Information Processing and Management 62 (2025) 104122

Contents lists available at ScienceDirect

Information Processing and Management

journal homepage: www.elsevier.com/locate/ipm

Efficient and explainable sequential recommendation with language model

Zihao Li, Lixin Zou, Chao Ma, Chenliang Li *

Key Laboratory of Aerospace Information Security and Trusted Computing, School of Cyber Science and Engineering, Wuhan University, Wuhan, 430072, Hubei, China

ARTICLE INFO

Keywords:
Sequential recommendation
Explainable recommendation
Parameter-efficient fine-tuning

ABSTRACT

Motivated by the outstanding success of large language models (LLMs) in a broad spectrum of NLP tasks, applying them for explainable recommendation become a cutting-edge recently. However, due to the inherent inconsistency in the information and knowledge focused, most existing solutions treat item recommendation and explanation generation as two distinct processes, incurring extensive computational costs and memory footprint. Besides, these solutions often pay more attention to the item-side (*i.e.,* item attributes and descriptions) for explanation generation while ignoring the user personalized preference. To close this gap, in this paper, we propose a personalized explainable sequential recommendation model, which aims to output the recommendation results as well as the corresponding personalized explanations via a single inference step. Moreover, to mitigate the substantial computational cost, we devise a rescaling adapter and a Fast Fourier Transform (FFT) adapter for parameter-efficient fine-tuning (PEFT). Theoretical underpinnings and experimental results demonstrate that compared with prevalent PEFT solutions, our adapter possesses three merits: (1) a larger receptive field across the entire sequence for long-term dependency modeling; (2) element product in orthogonal bases for noise attenuation and signal amplifying; (3) better alignment and uniformity properties for precise recommendation. Comprehensive experiments on three public datasets against nine sequential recommendation solutions and three explanation generation solutions illustrate our PLEASER outperforms the strong baselines significantly with only 5% parameter fine-tuning. Code available at https://github.com/WHUIR/PLEASER.

1. Introduction

Encouraged by the great performance of the large language models (LLMs) in natural language processing (NLP), modeling text information with language models for sequential recommendation and explanation attracts widespread attention from industry and academia. Reviewing the relevant works, existing efforts can be categorized into four lines: (1) *pre-training and fine-tuning*, which pre-training language models on a source domain, followed by fine-tuning on the target domain for recommendation (Ding, Ma, Deoras, Wang, & Wang, 2021; Li, Wang, et al., 2023; Zhou et al., 2020); (2) *In-context learning (ICL) or prompting*, which induces language models to predict the user's interested items by providing prompted samples or instruction instances, without pre-training needed (Hou et al., 2023; Zhang et al., 2021); (3) *Full parameter fine-tuning*, which fine-tuning entire language models in recommendation datasets straightforwardly to align the recommendation task (Mao, Wang, Du, & Wong, 2023; Wang et al., 2022); and (4) *Parameter efficient fine-tuning (PEFT)*, which only fine-tune a subset of parameters via reparameterized fine-tuning (Hu et al.,

* Corresponding author.
E-mail address: cllee@whu.edu.cn (C. Li).

https://doi.org/10.1016/j.ipm.2025.104122
Received 6 November 2024; Received in revised form 7 February 2025; Accepted 3 March 2025
Available online 26 March 2025

Table 1
The characteristic of different solutions with language model for sequential recommendation. In contrast to existing attempts, our solution can achieve personalized explanation recommendation (*i.e.,* output the recommendations and the correspondent explanations aligned with user tastes and item features simultaneously) with low resource consumption.

Methods	Performance retention	One-stage	Low cost	Personalized explanation
Pre-training and Fine-tuning	✓	✗	✗	✗
Full-parameter Fine-tuning	✓	✓	✗	✗
ICL or Prompting	✗	✓	✓	✗
Parameter-efficient Fine-tuning	✓	✓	✓	✗
Ours	✓	✓	✓	✓

2021), additive fine-tuning (Liu, Ji, et al., 2022) or adapters (Hu et al., 2023) for efficient fine-tuning (Bao et al., 2023; Cui, Ma, Zhou, Zhou, & Yang, 2022).

Despite the substantial benefits of LLMs in recommendation, these efforts may have some drawbacks, which preclude them from achieving optimal results. The characteristics of the four types of sequential recommendation recipes are summarized in Table 1. Specifically, most pre-training and fine-tuning methods require distinct optimization objections for each stage, rendering the overall training process somewhat complicated and redundant. Although in-context learning or prompting methods can output recommendation results without additional training, their performance generally remains inferior to the full parameter fine-tuning. Besides, the prompts design is an experience-driven and heuristic process, where the final results are sensitive to the wording of prompts (Webson & Pavlick, 2021; Zhao, Wallace, Feng, Klein, & Singh, 2021). In contrast, full parameter fine-tuning can acquire satisfactory results, but the massive storage and computation costs prevented its widespread application. To address this problem, PEFT, which updates only a small group of parameters without significant performance degradation, has emerged as a prominent solution for task-specific fine-tuning. However, the capabilities of whole sequence modeling and noise filtering are insufficient for existing adapter-based efficient parameter fine-tuning solutions. Due to the decoder-only model paradigm, existing LLM-driven methods can simultaneously generate the predicted items and their corresponding explanations. Despite this, the recommendation aims to capture the collaborative signal for user preference prediction, while the explanation relies more on the item characteristics and descriptions. The inherent inconsistency in optimization between these two tasks creates challenges in integrating them into a holistic framework. Besides, it is intractable to explicitly incorporate the predicted target item description as input during explanation generation within a one-step inference. Therefore, all the above methods model the explanation generation and recommendation as two distinct processes that require the model to be conducted twice to yield corresponding outputs, incurring redundant intermediate results and unessential computing costs. Furthermore, existing works concentrate exclusively on item features for explanation generation, neglecting the user's personality encapsulated in the historical interaction records, thus failing to deliver user-oriented personalized explanation recommendations.

To close this gap, we propose PLEASER for **p**erson**a**l**i**zed **e**xpl**a**nation **se**quential **r**ecommendation. As illustrated in Fig. 1, unlike conventional sequential recommendation, PLEASER can predict the next interested items based on her interacted records and also generate explanations consistent with both the user's personalized preference and the candidate item characteristics in a single go. Specifically, we choose T5, an encoder–decoder architecture, as the backbone. Given a user's historical interacted items and the correspondent text information (*i.e.,* title), the encoder is dedicated to capturing the long-term dependency and collaborate signal in the sequence for recommendation. Besides, the decoder will consider both the description of the predicted item yield by the encoder and the user's preference representation derived from the encoder module to align with user reviews as personalized explanations. Consequently, in the inference stage, the encoder–decoder framework allows for the generation of recommendations and explanations in a single step. Moreover, we devise two tailored adapters (*i.e.,* a rescaling adapter and a Fast Fourier Transform (FFT) adapter) for parameter-efficient fine-tuning. Theoretical underpinnings substantiate that compared with the prevalent adapters, the FFT adapter can be recognized as an alternative to the circular convolution but with a larger receptive field over the full sequence for long-term dependency modeling. Besides the FFT adapter is constructed by a series of orthogonal bases derived from the frequency domain, which will serve as filters to attenuate the noise and amplify the significant signals via learnable element-wise multiplication operations for representation augmentation. Experimental results illustrate that the item representation optimized by our method is superior in the properties of *alignment*[1] and *uniformity*[2] Wang and Isola (2020). Compared with existing LLMs-based recommendation solutions (Ding et al., 2021; Hou et al., 2022; Li, Wang, et al., 2023), our PLEASER displays an impressive zero-shot learning ability and achieves best results with only 5% parameters fine-tuning.

To encapsulate, the main contributions of this paper include the following:

- We proposed a unified framework, different from existing works, our solution can generate the recommendations as well as the corresponding personalized explanations via only one conduction.
- We devise a rescaling adapter and a Fast Fourier Transformer adapter for PEFT, which acquires superior performance by only 5% parameters updating. Theoretical analysis and experimental results further elaborate on the effectiveness of the FFT adapter from three perspectives.

[1] The representation of similar samples should be close to each other in the hypersphere. Conversely, the irrelevant samples should be far apart.
[2] Representation of samples should be uniformly distributed on the unit hypersphere.

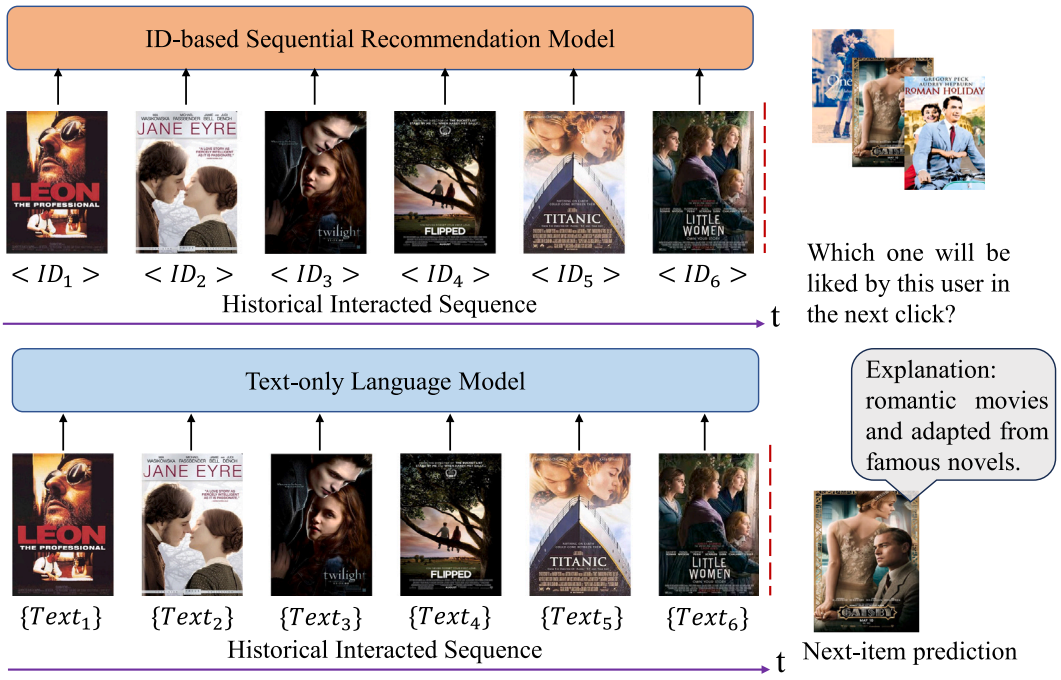

Fig. 1. A toy example to illustrate the comparison between item ID-based approaches (up) and text-based, language model-driven recommendation methods (below). Unlike the ID-based solution, the explainable sequential recommendation aims to yield the recommended items as well as the corresponding personalized explanations based on the item text information, such as titles and descriptions..

- Comprehensive experiments on three public datasets verify that our PLEASER is superior to nine competitive baselines on sequential recommendation and three baselines on explanation generation. The empirical analysis also elucidates the effectiveness of each module design and the remarkable ability of zero-short learning.

The remainder of this paper is organized as follows: Section 2 provides a review of related works, including ID-base sequential recommendation, language model-based sequential recommendation, and explainable recommendation, within this research domain. Section 3 introduces the notations and problem statement as preliminary knowledge for readability first. Then, an overview of our proposed PLEASER is presented in Section 3.3, which includes sequential recommendation and personalized explanation generation. Sections 3.4 to 3.6 introduce each module in detail, respectively. Section 3.7 analyzes the merits of our proposed adapter against existing solutions from mathematical perspectives. Section 4 reports the experimental results along with a comprehensive analysis. Finally, Section 5 offers the conclusion of this work.

2. Related work

We first present a concise review of the conventional ID-based models for recommendation. We then summarize the related studies for sequential recommendation and explainable recommendation with language models, the three areas most closely associated with our research.

