Huan Zhao, Wei Li, Pipei Huang, and Wenwu Ou. 2019. Behavior Sequence Transformer for E-Commerce Recommendation in Alibaba. In *DLP-KDD*.

Zheng Chen. 2023. PALR: Personalization Aware LLMs for Recommendation. *CoRR* (2023).

Zeyu Cui, Jianxin Ma, Chang Zhou, Jingren Zhou, and Hongxia Yang. 2022. M6-Rec: Generative Pretrained Language Models are Open-Ended Recommender Systems. *CoRR* (2022).

James Davidson, Benjamin Liebald, Junning Liu, Palash Nandy, Taylor Van Vleet, Ullas Gargi, Sujoy Gupta, Yu He, Mike Lambert, Blake Livingston, and Dasarathi Sampath. 2010. The YouTube video recommendation system. In *RecSys*.

Yuxiao Dong, Nitesh V. Chawla, and Ananthram Swami. 2017. metapath2vec: Scalable Representation Learning for Heterogeneous Networks. In *KDD*.

Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. 1996. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In *KDD*.

Wenqi Fan, Yao Ma, Qing Li, Yuan He, Yihong Eric Zhao, Jiliang Tang, and Dawei Yin. 2019. Graph Neural Networks for Social Recommendation. In *WWW*.

Wenqi Fan, Zihuai Zhao, Jiatong Li, Yunqing Liu, Xiaowei Mei, Yiqi Wang, Jiliang Tang, and Qing Li. 2023. Recommender Systems in the Era of Large Language Models (LLMs). *CoRR* (2023).

Yunfan Gao, Tao Sheng, Youlin Xiang, Yun Xiong, Haofen Wang, and Jiawei Zhang. 2023. Chat-REC: Towards Interactive and Explainable LLMs-Augmented Recommender System. *CoRR* (2023).

Shijie Geng, Shuchang Liu, Zuohui Fu, Yingqiang Ge, and Yongfeng Zhang. 2022. Recommendation as Language Processing (RLP): A Unified Pretrain, Personalized Prompt & Predict Paradigm (P5). In *RecSys*.

Yongjing Hao, Tingting Zhang, Pengpeng Zhao, Yanchi Liu, Victor S. Sheng, Jiajie Xu, Guanfeng Liu, and Xiaofang Zhou. 2023. Feature-Level Deeper Self-Attention Network With Contrastive Learning for Sequential Recommendation. *IEEE Trans. Knowl. Data Eng.* (2023).

Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2016. Session-based Recommendations with Recurrent Neural Networks. In *ICLR*.

Minjie Hong, Yan Xia, Zehan Wang, Jieming Zhu, Ye Wang, Sihang Cai, Xiaoda Yang, Quanyu Dai, Zhenhua Dong, Zhimeng Zhang, and Zhou Zhao. 2025. EAGER-LLM: Enhancing Large Language Models as Recommenders through Exogenous Behavior-Semantic Integration. In *WWW*.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-Rank Adaptation of Large Language Models. In *ICLR*.

Wenyue Hua, Shuyuan Xu, Yingqiang Ge, and Yongfeng Zhang. 2023. How to Index Item IDs for Recommendation Foundation Models. (2023).

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Yejin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of Hallucination in Natural Language Generation. *ACM Comput. Surv.* (2023).

Wang-Cheng Kang and Julian J. McAuley. 2018. Self-Attentive Sequential Recommendation. In *ICDM*.

Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *ICLR*.

Yuval Kluger, Ronen Basri, Joseph T Chang, and Mark Gerstein. 2003. Spectral biclustering of microarray data: coclustering genes and conditions. *Genome research* (2003).

Yehuda Koren, Robert M. Bell, and Chris Volinsky. 2009a. Matrix Factorization Techniques for Recommender Systems. *Computer* (2009).

Yehuda Koren, Robert M. Bell, and Chris Volinsky. 2009b. Matrix Factorization Techniques for Recommender Systems. *Computer* (2009).

Yehuda Koren, Steffen Rendle, and Robert M. Bell. 2022. Advances in Collaborative Filtering. In *Recommender Systems Handbook*.

Jinming Li, Wentao Zhang, Tian Wang, Guanglei Xiong, Alan Lu, and Gerard Medioni. 2023b. GPT4Rec: A Generative Framework for Personalized Recommendation and User Interests Interpretation. *CoRR* (2023).

Lei Li, Yongfeng Zhang, and Li Chen. 2020. Generate Neural Template Explanations for Recommendation. In *CIKM*.

Lei Li, Yongfeng Zhang, and Li Chen. 2021. Personalized Transformer for Explainable Recommendation. In *ACL/IJCNLP*.

