# Probabilistic Multi-Interest Representation for Personalized Recommendation

Sewon Lee*† Junhyeong Park*† Yebonn Han† Joonseok Lee†‡

**Abstract.** Recommender systems play a critical role in enhancing user experience on online platforms. Traditional models typically represent each user with a single embedding, but this potentially limits the capacity to reflect their diverse preferences. Multi-interest recommendation (MIR) addresses this by representing users with multiple embeddings, each corresponding to a distinct interest. However, since each interest is still treated as a fixed point in the latent space, they capture only the central tendency of each interest, failing to reflect its detailed structure. To fully capture the dispersion within each interest, we propose a probabilistic multi-interest recommender, representing each user as a mixture of multiple interest distributions. Each interest is modeled not just with a point embedding but also with its covariance, capturing both central tendency and spread of the preference. The final user representation is then composed by aggregating these multiple interests weighted by their relative importance. This distributional view allows a flexible and nuanced representation of user preferences, effectively modeling both their diversity and fine-grained structure. To further improve user-item matching, we introduce a distribution-aware scoring strategy that jointly considers the overall proximity and the interest spread. With extensive experiments, we verify that our method consistently outperforms existing MIR models at a negligible cost.

**1 Introduction.** Recommender systems (RS) are essential for delivering personalized content across various platforms [28, 33, 8, 25]. Among these, Sequential Recommendation (SR) [18, 21, 44, 51, 26, 20] has gained attention for modeling how user interests evolve over time. A typical SR approach embeds users and items into a shared latent space, where proximity indicates relevance. However, representing each user as a single embedding (Figure 1, left) fails to capture the multi-faceted nature of real-world preferences.

To address this, Multi-Interest Recommendation (MIR) models [28, 2, 50, 48, 5, 46, 35, 11, 37] represent each user with *multiple* embeddings, where each reflects a distinct interest facet (Figure 1, middle). Existing

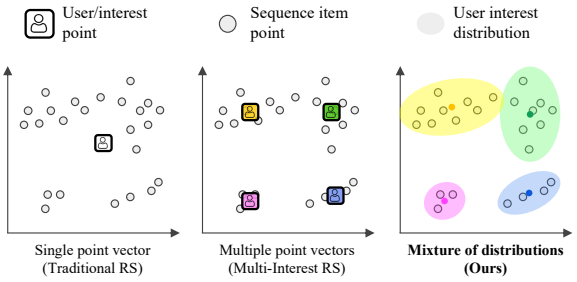



Figure 1: Different user representation approaches across model type.

MIR approaches, however, still treat each interest as a fixed point, overlooking the internal structure within each interest. Even within a single interest, preferences often span over multiple sub-concepts with varying dispersion; *e.g.*, a user may favor romantic comedies across a wide range of romance intensity while preferring comedy more narrowly. Such dimension-specific dispersion cannot be captured by point embeddings.

Figure 2 illustrates this limitation. The user has a spread of interests in the yellow-shaded region within the embedding space. If she is represented as a single point and we recommend by relying solely on the (Euclidean) proximity from it, a suboptimal item like B may be selected over A, even if A better aligns with the spread and structure of the user's preferences. Moreover, most MIR models score items based only on the best-matching interest, overlooking how multiple interests jointly influence user behavior.

To overcome this challenge, we propose **P**robabilistic **M**ulti-**I**nterest **Rec**ommender (PMIRec), which models each user interest as a probability distribution characterized by (i) a mean, (ii) a covariance capturing intra-interest dispersion, and (iii) an importance weight reflecting interest strength. Each user is represented as a mixture of such distributions, and candidate items are evaluated via a novel probabilistic matching mechanism. PMIRec can be seamlessly integrated into existing MIR models with minimal architectural change, imposing negligible inference overhead.

**2 Related Work.** MIR models [2, 28, 50, 48, 53, 27, 5, 46, 11, 35, 45, 37] represent users via multiple embeddings to capture diverse preferences. Early work

*Equal contribution
†Seoul National University
‡Corresponding author

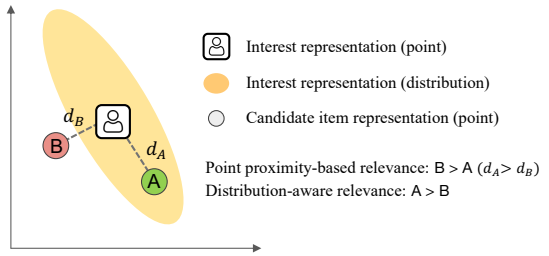


Figure 2: Motivation for probabilistic user modeling.

such as MIND [28] and ComiRec [2] uses dynamic routing and multi-head attention for multi-interest extraction, while subsequent models [27, 11, 50, 37] further improve interest disentanglement and training. However, all of these methods rely on point embeddings, lacking the ability to represent intra-interest dispersion.

To address this limitation, a natural direction is to model users and items as probability distributions rather than point vectors [10, 14, 15, 38], though they remain limited to unimodal or point-wise matching. More recently, diffusion-based methods [49, 30, 52, 34, 29] offer flexible preference distribution modeling without parametric constraints. However, their iterative denoising incurs substantial inference latency. Our method achieves both distributional flexibility and practical inference efficiency. Appendix A provides extended related work with more relevant work.

**3 Problem Formulation.** Let $\mathcal{U}$ and $\mathcal{I}$ represent the sets of users and items, respectively. For a user $u \in \mathcal{U}$, the interaction history is represented as an ordered sequence of items $\mathcal{S}_u = [v_1, v_2, \ldots, v_n; u]$, where $v_i \in \mathcal{I}$ denotes the $i$-th item that the user has interacted with. We fix the sequence length $n$ across users, either by truncating older interactions or padding as needed. Then, the goal is to predict the next items a user is likely to interact with, given their historical behaviors.

**4 Proposed Method: PMIRec.** We propose a novel multi-interest distribution modeling framework where each user is represented as a mixture of distributions over the item embedding space, enabling a fine-grained representation of diverse user interests. Considering the fact that user preferences are relatively more heterogeneous compared to item characteristics, we represent each user as a mixture of distributions, while representing each item as a single point. We adopt a pluggable design that can be seamlessly integrated into existing multi-interest recommenders.

The overall architecture of the proposed framework is illustrated in Figure 3. Item features are first projected into the shared embedding space through the embedding layer (§4.1). Then multiple user interests are extracted (§4.2) by leveraging a multi-interest recommender backbone [2, 28, 50, 11, 35, 48, 5, 27]. Our proposed multi-interest distribution modeling approach (§4.3) further extends these point-based interest representations into distributions by modeling the covariance and importance weight to capture the dispersion and relative strength of each interest. Finally, recommendation scores are predicted by our novel probabilistic user-item matching mechanism to evaluate candidate items based on the learned user distributions (§4.4). We summarize our training objective (§4.5) and analyze computational overhead (§4.6).

**4.1 Embedding Layer.** Given a user interaction sequence $\mathcal{S}_u$ with $n$ items, the embedding layer maps each item ID to an embedding through a look-up table, producing a sequence of item embeddings:

$$\mathbf{E_u} = [\mathbf{e}_1, \mathbf{e}_2, \cdots, \mathbf{e}_n; u]^\top \in \mathbb{R}^{n \times d},$$

where $\mathbf{e}_i \in \mathbb{R}^d$ denotes the embedding of the $i$-th item in the user sequence, and $d$ is the dimensionality of the embedding. To incorporate positional information in the sequence, we optionally add learnable positional embeddings $\mathbf{p}_i \in \mathbb{R}^d$ for the position $i$ within the given sequence: $\tilde{\mathbf{e}}_i = \mathbf{e}_i + \mathbf{p}_i$. To be uncluttered, we will use $\mathbf{e}_i$ to denote the item embedding henceforth.

**4.2 Multi-Interest Extraction.** We model each user with multiple interest representations, where each interest is formulated as a mean vector capturing the central tendency. These mean embeddings are extracted from the user's interaction sequence via an MIR backbone. Our framework is model-agnostic, allowing any MIR model to be adopted.