2.1. ID-based sequential recommendation

By modeling the transition probability matrix, Markov chain endeavors to imitate the user's ordered clicking behavior as a Markov decision process for sequential recommendation (Chen, Fan, & Wu, 2023; Shani, Heckerman, Brafman, & Boutilier, 2005) in the early stage. Encouraged by the incredible capacity of feature representation, deep learning methods, *e.g.,* CNN, LSTM, GRU, and the variants (Hidasi, Karatzoglou, Baltrunas, & Tikk, 2016; Tang & Wang, 2018; Xiao, Liang, & Meng, 2019), are carried out successively and deliver remarkable results in the sequential recommendation. Besides, to exploit the implicit correlation between any two items and model the short-term and long-term dependency relationships, attention mechanisms become mainstream in this task, *e.g.,* SASRec (Kang & McAuley, 2018) and BERT4Rec (Sun et al., 2019). Moreover, based on information propagation and aggregation, graph neural networks (GNNs) effectively capture high-order dependency encapsulated in sequences and are ubiquitous for next-item prediction (Chang et al., 2021; Li, Wang, Yang, et al., 2023; Li, Yang, et al., 2024; Liu et al., 2024; Wu et al., 2019). Apart from that, a variety of sophisticated methods, *e.g.,* memory network (Chen et al., 2018), contrastive learning (Duan, Zhu, Liang, Zhu, & Liu, 2023; Li, Sun, Zhao, et al., 2023; Yang et al., 2023), and reinforcement learning (Guo, Zhang, Chen, Wang, & Yin, 2022;

Tong, Wang, Li, Xia, & Niu, 2021), also emerge. As a new paradigm of generation task, the diffusion model achieves promising results in CV (Rombach, Blattmann, Lorenz, Esser, & Ommer, 2022) and NLP (Li, Thickstun, Gulrajani, Liang, & Hashimoto, 2022), thereby, it has become a hotspot in recommendation (Li, Sun, & Li, 2023; Wang et al., 2023). Despite the effectiveness of the aforementioned approaches, all of them use item ID for recommendation, neglecting the enriched external knowledge encapsulated in the item text for auxiliary information fusion.

2.2. Sequential recommendation with language models

Owing to the ground-breaking of LLMs in many NLP applications (Devlin, Chang, Lee, & Toutanova, 2019; Li, Xu, et al., 2024; Raffel et al., 2020; Touvron et al., 2023), applying language models for item text modeling has become a prominent solution in recommendation recently. UnisRec (Hou et al., 2022) is a pioneer work that first pre-trains a BERT-based language model by item text information (*e.g.,* titles, categories, and brands) on source domains. Subsequently, fine-tune the model using target domain data for recommendation. Similarly, Recformer (Li, Wang, et al., 2023) applies Longformer (Beltagy, Peters, & Cohan, 2020) as the backbone and uses item attributes as text information for model pre-training and fine-tuning. To comprehensively verify the zero-shot learning capacity of LLMs in recommendation, Wang and Lim (2023) designed different prompting strategies to investigate the performance of GPT-3 (Kojima, Gu, Reid, Matsuo, & Iwasawa, 2022) in the sequential recommendation. In line with this research, Hou et al. (2023) et al. devise various prompting templates and define the sequential recommendation as a conditional ranking task. Analyzing the zero-shot learning capability of GPT-3.5 (Ouyang et al., 2022). Although prompt engineering allows generating the recommendations without any parameter updating, they often encounter the challenges of performance degeneration. Conversely, P5 (Geng, Liu, Fu, Ge, & Zhang, 2022) and UniTRec (Mao et al., 2023) use T5 (Raffel et al., 2020) and BART (Lewis et al., 2020) as base models respectively and fine-tuning the full model for the accurate recommendation, the extensive computation cost hinders the widespread application of these methods in practical scenarios. Consequently, parameter-efficient fine-tuning (PEFT) becomes ubiquitous in LLMs-based sequential recommendation. For instance, TALLRec (Bao et al., 2023) selects LLaMA (Touvron et al., 2023) as the backbone and applies LoRA (Hu et al., 2021) for recommendation. M6-Rec (Cui et al., 2022) uses a visual-linguistic pre-trained model, *i.e.,* M6 (Lin et al., 2021), as the base model and proposes an improved prompt tuning, for task-specific parameter fine-tuning. ReLLa (Lin et al., 2023) chooses Vicuna (Chiang et al., 2023) and TALLRec (Bao et al., 2023) as base models and carries out a retrieval-enhanced instruction tuning for PEFT and sequential recommendation.

2.3. Explainable recommendation with natural language generation

Explainable recommendation (Zhang, Chen, et al., 2020) aims to provide the recommended items with convincing reasons for a more transparent and trustworthy system. Summarizing the existing works, the explanation formats can be categorized into various styles, *e.g.,* knowledge graph based (Fu et al., 2020), item features (He, Chen, Kan, & Chen, 2015), natural language description (Geng et al., 2022; Li, Zhang, & Chen, 2021, 2023), and image visualizations (Chen et al., 2019). In this paper, we specialize in explainable recommendation with natural language generation (Chen, Chen, Shi, & Zhang, 2021; Geng et al., 2022; Li et al., 2021; Li, Zhang, & Chen, 2023; Liu, Liu, Lv, Zhou, & Zhang, 2023; Yang, Wang, Deng, & Wang, 2021). In the early stages, given the circumscribe of model size and computational resources, the generation quality and input length are quite limited. Although LLMs can handle longer sentences, they emphasize more on the item features, ignoring the user's preference and personality encapsulated in the historical sequences and reviews, which are equally important for explanation generation. In addition, neither of them can generate the recommendation results and explanations simultaneously with one-step inference.

3. Methodology

We briefly introduce the formalization of personalized explainable sequential recommendation and our overall solution. Afterward, each module will be detailed respectively.

3.1. Notations

To enhance readability, we first provide a summary of the key notations employed in this paper in Table 2

3.2. Problem statement

Sequential Recommendation. Denote the item set and user set as $i \in \mathcal{I}$ and $u \in \mathcal{U}$, where each item i contains both an ID (i_{ID}) and the correspondent text information (i_{text}) (*e.g.,* title, description, review). Given a user u's historical sequence and the correspondent text information of each item, which can be organized chronologically as a sequence, *i.e.,* $s_u = [i_1, i_2, \dots, i_\ell]$, personalized explainable sequential recommendation aims to yield the probability of item $i_{\ell+1}$ the user will prefer at the $\ell + 1$ interacted time and also generate a reasonable explanation simultaneously, *i.e.,* $\left(P(i_{\ell+1}), E_i\right) = f(i_{\ell+1}, E_{\ell+1}|[i_{text_1}, \dots, i_{text_\ell}])$, where $E_{\ell+1}$ is the explanation of the predicted item $i_{\ell+1}$.[3]

[3] In our paper, $E_{\ell+1}$ is optimized to be consistent with the review of user u to the item $i_{\ell+1}$.

Table 2
Major notation.

Notation	Description	Notation	Description
i	Item	u	User
$\mathcal{I}$	Item set	$\mathcal{U}$	User set
i_{ID}	Item ID	i_{text}	Item text information
w_m	mth token	t_i	The title of item i
d_i	The description of item i	E_i	The personalized explanation of item i
$P(i_{\ell+1})$	The probability that the user u will like item i at $\ell+1$ step	y_i	Predict score of item i
s_u	User u's historical sequence organized chronologically	$f(\cdot)$	Sequence model
$\mathcal{F}(\cdot)$	Fast Fourier Transform	$\mathcal{F}(\cdot)^{-1}$	Inverse Fast Fourier Transform
$\mathbf{r}_i$	Representation of item i	$\mathbf{r}_s$	Representation of sequence s
$\mathbf{h}_m, \mathbf{h}'_m$	Hidden state of token m	$\mathbf{h}^f_m$	The fused hidden state of token m
$\mathbf{w}$	Learnable parameters	$\mathbf{X}$	Latent matrices

3.3. Overview

The framework of our proposed method is depicted in Fig. 2. Overall, it is constructed in two parts split by the dashed line: (1) the architecture of the adapter encoder–decoder backbone (subfigures (a) and (b)), which is a transformer-based model with two adapters (*i.e.,* a rescaling adapter integrated with attention module and an FFT adapter integrated with FFN module). (2) the pipeline of sequential recommendation (subfigure (c)) and explanation generation (subfigure (d)). Specifically, we encode the item title text information for the item and sequential representation generation and recommendation. After that, the description of the predicted item is obtained, thus, it is fed into the decoder module for explanation generation.

Sequential Recommendation. In PLEASER, we adopt the T5 (Raffel et al., 2020) encoder as the backbone and concate each item title along the timeline as the input, which will be fed into the encoder for next-item prediction. It can be formalized as below:

$$\begin{aligned} t_i &= [[CLS], w_1, \dots, w_m] \\ \mathbf{h}_i &\leftarrow \text{Adapter Encoder}(t_i) \\ \mathbf{h}_\ell &\leftarrow \text{Adapter Encoder}([t_1, \dots, t_\ell]) \\ \mathbf{r}_i &= \mathbf{h}_i,\ \mathbf{r}_s = \mathbf{h}_\ell \\ y_i &= \mathbf{r}_i \cdot \mathbf{r}_s^T \end{aligned} \tag{1}$$

where t_i is the title of item i, $w_1, \dots, w_m$ are the contained tokens. The token $[CLS]$ serves as a special indicator denoting the beginning of a sentence. $\mathbf{r}_i$ and $\mathbf{r}_s$ are the representation of candidate item i and historical sequence s, derived from the hidden state of the last tokens regarding the candidate item title and the concated title sequence respectively, *i.e.,* $\mathbf{r}_i = \mathbf{h}_i, \mathbf{r}_s = \mathbf{h}_\ell$. Following previous research (Sun et al., 2019), we calculate the inner product between $\mathbf{r}_i$ and $\mathbf{r}_s$ to obtain the prediction score for candidate item i.

Personalized Explanation Generation. We devise a personalized preference extraction (PPE) module to integrate both the characteristics of predicted item i encapsulated in the description d_i and the user's implicit preference extracted from the historical sequence representation $\mathbf{r}_s$. The fused representation is further fed into a decoder as a condition for personalized explanation generation.

$$\begin{aligned} d_i &= [[CLS], w_1, \dots, w_n] \\ [\mathbf{h}'_0, \mathbf{h}'_1, \dots, \mathbf{h}'_n] &\leftarrow \text{Adapter Encoder}(d_i) \\ [\mathbf{h}^f_0, \mathbf{h}^f_1, \dots, \mathbf{h}^f_n] &\leftarrow \text{PPE}(\mathbf{r}_s; [\mathbf{h}'_0, \mathbf{h}'_1, \dots, \mathbf{h}'_n]) \\ E_i &= \text{Adapter Decoder}([\mathbf{h}^f_0, \mathbf{h}^f_1, \dots, \mathbf{h}^f_n]) \end{aligned} \tag{2}$$

where $[\mathbf{h}'_0, \mathbf{h}'_1, \dots, \mathbf{h}'_n]$ are the hidden states derived from each token of item i's description. $[\mathbf{h}^f_0, \mathbf{h}^f_1, \dots, \mathbf{h}^f_n]$ are the hidden state representation fused with user preference. E_i is the personalized explanation.

3.4. Adaptation

Attribute to the efficiency, adapter (Hu et al., 2021; Liu, Tam, et al., 2022) is a popular fine-tuning strategy in LLMs, we, thereby, design two different adapters for PEFT. To be specific, we add an element-wise multiplication operation (*i.e.,* rescaling) in the self-attention layer and a Fast Fourier Transform (FFT) operation in the feed-forward neural network respectively as an extension of the vanilla transformer (Vaswani et al., 2017).