Xinhang Li, Chong Chen, Xiangyu Zhao, Yong Zhang, and Chunxiao Xing. 2023a. E4SRec: An Elegant Effective Efficient Extensible Solution of Large Language Models for Sequential Recommendation. *CoRR* (2023).

Chen Ma, Peng Kang, and Xue Liu. 2019. Hierarchical Gating Networks for Sequential Recommendation. In *KDD*.

Jianmo Ni, Jiacheng Li, and Julian McAuley. 2019. Justifying Recommendations using Distantly-Labeled Reviews and Fine-Grained Aspects. In *EMNLP-IJCNLP*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020a. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *J. Mach. Learn. Res.* (2020).

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020b. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *J. Mach. Learn. Res.* (2020).

Shashank Rajput, Nikhil Mehta, Anima Singh, Raghunandan H. Keshavan, Trung Vu, Lukasz Heldt, Lichan Hong, Yi Tay, Vinh Q. Tran, Jonah Samost, Maciej Kula, Ed H. Chi, and Maheswaran Sathiamoorthy. 2023. Recommender Systems with Generative Retrieval. *CoRR* (2023).

Ruiyang Ren, Zhaoyang Liu, Yaliang Li, Wayne Xin Zhao, Hui Wang, Bolin Ding, and Ji-Rong Wen. 2020. Sequential Recommendation with Self-Attentive Multi-Adversarial Network. In *SIGIR*.

Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian Personalized Ranking from Implicit Feedback. In *UAI*.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H. Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal V. Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Févry, Jason Alan Fries, Ryan Teehan, Teven Le Scao, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M. Rush. 2022. Multitask Prompted Training Enables Zero-Shot Task Generalization. In *ICLR*.

Badrul Munir Sarwar, George Karypis, Joseph A. Konstan, and John Riedl. 2001. Item-based collaborative filtering recommendation algorithms. In *WWW*.

Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 2019. BERT4Rec: Sequential Recommendation with Bidirectional Encoder Representations from Transformer. In *CIKM*.

Jiaxi Tang and Ke Wang. 2018. Personalized Top-N Sequential Recommendation via Convolutional Sequence Embedding. In *WSDM*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton,

Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open Foundation and Fine-Tuned Chat Models. *CoRR* (2023).

Jinpeng Wang, Jieming Zhu, and Xiuqiang He. 2021. Cross-Batch Negative Sampling for Training Two-Tower Recommenders. In *SIGIR*.

Qi-Wei Wang, Hongyu Lu, Yu Chen, Da-Wei Zhou, De-Chuan Zhan, Ming Chen, and Han-Jia Ye. 2023. Streaming CTR Prediction: Rethinking Recommendation Task for Real-World Streaming Data. *CoRR* (2023).

Wei-Yao Wang, Wei-Wei Du, and Wen-Chih Peng. 2022. RecFormer: personalized temporal-aware transformer for fair music recommendation. In *CIKM CEUR Workshops*.

Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. 2022. Finetuned Language Models are Zero-Shot Learners. In *ICLR*.

Chuhan Wu, Fangzhao Wu, Tao Qi, and Yongfeng Huang. 2021. Empowering News Recommendation with Pre-trained Language Models. In *SIGIR*.

Xu Xie, Fei Sun, Zhaoyang Liu, Shiwen Wu, Jinyang Gao, Jiandong Zhang, Bolin Ding, and Bin Cui. 2022. Contrastive Learning for Sequential Recommendation. In *ICDE*.

Neil Zeghidour, Alejandro Luebs, Ahmed Omran, Jan Skoglund, and Marco Tagliasacchi. 2022. SoundStream: An End-to-End Neural Audio Codec. *IEEE ACM Trans. Audio Speech Lang. Process.* (2022).

Jiaqi Zhai, Lucy Liao, Xing Liu, Yueming Wang, Rui Li, Xuan Cao, Leon Gao, Zhaojie Gong, Fangda Gu, Jiayuan He, Yinghai Lu, and Yu Shi. 2024. Actions Speak Louder than Words: Trillion-Parameter Sequential Transducers for Generative Recommendations. In *ICML*.

Junjie Zhang, Ruobing Xie, Yupeng Hou, Wayne Xin Zhao, Leyu Lin, and Ji-Rong Wen. 2023c. Recommendation as Instruction Following: A Large Language Model Empowered Recommendation Approach. *CoRR* (2023).

Yuhui Zhang, HAO DING, Zeren Shui, Yifei Ma, James Zou, Anoop Deoras, and Hao Wang. 2021. Language Models as Recommender Systems: Evaluations and Limitations. In *NeurIPS Workshop*.

Yang Zhang, Fuli Feng, Jizhi Zhang, Keqin Bao, Qifan Wang, and Xiangnan He. 2023a. CoLLM: Integrating Collaborative Embeddings into Large Language Models for Recommendation. *CoRR* (2023).

