Auto Encoding Neural Process for Multi-interest Recommendation

Yiheng Jiang[1], **Yuanbo Xu**[1,*], **Yongjian Yang**[1], **Funing Yang**[1], **Pengyang Wang**[2], **Chaozhuo Li**[3]
[1] Lab of Mobile Intelligent Computing, College of Computer Science and Technology, Jilin University
[2] Department of Computer and Information Science, The State Key Laboratory of Internet of Things for Smart City, University of Macau
[3] Beijing University of Aeronautics and Astronautics
jiangyh22@mails.jlu.edu.cn, yuanbox@jlu.edu.cn, yyj@jlu.edu.cn, yfn@jlu.edu.cn, pywang@um.edu.mo, lichaozhuo1991@gmail.com

Abstract

Multi-interest recommendation constantly aspires to an oracle individual preference modeling approach, one that satisfies the *diverse* and *dynamic* properties. Fueled by deep learning technologies, existing neural network (NN)-based recommender systems employ single-point or multi-point interest representation strategies for preference modeling, and boost the recommendation performance with remarkable margins. However, as the parameterized approximate function nature, NN-based methods remain deficiencies regarding the adaptability towards distinctive preference patterns cross different users, and the calibration over individual current intents. In this paper, we revisit multi-interest recommendation with the lens of stochastic process and Bayesian inference. Specifically, we propose to learn a distribution over functions to depict the individual diverse preferences rather than a unified function to approximate preference. Subsequently, the recommendation is equipped with the uncertainty estimation which conforms to the dynamic shifting intent. Along these lines, we establish the connection between multi-interest recommendation and neural processes by proposing **NP-Rec**, which realizes the flexible multiple interests modeling and uncertainty estimation, simultaneously. Empirical study on 4 real world datasets demonstrates that our NP-Rec attains superior recommendation performances to several state-of-the-art baselines, where the average improvement achieves up to 13.94%.

Code — https://anonymous.4open.science/r/NP-Rec-CF45

Introduction

In the era of information overload, recommender systems (RSs) arise to assist users by profiling preferences, screening irrelevant entries, and suggesting interested content (Jiang et al. 2024a; Xu et al. 2024b). Individual preferences possess the *diverse* property and *dynamic* tendency (Wu et al. 2023; Jiang et al. 2024b). As illustrated in Figure 1, the interest scope of a user comprises electronics and furniture, which alternatively generate the historical user-item interaction sequence. The dynamic nature lies in the observation that the user's current intent is shifting over time.

A necessary prerequisite for personalized recommendation is unraveling complex preference patterns. Fueled by deep learning technologies, modern RSs employ neural networks (NNs) to model preferences. According to the representation manner, existing methods can cast as: (1) Single-point Interest Representation (SIR) mode (Hidasi et al. 2016; Yuan et al. 2019; Kang and McAuley 2018), which utilizes a vector in the item embedding space to represent the interest and recommends according to the affinity regarding items; and (2) Multi-point Interest Representation (MIR) mode (Li et al. 2019; Cen et al. 2020; Zhang et al. 2022), which utilizes multiple (e.g. pre-defined K) vectors to depict a user's diverse interests in the item embedding space, one for each, respectively. During the recommendation stage, MIR is similar to SIR that makes predictions based on the similarity between interest and item (Wu et al. 2023).

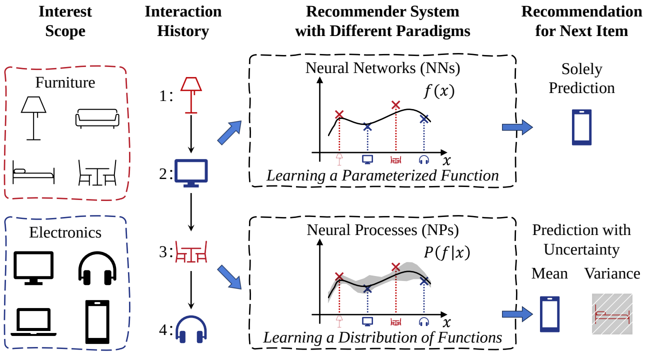

Interest Scope
Interaction History
Recommender System with Different Paradigms
Recommendation for Next Item
Furniture
Electronics
1:
2:
3:
4:
Neural Networks (NNs)
$f(x)$
x
Learning a Parameterized Function
Solely Prediction
Neural Processes (NPs)
$P(f|x)$
x
Learning a Distribution of Functions
Prediction with Uncertainty
Mean
Variance

Figure 1: An illustration of how RSs process the diverse and dynamic nature of a user's preference with different paradigms. NN-based methods focus on learning a preference function while lack the adaptability and uncertainty estimation. NPs, learning a distribution over functions, can generate the predictive mean and variance simultaneously.

Albeit NN-based RSs have achieved satisfactory recommendation performance, either SIR or MIR remains limitations regarding the efficiency, adaptability and uncertainty. Firstly, the limited expressive capability in SIR is hard to cover a user's diverse interests accurately (Zhang et al. 2023); in this view, subsequent methods scale up the embedding dimension whereas at the cost of efficiency inevitably. Secondly, the pre-defined interest quantity (or clustering threshold (Li et al. 2019)) in MIR restricts the adapt-

[*] Corresponding Author: Yuanbo Xu

ability towards distinctive preference patterns cross different users. Thirdly, from the dynamic perspective, NN-based RSs neglect the calibration towards individual current intent. As shown in the upper dashed box of Figure 1, it inherits from the NN essence that *learning a parameterized function to approximate interest representations* (Garnelo et al. 2018a,b), whereas overlooking the functional dynamics, i.e., the interest might change over time. Consequently, the recommendation (top right of Figure 1) is not qualified for the uncertainty estimation, which increases the risk of being overconfident in the current preference.

In this paper, we shift the viewpoint on *learning a distribution over functions, rather than a deterministic preference approximate function, to serve the diverse and dynamic requirements of interest representations.* In this view, we expect the distribution can be adaptive to difference preference patterns and provide the predictive recommendation along with the uncertainty estimation.

Recently, neural latent variable models emerge to perform inference on stochastic processes (Willi et al. 2019). Neural processes (NPs) combine the best of neural networks and Gaussian Processes (GPs) (Garnelo et al. 2018b), which offers an efficient NN-based formulation to approximate a stochastic process. As a probabilistic model, NPs compute a distribution over functions that map inputs to outputs, and use draws to make predictions with uncertainty estimation.

