Unified Metric Learning for Personalized Fashion Recommendations with Fast User Adaptation

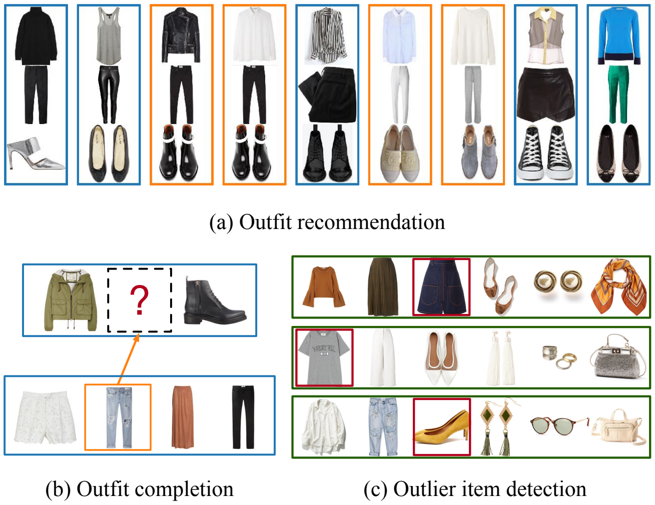
Zhi Lu, Yang Hu, Cong Yu, Yan Chen, *Senior Member* and Bing Zeng, *Fellow*

***Abstract*— Personlized fashion recommendation aims to model outfit composition quality and recommend compatible item combinations for users. However, existing studies often address outfit-level and item-level recommendations as separate problems, resulting in fragmented formulations. In this work, we revisit these problems from a unified mathematical perspective and extend fashion recommendation into a coherent family of composition tasks. These tasks, including outfit recommendation, outfit completion, outlier item detection, and new user profiling, are formulated within a common probabilistic framework that differs only in search constraints or conditioning. Based on this unified view, we propose a metric learning framework that embeds items, outfits, and users into a shared space where outfit compatibility is modeled as the probability of co-occurrence under composition utility. The learned embeddings provide consistent evaluation across tasks, enable personalized recommendation, and support fast user adaptation through plug-and-play user parameters without fine-tuning. Extensive experiments demonstrate that our approach achieves state-of-the-art performance across multiple composition tasks and establishes a unified foundation for generalizable and adaptive fashion recommendation.**

***Index Terms*—Fashion composition, item recommendation, outlier detection, representation learning, metric learning**

I. Introduction

UNDERSTANDING the relationships between fashion items within an outfit is a fundamental challenge in fashion recommendation [1]–[5]. Recent advancements have primarily focused on developing models that effectively capture compatibility among fashion items [6]–[10], especially in the outfit-composition domain [11]. Unlike complementary item recommendation [12]–[14], the relationships within an outfit involve more complex, high-order and implicit dependencies [15]. Various approaches have been proposed to capture such multiway relationships. For instance, FPITF [1] factorized the compatibility as pairwise interactions between fashion items and users, Bi-LSTM [2] treated the entire outfit as a whole sequence, while NGNN [16] additionally constructed a fashion graph. These approaches aim to better capture the underlying dependence and interaction among multiple fashion items, which is the crucial for outfit recommendation.


(a) Outfit recommendation

(b) Outfit completion

(c) Outlier item detection

Fig. 1. Examples of fashion outfit composition tasks. (a) Outfit recommendation. The outfits in orange boxes are positive outfits and those in blue boxes are negative outfits. (b) Outfit completion. The outfit in the blue box is incomplete. The item in the orange box matches the incomplete outfit best among a set of candidate items. (c) Outlier item detection. The outlier items are indicated in red boxes.

The outfit composition problem [11] involves developing a utility function that measures the compatibility of fashion items, guiding the selection of the most compatible outfits for users, as illustrated in Fig. 1 (a). The recommendation occasionally transitions from the entire outfit collection to a restricted subset, where each pair of outfits differs by just one item. For instance, as shown in Fig. 1 (b) and (c), two tasks emerge: the outfit completion task [2] recommends the best item to complete an outfit, and the outlier detection task [17] identifies the item disrupting the outfit's harmony most. These tasks aimed at outfit refinement are subtly distinct from general outfit recommendation tasks. When combined, they can assist users in refining their outfits piece by piece. For internal relationships, outfit recommendation prioritizes the overall pattern of an outfit, including style and occasion, while outfit refinement emphasizes the impact of individual items in outfits. Therefore, applying uniform item relationships across all tasks could be sub-optimal.

However, the item relationships are usually learned for outfit recommendation, with the completion task being treated as a means of model evaluation or a training strategy [2]. As a result, the difference between current outfit recommendation approaches largely lies in their way of modeling outfits and approaching the objective function. For example, decompos-

Zhi Lu and Yan Chen are with the School of Cyber Science and Technology, University of Science and Technology of China, Hefei 230026, China. (e-mail: zhilu@ustc.edu.cn; eecyan@ustc.edu.cn)

Cong Yu is with the Institute of Electronic Engineering, China Academy of Engineering Physics, Mianyang 621900, China. (e-mail: congyu@std.uestc.edu.cn)

Yang Hu is with the School of Information Science and Technology, University of Science and Technology of China, Hefei 230026, China. (e-mail: eeyhu@ustc.edu.cn)

Bing Zeng are with the School of Information and Communication Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China. (e-mail: eezeng@uestc.edu.cn)

Corresponding author: Yang Hu. This work was supported by the National Natural Science Foundation of China under Grant No. 62302471 and 62172381.

ing the multi-way relationship into pairwise interactions is a factorization way to formulate the compatibility [8], [9] that usually produce compact item representations for outfit completion task, but may overlook high-order relationships between items. Conversely, outfit-level strategies [2], [18], [19], [19], especially the graph-based approaches, view the outfit as an integrated entity to uncover high-order item interrelations. However, these approaches do not yield compact representations of outfits or items, rendering them less adaptable to outfit completion tasks. Therefore, recent research [18], [20] has tackled the outfit completion task by either constructing an additional network on top of the outfit recommendation model or through fine-tuning for better performance.

Additionally, performance can vary depending on the approach to optimize the compatibility. Learning the model with an outfit ranking approach can produce more discriminating results for outfit recommendation because it takes into account the distribution of all negative outfits. In contrast, emphasizing compatibility differences at the item level can lead to more discriminating results for outfit completion task, but may result in sub-optimal performance for outfit recommendation tasks. Although recent methods have significantly improved in measuring compatibility between fashion items, they often overlook the subtle differences among these tasks. The outlier item detection task, which hasn't been well studied, follows a similar trend of oversight.

Furthermore, in personalized cases [1], [9], [21]–[23], it is important to capture both the general preferences that reflect overall trends and patterns in the data, as well as the unique preferences of individual users that deviate from the general trends. With contemporary models growing rapidly in size [24], the ability to quickly adapt and accommodate new users with limited feedback [22], [25] becomes crucial, especially when user privacy is paramount and fine-tuning a large deep model is infeasible on user devices.

Hence, developing a unified learning framework capable of integrating different compatibility measures for outfits, embracing a broad spectrum of user preferences, while also being able to swiftly adjust to a new user with limited data becomes even challenging and also important for modern recommendation system.

To address these limitations and craft a model that can swiftly adaptable to new users, we propose a unified framework for learning compact outfit embeddings within the same metric space of items and users. The metric space is learned to maximize the probability of compatible outfits, with items being embedded to facilitate the outfit refinement and user profiles optimized for personalized recommendation and rapid adaptation. We treat outfits as set modeling problems [26]–[28] and use attention mechanisms [29] to capture high-order relations among the items. The learned outfit embeddings are permutation invariant to item orders and can be shared across different tasks. We summary our contribution as follows:

(1) We emphasize that the tasks of completing outfits and detecting outliers are different from outfit recommendation, a distinction that previous studies often overlook. We provide detailed definitions for each task and present them from a unified probabilistic viewpoint.

(2) We introduce a novel approach that learns a compact embedding for outfits in the same space as items and users. This method allows us to address each task within the unified framework and enables improved performance.

(3) Leveraging our unified framework, we address fast new user profiling by predicting plug-and-play user preferences with limited data, bypassing fine-tuning with a closed-form solution that exceeds the performance of conventional fine-tuning approaches.

While certain aspects of our study have been discussed in [17], [22], we thoroughly augment these works, i.e. extending them into a general framework for different outfit composition tasks, incorporating category information during representation learning, conducting more extensive experiments on multiple tasks and datasets.

II. Related Works

Fashion outfit recommendation [4], [24], [30], [31] has gained significant attention, focusing on uncovering complex item relationships. While deep learning is commonly used for feature extraction, existing methods fall into three categories based on interaction modeling: factorization-based, neural network-based, and graph-based approaches.

A. Factorization-based approaches

The factorization-based approaches [8], [9], [21], [32] mainly decomposes the multi-way compatibility into explicit interactions, such as pairwise similarities between items. These methods follow the conventional approach in recommendation to impose an explicit structure prior [33]. For example, Hu et al. [1] utilized tensor factorization [34] to model the interactions between users and fashion items. In [12], [13], a single latent space was used for measuring the compatibility. However, the relationships between a pair of items in different aspects (e.g. color, pattern, category) can be quite different or even contradictory.

To address the limitation, Veit et al. [14] proposed the conditional similarity networks (CSN) that compared different items in different conditions to improve the performance. Tan et al. [7] improved the CSN method by learning multiple conditional embeddings and using an attention mechanism to discover the relative importance of different conditions. Vasileva et al. [6] further enhance the CSN method by treating each category pair as a condition. To extend the fashion recommendation system with side information, Yang et al. [35] also proposed a translation-based network that learned the compatibility with the category-specific relations.

More recent approaches [36], [37] leverage attention mechanisms to dynamically weight feature interactions, enabling the model to learn more fine-grained and representative attentive features for users and items, thereby improving preference prediction. In general, fashion recommendation systems continue to build upon matrix factorization and collaborative filtering as foundational techniques, while incorporating deep learning methods to capture more complex patterns in fashion data.

Despite these advances, existing factorization-based models primarily decompose compatibility into explicit interactions, and often treat items, users, and outfits separately. A unified factorized framework that jointly models these entities remains an open challenge for achieving more holistic and context-aware fashion recommendations.

B. Neural network-based approaches

The neural network-based approaches [38]–[42] not only employ deep learning to learn advanced feature representations, but also predict compatibility through implicit neural architectures. Unlike traditional factorization-based models that rely on explicit interaction modeling, these methods capture the complex and nonlinear relationships among items and users in a more flexible manner. Consequently, they have become central to personalized fashion outfit recommendation, leveraging sophisticated network structures to model the nuanced interplay between compatibility and individual preferences.

These approaches typically extracted visual or textual features and then predicted compatibility scores directly [40], [43]. For example, Polania et al. [44] used fully connected layers for compatibility computation, while Tangseng et al. [45] concatenated all item features and applied a binary classifier. Han et al. [2] represented an outfit as a sequence and employed a bidirectional LSTM to learn compatibility, though such sequential modeling is order-sensitive and not permutation-invariant. Li et al. [46] addressed this limitation via an instance pooling mechanism using RNNs to aggregate item features. Transformer-based architectures were later introduced. For instance, the Personalized Outfit Generation (POG) model [47] used a Transformer [29] to decode users' click histories into outfits, treating outfit completion as a generation task [48]. Other methods employed Vision Transformer (ViT)–based embeddings [20].