Yi-Kai Zhang, Ting-Ji Huang, Yao-Xiang Ding, De-Chuan Zhan, and Han-Jia Ye. 2023b. Model Spider: Learning to Rank Pre-Trained Models Efficiently. In *NeurIPS*.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. 2023. A Survey of Large Language Models. *CoRR* (2023).

Kun Zhou, Hui Wang, Wayne Xin Zhao, Yutao Zhu, Sirui Wang, Fuzheng Zhang, Zhongyuan Wang, and Ji-Rong Wen. 2020. S3-Rec: Self-Supervised Learning for Sequential Recommendation with Mutual Information Maximization. In *CIKM*.

Appendix

We provide details omitted in the main paper.

- Appendix A: Workflow of META ID, encompassing the construction of OOV tokens.

- Appendix B: Details of memorization score (MS) and diversity score (DS).

- Appendix C: Experimental setups and implementation details of META ID.

- Appendix D: Additional experimental result analysis.

- Appendix E: Discussions and limitations of META ID.

A. Details of META ID

In the Section 5 of the main text, we elucidate the comprehensive workflow for generating META ID. This process encompasses three main steps, including (1) the extraction of meta-path-based embedding, (2) the generation of OOV tokens, and (3) the incorporation of META ID with LLMs, thereby handling with various downstream recommendation tasks.

A.1. How to extract the meta-path-based embedding

This section supplements the details of subsection 5.1, *i.e.*, the users / items representations extracted from a skip-gram model, including the sampling process of meta-paths as training data.

In META ID, we enable a skip-gram model to learn effective users / items representations from the sampled meta-paths, which is learning user representations $\mathbf{W}_U$ and item representations $\mathbf{W}_I$. The objective of the skip-gram model learning paradigm is to map the users and items in the meta-paths seqeuences into a lower-dimensional space as in (Dong et al., 2017)

Meta-paths sampling. Firstly, we constructs a node sequence based on random walks of meta-paths. A meta-path $p = P_1 \xrightarrow{R_1} P_2 \xrightarrow{R_2} ... \xrightarrow{R_{k-1}} P_k$ is a path that is defined on a graph, where R_i signifies a composite relation between different node P. We define user-item-user (*U-I-U*) as our meta-path, where paths only exist if users has given the same ratings R_i to one item. We sample 32 rounds starting from each user and item with the sampled length $k = 64$.

Skip-gram model training. In the second step, through sampling meta-paths based on random walks as training corpus, we train a skip-gram model thus learn the vector representations $(\mathbf{W}_U, \mathbf{W}_I)$ for all users and items. The objective of the skip-gram model is to maximize the conditional probability $P(n_i|v)$ for the node $v \in V$ of its neighboring node $n_i \in N_v$:

$$\arg\max_{\theta} \sum_{v \in V} \sum_{t \in U,I} \sum_{n_i \in N_v} \log P(n_i|v) \qquad (9)$$

and the $P(n_i|v)$ is calculated as:

$$P(n_i|v) = \frac{\exp(w_{n_i}^T w_v)}{\sum_{j \in V} \exp(w_j^T w_v)} \qquad (10)$$

where w means the representation of one user or item. In experiments, we set number of negative sampling to 5, and train the skip-gram model for 10 epoch with learning rate of 0.001.

A.2. How to generate OOV tokens

This section complements subsection 5.2, where we generate OOV tokens from users and items representations for constructing META ID. Essentially, we need to build a hierarchical classification system for IDs in order to express a wider range of items and users with as few OOV tokens as possible, so that similar items and users are under the same hierarchical branch.

This hierarchical construction mechanism is very reminiscent of clustering methods, as we apply in META ID, in which we use out-of-class indexes and in-class indexes as two levels of IDs. Though more sophisticated clustering method s for multi-levels structure can be applied, in experiments, we use the simple K-Means clustering, which is more suitable for large-scale data volume due to its simple and easy to optimize nature. We also demonstrate the effectiveness of this approach in Table 4.

In experiments, we cluster user and item representations together. We then create the between-cluster tokens $\langle CT_i \rangle$ for cluster i, and sort the in-cluster users / items based on the cosine distance to the cluster centroids to get in-cluster tokens $\langle y_i \rangle$. Finally, we add a prefix token $\langle Item \rangle$ or $\langle User \rangle$ to denote its type. For example, an item might be represented as "$\langle Item \rangle \langle CT_i \rangle \langle j \rangle$" labeled with three tokens ("$\langle Item \rangle$", "$\langle CT_i \rangle$", "$\langle j \rangle$"), where "$\langle Item \rangle$" denotes it as an item, "$\langle CT_i \rangle$" is its cluster token, and "$\langle y_i \rangle$" is its in-cluster token.

