Information Processing and Management 61 (2024) 103814

Contents lists available at ScienceDirect

Information Processing and Management

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

Non-autoregressive personalized bundle generation

Wenchuan Yang [a], Cheng Yang [b], Jichao Li [a], Yuejin Tan [a], Xin Lu [a,*], Chuan Shi [b]

[a] College of Systems Engineering, National University of Defense Technology, Changsha, 410073, PR China
[b] School of Computer Science, Beijing University of Posts and Telecommunications, Beijing, 100080, PR China

ARTICLE INFO

Keywords:
Personalized bundle generation
Non-autoregressive decoding
Transformer

ABSTRACT

The personalized bundle generation problem, which aims to create a preferred bundle for user from numerous candidate items, receives increasing attention in recommendation. However, existing works ignore the order-invariant nature of the bundle and adopt sequential modeling methods as the solution, which might introduce inductive bias and cause a large latency in prediction. To address this problem, we propose to perform the bundle generation via non-autoregressive mechanism and design a novel encoder–decoder framework named BundleNAT, which can effectively output the targeted bundle in one-shot without relying on any inherent order. In detail, instead of learning sequential dependency, we propose to adopt pre-training techniques and graph neural network to fully embed user-based preference and item-based compatibility information, and use a self-attention based encoder to further extract global dependency pattern. We then design a permutation-equivariant decoding architecture that is able to directly output the desired bundle in a one-shot manner. Experiments on three real-world datasets from Youshu and Netease show the proposed BundleNAT significantly outperforms the current state-of-the-art methods in average by up to 35.92%, 10.97% and 23.67% absolute improvements in Precision, Precision+, and Recall, respectively.

1. Introduction

The recommender system has now proved an effective tool for alleviating the information overload phenomenon in our daily life (Duan, Zhu, Liang, Zhu, & Liu, 2023; Hu, Li, Shi, Yang, & Shao, 2020; Sheng, Zhang, Zhang, & Gao, 2023). Generally, a recommender system predicts whether the target user will be interested in the item or not, and then returns a list of items from a vast candidate pool for potential choices. By doing so, users' demand could be satisfied to the maximum extent. Other than recommending the single item to users, delivering a size-K ($K \geq 2$) item-set named bundle becomes a common practice in online services (Kouki et al., 2019; Li, Bao, Chang, Xu, & Li, 2020; Zhu, Harrington, Li, & Tang, 2014). The bundle has two key characteristics: (1) items should be appealing to target users, and (2) constituent items should be compatible with each other and express the same topic or style. Some typical forms of the bundle are the playlist on Netease music platform, the game collection on Steam platform, and product set on shopping website Taobao.

The advantages of delivering a bundle are easily recognized (Chen, Liu, He, Gao, & Zheng, 2019; Ding, Mok, Ma, & Bin, 2023; Ma, He, Zhang, Wang, & Chua, 2022; Pathak, Gupta, & McAuley, 2017; Yang, Li, Tan, Tan, & Lu, 2023; Zhang, Du, & Tong, 2022). For users, item bundling could better tailor to the needs by helping users find surprisingly interested items. For service providers, recommending bundles could expose more items to users. Particularly, in online e-commerce, with an attractive discount rate, the bundling strategy could even possibly increase sales (Liu, Fu, Chen, Xiong, & Chen, 2017; Sun, Li, & Teo, 2021).

* Corresponding author.
E-mail addresses: xin.lu.lab@outlook.com (X. Lu), shichuan@bupt.edu.cn (C. Shi).

https://doi.org/10.1016/j.ipm.2024.103814
Received 24 November 2023; Received in revised form 12 March 2024; Accepted 15 June 2024

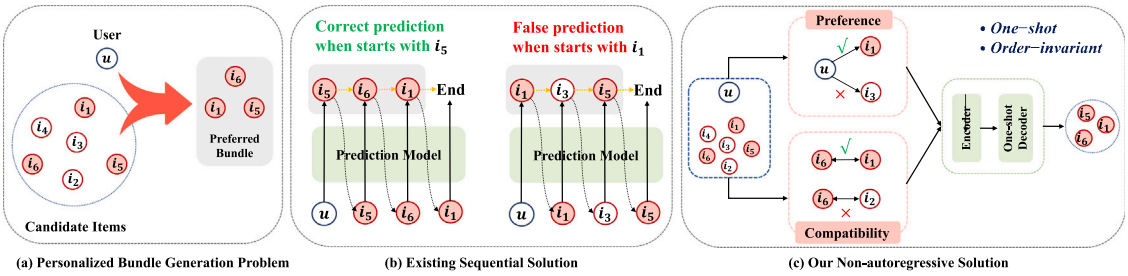


Fig. 1. A toy example illustrating personalized bundle generation. Fig. 1a: Given user u and candidate items $i_1, \ldots, i_6$, the goal is to generate the preferred bundle consists of i_1, i_5, and i_6. Fig. 1b shows the 4-step generating process based on sequence modeling, which is unaware of multiple optimal sequential orders and might result in inference failure when changing the sequential order. Fig. 1c illustrates the proposed non-autoregressive generation which aims to output the size-3 bundle in 1 step by utilizing preference and compatibility information (An encoder–decoder architecture is adopted in our paper).

There exist two lines of research on recommending bundles, one called pre-built bundle recommendation is in line with top-K recommendation which aims to rank the most likely preferred bundle (already existed) for users. The other called personalized bundle generation task is to investigate how to select items from the candidate set to composite suitable bundles for users as shown in Fig. 1a, which is the focus of our paper.

Although a decent portion of the literature focuses on the pre-built bundle recommendation (Chang, Gao, He, Jin, & Li, 2021; Chen et al., 2019; Ma et al., 2022; Vijaikumar, Shevade, & Murty, 2021; Zhang et al., 2022), relatively little effort has been made to push forward the development of bundle generation techniques, leaving a large space waiting for exploration. Existing works (Deng et al., 2021; Gong et al., 2019) generally model the bundle as a sequence and perform the generating process in an autoregressive manner, in which the items are selected one by one. And they propose to utilize sequential methods like PointerNet (Vinyals, Fortunato, & Jaitly, 2015) or reinforcement learning techniques (Schulman, Wolski, Dhariwal, Radford, & Klimov, 2017) to resolve it. Despite their effectiveness, we suggest that sequential modeling is still insufficient to find the optimal bundle. The reason is that it ignores the order-invariant property embedded in the bundle.

To be more specific, we detail the generation process based on sequential modeling in Fig. 1b. To search for the targeted $\{i_1, i_5, i_6\}$ bundle, it follows a step-by-step manner in which i_6 follows i_5, i_1 selected right after i_6, and $\{i_5 - i_6 - i_1\}$ is assumed to be the best ordering to recovering the bundle. However, there are $3!$ equally good solutions for the size-3 bundle (Sui, Zeng, Chen, Liu, & Zhao, 2023; Zaheer et al., 2017; Zhang, Hare, & Prugel-Bennett, 2019), when we place i_1 in the first place, the model could be confused since no dependency information is available, as i_1 is the last predicted item before. Hence, relying on a certain ordering might introduce unnecessary inductive bias and harm the generalization ability of the model. Besides, it requires 4 times inference for a size-3 bundle indicating the large latency in sequential prediction.

Aware of these setbacks, in this study, the bundle is structured into a set and we propose to perform the bundle generation via non-autoregressive mechanism which attempts to output the desired items in one-shot. Though the non-autoregressive solution seems like a straightforward approach as depicted in Fig. 1c, the problem remains highly non-trivial with the following unique challenges:

- How to make the generation process aware of the compatibility? Different from the sequential modeling in which compatibility could be promised by conditioning on the preceding ones, the non-autoregressive generation process is a one-shot process, consequently, we need to find a way to ensure the item compatibility globally.
- How to design a proper decoding approach that is equivariant to the permutations of constituent items? The main difficulty of non-autoregressive decoding lies in the property that items within the bundle are freely interchangeable. Imposing an inherent order (Deng et al., 2021; Gong et al., 2019) made it easier, but also can make the prediction highly sensitive to the input order.

To address these challenges, inspired by Non-Autoregressive Transformers (NATs), we here propose a novel encoder–decoder framework[1] for bundle composition named BundleNAT. Our proposed solution efficiently facilitates both user-based preference signal and item-compatibility signal as the overall dependency pattern and pairs it with a bundle-specific non-autoregressive decoding network. In BundleNAT, we first identify two key factors, i.e., user-based preference and compatibility among items, crucial for personalized generation. Inspired by pre-training techniques (Devlin, Chang, Lee, & Toutanova, 2018; Nowakowski, Ptaszynski, Murasaki, & Nieuważny, 2023; Shen, Li, Bouadjenek, Mai, & Sanner, 2023; Zhang, Han et al., 2019), we decide to utilize a conventional recommendation model to learn preference signals based on user–item interactions. For compatibility signals, since there is no ground-truth data on the relation of substitution and complement among items, we propose employing the co-occurrence relation as the approximation and use the graph neural network (GNN) as the extractor to capture compatibility signals. By employing a self-attention based encoding network, the model is able to further learn the global dependency patterns essential

[1] Although the proposed framework is mainly based on existing popular techniques, the novelty of our work lies in the effective combination along with scenario-specific modifications tailored for non-autoregressive generation. The empirical results further demonstrate the effectiveness and efficiency of proposed BundleNAT

for decoding. Secondly, inspired by NAT studies (Gu, Bradbury, Xiong, Li, & Socher, 2018; Huang, Tao, Zhou, Li, & Huang, 2022), we propose to adopt a non-autoregressive decoding mechanism, which is independent of specific ordering, to recover the bundle. However, simply applying the vanilla non-autoregressive decoding mechanism to the bundle generation scenario is not feasible. The reason is that the *multi-modality* issue (Jiang et al., 2021; Ma, Shao, Gui, Zhang, & Feng, 2023; Niwa, Takase, & Okazaki, 2023; Zhan et al., 2022), which refers to the inability to identify multiple equal combinations of items owing to the parallel output, is not solved and will result in poor generation in our scenario. To tackle the problem, we then propose a permutation-equivariant decoding network to alleviate this issue, meanwhile improving the performance by retrieving a base from the encoder via a novel copy mechanism. Lastly, considering the order-invariant feature of bundle, order-agnostic cross entropy (Du, Tu, & Jiang, 2021) is applied to guide the model training. Extensive experiments conducted on three real-world datasets validate the superiority of the proposed BundleNAT against state-of-the-art methods and the significance of the designed modules.