More recent systems further leverage large multimodal models (LMMs) to jointly encode visual and textual modalities, thereby enhancing the robustness of compatibility prediction [49], [50]. In addition, diffusion-based generative frameworks [51], [52] have been employed to jointly generate multiple fashion items [53], enabling the synthesis of visually compatible and personalized outfits and extending outfit recommendation into the generative paradigm.

While neural network–based methods have brought meaningful progress in feature representation, implicit interaction modeling, and user adaptation, building a unified and efficient framework [54] that jointly models items, users, and outfits remains an ongoing area of research.

C. Graph-based approaches

Graph-based approaches have recently gained significant attention by modeling fashion items, users, and their relationships as nodes and edges within a graph structure, allowing for the powerful application of Graph Neural Networks (GNNs) [41], [55]–[61]. These methods are adept at capturing complex, non-Euclidean relationships that are often missed by pairwise or sequence-based models.

Early studies constructed item–item or outfit graphs, where items were represented as nodes and co-occurrence relations as edges, reformulating outfit compatibility prediction as a link prediction task [16], [62]. Later extensions incorporated user–outfit interactions for personalized recommendation [19], [63]. Recent research emphasizes richer and more adaptive graph structures [64]–[67]. For instance, outfit-level graphs connect all items within an outfit and apply dot-attention or multi-head attention to encode fine-grained visual and semantic relations, aligning outfit and user embeddings for personalized matching [59], [61]. Hierarchical designs such as FGAT [60] organize users, outfits, and items into multi-tier graphs, jointly modeling compatibility and preference through attention-based propagation.

To enhance semantic richness and mitigate sparsity, knowledge graphs integrate attribute, brand, and category relations into GNN pipelines for high-order reasoning and attribute-aware matching. Advanced variants like FCSA-GNN [41] capture both low- and high-order connectivity. Transformer-based and federated GNNs further improve global relation modeling and privacy-preserving scalability [55], [66].

Graph-based models also support dynamic user preference modeling by combining long-term and short-term graphs [65] and aligning multi-intent representations for sequential fashion recommendation [64]. Despite substantial progress, challenges remain in graph construction quality, scalability, and over-smoothing, motivating research toward unified and efficient graph frameworks capable of fast user adaptation.

III. Persoanlized Outfit Composition tasks

Let an outfit be represented as a finite set of fashion items $o = \{x_i\}_{i=1}^n$, where x_i denotes the i-th item in the outfit, and let μ denote a user representation. We consider the following personalized outfit composition tasks [11].

Outfit recommendation. Evaluate how well an outfit o is composed for a given user μ. This is a well-studied problem that, unlike traditional personalized recommendation, requires modeling the compatibility among items within the outfit while accounting for user-specific preferences.

Outfit refinement. Perform item-level reasoning to improve the coherence of an outfit. The well-studied outfit completion task [18], [20], [68] focuses on predicting the missing item that best completes a partial outfit. Formally, given an incomplete outfit o_{-i} missing its i-th item, the objective is to select the item that best aligns with the remaining ones and the user's style preferences. We additionally consider identifying, given a complete outfit o, the item whose removal yields the most coherent remaining outfit, referred to as the *outlier detection* task.

User profiling. Given a new user with a limited number of observed outfits, we aim to infer a representative user embedding μ that captures the user's style preferences. This enables fast adaptation of personalized outfit composition from limited user samples.

Most existing studies address these tasks independently with task-specific objectives and formulations. In this work, we present a unified modeling framework that expresses these tasks within a common mathematical form, where the same formulation principle is applied to different reasoning targets depending on the task objective. This unified treatment provides a coherent theoretical basis, allowing analytical insights and methodological advances obtained from one task to be naturally transferred and generalized to others, thereby offering a consistent foundation for personalized outfit composition.

IV. Unified Prediction Formulation

We develop a unified probabilistic formulation as the foundation for all personalized outfit composition tasks.

A. Probabilistic formulation

Let $\boldsymbol{z} \in \mathbb{S}^{d-1}$ denote an embedding on the unit hypersphere, representing a generic entity such as an outfit or an item. Let Ω be the corresponding sample space. The predictive distribution of $\boldsymbol{z}$ is defined as

$$p(\boldsymbol{z}; \boldsymbol{\omega}, \kappa) = \frac{\exp(\kappa\, \boldsymbol{\omega}^{\intercal} \boldsymbol{z})}{\sum_{\boldsymbol{x} \in \Omega} \exp(\kappa\, \boldsymbol{\omega}^{\intercal} \boldsymbol{x})}, \tag{1}$$

where $\boldsymbol{\omega} \in \mathbb{S}^{d-1}$ is the mean direction and $\kappa \geq 0$ controls the sharpness [69]. This serves as a directional analogue of softmax, measuring alignment on the hypersphere.

B. Personalized modeling

To capture user-specific variation, we introduce a user embedding $\boldsymbol{\mu} \in \mathbb{S}^{d-1}$ representing deviation from the global preference $\boldsymbol{\omega}$, and a learnable weight $\lambda \geq 0$ controlling the strength of the global component. The personalized predictive model is

$$p(\boldsymbol{z}; \boldsymbol{\omega}, \boldsymbol{\mu}, \kappa) = \frac{\exp(\kappa(\lambda\boldsymbol{\omega} + \boldsymbol{\mu})^{\intercal} \boldsymbol{z})}{\sum_{\boldsymbol{x} \in \Omega} \exp(\kappa(\lambda\boldsymbol{\omega} + \boldsymbol{\mu})^{\intercal} \boldsymbol{x})}. \tag{2}$$

Rewriting it gives

$$\kappa' = \kappa\|\lambda\boldsymbol{\omega}+\boldsymbol{\mu}\|, \omega' = \frac{\lambda\boldsymbol{\omega} + \boldsymbol{\mu}}{\|\lambda\boldsymbol{\omega} + \boldsymbol{\mu}\|}, |\lambda - 1| \leq \frac{\kappa'}{\kappa} \leq |\lambda + 1| \tag{3}$$

It shows that personalization simultaneously rotates the global direction toward their preference and adjusts concentration. Greater deviation from the global preference makes the distribution flatter, whereas stronger alignment sharpens it.

C. Likelihood estimation

Each task maximizes the likelihood of observed samples under Eq. (2). Because Ω is large, we approximate the normalization term by sampling K candidates $\{\boldsymbol{z}_i\}_{i=1}^{K}$ with one positive and $K-1$ negatives. The likelihood objective is then

$$\mathcal{J}_{\text{MLE}} = \mathbb{E}\left[\frac{\exp\big(\kappa(\lambda\boldsymbol{\omega} + \boldsymbol{\mu})^{\intercal} \boldsymbol{z}_i\big)}{\sum_{k=1}^{K} \exp\big(\kappa(\lambda\boldsymbol{\omega} + \boldsymbol{\mu})^{\intercal} \boldsymbol{z}_k\big)}\right], \tag{4}$$

which follows the InfoNCE formulation and thus provides a lower bound on the mutual information $I(\boldsymbol{\mu}; \boldsymbol{z})$ [70]–[73].

Intuitively, maximizing $\mathcal{J}_{\text{MLE}}$ encourages the model to assign high likelihood to observed samples $p(\boldsymbol{z}|\boldsymbol{\mu})$. At the optimum, the critic satisfies

$$\kappa(\boldsymbol{\lambda}\boldsymbol{\omega} + \boldsymbol{\mu})^{\intercal} \boldsymbol{z} = \log p(\boldsymbol{z}|\boldsymbol{\mu}) + c(\boldsymbol{\mu}), \tag{5}$$

where $c(\boldsymbol{\mu})$ is a normalization constant depending only on $\boldsymbol{\mu}$ and thus does not affect relative scores among different $\boldsymbol{z}$ for a fixed $\boldsymbol{\mu}$. Rewriting the conditional term gives

$$\kappa(\lambda\boldsymbol{\omega} + \boldsymbol{\mu})^{\intercal} \boldsymbol{z} \propto \log p(\boldsymbol{z}|\boldsymbol{\mu})/p(\boldsymbol{z}) + \log p(\boldsymbol{z}), \tag{6}$$

which naturally decomposes into two components: $\lambda\boldsymbol{\omega}^{\intercal}\boldsymbol{z}$ represents the global popularity, while $\boldsymbol{\mu}^{\intercal}\boldsymbol{z}$ captures user-specific deviation, disentangled from the global bias.

D. User adaptation

In practice, new users may appear after the prediction model has been trained, requiring personalized parameters to be estimated without retraining the full model. Given a small set of S independent samples $\{\boldsymbol{z}_i\}_{i=1}^{S}$, the task reduces to estimating the posterior:

$$p(\boldsymbol{\mu}|\boldsymbol{z}_1, \ldots, \boldsymbol{z}_S) \propto p(\boldsymbol{\mu}) \prod_{i=1}^{S} p(\boldsymbol{z}_i|\boldsymbol{\mu}). \tag{7}$$

where S is typically small. Assuming a uniform prior $p(\boldsymbol{\mu})$, the problem is equivalent to maximum likelihood estimation, whose optimal solution is the mean direction of the samples:

$$\boldsymbol{\mu}^* = \bar{\boldsymbol{z}} = \frac{\sum_{i=1}^{S} \boldsymbol{z}_i}{\|\sum_{i=1}^{S} \boldsymbol{z}_i\|}. \tag{8}$$

This closed-form estimator is computationally efficient but depends on how well the global preference $\boldsymbol{\omega}$ and the user-specific preference $\boldsymbol{\mu}$ are decoupled as dicsussed in Eq. 6.

When this separation is imperfect, we adopt an ad-hoc refinement [22]. Let ψ denote the set of existing user embeddings. A new user is initialized by prototype matching:

$$\boldsymbol{\mu}^* = \arg\max_{\boldsymbol{\mu} \in \psi} \bar{\boldsymbol{z}}^{\intercal} \boldsymbol{\mu}. \tag{9}$$

To better account for diverse user tastes, each user can be represented by m variables in ψ, and the new user's embedding is constructed by aggregating the m most similar ones:

$$\{\boldsymbol{\mu}_1^*, \ldots, \boldsymbol{\mu}_m^*\} = \arg\max_{\boldsymbol{\mu}_i \in \psi} \sum_{i=1}^{m} \bar{\boldsymbol{z}}^{\intercal} \boldsymbol{\mu}_i. \tag{10}$$

Both approaches aim to maximize the posterior for new users. The closed-form estimator in Eq. (8) provides an analytic solution, while the prototype-based refinement improves robustness when global and personalized factors are not well separated.

E. Interpretation

The unified model provides a common probabilistic framework for all tasks. Each task applies this framework by defining probabilistic relations over different variables and sample spaces, for example between outfits, items, and user representations. From this perspective, the objectives of all tasks can be viewed as estimating directional likelihoods on the unit sphere and maximizing mutual information between learned representations and user preferences. This formulation offers a coherent theoretical foundation for analyzing each task within a single unified principle.