It is worth noting that a naive approach to generating user and item IDs is to assign an independent OOV token as ID (IID) that needs to be learned for each item and user. However, this is not applicable to modern recommender systems with a large number of items and users, as the training time may be too long if a large number of new tokens need to be created. We also show it in Table 8, where we use the meta-path-based embeddings with a linear projection layer for initialization, which shows that it is ineffective compared to META ID.

Table 6: Examples of prompts for recommendation tasks (Part 1).

	Sequential Recommendation	**Direct Recommendation**	**Rating Prediction**
Task Input:	Considering **user_2024** has interacted with items **item_1**, **item_2**. What is the next recommendation for the user?	What should we recommend for **user_2024**?	Which star rating will **user_2024** give to item **item_2**? (1 being the lowest and 5 being the highest).
Task Output:	**item_2024**	**item_2024**	5

Table 7: Examples of prompts for recommendation tasks (Part 2).

	Explanation	**Review**
Task Input:	According to the feature word quality, generate a 5-star explanation for **user_2** about **item_2**.	Write a short sentence to summarize the following product review from **user_2**: Absolutely great product. I bought this for ...
Task Output:	Absolutely great product!	Perfect!

Table 8: Comparison of independent ID (IID) with META ID on sequential recommendation task.

Methods	Sports				Beauty				Toys			
	HR@5	NDCG@5	HR@10	NDCG@10	HR@5	NDCG@5	HR@10	NDCG@10	HR@5	NDCG@5	HR@10	NDCG@10
IID	0.0114	0.0073	0.0208	0.0103	0.0302	0.0194	0.0494	0.0256	0.0146	0.0091	0.0217	0.0114
META ID	**0.0357**	**0.0256**	**0.0520**	**0.0308**	**0.0480**	**0.0336**	**0.0689**	**0.0403**	**0.0564**	**0.0391**	**0.0803**	**0.0468**

A.3. How to incorporate META ID with LLMs

As mentioned in subsection 3.1, we convert every recommendation tasks into question&answering template, in which we describe the recommendation tasks in natural language form, and replace user and item IDs with different dataset like a cloze test. The full templates for every tasks in this format is from (Geng et al., 2022), where we give some examples in Table 6 and Table 7.

Take rating prediction task as example. We might ask the LLM, "Which star rating will user $\langle \text{User} \rangle \langle CT_1 \rangle \langle 18 \rangle$ give item $\langle \text{Item} \rangle \langle CT_8 \rangle \langle 24 \rangle$?", and expect the LLM to answer "5".

We construct the fine-tuning and testing dataset for LLM in this unified way. Then LLM is able to acquire the generalized knowledge across different tasks, and even carve out user and item characteristics through those tokens constructing IDs to handle different recommendation tasks.

B. Details of memorization score (MS) and diversity score (DS)

This section complements subsection 4. The token embedding layer in LLMs transforms each input token into token embedding vectors. And we use ID representation to indicate that these token embedding vectors corresponding to an ID, *i.e.*, the representation of an item or user in LLMs.

The convergence of DS. DS is a metric designed to quantify the diversity of ID representations in LLMs. Given the high computational demand of calculating KL divergence for all embedding pairs in large datasets, DS employs a random sampling approach, thus we present a convergence analysis for DS in Figure 5. The stability of DS is evident across both datasets, demonstrating a trend towards convergence as the number of sampled pairs grows. This sampling strategy reducesthe computational complexity from $O(|I|^2 \cdot D)$ to $O(N \cdot D)$, where the $|I|$ means the size of items, D is the dimension of ID representation, which is equal to the LLM's token embedding dimension.

The approximate value of MS. The adjusted cosine similarity for items is given by:

$$\text{sim}(i, j) = \frac{\sum_{u \in U}(R_{u,i} - \bar{R}_u) \cdot (R_{u,j} - \bar{R}_u)}{\sqrt{\sum_{u \in U}(R_{u,i} - \bar{R}_u)^2 \cdot \sum_{u \in U}(R_{u,j} - \bar{R}_u)^2}} \tag{11}$$

where $R_{u,i}$ and $R_{u,j}$ denote user u's ratings for items i and j, respectively, while $\bar{R}_u$ is user u's average rating. To enhance computational efficiency, especially for large-scale datasets, we precalculate the rating deviation sums and squared sums for each item and user:

$$\text{sim}'(i, j) = \frac{\text{Dev}(i) \cdot \text{Dev}(j)}{\sqrt{\text{DevS}(i)} \cdot \sqrt{\text{DevS}(j)}} \tag{12}$$

where the rating deviation sums and squared sums for each item is:

$$\text{Dev}(i) = \sum_{u \in U_i}(R_{u,i} - \bar{R}_u), \ \text{DevS}(i) = \sum_{u \in U_i}(R_{u,i} - \bar{R}_u)^2 \tag{13}$$
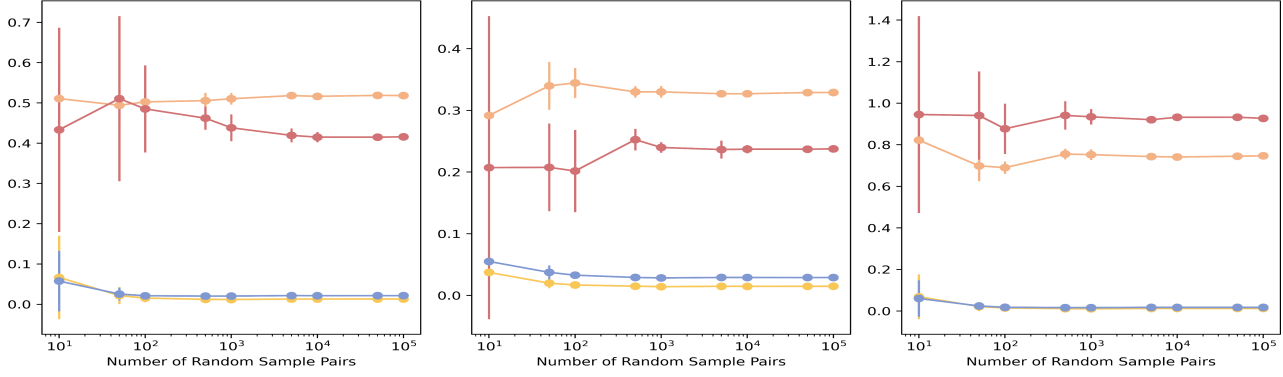
Figure 5: Convergence analysis of DS metric for Beauty and Sports datasets, with 95% confidence interval error bars indicating variability across 5 trials.

This approach reduces complexity from $O(|U| \cdot |I|^2)$ to $O(|I|^2)$, a significant improvement for large-scale datasets.

C. Experimental Setups and Implementation Details

Table 9: Basic statistics of the experimental datasets.

Dataset	Sports	Beauty	Toys
#Users	35,598	22,363	19,412
#Items	18,357	12,101	11,924
#Reviews	296,337	198,502	167,597
#Sparsity (%)	0.0453	0.0734	0.0724

C.1. Datasets Descriptions and Preprocessing

We conduct extensive experiments over three real-world datasets. The Amazon datasets are collected from *Amazon platform*[2] with user ratings and reviews on 29 categories of products. In this paper, we adopt three of them to evaluate our method, Sports&Outdoors, Beauty, Toys&Games. We follow (Geng et al., 2022) and use transaction records between January 1, 2019 to December 31, 2019. Detailed dataset statistics are available in Table 9.

We divide tasks into ratings, explanations, and reviews, adhering to the data-splitting approaches of similar studies (Geng et al., 2022; Li et al., 2020; 2021). For both sequential and direct recommendation tasks, we adopt the methodology of (Zhou et al., 2020; Geng et al., 2022; Ren et al., 2020), using the final item in a user's interaction sequence for testing while carefully structuring the training data to avoid leakage. For rating, explanation, and review task families, we randomly split each dataset into training (80%), validation (10%) and testing (10%) sets, and ensure

[2]https://nijianmo.github.io/amazon

that there is at least one instance included in the training set for each user and item.

C.2. Baselines

Our approach is compared to a variety of established models, spanning from CNN-based to LLM-based frameworks. Caser (Tang and Wang, 2018) applies CNNs to capture high-order Markov Chains in sequential recommendation. HGN (Ma et al., 2019) utilizes hierarchical gating networks for modeling long and short-term user interests. GRU4Rec (Hidasi et al., 2016) employs GRUs for session-based recommendation, representing items with embedding vectors. BERT4Rec (Sun et al., 2019), S^3-rec (Zhou et al., 2020) and SASRec (Kang and McAuley, 2018) employ self-attention mechanisms for sequential recommendation, focusing on bidirectional understanding and multi-head attention, respectively. FDSA (Hao et al., 2023) adopts feature-level self-attention for feature transitions. CL4SRec (Xie et al., 2022) introduces contrastive learning with data augmentation in sequential recommendation. P5 (Geng et al., 2022) learns different tasks with the same language modeling objective during pretraining, serving as the foundation model for various downstream recommendation tasks. P5 (Geng et al., 2022) is a recent method that uses a pretrained Large Language Model (LLM) to unify different recommendation tasks in a single model. Since there is no open source code for the recent work (Rajput et al., 2023) yet, we implemented the key ID construction of it ourselves in subsection 6.4.