The salient points of this work are:

- We take the first step to formulate the bundle generation task via the non-autoregressive mechanism, which is a more proper way to facilitate the order-invariance property of the bundle.
- We propose a novel non-autoregressive mechanism featured framework for personalized bundle generation, which can capture both the preference and compatibility pattern. We further design a bundle-specific decoding process combined with a copy mechanism to recover the targeted bundle.
- The experimental results demonstrate the effectiveness and efficiency of our proposed framework over existing state-of-the-art, achieving average 35.92%, 10.97%, and 23.67% absolute performance gain on Precision, Precision+, and Recall against the second-best baseline, respectively.

2. Related work

In this section, we briefly review the existing relevant literature in the personalized bundle generation domain.

2.1. Bundle generation

The core idea of bundle generation is to develop a model that is capable of picking up a desired set of compatible items from a vast candidate pool. Early works mainly adopt utility function and statistical methods to infer the target bundle. Xie, Lakshmanan, and Wood (2014) proposed a linear additive utility function based on implicit user feedback to generate personalized bundles, however, the function is unable to model complex dependencies among items. Ge, Zhang, Qian, and Yuan (2017) have studied the bundling strategy on e-commerce platforms and found that items with more reviews, particularly those with photos attached, are more likely to be included in the bundle. Liu et al. (2017) designed a probabilistic model named BPM to learn composition factors based on users' buying motives, and the preferred bundle is generated via finding complementary items concerning the target item.

Later, Pathak et al. (2017) built a greedy generation strategy upon the Bayesian Pairwise Ranking (BPR) (Rendle, Freudenthaler, Gantner, & Schmidt-Thieme, 2012) framework. The strategy first trained a bundle preference model, then the best bundle is picked from the candidate pool containing random variants of the initial bundle based on preference score. Vijaikumar et al. (2021) developed a monotone submodular function for scoring generated bundles and applied a greedy algorithm to approximate the optimality. However, both the generation process follows a pure heuristic way, which might result in unaffordable solving time cost in real-world practice.

Recently, Gong et al. (2019) interpreted the generation task as a maximal clique optimization problem on an item–item graph. It factorized the problem in a sequential modeling manner and then combined the reinforcement learning (RL) mechanism with an encoder–decoder framework to perform generation. In line with Gong et al. (2019), Deng et al. (2021) presented the problem as a multi-step Markov Decision Process and proposed a pure RL framework. The method first constructed a user–item preference and item compatibility model, then facilitated corresponding feedback signals along with the evaluation metrics as the reward function following curriculum training methodology. In a closely related work, Wei, Liu, Yang, Wang, and Zheng (2022) examined the superiority of non-autoregressive decoding manner and applied it to the bundle creative generation scenario, which takes heterogeneous objects into account. The method utilized a standard non-autoregressive encoder–decoder architecture in the NLP field, and considered an extra contrastive learning objective to further ensure generation quality.

However, our work is different from theirs in several main points:

Firstly, we are fully aware of the order-invariant nature of bundles and propose a novel non-autoregressive transformer architecture as the solution. While Bai et al. (2019), Deng et al. (2021), Gong et al. (2019) identifying the bundle as a sequence and utilizing sequential modeling methods as the solution, the generation performance will be affected by the contradiction between the order-invariant nature of the bundle and presumed optimal sequential order. Besides, the sequential methods generally suffer from long-term bottleneck and inference latency.

Secondly, we propose a proper way to encode the intrinsic compatibility among items into generating a feasible bundle, which is either ignored or insufficiently encoded in existing works. For example, in Deng et al. (2021), each bundle is seen as a sentence, and word2vec (Mikolov, Chen, Corrado, & Dean, 2013) is applied to obtain item compatibility, however, the bundle has no strict order like sentences and context-based learning only captures local dependency while items within the bundle are globally correlated.

Thirdly, as a similar task, bundle creative generation (Wei et al., 2022) aims to find a set of heterogeneous items that satisfy users' preference to the utmost. To improve the generation efficiency, the vanilla non-autoregressive architecture is directly adopted. In this

study, the motivation comes from the order-invariant nature of the bundle rather than seeking superior time efficiency. Meanwhile, the non-autoregressive decoding method used in Wei et al. (2022) is not applicable in bundle generation due to the ignorance of item compatibility and the multi-modality issue.

Lastly, we experiment with our model on real-world bundle recommendation datasets which truly demonstrate the characteristics of the bundle. In previous works, ground-truth bundle data is normally sampled from users' historical purchases in which totally irrelevant items have a great probability curated as a whole, generally containing lots of noisy information (Tzaban et al., 2020). The experimental results are more reliable to show the effectiveness of the architecture design.

2.2. Bundle completion

There is another line of work focusing on the personalized bundle generation task, in which the generation is essentially interpreted as a completion process. Thus, the goal of these studies switched to finding the next suitable item based on the incomplete existing bundle. For example, Bai et al. (2019) worked on a bundle-list recommendation problem and proposed a bundle generation network based on Long Short-Term Memory (LSTM) (Hochreiter & Schmidhuber, 1997) and Determinantal Point Process (Chen, Zhang, & Zhou, 2018) selection to find high-quality and diversified bundles. Chang et al. (2021) proposed to treat the bundle as a sparse-connected item graph and designed a GNN-based model named BGGN to predict both the right nodes and edges in the graph. Jeon, Jang, Kim, and Kang (2023) simplified bundle generation as the recovering process from the incomplete one and designed a neural network to learn the user-bundle (incomplete) pair embedding so that the rest of the items could be correctly retrieved.

Methods for bundle completion are excluded from further discussion in this paper, due to the totally different problem settings and incompatible hypothesis.

2.3. Non-autoregressive transformer

Non-Autoregressive Transformers (NATs) (Bin et al., 2023, 2022; Ding et al., 2020; Guo et al., 2021; Wang, Zhang, & Chen, 2018) have recently emerged as a powerful and popular class of text generation models due to the low decoding latency. Different from the sequential decoding strategy adopted by vanilla Autoregressive Transformers (ATs), NATs can generate the target sentence in parallel, which means each token can be predicted independently. Therefore, the decoding speed is significantly improved. However, the lack of token dependencies in decoding damages the performance of NATs when compared with ATs (Gu et al., 2018; Huang et al., 2022; Xiao et al., 2023).

To alleviate this problem while maintaining decoding efficiency, various advanced approaches have been proposed. For example, knowledge distillation (Ding et al., 2020, 2021; Liao, Jiang, Li, Wang, & Wang, 2023; Liu, Bao, Zhao & Huang, 2023; Shao, Wu, & Feng, 2022) is employed to supervise NATs with distilled target sentences generated by a pre-trained AT model, so that the quality of training corpus is improved. NAT-IR (Lee, Mansimov, & Cho, 2018) first introduced the refinement mechanism to balance the decoding speed and generation accuracy in NATs. The main idea is to refine the output of vanilla NATs or noisy target sentence iteratively (Ghazvininejad, Levy, Liu, & Zettlemoyer, 2019; Lu, Meng, & Peng, 2022; Ran, Lin, Li, & Zhou, 2021; Savinov, Chung, Binkowski, Elsen, & van den Oord, 2021) to improve the accuracy. Besides, several researchers have noticed that the traditional cross-entropy loss is not optimal (Du, Tu, Wang & Jiang, 2022; Li, Cui, Yin, & Zhang, 2022) for the NATs due to its sensitivity to strict alignment. For example, Du et al. (2021) proposed the order-agnostic cross-entropy (OAXE) loss to guide the NATs to focus on lexical matching rather than order errors of the prediction. Bin et al. (2023) designed an exclusive loss to alleviate the repetition issue in sentence ordering, which combines the loss for sentence choosing and position choosing to ensure the exclusiveness of optimal matching between positions and sentences. Moreover, various methods started to leverage the useful information embedded in pre-trained models to enhance the NAT models (Liao, Wang, & Wang, 2024; Shen, Bao, Gao, Zhou, & Zhao, 2024; Wang, He, Chen, Chen, & Jiang, 2022). For example, Wei, Wang, Zhou, Lin, and Sun (2019) used a well-trained AT model to supervise the decoding state of the NAT model. Guo, Tan et al. (2020) proposed to use curriculum learning along with an AT model to fine-tune the decoder of the NAT. AB-Net (Guo, Zhang et al., 2020) adopted two different BERT models as the encoder and decoder in non-autoregressive machine translation. Kim et al. (2023) proposed a novel architecture BiLD which employs a small autoregressive decoder to generate text and a large non-autoregressive decoder to refine the predictions.

Despite the empirical successes, the non-autoregressive generation and NATs are rarely discussed and explored in bundle generation scenario, since the bundle in nature is order-agnostic.