Formally, the goal of the Multi-Interest Extraction step is to generate $K$ distinct mean embeddings for each user, where $K$ is a hyperparameter denoting the number of interests. Given a user's sequence of item embeddings $\mathbf{E}_u \in \mathbb{R}^{n \times d}$, we compute the contribution weight of each item to the $k$-th interest. Specifically, a scoring network $h : \mathbb{R}^{n \times d} \to \mathbb{R}^{n \times K}$, commonly implemented with a capsule network [28, 11] or an MLP [2, 50], computes the item-interest relevance score $h(\mathbf{E}_u)$. The contribution weight $\alpha_i^{(k)}$ for the $i$-th item and the $k$-th interest is then given by

$$\alpha_i^{(k)} = \left(\operatorname{softmax}\left(h\left(\mathbf{E}_u\right)\right)\right)_{i,k} \in \mathbb{R}, \tag{4.1}$$

where the softmax is applied over the scores of all items in the sequence for each interest independently. The mean embedding $\boldsymbol{\mu}^{(k)} \in \mathbb{R}^d$ for the $k$-th interest is obtained by a weighted sum of sequence item embeddings:

$$\boldsymbol{\mu}^{(k)} = \sum_{i=1}^{n} \alpha_i^{(k)} \mathbf{e}_i. \tag{4.2}$$

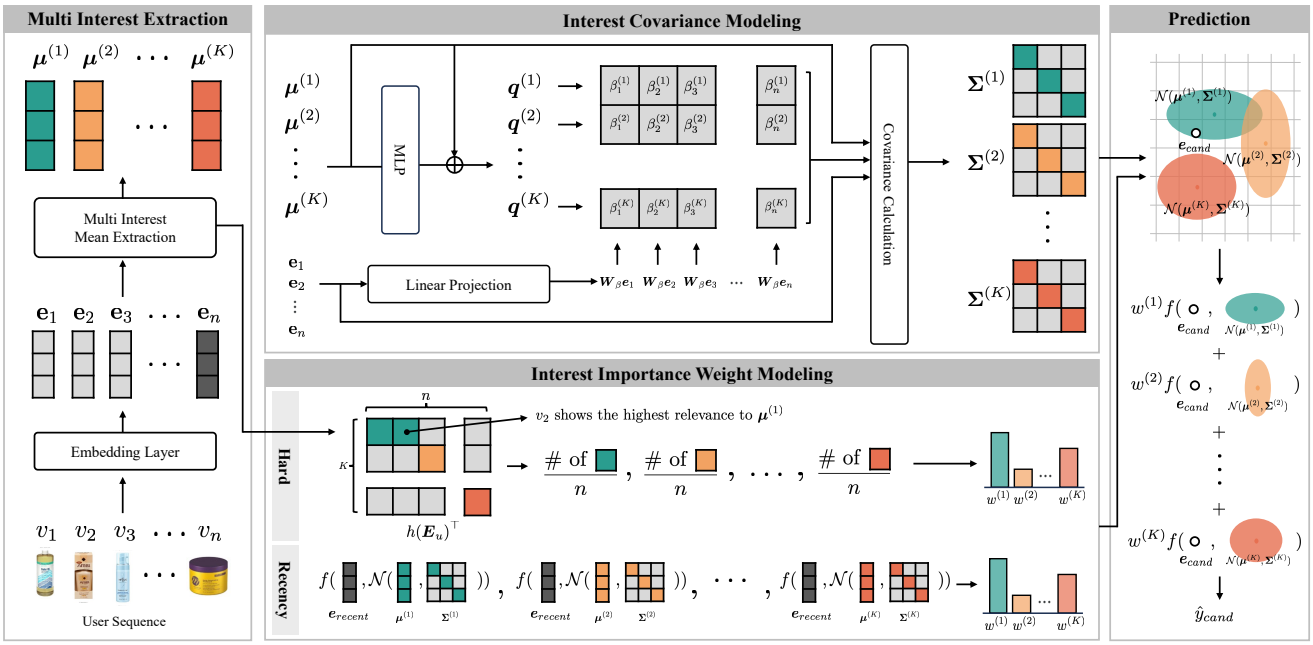



Figure 3: Overall architecture of our proposed model, PMIRec. Note that the low-rank approximation mechanism within the Interest Covariance Modeling module is omitted for visual clarity.

Repeating this process for $K$ interests yields $K$ interest mean embeddings $\boldsymbol{\mu}^{(1)}, \boldsymbol{\mu}^{(2)}, \cdots, \boldsymbol{\mu}^{(K)}$ for each user.

**4.3 Multi-Interest Distribution Modeling.** To model user preferences at a finer granularity, we propose to extend each point-based interest representation into a probability distribution. Specifically, we formulate each user's interest as a mixture of $K$ Gaussian distributions, each parameterized by a mean, a covariance, and an importance weight. This provides a flexible yet tractable way to model the continuous nature of user preferences while maintaining computational efficiency [47]. With this extended representation, our model is expected to effectively capture the detailed nuances of the preferences. The distribution modeling consists of two components: **Interest Covariance Modeling** (§4.3.1), and **Interest Importance Weight Modeling** (§4.3.2).

**4.3.1 Interest Covariance Modeling.** While prior works model user interests as point embeddings capturing only the mean, we further take into account the dispersion of each interest by introducing a covariance $\boldsymbol{\Sigma}^{(k)} \in \mathbb{R}^{d\times d}$ for each interest $k$. This allows the model to capture fine-grained structure within individual interests, offering a richer and more flexible user representation.

*Full covariance.* By definition, the full covariance matrix for each interest $k$ is formulated by taking the outer product of the difference between item and interest mean $(\mathbf{e}_i - \boldsymbol{\mu}^{(k)})$, with its transpose. Following [15], the full covariance matrix $\boldsymbol{\Sigma}^{(k)}$ is given by

$$\boldsymbol{\Sigma}^{(k)} = \mathbf{1}_{d\times d} + \mathrm{ELU}\left(\sum_{i=1}^{n} \beta_i^{(k)} \left(\mathbf{e}_i - \boldsymbol{\mu}^{(k)}\right)\left(\mathbf{e}_i - \boldsymbol{\mu}^{(k)}\right)^\top\right), \tag{4.3}$$

Here, we introduce a learnable weight $\beta_i^{(k)} \in \mathbb{R}$ that represents item-interest relevance, placing more emphasis on items strongly associated with the interest. To compute $\beta_i^{(k)}$, we first construct an interest-specific query vector $\mathbf{q}^{(k)}$ based on the interest mean embedding $\boldsymbol{\mu}^{(k)}$. Specifically, the query is defined as the sum of the interest mean and its transformation via an MLP, allowing for a more expressive interest representation:

$$\mathbf{q}^{(k)} = \boldsymbol{\mu}^{(k)} + \mathrm{MLP}\left(\boldsymbol{\mu}^{(k)}\right) \in \mathbb{R}^d. \tag{4.4}$$

This query is then matched against the projected item embedding to compute the item-interest relevance weight:

$$\beta_i^{(k)} = \mathrm{softmax}\left(\frac{(\mathbf{W}_\beta \mathbf{E}_u^\top)^\top \mathbf{q}^{(k)}}{\sqrt{d}}\right)_i \in \mathbb{R}, \tag{4.5}$$

where $\mathbf{W}_\beta \in \mathbb{R}^{d\times d}$ is a learnable projection matrix and $\mathbf{E}_u \in \mathbb{R}^{n\times d}$ represents the sequence item embeddings. Note that both $\alpha_i^{(k)}$ in Eq. (4.1) and $\beta_i^{(k)}$ in Eq. (4.5) conceptually measure relevance between the $i$-th item and the $k$-th interest, but we decide to model them separately, because modeling the spread would require distinct information than capturing the central tendency.

We empirically compare the use of $\beta_i^{(k)}$ versus $\alpha_i^{(k)}$ in §5.4.

In practice, directly constructing and utilizing a full $d \times d$ matrix is computationally prohibitive and memory-intensive when $d$ is large. To address this, we explore two strategies to simplify covariance modeling: diagonal covariance and low-rank approximation.

*Diagonal covariance.* Assuming independence across embedding dimensions, this approach considers only the diagonal entries of the covariance matrix (4.3), *i.e.*, per-dimension variances:

$$\mathbf{\Sigma}_{\text{diag}}^{(k)} = \text{diag}\left(\mathbf{\Sigma}^{(k)}\right) \in \mathbb{R}^d, \tag{4.6}$$

*Low-rank approximations.* To flexibly capture dependencies between embedding dimensions with reduced computational overhead, we approximate the full covariance matrix with a low-rank matrix [12], $\mathbf{\Sigma}^{(k)} \approx \mathbf{A}^{(k)}(\mathbf{A}^{(k)})^\top$ with $\mathbf{A}^{(k)} \in \mathbb{R}^{d \times r}$ and $r \ll d$. We consider two ways to construct $\mathbf{A}^{(k)}$: learned projection and top-$r$ item selection.