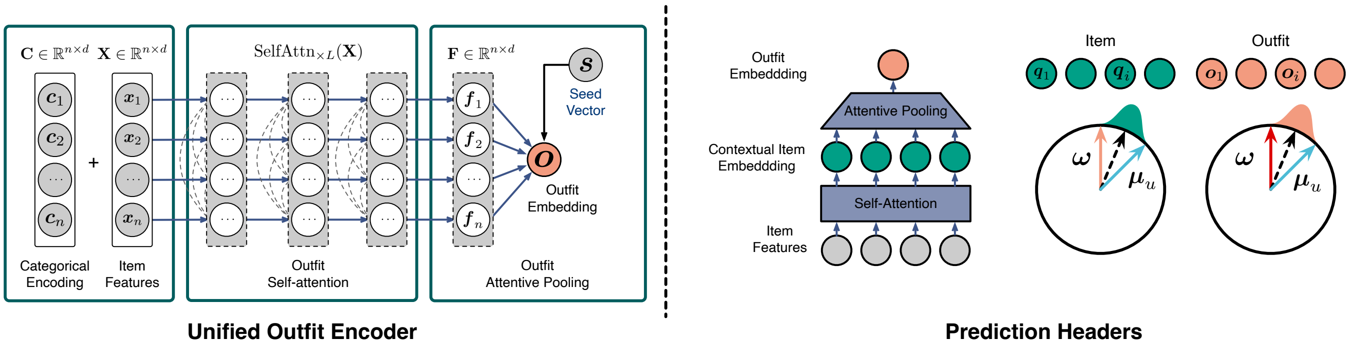


Fig. 2. Overall framework of the proposed model. Given an outfit, the inter-item relationships are captured through stacked self-attention layers, and the resulting features are aggregated into a compact outfit embedding via attentive pooling. These embeddings are then used by task-specific prediction headers for different outfit composition tasks.

V. Overall Framework

As illustrated in Fig. 2, our framework consists of a unified outfit encoder and several task-specific prediction headers. The encoder captures inter-item relationships within an outfit through stacked self-attention layers, producing contextual item embeddings $\mathbf{F} \in \mathbb{R}^{n \times d}$. These contextual embeddings are then aggregated via attentive pooling into a compact outfit embedding $\mathbf{o} \in \mathbb{R}^{d}$. The prediction headers leverage this shared representation to handle multiple outfit composition tasks in a unified manner, including personalized recommendation, outfit completion, and outlier detection. This section focuses on the encoder design and the learning of contextual and outfit embeddings using the self-attention mechanism [26], while the task-specific learning objectives are described in subsequent sections.

A. Categorical encoding

Each item belongs to a fashion category, such as top, bottom and shoes, which provides extra information for learning an informative outfit embedding. For example, an outfit with multiple pairs of shoes would be incongruous, while an outfit missing item from any major fashion category would be incomplete. The category information can determine the relative importance of items in an outfit, and thereby helps the outfit embedding learning. Previous works [1], [6] have also shown that learning compatibility in sub-spaces, which are conditioned on the categories, can greatly improve the performance.

We build a vocabulary $\mathbf{E} \in \mathbb{R}^{c \times d}$ for categorical embedding, where c is the number of item categories and d is the dimension. The elements of $\mathbf{E}$ are learnable parameters. The categorical embedding is added to the item embedding to enhance the item representation, i.e.

$$\mathbf{X}' = \mathbf{X} + \mathbf{C} \tag{11}$$

where $\mathbf{C} \in \mathbb{R}^{n \times d}$ with the i-th row fetched from $\mathbf{E}$ being the corresponding categorical embedding of the i-th item in $\mathbf{X}$. Similar to positional encoding [29], we use the additive embedding for category information and learn the underlying relationships between items by self-attention without explicitly modeling. For simplicity, we will omit the superscript "$'$" and use $\mathbf{X}$ to denote both item features with and without the categorical encoding.

B. Multi-head attention

We first introduce the attention mechanism, a basic component for augmenting item representation, and then introduce its extended form, the multi-head attention. Given a set of n query vectors $\mathbf{Q} \in \mathbb{R}^{n \times d}$, an attention function [29] updates them via m key-value pairs $\mathbf{K} \in \mathbb{R}^{m \times d}, \mathbf{V} \in \mathbb{R}^{m \times d}$. For brevity, we set the dimension of all vectors to be the same as the query vectors. Each output vector is a weighted sum of the value vectors in $\mathbf{V}$ with the weights being computed by the inner product between the query and the key vectors in $\mathbf{K}$.

$$\mathrm{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \mathrm{Softmax}(\frac{\mathbf{Q}\mathbf{K}^{\intercal}}{\sqrt{d}})\mathbf{V} \tag{12}$$

The multi-head attention extends the above function by projecting the triplet $(\mathbf{Q}, \mathbf{K}, \mathbf{V})$ into h subspaces and concatenating the outputs in each subspace. h is known as the number of heads. For ease of description, this process is succinctly expressed as $\mathrm{MultiHead}(\mathbf{Q}, \mathbf{K}, \mathbf{V})$.

C. Outfit self-attention

Given the features $\mathbf{X} \in \mathbb{R}^{n \times d}$ of the n items in an outfit, we employ the induced self-attention block (ISAB) [22], [26] to capture inter-item relationships while accounting for the unequal importance of different items.

$$\begin{aligned} \mathrm{ISAB}(\mathbf{X}) &= \mathrm{MultiHead}(\mathbf{X}, \mathbf{P}, \mathbf{P}) \in \mathbb{R}^{n \times d}, \\ \text{where } \mathbf{P} &= \mathrm{MultiHead}(\mathbf{I}, \mathbf{X}, \mathbf{X}) \in \mathbb{R}^{p \times d}. \end{aligned} \tag{13}$$

Here, $\mathbf{I} \in \mathbb{R}^{p \times d}$ denotes p learnable inducing points that aggregate global information from the n items and decode it back to item-level representations. As a simpler alternative, a standard self-attention layer can also be used:

$$\mathrm{SAB}(\mathbf{X}) = \mathrm{MultiHead}(\mathbf{X}, \mathbf{X}, \mathbf{X}) \in \mathbb{R}^{n \times d}, \tag{14}$$

which directly computes full item-to-item interactions without introducing inducing points.

The output is refined using residual connections, layer normalization, and a feedforward network:

$$\begin{aligned} \mathrm{SelfAttn}(\mathbf{X}) &= \mathrm{LayerNorm}(\mathbf{H} + \sigma(\mathbf{H})), \\ \text{where } \mathbf{H} &= \mathrm{LayerNorm}\left(\mathbf{X} + \mathrm{ISAB}(\mathbf{X})\right). \end{aligned} \tag{15}$$

$\sigma(\cdot)$ is a row-wise feedforward layer, and $\mathrm{LayerNorm}(\cdot)$ denotes layer normalization [74].

Stacking multiple $\mathrm{SelfAttn}(\cdot)$ layers models higher-order relationships among outfit items:

$$\mathbf{F} = \mathrm{SelfAttn}_{\times L}(\mathbf{X}) \in \mathbb{R}^{n\times d}, \tag{16}$$

where L is the number of layers.

D. Outfit attentive Pooling

To get a fixed length outfit representation regardless of the number of items in it, we use a vector $\boldsymbol{s} \in \mathbb{R}^d$ to aggregate the previous output $\mathbf{F}$ as follows:

$$\begin{gathered}\boldsymbol{o} = \mathrm{LayerNorm}(\boldsymbol{h} + \sigma(\boldsymbol{h}))\\ \text{where } \boldsymbol{h} = \mathrm{LayerNorm}\left(\boldsymbol{s} + \mathrm{MultiHead}(\boldsymbol{s}, \mathbf{F}, \mathbf{F})\right)\end{gathered} \tag{17}$$

where $\boldsymbol{s}$ is learned as a model parameter. The outfit attentive pooling produces a single compact vector $\boldsymbol{o} \in \mathbb{R}^d$ for an outfit. It can be easily seen that the transformation from n items $\mathbf{X}$ to a single outfit embedding $\boldsymbol{o}$ is invariant to the permutation of items. With the compact outfit embedding, different downstream tasks can be handled easily and accordantly.

VI. Optimization

Based on the unified probabilistic formulation in Section IV, we now instantiate the prediction model and objective for specific tasks.

A. Outfit recommendation

The task aims to assess whether a set of items $\{x_i\}_{i=1}^n$ form a compatible outfit that matches the user's style. Following Eq. (2), we define the personalized scoring function as:

$$f(\boldsymbol{o}; \boldsymbol{\omega}, \boldsymbol{\mu}_u, \lambda) = (\lambda\boldsymbol{\omega} + \boldsymbol{\mu}_u)^\intercal \boldsymbol{o}, \tag{18}$$

where $\boldsymbol{\omega}$ encodes global compatibility and $\boldsymbol{\mu}_u$ captures user-specific preference. This task-specific instantiation of the unified framework is denoted as *model-r*.

Objective function: For each user, created outfits are treated as positive samples, and others as negatives sampled from Ω. Given K candidate outfits $\boldsymbol{o}_{1:K}$ with one positive, the objective directly follows the likelihood in Eq. (4):

$$\mathcal{L}_r = \mathbb{E}\left[-\log\frac{\exp\left(\kappa(\lambda\boldsymbol{\omega} + \boldsymbol{\mu}_u)^\intercal \boldsymbol{o}_i\right)}{\sum_{k=1}^K \exp\left(\kappa(\lambda\boldsymbol{\omega} + \boldsymbol{\mu}_u)^\intercal \boldsymbol{o}_k\right)}\right]. \tag{19}$$

This corresponds to maximizing the mutual information between users and their preferred outfits.

New user profiling: For a new user v with S outfits $\{\boldsymbol{o}_i\}_{i=1}^S$, the personalized embedding is estimated in closed form from Eq. (8):

$$\boldsymbol{\mu}_v^* = \frac{\sum_{i=1}^S \boldsymbol{o}_i}{\|\sum_{i=1}^S \boldsymbol{o}_i\|}. \tag{20}$$

B. Outfit completion

The goal of outfit completion is to select the most compatible item for a partial outfit. We define the scoring function as

$$f(x_i; o_{-i}, \boldsymbol{\omega}, \boldsymbol{\mu}_u, \lambda) = (\lambda\boldsymbol{\omega} + \boldsymbol{\mu}_u)^\intercal \boldsymbol{q}_i, \tag{21}$$

where $\boldsymbol{q}_i$ is the item representation before outfit-attentive pooling. This task-specific instantiation of the unified framework is denoted as *model-c*.

Objective function: For each user, the ground-truth completing item is treated as a positive sample, and other sampled items as negatives. Given K candidate items $\boldsymbol{q}kk = 1^K$ with one positive, the objective directly follows the likelihood in Eq. (4):

$$\mathcal{L}_c = \mathbb{E}\left[-\log\frac{\exp(\kappa(\boldsymbol{o}_{-i} + \lambda\boldsymbol{\omega} + \boldsymbol{\mu}_u)^\intercal \boldsymbol{q}_i)}{\sum_{k=1}^K \exp(\kappa(\boldsymbol{o}_{-i} + \lambda\boldsymbol{\omega} + \boldsymbol{\mu}_u)^\intercal \boldsymbol{q}_k)}\right]. \tag{22}$$

This corresponds to maximizing the conditional mutual information between users and items given the partial outfit.