In particular, we provide P5 with its variations (Hua et al., 2023), equipped with different ID constructs like Sequential IDs (SID), Random IDs (RID), and Collaborative IDs (CID) as a benchmark for exploring the impact of different ID strategies. RID Assigns each item with a random number as the item ID. The number is further tokenized into a sequence of sub-tokens, as did in P5. For example, an

Table 10: Table of the size of OOV tokens used by IDs.

OOV Tokens Size	Sports	Beauty	Toys
RID	0	0	0
SID	0	0	0
CID	448	437	487
IID	18357	12101	11924
META ID	1600	1319	727

item is randomly assigned the number "2024", and "2024" is represented as a sequence of tokens "20""24". SID is a straightforward method to leverage collaborative information for item indexing, where items interacted consecutively by a user are assigned consecutive numerical indices, reflecting their co-occurrence. CID approach employs spectral clustering based on spectral matrix factorization to generate item IDs. This method is based on the premise that items with more frequent co-occurrence are more similar and should share more overlapping tokens in ID construction. The results for all baselines except P5 with its variations are reproduced through open source code (Zhou et al., 2020).

C.3. Implementation Details

As mentioned in Appendix A, we generate the META ID for users and items for each dataset, generalized to all experiments below. For constructing META IDs, we sampling rating-based meta-paths in each dataset, where adjacent users assigned equal ratings to an item. We set the sampling path length to 64, and use a skip-gram model for training, with a window size of 5 and learning rate set at $1e^{-3}$. The embedding clusters groups are limited to $|G| = 100$ (200 for Toys) to manage vocabulary size effectively. The OOV tokens size is shown in Table 10.

C.3.1. EVALUATION ON SEQUENTIAL AND DIRECT RECOMMENDATION TASKS

We first evaluate META ID on sequential recommendation tasks and direct recommendation tasks following (Hua et al., 2023). Our implementation first utilizes T5 (Raffel et al., 2020b) as the backbone with parameters around 60.75 million. As mentioned in subsection 5.3, we add a linear layer where the OOV tokens undergo an extra linear transformation before the token embedding layer for a better initialization with $\alpha = 0.1$. We also consider decoder-only architecture LLaMA2-7b (Touvron et al., 2023) with 7B parameters. For tokenization, we use the default SentencePiece tokenizer with extended OOV tokens for parsing sub-word units. We use the same sequential recommendation and direct recommendation prompts in P5 (Geng et al., 2022) to convert sequential information into texts.

For LLM fine-tuning, we pre-train T5 for 10 epochs using AdamW optimizer on two NVIDIA RTX 3090 GPUs with

a batch size of 64, a peak learning rate of $1e^{-3}$. We apply warm-up for the first 5% of all training steps to adjust the learning rate, a maximum input token length of 1024. We use the lora (Hu et al., 2022) technique to fine-tune the token embedding layer and linear head layer of LLaMA2-7b for 1 epochs using AdamW optimizer on two NVIDIA RTX A6000 GPUs with a batch size of 28, a peak learning rate of $1e^{-5}$, the lora attention dimension of 16 and the alpha parameter of 32.

For LLM inferencing, beam search is utilized to generate a list of potential next items, evaluated under the all-item setting. To prevent the generation of non-existent IDs, we apply a constrained decoding method (Hua et al., 2023), setting the generation probability of invalid IDs to zero.

C.3.2. EVALUATION ON RATING, EXPLANATION AND REVIEW TASKS

To validate META ID's adaptability, we extend our evaluation to rating prediction, explanation generation, and review tasks, akin to P5. We use the same prompts in P5 (Geng et al., 2022) to convert all information into training texts.

For LLM fine-tuning, we pre-train T5 for 10 epochs using AdamW optimizer on two NVIDIA RTX 3090 GPUs with a batch size of 32, a peak learning rate of $1e^{-3}$. We apply warm-up for the first 5% of all training steps to adjust the learning rate, a maximum input token length of 512 and maximum generation length of 64.

For LLM inferencing, greedy decoding is applied for rating prediction, explanation generation, and review tasks.

The full results are shown in Table 11.