Along these lines, we establish the connection between multi-interest recommendation and NPs by proposing **NP-Rec**. The NP-based recommendation paradigm is depicted in Figure 1. Specifically, we assume the user-item interaction history is generated from a stochastic process, which is defined as a collection of random variables (i.e., preference functions) on a probability space (Arthur, O., and Pittenger 1979). The variables are indexed by the observed interaction at corresponding time steps (i.e., order), and take values in a common measure space. The NP paradigm, as learning a distribution over preferences, naturally meets the adaptability requirement regarding diverse patterns cross users. Credited to the inherent Bayesian inference mechanism, as shown in the lower right of Figure 1, the paradigms provide predictive means with variances, which accords with the dynamic shifting property of individual current interest. Our contributions are summarized as follows:

- To our best knowledge, it is the first work to model the multi-interest representations with the lens of learning a distribution over preference functions.
- We extend neural process to multi-interest recommendation scenario by proposing NP-Rec, which satisfies the diverse and dynamic requirement of interest modeling. Notably, NP-Rec can provide recommendations with uncertainty estimations.
- We validate the proposed algorithm on 4 real-world datasets from 2 benchmarks, and experimental results show that NP-Rec attains superior recommendation performance to several state-of-the-art SIR and MIR methods, where the average improvement achieves up to 13.94%. We verify the effectiveness and extensibility of NP model through an ablation study.

Related Work

Multi-interest Recommendation

Multi-interest recommendation aims at extract preference representation(s) to depict a user's interest scope. Conventional recommender systems (Hidasi et al. 2016; Jiang et al. 2024a; Yuan et al. 2019; Kang and McAuley 2018; Xu et al. 2024a, 2022a) employ the single-point interest representation (SIR) manner to map the preference as a vector in the item embedding space. SIR commonly suffers from the expressive capability issue (Zhang et al. 2023; Zhuo et al. 2024) that limits the accuracy and diversity in item retrieval. In a separate line, MaxMF (Weston, Weiss, and Yee 2013) first proposes the multi-point interest representation (MIR) strategy to represent a user's multiple interests with K vectors. The pioneering MIND (Li et al. 2019) utilizes the dynamic routing mechanism to achieve the interest clustering. Take a step further, ComiRec (Cen et al. 2020) considers the diversity v.s. precision trade-off. PIMI (Chen et al. 2021) incorporates the periodicity and interactivity contained in the user-item interaction history. Zhang et al. (2022) proposes Re4 to reexamine the learned interest representations with explicit regularization. Recently, REMI (Xie et al. 2023) achieves the state-of-the-art recommendation performance by introducing a novel negative sampling strategy and the routing regularization method.

Despite the improvements achieved by these methods, the adaptability to distinctive preference patterns cross users and uncertainty estimation towards the current intent are undergo. Our NP-Rec alleviates the above concerns with the lens of modeling the distribution over preference functions.

Neural Processes Family

Neural Processes (NPs) (Garnelo et al. 2018b) combines the best of neural networks and Gaussian Process (GP), which introduces a neural network-based formulation that learns an approximation of a stochastic process. NPs inherits some fundamental properties from GP, that models distributions over functions and provides uncertainty estimation over predictions conditioned on the context observations (Jha et al. 2022). ANP (Kim et al. 2019) improves the NP fitting ability by introducing the self-attention mechanism (Vaswani et al. 2017). Singh et al. (2019); Willi et al. (2019); Qin et al. (2019) extend NP to the sequential scenarios. Nguyen and Grover (2022) and Bruinsma et al. (2023) exploits the feasibility of adopting auto-regressive inference in NP. Credited to the promising potential, NP demonstrates the capabilities in various domains including set-based representation learning (Zaheer et al. 2017), meta-learning (Ton et al. 2021), Bayesian learning (Hewitt et al. 2018) and generative modeling (Eslami et al. 2018). In recommender systems, TaNP (Lin et al. 2021) assumes each task or user as an instantiation of a stochastic process, and exploits a latent variable structure, a customized module and an adaptive decoder to address the cold-start issue.

To our best knowledge, the proposed NP-Rec in this paper is the first to establish the connection between multi-interest recommendation and NP, which overcomes the adaptability and uncertainty deficiencies in NN-based methods.

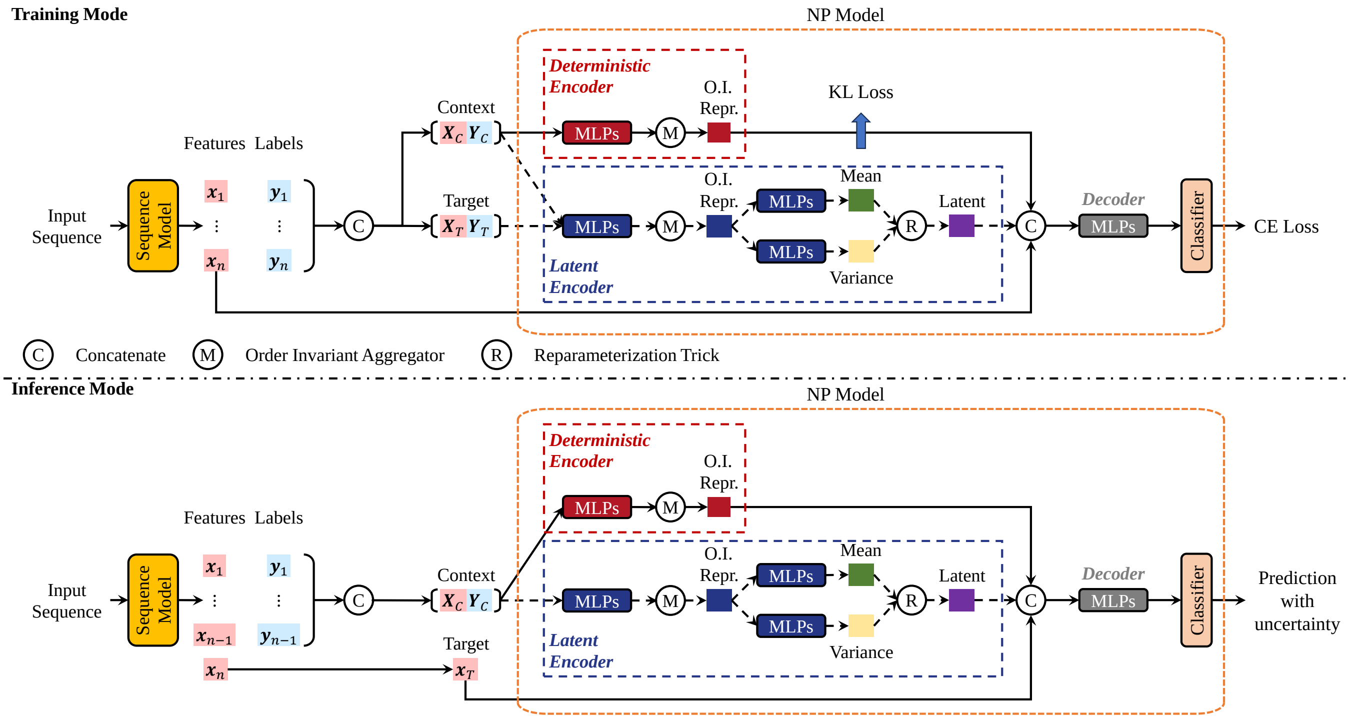


Figure 2: Overview of NP-Rec: it contains a Sequence model and an NP model. The upper and lower part illustrates the train and inference mode, respectively. As revealed in the orange dashed box, the NP model is built entirely upon MLPs which comprises a Encoder-Decoder architecture and a Classifier. "O.I. Repr." is short for the order invariant representations.