3. Problem formulation

Given a candidate items set $I = \{v_1, v_2, \dots, v_n\}$ with n stands for the set size, the goal of personalized bundle generation is to find the specific item-set $B_u = \{v_j \mid j = 1, 2, \dots, K, v_j \in I\}$, so that B_u is most likely the correct and preferred combination of items for each user u. We therefore derive the problem formulation as follows:

$$\max \sum_u P\left(B_u \mid u, I\right), \tag{1}$$

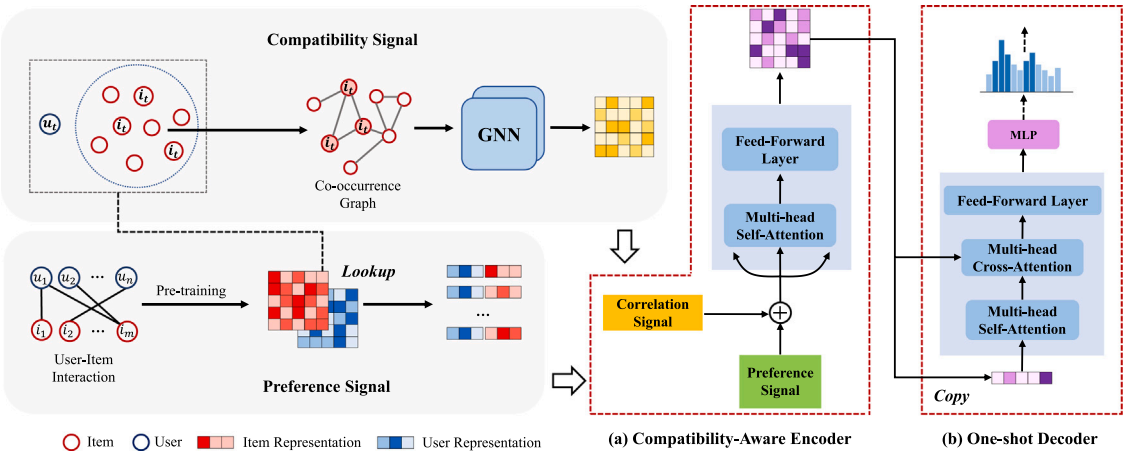


Fig. 2. The overall architecture of the proposed framework. u_t stands for the target user, the items within the ground-truth bundle are marked by i_t.

where $P\left(B_u \mid u, I\right)$ measures the probability of generated B_u being satisfied. However, simply conditioning on user preference cannot ensure the rationality of the generated bundle, we introduce item compatibility information to further supervise the generation process:

$$\max \sum_{u} P\left(B_u \mid u, I, O\right), \tag{2}$$

where O represents the compatibility pattern.

Inspired by the order invariant property of the bundle, we can therefore model the task in non-autoregressive manner, in which the sequential order is no longer needed for generating. We should be able to find the optimal bundle by predicting an item set from the candidate pool in one-shot.

Thus, the probability of inducing a bundle could be factorized as follows.

$$P\left(B_u \mid u, I, O\right) = \sum_{j=1, v_j \in I}^{K} p\left(v_j \mid u, O\right), \tag{3}$$

Relying on this factorization, the entire generation process is fully aware of users' preference and items' compatibility. Meanwhile, the formulation is order-invariant since the summing operation is commutative, the prediction remains constant no matter how the order changes.

4. Proposed framework

In this section, more details are presented regarding the proposed BundleNAT framework. We first give an overview of the framework, and then we further clarify the design of two main modules of NAT, i.e., compatibility-aware encoder and one-shot decoder.

4.1. Overview

The overall architecture of the proposed framework is shown in Fig. 2. The BundleNAT falls into two main parts: compatibility-aware encoder and one-shot decoder. For the encoder, we have introduced two kinds of signal to learn the needed global dependency information, i.e., the preference signal and the item compatibility signal. We propose to capture item compatibility signal from the co-occurrence graph to strengthen the intra-relatedness within the bundle, meanwhile, the preference signal extracted from user–item interactions is utilized to ensure each item retrieved is appealing to the user. Then a self-attention based encoding network is employed to further learn the global dependency pattern. When designing the decoding strategy, inspired by vanilla NAT framework, we revise the original decoder architecture from parallel decoding into a one-shot decoding manner. Besides, a copy mechanism is proposed to guide the decoding process towards the optimal solution and an MLP-based projection module is added to enable the model to directly output the item set from the predicted distribution. In the end, we utilize order-agnostic cross entropy as the training loss to guide model optimization.

4.2. Compatibility-aware encoder

Preference signal. The generated bundle should well satisfy users' need, which means each item within the bundle is favorable. Given the candidate item-set I and user u, we facilitate the preference signal $\boldsymbol{P}$ to ensure the personalized property, which is derived as follows:

$$\begin{aligned} &\boldsymbol{P} = [p_1, p_2, \dots, p_n] \\ &\text{with } p_i = e_{v_i} \oplus e_u, \end{aligned} \tag{4}$$

where $p_i \in \boldsymbol{R}^d$ represents the preference signal embedded in $u - v_i$ pair, $e_{v_i} \in I$, e_u denote feature vectors for item v_i and user u respectively, $\oplus$ refers to concatenation operation. Here, e_{v_i}, e_u are learned by matrix factorization model optimized by BPR (Rendle et al., 2012) based on user–item interaction data. Consequently, p_i can migrate the user's preference into item selection for the bundle. Notably, the choice of preference model is flexible, for example, Wide-and-Deep (Cheng et al., 2016) and LightGCN (He et al., 2020) can also be leveraged to learn e_{v_i}, e_u.

Compatibility signal. The key challenge in non-autoregressive bundle generation is how to guarantee the rationality of the selected item-set, in other words, constitute items should be compatible. Selection only relies on preference could possibly result in meaningless bundle in which incompatible items could appear together. Here, we first construct a co-occurrence graph G that implies the item compatibility pattern. Specifically, for each item v_i in the candidate set I, there is an occurrence frequency vector $f_{v_i} \in R^{N_b}$ retrieved from ground-truth bundle-item affiliations where N_b denotes the number of bundles. Specifically, G is built as follows,

$$\begin{aligned} \boldsymbol{G} &= \boldsymbol{D}^{-1/2} \cdot \left(\boldsymbol{F} \cdot \boldsymbol{F}^T\right) \cdot \boldsymbol{D}^{-1/2}, \\ \boldsymbol{F} &= \left[f_{v_1}, f_{v_2}, \dots, f_{v_n}\right]^T, \boldsymbol{G} \in \boldsymbol{R}^{n \times n}, \end{aligned} \tag{5}$$

where $\boldsymbol{D}$ is the degree matrix of $\boldsymbol{F} \cdot \boldsymbol{F}^T$ utilized for normalization, $g_{ij} \in G$ shows the compatibility degree between item v_i and item v_j. Thus, we adopt the GNN model to transform the degree information into vectorized representation $c_{v_i} \in R^d$,

$$c_{v_i} = \sum_{v \in N(v_i)} GNN_G\left(v_i\right), \tag{6}$$

where $N(v_i)$ stands for the neighbor set of item v_i including itself. The detailed learning mechanism is expressed as,

$$\begin{aligned} c_{v_i}^{(k)} &= \sigma\left(W_{(k)}\left(c_{v_i}^{(k-1)} + \mathrm{Agg}\left(c_v^{(k-1)}\right)\right) + b_{(k)}\right), \\ c_{v_i}{}^{(0)} &= z_{v_i}, \end{aligned} \tag{7}$$

where k means the kth propagation layer, $W_{(k)}$, and $b_{(k)}$ are the corresponding learnable weight matrix and bias, respectively. $c_{v_i}^{(k-1)}$ stands for the preceding learned representation of v_i, $c_v^{(k)}$ is the representation of v_i's neighboring nodes, Agg stands for the weighted aggregation operation, with the weight decided by g_{ij}. z_{v_i} denotes the initial trainable node feature of v_i.

The input. Positional encoding is one of the most successful designs in Transformer architecture that help the encoder network aware of the position information of tokens, thus enhancing the performance in various language tasks (Liu et al., 2019; Radford et al., 2019; Vaswani et al., 2017). Here, we employ the Absolute-Positional-Encoding-style (APE-style) input as the backbone, in which the compatibility signal is seen as the positional encoding for the candidate items. Although simple, APE enjoys a great property proved by Luo et al. (2022), Yun, Bhojanapalli, Rawat, Reddi, and Kumar (2019) that Transformer architecture with APE is capable of approximating any continuous sequence-to-sequence function in a compact domain. Meanwhile, Relative positional encoding fails in this theorem because the positional information will be suppressed into stochastic signals when placed in softmax exponentiation. Thus, in the proposed BundleNAT framework, we derive the input for encoding as $\boldsymbol{X} \in \boldsymbol{R}^{n \times d} = \boldsymbol{P} + \boldsymbol{C}$, where $\mathbf{C} = \left[c_{v_1}, c_{v_2}, \dots, c_{v_n}\right]^T$.

Encoding network. We adopt an attention-based encoding network to learn a solid dependency pattern for subsequent decoding. The basic unit of the encoding network consists of two kinds of layers: a self-attention layer $Attention$ followed by a point-wise feed-forward layer FFN. Given the input $\boldsymbol{X}$, the workflow of a unit is defined as follow,

$$\begin{aligned} \boldsymbol{X_A} = Attention(\boldsymbol{X}) &= softmax\left(\frac{\left(W_Q \boldsymbol{X}\right) \cdot \left(W_K \boldsymbol{X}\right)^T}{\sqrt{d}}\right) \cdot \left(W_V \boldsymbol{X}\right), \\ \boldsymbol{X_F} = FFN(\boldsymbol{X}) &= \boldsymbol{X_A} + \sigma(\boldsymbol{X_A} W_1) W_2, \end{aligned} \tag{8}$$

where W_Q, W_K and W_V in self-attention layer are learnable projection vectors, d stands for the embedding dimension of $\boldsymbol{X}$. W_1 and W_2 are learnable weight matrices in feed-forward layer, with activation function σ set to be $ReLu$ function. Here, multi-head attention is further adopted to capture different dependency patterns among the candidate set, those learned patterns will then be concatenated as the output,

$$\begin{aligned} \boldsymbol{X_A} = Multihead\ Atention(\boldsymbol{X}) &= (head_1 \oplus head_2 \oplus \cdots \oplus head_k) \cdot W_M, \\ head_i = softmax\left(\frac{\left(W_Q^i \boldsymbol{X}\right) \cdot \left(W_K^i \boldsymbol{X}\right)^T}{\sqrt{d}}\right) &\cdot \left(W_V^i \boldsymbol{X}\right), \end{aligned} \tag{9}$$

where W_Q^i, W_K^i and W_V^i are learnable projection vectors for ith head, W_M denotes the parameterized transformation matrix, $\oplus$ stands for concatenation.