Rescaling Adapter. We apply an element-wise multiplication operation as the adaptation of the self-attention layer. Specifically, we implement the rescaling adaptation on the keys and values (*i.e.,* $\mathbf{w}_k$ and $\mathbf{w}_v$ vectors for key and value), which can be formalized

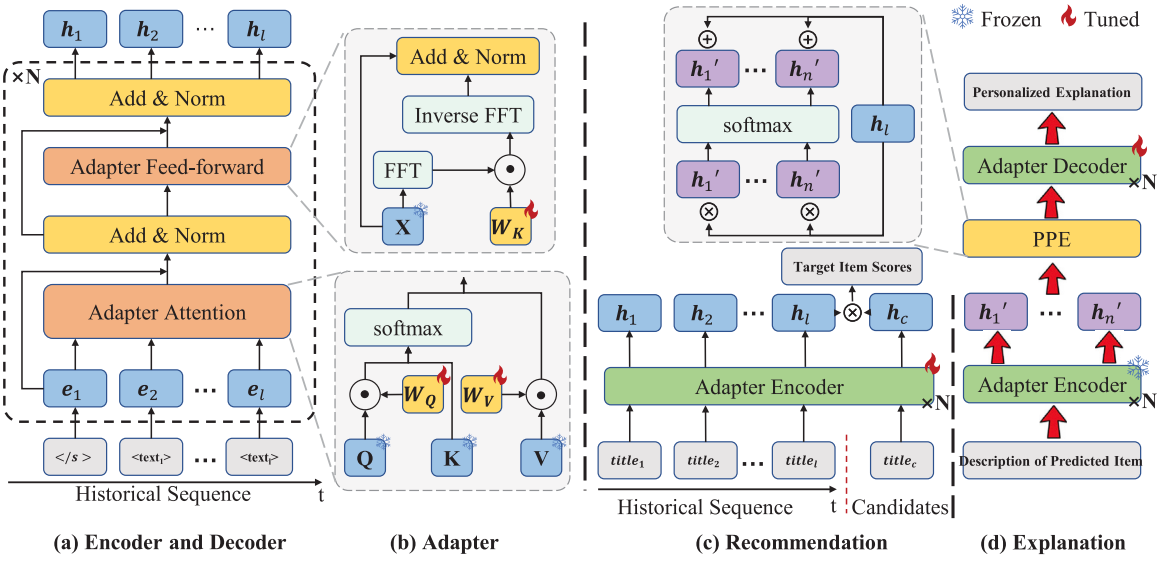

Fig. 2. Framework of PLEASER. In general, it can be split into two parts by the dashed line. The left part subfigures (a) and (b) present the overall architecture of the adapter encoder and adapter decoder. The right part depicts the pipeline of sequential recommendation (subfigure (c)) and explanation generation (subfigure (d)). Specifically, each item title text information is obtained and contacted as a sequence which will be fed into the encoder for sequential representation generation and recommendation. For explanation generation, we first acquire the predicted target item yield by the encoder, then obtain its description, which will be fed into the decoder for explanation generation..

as below:

$$\text{softmax}\left(\frac{\mathbf{Q}(\mathbf{w}_k \odot \mathbf{K})^T}{\sqrt{d_k}}\right)(\mathbf{w}_v \odot \mathbf{V}) \tag{3}$$

where $\mathbf{Q}, \mathbf{K}, \mathbf{V} \in \mathbb{R}^{\ell \times d}$ is the hidden states of query, key, and value. ℓ is the sequence the length, $\mathbf{w} \in \mathbb{R}^{1 \times d}$ is a learnable vector.

Fast Fourier Transform Adapter. FFT is extensively employed across many avenues (Lee-Thorp, Ainslie, Eckstein, & Ontanon, 2021; Rao, Zhao, Zhu, Lu, & Zhou, 2021; Zhou, Yu, Zhao, & Wen, 2022). Theoretically, the FFT adaptation is equivalent to the circular convolution in the time domain, while attaining a large receptive field covering the entire sequence. Thus, it facilitates the model to capture the long-term dependency between items. Additionally, applying the element-wise multiplication in the frequency domain could be construed as a filter for noise attenuation and valuable signal amplification (ref. Section 3.7 for details). Therefore, an FFT adaptation is incorporated into the feed-forward layer of the Transformer block. More concretely, we first transform the input representation into the frequency domain via FFT, also an element-wise multiplication is applied to the inputs. After that, it will undergo an inverse FFT procedure to recover the inputs to the time domain. This can be formulated as follows:

$$\begin{aligned}\tilde{\mathbf{X}} &= \mathcal{F}^{-1}\left(\mathbf{W}_f \odot \mathcal{F}(\mathbf{X})\right) \\ \tilde{\mathbf{X}} &= \text{LayerNorm}\left(\mathbf{X} + \text{Dropout}\left(\tilde{\mathbf{X}}\right)\right)\end{aligned} \tag{4}$$

where $\mathcal{F}(\cdot)$ and $\mathcal{F}^{-1}(\cdot)$ represent the 1D FFT and 1D inverse FFT respectively, which enable the conversion between the frequency domain and time domain. $\mathbf{W}_f \in C^{\ell \times d}$ is a learnable adapter. The layer normalization, dropout, and residual connection operations remain the same as the vanilla Transformer for a stable convergence. The augmented $\tilde{\mathbf{X}}$ will be further fed into the standard feed-forward layer for representation updating.

3.5. Personalized Preference Extraction (PPE)

To integrate the user's personalized preference from the historical interacted sequence with the characteristics of the predicted items, we design a personalized preference extraction module (*i.e.,* PPE). More concretely, given an interacted historical sequence $s = [t_1, \ldots, t_\ell]$ consisting of the item title t and the description $d_i = [[CLS], w_1, \ldots, w_n]$ of predicted item i, we fed them into the adapter-fused recommendation encoder to yield the personalized preference representation and item characteristic representation. Then, the PPE module is adopted for the representation fusion, which can be formalized as below:

$$\begin{aligned}\mathbf{r}'_s &= \mathbf{W}_p \cdot \mathbf{r}_s \\ \alpha_j &= \frac{\mathbf{r}'_s \cdot \mathbf{h}'_j}{\sum_{j=0}^{n} \mathbf{r}'_s \cdot \mathbf{h}'_j} \\ \mathbf{h}^f_j &= \alpha_j \mathbf{h}'_j + \mathbf{r}'_s\end{aligned} \tag{5}$$

Algorithm 1: Adapter-fused Encoder fine-tuning

Input:
  Historical sequence: $s = [t_1, ..., t_\ell]$;
  Target item: $i_{\ell+1} = [[CLS], w_1, .., w_m]$;
  Training Epoch: T;
  Adapter-fused Encoder: $f_\theta^E(\cdot)$;
Output:
  Predicted Target Item: $i_{\ell+1}$;
Process:
for $t = 1, 2, .., T$ **do**
  $\mathbf{h}_\ell = f_\theta^E([t_1, ..., t_\ell])$; // Sequence Representation
  $\mathbf{h}_i = f_\theta^E(t_i)$; // Target Item Representation
  $\mathcal{L}_{rec} = \frac{1}{|\mathcal{U}|}\sum_{i\in\mathcal{U}} -\log P_i,\ P_i = \frac{\exp(\mathbf{h}_i \cdot \mathbf{h}_\ell^T)}{\sum_{i\in I}\exp(\mathbf{h}_i \cdot \mathbf{h}_\ell^T)}$; // Loss
  $\theta_{fE} \leftarrow \theta_{fE} - \epsilon \nabla_{\theta_{fE}} \mathcal{L}_{rec}$; // Encoder fine-tuning
end for

where $\mathbf{W}_p \in \mathbb{R}^{d\times d}$ is a linear projection transformation. $\mathbf{h}_j$ is the hidden states of each token in d_i. $\mathbf{r}_s$ is the representation of sequence s yield by Eq. (1). Finally, the fused hidden states $\mathbf{h}_j^f$ will be fed into the decoder for explanation generation.

3.6. Loss function and optimization

Sequential Recommendation Loss. As mentioned in Eq. (1), we apply the inner production operation between candidate item representation $\mathbf{r}_i$ and sequence representation $\mathbf{r}_s$ for recommendation. Hence, we leverage cross-entropy loss for model optimization, which can be formalized as below:

$$\begin{aligned} P_i &= \frac{\exp(\mathbf{r}_i \cdot \mathbf{r}_s^T)}{\sum_{i\in I}\exp(\mathbf{r}_i \cdot \mathbf{r}_s^T)} \\ \mathcal{L}_{rec} &= \frac{1}{|\mathcal{U}|}\sum_{i\in\mathcal{U}} -\log P_i \end{aligned} \tag{6}$$

Explanation Generation Loss. Same as the auto-regressive text generation paradigm in many works (Devlin et al., 2019; Lewis et al., 2020) (*i.e.,* based on the preceding text $[w_1, \ldots, w_m]$ to predict the next token w_{m+1}), we apply cross-entropy loss for explanation generation:

$$\mathcal{L}_{exp} = -\log P(w_{m+1}|w_1, w_2, \ldots, w_m) \tag{7}$$

We denote the ground-truth of the explanation as the corresponding review of the target item i and optimize the E to align with the review via Eq. (7), as we believe the item reviews can reveal both the users' real personalized preferences and item features from the user's perspective.

Model Optimization. The proposed adapter allows us to fine-tune the language model in an efficient and effective manner. Specifically, we first fine-tune the adapter-fused encoder of T5 to align with the sequential recommendation task. Then, we freeze the encoder and fine-tune the adapter-fused decoder and PPE module to realize the explanation generation. Consequently, during the inference stage, each item title in the sequence is concatenated into a sentence, which is then input into the adapter encoder to obtain the sequence representation for recommendation. Afterward, we can acquire the description of predicted items. The description, along with the sequence representation, is subsequently passed into the PPE module and adapter decoder to generate the explanation. As a result, we could output the recommendation and explanation via only a single go based on the PLEASER. The complete training process is illustrated by the following pseudocode.

3.7. Discussion

Learnable Parameters. Suppose ℓ is the length of sequence and d_k, d_v, d_f are the dimensions of the learnable vectors in adapters. K is the number of Transformer blocks. Hence, a total of $K(d_k + d_v + \ell d_f)$ new parameters will be supplemented for the adapter fine-tuning.

Theoretical Analysis of FFT Adaptation. Here, we give a theoretical analysis of the FFT adaptation.

Preliminary 1. Discrete Fourier Transform (DFT). Discrete Fourier Transform is widely used in digital signal processing and many practical applications. Given a sequence consisted of a series of continuous numbers $\{x_0, \ldots, x_{N-1}\}, x_n \in \mathbb{R} \quad \forall n \in \{0, 1, \ldots, N-1\}$,

Algorithm 2: Adapter-fused Decoder fine-tuning

Input:
Historical sequence: $s = [t_1, ..., t_\ell]$;
Target item description: $d_i = [[CLS], w_1, .., w_n]$;
Training Epoch: T;
Frozen Adapter Encoder: $f_\theta^E(\cdot)$;
Adapter Decoder: $f_\theta^D(\cdot)$;
Output:
Target Item Personalized Explanation: E;
Process:
for $t = 1, 2, .., T$ **do**
  $\mathbf{h}_\ell = f_\theta^E([t_1, ..., t_\ell])$; // Sequence Representation
  $[\mathbf{h}'_1, ...\mathbf{h}'_n] = f_\theta^E(d_i)$; // Description Representation
  $[\mathbf{h}_1^f, ..., \mathbf{h}_n^f] = \text{PPE}(\mathbf{h}_\ell; [\mathbf{h}'_1, ...\mathbf{h}'_n])$; // Representation Fusion
  $E_i = f_\theta^D([\mathbf{h}'_1, ..., \mathbf{h}'_n], [\mathbf{h}_1^f, ..., \mathbf{h}_n^f])$; // Explanation Generation
  $\mathcal{L}_{exp} = -\log p(w_m | w_1, ..., w_{m-1})$; // Loss
  $\theta_{fD} \leftarrow \theta_{fD} - \epsilon \nabla_{\theta_{fD}} \mathcal{L}_{rec}$; // Decoder fine-tuning
end for

the 1D DFT is given by:

$$X_k = \sum_{n=0}^{N-1} x_n e^{-\frac{i2\pi}{N} nk}, \quad k = 0, 1, \ldots, N-1 \tag{8}$$

where X_k is represents a complex number that encapsulates both the amplitude and phase of a complex sinusoidal component $e^{-\frac{i2\pi}{N} nk}$ of original input x_n. Therefore, the DFT could project the original sequence $\{x_n\}$ into a spectrum space with the frequency $w_k = \frac{2\pi k}{N}$. Note that the discrete Fourier transform is invertible, thus, we could reverse the original sequence from the X_k via the inverse DFT (IDFT), which can be formalized as follows:

$$x_n = \frac{1}{N} \sum_{k=0}^{N-1} X_k e^{\frac{i2\pi}{N} nk} \tag{9}$$

Preliminary 2. Fast Fourier Transform. A FFT (Van Loan, 1992) is an algorithm that efficiently computes the DFT transformation by factorizing the DFT matrix into a product of sparse factors, which predominantly consist of zeros. Such that, it manages to reduce the computational complexity associated with DFT from $O(N^2)$ to $O(NlogN)$ for the sequence of length N. For instance, Cooley–Tukey algorithm (Cooley & Tukey, 1965) is the most commonly used FFT on a wide range of published theories by far. It employs a divide-and-conquer strategy, recursively decomposing the original DFT of a size N into two smaller DFTs of sizes N_1 and N_2 ($N = N_1 N_2$), along with $O(N)$ multiplications via complex roots of unity, also known as *twiddle factors*. Similar to the DFT, the original inputs can also be recovered from the spectrum domain efficiently via the inverse FFT (IFFT).