New user profiling: Let $\{o_i\}_{i=1}^S$ be the set of outfits for a new user v, and $\{\boldsymbol{q}_{ij}\}_{j=1}^{n_i}$ the corresponding items in the i-th outfit with n_i items. The closed-form estimator is computed as:

$$\boldsymbol{\mu}_v^* = \frac{\sum_{i=1}^S \sum_{j=1}^{n_i} \boldsymbol{q}_{ij}}{\left\|\sum_{i=1}^S \sum_{j=1}^{n_i} \boldsymbol{q}_{ij}\right\|}. \tag{23}$$

Decoupling $\boldsymbol{o}$ and $\boldsymbol{\mu}$ in outfit completion is challenging due to conditional dependence, the ad-hoc solution is expected to outperform the closed-form solution.

C. Outlier item detection

The goal of outlier item detection is to identify elements within an outfit that are incompatible with the overall style. We define the scoring function as

$$f(x_i; o_{-i}, \boldsymbol{\omega}, \boldsymbol{\mu}_u, \lambda) = (\lambda\boldsymbol{\omega} + \boldsymbol{\mu}_u)^\intercal \boldsymbol{g}_i, \tag{24}$$

where $\boldsymbol{g}_i$ is the item representation before outfit-attentive pooling. This task-specific instantiation of the unified framework is denoted as *model-d*.

Objective function. The item set Ω for this task contains all items within an outfit, the objective function is then defined as:

$$\mathcal{L}_d = \mathbb{E}\left[-\log\frac{\exp(-\kappa(\lambda\boldsymbol{\omega} + \boldsymbol{\mu}_u)^\intercal \boldsymbol{g}_i)}{\sum_{k=1}^n \exp(-\kappa(\lambda\boldsymbol{\omega} + \boldsymbol{\mu}_u)^\intercal \boldsymbol{g}_k)}\right]. \tag{25}$$

User adaptation. For new users providing only positive outfits, we estimate user preference by the mean direction of all attended item embeddings:

$$\boldsymbol{\mu}_v^* = \frac{\sum_{i=1}^S \sum_{j=1}^{n_i} \boldsymbol{g}_{ij}}{\|\sum_{i=1}^S \sum_{j=1}^{n_i} \boldsymbol{g}_{ij}\|}. \tag{26}$$

TABLE I
COMPARISON OF DIFFERENT METHODS ON COMPATIBILITY PREDICTION TASK.

Method	Category	Polyvore Maryland			Polyvore Outfit-D			Polyvore Outfits		
	Encoding	AUC	NDCG	FITB	AUC	NDCG	FITB	AUC	NDCG	FITB
SiameseNet [13]	–	0.9165	0.8883	0.6203	0.8444	0.7817	**0.5774**	0.8744	0.8316	0.5823
Bi-LSTM [2]	–	0.9424	0.9200	**0.7016**	0.7577	0.6970	0.5442	0.8055	0.7515	0.5859
CSN [6]	✓	0.9314	0.9013	0.6580	0.8502	0.7889	0.5580	0.8798	0.8333	0.5809
SCE-Net [7]	–	0.9160	0.8890	0.6170	0.8506	0.7906	0.5534	0.8998	0.8592	0.6030
NGNN [16]	✓	0.9485	0.9239	0.5811	0.8416	0.7958	0.4834	0.9037	0.8766	0.5500
Model-r	✓	**0.9735**	**0.9600**	0.6989	**0.8731**	**0.8377**	0.5436	**0.9259**	**0.9049**	**0.6039**

D. Alternative modeling

The formulations of outfit completion and outlier detection are not unique. For example, in the outfit completion task, the similarity between a candidate item and the embedding of the incomplete outfit can also be used for prediction. Similarly, for outlier detection, the similarity between the outfit embedding and each item representation can be computed to identify inconsistent elements within the outfit. In this work, we keep the formulation simple and unified across tasks, while exploring richer outfit-item interactions is left for future research.

VII. Experiments

In this section, we compare our approach with state-of-the-art methods on various fashion datasets.

Fashion datasets. We consider four datasets: Polyvore Maryland [2], Polyvore UIUC [6], Polyvore-Us [8], and IQON-3000 [21]. Polyvore Maryland and Polyvore UIUC, widely used for compatibility prediction, lack user data. Polyvore UIUC has two versions: Polyvore Outfits-D, where items don't overlap between training and testing sets, making it more challenging, and Polyvore Outfits. Polyvore-Us datasets, defined by user count U, have four versions: Polyvore-519, Polyvore-630, Polyvore-32, and Polyvore-53. We use Polyvore-630/519 for personalized outfit recommendation and Polyvore-53/32 for new user profiling. Each user has 200 training and 40 testing outfits, each containing 3 items from different categories. For IQON-3000, we filter 608 users, retaining those with 85 training and 20 testing outfits, each containing 3–8 items across 8 categories. Users are split into two groups, forming IQON-550 and IQON-58.

Evaluation metrics. To evaluate outfit recommendation accuracy, we consider the Area Under the ROC curve (AUC) and the Normalized Discounted Cumulative Gain (NDCG) used in previous works [1], [8]. These metrics assess the ranking quality of positive and negative outfits. The testing set maintains a 1:10 ratio, with performance averaged across users for personalized datasets. For outfit completion, we measure FITB [2] accuracy using varying candidate sizes. Outlier item detection is evaluated via averaged detection accuracy. As no existing dataset is available, we generate one by randomly replacing an item in each outfit.

Baseline methods. We compare our models with several state-of-the-art methods: SiameseNet [13], Bi-LSTM [2], CSN [6], SCE-Net [7], NGNN [16], FNH [9], and Outfit-Net [18]. SiameseNet maximizes similarity between positive item pairs and minimizes it for negative pairs using metric learning. Bi-LSTM [2] treats compatibility prediction as an item prediction problem, using a bidirectional LSTM to maximize the likelihood of positive outfits. CSN [6] embeds item pairs into distinct subspaces to learn conditional similarities [14] by incorporating category information in item embeddings. SCE-Net learns multiple conditional item embeddings, weighting each via an attention mechanism. NGNN constructs a fashion graph from category co-occurrence and trains outfit compatibility using a graph convolutional network. FNH models outfit compatibility and user preferences in a pairwise manner. Since FNH is hash-based, we train it without binarization for fair comparison. Outfit-Net employs multiple-instance learning and attention to capture users' fashion preferences for personalized recommendations.

Implementation details. The item features are extracted from images with ResNet-34 [75] and used as the input for all methods for fair comparison. We set the latent dimension to 128 and SGD with momentum [76] is used for all methods. For our methods, we use $\kappa = 10$ buy default, and set the number of self-attention layer to 2. The learning rates are reduced when the accuracy stops increasing on validation set and the initial learning rate is 0.01 for all tasks. All methods are implemented with PyTorch. For the outfit recommendation task and outfit completion task, we set the number of negatives to 32 and use ISAB to learn outfit embedding.

A. Outfit recommendation

Compatibility prediction is the essential problem for outfit recommendation. We first use Polyvore Maryland [2] and Polyvore UIUC [6] to evaluate the performance on compatibility prediction tasks. The comparison results are shown in Table I. Our models achieve the superior performance on ranking metrics, i.e. AUC and NDCG, across all datasets even without the categorical information. We believe that due to the self-attention mechanism [26], [29], the resulting outfit embedding can better capture the underlying high-order relationships among multiple items. Besides, adding the categorical information can sustainably improve the recommendation accuracy as expected.

Another finding is that models that perform well in ranking accuracy may not necessarily perform well in the outfit completion task, and vice versa. For example, while NGNN achieves the highest AUC in the baseline methods across all datasets, it obtains the lowest FITB accuracy. On the other hand, the Bi-LSTM model performs worst poorly on the AUC

TABLE II
COMPARISON OF DIFFERENT METHODS ON PERSONALIZED OUTFIT RECOMMENDATION TASK.

Method	Category Encoding	Polyvore-519			Polyvore-630			IQON-550		
		AUC	NDCG	FITB	AUC	NDCG	FITB	AUC	NDCG	FITB
SiameseNet [13]	–	0.7956	0.6139	0.5250	0.7737	0.6019	0.5065	0.8065	0.6532	0.4778
Bi-LSTM [2]	–	0.8142	0.6467	0.5427	0.7846	0.6336	0.5156	0.8127	0.6650	0.5139
CSN [6]	✓	0.7825	0.6127	0.5043	0.7718	0.6038	0.4983	0.7980	0.6381	0.4797
SCE-Net [7]	–	0.8078	0.6527	0.5336	0.8020	0.6580	0.5297	0.8228	0.6823	0.4896
Outfit-Net [18]	–	0.8175	0.6671	0.4621	0.8411	0.7152	0.5002	0.8308	0.6956	0.4291
NGNN [16]	✓	0.8230	0.6640	0.5280	0.7784	0.5918	0.4935	0.8601	0.7613	0.4920
FHN [8]	✓	0.9015	0.8298	0.6038	0.8960	0.8229	0.6060	0.8973	0.8152	0.5402
Model-r	–	0.9215	0.8607	0.6381	0.9006	0.8233	0.6099	0.9325	0.8853	0.5740
Model-r	✓	**0.9294**	**0.8704**	**0.6581**	**0.9075**	**0.8342**	**0.6258**	**0.9427**	**0.8967**	**0.6024**

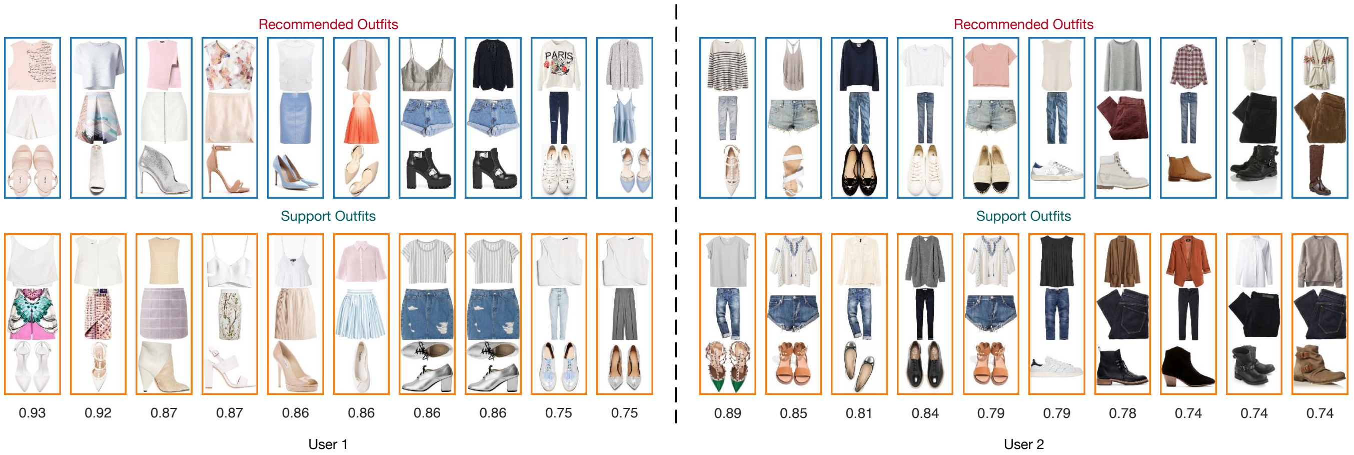


Fig. 3. The outfit recommendation results for different users.

in two Polyvore UIUC datasets, but still achieves a comparable FITB accuracy due to its item prediction pipeline. The reason for such discrepancies is that the outfit completion task focuses more on individual items, while the outfit recommendation task focuses more on global style. As a result, the compatibility score needs to be sensitive to changes in individual items, which may not always be fulfilled. To address this limitation, an appropriate strategy is needed for the outfit completion task. We provide the compatibility-based FITB accuracy as a reference for outfit recommendation tasks and leave a detailed discussion for later sections.