With learned projection, the item-interest difference matrix $\mathbf{D}^{(k)}$ is projected onto a lower-dimensional space before aggregation:

$$\begin{aligned} \mathbf{D}^{(k)} &= \left(\mathbf{E}_u - \mathbf{1}_n \left(\boldsymbol{\mu}^{(k)}\right)^\top\right) \odot \left(\sqrt{\boldsymbol{\beta}^{(k)}} \mathbf{1}_d^\top\right) \in \mathbb{R}^{n \times d}, \\ \mathbf{A}^{(k)} &= \left(\mathbf{D}^{(k)}\right)^\top \left(\mathbf{D}^{(k)} \mathbf{W}_p\right) \in \mathbb{R}^{d \times r} \end{aligned} \tag{4.7}$$

where $\mathbf{E}_u \in \mathbb{R}^{n \times d}$ is sequence item embeddings, $\mathbf{W}_{\text{proj}} \in \mathbb{R}^{d \times r}$ is a learnable projection matrix, and $\odot$ is an element-wise multiplication that scales each difference vector row by square-rooted item-interest relevance weight $\sqrt{\boldsymbol{\beta}^{(k)}} \in \mathbb{R}^n$. $\mathbf{1}_n$ and $\mathbf{1}_d$ are all-ones vectors used for broadcasting.

As an alternative, we select the top-$r$ items from each user sequence based on relevance weights $\boldsymbol{\beta}^{(k)}$. Then, we construct $\mathbf{A}^{(k)}$ by stacking the weighted differences between those selected items and interest mean embeddings:

$$\left(\mathbf{A}^{(k)}\right)_{:,l} = \sqrt{\beta_l^{(k)}} \left(\mathbf{e}_l^{(k)} - \boldsymbol{\mu}^{(k)}\right) \in \mathbb{R}^d, \tag{4.8}$$

where $\mathbf{e}_l^{(k)}$ is the $l$-th selected item for $k$-th interest ranked by $\boldsymbol{\beta}^{(k)}$.

**4.3.2 Interest Importance Weight Modeling.** To better reflect the varying strengths of diverse user preferences, we aggregate them as a weighted mixture over interests. We apply an importance weight $w^{(k)}$ for each $k$-th interest with two strategies: hard aggregation and recency-based aggregation.

*Hard Aggregation.* This approach estimates the strength of each interest based on how frequently it is associated with items in the user's interaction sequence. Each item is assigned to the most relevant interest based on the item-interest relevance score $h(\mathbf{E}_u)$ in (4.1). The interest weight $w^{(k)}$ is then calculated as the proportion of items assigned to the $k$-th interest:

$$w^{(k)} = \frac{\#\text{ of items assigned to the } k\text{-th interest}}{\text{sequence length } (n)} \in \mathbb{R}.$$

*Recency-based Aggregation.* This strategy emphasizes interests that align closely with the user's recent preference, particularly well-suited to the sequential recommendation setting. Specifically, the interest weight is calculated based on the similarity between each interest distribution and the most recent item embedding, $\mathbf{e}_{\text{recent}}$ (*i.e.*, the latest item available) by

$$w^{(k)} = \left(\text{softmax}\left\{f\left(\mathbf{e}_{\text{recent}}, \mathcal{N}\left(\boldsymbol{\mu}^{(m)}, \mathbf{\Sigma}^{(m)}\right)\right)\right\}_{m=1}^{K}\right)_k,$$

where $f(\cdot,\cdot)$ is a similarity function such as dot-product using only mean or a distribution-aware similarity measure. The specific choice of this function is detailed in the following section.

**4.4 Prediction.** To predict the likelihood of a user interacting with a candidate item $\hat{y}_{\text{cand}}$, we comprehensively consider the user's multiple interests, while capturing the nuanced alignment between the item and the user's complex preference structure. To this end, we aggregate the matching scores between the item embedding and each interest distribution by:

$$\hat{y}_{\text{cand}} = \sum_{k=1}^{K} w^{(k)} f\left(\mathbf{e}_{\text{cand}}, \mathcal{N}\left(\boldsymbol{\mu}^{(k)}, \mathbf{\Sigma}^{(k)}\right)\right),$$

where $w^{(k)}$ is the importance of the user's $k$-th interest, $f(\cdot,\cdot)$ is a similarity measure, and $\mathbf{e}_{\text{cand}}$ is the candidate item embedding.

*Mahalanobis Distance for Probabilistic Matching.* For $f(\cdot,\cdot)$, we adopt a distribution-aware similarity function to fully leverage the distributional characteristics of user interests. In particular, we use the negative Mahalanobis distance [4], $-\mathcal{D}_{\text{M}}$, which captures the distance between the item embedding and the center of each interest while considering its dispersion:

$$\mathcal{D}_{\text{M}}\left(\mathbf{e}, \mathcal{N}\left(\boldsymbol{\mu}, \mathbf{\Sigma}\right)\right) = \sqrt{(\mathbf{e}-\boldsymbol{\mu})^\top \mathbf{\Sigma}^{-1} (\mathbf{e}-\boldsymbol{\mu})}. \tag{4.9}$$

Recall that each user in our model is represented as a mixture of Gaussian distributions, with each interest modeled by a learnable mean and covariance, while items are represented as point embeddings. This setting

requires a similarity measure that compares a point to a distribution, while accounting for the internal spread of each interest. The negative Wasserstein distance [22] is commonly used to measure distribution-aware similarity [15, 14, 43, 13], but this is not well-suited in our setting, as each item embedding is a single point, *i.e.*, equivalent to the Dirac delta distribution with zero covariance [1, 16]. In this case, the 2-Wasserstein distance $\mathcal{D}_{\mathrm{W}_2}(\cdot,\cdot)$ collapses to the sum of the squared Euclidean distance and the trace of the user interest covariance:

$$\mathcal{D}_{\mathrm{W}_2}\left(\mathbf{e}, \mathcal{N}\left(\boldsymbol{\mu}, \boldsymbol{\Sigma}\right)\right) = \|\mathbf{e}-\boldsymbol{\mu}\|_2^2 + \mathrm{Tr}\left(\boldsymbol{\Sigma}\right). \tag{4.10}$$

This compresses the entire covariance structure into a scalar, capturing only how broad the distribution is, not *where* or *how* it is spread across dimensions. To illustrate, consider two elliptical distributions $\mathcal{P}$ and $\mathcal{Q}$ with the same covariance magnitude and the same Euclidean distance to an item $i$. With the Wasserstein distance in Eq. (4.10), their similarity to $i$ would be the same. However, if the item $i$ lies within the core region of $\mathcal{P}$ but outside that of $\mathcal{Q}$, $\mathcal{P}$ should be intuitively more relevant to $i$. Wasserstein distance fails to distinguish this due to its lack of directional sensitivity.

In contrast, the Mahalanobis distance $\mathcal{D}_{\mathrm{M}}$ in (4.9) captures this difference through inverse covariance weighting. It penalizes deviations more severely along the tightly concentrated dimensions, while more gently along broad ones, distinguishing whether the item lies within or outside of the core region of the interest distribution. For instance, in the romantic-comedy example from section 1, suppose a user's romance interest is broad (high variance) while their comedy interest is narrow (low variance). If a candidate movie has slightly too much romance, the Mahalanobis distance will gently penalize that deviation; but if it lacks the precise comedic intensity the user expects, it will punish that discrepancy more significantly. This better reflects the internal structure of user interests and aligns with our probabilistic formulation. Empirical comparisons with other similarity measures are provided in §5.4.

**4.5 Training.** We train our model with three objectives: recommendation loss, regularization loss, and independence loss.

*Recommendation Loss* $\mathcal{L}_{\mathrm{rec}}$**.** We minimize cross-entropy for the predicted score of the target item relative to all candidate items:

$$\mathcal{L}_{\mathrm{rec}} = -\log\left(\frac{\exp(\hat{y}_{\mathrm{target}})}{\sum_{j\in\mathcal{I}}\exp(\hat{y}_j)}\right),$$

where $\hat{y}_{\mathrm{target}}$ is the matching score of the target positive item. In practice, more scalable approximations, such as sampled softmax [50] can be employed instead of the full softmax over $\mathcal{I}$.