Lemma 1. *FFT is equivalent to circular convolution but with a larger receptive field covering the full sequence.*

Proof. The DFT of the sequence $\{f^{(t)} * x^{(t)}\}[n]$ can be derived as below:

$$\begin{aligned}
\text{DFT}\{f_N * x\}[k] &\triangleq \sum_{n=0}^{N-1} \left(\sum_{m=0}^{N_1} f_N[m] \cdot x_N[n-m] \right) e^{\frac{-i2\pi kn}{N}} \\
&= \sum_{m=0}^{N-1} f_N[m] \left(\sum_{n=0}^{N-1} x_N[n-m] \cdot e^{\frac{-i2\pi kn}{N}} \right) \\
&= \sum_{m=0}^{N-1} f_N[m] e^{\frac{-i2\pi kn}{N}} \underbrace{\left(\sum_{n=0}^{N-1} x_N[n-m] \cdot e^{\frac{-i2\pi k(n-m)}{N}} \right)}_{\text{DFT}\{x_N\}[k] \text{ due to periodicity}} \\
&= \underbrace{\left(\sum_{m=0}^{N-1} f_N[m] e^{-i2\pi kn/N} \right)}_{\text{DFT}\{f_N\}[k]} (\text{DFT}\{x_N\}[k])
\end{aligned} \tag{10}$$

Consequently, we have,

$$\{f_N * x\}[n] = \mathcal{F}^{-1}\{\mathcal{F}(f) \cdot \mathcal{F}(x)\} \tag{11}$$

Defining the $\mathcal{F}(f) = \mathbf{w}$, *i.e.,* the learnable adapter in Eq. (4), the Eq. (11) can be further formalized as below:

$$f^{(t)} * x^{(t)} = \mathcal{F}^{-1}\left(\mathbf{w}^{(t)} \odot x^{(t)}\right) \tag{12}$$

where $\odot$ and $*$ are element-wise multiplication and circular convolution, respectively, consequently, we manifest that inserting an element-wise multiplication adapter in the frequency domain can be regarded as an alternative to the circular convolution operation with the kernel $f^{(t)}$. Moreover, compared with the conventional linear convolution operation, the circular convolution could model the sequential dependency with a larger receptive field covering the full sequence.

Lemma 2. *Applying the learnable element-wise multiplication in orthonormal bases of frequency domain can be construed as a filter to attenuate the noise and amplify the significant signals for better recommendation.*

Proof. Denote a $n \times n$ matrix whose (j,k)-entry is $\xi^{jk} = \omega^{(-jk)}$, where $\xi = e^{-2\pi i/n} = \cos\frac{2\pi}{n} - i\sin\frac{2\pi}{n} = \varpi$, $\xi^{-k} = \bar{\xi^k} = \omega^k$, for $0 \le j,k \le n-1$ as the Fourier matrix of order n, and it has the form,

$$\mathbf{F}_n = \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \xi & \xi^2 & \cdots & \xi^{n-1} \\ 1 & \xi^2 & \xi^4 & \cdots & \xi^{n-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \xi^{n-1} & \xi^{n-2} & \cdots & \xi \end{pmatrix}_{n\times n} \tag{13}$$

It is noteworthy that $\mathbf{F}_n$ is a special instance of the Vandermonde matrix and any two columns in $\mathbf{F}_n$, *e.g.,* the rth and sth, are mutually orthogonal,

$$\mathbf{F}_{*r}\mathbf{F}_{*s} = \sum_{j=0}^{n-1} \bar{\xi^{jr}}\xi^{js} = \sum_{j=0}^{n-1} \xi^{-jr}\xi^{js} = \sum_{j=0}^{n-1} \xi^{j(s-r)} = 0$$

Hence, the complex exponentials form a series of orthonormal bases in the frequency domain, and the magnitude of each basis can reveal some important characteristics of the raw signal. Compared with the rescaling adapter on the original space, the learnable element-wise multiplication, *i.e.,* $\mathbf{w}$ in Eq. (12), adopted on the orthonormal basis can be recognized as a filter for noise information attenuation, thereby enhancing the capability of models to discern and capture crucial features straightforwardly for an effective recommendation.

4. Experiments

We conduct extensive experiments on three popular datasets against nine baselines for sequential recommendation performance evaluation and three baselines for explanation generation. Overall, we aim to answer the following research questions:

- **RQ1.** How does the Pleaser perform compared with sequential recommendation baselines and explanation generation solutions?
- **RQ2.** How about the zero-shot learning ability of Pleaser against other representative alternatives?
- **RQ3.** How does each design choice made in Pleaser and different fine-tuning strategies affect its performance?
- **RQ4.** What is the efficiency of model training across different model sizes and learnable parameters?
- **RQ5.** How about the explanation generation quality and item representation distributions?

4.1. Datasets

We select three public datasets (*i.e., Amazon Instruments*, *Amazon Arts*, and *Amazon Office*) to fine-tune our Pleaser for recommendation performance evaluation. Besides, the other two datasets (*i.e., Amazon Software* and *Amazon Tools*) are utilized to assess the effectiveness of zero-short learning. All these datasets are collected from *Amazon* review (Ni, Li, & McAuley, 2019), which contain numerous user–item interactions; user reviews and ratings; item descriptions and titles; and so on. All of them are widely used in text-based sequential recommendation. Following the previous works (Kang & McAuley, 2018; Li, Xie, et al., 2024; Zhou et al., 2022), we recognize all the user–item ratings as interactions and filter out unpopular items and inactive users whose interacted times are less than five for all the datasets. We organize them along the timeline for sequence construction. In terms of personalized explanation generation, we obtain the review summaries from source datasets and remove the records without item descriptions. We truncate the item title longer than 15 tokens and set the maximum sequence length to 10 (item numbers) for all three datasets.[4] Regarding the item description and user review summary, the maximum token numbers are set to 128 as most of them are shorter than 100 tokens. Furthermore, we adopt the commonly used leave-one-out strategy for the test dataset, validation dataset, and training dataset split, *i.e.,* for all the datasets, given a sequence, we obtain the most recent interaction for testing, the penultimate interaction for validation, and the remains for training. The statistics for these datasets are presented in Table 3. We could observe that the average sequence lengths and dataset sizes exhibit significant variation, encompassing a wide range of real-world scenarios.

[4] It proves to be sufficient as most sequences are shorter than 7.

Table 3
Statistics of datasets after preprocessing. *Inters.* denotes the interactions. *Avg. s*, *Avg. t*, *Avg. d* and *Avg. r* are the average length of sequences, and average token numbers of item title, description, and review summary, respectively.

Datasets	#Users	#Items	#Inters.	Avg. s	Avg. t	Avg. d	Avg. r
Instruments	24,962	9,964	208,926	8.37	68.90	487.59	25.31
Arts	45,486	21,019	395,150	8.69	71.64	416.86	21.57
Office	87,436	25,986	684,837	7.84	89.21	449.64	23.88
Software	2,762	1,032	12,454	7.00	49.56	1378.40	32.81
Tools	219,151	70,914	2,057,668	8.60	86.72	534.60	25.37

4.2. Baselines and evaluation metrics

We select three groups of sequential recommendation methods for performance comparison. (1) *conventional ID-based methods*; (2) *ID-text solutions*; (3) *Text-based solutions with language models*, which will be further split into two categories: (a) pre-training and fine-tuning models, (b) full-parameter fine-tuning models; So as to the explanation generation, we select three methods, *i.e.,* **NRT** (Li, Wang, Ren, Bing, & Lam, 2017), **P5** (Geng et al., 2022), and **PETER** (Li et al., 2021).

- **Conventional ID-based Sequential Recommendation.** We select four representative sequential recommendation methods with different model architectures. **GRU4Rec**[5] (Hidasi et al., 2016) applies GRU to capture the sequential information for the next-item recommendation. **Caser**[6] (Tang & Wang, 2018) specializes in user's short-term interests modeling with integrated horizontal and vertical CNN. **SASRec**[7] (Kang & McAuley, 2018) adopts a uni-directional Transformer to capture the implicit connection between items for sequential recommendation. **BERT4Rec**[8] (Sun et al., 2019), in contrast, utilizes a bi-directional Transformer as the backbone and introduces the future behaviors as auxiliary information for current interaction prediction.
- **ID-Text Models.** We select two ID-text mixed models: **S^3-Rec**[9] (Zhou et al., 2020) and **FDSA** (Zhang et al., 2019), which introduce item's correspondent text (*e.g.,* category, brand, description, etc.), as side information for recommendation. Specifically, S^3-Rec proposes four objectives with the maximization (MIM) principle to capture the correlations among items, attributes, segments, and sequences for recommendation. FDSA proposes two types of self-attention modules for feature and item modeling respectively, then, a fully-connected layer is utilized to integrate the outputs from these two blocks for next-item prediction.
- **Text-based Models.** Moreover, we include five representative text-based models for comparison. **UniSRec**[10] (Hou et al., 2022) and **Recformer**[11] (Li, Wang, et al., 2023) use BERT (Devlin et al., 2019) and Longformer (Beltagy et al., 2020) respectively as base language model and dedicate different auxiliary task for representation enhancement and sequential recommendation. For UniSRec, we select transductive learning for recommendation, which has been verified to perform better against inductive learning. **P5**[12] (Geng et al., 2022) model also uses T5 as the base model for full-parameter fine-tuning and recommendation. We select P5-small and pre-train this model with dedicated prompts provided by the authors on the evaluation datasets for comparison. [**LLaRA**] (Liao et al., 2024)[13] is a prompt-driven LLMs-based method, which applies curriculum learning for sequential recommendation. **LITE-LLM4Rec** (Wang et al., 2024) removes the beam-search decoding process, instead, a projection head tailored to recommendation is integrated for an efficient recommendation.
- **Explanation Generation Models.** We adopt three baselines with respect to the task of explanation generation. **NRT** (Li et al., 2017) use generated explanation for recommendation performance improvement. Instead of GRU in NRT, **PETER** (Li et al., 2021) applies a uni-directional Transformer as the backbone for explanation generation. Based on prompt engineering, **P5** considers the rating scores, user, item, and its feature for explanation generation.

We adopt HR@K (Hit Rate) and NDCG@K (Normalized Discounted Cumulative Gain), the two most popular metrics for recommendation precision and ranking quality evaluation. Specifically, HR@K reveals the number of recommended hits within the top-K list. NDCG@K further provides a more nuanced evaluation of ranking performance by taking into account the positions of these hits within the list. Higher values of these metrics indicate superior performance. Let R_K be the recommended list including the items with Top K predicted scores and R represents the ground-truth, *i.e.,* the item the user clicked in the next step. Suppose

[5] https://github.com/hidasib/GRU4Rec
[6] https://github.com/graytowne/caser
[7] https://github.com/kang205/SASRec
[8] https://github.com/FeiSun/BERT4Rec
[9] https://github.com/RUCAIBox/CIKM2020-S3Rec
[10] https://github.com/RUCAIBox/UniSRec
[11] https://github.com/AaronHeee/RecFormer
[12] https://github.com/jeykigung/P5
[13] https://github.com/ljy0ustc/LLaRA

we have N samples, HR and NDCG can be calculated as follows:

$$\begin{aligned} \text{HR@}K &= \frac{\sum_{i=1}^{N} |R_{K_i} \cap R_i|}{N} \\ \text{NDCG@}K &= \frac{1}{N}\sum_{i=1}^{N}\frac{1}{Z}\text{DCG@}K = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{1}{Z}\sum_{j=1}^{K}\frac{2^{r_{ij}}-1}{\log_2(j+1)}\right) \end{aligned} \tag{14}$$

where $r_{ij} \in \{0, 1\}$ indicates the presence of the ground-truth number j appears in the recommended list for sample i. Z represents a normalization constant, defined as the maximum possible value of DCG. The value of NDCG@K is assigned a value of zero when the rank exceeds K. In this paper, we present the experimental results at $K = 5, 10, 20$. As the evaluation results will be inconsistent with the practical scenario when the sample size of negative items is limited (Krichene & Rendle, 2020), we, thereby, randomly sample 1000 negative samples that do not appear in the user's historical interacted sequence as candidates for each ground-truth, to balance the efficiency and authenticity of evaluation.