D. Additional Experimental Results

Initialization Approaches. We explore the impact of different token initialization methods on the performance of META ID. Recognizing that the LLM's vocabulary includes numeric tokens for linguistic IDs, we first consider whether reinitializing these numeric tokens helps LLM for recommendation. As shown in Figure 6, our experiment, contrasting random initializing numeric tokens (Random Init.) with keeping T5's original token embeddings (Embedding Init.), reveals that random initialization does not enhance performance, and be detrimental on Sports and Toys datasets. This suggests that the influence of pre-training on these tokens cannot be effectively negated through simple random initialization, thus not suitable for building IDs. This result emphasizes the importance of introducing extra tokens for IDs in LLM-based recommender systems. We also compare two initialization approaches for META ID: random initialization (Random Init.) and initializing OOV tokens using the augmentation (Embedding Init.) (See Section 5.2). Our

Table 11: Performance comparison of different methods on rating, explanation, and review task.

| Task Type	Metric	Sports				Beauty				Toys			
		RID	SID	CID	**META ID**	RID	SID	CID	**META ID**	RID	SID	CID	**META ID**
Rating	RMSE	1.0382	1.0486	1.0383	**1.0327**	1.2829	1.3098	1.2819	**1.2818**	1.0725	**1.0693**	1.0766	1.0770
Explan.	BLEU-1	16.2567	16.5825	16.6121	**16.9005**	18.2299	18.3981	19.3499	**19.5106**	19.9858	20.2198	**20.4570**	20.2270
	BLEU-4	2.1782	2.1944	2.2332	**2.3481**	2.9027	2.8071	**3.0626**	3.0592	4.3495	4.3701	**4.5844**	4.4945
Review	BLEU-1	7.6140	7.7948	7.6586	**7.8819**	6.2282	6.5055	6.5854	**7.0500**	**8.5336**	8.0862	7.3846	8.3080
	BLEU-4	2.3228	1.2406	2.4109	**2.6546**	1.9891	1.2406	1.9718	**2.7485**	1.1315	**1.8366**	1.2128	1.7061

(a) Sports (b) Beauty (c) Toys
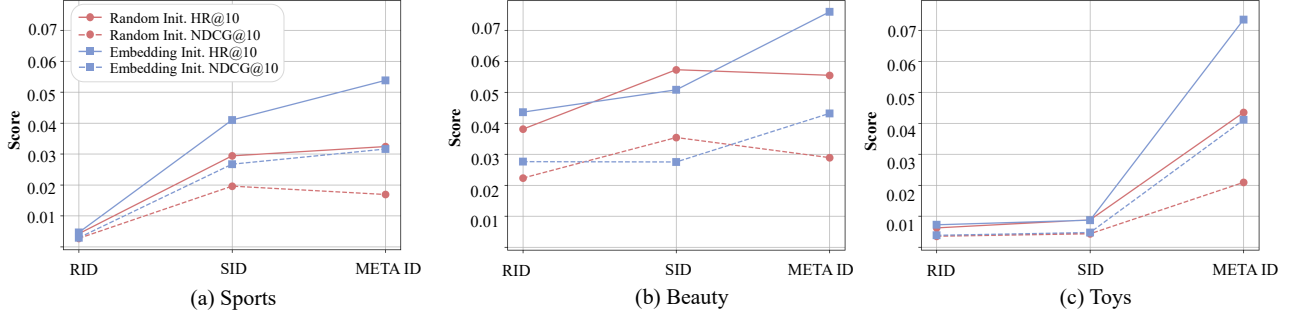
Figure 6: Performance comparision between random initialization (Random Init.) and initializing ID token embeddings using pre-trained ones (Embedding Init.). For RID and SID, random initialization does not have a positive effect compared to keeping T5's original token embeddings. For META ID, initializing using a learned strategy with the help of cluster centroid embeddings (Embedding Init.) learned from meta-paths can significantly enhance the recommendation performance of META ID compared to random initialization.
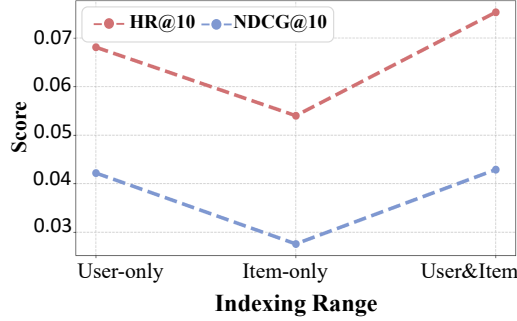

Figure 7: Performance comparison of whether indexing both items and users on Beauty dataset.

findings show that the latter method substantially improves META ID's performance, underlining the critical nature of the token initialization method in achieving better results.

User or Item Indexing. Previous ID strategies for LLMs only consider indexing for items, which come from the convention that users are typically represented by a sequence of interacted items in sequential recommendation (Geng et al., 2022; Hua et al., 2023; Rajput et al., 2023). While META ID models users and items, as shown in Table 7, reveals that the combined user-item indexing (User&Item) outperforms either user-only or item-only indexing. This result

shows the importance of incorporating user preferences and item attributes for LLMs to enhance the accuracy of the recommendations.