*Regularization Loss* $\mathcal{L}_{\mathrm{reg}}$**.** To prevent the interest distributions from collapsing or drifting excessively, we impose a KL divergence regularization that pulls each interest distribution towards the standard normal distribution $\mathcal{N}(\mathbf{0},\mathbf{I})$, inspired by [41, 7]:

$$\mathcal{L}_{\mathrm{reg}} = \frac{1}{K}\sum_{k=1}^{K}\mathcal{D}_{\mathrm{KL}}\left(\mathcal{N}\left(\boldsymbol{\mu}^{(k)},\boldsymbol{\Sigma}^{(k)}\right)\,\middle\|\,\mathcal{N}\left(\mathbf{0},\mathbf{I}\right)\right).$$

*Independence Loss* $\mathcal{L}_{\mathrm{ind}}$**.** To encourage diversity among a user's multiple interests, we penalize pairwise similarities between different interest distributions. Specifically, we use the KL divergence between interest distributions for this:

$$\mathcal{L}_{\mathrm{ind}} = -\frac{\sum_{k\neq k'}\mathcal{D}_{\mathrm{KL}}\left(\mathcal{N}\left(\boldsymbol{\mu}^{(k)},\boldsymbol{\Sigma}^{(k)}\right)\,\middle\|\,\mathcal{N}\left(\boldsymbol{\mu}^{(k')},\boldsymbol{\Sigma}^{(k')}\right)\right)}{K(K-1)}.$$

*Overall Loss* $\mathcal{L}$*.* We train our model by minimizing $\mathcal{L} = \mathcal{L}_{\mathrm{rec}} + \lambda_{\mathrm{reg}}\mathcal{L}_{\mathrm{reg}} + \lambda_{\mathrm{ind}}\mathcal{L}_{\mathrm{ind}}$, where $\lambda_{\mathrm{reg}}$ and $\lambda_{\mathrm{ind}}$ control the strength of the regularization and independence losses, respectively.

**4.6 Time Complexity Analysis.** We compare the inference time complexity of our method with previous MIR models, assuming the most common dot-product scoring with best-matching interest selection setup in MIR [2, 50, 5, 53, 27, 35, 3]. Note that we omit the multi-interest extraction step, as our model is agnostic to the specific extraction procedure.

In general, MIR models compute dot product similarities between $K$ interest vectors and all items in $\mathcal{I}$, requiring $O(Kd|\mathcal{I}|)$, followed by interest-wise max-pooling, taking $O(K|\mathcal{I}|)$. Including the cost $O(|\mathcal{I}|\log|\mathcal{I}|)$ to sort all $|\mathcal{I}|$ scores, the overall complexity is given by $O(Kd|\mathcal{I}| + K|\mathcal{I}| + |\mathcal{I}|\log|\mathcal{I}|)$. On top of this, our method incorporates an additional distribution modeling step. We assume that the projected item embeddings $\mathbf{W}_\beta\mathbf{E}_u^\top$ in (4.5) are precomputed for all items at inference time. Constructing the $K$ interest-specific query vectors in (4.4) requires $O(Kd^2)$ operations, assuming a fixed-depth MLP whose hidden dimensionality is proportional to $d$. Computing the item-interest relevance weights in (4.5) for the $K$ interests over $n$ interactions then requires $O(Kdn)$ operations. With the diagonal covariance for practical efficiency, we directly compute its diagonal entries without constructing the full covariance matrix, which also requires $O(Kdn)$ operations. Therefore, the additional distribution modeling step requires $O(Kdn + Kd^2)$ operations. For scoring, the Mahalanobis distances are computed in $O(Kd|\mathcal{I}|)$, followed by weighted aggrega-

tion in $O(K|\mathcal{I}|)$. The ranking step remains the same. Hence, the overall complexity of our method is given by $O(\boldsymbol{Kdn + Kd^2} + Kd|\mathcal{I}| + K|\mathcal{I}| + |\mathcal{I}| \log |\mathcal{I}|)$. Comparing it with the general MIR cases, the additional cost $O(Kdn + Kd^2)$ remains marginal, since $n, d \ll |\mathcal{I}|$. This small additional overhead is well-justified by the significant performance gains achieved through distribution-based interest modeling. We provide an empirical comparison of inference time in §5.6.

## 5 Experiments.

We conduct extensive experiments to validate the effectiveness, generalizability, and design choices of PMIRec. We further examine its computational efficiency and provide a case study.

### 5.1 Experimental Settings.

In this section, we first outline the overall experimental setup.

*Datasets.* We take four widely-used benchmarks from Amazon review datasets [40]: Sports, Beauty, Toys, and Tools. All datasets are processed under the 5-core setting [2, 50, 35, 15]. The preprocessed dataset statistics are provided in Table 1.

Table 1: Dataset statistics after preprocessing.

| Dataset | # Users | # Items | # Actions | Avg. A./U. | Density |
|---|---|---|---|---|---|
| Sports | 35,598 | 18,357 | 296,337 | 8.32 | 0.045% |
| Beauty | 22,363 | 12,101 | 198,502 | 8.76 | 0.073% |
| Toys | 19,412 | 11,924 | 167,597 | 8.63 | 0.072% |
| Tools | 16,638 | 10,217 | 134,476 | 8.08 | 0.079% |

*Evaluation Protocol.* In line with prior studies [2, 50], we partition users into train, valid, and test sets with a ratio of 8:1:1, ensuring no user overlaps across splits. For evaluation, we use the first 80% of their interaction history to construct user representations and evaluate on the remaining 20%. This setup, often referred to as strong generalization [31], requires the model to generalize to entirely unseen users based solely on their partial histories. This makes it more challenging than leave-one-out [21, 44], where the same user's partial behavior is available at training [2, 32]. We report Hit Rate (HR) and Normalized Discounted Cumulative Gain (NDCG) with the top-$N$ matched items, where $N \in \{5, 10, 20\}$, following [19, 23].

*Baselines.* We compare with 3 categories of baselines: general SR (GRU4Rec [18] and SASRec [21]), probabilistic representation (PreferDiff [34], STOSA [15], and MacridVAE [38]), and other state-of-the-art MIR models (MIND [28], ComiRec-SA [2], ComiRec-DR [2], REMI [50], TiMiRec+ [48], DisMIR [11], and DMI-GNN [37]).

*Implementation Details.* We take ComiRec-SA [2] as our default backbone model for efficiency, unless noted otherwise. Among the proposed covariance modeling strategies (§4.3.1), we primarily use the diagonal covariance for its efficiency and scalability unless noted otherwise. We use Adam optimizer [24] with a learning rate from $\{0.001, 0.005\}$, training for up to 1 million iterations with early stopping if the validation NDCG@20 does not increase for 50 consecutive checks. We set the hidden dimension $d$ to 64, the batch size to 256, and the maximum sequence length to 20 for all models. For the overall comparison (§5.2), we perform grid search over the number of interests $K \in \{2, 4, 8\}$, regularization loss coefficient $\lambda_{\text{reg}} \in \{0, 0.1, 0.3, 0.5, 0.7, 0.9, 1\}$, and independence loss coefficient $\lambda_{\text{ind}} \in \{0, 10^{-8}, 10^{-6}, 10^{-4}, 10^{-2}\}$. For other experiments, we set $K = 2$ by default. We implement the baselines based on the official code and the code reproduced by REMI [50], and we follow the original papers for model-specific hyperparameter settings.

### 5.2 Overall Performance.

We compare in Table 2 the performance of PMIRec against baselines.

*Effectiveness of PMIRec.* PMIRec achieves the best performance across all datasets with substantial performance gain. This improvement is particularly notable in NDCG, indicating the model's superior ability to capture fine-grained user preferences and accurately rank relevant items. This verifies the effectiveness of probabilistic multi-interest modeling and our distribution-aware retrieval.

*Single vs. Multi-interest Modeling.* Multi-interest models generally outperform single-interest models (*e.g.*, SASRec), confirming the effectiveness of multiple interest embeddings. This advantage is prominent in advanced MIR models (*e.g.*, DMI-GNN, DisMIR, TiMiRec+) than earlier models like ComiRec and MIND, performing comparably to or worse than SASRec, likely due to their relatively shallow architectures, aligned with the observations from previous work [48, 11].

*Point vs. Distributional Modeling.* Among single-interest models, STOSA employs distributional representations, while SASRec adopts point embeddings. STOSA consistently outperforms SASRec at smaller top-$N$ (*e.g.*, HR@5, NDCG@5), suggesting that distributional modeling enables more precise ranking of highly relevant items. Similarly, our method consistently improves point-based MIR models, preserving the benefit of multi-interest modeling while enriching representational flexibility.

### 5.3 Generalizability of PMIRec.

As mentioned earlier, PMIRec is applicable to any MIR backbone. To verify this generalizability, we plug it into four representative MIR backbones: ComiRec-SA, ComiRec-DR, MIND, and TiMiRec+. We report results on Sports, Beauty, and Toys domains in Table 3.