For personalized outfit recommendation task, we compare our model with state-of-the-art approaches in Table II, where Polyvore-Us [8] and IQON [22] are used. As we can see, our models achieve the best performance on all metrics including the FITB accuracy. On Polyvore-630, the improvement in AUC is relatively small because all outfits contain exactly three items with clear category boundaries, where the rigid pairwise tensor decomposition in models such as FHN can already capture most compatibility relations. Nevertheless, our model achieves higher NDCG and FITB scores, indicating more accurate ranking calibration and finer item-level reasoning. For Polyvore-519 and IQON-550, where outfit sizes vary, the advantages of our model are more evident across all metrics.

Interpretability of recommendations. With the compact outfit embedding, for each recommended outfit, we can retrieve similar outfits from the user's past selections to support the recommendation results. Since other baseline methods do not establish the similarity measurements between outfits, we only show examples of ours in Fig. 3. Outfits at the top are the recommended ones, and outfits at the bottom are support outfits from the training set. We also show the cosine similarities between the recommended ones and support ones.

TABLE III
COMPARISON OF DIFFERENT METHODS ON THE COMPLETION TASK.

Method	Polyvore Maryland	Polyvore Outfits-D	Polyvore Outfits	Polyvore 519	Polyvore 630	IQON 550
Bi-LSTM	0.7016	0.5442	0.5859	0.5427	0.5156	0.5139
CSN	0.6580	0.5580	0.5809	0.5043	0.4983	0.4797
SCE-Net	0.6170	0.5534	0.6030	0.5336	0.5297	0.4896
NGNN	0.5811	0.4788	0.5500	0.5276	0.4897	0.4920
Outfit-Net	–	–	–	0.4621	0.5002	0.4291
FHN	–	–	–	0.6038	0.6060	0.5402
Model-r	0.6981	0.5436	0.6039	0.6581	0.6258	0.6024
Model-c	**0.7291**	**0.5781**	**0.6483**	**0.6660**	**0.6323**	**0.6153**

TABLE IV
COMPARISON OF DIFFERENT METHODS ON OUTLIER ITEM DETECTION.

Method	Polyvore Maryland	Polyvore Outfits-D	Polyvore Outfits	Polyvore 519	Polyvore 630	IQON 550
Bi-LSTM	0.5663	0.4033	0.4225	0.5066	0.4998	0.3530
CSN	0.5433	0.4971	0.5099	0.5047	0.5160	0.3883
SCE-Net	0.5233	0.5030	0.5325	0.5261	0.5128	0.4175
Model-d	**0.6653**	**0.5342**	**0.5885**	**0.6849**	**0.6777**	**0.5580**

TABLE V
PER-CATEGORY PERFORMANCE ON THE IQON-550 DATASET FOR OUTFIT COMPLETION (MODEL-c) AND OUTLIER DETECTION (MODEL-d). HIGHER VALUES INDICATE BETTER COMPATIBILITY UNDERSTANDING.

Type	Accessories	Bag	Bottom	Coat	Dress	Hat	Shoes	Top
Model-c	0.6048	0.6110	0.5957	0.5654	0.6368	0.6526	0.6756	0.6058
Model-d	0.5403	0.5485	0.5372	0.5042	0.6074	0.5919	0.6090	0.5519

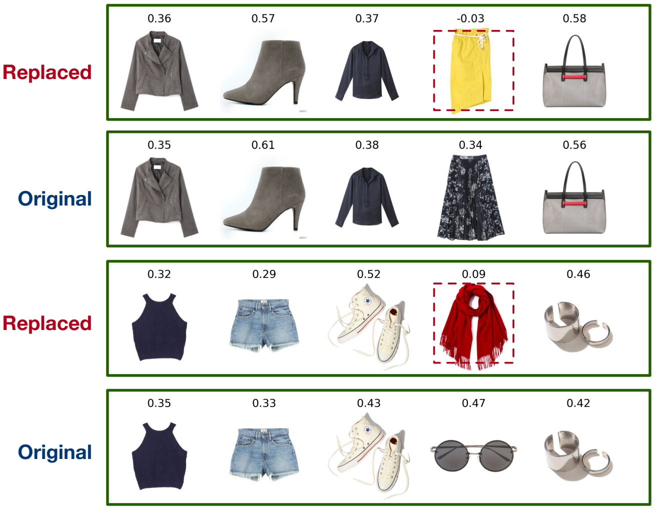


Fig. 4. The outlier item detection results. Scores indicate the compatibility of each item with the outfit, with a red box highlighting the ground-truth item that was replaced.

As we can see, the similar visual style of the support outfits do help to interpret the recommendation results.

B. Outfit refinement

Outfit completion and outlier item detection are the two basic outfit refinement tasks.

Outfit completion. Outfit completion involves selecting the most suitable item to fit an incomplete outfit from a set of candidate items, while outlier item detection involves detecting the most incompatible item for a given outfit. As discussed, using the overall compatibility for item prediction is suboptimal, so we propose a new formula in Eq. (21) to focus more on individual items. The comparison results are shown in Table III. For Outfit-Net and FHN, we only use their performance on personalized datasets. As we can see, our proposed model-c not only improve the performance over the compatibility-based model, but also significantly outperforms other baselines.

Outlier item detection. To evaluate outlier item detection, we created a new dataset by randomly replacing one item from each positive outfit, and treating the replaced item as the outlier. We only consider the case where there is only one outlier and the results are shown in Table IV. Since the task has not been studied before, we only compare with the pairwise models and Bi-LSTM. For pairwise models, we use the averaged similarity of an item with others as the compatibility score. For the Bi-LSTM, we use the log probability of each item as the compatibility score. And the item with the lowest compatibility is selected as the outlier. As we can see, our model significantly outperforms the baselines on all datasets. Fig. 4 shows examples of outlier detection results where the outlier items are successfully detected.

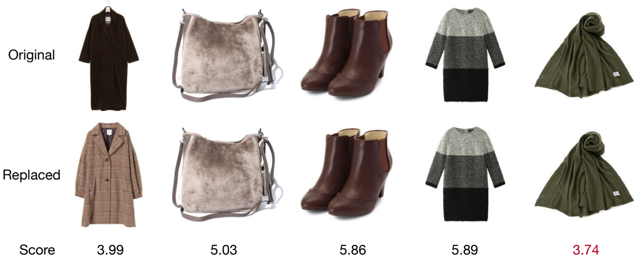


Fig. 5. A failure case of outlier detection on the IQON-550 dataset. The top row shows the original compatible outfit, while the bottom row displays the outfit after replacing the coat. The model incorrectly identifies the scarf as the outlier instead of the replaced coat.

Per-category analysis. To further understand task-specific behavior, we analyze the per-category performance of our models, as shown in Table V. Categories such as *dress*, *hat*, and *shoes* achieve the highest accuracy because they are visually salient and stylistically distinctive. Replacing these items easily disrupts outfit coherence, making inconsistencies easier to detect or complete. In contrast, the *coat* category shows the lowest performance, as it often dominates the overall outfit style; replacing it drastically changes the visual context, while predicting it is challenging due to its large intra-class variation. Meanwhile, *accessories* and *bags* yield moderate accuracy since they exert weaker influence on the global style and appear less frequently in the dataset.

These quantitative findings motivate a closer look at typical failure cases. As illustrated in Fig. 5, the model incorrectly identifies the scarf as the outlier instead of the replaced coat. This occurs because the coat largely determines the outfit's overall style, making it difficult for the model to discern whether the incompatibility arises from the replaced coat itself or from its strong stylistic influence on the remaining items.

C. New user profiling

In real-world applications, learning the preferences of new users with limited feedback is an important problem. Fine-tuning the whole model for new users is computationally expensive. Besides, since newly joined users usually have limited data, the model is prone to over-fitting. This raises the question of how well we can learn from such limited data. In this section, we test the performance when each new user has only 1, 5, or 10 outfits for learning.

We first show the performance of different strategies in Table VI, where we use 16 vectors to expand the search space for the ad-hoc strategy. We further use two models for comparison, one trained without $\boldsymbol{\mu}$ and the other a personalized model that drop $\boldsymbol{\mu}$ during evaluation. The model trained without $\boldsymbol{\mu}$ usually achieves better performance, and the personalized model is used to demonstrate how much improvement we can get from limited data. As we can see, for outfit recommendation task, both ad-hoc and close-from solutions achieve notable improvement. This shows that our proposed methods can efficiently

TABLE VI
NEW USER PROFILING ON DIFFERENT TASKS WITH DIFFERENT NUMBER OF AVAILABLE OUTFITS. FOR OUTFIT RECOMMENDATION TASK WE USE AUC AS THE METRIC, FOR OUTFIT COMPLEATION TASK WE USE 4 CANDIDATE ITEMS TO EVALUATE THE ACCURACY, AND FOR OUTLIER DETECTION TASKS, WE USE THE DETECTION ACCURACY.

Model	Strategy	Polyvore-32			Polyvore-53			IQON-58		
		1	5	10	1	5	10	1	5	10
Model-r	w/o $\boldsymbol{\mu}$	0.8459 / 0.8346			0.7710 / 0.7692			0.8756 / 0.8783		
Model-r	Ad-hoc	0.8491	0.8672	0.8807	0.7917	0.8240	0.8387	**0.8920**	**0.8967**	**0.9047**
Model-r	Closed-form	**0.8505**	**0.8793**	**0.8921**	**0.7924**	**0.8319**	**0.8480**	0.8890	0.8962	0.9044
Model-c	w/o $\boldsymbol{\mu}$	0.5797 / 0.5634			0.5218 / 0.5021			0.5495 / 0.5362		
Model-r	Ad-hoc	0.8491	0.8672	0.8807	0.7917	0.8240	0.8387	**0.8920**	**0.8967**	**0.9047**
Model-r	Closed-form	**0.8505**	**0.8793**	**0.8921**	**0.7924**	**0.8319**	**0.8480**	0.8890	0.8962	0.9044
Model-d	w/o $\boldsymbol{\mu}$	0.5855 / 0.5550			0.5467 / 0.4902			0.4913 / 0.4758		
Model-d	Ad-hoc	0.5699	0.6037	0.6210	**0.5341**	**0.5486**	**0.5684**	0.4882	**0.4982**	**0.4983**
Model-d	Closed-form	**0.5925**	**0.6164**	**0.6225**	0.5029	0.5367	0.5445	**0.4937**	0.4969	0.4958

TABLE VII
COMPARISON OF BASELINE METHODS ON NEW USER TASKS WITH DIFFERENT NUMBER OF AVAILABLE OUTFITS.