Preliminaries

We utilize $\mathcal{U}$ and $\mathcal{I}$ to denote the user and item set, which contain $|\mathcal{U}|$ users and $|\mathcal{I}|$ items, separately. Accordingly, we have the following basic definitions.

Basic Definition

Definition 1: (User-item Interaction) A user-item interaction is denoted by a triplet $x = \langle u, i, t \rangle$, which means that a user $u \in \mathcal{U}$ interacted with an item $i \in \mathcal{I}$ at time t.

Definition 2: (History Sequence) A user u's history sequence chronologically records the user's $|X^u|$ interactions with items, denoted as $X^u = x_1 \to x_2 \to \cdots \to x_{|X^u|}$.

Problem Statement

Multi-interest Recommendation Given a user u's history sequence X^u, the multi-interest recommendation problem models the user's diverse and dynamic preferences, and provides the top-N recommendation list which contains N items that the user might interact with in the next time step. It can summarized with the following equation,

$$\mathcal{R}^u = \mathrm{MiRS}(X^u) \qquad (1)$$

where $\mathcal{R}^u$ is the top-N recommendation list and $\mathrm{MiRS}(\cdot)$ denotes any multi-interest recommender system that takes as input a user's history sequence.

Methodology

In this section, we start from introducing neural processes (NPs), and followed by a detailed description of NP-Rec.

Neural Processes (NPs)

NPs aim at mapping an input $\boldsymbol{x}_i \in \mathbb{R}^{d_x}$ to the corresponding output $\boldsymbol{y}_i \in \mathbb{R}^{d_y}$ based on an (infinite) family of conditional distributions. In particular, one may condition on an arbitrary number of observed *Contexts* $(\boldsymbol{X}_C, \boldsymbol{Y}_C) = (\boldsymbol{x}_C, \boldsymbol{y}_C)_{i \in C}$ to model an arbitrary number of *Targets* $(\boldsymbol{X}_T, \boldsymbol{Y}_T) = (\boldsymbol{x}_T, \boldsymbol{y}_T)_{i \in T}$. The arbitrary property requires the mapping procedure should be non-sensitive towards the order of contexts or targets. The conditional distribution is

$$p(\boldsymbol{Y}_T|\boldsymbol{X}_T, \boldsymbol{X}_C, \boldsymbol{Y}_C) = \int p(\boldsymbol{Y}_T|\boldsymbol{X}_T, \boldsymbol{r}_C, \boldsymbol{z}) p(\boldsymbol{z}|\boldsymbol{s}_C) d\boldsymbol{z}, \qquad (2)$$

where $\boldsymbol{r}_C = r(\boldsymbol{x}_C, \boldsymbol{y}_C) \in \mathbb{R}^d$ and $\boldsymbol{s}_C = s(\boldsymbol{x}_C, \boldsymbol{y}_C) \in \mathbb{R}^d$ are the finite dimensional representations. $r(\cdot)$ is an order invariant *deterministic* function which aggregates contexts, and $s(\cdot)$ is the *latent* one of the same properties. Given the observation $(\boldsymbol{x}_C, \boldsymbol{y}_C)$, the global latent $\boldsymbol{z} \in \mathbb{R}^d$ accounts for incorporating uncertainties in the predictions $\boldsymbol{Y}_T$ which is modeled by a factorized Gaussian parameterized $\boldsymbol{s}_C$.

Given a random subset of contexts C and targets T, NPs learn the parameters in the encoder-decoder architecture by maximizing the following ELBO with reparameterization trick (Kingma and Welling 2014),

$$\log p(\boldsymbol{Y}_T|\boldsymbol{X}_T, \boldsymbol{X}_C, \boldsymbol{Y}_C) \geq \\ \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{s}_T)}[\log p(\boldsymbol{Y}_T|\boldsymbol{X}_T, \boldsymbol{r}_C, \boldsymbol{z})] - D_{KL}(q(\boldsymbol{z}|\boldsymbol{s}_T)||q(\boldsymbol{z}|\boldsymbol{s}_C)), \tag{3}$$

where q, r and s form the encoder part, and the likelihood p is referred as the decoder.

NP-Rec

As Figure 2 shows, NP-Rec mainly comprises with two components: a sequence model and an NP model. The sequence model takes as input the history sequence, and obtains the high dimensional features, and the NP model receives the features to learn a distribution over preference functions for classification.

Sequence Model Given a history sequence X^1, NP-Rec employs a sequence model to get the high dimensional features. We expect the features can contain the sequential dependency to model the individual preference transition pattern. In practical, we implement the sequence model with a embedding layer and the Mamba architecture (Gu and Dao 2023), which is a selective state space sequence model. Since the history sequences of different users might be inconsistent in length, we follow the "clip-and-pad" strategy in (Jiang et al. 2024a; Xu et al. 2022b) to uniform the input length as n, and obtain the representation $\boldsymbol{X} \in \mathbb{R}^{n \times d}$.

NP Model for Recommendation We regard the recommendation as a classification task that selects the most relevant N items, i.e., classes, from $|\mathcal{I}|$ candidates. Considering the discrete characteristics, we define a categorical distribution for the decoder in Eq. 2 instead of a Gaussian distribution. Specifically, we implement the categorical distribution with the Classifier, which comprises a weight matrix $\boldsymbol{W}_{cls}$ and a Softmax function, as follows,

$$\begin{aligned} p(\boldsymbol{Y}_T|\boldsymbol{X}_T, \boldsymbol{r}_C, \boldsymbol{z}) &= \text{Classifier}(p(\boldsymbol{Y}_T|\boldsymbol{X}_T, \boldsymbol{r}_C, \boldsymbol{z})) \\ &= \text{Softmax}(\boldsymbol{W}_{cls} p(\boldsymbol{Y}_T|\boldsymbol{X}_T, \boldsymbol{r}_C, \boldsymbol{z})) \\ &\sim \mathcal{C}(p_1, \cdots, p_{|\mathcal{I}|}) \quad \text{s.t.} \quad \sum_{i=1}^{|\mathcal{I}|} p_i = 1, \end{aligned} \tag{4}$$

where $\mathcal{C}$ denotes the categorical distribution.