PMIRec consistently improves performance across

Table 2: Overall performance comparison (%) on Sports, Beauty, Toys, and Tools datasets in Hit Rate and NDCG. The best performance is boldfaced and the second best is underlined.

| Dataset | Metric | GRU4Rec | SASRec | STOSA | MacridVAE | PreferDiff | MIND | Comi-SA | Comi-DR | REMI | TiMiRec+ | DisMIR | DMI-GNN | PMIRec (Ours) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Sports | HR@5 | 2.47 | 3.09 | 3.15 | 4.58 | 2.89 | 4.47 | 1.91 | 3.17 | 3.82 | 4.24 | 4.80 | 4.24 | **5.11** |
| | HR@10 | 4.19 | 5.34 | 4.97 | 7.42 | 3.68 | 6.71 | 3.31 | 4.75 | 6.35 | 7.39 | 7.33 | 6.26 | **8.12** |
| | HR@20 | 6.43 | 8.06 | 6.97 | 11.38 | 5.14 | 9.30 | 5.39 | 6.85 | 9.75 | 11.32 | 11.74 | 8.12 | **11.99** |
| | NDCG@5 | 1.79 | 1.91 | 2.17 | 3.06 | 2.14 | 2.65 | 1.24 | 1.76 | 2.13 | 2.38 | 3.09 | 2.80 | **3.47** |
| | NDCG@10 | 2.35 | 2.63 | 2.74 | 3.97 | 2.32 | 3.07 | 1.62 | 2.06 | 2.85 | 3.31 | 3.92 | 3.44 | **4.43** |
| | NDCG@20 | 2.91 | 3.30 | 3.20 | 4.92 | 2.82 | 3.55 | 1.98 | 2.56 | 3.50 | 4.12 | 4.95 | 4.41 | **5.39** |
| Beauty | HR@5 | 5.10 | 5.10 | 5.23 | 6.89 | 5.72 | 5.81 | 4.96 | 4.65 | 5.95 | 6.57 | 6.15 | 7.15 | **7.29** |
| | HR@10 | 7.51 | 8.18 | 8.14 | 11.09 | 6.88 | 9.12 | 8.14 | 7.24 | 10.06 | 10.15 | 10.61 | 10.37 | **11.35** |
| | HR@20 | 10.01 | 13.41 | 12.29 | 15.61 | 7.73 | 13.41 | 11.89 | 10.73 | 15.11 | 15.56 | 16.74 | 14.93 | **16.76** |
| | NDCG@5 | 3.40 | 3.11 | 3.38 | 4.47 | 4.11 | 3.45 | 3.04 | 2.80 | 3.47 | 3.91 | 3.31 | 4.61 | **4.82** |
| | NDCG@10 | 4.13 | 4.11 | 4.28 | 5.78 | 4.52 | 4.22 | 3.78 | 3.26 | 4.49 | 4.65 | 4.70 | 5.60 | **6.08** |
| | NDCG@20 | 4.76 | 5.38 | 5.24 | 6.86 | 4.72 | 5.20 | 4.60 | 4.02 | 5.59 | 5.99 | 6.30 | 6.70 | **7.35** |
| Toys | HR@5 | 3.60 | 4.99 | 5.15 | 6.13 | 5.20 | 5.56 | 4.84 | 2.63 | 6.08 | 6.90 | 5.77 | 6.85 | **7.52** |
| | HR@10 | 5.61 | 8.55 | 7.67 | 9.27 | 6.39 | 7.98 | 7.36 | 3.96 | 8.96 | 10.56 | 8.60 | 9.63 | **11.74** |
| | HR@20 | 8.29 | 13.85 | 10.92 | 12.31 | 7.42 | 11.95 | 11.07 | 6.08 | 13.29 | 14.78 | 13.18 | 13.65 | **16.07** |
| | NDCG@5 | 2.67 | 2.77 | 3.39 | 4.25 | 3.96 | 3.43 | 2.90 | 1.65 | 3.69 | 4.51 | 3.10 | 4.60 | **5.13** |
| | NDCG@10 | 3.32 | 3.89 | 4.21 | 5.27 | 4.39 | 3.82 | 3.42 | 1.80 | 4.23 | 5.12 | 3.91 | 5.52 | **6.48** |
| | NDCG@20 | 4.00 | 5.24 | 5.03 | 6.02 | 4.55 | 4.24 | 4.04 | 2.19 | 4.88 | 5.40 | 5.19 | 6.47 | **7.56** |
| Tools | HR@5 | 2.10 | 3.19 | 3.43 | 4.03 | 3.30 | 3.91 | 1.98 | 2.28 | 2.22 | 4.42 | 3.91 | 4.21 | **4.45** |
| | HR@10 | 3.49 | 5.77 | 5.47 | 6.55 | 4.99 | 4.75 | 4.15 | 3.67 | 4.27 | 7.48 | 6.73 | 6.91 | **8.05** |
| | HR@20 | 5.29 | 9.38 | 8.35 | 11.00 | 5.95 | 6.85 | 7.33 | 4.93 | 6.01 | 11.47 | 10.58 | 10.34 | **11.90** |
| | NDCG@5 | 1.24 | 1.94 | 2.09 | 2.60 | 2.04 | 2.32 | 1.11 | 1.55 | 1.34 | 2.61 | 2.28 | 2.58 | **3.03** |
| | NDCG@10 | 1.68 | 2.77 | 2.75 | 3.40 | 2.88 | 2.28 | 1.90 | 1.86 | 2.06 | 3.47 | 3.08 | 3.46 | **4.19** |
| | NDCG@20 | 2.11 | 3.70 | 3.49 | 4.52 | 3.28 | 2.68 | 2.75 | 2.34 | 2.49 | 4.34 | 4.05 | 4.31 | **5.13** |

Table 3: Generalizability analysis on MIR backbones.

| Backbone | +Ours | Sports N@10 | Sports N@20 | Beauty N@10 | Beauty N@20 | Toys N@10 | Toys N@20 |
|---|---|---|---|---|---|---|---|
| ComiRec-SA | ✗ | 1.62 | 1.98 | 3.78 | 4.60 | 3.42 | 4.04 |
| ComiRec-SA | ✓ | **4.43** | **5.39** | **6.08** | **7.35** | **6.48** | **7.56** |
| Relative Improvement | | 173% | 172% | 61% | 60% | 89% | 87% |
| ComiRec-DR | ✗ | 2.06 | 2.56 | 3.26 | 4.02 | 1.80 | 2.19 |
| ComiRec-DR | ✓ | **2.78** | **3.38** | **4.61** | **5.60** | **3.65** | **4.49** |
| Relative Improvement | | 35% | 32% | 41% | 39% | 103% | 105% |
| MIND | ✗ | 3.07 | 3.55 | 4.22 | 5.20 | 3.82 | 4.24 |
| MIND | ✓ | **3.23** | **3.89** | **5.13** | **5.88** | **4.70** | **5.85** |
| Relative Improvement | | 5% | 10% | 22% | 13% | 23% | 38% |
| TiMiRec+ | ✗ | 3.31 | 4.12 | 4.65 | 5.99 | 5.12 | 5.40 |
| TiMiRec+ | ✓ | **4.16** | **4.97** | **5.97** | **7.06** | **6.28** | **7.45** |
| Relative Improvement | | 26% | 21% | 28% | 18% | 23% | 38% |

Table 4: Ablation on covariance and interest weights.

| Covariance | Weight | Sports N@10 | Sports N@20 | Beauty N@10 | Beauty N@20 | Toys N@10 | Toys N@20 |
|---|---|---|---|---|---|---|---|
| ✗ | ✗ | 1.62 | 1.98 | 3.78 | 4.60 | 3.42 | 4.04 |
| ✗ | ✓ | 2.00 | 2.63 | 4.18 | 5.02 | 4.11 | 5.15 |
| ✓ | ✗ | 3.58 | 4.53 | 5.75 | 6.97 | 5.83 | 7.05 |
| ✓ (with $\boldsymbol{\alpha}$) | ✓ | 3.85 | 4.79 | 5.83 | 6.81 | 5.89 | 6.85 |
| ✓ | ✓ (recency) | 4.25 | 5.16 | 5.86 | 7.26 | **6.48** | **7.56** |
| ✓ | ✓ (hard) | **4.43** | **5.39** | **6.08** | **7.35** | 6.22 | 7.34 |

Table 5: Ablation on probabilistic matching.

| Train | Inference | Sports N@10 | Sports N@20 | Beauty N@10 | Beauty N@20 | Toys N@10 | Toys N@20 |
|---|---|---|---|---|---|---|---|
| $\boldsymbol{\mu}$ only (IP) | $\boldsymbol{\mu}$ only (IP) | 2.00 | 2.63 | 4.18 | 5.02 | 4.11 | 5.15 |
| $\boldsymbol{\mu}$ only (L2) | $\boldsymbol{\mu}$ only (L2) | 1.86 | 2.40 | 3.55 | 4.51 | 3.36 | 4.08 |
| Wasserstein | $\boldsymbol{\mu}$ only (L2) | 1.73 | 2.37 | 3.18 | 4.22 | 3.09 | 4.04 |
| Mahalanobis | $\boldsymbol{\mu}$ only (L2) | 2.94 | 3.72 | 4.85 | 5.91 | 4.52 | 5.50 |
| Wasserstein | Wasserstein | 1.94 | 2.47 | 3.72 | 4.46 | 3.65 | 4.30 |
| Mahalanobis | Mahalanobis | **4.43** | **5.39** | **6.08** | **7.35** | **6.48** | **7.56** |

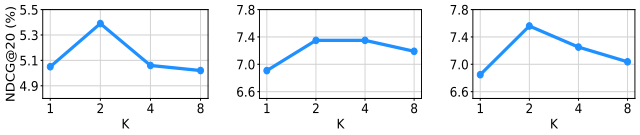
all backbones and datasets. It achieves average relative NDCG@20 gains of 59%, 33% and 67% on Sports, Beauty, and Toys, respectively, where the largest gain is observed with the ComiRec-SA backbone. These results demonstrate that PMIRec can be effectively integrated into diverse MIR architectures, enhancing their ability to model fine-grained user interests.