4.3. One-shot decoder

Non-autoregressive decoding has received lots of attention in the NLP field in recent years (Ren et al., 2020; Xiao et al., 2023), different from auto-regressive decoding (or sequential decoding), non-autoregressive decoding aims to output tokens parallel so that the inference latency is largely reduced. In this study, we suggest a sequential order is not necessary for the generation by recognizing the order-invariant nature of the bundle. Therefore non-autoregressive decoding mechanism becomes a natural option for bundle composition. However, directly utilizing the vanilla non-autoregressive decoding from NLP is not feasible resulting from two challenges: (1) multi-modality, referring to the phenomenon that parallel output is unaware of the multi-combination distribution of target bundle. For example, $\{1, 3, 5, 7\}$ along with $\{5, 3, 1, 7\}$ represents the same bundle, for parallel decoding it might lead to $\{5, 3, 5, 7\}$ due to independence of multi-channel prediction; (2) the performance degradation attributes to the removal of sequential dependency in prediction.

To deal with the challenging issue, we here revise the vanilla NAT architecture and propose a one-shot decoder with a copy mechanism to directly output the desired item set.

The input. As demonstrated by Niwa et al. (2023), the poor performance of NAT could resulted from the from-scratch decoding process, which means if the decoder starts from low-quality input, it is difficult to reach the demanded output even through several iterations. Besides utilizing the learned global dependency pattern to guide the decoding, we suggest a high-quality "start" is also vital for further improving the performance and generating the satisfied bundle effectively. Subsequently, we propose to copy the output $\boldsymbol{X_F} \in R^{n\times d}$ from the encoder via a mean pooling function, to get abstract global dependency feature, which serves as the start point for decoding,

$$h_m = \frac{1}{|P_m|}\sum_{(i,j)\in P_m} x_{ij}, m = 1,2,\dots,d, \tag{10}$$

where $x_{ij} \in \boldsymbol{X_F}$, P_m denotes the pooling area to learn h_m which belongs to $\boldsymbol{R}^{n\times 1}$, and we have $d \cdot P_m \in R^{n\times d}$. $|\cdot|$ indicates the number of elements in the area, and the input for the decoder is formulated as $\boldsymbol{h} = \{h_m \mid m = 1, 2, \dots, d\}, \boldsymbol{h} \in R^{1\times d}$.

Notably, with a single feature vector $\boldsymbol{h} \in R^{1\times d}$ as input, we transform the vanilla NAT architecture into a one-shot decoder. By reducing the input to $R^{1\times d}$, the advantage is twofold: firstly, the decoder utilizes a single channel to predict the item set rather than multiple channels (Zhang et al., 2019), thus the repetition issue brought by the independence of multiple predictions could be alleviated; secondly, it enables the decoder to work for variable-size bundles, while the existing method (Wei et al., 2022) needs to alter the number of prediction channels once the bundle size changed.

Decoding network. The basic block of proposed decoder contains three kinds of layers: one-token self-attention layer, cross-attention layer, and feed-forward layer. Specifically, the one-token self-attention layer is derived as follows:

$$\boldsymbol{h'} = Attention(\boldsymbol{h}) = softmax\Big(\frac{(W_Q^{d1}\boldsymbol{h})\cdot(W_K^{d1}\boldsymbol{h})^T}{\sqrt{d}}\Big)\cdot(W_V^{d1}\boldsymbol{h}), \tag{11}$$

where $W_*^{d1} \in R^{d\times d}$ is the parameterized transformation vector. The cross-attention layer is defined based on the interaction between encoder output and decoder input, the intuition is to suppress the learned fine-grained dependency information into decoding process,

$$\boldsymbol{h''} = CrossAttention(\boldsymbol{h'}) = softmax\Big(\frac{\left(W_Q^{d2}\boldsymbol{X_F}\right)\cdot(W_K^{d2}\boldsymbol{h'})^T}{\sqrt{d}}\Big)\cdot\left(W_V^{d2}\boldsymbol{h'}\right), \tag{12}$$

where $W_*^{d2} \in R^{d\times d}$ is the trainable parameter matrix, and the learned compatibility $\boldsymbol{X_F}$ is served as the *query* vector. $\boldsymbol{h''} \in R^{1\times d}$ is then fed into a feed-forward layer, and obtains the output representation $\boldsymbol{h_d}$:

$$\boldsymbol{h_d} = \boldsymbol{h''} + \sigma(\boldsymbol{h''}W_1^{d3})W_2^{d3}, \tag{13}$$

where $W_*^{d3} \in R^{d\times d}$ denotes the parameterized weight matrix.

Notably, multi-head attention mechanism is also utilized in both one-token self-attention layer and cross-attention layer.

Prediction. We further add an MLP-based network module to project back to the full-size item-set and return the predicted distribution, consequently, we can pick the desired item directly instead of inferring the item position (Wei et al., 2022).

$$\boldsymbol{h}_o = \sigma(\boldsymbol{h}_d W_o + b_o), \tag{14}$$

where W_o is a learnable parameter matrix whose dimension is defined as $d \times N$, N denotes the number of all items, b_o is the bias factor, $\boldsymbol{h}_o \in \boldsymbol{R}^{1\times N}$ is the predicted distribution for final inference.

Table 1
The detailed information for datasets.

	Youshu	Netease
# Users	8,006	2,532
# Bundles	4,771	5,586
# Items	32,770	51,298
# User–item Interactions	138,515	160,318
# User-bundle Interactions	51,377	75,536
# Bundle-item Affiliations	176,667	317,608
Datasets for bundle generation		
Youshu (K = 5, N = 100)	51,377 instances	
Youshu (K = 20, N = 200)	45,115 instances	
Netease (K = 5, N = 100)	75,536 instances	

4.4. Model optimization

For inference, we use *argmax* to produce the size-K bundle based on $\boldsymbol{h}_o = \{h_o^1, h_o^2, \ldots, h_o^N\}$. For loss calculation, cross-entropy loss is a natural choice for training the structured prediction problem, however, in bundle generation, the strictly aligned cross entropy would falsely penalize the inference due to the ignorance of order-invariant property. Inspired by Du et al. (2021), we propose a bundle-specific prediction learning objective to guide the model training. To illustrate, the inferred items are denoted as $\tilde{B} = \{v_{b_i} | b_i \in N, |b_i| = k\}$, the corresponding predicted distribution is denoted by $\tilde{b} = \{h_o^{b_i}\}$. We have an ordering space $\boldsymbol{B} = \{\tilde{B}_{(1)}, \tilde{B}_{(2)}, \ldots, \tilde{B}_{(N!)}\}$ which contains $N!$ kinds of permutation of predicted items, the bundle generation objective is formulated as retrieving the best ordering $\tilde{B}_{(i)}$ to minimize the cross-entropy loss,

$$L_{bundle} = arg \min_{\tilde{B}(i) \in \boldsymbol{B}} \left(XE\left(\tilde{B}_{(i)}, Y\right)\right) \tag{15}$$

$$= arg \min_{\tilde{B}(i) \in \boldsymbol{B}} \left(- \sum_{y_i \in Y} y_i log\left(h_o^{b_i}\right) + (1 - y_i) log\left(1 - h_o^{b_i}\right)\right), \tag{16}$$

where $XE(\cdot)$ is the standard cross entropy loss, Y is the ground-truth bundle, y_i represents each label belongs to Y. Specifically, to search for the best ordering $\tilde{B}_{(i)}$ efficiently, we leverage the Hungarian Matching algorithm (Kuhn, 1955).

5. Experiments

In this section, we detail the experimental settings and present empirical results to demonstrate the effectiveness of the proposed BundleNAT. The experiments answer the following Research Questions (RQs):

- **RQ1**: Does the proposed BundleNAT yield better generation performance compared with existing state-of-the-art?
- **RQ2**: How do different components (*e.g.*, compatibility signal extraction) contribute to the performance of BundleNAT?
- **RQ3**: How well is the efficiency of BundleNAT for bundle generation?
- **RQ4**: How do different hyper-parameters (*e.g.*, the depth of encoding/decoding network) affect the performance of BundleNAT?

5.1. Datasets

The evaluation datasets for the experiment are constructed based on two popular real-world bundle recommendation datasets, and the detailed statistics of original and constructed datasets are stated in Table 1.

- **Youshu**. The dataset is collected by Chen et al. (2019) from Youshu, a book-review website in China. Every bundle is a list of books that users might be interested in. To obtain the standard dataset for evaluation, we first create the ground-truth recommended size-K bundles by randomly sampling K items from each interacted bundle for a user, ensuring the size-K bundles a user interacted with are unique. Then we construct a candidate set by pairing $M - K$ items with each size-K bundle for a user, where the $M - K$ items are uniformly sampled from the whole items set. At last, we develop two datasets: (1) size-5 bundle along with $M = 100$ as the size of the candidate set and (2) size-20 bundle with $M = 200$ as the size of the candidate set. We name the above two datasets as Youshu (K = 5, M = 100) and Youshu (K = 20, M = 200).
- **Netease**. The dataset is crawled from Netease (Cao et al., 2017), a music streaming platform in which playlists containing various songs are seen as bundles. Due to a larger data size compared to the Youshu dataset, we first perform a sampling process to obtain a relatively smaller dataset. Here, each bundle consists of at least 10 items and each item appears in at least 15 bundles. Further, each user in the dataset should be interacted with at least 15 bundles and 15 items. And following the same processing procedure as Youshu, we correspondingly construct a dataset called Netease (K = 5, M = 100).