NP-Rec Pipeline The orange dashed box in Figure 2 reveals our configuration about the NP model. The NP model employs the Encoder-Decoder architecture and followed by a Classifier. To achieve the most expressive model, we follow Garnelo et al. (2018b) to exploit the both of deterministic and latent path, which are marked with the red and blue dashed box in Figure 2, separately. The workflows of NP-Rec are different during the training and inference procedure. Next, we elaborate on the details.

Training Mode. We set the input history sequence $X = x_1 \to \cdots \to x_n$ and label sequence $Y = y_1 \to \cdots \to y_n$ in line with the sequential recommendation scheme (Jiang et al. 2024a; Yang et al. 2023), where $y_i = x_{i+1}$ for $i = 1, \cdots, n$. At each time step i, we make prediction for the next step.

Given the sequence representation $\boldsymbol{X}$ and the corresponding one-hot labels $\boldsymbol{Y} \in \mathbb{R}^{n \times I}$, we concatenate them along the last dimension to form the input-label pairs, i.e., $[\boldsymbol{X}, \boldsymbol{Y}] \in \mathbb{R}^{n \times (d+|\mathcal{I}|)}$. Then, we split the pairs into two non-overlapping parts C context pairs $[\boldsymbol{X}_C, \boldsymbol{Y}_C] \in \mathbb{R}^{C \times (d+|\mathcal{I}|)}$ and T target pairs $[\boldsymbol{X}_T, \boldsymbol{Y}_T] \in \mathbb{R}^{T \times (d+|\mathcal{I}|)}$, where $n = C + T$. The detailed process is summarized as follows.

As for the latent encoder, it processes both the contexts and targets. Take the context for instance, it maps the input into the latent space with MLPs $s(\cdot)$. After the mean aggregator, it further employs two MLPs to get the mean and variance vector, separately, i.e., the prior distribution $q(\boldsymbol{z}|\boldsymbol{s}_C)$. Then, it samples K latent representations via reparameterization trick $\boldsymbol{z}_C \in \mathbb{R}^{K \times d}$. Towards the target input, the latent performs in the same manner to get the posterior distribution $q(\boldsymbol{z}|\boldsymbol{s}_T)$ and latent representations $\boldsymbol{z}_T \in \mathbb{R}^{K \times d}$.

As for the deterministic encoder, it takes as input only the context. Firstly, it processes the input context with MLPs $r(\cdot)$, and then averages the output along the sequence length dimension to get the order invariant representation $\boldsymbol{r}_C \in \mathbb{R}^d$.

The decoder takes as input the sequence features $\boldsymbol{X}$, deterministic representation $\boldsymbol{r}_C$ and posterior latent $\boldsymbol{z}_T$ for classification. Since we aim at maximizing the ELBO of the complete history sequence, we duplicate $\boldsymbol{X}$, $\boldsymbol{z}_T$ and $\boldsymbol{r}_C$ by K, n and $n \times K$ times, separately. We concatenate the representations along the last dimension to form the decoder input. After the Classifier, we get K predicted categorical distributions for each point in the sequence. The final prediction is achieved by averaging these K distributions, where the uncertainty is computed as the entropy of mean (Wang et al. 2022).

The learning objective is formulated in Eq. 3. As shown in the upper part of Figure 2, the likelihood $p(\boldsymbol{Y}_T|\boldsymbol{X}_T, \boldsymbol{r}_C, \boldsymbol{z})$ can be calculated with the cross entropy loss between the predicted categorical distribution and the one-hot label vectors. The KL divergence loss $D_{KL}(\cdot)$ is computed with the prior and posterior distributions.

Inference Mode. Given a history sequence $X = x_1 \to \cdots \to x_n$, NP-Rec aims at making predictions for the next time step x_{n+1}. Since the label for the n step is unavailable, as shown in the lower part of Figure 2, we set the previous $n - 1$ entries in the sequence representations and corresponding one-hot label vectors as contexts, i.e., $[\boldsymbol{X}_C, \boldsymbol{Y}_C] = [\boldsymbol{X}_{1:n-1}, \boldsymbol{Y}_{1:n-1}] \in \mathbb{R}^{(n-1) \times (d+|\mathcal{I}|)}$. The target input is denoted as $\boldsymbol{X}_T = \boldsymbol{X}_n \in \mathbb{R}^{1 \times d}$.

During the inference procedure, only the contexts would pass through the encoders. As for the latent one, it firsts get the mean and variance vector of the prior distribution $q(\boldsymbol{z}|\boldsymbol{s}_C)$ with MLPs $s(\cdot)$, and samples K latent representations $\boldsymbol{z}_C$. Towards the deterministic encoder, it generates the order invariant $\boldsymbol{r}_C$ with the same manner in the training mode. Recall the multiple sampling mechanism, we make K copies of target $\boldsymbol{X}_T$ and deterministic representation $\boldsymbol{r}_C$, and concatenate them with the latent ones $\boldsymbol{z}_C$ to form the decoder input. Similarly, Classifier outputs K categorical distributions over the next item, and makes the final recommendation based on averaged distribution where the uncertainty is naturally included.

[1] We omit the superscript u for concise, which specifies the user.

Experiments

In this section, we start from introducing the experimental settings, and then discuss the experimental results. We conduct an amount of experiments to answer the following research questions:

- **RQ 1:** Can NP-Rec provide competitive or superior recommendation performance against state-of-the-art recommender systems, including both the single-point and multi-point interest representations?
- **RQ 2:** How is the effectiveness of major components under the NP-Rec framework, including the sequence model and NP model?
- **RQ 3:** Can the NP paradigm be extended to existing recommendation models and benefit the performance by incorporating uncertainty estimations?
- **RQ 4:** How is the model sensitivity with respect to the sampling times K?

Dataset

We conduct the validation on four widely studied datasets from two benchmarks MovieLens (Russo et al. 2018) and Foursquare (Yang, Zhang, and Qu 2016), including ML-100K, ML-1M, NYC and TKY. To ensure the data quality, we follow (Jiang et al. 2024a; Wang et al. 2023) to screen out the "unpopular" items which are interacted by less than 10 times, and "inactive" users whose interactions are fewer than 20 times. The statistics after processed are listed in Table 1. We set the maximum sequence length n of each dataset according to the average one, that $n = 100$ in ML-100K, $n = 160$ in ML1M, $n = 30$ in NYC and $n = 50$ in TKY. Towards the data partition, we select each user's last previously un-interacted item as the target during recommendation procedure[2], and all the prior items for training.