**5.4 Ablation Study.** We breakdown the contributions of the key components of PMIRec: interest covariance modeling (§4.3.1), interest importance weight modeling (§4.3.2), and probabilistic matching (§4.4).

*Effect of Covariance Modeling.* To assess the impact of modeling user interest dispersion, we compare variants of our method with different covariance settings in Table 4: no covariance at all (✗ in "Covariance"), covariance reusing contribution weights $\boldsymbol{\alpha}$ from interest mean extraction (4.1), and our proposed covariance (✓ in "Covariance").

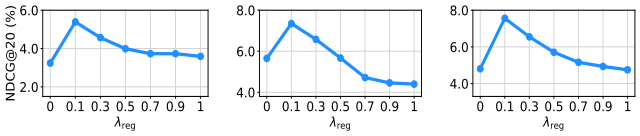
Removing covariance entirely leads to a significant performance drop, verifying the importance of modeling dispersion in user interests. The $\boldsymbol{\alpha}$-weighted covariance (with $\boldsymbol{\alpha}$) slightly underperforms our full model that learns dedicated relevance weights for covariance. Overall, these results demonstrate that learning user-specific covariance with relevance-aware weighting is crucial for effective personalization.

*Effect of Weighted Mixture Modeling.* We also examine the effect of mixture modeling for aggregating multiple interests in Table 4. Using only the best matching interest (✗ in "Weight"), the most common strategy in existing MIR models, results in inferior performance compared to our approach in general. This demonstrates the benefit of adaptively leveraging the multi-interest structure to better capture diverse user preferences. The two interest weighting strategies, recency-based and hard aggregation, generally perform well, exhibiting slightly different competence by datasets. Moreover, since variants that include either covariance modeling or interest weight modeling outperform the baseline without both, each component independently contributes to improved recommendation.



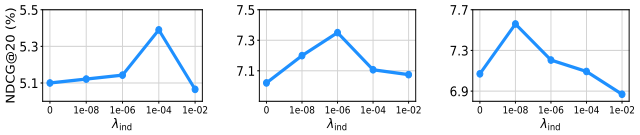

(a) Sports (b) Beauty (c) Toys

Figure 4: Performance variation with respect to $K$.




(a) Sports (b) Beauty (c) Toys

Figure 5: Performance variation with respect to $\lambda_{\text{reg}}$.




(a) Sports (b) Beauty (c) Toys

Figure 6: Performance variation with respect to $\lambda_{\text{ind}}$.

*Effect of Probabilistic Matching.* We empirically evaluate the effect of various similarity metrics between user distributions and item point embeddings in Table 5. We consider two types of similarity metrics: 1) point-based ones utilizing only the means, including inner product (IP) and negative L2 distance, and 2) distribution-aware ones, including negative Wasserstein and Mahalanobis distances. Since Mahalanobis and Wasserstein distances are built upon L2 distance, L2 can be interpreted as a covariance-ablated variant of these metrics.

Overall, the distribution-aware Mahalanobis distance achieves the best performance. Its significant improvement over L2 demonstrates the benefit of incorporating user-specific covariance. Notably, Mahalanobis distance consistently outperforms Wasserstein distance, which sometimes underperforms even the simpler point-based methods, verifying our claim in §4.4.

We also observe that using Mahalanobis distance at both training and inference stages is most effective, highlighting the benefit of consistent distributional modeling for stable generalization. Adopting Mahalanobis distance only at training while using L2 at inference somewhat degrades the performance, but considering its second-best performance, this can be an attractive alternative when the inference latency is particularly critical.

**5.5 Effect of Hyperparameters.** We analyze how key modeling choices and hyperparameters affect the performance of PMIRec, including the interest number $K$, the weights for the loss terms ($\lambda_{\text{reg}}$, $\lambda_{\text{ind}}$), and the effect of different covariance construction methods.

*Number of User Interests $K$.* We analyze how the number of interests $K \in \{1, 2, 4, 8\}$ affects model performance. As shown in Figure 4, multi-interest modeling ($K \geq 2$) generally outperforms the single-interest variant ($K = 1$), while the best $K$ varies across datasets. A larger $K$ tends to offer greater capacity to capture diverse preferences, but too high $K$ can lead to overfitting, learning noisy behaviors resulting in degraded performance. Interestingly, even with $K = 1$, ours consistently outperforms point-based single-interest models such as SASRec (see Table 2). This suggests the benefit of distribution modeling even in the single interest setting.

*Loss Weights $\lambda_{reg}$, $\lambda_{ind}$.* As illustrated in Figure 5-6, both losses exhibit a similar trend, where the performance increases with higher weights, peaks at a moderate value, then declines. This reflects the trade-off between guiding the learning process and allowing representational flexibility. Notably, removing either loss ($\lambda = 0$) leads to a significant performance drop compared to the peak, highlighting the important role of these losses. Still, even without them, ours outperforms the backbone ComiRec-SA, confirming that the performance gains are not only from the auxiliary losses but also from the core probabilistic multi-interest design.

*Covariance Modeling Strategies.* We compare the covariance modeling strategies introduced in §4.3.1: full covariance, diagonal covariance, and low-rank approximations, including learned projection (4.7) and top-$r$ item selection (4.8). We set $r = 2$, selected from the search space $r \in \{1, 2, 4\}$ via cross-validation. Under our 5-core setting, the minimum training sequence length is 4, limiting the maximum feasible rank to $r = 4$.

Table 6 shows that the full covariance strategy consistently performs the worst among the covariance-based methods, likely due to its instability in estimating high-dimensional covariances from limited interaction data. Still, it outperforms the one without covariance, confirming the benefit of probabilistic user interest modeling. Between the two low-rank approaches, results vary across datasets with no clear winner, suggesting that the optimal choice may depend on the dataset characteristics. Diagonal covariance achieves comparable or superior performance than other strategies, demonstrating that this simplified yet expressive modeling of interest dispersion is not only computationally efficient but also effective.

**5.6 Computational Efficiency.** We compare the actual per-user inference time across models, including SASRec, MIR models (ComiRec-SA/DR, MIND and TiMiRec+), and our method applied to those MIR backbones. All results are measured on a single NVIDIA

Table 6: Comparison for different covariance forms.

| Covariance | Sports | | Beauty | | Toys | |
|---|---|---|---|---|---|---|
| | N@10 | N@20 | N@10 | N@20 | N@10 | N@20 |
| ✗ | 1.66 | 2.18 | 3.86 | 4.92 | 2.86 | 3.46 |
| full | 3.27 | 3.97 | 4.60 | 5.50 | 4.57 | 5.62 |
| low-rank (proj.) | 4.12 | 4.93 | 5.91 | 7.03 | 6.14 | 7.25 |
| low-rank (top-$r$) | **4.64** | **5.46** | 5.68 | 6.88 | 5.55 | 6.74 |
| diagonal | 4.43 | 5.39 | **6.08** | **7.35** | **6.48** | **7.56** |

A6000 GPU with 4 CPU cores, and we report the average over five runs in Table 7.

Integrating our method into existing MIR baselines (✓ on "+Ours") leads to notable performance improvements with minimal increase in inference time. On ComiRec-SA, for example, our method improves NDCG@20 by 106% on average, while per-user inference time increases by only 0.92%. This demonstrates that our method can be seamlessly integrated into existing models to enhance performance without compromising efficiency.