5.2. Experimental settings

Evaluation protocol. We perform an 80/20 split to construct the training and testing set for all datasets in line with Deng et al. (2021), Gong et al. (2019). To evaluate the performance of proposed model, three bundle-specific metrics are utilized (Deng et al., 2021):

$$Precision@K = \frac{1}{|Y|}\sum_{i}^{|Y|} I(v_{\tilde{B}_i}^{(0)} = v_{Y_i}^{(0)}), \tag{17}$$

$Precision@K$ demonstrates whether the predicted next-item is the same as the one in the ground-truth bundle. Here, $Y = \{Y_i\}, \widetilde{B} = \{\widetilde{B}_i\}$ are the set of ground-truth and generated bundles, respectively. $v_*^{(0)}$ denotes the first-place item in the bundle. $|\cdot|$ returns the size of the set, with $|Y| = |\widetilde{B}|$. $I(\cdot)$ is the indicator function and K indicates the size of bundle.

$$Precision^{+}@K = \frac{1}{|Y|}\sum_{i=1}^{|Y|} I(v_{\tilde{B}_i}^{*} \in Y_i), \tag{18}$$

$Precision^{+}@K$ measures where there is an overlap between the generated bundle and the ground-truth bundle. $v_{\tilde{B}_i}^{*}$ stands for arbitrary item in the generated bundle $\tilde{B}_i$.

$$Recall@K = \frac{1}{|Y|}\sum_{i=1}^{|Y|} \frac{\tilde{B}_i \cap Y_i}{K}, \tag{19}$$

$Recall@K$ indicates how many predicted items of $\tilde{B}_i$ are in the ground-truth bundle Y_i. Compared with $Precision^{+}@K$, $Recall@K$ is a more specific measure and gives a more prominent illustration on the generation quality.

Baselines. We compare the proposed BundleNAT[2] with several competitive models in the experiments.

- POP (Jeon et al., 2023): It chooses the top-k popular items.
- BPR (Rendle et al., 2012): It is a well-known and effective pairwise ranking model, which is learned by optimizing the user-item pairwise ranking loss under the matrix factorization framework. The bundle here is generated based on top-K ranked user-item pairs from the candidate set.
- NCF (He et al., 2017): A competitive collaborative filtering method that combines the neural network architecture and traditional matrix factorization model to capture the non-linear interactions between users and items. We use the model to predict top-K items that user is most likely to interact with as the bundle for each user.
- UltraGCN (Mao et al., 2021): It is the ultra simplified GCN model for collaborative filtering, which further removes message passing on the basis of LightGCN and uses a constraint loss to approximate infinite-layer graph convolutions.
- SASRec (Kang & McAuley, 2018): It is a sequential recommendation model based on self-attention mechanism which is capable of capturing long-range dependency so that high-quality prediction can be made when learning from user's historical behaviors. To recover the size-K bundle, we employ the model to take turns to predict K items.
- Exact-k (Gong et al., 2019): This is the state-of-the-art model for building bundles which tries to find the maximal clique (bundle) in the graph composed by candidate items by employing an autoregressive-style encoder–decoder framework along with the demonstration mechanism from reinforcement learning.
- BYOB (Deng et al., 2021): It is a reinforcement-learning-based method recent proposed for bundle generation task, which tries to generate the personalized item set for users through the combination of proximal policy optimization and curriculum learning mechanism.

BGGN (Chang et al., 2021) and BGN (Bai et al., 2019) mentioned in related works are excluded from the discussions due to the totally incompatible problem definition. Specifically, BGN is defined as a sequential recommendation task, the next bundle generation is merely based on historical user-bundle interactions; BGGN in fact treats the bundle generation as the graph completion task where items within a bundle are assumed to be sparsely related. We also do not include pre-built recommendation methods (Ma et al., 2022; Zhang et al., 2022) as baselines, which focus on finding the top-k bundles rather than generating a bundle from scratch.

Implementation details. We choose Adaptive Moment Estimation (Adam) (Kingma & Ba, 2014) to optimize our BundleNAT. The dimension of the preference signal is set to be 64, and we use a 2-layer GNN to extract compatibility signal with an embedding size 128. Both encoding and decoding network is fixed to 2-block depth. Specifically, when utilizing a MF-BPR model for learning preference signal, a leave-one-out strategy (He, Zhang, Kan, & Chua, 2016; Petrov & Macdonald, 2022; Sun et al., 2019) is utilized to split user–item interactions for training. Grid search is adopted to find the best hyperparameter: the learning rate is searched within $\{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$, the dropout ratio is tuned amongst $\{0.0, 0.1, \ldots, 0.4\}$ and the coefficient of weight decay is in $\{10^{-5}, 10^{-4}, \ldots, 10^{-1}\}$. All the experiments were conducted on a server with an AMD EPYC 7402 CPU and an NVIDIA GeForce RTX 3090 24G GPU. The server was running Ubuntu 22.04 with PyTorch[3] v1.9 and Python v3.6.

[2] The code is available at https://github.com/Chuan1997/BundleNAT

[3] https://pytorch.org/

Table 2
Performance comparison on three datasets.

Method	Youshu (K = 5, M = 100)			Youshu (K = 20, M = 200)			Netease (K = 5, M = 100)		
	Precision@5	Precision+@5	Recall@5	Precision@20	Precision+@20	Recall@20	Precision@5	Precision+@5	Recall@5
POP	0.1000	0.8511	0.3949	0.0361	0.9917	0.5047	0.0493	0.6336	0.2046
BPR	0.4851	0.8313	0.3733	0.6425	0.9876	0.4745	0.2343	0.5691	0.1771
NCF	0.4943	0.8391	0.3770	0.6476	0.9862	0.4787	0.2946	0.6572	0.2297
UltraGCN	0.1039	0.8649	0.4025	0.0357	0.9907	0.4977	0.0639	0.6831	0.2449
SASRec	0.4998	0.7884	0.2886	0.5875	0.9870	0.3578	0.2255	0.4920	0.1283
BYOB	0.4700	0.7900	0.3042	0.6363	0.9797	0.1934	0.0595	0.4536	0.1070
Exact-k	0.5500	0.8929	0.4307	0.7199	0.9918	0.5490	0.2565	0.6890	0.2142
BundleNAT	**0.8091**	**0.9582**	**0.5843**	**0.9779**	**0.9996**	**0.7566**	**0.8551**	**0.9451**	**0.5937**
Improved	0.2591	0.0653	0.1536	0.2580	0.0078	0.2076	0.5605	0.2561	0.3488

5.3. Overall performance comparison (RQ1)

We have tested the performance of the proposed BundleNAT against competitive baselines on three datasets, and the overall result is summarized in Table 2. The best and second-best results are emphasized by **bold** and underlined fonts. We have observed the following findings:

Based on the empirical results, it can be seen that our method exceeds all the baselines by a large margin significantly. In detail, we can observe that absolute improvements in terms of Precision@5, Precision+@5, and Recall@5 are 25.91%, 6.53%, and 15.36% against the second-best model, respectively, on Youshu datasets. As for the Netease dataset, the proposed BundleNAT still acquires a stably great performance, while the baselines apparently have difficulty in recovering the best bundle, with the absolute performance gains are 56.05%/25.61%/34.88% on Precision@5/Precision+@5/Recall@5. Notably, existing baselines can gain a close performance regarding the Precision+@K metric, however, the poor performance on Recall@K metrics demonstrates the inferior ability to generate the exact bundle. The superiority of BundleNAT could be attributed to several factors: (1) BundleNAT employs the one-shot decoding manner which is fully aware of the order-invariant property; (2) BundleNAT effectively encodes the global dependency by incorporating the compatibility and preference signal.

It is surprising to find that POP gains a good performance on Precision+@k and Recall@k metric, which reveals that popular items are more likely to be chosen when forming a bundle. Our method outperforms the POP in every case, demonstrating that BundleNAT is able to generate bundles consisting of unpopular items as well as popular items.

BPR and NCF obtain close performance when compared with other specially designed models. The unexpected results suggest that the preference signal is informative for selecting the right item to composite the bundle, however, the significant performance gap indicates preference is not enough for high-quality generation.

Though UltraGCN shows remarkable performance in retrieving the desired bundle, e.g., obtaining the second-best performance on Recall@5 for the Netease dataset, its poor performance on the Precision metric indicates its ineffectiveness in finding the first appealing item in a bundle.

As shown in the Table 2, SASRec loses its superiority when applied to generate a bundle. A possible reason is that the inference of SASRec mainly focuses on modeling the sequential dependency between historical behaviors and the next interaction, while in bundle generation scenario we need to predict the next K interactions. SASRec is not capable of capturing the dependency among the K interactions, therefore resulting in poor performance in size-K bundle generation.

As far as we can see, Exact-k is the strongest competitor and obtains consistent good performance among all three datasets. When choosing an item, Exact-k takes the relations attended to preceding items into account via an attention-based module, consequently, the quality of the bundle is ensured. Meanwhile, BYOB generally falls behind the Exact-K on all the evaluation metrics, sometimes worse than other baselines. The performance degradation could be attributed to several reasons: (1) item compatibility information from the dataset is also leveraged to generate the bundle, however, the learning module based on word2vec is insufficient to capture a global dependency pattern due to fixed context size, consequently, the reinforcement learning framework is guided by a flawed signal; (2) the characteristics of real-world datasets are different from the synthetic ones for original BYOB evaluation, the specific-designed training order in curriculum learning could possibly fail in real-world bundle generation scenario.

5.4. Ablation study (RQ2)

We further conduct experiments on Netease (K = 5, M = 100) dataset to investigate the utility of various components in BundleNAT. To better demonstrate their effectiveness, we develop several ablated models for comparison. Table 3 shows the performance of the default BundleNAT and ablated models.

Only with random feature (o/w random for short). The o/w random model refers to removing the preference and compatibility signal in the encoder, and only randomly initialized feature is used for generation. As shown in the Table 3, the o/w random model is generally the worst variant, which is not surprising since no informative characteristics are used for decoding.

Only with preference signal (o/w preference for short). The variant keeps preference signal and removes compatibility signal for encoding, thus the items within the bundle are selected based on users' preference. It can be observed from the Table 3 that the

Table 3
Performance comparison on Netease (K = 5, M = 100) regarding major designs.