Metrics

We adopt the following two metrics to measure the recommendation performance, including

- **Hit Ratio (HR)** counts the frequency that the top-N recommendation list contains the target;
- **Normalized Discounted Cumulative Gain (NDCG)** emphasizes the order inside the recommendation list.

The higher the metric values, the better the recommendation performances. We report $\mathrm{N} = \{5, 10\}$ in our experiments.

Baselines

We select the following 9 competitors from two categories. Single-point Interest Representation (SIR) methods employ a unified vector to model a user's preference, which are common in sequential recommendation models, we choose the following 4 representative baselines which are related to different neural networks separately:

- **GRU4Rec** (Hidasi et al. 2016) first utilizes the gated recurrent unit for recommendation.

Dataset	#User	#Item	#Inter.	A. Len.	Sparsity
ML-100K	932	1,152	97,746	104.88	90.90%
ML-1M	6,034	3,260	998,428	165.47	94.92%
NYC	568	1,211	18,338	32.29	97.33%
TKY	1,962	2,876	97,746	49.82	98.27%

Table 1: The statistics of datasets after processed. "Interac." is the abbreviation of interactions, and "A. Len." is short for the average historical sequence length.

- **NextItNet** (Yuan et al. 2019) incorporates the hierarchical CNN for the long- and short-term preference modeling.
- **SASRec** (Kang and McAuley 2018) introduces the self-attention mechanism.
- **TriMLP** (Jiang et al. 2024a) exploits a triangle mixer and boosts the recommendation performance of MLPs.

Multi-point Interest Representation (MIR) methods utilize multiple vectors to depict a user's diverse interest scope. We choose the following 5 MIR recommendation models as baselines including the classic and modern ones.

- **MIND** (Li et al. 2019) is the pioneered MIR-based method, which employs the capsule network to extract multiple interest vectors with dynamic routing.
- **ComiRec**(Cen et al. 2020) further introduces a controller to reconcile the diversity v.s. precision trade-off.
- **PIMI**(Chen et al. 2021) considers the periodicity and interactivity contained in the history sequence.
- **Re4** (Zhang et al. 2022) focuses on regularizing the interest vectors learning procedure with the proposed backward reexamine.
- **REMI** (Xie et al. 2023) exploits a novel interest-aware negative sampling mechanism and the routing regularization strategy.

Implementation Details

We configure the proposed NP-Rec as follows. Towards the sequence model, we set the dimension $d = 64$ and employ a layer normalization after the embedding layer along with a dropout ratio 0.3. We stack 2 Mamba layers to generate the sequence features where the internal configuration follows the original hyper-parameter settings[3]. As for the NP model, we employ the ReLU activation function for all MLPs to inject non-linearity, and the intermediate dimensions are set as 64. The experiments are conducted on a single server with Intel 13900K CPU and NVIDIA RTX 4090 GPU.

Overall Recommendation Performance (RQ 1)

The overall recommendation performance[4] of all compared methods are listed in Table 2. Accordingly, we have the following observations.

[2] Since the recommendation target is "new" to each user, our evaluation is performed on the purely exploration scenario which naturally avoids the repetition bias.

[3] https://github.com/state-spaces/mamba

[4] To achieve the fair comparison, we follow (Jiang et al. 2024a) to uniform the width and depth of all compared methods, which guarantees the same parameter scale. Other specific settings in baseline are same with the original implementations.

Dataset	Metric (%)	SIR				MIR					Ours	
		TriMLP	GRU4Rec	NextItNet	SASRec	MIND	ComiRec	PIMI	Re4	REMI	NP-Rec	Impv.
ML-100K	H@5	6.65	6.01	4.61	4.94	6.47	8.14	9.31	8.42	13.46	**18.78**	39.52%
	N@5	3.92	3.97	2.70	2.90	3.50	5.09	5.97	7.69	7.91	**10.92**	31.25%
	H@10	13.63	11.48	9.12	10.73	12.83	14.68	13.05	12.36	17.93	**22.75**	26.88%
	N@10	6.14	5.71	4.14	4.72	5.24	7.23	8.70	7.88	8.32	**10.22**	22.84%
ML-1M	H@5	14.70	15.38	16.27	16.29	12.01	12.87	14.23	15.34	16.68	**19.59**	17.44%
	N@5	9.84	10.08	11.08	11.00	7.36	8.27	9.49	10.38	11.37	**12.70**	11.69%
	H@10	21.88	23.47	24.49	24.15	21.82	23.39	24.12	24.59	26.58	**28.86**	8.57%
	N@10	12.15	12.68	13.85	13.52	9.90	11.04	13.19	14.30	13.42	**15.42**	7.83%
NYC	H@5	6.16	7.39	6.34	6.51	6.39	5.28	7.42	6.77	7.18	**7.75**	4.45%
	N@5	3.84	4.32	3.49	3.87	2.90	2.82	4.87	3.52	4.07	**5.08**	4.31%
	H@10	9.33	10.74	9.86	10.30	9.50	9.68	11.50	10.83	12.33	**13.91**	12.81%
	N@10	4.83	6.00	4.60	5.05	4.59	4.18	6.14	5.97	6.19	**6.94**	12.12%
TKY	H@5	11.37	11.11	9.43	11.06	11.57	11.21	13.97	14.43	14.97	**15.44**	3.14%
	N@5	7.60	6.96	5.75	7.44	7.56	5.46	7.53	7.72	7.90	**8.40**	5.95%
	H@10	15.60	14.68	13.20	14.83	14.48	14.58	16.68	18.37	22.34	**24.36**	8.86%
	N@10	8.98	8.13	6.97	8.67	8.47	6.54	8.79	9.56	10.57	**11.15**	5.49%

Table 2: Overall recommendation performance. The best and second scores are marked with **boldface** and underline forms, separately. The last column "Impv." stands for the improvement of our method against the strongest baseline.

- Although SIR methods default to represent the interest with a single vector, they outperforms the pioneered MIR methods, e.g. MIND and ComiRec, on most datasets. SIR methods focus on mining the sequential dependency contained in history sequence to model the dynamically shifting preferences, while MIND and ComiRec concentrates on depicting individual diverse interest. The underlying mechanism in SIR is more suitable for handling longer input sequences.
- Modern MIR methods like PIMI, Re4 and REMI reveal the remarkable superiority to the classical ones. REMI is the strongest competitor which achieves decent recommendation performance in most scenarios. It is credited to the proposed negative sampling and routing regularization mechanism, which leads to the more robust interest representations.
- Notably, our proposed NP-Rec consistently attains the best recommendation performances cross all metrics over all validated datasets, where the average improvement achieves up to 30.12% on ML-100K, 11.38% on ML-1M, 8.42% on NYC and 5.86% on TKY, respectively. It inherits the advantages of sequential dependency modeling from sequence model and extends to cover the diverse individual interest scope via the multiple sampling mechanism in the latent encoder of NP model. Moreover, NP-Rec averages multiple categorical distributions before the final recommendation, where incorporated entropy provides uncertainty estimation to serve the preference dynamic shifting property.