Method	Category Encoding	Polyvore-32			Polyvore-53			IQON-58		
		1	5	10	1	5	10	1	5	10
SiameseNet	–	0.7690	0.7870	0.7925	0.7363	0.7439	0.7478	0.7610	0.7655	0.7701
Bi-LSTM [2]	–	0.8152	0.8140	0.8106	0.7657	0.7634	0.7636	0.7929	0.7926	0.7943
CSN [6]	✓	0.7676	0.7829	0.7849	0.7386	0.7471	0.7518	0.7574	0.7591	0.7635
SCE-Net [7]	–	0.7974	0.8064	0.8093	0.7752	0.7768	0.7680	0.7829	0.7764	0.7801
Outfit-Net	–	0.6014	0.6794	0.7124	0.6005	0.6925	0.7167	0.6072	0.6578	0.7171
NGNN	✓	0.8154	0.8231	0.8276	0.7554	0.7635	0.7653	0.8346	0.8425	0.8435
FNH	✓	0.7537	0.8267	0.8528	0.7342	0.8066	0.8365	0.7982	0.8440	0.8610
Model-r	–	0.8453	0.8693	0.8843	0.7876	0.8332	0.8497	0.8679	0.8790	0.8884
Model-r	✓	**0.8505**	**0.8793**	**0.8921**	**0.7920**	**0.8345**	**0.8498**	**0.8890**	**0.8962**	**0.9044**

learn user preferences with limited feedback. Besides, the closed-form solution achieves the best accuracy on Polyvore-32/53 and similar performance on IQON-58. The reason is that, in outfit recommendation task, the general preference and user preference are well decoupled, i.e. the general preference can sufficiently captures the user indenepent term in Eq. (6). However, for outfit completion, this task is more challenging as it involves modeling the conditional mutual information between users and items. As a result, our method does not show a clear advantage, but performs comparably to the ad-hoc strategy. The same holds for outlier detection task.

We further evaluate the outfit recommendation task under the new user profiling setting. As shown in Table VII, our model achieves the best overall performance. For non-personalized methods, the entire pre-trained model is fine-tuned, while personalized ones re-train only the user-related parameters. Our method uses the closed-form solution. As illustrated in Fig. 6, when only a few outfits are available (e.g., fewer than 10), the closed-form solution performs notably better. As the number increases, fine-tuning gradually surpasses it due to the use of negative outfits for learning a more discriminative decision function. Nevertheless, the closed-form solution remains highly competitive.

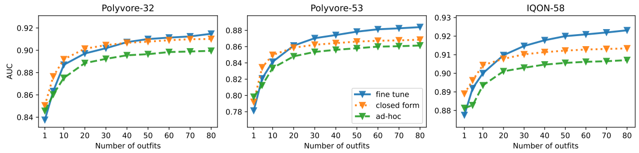


Fig. 6. Detailed performance of new user profiling tasks.

TABLE VIII
EFFECT OF CATEGORY ENCODING ON DIFFERENT MODELS

Method	Category Encoding	Polyvore-519	Polyvore-630	IQON-550
Model-r	–	0.9215	0.9006	0.9325
Model-r	✓	**0.9294**	**0.9075**	**0.9427**
Model-c	–	0.6626	0.6385	0.5964
Model-c	✓	**0.6720**	**0.6354**	**0.6320**
Model-d	–	0.6780	0.6736	0.5247
Model-d	✓	**0.6934**	**0.6839**	**0.5770**

TABLE IX
PERFORMANCE OF MODEL-r ON THE OUTFIT RECOMMENDATION TASK.

Model	ω	Polyvore-519		Polyvore-630		IQON-550	
		AUC	NDCG	AUC	NDCG	AUC	NDCG
Model-r	–	0.9279	0.8708	0.9082	0.8358	0.9465	0.9043
Model-r	✓	0.9294	0.8704	0.9075	0.8342	0.9427	0.8967

D. Ablation study

In the prediction model, we use general preference and user preference to learn different information about the data. For outfit recommendation, the general preference is a learnable parameter, and for outfit refinement, the general preference is the outfit embedding. In this section, we show the contribution of different preference terms.

On categorical encoding: We conduct an ablation study to examine the influence of category encoding across different

TABLE X
EVALUATION ON HARD NEGATIVES USING ONLY PRETRAINED USER EMBEDDINGS $\boldsymbol{\mu}$ (I.E., ELIMINATING GENERAL PREFERENCE $\boldsymbol{\omega}$ DURING EVALUATION).

Model	$\boldsymbol{\omega}$	Polyvore-519		Polyvore-630		IQON-550	
		AUC	NDCG	AUC	NDCG	AUC	NDCG
Model-r	–	0.7622	0.6050	0.7817	0.6240	0.7011	0.5375
Model-r	✓	**0.8480**	**0.7086**	**0.8406**	**0.7009**	**0.8220**	**0.6568**

TABLE XI
CONTRIBUTION OF GENERAL PREFERENCE $\boldsymbol{\omega}$ ON NEW USER PROFILING TASKS ON MODEL-r.

Strategy	$\boldsymbol{\omega}$	Polyvore-32			Polyvore-53			IQON-58		
		1	5	10	1	5	10	1	5	10
Ad-hoc	–	0.844	0.858	0.872	0.790	0.805	0.820	0.880	0.893	0.897
	✓	0.849	0.867	0.881	0.792	0.824	0.839	**0.892**	**0.897**	**0.905**
Closed-form	–	0.676	0.805	0.840	0.631	0.718	0.749	0.659	0.766	0.810
	✓	**0.851**	**0.879**	**0.892**	**0.792**	**0.832**	**0.848**	0.889	0.896	0.904

TABLE XII
THE CONTRIBUTION OF USER PREFERENCE $\boldsymbol{\mu}$ FOR OUTFIT RECOMMENDATION AND OUTFIT REFINEMENT TASKS. FOR MODEL-r, WE USE AUC AS THE METRIC, AND FOR MODEL-c, WE USE 4 CANDIDATE ITEMS TO EVALUATE THE ACCURACY.

Method	$\boldsymbol{\mu}$	Polyvore-519	Polyvore-630	IQON-550
Model-r	–	0.8521	0.8108	0.9094
Model-r	✓	**0.9294**	**0.9075**	**0.9427**
Model-c	–	0.5991	0.5539	0.5896
Model-c	✓	**0.6720**	**0.6354**	**0.6265**
Model-d	–	0.5971	0.5742	0.5251
Model-d	✓	**0.6934**	**0.6839**	**0.5770**

tasks, as shown in Table VIII. The results show that enabling category encoding consistently improves performance on all tasks, with the most significant gains observed in the outfit completion and outlier detection tasks. This suggests that category encoding plays a more critical role when the model needs to reason about item categories, such as identifying missing or inconsistent items, rather than when performing overall outfit scoring.

On general preference: We study the role of the general preference vector $\boldsymbol{\omega}$ in the outfit recommendation task. While $\boldsymbol{\omega}$ captures user-independent signals, our re-parametrization shows that the model can be trained equally well without it. As shown in Table IX, using only the user-specific embedding $\boldsymbol{\mu}$ yields nearly identical performance, suggesting that $\boldsymbol{\omega}$ is not essential for accurate recommendations on these datasets.

However, we find that training with the general preference vector $\boldsymbol{\omega}$ still results in a more structured latent space. To demonstrate this, we evaluate the pretrained models on hard negatives, which refer to outfits that are positively rated by other users, and present the results in Table X. During this evaluation, we exclude $\boldsymbol{\omega}$ from the scoring function in both variants and rely solely on the user-specific embedding $\boldsymbol{\mu}$. Despite the absence of $\boldsymbol{\omega}$ at inference, the model trained with it consistently outperforms the one trained without it. This indicates that incorporating general preference during training helps decouple user-specific and global signals, leading to better generalization even when only user-specific information is used at test time.

TABLE XIII
MULTI-TASK LEARNING. AUC FOR OUTFIT RECOMMENDATION (T_r), ACCURACY WITH FOUR CANDIDATE ITEMS FOR OUTFIT COMPLETION (T_c), AND DETECTION ACCURACY FOR OUTLIER DETECTION (T_d).

Dataset	Model	T_r	T_c	T_d
Polyvore-519	Single-task	0.9289	0.6660	0.6849
	Multi-task	**0.9308**	**0.6701**	**0.6894**
Polyvore-630	Single-task	0.9079	0.6323	0.6777
	Multi-task	**0.9108**	**0.6396**	**0.6819**
IQON-550	Single-task	0.9401	0.6153	0.5584
	Multi-task	**0.9433**	**0.6225**	**0.5621**

This decoupling is critical in the new user profiling scenario, as shown in Table XI. The closed-form solution, which relies on clean separation between $\boldsymbol{\omega}$ and $\boldsymbol{\mu}$, performs poorly without $\boldsymbol{\omega}$. In contrast, the ad-hoc strategy remains more robust because the learned user embeddings implicitly absorb both general and specific preferences. Still, both strategies benefit from explicitly modeling $\boldsymbol{\omega}$ during training.

On user preference: In personalized outfit recommendation, user information is important to the performance as shown in previous works. In this section, we further show that user preference also contributes to the outfit refinement tasks, i.e outfit completion and outlier detection. We report the performance of whether using user preference on Table XII. As we can see, introducing the user preference term can sustainably improve the performance on different tasks. Similar to outfit recommendation, where different users have different bias on outfit, users also have different preferences for different items in the outfit refinement tasks.

On multi-tasking learning: We further explore the potential of a unified framework to jointly handle multiple outfit-based recommendation tasks. To this end, we optimize the following joint objective:

$$\mathcal{L}_m = \mathcal{L}_r + \mathcal{L}_c + \mathcal{L}_d, \tag{27}$$

where $\mathcal{L}_r$, $\mathcal{L}_c$, and $\mathcal{L}_d$ correspond to outfit recommendation, outfit completion, and outlier detection, respectively.

As shown in Table XIII, the multi-task model achieves consistently better performance across all datasets and tasks compared with single-task training. These results demonstrate the benefit of joint optimization and confirm that the learned outfit encoder and user preference representation can be effectively shared across different tasks within a unified framework.

E. Hyper-parameter sensitivity

The concentration parameter κ and the number of negatives K are two important hyper-parameters. The concentration parameter defines the sharpness of the distribution which can be sensitive to the performance. The number of negatives is used for the estimation of likelihood, which gives better estimation when more negatives are sampled. Therefore, in this section, we evaluate the performance of our models on these hyper-parameters.

On different concentration parameters: The prediction model with small κ is flatter and may not be discriminative between positive and negative samples. On the other hand,

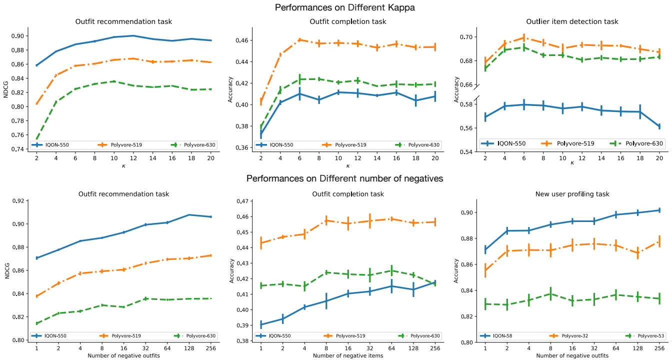


Fig. 7. Performance of different settings. Top row: Impact of different κ values. Bottom row: Impact of different numbers of negatives.

TABLE XIV
Comparison of methods on the Polyvore Outfit dataset with a ResNet-18 backbone, adapted from Table 3 in [20].

Method	Category	Backbone	AUC
Outfit Transformer [20]	–	Frozen	0.82
	–	Fine-tuned	0.91
	✓	Fine-tuned	0.92
Model-r	–	Frozen	0.87
	–	Fine-tuned	0.92
	✓	Fine-tuned	0.92

TABLE XV
Comparison of different methods with frozen ResNet-18.