Method	Precision@5		Precision+@5		Recall@5	
o/w random	0.4610	−46.09%	0.6165	−34.77%	0.2415	−59.32%
o/w preference	0.4908	−42.6%	0.6431	−31.95%	0.2699	−54.54%
o/w compatibility	0.8342	−2.44%	0.9291	−1.69%	0.5359	−9.74%
w/o copy	0.7221	−15.55%	0.8123	−14.05%	0.3257	−45.14%
r/w max-pooling	0.8387	−1.92%	0.9416	−0.37%	0.5709	−3.99%
r/w LapPE	0.4742	−44.54%	0.6314	−33.19%	0.2582	−56.51%
r/w RWSE	0.4497	−47.41%	0.5814	−38.48%	0.2311	−61.07%
r/w RoPE	0.4978	−41.78%	0.6360	−32.7%	0.2726	−54.08%
r/w CE	0.8474	−0.9%	0.9341	−1.16%	0.5799	−3.84%
BundleNAT	**0.8551**		**0.9451**		**0.5937**	

utilization of preference signal can indeed find a better combination of items compared with o/w random variant, but still falls largely behind the default design, indicating that focus merely on preference is insufficient for bundle generation.

Only with compatibility signal (o/w compatibility for short). We combine the compatibility signal with the random initialized preference signal to formulate the encoder input. As we can see from the experimental result, the leverage of compatibility signal largely enhances the generation performance compared with o/w preference variant, demonstrating the necessity of properly encoding the item compatibility towards accurate bundle composition.

Replace with various positional encodings. To investigate the effectiveness of the proposed compatibility signal learning module, we conduct comparisons between existing state-of-the-art strategies for encoding positional information. Laplacian positional encoding (LapPE) and Random-walk structural encoding (RWSE) (Liu, Cantürk et al., 2023; Müller, Galkin, Morris, & Rampášek, 2023; Rampášek, Galkin, Dwivedi, Luu, Wolf, & Beaini, 2022) are the two most prevalent methods for incorporating structure bias into graph-based transformers. Rotary position embedding (RoPE) (Su et al., 2024) is widely used in nowadays large language models (Du, Qian et al., 2022; Touvron et al., 2023) for sequence-based positional encoding. We apply these three strategies on the constructed co-occurrence graph and the experimental results are shown in Table 3. We observe that LapPE and RWSE cannot capture the dependency pattern among items well. Among them, RoPE obtains the best performance but still largely falls behind our method. The reason might be that RoPE is originally designed for strictly ordered and structural text, while the candidate set for bundle generation is out-of-order. Thus, it is difficult to retrieve dependency information without sequential order.

Without copy mechanism (w/o copy for short). Here, we remove the copy mechanism designed for decoder and the input of decoder is replaced by a randomly initialized trigger embedding (Wei et al., 2022). Compared with the complete BundleNAT design, w/o copy variant experiences 15.55%, 14.05%, and 45.14% performance drop on Precision@5, Precision+@5, and Recall@5, respectively. It validates that a relatively high-quality start for decoding process is empirically beneficial, while from-scratch decoding has difficulty in recovering desired bundle.

Replace with max-pooling (r/w max-pooling for short). We replace the pooling function embedded in the copy mechanism with max-pooling method to obtain the decoder input. We can conclude from the result that r/w max-pooling is a good choice but still significantly inferior to the original design with regard to Precision@5 and Recall@5. Compared with the mean-pooling strategy, max-pooling tends to capture the most prominent local dependency which might cause global information loss.

Replace with cross-entropy loss (r/w CE for short). We replace the order-invariant loss with the traditional cross-entropy loss for model optimization. It can be seen in the table that cross-entropy loss is inferior to order-agnostic one and is not an optimal choice for non-autoregressive generation, since it will penalize a correct prediction with a different order.

5.5. Empirical analysis on time efficiency (RQ3)

In this section, we compare the proposed BundleNAT with the existing bundle-specific methods Exact-k and BYOB in terms of training time efficiency and inference latency. Fig. 3 and Table 4 show the total training time and inference latency comparison, respectively, among the three models denoted by the multiplier.

In detail, we set the time cost and inference latency of BundleNAT as the base, Youshu-5, Youshu-20, and Netease-5 are the abbreviation for the three datasets.

In Fig. 3, we can observe that, when the target bundle size is 5 (Youshu-5, Netease-5), the second-best Exact-k is on average 7.99 times slower than our method. As the target bundle size grows, i.e., increases to 20, the time cost of Exact-k experiences a large increase due to the nonlinear expansion of search space, which becomes 32.273 times slower than BundleNAT. In addition, we can also find that, BundleNAT shows higher time efficiency than BYOB with average 2.03 times faster in overall runtime, although BYOB adopts a parallel speed-up computation framework (Moritz et al., 2018) to speed up the RL-based inference process.

As we can see from Table 4, the BundleNAT fully exploits the inference advantage from non-autoregressive modeling, and the time spent in generating a bundle is significantly faster than Exact-k and BYOB. Besides, we observe that the proposed BundleNAT maintains the inference efficiency with the varying size of candidate sets. The autoregressive model Exact-k has experienced a sharp increase as the size of candidate set and bundle becomes larger.

The empirical results demonstrate the superior generation efficiency of the proposed non-autoregressive mechanism.

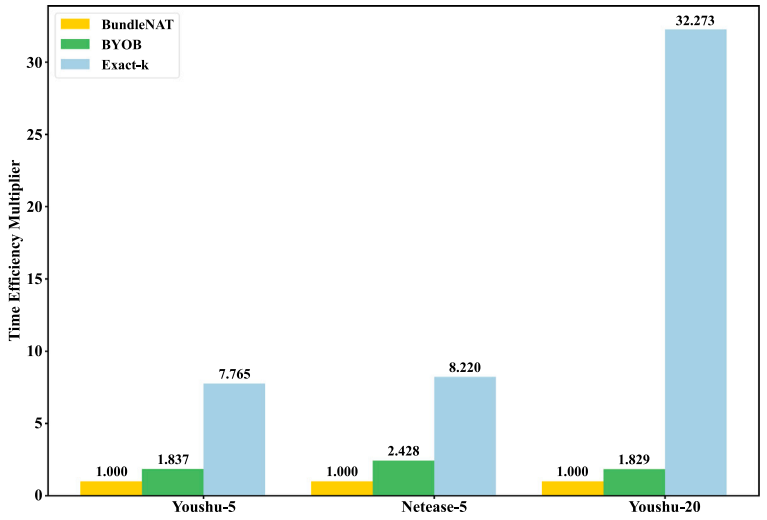
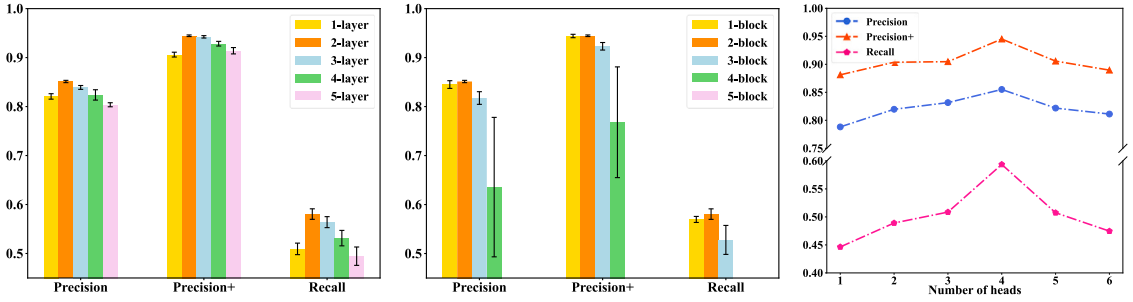


Fig. 3. Overall training time comparison on all the three datasets. We compare the proposed BundleNAT with bundle-specific methods, i.e., Exact-k and BYOB.

Table 4
Inference latency (per bundle) comparison of autoregressive and non-autoregressive models. **ratio** denotes the accelerating ratio comparing BundleNAT to Exact-k and BYOB.

Dataset	BundleNAT	Exact-k		BYOB	
	time	time	ratio	time	ratio
Youshu (K = 5, M = 100)	4.58 (10^{-5} s)	8.52 (10^{-4} s)	18.60×	4.97 (10^{-3} s)	108.41×
Youshu (K = 20, M = 200)	5.24 (10^{-5} s)	6.62 (10^{-3} s)	126.47×	5.94 (10^{-3} s)	113.40×
Netease (K = 5, M = 100)	4.94 (10^{-5} s)	8.14 (10^{-4} s)	16.47×	3.52 (10^{-3} s)	71.29×



(a) Effect of the *Num.* of GNN layer (b) Effect of the *Num.* of block (c) Effect of the *Num.* of heads

Fig. 4. Influence of hyper-parameters on Netease (K = 5, N = 100) dataset.

5.6. Parameters analysis (RQ4)

In this subsection, we perform sensitivity analysis of some important hyper-parameters in BundleNAT, including the number of layers of GNN to extract compatibility signal, the depth of encoding and decoding network, and the number of heads in the self-attention layer. We run 5 trials for each parameter during the analysis and report the averaged results in Fig. 4.

The number of layers. The number of GNN layers is searched in $\{1, 2, 3, 4, 5\}$. The corresponding results are presented in Fig. 4(a). We can observe a 2-layer GNN is able to well retrieve higher-order signals and obtain a robust compatibility pattern with a relatively smaller standard deviation. As the number of GNN layers increases, the performance experiences a continuous drop resulted from the over-smoothing issue. It also can be seen that a 1-layer GNN is insufficient to capture the compatibility signal with only one-time information propagation.

The depth of encoding/decoding network. The depth refers to the number of encoding/decoding units in the Transformer architecture, which is searched from $\{1, 2, 3, 4, 5\}$. Fig. 4(b) shows the empirical results. The BundleNAT obtains consistent performance gains as the depth increases, however, when the depth reaches a relatively larger number, i.e., greater than 2, the model starts to fall behind demonstrating that increased model complexity has no positive impact on model performance. When the number

of encoding/decoding units is 2, it shows the greatest generation performance with the smallest deviation which illustrates the robustness of encoder–decoder architecture with depth 2.

The number of heads. Here, we investigate the impact of the number of heads in the self-attention module and tune among $\{1, 2, 3, 4, 5, 6\}$. As we can see from Fig. 4(c), single-head is not a preferred choice and results in poor performance without identifying various dependency patterns. Generally, the performance increases as the number of heads enlarges, while too many heads might begin to bring noisy information as the performance starts to fall when exceeding 4-head attention.