Ablation Study (RQ 2)

The proposed NP-Rec framework comprises two major components, including a Mamba-based sequence model and a NP model. Accordingly, we derive the following two variants to validate the effectiveness.

Dataset	ML-100K				ML-1M			
Metric (%)	H@5	N@5	H@10	N@10	H@5	N@5	H@10	N@10
Original	18.78	10.92	22.75	10.22	19.59	12.70	28.86	15.42
w.o. NP	13.95	6.87	15.49	6.26	13.82	12.24	24.76	13.18
w.o. SM	5.47	3.52	9.66	4.88	7.94	4.71	13.77	6.56

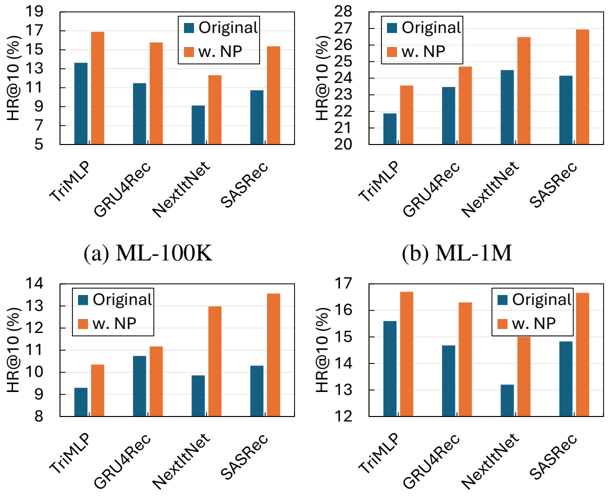
Dataset	NYC				TKY			
Metric (%)	H@5	N@5	H@10	N@10	H@5	N@5	H@10	N@10
Original	7.75	5.08	13.91	6.94	15.44	8.40	24.36	11.15
w.o. NP	4.75	2.63	8.87	4.39	9.46	6.17	13.18	7.35
w.o. SM	3.98	1.93	8.26	4.92	8.31	5.36	14.83	7.55

Table 3: Ablation study. "Original" denotes the complete NP-Rec framework. The variants that remove the neural process and sequence model are tagged as "w.o. NP" and "w.o. SM", separately.

- **w.o. NP** removes the NP model, and performs the same training and inference manner with sequential recommendation models.
- **w.o. SM** removes the Mamba architecture, and the subsequent NP model directly processes the input sequence embedding.

The experimental results are summarized in Table 3. Accordingly, we have the following findings.

- *Finding 1: NP model proves to be beneficial fro improving the recommendation performances.* Compared the original implementation with the variant w.o. NP, the incorporation of neural process averagely improves performance by 49.37% on all datasets. The reasons are twofolds. Firstly, the multiple sampling strategy in the latent encoder extends the single-point output of sequence model to depict the diverse interest scope. Further, the subsequent average operation provides the final recommendation with uncertainty estimations, which avoids the risk to be overconfident about the current preference.
- *Finding 2: The sequential modeling is necessary to achieve meaningful interest representations.* The variant



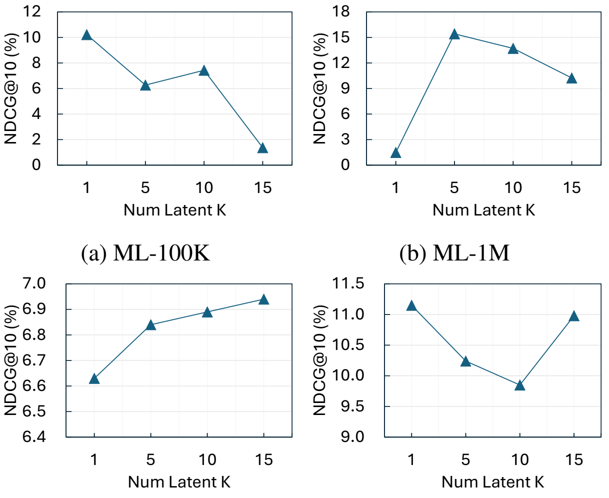
(a) ML-100K

(b) ML-1M

(c) NYC

(d) TKY

Figure 3: Extensibility of the proposed neural process recommendation paradigm. We report the HR@10 scores.



(a) ML-100K

(b) ML-1M

(c) NYC

(d) TKY

Figure 4: Sensitivity with respect to the latent sampling times K. We report the NDCG@10 scores.

w.o. SM dramatically decreases the recommendation performances. The core reason lies at the evaluation scenario that recommends for the next step. Without the sequential dependency endowed with Mamba, the subsequent NP model is hard to establish a meaningful mapping from the previous step item embedding to the next step item label since there is no preference transaction pattern included. Our NP-Rec takes the both advantages of sequence model and neural process, and achieves the superior recommendation performances.

Extensibility Analysis (RQ 3)

Since our NP-Rec can be viewed as performing the distribution modeling over the single-point output of the sequence model, we explore the extensibility of neural process paradigm and corresponding inter-sequence partition strategy, i.e., split the input sequence pairs into non-overlapping contexts and targets. Specifically, we select 4 SIR methods including TriMLP, GRU4Rec, NextItNet and SASRec as backbones which corresponds to the MLP-, RNN-, CNN- and Transformer-based sequence model, separately. Then, we replace the original Mamba architecture in NP-Rec with backbones to verify whether the subsequent NP model improves the recommendation performances.

The experimental results are shown in Figure 3. Obviously, the introduction of NP model largely boosts the recommendation performances. It demonstrates the insight that learning a distribution over function is more in line with the diverse and dynamic interest modeling requirements.

Sensitivity w.r.t. Hyper-parameters (RQ 4)

We mainly investigate the influence of sampling times K in the latent encoder of NP-Rec. The hyper-parameter K decides the number of predicted categorical distributions over candidate items, which is inline with the idea of MIR methods. We verify the setting range as $K = \{1, 5, 10, 15\}$.

The experimental results are revealed in Figure 4. We find that NP-Rec is sensitive towards different K. The most suitable settings are listed as follows: $K = 1$ for ML-100K and TKY, $K = 5$ for ML-1M and $K = 15$ for NYC.