So as to the personalized explanation generation, we consider two commonly used metrics (*i.e.,* BLEU and ROUGE scores) to evaluate the quality of our generated personalized explanations in the field of NLP. To be specific, the BLEU score measures the precision of the n-grams (*i.e.,* contiguous sequences of n words) in the generated sentence by comparing it to the reference ground-truth. To avoid a shorter generation, a brevity penalty (BP) is employed to modify the precision, *i.e.,* the shorter the generation, the smaller the BP will be. This process can be formalized as below,

$$\begin{aligned} \text{BLEU} &= \text{BP} * \exp(\sum_{n=1}^{N} w_n \mathbf{log} p_n) \\ \text{BP} &= \begin{cases} 1 & \text{if } c > r \\ \exp(1 - r/c) & \text{if } c \le r \end{cases} \end{aligned} \tag{15}$$

where r and c are the length (*i.e.,* total number of words) of the reference and generated sentence, respectively. p_n is the precision of n-grams, which is counted as the number of n-grams that appear in both generation and reference text divided by the total number of n-grams in the reference text. w_n is a uniform weight, *i.e.,* $n = 1/N$.

The ROUGE score measures the similarity via the recall of n-grams. The formula for calculating the ROUGE score is presented below:

$$\text{ROUGE} = \sum_{n=1}^{N}(\text{Recall of } n\text{-grams}) \tag{16}$$

where Recall of n-grams is defined as the ratio of the number of n-grams present in both the reference and the generated text to the total number of n-grams in the reference text.

4.3. Experimental settings

For a fair comparison, we followed (Kang & McAuley, 2018; Zhou et al., 2022) and used the Adam optimizer, setting the initial learning rate to $5e-4$. We obtain T5-small[14] as our backbone and also explore the impacts of other model sizes on the final performance (ref. Section 4.6). We utilize the early-stop strategy for three datasets, *i.e.,* no improvement over three consecutive epochs to relieve the overfitting issue. The monitor indicators are HR@10 and BLEU-1 for sequential recommendation and explanation generation respectively. We conduct the student *t-test* for the statistical significance test.

4.4. Overall comparison (RQ1)

Sequential Recommendation. The overall comparison results of PLEASER against other baselines are illustrated in Table 4. Here, we could draw the following observations.

First, compared with the conventional sequential modeling methods, introducing the item text as side information can acquire superior results in general. Specifically, S^3-Rec and FDSA achieve comparable results across the three datasets. On *Amazon Arts* dataset, S^3-Rec outperforms FDSA, in turn, on *Amazon Office* dataset we have the opposite conclusion. On *Amazon Instruments* dataset, none of them are capable of achieving absolute dominance on all the metrics. Therefore, we believe text modeling and fusion methods proposed by FDSA and S^3-Rec may not be adapted well to all the scenarios, a more robust method is expected to be explored.

Text-based sequential recommendation with language models achieves optimal results against ID-based models. More concretely, UniSRec and Recformer are the best and second-best performers excluding our PLEASER across all the datasets, but the performance gap between them is not significant. We argue this observation might be caused by the framework similarity of these two models, *i.e.,* both of them are pre-training and fine-tuning pipelines. Through prompts learning, P5 and LLaRA concates the item ID as a sentence and obtain the next generated tokens as the predicted item for recommendation. As this strategy does not introduce any

[14] https://huggingface.co/t5-small

Table 4
Sequential recommendation comparison. The best results are highlighted in boldface, and the second-best results are indicated with underlining. The symbol ▲% represents the relative improvement of Pleaser compared to the best baseline. * denotes a significant improvement of Pleaser over the best baseline results, as determined by a t-test with a significance level of $P < 0.05$.

Datasets	Metric	GRU4Rec	Caser	SASRec	BERT4Rec	FDSA	S^3-Rec	UniSRec	Recformer	P5	LLaRA	LITE-LLM4Rec	Pleaser	▲%
Instruments	HR@5	0.1024	0.1167	0.1058	0.1119	0.1118	0.1123	0.1593	0.1466	0.0770	0.1262	0.1532	**0.1892***	18.76%
	HR@10	0.1586	0.1662	0.1534	0.1458	0.1544	0.1522	0.2098	0.1937	0.0881	0.1656	0.1835	**0.2486***	18.49%
	HR@20	0.2280	0.2243	0.2152	0.1948	0.2116	0.2031	0.2703	0.2572	0.1033	0.2397	0.2774	**0.3228***	16.37%
	NDCG@5	0.0672	0.0784	0.0701	0.0899	0.0795	0.0810	0.1191	0.1123	0.0682	0.0887	0.1161	**0.1411***	18.56%
	NDCG@10	0.0852	0.0944	0.0854	0.1008	0.0932	0.0939	0.1354	0.1278	0.0718	0.1143	0.1346	**0.1603***	18.44%
	NDCG@20	0.1027	0.1091	0.1010	0.1132	0.1076	0.1067	0.1506	0.1438	0.0756	0.1369	0.1581	**0.1790***	18.86%
Arts	HR@5	0.0546	0.0742	0.0747	0.0788	0.1160	0.1490	0.2343	0.2176	0.0720	0.2084	0.2232	**0.2486***	6.10%
	HR@10	0.0974	0.1150	0.1125	0.1022	0.1500	0.2087	0.2976	0.2802	0.0799	0.2432	0.2971	**0.3195***	7.36%
	HR@20	0.1671	0.1713	0.1742	0.1392	0.1964	0.2823	0.3732	0.3530	0.0867	0.3063	0.3812	**0.3992***	4.72%
	NDCG@5	0.0336	0.0490	0.0485	0.0636	0.0913	0.1061	0.1789	0.1654	0.0630	0.1402	0.1659	**0.1877***	4.92%
	NDCG@10	0.0474	0.0620	0.0607	0.0711	0.1022	0.1254	0.1993	0.1855	0.0656	0.1617	0.2061	**0.2106***	2.18%
	NDCG@20	0.0649	0.0762	0.0762	0.0804	0.1139	0.1439	0.2183	0.2039	0.0673	0.1821	0.2205	**0.2307***	5.68%
Office	HR@5	0.0546	0.1003	0.0868	0.1177	0.1439	0.1278	0.2160	0.1865	0.0867	0.1634	**0.2218**	0.2216	2.59%
	HR@10	0.0974	0.1593	0.1339	0.1501	0.1821	0.1809	0.2675	0.2286	0.0928	0.2483	0.2751	**0.2782***	4.00%
	HR@20	0.1671	0.2352	0.2051	0.1990	0.2365	0.2519	0.3313	0.2810	0.0996	0.2866	0.3215	**0.3486***	5.22%
	NDCG@5	0.0336	0.0656	0.0604	0.0972	0.1166	0.0903	0.1707	0.1493	0.0776	0.0904	0.1735	**0.1737***	1.76%
	NDCG@10	0.0474	0.0845	0.0755	0.1076	0.1289	0.1073	0.1873	0.1629	0.0795	0.1549	0.1882	**0.1919***	2.46%
	NDCG@20	0.0649	0.1036	0.0934	0.1199	0.1425	0.1252	0.2034	0.1760	0.0813	0.1727	0.2010	**0.2096***	3.05%

Table 5
Performance (%) comparison of our Pleaser against the other three baselines on explanation generation task. The best results are highlighted in boldface, and the second-best results are underlined.

Methods	Amazon instruments			Amazon arts			Amazon office		
	BLEU-1	ROUGE-1	ROUGE-L	BLEU-1	ROUGE-1	ROUGE-L	BLEU-1	ROUGE-1	ROUGE-L
NRT	1.0194	6.9603	5.9234	1.1712	10.7163	9.0166	1.8392	11.8323	11.4915
PETER	2.5976	9.4303	9.4150	5.4227	14.2032	14.2021	4.1470	12.3638	12.3612
P5	6.0748	**18.0630**	**17.7641**	12.2219	**31.3314**	**31.3305**	8.6386	**26.2570**	**26.2551**
Pleaser	**6.1417**	15.3271	14.8982	**12.8414**	30.9417	30.9074	**8.7909**	25.2370	25.1840

text information associated with items, the domain knowledge encapsulated in the language model remains insufficiently exploited. Consequently, the results are only close to conventional baselines and inferior to the text-based solutions. On the contrary, LITE-LLM4Rec adopts head projection to predict the candidate items probability straightforwardly for the next-item recommendation, achieving suboptimal results.

Our method consistently demonstrates superior performance compared to all baselines. Specifically, in comparison to the best baseline, Pleaser achieves improvements of up to 19.42%/18.76%, 7.36%/5.68% and 5.22%/3.05% (HR/NDCG) improvements on *Amazon Instruments*, *Amazon Arts*, and *Amazon Office* datasets respectively. However, as the dataset size increases, the performance improvement gradually diminishes, *i.e.,* on *Amazon Office* dataset, Pleaser only achieves up to 5.22%/3.05% (HR/NDCG) improvements. We believe the parameter-efficient fine-tuning might be sufficient for small-scale datasets (*e.g.,* the *Amazon Instruments*) However, as the size of the dataset expands, more parameters necessitate being optimized to learn the implicit dependency and complicated patterns inherent in the text. Hence, we can increase the size of learnable parameters for performance further improvement (ref. Table 7 for details).

Personalized Explanation Generation. Table 5 reports the overall performance of Pleaser against the other three models on *Amazon Instruments*, *Amazon Arts*, and *Amazon Office* datasets. We could observe that the NRT and PETER obtain the worst and second worst performance since neither of them applies the language model as backbones. Besides, the sentence length of the input and the generated counterpart are quite short (*i.e.,* 20 and 15 respectively on the paper settings). Compared with P5, our Pleaser achieves the best performance on BLEU-1, while inferior to P5 on the ROUGE metrics. However, it is noteworthy that although both two methods select T5 as the base model, the training time and trainable parameters of our method are significantly less than P5 (*i.e.,* 5.2M *vs* 78.5M). If we adopt the full-parameter fine-tuning, the performance of Pleaser could be improved significantly (ref. Table 7 for details).

4.5. Zero-shot learning (RQ2)

We compare our Pleaser with four representative baselines, including SASRec, S^3-Rec, UniSRec, and P5, to investigate the zero-learning ability.[15] Observing Fig. 3, we could find that SASRec achieves the worst results as the item ID does not contain any knowledge across different domains. Therefore, it is intractable to leverage the ID-based sequential modeling methods for the zero-shot recommendation. S^3-Rec obtains the second-worst results, this is because it only leverages the domain-specific text information

[15] For a fair comparison, we select the well-trained models with the best performance on *Amazon Office* dataset and evaluate the zero-shot learning ability on the other two datasets (*i.e., Amazon Software* and *Amazon Tools*).