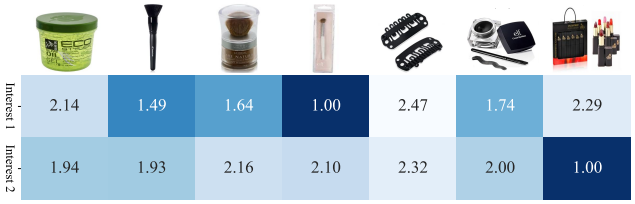
Table 7: Per-user inference time comparison (ms).

| Model | +Ours | Sports | | Beauty | | Toys | |
|---|---|---|---|---|---|---|---|
| | | Time | N@20 | Time | N@20 | Time | N@20 |
| SASRec | ✗ | 1.2812 | 3.30 | 0.8001 | 5.38 | 0.7877 | 5.24 |
| ComiRec-SA | ✗ | 1.2927 | 1.98 | 0.8006 | 4.60 | 0.7890 | 4.04 |
| ComiRec-SA | ✓ | 1.2932 | 5.39 | 0.8080 | 7.35 | 0.7915 | 7.56 |
| ComiRec-DR | ✗ | 1.2895 | 2.56 | 0.8067 | 4.02 | 0.7900 | 2.19 |
| ComiRec-DR | ✓ | 1.2999 | 3.38 | 0.8107 | 5.60 | 0.7954 | 4.49 |
| MIND | ✗ | 1.2940 | 3.55 | 0.8042 | 5.20 | 0.7903 | 4.24 |
| MIND | ✓ | 1.2985 | 3.89 | 0.8083 | 5.88 | 0.7935 | 5.85 |
| TiMiRec+ | ✗ | 1.3233 | 4.12 | 0.8241 | 5.99 | 0.8115 | 5.40 |
| TiMiRec+ | ✓ | 1.3380 | 4.97 | 0.8301 | 7.06 | 0.8224 | 7.45 |

**5.7 Case Study.** To qualitatively validate the interpretability of PMIRec, we conduct a case study on a couple of sample users in Amazon Beauty test set. We visualize the relationship between the user's interaction sequence and the learned interest distributions by plotting the Mahalanobis distance. Since only the relative distance matters in ranking-based recommendation, these distances are scaled relative to the minimum distance within each interest's sequence. In addition, we report the determinant of the covariance matrix for each interest to quantify the volumetric spread.

As depicted in Figure 7a, the model successfully disentangles intents of the user 22311 into two distinct interests. Interest 1 (top row) exhibits a sharp focus on nail artistry, showing strong affinity exclusively with the nail polish and nail art stamping plates. This high specificity results in a minimal determinant (≈ 1.00). In contrast, Interest 2 (bottom row) captures a broad spectrum of daily personal care essentials, encompassing facial makeup, hair care, and hygiene products. This heterogeneity is quantitatively reflected in its larger covariance determinant (≈ 6.95).


(a) User 22311




(b) User 20410

Figure 7: Case Studies for User 22311 and User 20410

Further validating this geometric interpretability, the case of user 20410 in Figure 7b highlights the model's capacity to isolate distinct sub-preferences while filtering out irrelevant categories. Interest 1 (top row) aggregates a composite preference for facial cosmetics and application tools, showing high affinity with makeup brushes, and cream eyeliner. Reflecting this combination of heterogeneous items, this interest yields a large covariance determinant (≈ 39.05). In contrast, Interest 2 (bottom row) demonstrates a singular, sharp focus on lip color products, responding strongly only to the lipstick set. This exclusivity results in a minimal determinant (≈ 1.00). Crucially, both interests maintain large distances from hair styling products (gel and clips), confirming that the model successfully disentangles makeup-related intents from hair-care needs.

**6 Conclusion.** We present PMIRec, a novel probabilistic framework for multi-interest recommendation that models each user as a mixture of interest distributions. By capturing both the diversity across distinct interests and internal dispersion of user preferences, it enables more expressive and fine-grained user representations. PMIRec seamlessly integrates into existing multi-interest recommenders, improving performance with minimal computational overhead. Through extensive experiments, we demonstrate its effectiveness. An interesting future direction would be to model items as distributions as well, enabling more expressive user-item interactions by capturing multi-faceted item semantics.

**Acknowledgements.** This work was supported by NRF (RS-2021-NR05515, RS-2024-00336576, RS-2023-0022663) and IITP (RS-2022-II220264, RS-2024-00353131) grants from Korean government.

## References


[1] E. Bernton, P. E. Jacob, M. Gerber, and C. P. Robert, *On parameter estimation with the wasserstein distance*, Information and Inference: A Journal of the IMA, 8 (2019), pp. 657–676.

[2] Y. Cen, J. Zhang, X. Zou, C. Zhou, H. Yang, and J. Tang, *Controllable multi-interest framework for recommendation*, in KDD, 2020.

[3] Z. Chai, Z. Chen, C. Li, R. Xiao, H. Li, J. Wu, J. Chen, and H. Tang, *User-aware multi-interest learning for candidate matching in recommenders*, in SIGIR, 2022.

[4] M. P. Chandra, *On the generalised distance in statistics*, Sankhya A, 80 (2018), pp. 1–7.

[5] G. Chen, X. Zhang, Y. Zhao, C. Xue, and J. Xiang, *Exploring periodicity and interactivity in multi-interest framework for sequential recommendation*, in IJCAI, 2021.

[6] J. Chen, Y. Xu, and Y. Jiang, *Unlocking the power of diffusion models in sequential recommendation: A simple and effective approach*, in KDD, 2025.

[7] S. Chun, S. Oh, R. S. De Rezende, Y. Kalantidis, and D. Larlus, *Probabilistic embeddings for cross-modal retrieval*, in CVPR, 2021.

[8] P. Covington, J. Adams, and E. Sargin, *Deep neural networks for youtube recommendations*, in RecSys, 2016.

[9] X. Dong, X. Song, T. Liu, and W. Guan, *Prompt-based multi-interest learning method for sequential recommendation*, IEEE TPAMI, (2025).

[10] L. Dos Santos, B. Piwowarski, and P. Gallinari, *Gaussian embeddings for collaborative filtering*, in SIGIR, 2017.

[11] Y. Du, Z. Wang, Z. Sun, Y. Ma, H. Liu, and J. Zhang, *Disentangled multi-interest representation learning for sequential recommendation*, in KDD, 2024.

[12] C. Eckart and G. Young, *The approximation of one matrix by another of lower rank*, Psychometrika, 1 (1936), pp. 211–218.

[13] Z. Fan, Z. Liu, H. Peng, and P. Yu, *Mutual Wasserstein discrepancy minimization for sequential recommendation*, in WWW, 2023.

[14] Z. Fan, Z. Liu, S. Wang, L. Zheng, and P. S. Yu, *Modeling sequences as distributions with uncertainty for sequential recommendation*, in CIKM, 2021.

[15] Z. Fan, Z. Liu, Y. Wang, A. Wang, Z. Nazari, L. Zheng, H. Peng, and P. S. Yu, *Sequential recommendation via stochastic self-attention*, in WWW, 2022.

[16] C. Frogner, C. Zhang, H. Mobahi, M. Araya, and T. Poggio, *Learning with a wasserstein loss*, in NeurIPS, 2015.

[17] G. Hu, A. Z. Liu, W. Mao, J. Wu, X. Yang, X. Li, L. Hu, H. Li, K. Gai, X. Wang, et al., *Fading to grow: Growing preference ratios via preference fading discrete diffusion for recommendation*, in NeurIPS, 2025.

[18] D. Jannach and M. Ludewig, *When recurrent neural networks meet the neighborhood for session-based recommendation*, in RecSys, 2017.

[19] K. Järvelin and J. Kekäläinen, *Cumulated gain-based evaluation of ir techniques*, ACM Trans. on Information Systems, 20 (2002), pp. 422–446.

[20] Y. Jeon, J. Kim, and J. Lee, *Local large language models for recommendation*, in CIKM, 2025.

[21] W.-C. Kang and J. McAuley, *Self-attentive sequential recommendation*, in ICDM, 2018.

[22] L. V. Kantorovich, *Mathematical methods of organizing and planning production*, Management science, 6 (1960), pp. 366–422.

[23] G. Karypis, *Evaluation of item-based top-n recommendation algorithms*, in CIKM, 2001.

[24] D. Kingma and J. Ba, *Adam: a method for stochastic optimization*, arXiv:1412.6980, (2014).

[25] P. P.-H. Kung, Z. Fan, T. Zhao, Y. Liu, Z. Lai, J. Shi, Y. Wu, J. Yu, N. Shah, and G. Venkataraman, *Improving embedding-based retrieval in friend recommendation with ann query expansion*, in SIGIR, 2024.