Conclusion

In this paper, we concentrate on the diverse and dynamic property of individual interests, and propose NP-Rec for multi-interest recommendation. Instead of learning a parameterized function to approximate preference, we shift the viewpoint on learning a distribution over functions. By taking the both of sequence model and neural process, NP-Rec makes the predictions along with uncertainty estimations. Empirical observations on 4 real-world datasets demonstrate the superiority of NP-Rec. Besides, we prove that the proposed NP paradigm can benefit single-point interest representation methods.

In future, we would further improve the model's expressive capability by introducing the attention mechanism for order invariant aggregation and auto-regressive inference.

Acknowledgments

This work is supported by the Natural Science Foundation of China No. 62472196; Jilin Science and Technology Research Project No. 20230101067JC; Science and Technology Development Fund (FDCT), Macau SAR (file no. 0123/2023/RIA2, 001/2024/SKL) and National Natural Science Foundation of China General Program No. 62072209.

References

Arthur; O.; and Pittenger. 1979. Stochastic Processes: A Survey of the Mathematical Theory (John Lamperti). *SIAM Review*, 21(3): 421–422.

Bruinsma, W. P.; Markou, S.; Requeima, J.; Foong, A. Y. K.; Andersson, T. R.; Vaughan, A.; Buonomo, A.; Hosking, J. S.; and Turner, R. E. 2023. Autoregressive Conditional Neural Processes. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Cen, Y.; Zhang, J.; Zou, X.; Zhou, C.; Yang, H.; and Tang, J. 2020. Controllable Multi-Interest Framework for Recommendation. In Gupta, R.; Liu, Y.; Tang, J.; and Prakash, B. A., eds., *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*, 2942–2951. ACM.

Chen, G.; Zhang, X.; Zhao, Y.; Xue, C.; and Xiang, J. 2021. Exploring Periodicity and Interactivity in Multi-Interest Framework for Sequential Recommendation. In Zhou, Z., ed., *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, Canada, 19-27 August 2021*, 1426–1433. ijcai.org.

Eslami, S. M. A.; Rezende, D. J.; Besse, F.; Viola, F.; Morcos, A. S.; Garnelo, M.; Ruderman, A.; Rusu, A. A.; Danihelka, I.; Gregor, K.; Reichert, D. P.; Buesing, L.; Weber, T.; Vinyals, O.; Rosenbaum, D.; Rabinowitz, N.; King, H.; Hillier, C.; Botvinick, M.; Wierstra, D.; Kavukcuoglu, K.; and Hassabis, D. 2018. Neural scene representation and rendering. *Science*, 360(6394): 1204–1210.

Garnelo, M.; Rosenbaum, D.; Maddison, C.; Ramalho, T.; Saxton, D.; Shanahan, M.; Teh, Y. W.; Rezende, D. J.; and Eslami, S. M. A. 2018a. Conditional Neural Processes. In Dy, J. G.; and Krause, A., eds., *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, 1690–1699. PMLR.

Garnelo, M.; Schwarz, J.; Rosenbaum, D.; Viola, F.; Rezende, D. J.; Eslami, S. M. A.; and Teh, Y. W. 2018b. Neural Processes. *CoRR*, abs/1807.01622.

Gu, A.; and Dao, T. 2023. Mamba: Linear-Time Sequence Modeling with Selective State Spaces. *CoRR*, abs/2312.00752.

Hewitt, L. B.; Nye, M. I.; Gane, A.; Jaakkola, T. S.; and Tenenbaum, J. B. 2018. The Variational Homoencoder: Learning to learn high capacity generative models from few examples. In Globerson, A.; and Silva, R., eds., *Proceedings of the Thirty-Fourth Conference on Uncertainty in Artificial Intelligence, UAI 2018, Monterey, California, USA, August 6-10, 2018*, 988–997. AUAI Press.

Hidasi, B.; Karatzoglou, A.; Baltrunas, L.; and Tikk, D. 2016. Session-based Recommendations with Recurrent Neural Networks. In Bengio, Y.; and LeCun, Y., eds., *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

Jha, S.; Gong, D.; Wang, X.; Turner, R. E.; and Yao, L. 2022. The Neural Process Family: Survey, Applications and Perspectives. *CoRR*, abs/2209.00517.

Jiang, Y.; Xu, Y.; Yang, Y.; Yang, F.; Wang, P.; Li, C.; Zhuang, F.; and Xiong, H. 2024a. TriMLP: A Foundational MLP-like Architecture for Sequential Recommendation. *ACM Trans. Inf. Syst.* Just Accepted.

Jiang, Y.; Yang, Y.; Xu, Y.; and Wang, E. 2024b. Spatial-Temporal Interval Aware Individual Future Trajectory Prediction. *IEEE Transactions on Knowledge and Data Engineering*, 36(10): 5374–5387.

Kang, W.; and McAuley, J. J. 2018. Self-Attentive Sequential Recommendation. In *IEEE International Conference on Data Mining, ICDM 2018, Singapore, November 17-20, 2018*, 197–206. IEEE Computer Society.

Kim, H.; Mnih, A.; Schwarz, J.; Garnelo, M.; Eslami, S. M. A.; Rosenbaum, D.; Vinyals, O.; and Teh, Y. W. 2019. Attentive Neural Processes. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Kingma, D. P.; and Welling, M. 2014. Auto-Encoding Variational Bayes. In Bengio, Y.; and LeCun, Y., eds., *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*.

Li, C.; Liu, Z.; Wu, M.; Xu, Y.; Zhao, H.; Huang, P.; Kang, G.; Chen, Q.; Li, W.; and Lee, D. L. 2019. Multi-Interest Network with Dynamic Routing for Recommendation at Tmall. In Zhu, W.; Tao, D.; Cheng, X.; Cui, P.; Rundensteiner, E. A.; Carmel, D.; He, Q.; and Yu, J. X., eds., *Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019, Beijing, China, November 3-7, 2019*, 2615–2623. ACM.

Lin, X.; Wu, J.; Zhou, C.; Pan, S.; Cao, Y.; and Wang, B. 2021. Task-adaptive Neural Process for User Cold-Start Recommendation. In Leskovec, J.; Grobelnik, M.; Najork, M.; Tang, J.; and Zia, L., eds., *WWW '21: The Web Conference 2021, Virtual Event / Ljubljana, Slovenia, April 19-23, 2021*, 1306–1316. ACM / IW3C2.

Nguyen, T.; and Grover, A. 2022. Transformer Neural Processes: Uncertainty-Aware Meta Learning Via Sequence Modeling. In Chaudhuri, K.; Jegelka, S.; Song, L.; Szepesvári, C.; Niu, G.; and Sabato, S., eds., *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, 16569–16594. PMLR.

Qin, S.; Zhu, J.; Qin, J.; Wang, W.; and Zhao, D. 2019. Recurrent Attentive Neural Process for Sequential Data. *CoRR*, abs/1910.09323.