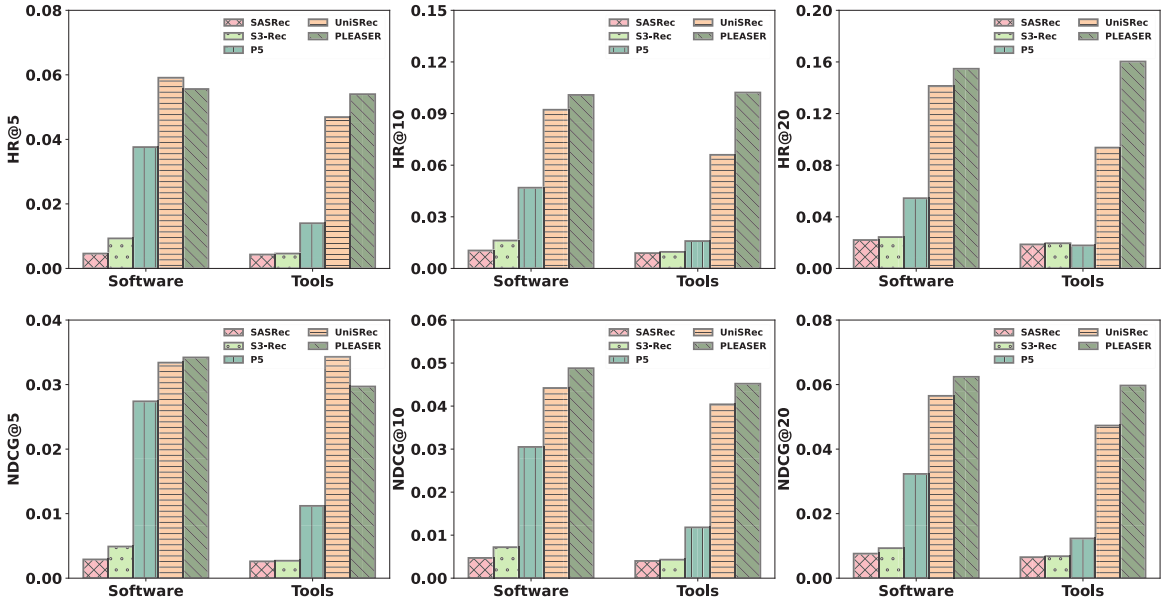

Fig. 3. The completed zero-shot learning results of our PLEASER against the other four recipes on *Amazon Software* and *Amazon Tools* dataset.

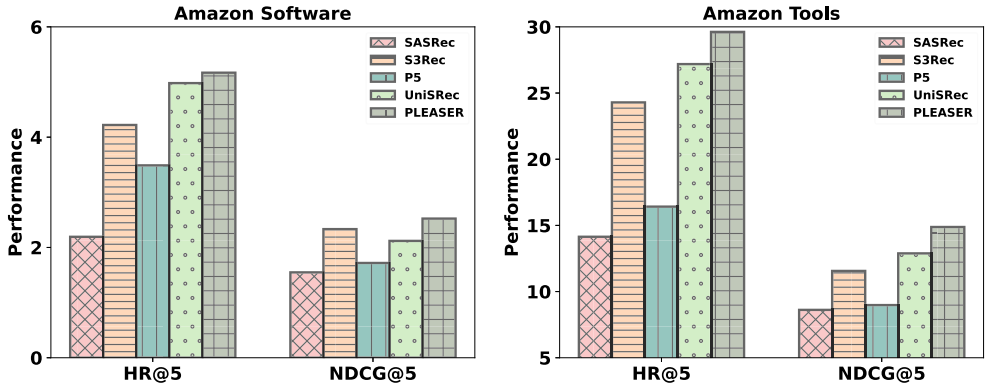

Fig. 4. The fine-tuning performance of our PLEASER against the other four recipes on *Amazon Software* and *Amazon Tools* dataset..

e.g., brand, and category, which contains limited cross-domain information for knowledge sharing and transfer. It is encouraging to note that the results of the P5 model are ranked in the middle, which illustrates the potential of language models for zero-shot learning. Our method and UniSRec achieve the best and second-best results since they both consider item titles as text information and leverage the language models as backbones for next-click item prediction. Additionally, we present the fine-tuning results of our PLEASER in comparison to representative models on the *Amazon Software* and *Amazon Tools* datasets, as shown in Fig. 4, to comprehensively evaluate the effectiveness of our approach. The results demonstrate that our method consistently outperforms the baselines across all evaluation metrics. Consequently, we believe the unified text representation and the general knowledge encapsulated in the language models not only improve overall performance but also can bridge the gap between different domains for zero-shot learning capability enhancement.

4.6. Ablation study (RQ3)

To evaluate the effectiveness of each module design and also investigate the influence of fine-tuning solutions and model size on the final improvements, we consider full-parameter fine-tuning (FPFT) and PEFT with regard to various model sizes, observing the performance variation.

- **w/o Rescaling.** Removing rescaling adapter from PLEASER for sequential recommendation.
- **w/o FFT.** Removing Fast Fourier Transformer adapter from PLEASER for sequential recommendation.
- **w/o PPE.** Removing personalized preference extraction module from PLEASER for explanation generation.

Table 6
Results of ablation experiments. The best results are highlighted in boldface.

Dataset	Ablation	HR@5	HR@10	HR@20	NDCG@5	NDCG@10	NDCG@20	BLEU-1	ROUGE-1	ROUGE-L
Instruments	w/o Rescaling	0.1773	0.2341	0.3040	0.1230	0.1503	0.1680	–	–	–
	w/o FFT	0.1691	0.2227	0.2908	0.1262	0.1436	0.1607	–	–	–
	w/o PPE	–	–	–	–	–	–	0.0554	0.1244	0.1221
	PLEASER	**0.1892**	**0.2486**	**0.3228**	**0.1411**	**0.1603**	**0.1790**	**0.0614**	**0.1533**	**0.1490**
Arts	w/o Rescaling	0.2227	0.2927	0.3706	0.1652	0.1878	0.2074	–	–	–
	w/o FFT	0.2239	0.2927	0.3707	0.1661	0.1883	0.2080	–	–	–
	w/o PPE	–	–	–	–	–	–	0.1130	0.2976	0.2975
	PLEASER	**0.2486**	**0.3195**	**0.3992**	**0.1877**	**0.2106**	**0.2307**	**0.1284**	**0.3094**	**0.3091**

Table 7
Ablation experimental results of various model sizes (small vs. base vs. large) with FPFT strategy and PEFT strategy. OOM means out-of-memory. The third and fourth columns indicate the whole number of learnable parameters vs. the model size with regard to the sequential recommendation (Rec.) task and explanation generation (Exp.) task. The best results are highlighted in boldface.

Dataset	Ablation	Rec. parameters	Exp. parameters	HR@5	HR@10	NDCG@5	NDCG@10	BLEU-1	ROUGE-1	ROUGE-L
Instruments	Small & PEFT	3.7M/35.9M	5.2M/78.5M	0.1892	0.2486	0.1411	0.1603	0.0614	0.1533	0.1490
	Small & FPFT	35.9M/35.9M	42.7M/78.5M	**0.1933**	**0.2559**	**0.1461**	**0.1662**	0.0769	0.2064	0.2060
	Base & PEFT	8.7M/111M	12M/251M	0.1842	0.2459	0.1372	0.1571	0.0726	0.1769	0.1758
	Base & FPFT	111M/111M	140M/251M	0.1852	0.2453	0.1377	0.1571	**0.0790**	**0.2124**	**0.2087**
	Large & PEFT	16.8M/339M	22.6M/780M	0.1723	0.2310	0.1291	0.1480	0.0740	0.1840	0.1813
	Large & FPFT	339M/339M	441M/780M	OOM	OOM	OOM	OOM	OOM	OOM	OOM

- **w LoRA.** Replacing the FFT adapter and rescaling adapter with LoRA (Hu et al., 2021) for PEFT.
- **w FPFT.** Replacing the PEFT with FPTT.[16]
- **w Base.** Replacing the T5-small with T5-base as the backbone of PLEASER.
- **w Large.** Replacing the T5-small with T5-large as the backbone of PLEASER.

Ablation on Module Design. Table 6 shows the ablation results of module design. We could find that the performance of both sequential recommendation and explanation generation will decrease when any of the rescaling adapter, FFT adapter, and PPE are removed. Despite only one type of adapter being used for PEFT, our PLEASER still outperforms the strongest sequential recommendation baselines (ref. Table 4). Also, PLEASER achieves superior results compared with LoRA, demonstrating the effectiveness of our dedicated adapters. Moreover, in comparison to the rescaling adapter, the FFT adapter obtain better performance manifesting the superiority of the circular convolution for sequential dependency modeling. Furthermore, as to the explanation generation, the significant degeneration in performance (*i.e.,* for *Amazon Arts* dataset the BLEU-1 and ROUGE-1 decrease by 9.77% and 18.85% respectively) also validates the effectiveness of the PPE module.

Ablation on Model Size and Learnable Parameters. Table 7 illustrates the recommendation and explanation generation performance of our PLEASER under different model sizes and fine-tuning strategies. As for the sequential recommendation, given the same fine-tuning strategies, the performance will slightly decline with an increase in model size. However, as to the explanation generation, a contrasting trend is observed, *i.e.,* the larger language models tend to attain high-quality explanations. Although the PEFT performance is superior to the FPFT under the same model size, the trainable parameters of PEFT are only 5% (16.8M vs. 339M) to 10% (3.7M vs. 35.9M) against FPFT, thus, we believe the PEFT is a competitive and effective solution for model fine-tuning under limited computational resources scenario.

4.7. Training efficiency (RQ4)

Mode Size. We further analyze the effects of the number of learnable parameters and model size on the training time and efficiency. For a fair comparison, all experiments were conducted on four NVIDIA A100 GPUs with 40GB memory and AMD EPYC 7543 2.8 GHz CPU.

To be specific, for varying model sizes with different fine-tuning strategies, we train one epoch on *Amazon Instruments* dataset and plot the training time with regard to different learnable parameters, as depicted in Fig. 5. We could find that the training time of PEFT is considerably shorter than FPFT, (*i.e.,* about half at most when the base model is T5-small). Additionally, the trainable

[16] For a fair comparison, we freeze the well-trained encoder block and token embedding from recommendation task and only fine-tune all the decoders for explanation generation evaluation.

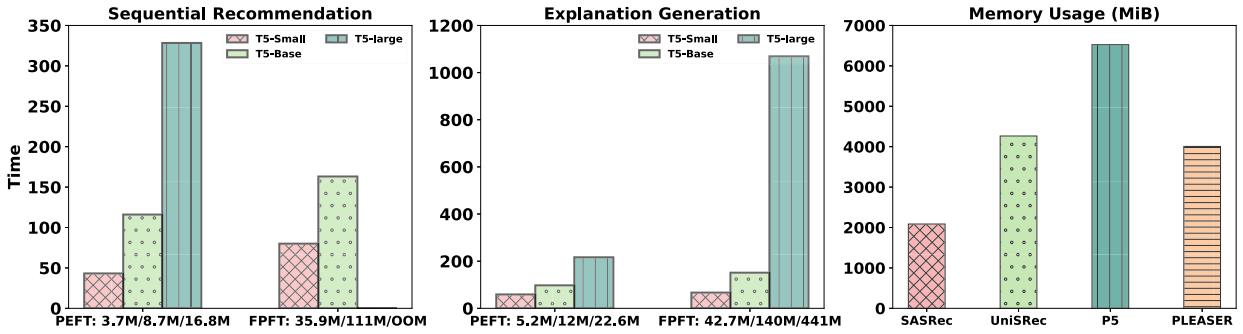

Fig. 5. One epoch train time of sequential recommendation (left), explanation generation (middle) under different learnable parameters and fine-tuning strategies and peak GPU memory usage (right) on *Amazon Instrument* datasets. The horizontal axis of the two sub-figures on the left showcases the learnable parameter numbers. OOM means out-of-memory..

Table 8
Example cases of explanation generation yield by PLEASER on *Amazon Instruments* dataset. Purple text indicates the subject of the comment and the orange text reveals the users emotion and attitude.

Case	Generation	Reference
Case 1	Great stand for cradle.	Nice Stand.
Case 2	Elixir Strings 80/20 Bronze Acoustic Guitar Strings.	Fits my 12 string guitar perfectly!!.
Case 3	It is perfect to use.	Small PA is perfect for us!
Case 4	Great picks.	Nice picks!.
Case 5	Great sound, but I've got my C01U mic.	Currently, the best mic for under 50 (maybe 60 or 70).
Case 6	Nice Electric Guitar (Gloss Black).	Satisfactory for classical guitars.

parameters of PEFT are only 5% compared to FPFT (*i.e.,* 16.8M vs 339M with T5-large), validating the efficiency and effectiveness of the proposed method. As for the explainable generation, we could also make similar conclusions.

Memory Usage. We further investigate one-batch peak per GPU memory usage of our PLEASER against three representative baselines, including SASRec,[17] UniSRec, and P5, as shown in Fig. 5. Owing to the adapter fine-tuning, the memory usage of our solution is comparable to UniSRec, higher than SASRec but significantly less than P5, a T5-based recommendation model.