6. Conclusion

6.1. Research implications

In this paper, we focus on the personalized bundle generation problem which aims to find the optimal bundle for users over a set of candidate items. Different from the previous study, we highlight the order-invariant property of the bundle and suggest following a sequential order is not suitable for generating the bundle resulting from inductive bias. We take the first step to formulate the bundle generation task via the non-autoregressive manner, and identify the corresponding challenges. To tackle this specific problem, we propose a novel encoder–decoder framework named BundleNAT. Specifically, we first design a self-attention based network to encode both preference signal and compatibility signal. Then we propose a non-autoregressive decoder to predict the targeted bundle in one-shot. We further propose a copy mechanism that facilitates the encoded pattern as the initial state of the decoder to ensure the generation towards the optimal solution.

Extensive experiments on three real-world datasets demonstrate the effectiveness of the proposed BundleNAT, as BundleNAT significantly outperforms the existing state-of-the-art methods by a large margin. After time efficiency comparison, it can be seen that BundleNAT also shows significant advantages in generation efficiency.

BundleNAT is closely related to the product bundling strategy in modern marketing and has many real-world applications. For example, on e-commerce platforms, it can select a set of appealing and compatible items from numerous candidate items for users efficiently and accurately based on historical purchases. It can also be effective on content platforms like music-streaming platform, it can deliver a satisfying playlist for listeners based on their likes. Besides, BundleNAT can perfectly fit with the real-time services in real-world practices due to the speed advantage brought by the non-autoregressive decoding.

6.2. Future work

In future work, we aim to explore a new compatibility signal learning module which is able to consider heterogeneous and dynamic correlations among items, so that we can learn a more comprehensive dependency pattern. Moreover, since the candidate set could be much larger in real-world scenarios, we would like to investigate the efficient deployment of BundleNAT for online billion-scale bundle generation.

CRediT authorship contribution statement

Wenchuan Yang: Writing – original draft, Software, Methodology, Conceptualization. **Cheng Yang:** Writing – review & editing, Methodology. **Jichao Li:** Data curation. **Yuejin Tan:** Funding acquisition, Validation. **Xin Lu:** Resources, Supervision. **Chuan Shi:** Supervision, Writing – review & editing.

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.

Data availability

Data will be made available on request.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (72025405, 72088101, 72001211, 72371244, 72301285), the National Social Science Foundation of China (22ZDA102), the Hunan Science and Technology Plan Project (2020TP1013, 2020JJ4673, 2023JJ40685), the Innovation Team Project of Colleges in Guangdong Province (2020KCXTD040), and the Science Foundation for Outstanding Youth Scholars of Hunan Province (2022JJ20047).

References

Bai, J., Zhou, C., Song, J., Qu, X., An, W., Li, Z., et al. (2019). Personalized bundle list recommendation. In *The world wide web conference* (pp. 60–71).

Bin, Y., Han, M., Shi, W., Wang, L., Yang, Y., & Shen, H. T. (2023). Non-autoregressive math word problem solver with unified tree structure. arXiv preprint arXiv:2305.04556.

Bin, Y., Shi, W., Zhang, J., Ding, Y., Yang, Y., & Shen, H. T. (2022). Non-autoregressive cross-modal coherence modelling. In *Proceedings of the 30th ACM international conference on multimedia* (pp. 3253–3261).

Cao, D., Nie, L., He, X., Wei, X., Zhu, S., & Chua, T.-S. (2017). Embedding factorization models for jointly recommending items and user generated lists. In *Proceedings of the 40th international ACM SIGIR conference on research and development in information retrieval* (pp. 585–594).

Chang, J., Gao, C., He, X., Jin, D., & Li, Y. (2021). Bundle recommendation and generation with graph neural networks. *IEEE Transactions on Knowledge and Data Engineering*.

Chen, L., Liu, Y., He, X., Gao, L., & Zheng, Z. (2019). Matching user with item set: Collaborative bundle recommendation with deep attention network. In *International joint conference on artificial intelligence* (pp. 2095–2101).

Chen, L., Zhang, G., & Zhou, E. (2018). Fast greedy map inference for determinantal point process to improve recommendation diversity. *Advances in Neural Information Processing Systems*, *31*.

Cheng, H.-T., Koc, L., Harmsen, J., Shaked, T., Chandra, T., Aradhye, H., et al. (2016). Wide & deep learning for recommender systems. In *Proceedings of the 1st workshop on deep learning for recommender systems* (pp. 7–10).

Deng, Q., Wang, K., Zhao, M., Wu, R., Ding, Y., Zou, Z., et al. (2021). Build your own bundle-a neural combinatorial optimization method. In *Proceedings of the 29th ACM international conference on multimedia* (pp. 2625–2633).

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

Ding, Y., Mok, P., Ma, Y., & Bin, Y. (2023). Personalized fashion outfit generation with user coordination preference learning. *Information Processing & Management*, *60*(5), Article 103434.

Ding, L., Wang, L., Liu, X., Wong, D. F., Tao, D., & Tu, Z. (2020). Understanding and improving lexical choice in non-autoregressive translation. In *International conference on learning representations*.

Ding, L., Wang, L., Liu, X., Wong, D. F., Tao, D., & Tu, Z. (2021). Rejuvenating low-frequency words: Making the most of parallel data in non-autoregressive translation. 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. 3431–3441).

Du, Z., Qian, Y., Liu, X., Ding, M., Qiu, J., Yang, Z., et al. (2022). GLM: General language model pretraining with autoregressive blank infilling. In *Proceedings of the 60th annual meeting of the association for computational linguistics (volume 1: long papers)* (pp. 320–335).

Du, C., Tu, Z., & Jiang, J. (2021). Order-agnostic cross entropy for non-autoregressive machine translation. In *International conference on machine learning* (pp. 2849–2859). PMLR.

Du, C., Tu, Z., Wang, L., & Jiang, J. (2022). Ngram-OAXE: Phrase-based order-agnostic cross entropy for non-autoregressive machine translation. In *Proceedings of the 29th international conference on computational linguistics* (pp. 5035–5045).

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.

Ge, X., Zhang, Y., Qian, Y., & Yuan, H. (2017). Effects of product characteristics on the bundling strategy implemented by recommendation systems. In *2017 international conference on service systems and service management* (pp. 1–6). IEEE.

Ghazvininejad, M., Levy, O., Liu, Y., & Zettlemoyer, L. (2019). Mask-predict: Parallel decoding of conditional masked language models. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing* (pp. 6112–6121).

Gong, Y., Zhu, Y., Duan, L., Liu, Q., Guan, Z., Sun, F., et al. (2019). Exact-k recommendation via maximal clique optimization. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining* (pp. 617–626).

Gu, J., Bradbury, J., Xiong, C., Li, V., & Socher, R. (2018). Non-autoregressive neural machine translation. In *International conference on learning representations*.

Guo, J., Tan, X., Xu, L., Qin, T., Chen, E., & Liu, T.-Y. (2020). Fine-tuning by curriculum learning for non-autoregressive neural machine translation. In *Proceedings of the AAAI conference on artificial intelligence*: *vol. 34*, (no. 05), (pp. 7839–7846).

Guo, J., Wang, M., Wei, D., Shang, H., Wang, Y., Li, Z., et al. (2021). Self-distillation mixup training for non-autoregressive neural machine translation. arXiv preprint arXiv:2112.11640.

Guo, J., Zhang, Z., Xu, L., Wei, H.-R., Chen, B., & Chen, E. (2020). Incorporating bert into parallel sequence decoding with adapters. *Advances in Neural Information Processing Systems*, *33*, 10843–10854.

He, X., Deng, K., Wang, X., Li, Y., Zhang, Y., & Wang, M. (2020). Lightgcn: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval* (pp. 639–648).

He, X., Liao, L., Zhang, H., Nie, L., Hu, X., & Chua, T.-S. (2017). Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web* (pp. 173–182).

He, X., Zhang, H., Kan, M.-Y., & Chua, T.-S. (2016). Fast matrix factorization for online recommendation with implicit feedback. In *Proceedings of the 39th international ACM SIGIR conference on research and development in information retrieval* (pp. 549–558).

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735–1780.

Hu, L., Li, C., Shi, C., Yang, C., & Shao, C. (2020). Graph neural news recommendation with long-term and short-term interest modeling. *Information Processing & Management*, *57*(2), Article 102142.

Huang, F., Tao, T., Zhou, H., Li, L., & Huang, M. (2022). On the learning of non-autoregressive transformers. In *International conference on machine learning* (pp. 9356–9376). PMLR.

Jeon, H., Jang, J.-G., Kim, T., & Kang, U. (2023). Accurate bundle matching and generation via multitask learning with partially shared parameters. *Plos one*, *18*(3), Article e0280630.

Jiang, T., Huang, S., Zhang, Z., Wang, D., Zhuang, F., Wei, F., et al. (2021). Improving non-autoregressive generation with mixup training. arXiv preprint arXiv:2110.11115.

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

Kim, S., Mangalam, K., Malik, J., Mahoney, M. W., Gholami, A., & Keutzer, K. (2023). Big little transformer decoder. arXiv preprint arXiv:2302.07863.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

Kouki, P., Fountalis, I., Vasiloglou, N., Yan, N., Ahsan, U., Jadda, K. A., et al. (2019). Product collection recommendation in online retail. In *Proceedings of the 13th ACM conference on recommender systems* (pp. 486–490).

Kuhn, H. W. (1955). The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, *2*(1–2), 83–97.

Lee, J., Mansimov, E., & Cho, K. (2018). Deterministic non-autoregressive neural sequence modeling by iterative refinement. In *Proceedings of the 2018 conference on empirical methods in natural language processing*. Association for Computational Linguistics.

Li, M., Bao, X., Chang, L., Xu, Z., & Li, L. (2020). A survey of researches on personalized bundle recommendation techniques. In *Machine learning for cyber security: third international conference, ML4CS 2020, guangzhou, China, October 8–10, 2020, proceedings, part II 3* (pp. 290–304). Springer.