Russo, D.; Roy, B. V.; Kazerouni, A.; Osband, I.; and Wen, Z. 2018. A Tutorial on Thompson Sampling. *Found. Trends Mach. Learn.*, 11(1): 1–96.

Singh, G.; Yoon, J.; Son, Y.; and Ahn, S. 2019. Sequential Neural Processes. In Wallach, H. M.; Larochelle, H.; Beygelzimer, A.; d'Alché-Buc, F.; Fox, E. B.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, 10254–10264.

Ton, J.; Chan, L.; Teh, Y. W.; and Sejdinovic, D. 2021. Noise Contrastive Meta-Learning for Conditional Density Estimation using Kernel Mean Embeddings. In Banerjee, A.; and Fukumizu, K., eds., *The 24th International Conference on Artificial Intelligence and Statistics, AISTATS 2021, April 13-15, 2021, Virtual Event*, volume 130 of *Proceedings of Machine Learning Research*, 1099–1107. PMLR.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is All you Need. In Guyon, I.; von Luxburg, U.; Bengio, S.; Wallach, H. M.; Fergus, R.; Vishwanathan, S. V. N.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, 5998–6008.

Wang, E.; Xu, Y.; Yang, Y.; Jiang, Y.; Yang, F.; and Wu, J. 2023. Zone-Enhanced Spatio-Temporal Representation Learning for Urban POI Recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 35(9): 9628–9641.

Wang, J.; Lukasiewicz, T.; Massiceti, D.; Hu, X.; Pavlovic, V.; and Neophytou, A. 2022. NP-Match: When Neural Processes meet Semi-Supervised Learning. In Chaudhuri, K.; Jegelka, S.; Song, L.; Szepesvári, C.; Niu, G.; and Sabato, S., eds., *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, 22919–22934. PMLR.

Weston, J.; Weiss, R. J.; and Yee, H. 2013. Nonlinear latent factorization by embedding multiple user interests. In Yang, Q.; King, I.; Li, Q.; Pu, P.; and Karypis, G., eds., *Seventh ACM Conference on Recommender Systems, RecSys '13, Hong Kong, China, October 12-16, 2013*, 65–68. ACM.

Willi, T.; Masci, J.; Schmidhuber, J.; and Osendorfer, C. 2019. Recurrent Neural Processes. *CoRR*, abs/1906.05915.

Wu, H.; Meshi, O.; Zoghi, M.; Diaz, F.; Liu, X.; Boutilier, C.; and Karimzadehgan, M. 2023. Density-based User Representation through Gaussian Process Regression for Multi-interest Personalized Retrieval. *CoRR*, abs/2310.20091.

Xie, Y.; Gao, J.; Zhou, P.; Ye, Q.; Hua, Y.; Kim, J. B.; Wu, F.; and Kim, S. 2023. Rethinking Multi-Interest Learning for Candidate Matching in Recommender Systems. In Zhang, J.; Chen, L.; Berkovsky, S.; Zhang, M.; Noia, T. D.; Basilico, J.; Pizzato, L.; and Song, Y., eds., *Proceedings of the 17th ACM Conference on Recommender Systems, RecSys 2023, Singapore, Singapore, September 18-22, 2023*, 283–293. ACM.

Xu, Y.; Wang, E.; Yang, Y.; and Chang, Y. 2022a. A Unified Collaborative Representation Learning for Neural-Network Based Recommender Systems. *IEEE Transactions on Knowledge and Data Engineering*, 34(11): 5126–5139.

Xu, Y.; Wang, E.; Yang, Y.; and Xiong, H. 2024a. GS-RS: A Generative Approach for Alleviating Cold Start and Filter Bubbles in Recommender Systems. *IEEE Transactions on Knowledge and Data Engineering*, 36(2): 668–681.

Xu, Y.; Yang, Y.; Wang, E.; Zhuang, F.; and Xiong, H. 2022b. Detect Professional Malicious User With Metric Learning in Recommender Systems. *IEEE Transactions on Knowledge and Data Engineering*, 34(9): 4133–4146.

Xu, Y.; Zhuang, F.; Wang, E.; Li, C.; and Wu, J. 2024b. Learning without Missing-At-Random Prior Propensity-A Generative Approach for Recommender Systems. *IEEE Transactions on Knowledge and Data Engineering*, 1–13.

Yang, D.; Zhang, D.; and Qu, B. 2016. Participatory Cultural Mapping Based on Collective Behavior Data in Location-Based Social Networks. *ACM Trans. Intell. Syst. Technol.*, 7(3): 30:1–30:23.

Yang, L.; Shi, R.; Zhang, Q.; Niu, B.; Wang, Z.; Cao, X.; and Wang, C. 2023. Self-supervised Graph Neural Networks via Low-Rank Decomposition. In Oh, A.; Naumann, T.; Globerson, A.; Saenko, K.; Hardt, M.; and Levine, S., eds., *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Yuan, F.; Karatzoglou, A.; Arapakis, I.; Jose, J. M.; and He, X. 2019. A Simple Convolutional Generative Network for Next Item Recommendation. In Culpepper, J. S.; Moffat, A.; Bennett, P. N.; and Lerman, K., eds., *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, WSDM 2019, Melbourne, VIC, Australia, February 11-15, 2019*, 582–590. ACM.

Zaheer, M.; Kottur, S.; Ravanbakhsh, S.; Póczos, B.; Salakhutdinov, R.; and Smola, A. J. 2017. Deep Sets. In Guyon, I.; von Luxburg, U.; Bengio, S.; Wallach, H. M.; Fergus, R.; Vishwanathan, S. V. N.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, 3391–3401.

Zhang, S.; Yang, L.; Yao, D.; Lu, Y.; Feng, F.; Zhao, Z.; Chua, T.; and Wu, F. 2022. Re4: Learning to Re-contrast, Re-attend, Re-construct for Multi-interest Recommendation. In Laforest, F.; Troncy, R.; Simperl, E.; Agarwal, D.; Gionis, A.; Herman, I.; and Médini, L., eds., *WWW '22: The ACM Web Conference 2022, Virtual Event, Lyon, France, April 25 - 29, 2022*, 2216–2226. ACM.

Zhang, X.; Liu, J.; Chang, S.; Gong, P.; Wu, Z.; and Han, B. 2023. MIRN: A multi-interest retrieval network with sequence-to-interest EM routing. *PLOS ONE*, 18.

Zhuo, J.; Qin, F.; Cui, C.; Fu, K.; Niu, B.; Wang, M.; Guo, Y.; Wang, C.; Wang, Z.; Cao, X.; and Yang, L. 2024. Improving Graph Contrastive Learning via Adaptive Positive Sampling. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, 23179–23187. IEEE.