Method		Polyvore-630		Polyvore-519	
		AUC	NDCG	AUC	NDCG
Dot-GAT [61]	$d = 256$	0.9107	0.8412	0.9237	0.8575
Model-r	$d = 256$	**0.9122**	**0.8457**	**0.9320**	**0.8780**
Model-r	$d = 128$	0.9092	0.8370	0.9282	0.8696
Model-r	$d = 64$	0.8999	0.8196	0.9207	0.8530

model with large κ may overfit, while the back-propagation can be unstable due to large gradient. Therefore, there is an optimal κ for each task as shown in Fig. 7.

On number of negatives: For outlier item detection task, since the likelihood is evaluated over the entire outfit, the probability can be fully computed. Therefore, we only show the impact of using different number of negatives for outfit recommendation and outfit completion tasks. The performance is shown in Fig. 7. For both tasks, using more negatives can improve the performance as the likelihood estimation becomes more accurate with more negatives.

On backbones and feature dimension: n this paper, we extract features using a pre-trained ResNet [75] and project these to a dimensionality of $d = 128$ as the input. Fine-tuning the backbone and increasing the dimensionality usually results in better performance. To demonstrate this, we evaluate the recommendation tasks using different strategies, focusing on whether the backbone is frozen, as illustrated in Table XIV, and altering the dimensionality when frozen, as shown in Table XV.

VIII. Conclusion

In this paper, we have explored various tasks related to outfit composition, which require a deep understanding of the intricate relationships between items in an outfit. We propose a unified prediction model that can be used for all tasks by learning a compact outfit embedding in the same metric of users and items. With the unified framework, we propose a new user profiling strategy to adapt the model for users with limited feedback without fine-tuning the model. We conduct extensive experiments on several large-scale real-world datasets, demonstrating the superiority of our proposed methods across different tasks.

References

[1] Y. Hu, X. Yi, and L. S. Davis, "Collaborative Fashion Recommendation: A Functional Tensor Factorization Approach," in *ACM MM*. ACM, 2015, pp. 129–138.

[2] X. Han, Z. Wu, Y.-G. Jiang, and L. S. Davis, "Learning fashion compatibility with bidirectional lstms," in *Proceedings of the 25th ACM International Conference on Multimedia*, 2017, pp. 1078–1086.

[3] K. Laenen and M.-F. Moens, "A comparative study of outfit recommendation methods with a focus on attention-based fusion," *Information Processing & Management*, vol. 57, no. 6, p. 102316, 2020.

[4] S. Jaradat, N. Dokoohaki, H. J. C. Pampín, and R. Shirvany, "Fashion recommender systems," in *Recommender Systems Handbook*. Springer, 2022, pp. 1015–1055.

[5] Y. Deldjoo, M. Schedl, P. Cremonesi, and G. Pasi, "Recommender systems leveraging multimedia content," *ACM Computing Surveys*, vol. 53, no. 5, pp. 1–38, 2020.

[6] M. I. Vasileva, B. A. Plummer, K. Dusad, S. Rajpal, R. Kumar, and D. A. Forsyth, "Learning type-aware embeddings for fashion compatibility," in *ECCV*. Springer International Publishing, 2018, pp. 390–405.

[7] R. Tan, M. I. Vasileva, K. Saenko, and B. A. Plummer, "Learning similarity conditions without explicit supervision," in *ICCV*, 2019, p. 10.

[8] Z. Lu, Y. Hu, Y. Jiang, Y. Chen, and B. Zeng, "Learning binary code for personalized fashion recommendation," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2019, pp. 10 554–10 562.

[9] Z. Lu, Y. Hu, C. Yu, Y. Jiang, Y. Chen, and B. Zeng, "Personalized fashion recommendation with discrete content-based tensor factorization," *IEEE Transactions on Multimedia*, vol. 25, pp. 5053–5064, 2023.

[10] Y.-L. Lin, S. Tran, and L. S. Davis, "Fashion outfit complementary item retrieval," in *CVPR*, 2020, pp. 3311–3319.

[11] Y. Deldjoo, F. Nazary, A. Ramisa, J. McAuley, G. Pellegrini, A. Bellogin, and T. D. Noia, "A review of modern fashion recommender systems," *ACM Computing Surveys*, vol. 56, no. 4, pp. 87:1–87:37, 2023.

[12] J. McAuley, C. Targett, Q. Shi, and A. van den Hengel, "Image-based recommendations on styles and substitutes," in *SIGIR*. ACM Press, 2015, pp. 43–52.

[13] A. Veit, B. Kovacs, S. Bell, J. McAuley, K. Bala, and S. Belongie, "Learning visual clothing style with heterogeneous dyadic cooccurrences," in *2015 IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2015, pp. 4642–4650.

[14] A. Veit, S. Belongie, and T. Karaletsos, "Conditional similarity networks," in *CVPR*, 2017, pp. 830–838.

[15] Z. Lu, Y. Hu, C. Yu, Y. Chen, and B. Zeng, "Learning fashion compatibility with context conditioning embedding," *IEEE Transactions on Multimedia*, vol. 25, pp. 5516–5526, 2023.

[16] Z. Cui, Z. Li, S. Wu, X.-Y. Zhang, and L. Wang, "Dressing as a whole: Outfit compatibility learning based on node-wise graph neural networks," in *International World Wide Web Conferences*, 2019, pp. 307–317.

[17] Z. Lu, Y. Hu, Y. Chen, and B. Zeng, "Outlier item detection in fashion outfit," in *2022 the 6th International Conference on Innovation in Artificial Intelligence (ICIAI)*. ACM, 2022, pp. 166–171.

[18] Y. Lin, M. Moosaei, and H. Yang, "OutfitNet: Fashion outfit recommendation with attention-based multiple instance learning," in *International World Wide Web Conferences*. Association for Computing Machinery, 2020, pp. 77–87.

[19] X. Li, X. Wang, X. He, L. Chen, J. Xiao, and T.-S. Chua, "Hierarchical fashion graph network for personalized outfit recommendation," in *SIGIR*, 2020.

[20] R. Sarkar, N. Bodla, M. Vasileva, Y.-L. Lin, A. Beniwal, A. Lu, and G. Medioni, "OutfitTransformer: Outfit representations for fashion recommendation," in *WACV*. IEEE, 2022, pp. 2262–2266.

[21] X. Song, X. Han, Y. Li, J. Chen, X.-S. Xu, and L. Nie, "GP-BPR: Personalized compatibility modeling for clothing matching," in *ACM MM*. Association for Computing Machinery, 2019, pp. 320–328.

[22] Z. Lu, Y. Hu, Y. Chen, and B. Zeng, "Personalized outfit recommendation with learnable anchors," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2021, pp. 12 717–12 726.

[23] C. Yu, Y. Hu, Y. Chen, and B. Zeng, "Personalized fashion design," in *ICCV*, 2019, p. 10.

[24] C. Huang, T. Yu, K. Xie, S. Zhang, L. Yao, and J. McAuley, "Foundation models for recommender systems: A survey and new perspectives," 2024.

[25] P. Massa and P. Avesani, "Trust-aware collaborative filtering for recommender systems," in *OTM*. Springer, 2004, pp. 492–508.

[26] J. Lee, Y. Lee, J. Kim, A. Kosiorek, S. Choi, and Y. W. Teh, "Set transformer: A framework for attention-based permutation-invariant neural networks," in *ICML*, 2019, pp. 3744–3753.

[27] Y. Zhang, J. Hare, and A. Prügel-Bennett, "Deep set prediction networks," in *NeurIPS*, 2019.

[28] M. Zaheer, S. Kottur, S. Ravanbakhsh, B. Poczos, R. R. Salakhutdinov, and A. J. Smola, "Deep sets," in *NeurIPS*, 2017, pp. 3394–3404.

[29] A. Vaswani, N. M. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Annual Conference on Neural Information Processing Systems*, vol. 30. Curran Associates, Inc., 2017, pp. 5998–6008.

[30] Y. Deldjoo, N. Rafiee, and M. Ravanbakhsh, "Agentic Personalized Fashion Recommendation in the Age of Generative AI: Challenges, Opportunities, and Evaluation," 2025.

[31] Y. Ding, Z. Lai, P. Mok, and T.-S. Chua, "Computational Technologies for Fashion Recommendation: A Survey," *ACM Computing Surveys*, vol. 56, no. 5, pp. 1–45, 2024.

[32] Y. Wang, L. Liu, X. Fu, and L. Liu, "MCCP: Multi-modal fashion compatibility and conditional preference model for personalized clothing recommendation," *Multimedia Tools and Applications*, vol. 83, no. 4, pp. 9621–9645, 2024.

[33] S. Rendle and L. Schmidt-Thieme, "Pairwise interaction tensor factorization for personalized tag recommendation," in *WSDM*. ACM Press, 2010, pp. 81–90.

[34] Z. Lu, Y. Hu, and B. Zeng, "Sampling for approximate maximum search in factorized tensor," in *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*. International Joint Conferences on Artificial Intelligence Organization, 2017, pp. 2400–2406.

[35] X. Yang, Y. Ma, L. Liao, M. Wang, and T.-S. Chua, "TransNCFM: Translation-based neural fashion compatibility modeling," in *AAAI*, 2019, pp. 403–410.

[36] D. Jang, Q. Li, C. Lee, and J. Kim, "Attention-based multi attribute matrix factorization for enhanced recommendation performance," *Information Systems*, vol. 121, p. 102334, 2024.

[37] Z. Li, D. Jin, and K. Yuan, "Attentional factorization machine with review-based user–item interaction for recommendation," *Scientific Reports*, vol. 13, no. 1, p. 13454, 2023.

[38] X. Song, F. Feng, X. Han, X. Yang, W. Liu, and L. Nie, "Neural compatibility modeling with attentive knowledge distillation," in *SIGIR*, ser. SIGIR '18. ACM, 2018, pp. 5–14.

[39] P. Jing, K. Cui, W. Guan, L. Nie, and Y. Su, "Category-aware multimodal attention network for fashion compatibility modeling," *IEEE Transactions on Multimedia*, vol. 25, pp. 9120–9131, 2023.

[40] D. Mo, X. Zou, K. Pang, and W. K. Wong, "Towards private stylists via personalized compatibility learning," *Expert Systems with Applications*, vol. 219, p. 119632, 2023.

[41] H. Liu, L. Li, N. Yu, K. Ma, T. Peng, and X. Hu, "Outfit compatibility model using fully connected self-adjusting graph neural network," *The Visual Computer*, vol. 40, no. 11, pp. 8331–8343, 2024.

[42] D. Mo, X. Zou, and W. Wong, "Personalized Fashion Recommendation via Deep Personality Learning," in *Proceedings of the 34th British Machine Vision Conference 2023*, 2023, pp. 385–393.

[43] S. Shirkhani, H. Mokayed, R. Saini, and H. Y. Chai, "Study of AI-Driven Fashion Recommender Systems," *SN Computer Science*, vol. 4, no. 5, p. 514, 2023.

[44] L. F. Polanía and S. Gupte, "Learning fashion compatibility across apparel categories for outfit recommendation," in *ICIP*. IEEE, 2019, pp. 4489–4493.