Li, Y., Cui, L., Yin, Y., & Zhang, Y. (2022). Multi-granularity optimization for non-autoregressive translation. In *Proceedings of the 2022 conference on empirical methods in natural language processing* (pp. 5073–5084).
Liao, Y., Jiang, S., Li, Y., Wang, Y., & Wang, Y. (2023). Self-improvement of non-autoregressive model via sequence-level distillation. In *Proceedings of the 2023 conference on empirical methods in natural language processing* (pp. 14202–14212).
Liao, Y., Wang, Y., & Wang, Y. (2024). Leveraging diverse modeling contexts with collaborating learning for neural machine translation. arXiv preprint arXiv:2402.18428.
Liu, M., Bao, Y., Zhao, C., & Huang, S. (2023). Selective knowledge distillation for non-autoregressive neural machine translation. arXiv preprint arXiv:2303.17910.
Liu, R., Cantürk, S., Lapointe-Gagné, O., Létourneau, V., Wolf, G., Beaini, D., et al. (2023). Graph positional and structural encoder. arXiv preprint arXiv:2307.07107.
Liu, G., Fu, Y., Chen, G., Xiong, H., & Chen, C. (2017). Modeling buying motives for personalized product bundle recommendation. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, *11*(3), 1–26.
Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., et al. (2019). Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692.
Lu, S., Meng, T., & Peng, N. (2022). Insnet: An efficient, flexible, and performant insertion-based text generation model. *Advances in Neural Information Processing Systems*, *35*, 7011–7023.
Luo, S., Li, S., Zheng, S., Liu, T.-Y., Wang, L., & He, D. (2022). Your transformer may not be as powerful as you expect. arXiv preprint arXiv:2205.13401.
Ma, Y., He, Y., Zhang, A., Wang, X., & Chua, T.-S. (2022). CrossCBR: Cross-view contrastive learning for bundle recommendation. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining* (pp. 1233–1241).
Ma, Z., Shao, C., Gui, S., Zhang, M., & Feng, Y. (2023). Fuzzy alignments in directed acyclic graph for non-autoregressive machine translation. arXiv preprint arXiv:2303.06662.
Mao, K., Zhu, J., Xiao, X., Lu, B., Wang, Z., & He, X. (2021). UltraGCN: Ultra simplification of graph convolutional networks for recommendation. In *Proceedings of the 30th ACM international conference on information & knowledge management* (pp. 1253–1262).
Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.
Moritz, P., Nishihara, R., Wang, S., Tumanov, A., Liaw, R., Liang, E., et al. (2018). Ray: A distributed framework for emerging {AI} applications. In *13th {USeNIX} symposium on operating systems design and implementation* (pp. 561–577).
Müller, L., Galkin, M., Morris, C., & Rampášek, L. (2023). Attending to graph transformers. arXiv preprint arXiv:2302.04181.
Niwa, A., Takase, S., & Okazaki, N. (2023). Nearest neighbor non-autoregressive text generation. *Journal of Information Processing*, *31*, 344–352.
Nowakowski, K., Ptaszynski, M., Murasaki, K., & Nieuważny, J. (2023). Adapting multilingual speech representation model for a new, underresourced language through multilingual fine-tuning and continued pretraining. *Information Processing & Management*, *60*(2), Article 103148.
Pathak, A., Gupta, K., & McAuley, J. (2017). Generating and personalizing bundle recommendations on steam. In *Proceedings of the 40th international ACM SIGIR conference on research and development in information retrieval* (pp. 1073–1076).
Petrov, A., & Macdonald, C. (2022). Effective and efficient training for sequential recommendation using recency sampling. In *Proceedings of the 16th ACM conference on recommender systems* (pp. 81–91).
Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. (2019). Language models are unsupervised multitask learners. *OpenAI blog*, *1*(8), 9.
Rampášek, L., Galkin, M., Dwivedi, V. P., Luu, A. T., Wolf, G., & Beaini, D. (2022). Recipe for a general, powerful, scalable graph transformer. *Advances in Neural Information Processing Systems*, *35*, 14501–14515.
Ran, Q., Lin, Y., Li, P., & Zhou, J. (2021). Guiding non-autoregressive neural machine translation decoding with reordering information. In *Proceedings of the AAAI conference on artificial intelligence*: *vol. 35*, (no. 15), (pp. 13727–13735).
Ren, Y., Liu, J., Tan, X., Zhao, Z., Zhao, S., & Liu, T.-Y. (2020). A study of non-autoregressive model for sequence generation. arXiv preprint arXiv:2004.10454.
Rendle, S., Freudenthaler, C., Gantner, Z., & Schmidt-Thieme, L. (2012). BPR: Bayesian personalized ranking from implicit feedback. arXiv preprint arXiv:1205.2618.
Savinov, N., Chung, J., Binkowski, M., Elsen, E., & van den Oord, A. (2021). Step-unrolled denoising autoencoders for text generation. In *International conference on learning representations*.
Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347.
Shao, C., Wu, X., & Feng, Y. (2022). One reference is not enough: Diverse distillation with reference selection for non-autoregressive translation. In *Proceedings of the 2022 conference of the North American chapter of the association for computational linguistics: human language technologies* (pp. 3779–3791).
Shen, Y., Bao, W., Gao, G., Zhou, M., & Zhao, X. (2024). Unsupervised multilingual machine translation with pretrained cross-lingual encoders. *Knowledge-Based Systems*, *284*, Article 111304.
Shen, T., Li, J., Bouadjenek, M. R., Mai, Z., & Sanner, S. (2023). Towards understanding and mitigating unintended biases in language model-driven conversational recommendation. *Information Processing & Management*, *60*(1), Article 103139.
Sheng, Z., Zhang, T., Zhang, Y., & Gao, S. (2023). Enhanced graph neural network for session-based recommendation. *Expert Systems with Applications*, *213*, Article 118887.
Su, J., Ahmed, M., Lu, Y., Pan, S., Bo, W., & Liu, Y. (2024). Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, *568*, Article 127063.
Sui, D., Zeng, X., Chen, Y., Liu, K., & Zhao, J. (2023). Joint entity and relation extraction with set prediction networks. *IEEE Transactions on Neural Networks and Learning Systems*.
Sun, H., Li, X., & Teo, C.-P. (2021). Product bundle recommendation and pricing: How to make it work? Available at SSRN 3874843.
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 *Proceedings of the 28th ACM international conference on information and knowledge management* (pp. 1441–1450).
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 arXiv:2302.13971.
Tzaban, H., Guy, I., Greenstein-Messica, A., Dagan, A., Rokach, L., & Shapira, B. (2020). Product bundle identification using semi-supervised learning. In *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval* (pp. 791–800).
Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., et al. (2017). Attention is all you need. In *Advances in neural information processing systems*: *vol. 30*.
Vijaikumar, M., Shevade, S., & Murty, M. N. (2021). Gram-smot: Top-n personalized bundle recommendation via graph attention mechanism and submodular optimization. In *Machine learning and knowledge discovery in databases: European conference* (pp. 297–313). Springer.
Vinyals, O., Fortunato, M., & Jaitly, N. (2015). Pointer networks. In *Advances in neural information processing systems*: *vol. 28*.
Wang, Y., He, S., Chen, G., Chen, Y., & Jiang, D. (2022). Xlm-d: Decorate cross-lingual pre-training model as non-autoregressive neural machine translation. In *Proceedings of the 2022 conference on empirical methods in natural language processing* (pp. 6934–6946).
Wang, C., Zhang, J., & Chen, H. (2018). Semi-autoregressive neural machine translation. In *Proceedings of the 2018 conference on empirical methods in natural language processing* (pp. 479–488).
Wei, P., Liu, S., Yang, X., Wang, L., & Zheng, B. (2022). Towards personalized bundle creative generation with contrastive non-autoregressive decoding. In *Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval* (pp. 2634–2638).
Wei, B., Wang, M., Zhou, H., Lin, J., & Sun, X. (2019). Imitation learning for non-autoregressive neural machine translation. In *Proceedings of the 57th annual meeting of the association for computational linguistics* (pp. 1304–1312).

Xiao, Y., Wu, L., Guo, J., Li, J., Zhang, M., Qin, T., et al. (2023). A survey on non-autoregressive generation for neural machine translation and beyond. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Xie, M., Lakshmanan, L. V., & Wood, P. T. (2014). Generating top-k packages via preference elicitation. *Proceedings of the VLDB Endowment*, *7*(14), 1941–1952.

Yang, W., Li, J., Tan, S., Tan, Y., & Lu, X. (2023). A heterogeneous graph neural network model for list recommendation. *Knowledge-Based Systems*, Article 110822.

Yun, C., Bhojanapalli, S., Rawat, A. S., Reddi, S. J., & Kumar, S. (2019). Are transformers universal approximators of sequence-to-sequence functions? arXiv preprint arXiv:1912.10077.

Zaheer, M., Kottur, S., Ravanbakhsh, S., Poczos, B., Salakhutdinov, R. R., & Smola, A. J. (2017). Deep sets. In *Advances in neural information processing systems*: *vol. 30*.

Zhan, J., Chen, Q., Chen, B., Wang, W., Bai, Y., & Gao, Y. (2022). Non-autoregressive translation with dependency-aware decoder. arXiv preprint arXiv:2203.16266.

Zhang, Z., Du, B., & Tong, H. (2022). SuGeR: A subgraph-based graph convolutional network method for bundle recommendation. In *Proceedings of the 31st ACM international conference on information & knowledge management* (pp. 4712–4716).

Zhang, Z., Han, X., Liu, Z., Jiang, X., Sun, M., & Liu, Q. (2019). ERNIE: Enhanced language representation with informative entities. arXiv preprint arXiv:1905.07129.

Zhang, Y., Hare, J., & Prugel-Bennett, A. (2019). Deep set prediction networks. *Advances in Neural Information Processing Systems*, *32*.

Zhu, T., Harrington, P., Li, J., & Tang, L. (2014). Bundle recommendation in ecommerce. In *Proceedings of the 37th international ACM SIGIR conference on research & development in information retrieval* (pp. 657–666).