4.8. Case study (RQ5)

Explanation Generation. Table 8 provides some explanation examples on *Amazon Instruments* dataset generated by PLEASER with T5-small parameter-efficient fine-tuning. We can observe that our PLEASER can capture the subject (purple text) of the description and also generate explanations that are consistent with the sentiment (orange text) of the reference.

Uniformity and Alignment of Item Representation. To conduct a more in-depth analysis of the efficacy of our proposed method in item representation distribution, following (Wang & Isola, 2020), we analyze the *alignment*[18] and *uniformity*,[19] the two properties, of our PLEASER against SASRec, a representative conventional sequential modeling method. Concretely, we randomly sample 2048 items and obtain their representations generated by the SASRec and PLEASER respectively. Plotting their Min-Max normalized ℓ_2 distance distribution and representation distribution on $\mathcal{S}^1$ space with $\arctan2(y, x)$ coordinates, as shown in Fig. 6. **Alignment analysis**: comparing the histogram of ℓ_2 distance distribution, we could find that the mean value of our PLEASER is significantly bigger than SASRec, which renders the irrelevant item representations generated by our PLEASER acquire a distinct separation in hyperspace. **Uniformity analysis**: the $\mathcal{S}^1$ distribution (top-right subfigure) indicates that the item representation generated by our PLEASER spreads on the whole space. The histogram (bottom-right subfigure) further illustrates that our PLEASER has a more uniform distribution compared to SASRec, which manifests three distinct peaks on the KDE curve. Overall, different from conventional sequential recommendation with item ID, we believe the item representation yield by our method could be enhanced from both **uniformity** and **alignment** two perspectives, which will be facilitated for downstream tasks.

[17] For a fair comparison, we set the hidden size as the same as P5 and PLEASER, *i.e.,* $d = 512$.

[18] The representation of relevant or similar samples should be mapped to nearby features and be close to each other in the hypersphere, on the contrary, the irrelevant samples should be far apart in the hypersphere.

[19] Representations of samples should be uniformly distributed on the unit hypersphere.

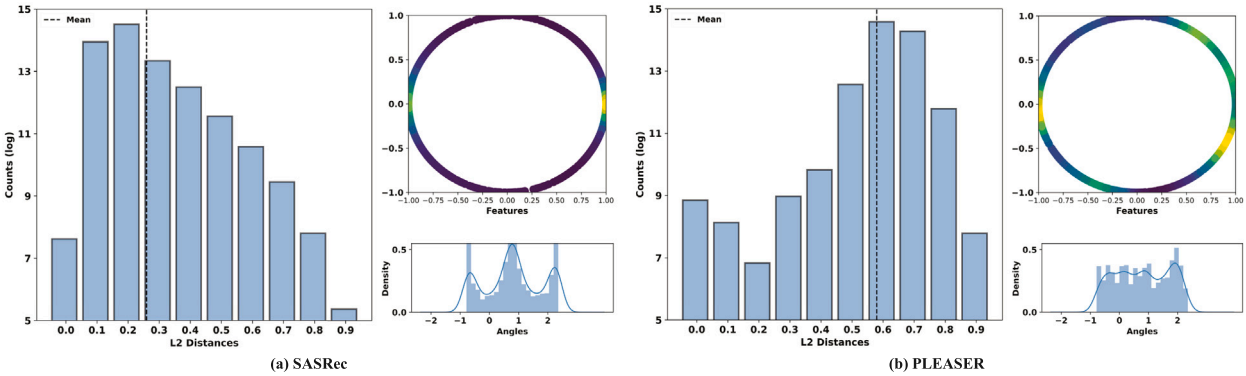

Fig. 6. *Uniformity* and *Alignment* analysis for the item representation generated by SASRec (left) and PLEASER (right). *Alignment analysis*: the histograms elucidate the Min-Max normalization distribution of ℓ_2 distance between all item pair representations, where the black dotted lines show the mean value of the distance. We could find that our method acquires a bigger mean value against SASRec, illustrating a distinct distance between irrelevant items in the hyperspace. *Uniformity analysis*: the subfigure (top-right) plots the item representation distribution on S^1 (two-dimensional space with angle coordinates *i.e.,* arctan2(y, x)) via t-SNE. The darkness of the color indicates the density of distributions, *i.e.,* the darker the color, the denser the distribution. The bottom-right histogram further illustrates the angle distribution with Gaussian kernel density estimation (KDE), which showcases our PLEASER possesses a more uniform item representation distribution against SASRec.

5. Conclusion

This work attempts to achieve personalized explainable sequential recommendation with a language model in a holistic, efficient yet effective way. To instantiate this idea, we propose PLEASER, which selects T5, an encoder–decoder model architecture, as the backbone and designs a rescaling adapter and a Fast Fourier Transform adapter for parameter-efficient fine-tuning. Besides, a personalized preference extraction (PPE) module is adopted to consider both the user's historical preference and the target item characteristics simultaneously for personalized explanation generation. Through a single inference step, our PLEASER could yield the recommended results and the corresponding personalized explanations simultaneously. Comprehensive experiments on three public datasets show that with only 5% parameter fine-tuning, our PLEASER outperforms the competitive baselines on both sequential recommendation and explanation generation tasks. Further analysis shows the remarkable ability of PLEASER on zero-shot learning and the effectiveness of each module design. The case studies also illustrate the superiority of our method on both uniformity and alignment, two perspectives. In future work, we will consider conducting human evaluations in addition to automated metrics, *i.e.,* BLEU, ROUGE for assessing the quality of personalized explanations from a user's perspective. Note that calculating the softmax of the inner product between sequential representation and item representation is an essential operation for the sequential recommendation. However, this process will incur extensive computing costs, which are particularly prominent in practical scenarios, given the number of items is increased significantly. Moreover, due to the inherent limitations of the attention mechanism, LLMs face challenges in efficiently modeling very long sequences, hindering the widespread application in long-sequence recommendation tasks. In the future, we aim to alleviate these issues without performance sacrifice by leveraging linear attention techniques.

CRediT authorship contribution statement

Zihao Li: Writing – review & editing, Writing – original draft, Visualization, Validation, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Lixin Zou:** Writing – review & editing, Methodology, Investigation. **Chao Ma:** Writing – review & editing, Investigation. **Chenliang Li:** Writing – review & editing, Supervision, Resources, Project administration, Methodology, Investigation, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by National Natural Science Foundation of China (No. 62272349, No. U23A20305, and No. 62302345); Natural Science Foundation of Hubei Province under Grant Numbers 2023BAB160 and 2022CFB012; the Ministry of Education Humanities and Social Sciences Project under Grant No. 24JDSZ3073; and CAAI-Ant Group Research Fund (No. CAAI-MYJJ2024-01). The numerical calculations in this paper have been done on the supercomputing system at the Supercomputing Center of Wuhan University. Chenliang Li is the corresponding author.

References

Bao, K., Zhang, J., Zhang, Y., Wang, W., Feng, F., & He, X. (2023). Tallrec: An effective and efficient tuning framework to align large language model with recommendation. arxiv preprint URL https://arxiv.org/abs/2305.00447.

Beltagy, I., Peters, M. E., & Cohan, A. (2020). Longformer: The long-document transformer. arxiv preprint URL https://arxiv.org/abs/2004.05150.

Chang, J., Gao, C., Zheng, Y., Hui, Y., Niu, Y., Song, Y., et al. (2021). Sequential recommendation with graph neural networks. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval* (pp. 378–387).

Chen, H., Chen, X., Shi, S., & Zhang, Y. (2021). Generate natural language explanations for recommendation. arxiv preprint URL https://arxiv.org/abs/2101.03392.

Chen, X., Chen, H., Xu, H., Zhang, Y., Cao, Y., Qin, Z., et al. (2019). Personalized fashion recommendation with visual explanations based on multimodal attention network: Towards visually explainable recommendation. In B. Piwowarski, M. Chevalier, E. Gaussier, Y. Maarek, J. Nie, F. Scholer (Eds.), *Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval* (pp. 765–774). ACM, http://dx.doi.org/10.1145/3331184.3331254.

Chen, R., Fan, J., & Wu, M. (2023). MC-RGN: Residual Graph Neural Networks based on Markov Chain for sequential recommendation. *Information Processing & Management*, *60*(6), Article 103519.

Chen, X., Xu, H., Zhang, Y., Tang, J., Cao, Y., Qin, Z., et al. (2018). Sequential recommendation with user memory networks. In Y. Chang, C. Zhai, Y. Liu, Y. Maarek (Eds.), *Proceedings of the eleventh ACM international conference on web search and data mining* (pp. 108–116). ACM, http://dx.doi.org/10.1145/3159652.3159668.

Chiang, W.-L., Li, Z., Lin, Z., Sheng, Y., Wu, Z., Zhang, H., et al. (2023). Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. See https://vicuna.lmsys.org. (Accessed 14 April 2023).

Cooley, J. W., & Tukey, J. W. (1965). An algorithm for the machine calculation of complex Fourier series. *Mathematics of Computation*, *19*(90), 297–301.

Cui, Z., Ma, J., Zhou, C., Zhou, J., & Yang, H. (2022). M6-rec: Generative pretrained language models are open-ended recommender systems. arxiv preprint URL https://arxiv.org/abs/2205.08084.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)* (pp. 4171–4186). Minneapolis, Minnesota: Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/N19-1423, URL https://aclanthology.org/N19-1423.

Ding, H., Ma, Y., Deoras, A., Wang, Y., & Wang, H. (2021). Zero-shot recommender systems. arXiv preprint URL https://arxiv.org/abs/2105.08318.

Duan, H., Zhu, Y., Liang, X., Zhu, Z., & Liu, P. (2023). Multi-feature fused collaborative attention network for sequential recommendation with semantic-enriched contrastive learning. *Information Processing & Management*, *60*(5), Article 103416.

Fu, Z., Xian, Y., Gao, R., Zhao, J., Huang, Q., Ge, Y., et al. (2020). Fairness-aware explainable recommendation over knowledge graphs. In J. Huang, Y. Chang, X. Cheng, J. Kamps, V. Murdock, J. Wen, Y. Liu (Eds.), *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval* (pp. 69–78). ACM, http://dx.doi.org/10.1145/3397271.3401051.

Geng, S., Liu, S., Fu, Z., Ge, Y., & Zhang, Y. (2022). Recommendation as language processing (rlp): A unified pretrain, personalized prompt & predict paradigm (p5). In *Proceedings of the 16th ACM conference on recommender systems* (pp. 299–315).

Guo, L., Zhang, J., Chen, T., Wang, X., & Yin, H. (2022). Reinforcement learning-enhanced shared-account cross-domain sequential recommendation. *IEEE Transactions on Knowledge and Data Engineering*.

He, X., Chen, T., Kan, M., & Chen, X. (2015). TriRank: Review-aware explainable recommendation by modeling aspects. In J. Bailey, A. Moffat, C. C. Aggarwal, M. de Rijke, R. Kumar, V. Murdock, T. K. Sellis, & J. X. Yu (Eds.), *Proceedings of the 24th ACM international conference on information and knowledge management* (pp. 1661–1670). ACM, http://dx.doi.org/10.1145/2806416.2806504.

Hidasi, B., Karatzoglou, A., Baltrunas, L., & Tikk, D. (2016). Session-based recommendations with recurrent neural networks. In Y. Bengio, & Y. LeCun (Eds.), *4th international conference on learning representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, conference track proceedings*. URL http://arxiv.org/abs/1511.06939.

Hou, Y., Mu, S., Zhao, W. X., Li, Y., Ding, B., & Wen, J.-R. (2022). Towards universal sequence representation learning for recommender systems. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining* (pp. 585–593).

Hou, Y., Zhang, J., Lin, Z., Lu, H., Xie, R., McAuley, J., et al. (2023). Large language models are zero-shot rankers for recommender systems. arXiv preprint URL https://arxiv.org/abs/2305.08845.

Hu, E. J., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., et al. (2021). LoRA: Low-rank adaptation of large language models. In *International conference on learning representations*.