[26] S. Lee, K. Yeo, E. Kim, J. Kim, C. Kim, Y. Jeon, J. Kim, and J. Lee, *Mixture of conditional attention for multimodal fusion in sequential recommendation*, in PAKDD, 2025.

[27] B. Li, B. Jin, J. Song, Y. Yu, Y. Zheng, and W. Zhou, *Improving micro-video recommendation via contrastive multiple interests*, in SIGIR, 2022.

[28] C. Li, Z. Liu, M. Wu, Y. Xu, H. Zhao, P. Huang, G. Kang, Q. Chen, W. Li, and D. L. Lee, *Multi-interest network with dynamic routing for recommendation at tmall*, in CIKM, 2019.

[29] W. Li, R. Huang, H. Zhao, C. Liu, K. Zheng, Q. Liu, N. Mou, G. Zhou, D. Lian, Y. Song, et al., *DimeRec: a unified framework for enhanced sequential recommendation via generative diffusion models*, in WSDM, 2025.

[30] Z. Li, A. Sun, and C. Li, *DiffuRec: A diffusion model for sequential recommendation*, ACM Trans. on Information Systems, 42 (2023), pp. 1–28.

[31] D. Liang, R. G. Krishnan, M. D. Hoffman, and T. Jebara, *Variational autoencoders for collaborative filtering*, in WWW, 2018.

[32] D. Liang, R. G. Krishnan, M. D. Hoffman, and T. Jebara, *Variational autoencoders for collaborative filtering*, in WWW, 2018.

[33] J. Liu, Q. Chen, J. Xu, J. Li, B. Li, and S. Xu, *A unified search and recommendation framework based on multi-scenario learning for ranking in E-commerce*, in SIGIR, 2024.

[34] S. Liu, A. Zhang, G. Hu, H. Qian, and T.-s. Chua, *Preference diffusion for recommendation*, in ICLR, 2025.

[35] Y. Liu, X. Zhang, M. Zou, and Z. Feng, *Attribute simulation for item embedding enhancement in multi-interest recommendation*, in WSDM, 2024.

[36] M. Luo, Y. Li, and C. Lin, *Enhancing sequential recommendation with global diffusion*, in AAAI, 2025.

[37] M. Lv, X. Liu, and Y. Xu, *Dynamic multi-interest graph neural network for session-based recommendation*, in AAAI, 2025.

[38] J. Ma, C. Zhou, P. Cui, H. Yang, and W. Zhu, *Learning disentangled representations for recommendation*, in NeurIPS, 2019.

[39] W. Mao, Z. Yang, J. Wu, H. Liu, Y. Yuan, X. Wang, and X. He, *Addressing missing data issue for diffusion-based recommendation*, in SIGIR, 2025.

[40] J. McAuley, C. Targett, Q. Shi, and A. Van Den Hengel, *Image-based recommendations on styles and substitutes*, in SIGIR, 2015.

[41] S. J. Oh, K. P. Murphy, J. Pan, J. Roth, F. Schroff, and A. C. Gallagher, *Modeling uncertainty with hedged instance embedding*, in ICLR, 2019.

[42] Y. Qu and H. Nobuhara, *Intent-aware diffusion with contrastive learning for sequential recommendation*, in SIGIR, 2025.

[43] E. Schonfeld, S. Ebrahimi, S. Sinha, T. Darrell, and Z. Akata, *Generalized zero-and few-shot learning via aligned variational autoencoders*, in CVPR, 2019.

[44] F. Sun, J. Liu, J. Wu, C. Pei, X. Lin, W. Ou, and P. Jiang, *BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer*, in CIKM, 2019.

[45] Q. Tan, J. Zhang, J. Yao, N. Liu, J. Zhou, H. Yang, and X. Hu, *Sparse-interest network for sequential recommendation*, in WSDM, 2021.

[46] Y. Tian, J. Chang, Y. Niu, Y. Song, and C. Li, *When multi-level meets multi-interest: A multi-grained neural model for sequential recommendation*, in SIGIR, 2022.

[47] L. Vilnis and A. McCallum, *Word representations via gaussian embedding*, in ICLR, 2015.

[48] C. Wang, Z. Wang, Y. Liu, Y. Ge, W. Ma, M. Zhang, Y. Liu, J. Feng, C. Deng, and S. Ma, *Target interest distillation for multi-interest recommendation*, in CIKM, 2022.

[49] W. Wang, Y. Xu, F. Feng, X. Lin, X. He, and T.-S. Chua, *Diffusion recommender model*, in SIGIR, 2023.

[50] Y. Xie, J. Gao, P. Zhou, Q. Ye, Y. Hua, J. B. Kim, F. Wu, and S. Kim, *Rethinking multi-interest learning for candidate matching in recommender systems*, in RecSys, 2023.

[51] Y. Xie, P. Zhou, and S. Kim, *Decoupled side information fusion for sequential recommendation*, in SIGIR, 2022.

[52] Z. Yang, J. Wu, Z. Wang, X. Wang, Y. Yuan, and X. He, *Generate what you prefer: Reshaping sequential recommendation via guided diffusion*, in NeurIPS, 2023.

[53] S. Zhang, L. Yang, D. Yao, Y. Lu, F. Feng, Z. Zhao, T.-S. Chua, and F. Wu, *Re4: Learning to re-contrast, re-attend, re-construct for multi-interest recommendation*, in WWW, 2022.

## Appendix

**A Expanded Related Work.** This section presents an extended version of the related work discussed in the main paper.

*Multi-Interest Recommendation (MIR).* To effectively capture users' diverse preferences, MIR models [2, 28, 50, 48, 53, 27, 5, 46, 11, 35, 45, 37] explicitly represent multiple interests of each user. Early approaches such as MIND [28] and ComiRec [2] extract multiple user embeddings through dynamic routing and multi-head self-attention, offering more granular representations of user intent. Building upon this, recent models focus on disentangling user interests. For example, CMI [27] applies contrastive learning with implicit category guidance, and DisMIR [11] partitions the item space into semantically coherent clusters, allowing each interest embedding to reflect a distinct facet of user preference.

Complementing these architectural improvements, other works focus on how MIR models are trained and used. REMI [50] reveals that uniform negative sampling and routing collapse limit expressiveness. It addresses these by interest-aware hard negative mining and routing regularization. DMI-GNN [37] introduces another dynamic regularization strategy, adjusting the differentiation among interest representations according to sequence length. TiMiRec [48] infers user intent by estimating context-dependent weights over multiple interests through a dedicated predictor.

In spite of enhanced interest modeling, however, these methods rely on point embeddings, lacking the capacity to represent internal dispersion within individual user interests. While PoMRec [9] introduces a distributional perspective by directly adding dispersion to interest embeddings, this simple element-wise addition offers only a limited approximation of probabilistic representations.

*Probabilistic Representations in Recommender Systems.* While most RS models represent users and items as fixed point vectors, recent works have proposed modeling them as probability distributions to enable uncertainty modeling. GER [10] models each user and item as a Gaussian distribution, improving recommendation performance in scenarios with high-uncertainty, such as cold-start recommendations. DT4SR [14] extends probabilistic modeling into the sequential recommendations, representing each interaction sequence as a Gaussian distribution learned via separate transformers. STOSA [15] incorporates uncertainty directly into the attention mechanism, replacing dot-product attention with Wasserstein self-attention. While these models have taken a step forward in capturing uncertainty in recommendation, probabilistic modeling remain underexplored, particularly in representing the multi-faceted nature of user preferences. These models still rely on unimodal user representations with limited expressiveness. Exceptionally, MacridVAE [38] models multiple interests as probability distributions by learning Gaussian posteriors for concept-specific user factors; however, its matching is still point-wise, operating on mean vectors.

More recently, generative diffusion models [34, 42, 39, 6, 36, 17] have been employed to capture complex user preference distributions. DiffRec [49] formulates recommendation as a conditional generation task, learning the distribution of user interactions. Adapting this to sequential settings, DiffuRec [30] and DreamRec [52] generate future item sequences or latent representations of user intent. PreferDiff [34] formulates the ranking objective into a generative framework. DimeRec [29] adopts a multi-interest framework that extracts a set of interest embeddings as high-level stationary abstractions of user intent. It then uses a diffusion process conditioned on these embeddings to synthesize the final user embedding. These diffusion-based methods aim to learn the distribution of user preferences more flexibly than typical MIR models, without imposing parametric forms like Gaussian mixture in our method. However, due to the iterative nature of the denoising process, diffusion-based recommenders incur substantial inference latency, which often becomes a practical drawback. Our method, on the other hand, achieves both reasonable flexibility in capturing preference distribution and practical inference time.