[45] P. Tangseng, K. Yamaguchi, and T. Okatani, "Recommending outfits from personal closet," in *ICCV*, 2017, pp. 2275–2279.

[46] Y. Li, L. Cao, J. Zhu, and J. Luo, "Mining fashion outfit composition using an end-to-end deep learning approach on set data," *IEEE Transactions on Multimedia*, vol. 19, no. 8, pp. 1946–1955, 2017.

[47] W. Chen, P. Huang, J. Xu, X. Guo, C. Guo, F. Sun, C. Li, A. Pfadler, H. Zhao, and B. Zhao, "POG: Personalized outfit generation for fashion recommendation at alibaba iFashion," in *KDD*. Association for Computing Machinery, 2019, pp. 2662–2670.

[48] Y. Ding, P. Mok, Y. Ma, and Y. Bin, "Personalized fashion outfit generation with user coordination preference learning," *Information Processing & Management*, vol. 60, no. 5, p. 103434, 2023.

[49] C.-L. Chang, Y.-L. Chen, and D.-X. Jiang, "Using large multimodal models to predict outfit compatibility," *Decision Support Systems*, vol. 194, p. 114457, 2025.

[50] H. Liu, X. Tang, T. Chen, J. Liu, I. Indu, H. P. Zou, P. Dai, R. F. Galan, M. D. Porter, D. Jia, N. Zhang, and L. Xiong, "Sequential LLM Framework for Fashion Recommendation," in *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2024, pp. 1276–1285.

[51] M. Yu, Y. Ma, L. Wu, C. Wang, X. Li, and L. Meng, "FashionDPO:Finetune Fashion Outfit Generation Model using Direct Preference Optimization," in *The 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 2025, pp. 212–222.

[52] Y. Xu, W. Wang, F. Feng, Y. Ma, J. Zhang, and X. He, "Diffusion models for generative outfit recommendation," in *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2024, pp. 1350–1359.

[53] D. Zhou, H. Zhang, J. Ma, and J. Shi, "BC-GAN: A Generative Adversarial Network for Synthesizing a Batch of Collocated Clothing," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 34, no. 5, pp. 3245–3259, 2024.

[54] A. H. Vo, T. B. T. Le, H. V. Pham, and B. T. Nguyen, "An efficient framework for outfit compatibility prediction towards occasion," *Neural Computing and Applications*, vol. 35, no. 19, pp. 14 213–14 226, 2023.

[55] F. Becattini, F. M. Teotini, and A. D. Bimbo, "Transformer-Based Graph Neural Networks for Outfit Generation," *IEEE Transactions on Emerging Topics in Computing*, vol. 12, no. 1, pp. 213–223, 2024.

[56] H. Zhan, J. Lin, K. E. Ak, B. Shi, L.-Y. Duan, and A. C. Kot, "A3-FKG: Attentive attribute-aware fashion knowledge graph for outfit preference prediction," *IEEE Transactions on Multimedia*, vol. 24, pp. 819–831, 2022.

[57] J. Tang, X. Shu, Z. Li, Y.-G. Jiang, and Q. Tian, "Social anchor-unit graph regularized tensor completion for large-scale image retagging," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 8, pp. 2027–2034, 2019.

[58] J. Zhou, G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun, "Graph neural networks: A review of methods and applications," *AI Open*, vol. 1, pp. 57–81, 2020.

[59] Y. Li, G. Li, J. Zhang, P. Jing, and X. Lu, "Research on type-aware fashion compatibility prediction based on a hybrid attention mechanism," *Multimedia Tools and Applications*, vol. 83, no. 30, pp. 74 003–74 020, 2024.

[60] S. Saed and B. Teimourpour, "Hybrid-Hierarchical Fashion Graph Attention Network for Compatibility-Oriented and Personalized Outfit Recommendation," 2025.

[61] B. S. Vivek, G. Bhattacharya, J. Gubbi, B. L. V, A. Pal, and P. Balamuralidhar, "Personalized outfit compatibility prediction using outfit graph network," in *2023 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2023, pp. 1–8.

[62] G. Cucurull, P. Taslakian, and D. Vazquez, "Context-aware visual compatibility prediction," in *CVPR*, 2019.

[63] X. Yang, X. Du, and M. Wang, "Learning to match on graph for fashion compatibility modeling," in *AAAI*, 2020.

[64] R. Xu, J. Wang, and Y. Li, "Cross-Intent Graph Contrastive Learning for Fashion Sequential Recommendation," in *2023 28th International Conference on Automation and Computing*. IEEE, 2023, pp. 1–6.

[65] J. Wu, Y. Xu, B. Zhang, Z. Xu, and B. Li, "Graph-based Dynamic Preference Modeling for Personalized Recommendation," in *Advances in Knowledge Discovery and Data Mining*. Springer Nature, 2024, pp. 356–368.

[66] Z. Han, C. Hu, T. Li, Q. Qi, P. Tang, and S. Guo, "Subgraph-level federated graph neural network for privacy-preserving recommendation with meta-learning," *Neural Networks*, vol. 179, p. 106574, 2024.

[67] G. Tan, "NAH-GNN: A graph-based framework for multi-behavior and high-hop interaction recommendation," *PLOS ONE*, vol. 20, no. 4, p. e0321419, 2025.

[68] A. Yan, C. Dong, Y. Gao, J. Fu, T. Zhao, Y. Sun, and J. Mcauley, "Personalized complementary product recommendation," in *Companion*

Proceedings of the Web Conference, ser. WWW '22. Association for Computing Machinery, 2022, pp. 146–151.

[69] K. V. Mardia, P. E. Jupp, and K. V. Mardia, *Directional Statistics*. Wiley Online Library, 1999, vol. 2.

[70] K. Sohn, "Improved deep metric learning with multi-class n-pair loss objective," in *NeurIPS*, 2016, pp. 1857–1865.

[71] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *ICML*. PMLR, 2020, pp. 1597–1607.

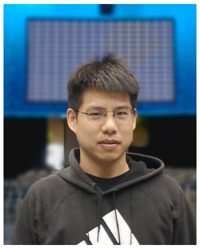
[72] B. Poole, S. Ozair, A. van den Oord, A. A. Alemi, and G. Tucker, "On variational bounds of mutual information," in *ICML*, 2019.

[73] Z. Ma and M. Collins, "Noise contrastive estimation and negative sampling for conditional models: Consistency and statistical efficiency," in *EMNLP*, 2018.

[74] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," 2016.

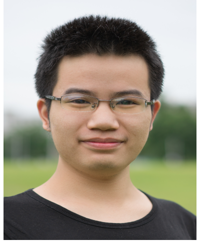
[75] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2016, pp. 770–778.

[76] N. Qian, "On the momentum term in gradient descent learning algorithms," *Neural Networks*, vol. 12, no. 1, pp. 145–151, 1999.

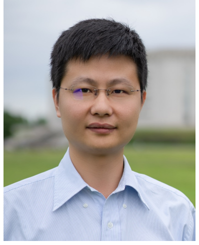


Zhi Lu received the B.S., M.Phil., and Ph.D. degrees from the University of Electronic Science and Technology of China, Chengdu, China, in 2015, 2018, and 2022, respectively. He is currently a Research Associate Professor at the School of Cyber Science and Technology, University of Science and Technology of China, Hefei, China. His current research interests include computer vision, machine learning, and multimedia signal processing.

Yang Hu received the B.S. and Ph.D. degrees in electrical engineering from the University of Science and Technology of China, Hefei, China, in 2004 and 2009 respectively. She was with the University of Maryland Institute for Advanced Computer Studies as a research associate from 2010 to 2015. She is currently an associate professor with the School of Information Science and Technology at the University of Science and Technology of China. Her current research interests include computer vision, machine learning and multimedia signal processing.



Cong Yu received the B.S. and Ph.D degrees from University of Electronic Science and Technology of China, Chengdu, China, in 2019 and 2023, respectively. He is currently an Assistant Researcher at Institute of Electronic Engineering, China Academy of Engineering Physics, Mianyang, China. His research interests include lightweight model, object detection, and wireless sensing.



Yan Chen (SM'14) received the bachelor degree from the University of Science and Technology of China in 2004, the M.Phil. degree from the Hong Kong University of Science and Technology in 2007, and the Ph.D. degree from the University of Maryland, College Park, MD, USA, in 2011. He was with Origin Wireless Inc. as a Founding Principal Technologist. From Sept. 2015 to Feb. 2020, he was a Professor with the School of Information and Communication Engineering at the University of Electronic Science and Technology of China. He is currently a Professor with the School of Cyber Science and Technology at the University of Science and Technology of China.

Dr. Chen's research interests include multimodal sensing and imaging, multimedia signal processing, and wireless multimedia. He is a co-author of "Reciprocity, Evolution, and Decision Games in Network and Data Science" (Cambridge University Press, 2021) and "Behavior and Evolutionary Dynamics in Crowd Networks: An Evolutionary Game Approach" (Springer, 2020), as well as co-author of over 200 technical papers including more than 100 IEEE journal papers. He is the Associate Editor for IEEE Transactions on Network Science and Engineering (TNSE) and IEEE Transactions on Signal and Information Processing over Networks (TSIPN). He is the Chair for APSIPA Signal and Information Processing Theory and Methods (SIPTM) Technical Committee, a Distinguished Lecturer for APSIPA, the Secretary-General for the CES Young Scientist Network Multimedia Technical Committee. He is an Organizing Co-Chair of PCM 2017, a Special Session Co-Chair of APSIPA ASC 2017, the 10K Best Paper Award Committee Member of ICME 2017, the Multimedia Communications Symposium Lead Chair of WCSP 2019, an Area Chair for ACM Multimedia 2021, a TPC Co-Chair of APSIPA ASC 2021. He was the recipient of multiple honors and awards, including an Excellent Editor for IEEE TNSE in 2021, the best paper award at the APSIPA ASC in 2020, the best student paper award at the PCM in 2017, the best student paper award at the IEEE ICASSP in 2016, the best paper award at the IEEE GLOBECOM in 2013.

Bing Zeng (M'91-SM'13) received the B.E. and M.Sc. degrees in Electronic Engineering from the University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 1983 and 1986, respectively, and the Ph.D. degree in Electrical Engineering from the Tampere University of Technology, Tampere, Finland, in 1991. He worked as a Postdoctoral Fellow with the University of Toronto from September 1991 to July 1992 and as a Researcher with Concordia University from August 1992 to January 1993. Then he joined the Hong Kong University of Science and Technology (HKUST). After 20 years of service, he returned to the UESTC in the summer of 2013, through Chinas 1000-Talent-Scheme. At UESTC, he leads the Institute of Image Processing, which works on image and video processing, 3-D and multiview video technology, and visual big data. During his tenure with the HKUST and UESTC, he graduated more than 30 Master's and Ph.D. students, received about 20 research grants, filed 8 international patents, and published more than 250 papers. He served as an Associate Editor for the IEEE TCSVT for 8 years and received the Best Associate Editor Award in 2011. He was General Co-Chair of the IEEE VCIP-2016, held in Chengdu, China, in November 2016. He is currently on the Editorial Board of the Journal of Visual Communication and Image Representation and serves as General Co-Chair of PCM-2017. He was the recipient of a 2nd Class Natural Science Award (the first recipient) from the Ministry of Education of China in 2014 and was elected as an IEEE Fellow in 2016 for contributions to image and video coding.