Hu, Z., Wang, L., Lan, Y., Xu, W., Lim, E.-P., Bing, L., et al. (2023). LLM-adapters: An adapter family for parameter-efficient fine-tuning of large language models. In *Proceedings of the 2023 conference on empirical methods in natural language processing* (pp. 5254–5276).

Kang, W.-C., & McAuley, J. (2018). Self-attentive sequential recommendation. In *ICDM* (pp. 197–206). IEEE.

Kojima, T., Gu, S. S., Reid, M., Matsuo, Y., & Iwasawa, Y. (2022). Large language models are zero-shot reasoners. *Advances in Neural Information Processing Systems*, *35*, 22199–22213.

Krichene, W., & Rendle, S. (2020). On sampled metrics for item recommendation. In R. Gupta, Y. Liu, J. Tang, & B. A. Prakash (Eds.), *KDD '20: the 26th ACM SIGKDD conference on knowledge discovery and data mining* (pp. 1748–1757). ACM, URL https://dl.acm.org/doi/10.1145/3394486.3403226.

Lee-Thorp, J., Ainslie, J., Eckstein, I., & Ontanon, S. (2021). Fnet: Mixing tokens with fourier transforms. arxiv preprint URL https://arxiv.org/abs/2105.03824.

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., et al. (2020). BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th annual meeting of the association for computational linguistics* (pp. 7871–7880). Online: Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/2020.acl-main.703, URL https://aclanthology.org/2020.acl-main.703.

Li, Z., Sun, A., & Li, C. (2023). DiffuRec: A diffusion model for sequential recommendation. arxiv preprint URL https://arxiv.org/abs/2304.00686.

Li, X., Sun, A., Zhao, M., Yu, J., Zhu, K., Jin, D., et al. (2023). Multi-intention oriented contrastive learning for sequential recommendation. In *Proceedings of the sixteenth ACM international conference on web search and data mining* (pp. 411–419).

Li, X., Thickstun, J., Gulrajani, I., Liang, P. S., & Hashimoto, T. B. (2022). Diffusion-lm improves controllable text generation. *Advances in Neural Information Processing Systems*, *35*, 4328–4343.

Li, J., Wang, M., Li, J., Fu, J., Shen, X., Shang, J., et al. (2023). Text is all you need: Learning language representations for sequential recommendation. arXiv preprint URL https://arxiv.org/abs/2305.13731.

Li, P., Wang, Z., Ren, Z., Bing, L., & Lam, W. (2017). Neural rating regression with abstractive tips generation for recommendation. In N. Kando, T. Sakai, H. Joho, H. Li, A. P. de Vries, & R. W. White (Eds.), *Proceedings of the 40th international ACM SIGIR conference on research and development in information retrieval* (pp. 345–354). ACM, http://dx.doi.org/10.1145/3077136.3080822.

Li, Z., Wang, X., Yang, C., Yao, L., McAuley, J., & Xu, G. (2023). Exploiting explicit and implicit item relationships for session-based recommendation. In *Proceedings of the sixteenth ACM international conference on web search and data mining* (pp. 553–561).

Li, Z., Xie, Y., Zhang, W. E., Wang, P., Zou, L., Li, F., et al. (2024). Disentangle interest trend and diversity for sequential recommendation. *Information Processing & Management*, *61*(3), Article 103619.

Li, Z., Xu, X., Tang, Z., Zou, L., Wang, Q., & Li, C. (2024). Spectral and geometric spaces representation regularization for multi-modal sequential recommendation. In *Proceedings of the 33rd ACM international conference on information and knowledge management* (pp. 1336–1345).

Li, Z., Yang, C., Chen, Y., Wang, X., Chen, H., Xu, G., et al. (2024). Graph and sequential neural networks in session-based recommendation: A survey. *ACM Computing Surveys*, *57*(2), 1–37.

Li, L., Zhang, Y., & Chen, L. (2021). Personalized transformer for explainable recommendation. In *Proceedings of the 59th annual meeting of the association for computational linguistics and the 11th international joint conference on natural language processing (volume 1: long papers)* (pp. 4947–4957). Online: Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/2021.acl-long.383, URL https://aclanthology.org/2021.acl-long.383.

Li, L., Zhang, Y., & Chen, L. (2023). Personalized prompt learning for explainable recommendation. *ACM Transactions on Information Systems*, *41*(4), 1–26.

Liao, J., Li, S., Yang, Z., Wu, J., Yuan, Y., Wang, X., et al. (2024). Llara: Large language-recommendation assistant. In *Proceedings of the 47th international ACM SIGIR conference on research and development in information retrieval* (pp. 1785–1795).

Lin, J., Men, R., Yang, A., Zhou, C., Zhang, Y., Wang, P., et al. (2021). M6: Multi-modality-to-multi-modality multitask mega-transformer for unified pretraining. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining* (pp. 3251–3261).

Lin, J., Shan, R., Zhu, C., Du, K., Chen, B., Quan, S., et al. (2023). ReLLa: Retrieval-enhanced large language models for lifelong sequential behavior comprehension in recommendation. arxiv preprint URL https://arxiv.org/abs/2308.11131.

Liu, X., Ji, K., Fu, Y., Tam, W., Du, Z., Yang, Z., et al. (2022). P-tuning: Prompt tuning can be comparable to fine-tuning across scales and tasks. In *Proceedings of the 60th annual meeting of the association for computational linguistics (volume 2: short papers)* (pp. 61–68).

Liu, B., Li, D., Wang, J., Wang, Z., Li, B., & Zeng, C. (2024). Integrating user short-term intentions and long-term preferences in heterogeneous hypergraph networks for sequential recommendation. *Information Processing & Management*, *61*(3), Article 103680.

Liu, J., Liu, C., Lv, R., Zhou, K., & Zhang, Y. (2023). Is chatgpt a good recommender? a preliminary study. arxiv preprint URL https://arxiv.org/abs/2304.10149.

Liu, H., Tam, D., Muqeeth, M., Mohta, J., Huang, T., Bansal, M., et al. (2022). Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. *Advances in Neural Information Processing Systems*, *35*, 1950–1965.

Mao, Z., Wang, H., Du, Y., & Wong, K.-f. (2023). UniTRec: A unified text-to-text transformer and joint contrastive learning framework for text-based recommendation. arXiv preprint URL https://arxiv.org/abs/2305.15756.

Ni, J., Li, J., & McAuley, J. (2019). Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing* (pp. 188–197). Hong Kong, China: Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/D19-1018, URL https://aclanthology.org/D19-1018.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., et al. (2022). Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, *35*, 27730–27744.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., et al. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, *21*(1), 5485–5551.

Rao, Y., Zhao, W., Zhu, Z., Lu, J., & Zhou, J. (2021). Global filter networks for image classification. *Advances in Neural Information Processing Systems*, *34*, 980–993.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., & Ommer, B. (2022). High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 10684–10695).

Shani, G., Heckerman, D., Brafman, R. I., & Boutilier, C. (2005). An MDP-based recommender system. *Journal of Machine Learning Research*, *6*(9).

Sun, F., Liu, J., Wu, J., Pei, C., Lin, X., Ou, W., et al. (2019). BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer. In W. Zhu, D. Tao, X. Cheng, P. Cui, E. A. Rundensteiner, D. Carmel, Q. He, J. X. Yu (Eds.), *Proceedings of the 28th ACM international conference on information and knowledge management* (pp. 1441–1450). ACM, http://dx.doi.org/10.1145/3357384.3357895.

Tang, J., & Wang, K. (2018). Personalized top-N sequential recommendation via convolutional sequence embedding. In Y. Chang, C. Zhai, Y. Liu, & Y. Maarek (Eds.), *Proceedings of the eleventh ACM international conference on web search and data mining* (pp. 565–573). ACM, http://dx.doi.org/10.1145/3159652.3159656.

Tong, X., Wang, P., Li, C., Xia, L., & Niu, S. (2021). Pattern-enhanced contrastive policy learning network for sequential recommendation. In *IJCAI* (pp. 1593–1599).

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., et al. (2023). Llama: Open and efficient foundation language models. arxiv preprint URL https://arxiv.org/abs/2302.13971.

Van Loan, C. (1992). *Computational frameworks for the fast Fourier transform*. SIAM.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., et al. (2017). Attention is all you need. In I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, R. Garnett (Eds.), *Advances in neural information processing systems 30: annual conference on neural information processing systems 2017* (pp. 5998–6008). URL https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

Wang, T., & Isola, P. (2020). Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *Proceedings of machine learning research: vol. 119*, *Proceedings of the 37th international conference on machine learning* (pp. 9929–9939). PMLR, URL http://proceedings.mlr.press/v119/wang20k.html.

Wang, L., & Lim, E.-P. (2023). Zero-shot next-item recommendation using large pretrained language models. arxiv preprint URL https://arxiv.org/abs/2304.03153.

Wang, H., Liu, X., Fan, W., Zhao, X., Kini, V., Yadav, D., et al. (2024). Rethinking large language model architectures for sequential recommendations. arXiv preprint arXiv:2402.09543.

Wang, W., Xu, Y., Feng, F., Lin, X., He, X., & Chua, T.-S. (2023). Diffusion recommender model. In *Proceedings of the 46th international ACM SIGIR conference on research and development in information retrieval* (pp. 832–841).

Wang, J., Yuan, F., Cheng, M., Jose, J. M., Yu, C., Kong, B., et al. (2022). TransRec: Learning transferable recommendation from mixture-of-modality feedback. arXiv preprint URL https://arxiv.org/abs/2206.06190.

Webson, A., & Pavlick, E. (2021). Do prompt-based models really understand the meaning of their prompts?. arxiv preprint URL https://arxiv.org/abs/2109.01247.

Wu, S., Tang, Y., Zhu, Y., Wang, L., Xie, X., & Tan, T. (2019). Session-based recommendation with graph neural networks. In *Proceedings of the AAAI conference on artificial intelligence, vol. 33, no. 01* (pp. 346–353).

Xiao, T., Liang, S., & Meng, Z. (2019). Hierarchical neural variational model for personalized sequential recommendation. In L. Liu, R. W. White, A. Mantrach, F. Silvestri, J. J. McAuley, R. Baeza-Yates, & L. Zia (Eds.), *The world wide web conference* (pp. 3377–3383). ACM, http://dx.doi.org/10.1145/3308558.3313603.

Yang, Y., Huang, C., Xia, L., Huang, C., Luo, D., & Lin, K. (2023). Debiased contrastive learning for sequential recommendation. In *Proceedings of the ACM web conference 2023* (pp. 1063–1073).

Yang, A., Wang, N., Deng, H., & Wang, H. (2021). Explanation as a defense of recommendation. In *Proceedings of the 14th ACM international conference on web search and data mining* (pp. 1029–1037).

Zhang, Y., Chen, X., et al. (2020). Explainable recommendation: A survey and new perspectives. *Foundations and Trends® in Information Retrieval*, *14*(1), 1–101.

Zhang, Y., Ding, H., Shui, Z., Ma, Y., Zou, J., Deoras, A., et al. (2021). Language models as recommender systems: Evaluations and limitations.

Zhang, T., Zhao, P., Liu, Y., Sheng, V. S., Xu, J., Wang, D., et al. (2019). Feature-level deeper self-attention network for sequential recommendation. In S. Kraus (Ed.), *Proceedings of the twenty-eighth international joint conference on artificial intelligence* (pp. 4320–4326). ijcai.org, http://dx.doi.org/10.24963/ijcai.2019/600.

Zhao, Z., Wallace, E., Feng, S., Klein, D., & Singh, S. (2021). Calibrate before use: Improving few-shot performance of language models. In *International conference on machine learning* (pp. 12697–12706). PMLR.

Zhou, K., Wang, H., Zhao, W. X., Zhu, Y., Wang, S., Zhang, F., et al. (2020). S3-rec: Self-supervised learning for sequential recommendation with mutual information maximization. In M. d'Aquin, S. Dietze, C. Hauff, E. Curry, P. Cudré-Mauroux (Eds.), *CIKM '20: the 29th ACM international conference on information and knowledge management* (pp. 1893–1902). ACM, http://dx.doi.org/10.1145/3340531.3411954.

Zhou, K., Yu, H., Zhao, W. X., & Wen, J.-R. (2022). Filter-enhanced MLP is all you need for sequential recommendation. In *Proceedings of the ACM web conference 2022* (pp. 2388–2399).