Visualization of ID-related tokens. Directly applying in-vocabulary tokens to construct IDs (RID and SID) brings poor performance in Toys dataset. In Figure 2b, We use t-SNE visualization to map ID token embeddings and observed that these tokens tend to be homogeneous, whereas CID and META IDs that use OOV tokens to construct IDs have a wider distribution, reflecting difference and diversity between their representations.

To further illustrate the impact on representation, we visualized the attention mechanism in sequence recommendation generation in Figure 8. This revealed that SID leads to uniform attention patterns, not distinguishing between different items and user IDs. In contrast, META ID demonstrates distinct attention patterns, successfully differentiating items and emphasizing user IDs, thereby allowing models to grasp more personalized and distinct information.

Statistics on training & Inference Time. We provide statistics on the training and inference time of P5 models, we collect the running time on the Toys dataset. As mentioned in subsection C, we trained and test our models on two RTX 3090 GPUs. For training on sequential recommendation and direct recommendation tasks, the T5 model spent 3.5 hours
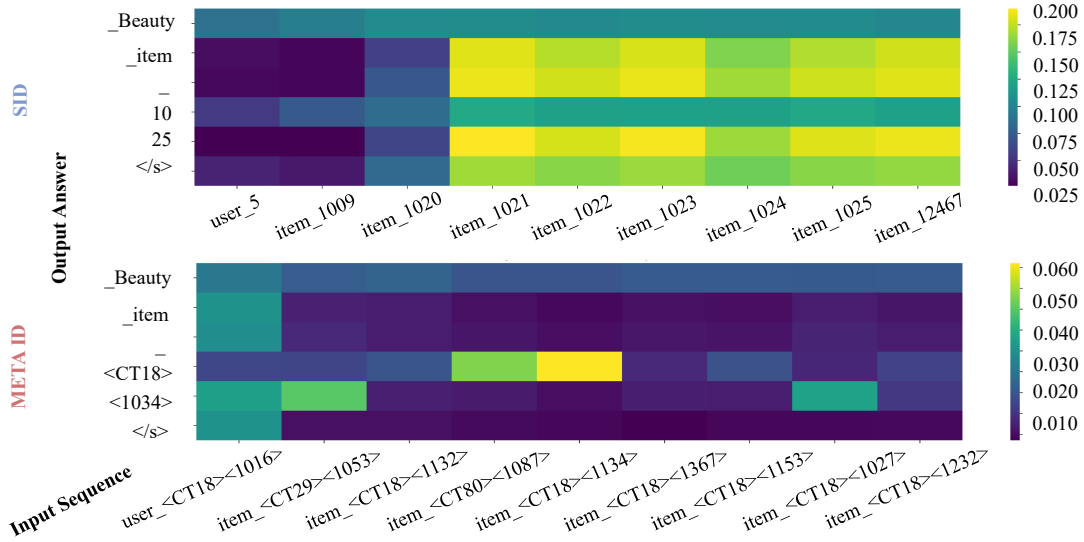
Figure 8: Visualization of attention between input and output tokens in sequence recommendation generation, where we aggregate tokens within one ID for clarity. The visualization indicates that SID leads to undifferentiated attention across items and neglects the user ID, whereas META ID distinctly allocates attention to sequence items, outputting item containing the " ⟨CT18⟩" token in both the user ID and item ID with the highest attention values.

Table 12: The standard error of the metrics for META ID(T) on sequential recommendation task.

Datasets	HR@5	NCDG@5	HR@10	NCDG@10
Sports	0.0322 ± 0.00061	0.0223 ± 0.00060	0.0487 ± 0.00073	0.0277 ± 0.00055
Beauty	0.0510 ± 0.00038	0.0351 ± 0.00044	0.0753 ± 0.00131	0.0429 ± 0.00075
Toys	0.0533 ± 0.00091	0.0372 ± 0.00067	0.0761 ± 0.00138	0.0441 ± 0.00078

Table 13: Average inference time (in milliseconds) of T5 with META ID on different tasks on the Toy dataset.

Models	per user		per user-item pair		per review	
	Sequential	Direct	Rating	Explanation	Summarization	Preference
META ID (T)	74.05	68.60	5.21	17.28	9.67	8.55

to finish training. The average inference time of T5 model on dferent tasks are presented in Table 13. Sequential and direct recommendation tasks require much longer inference time than other tasks due to the beam search step. Overall, the inference is very fast. It is also promising to further reduce the training and inference time with the help of effcient Transformer techniques.

E. Discussions

There are two promising directions of META ID. First, META ID uses a fixed database of users and items, while newly appearing items and users do not have interaction history. This could be solved using methods related to the cold start issue. Second, META ID applies two-level tokens for constructing IDs, while a more complicated hierarchical structure could be considered. Then, META ID could be applied to modern recommender system containing trillions of users and items.