applied sciences

MDPI

Article

Clothing Recommendation with Multimodal Feature Fusion: Price Sensitivity and Personalization Optimization

Chunhui Zhang [1], Xiaofen Ji [2,3] and Liling Cai [2,*]

[1] School of Fashion Design and Engineering, Zhejiang Sci-Tech University, Hangzhou 310018, China; zhscitech@gmail.com
[2] International Fashion Technology College, Zhejiang Sci-Tech University, Hangzhou 310018, China; xiaofenji@zstu.edu.cn
[3] China National Silk Museum, Hangzhou 310002, China
* Correspondence: caililing@zstu.edu.cn

Abstract: The rapid growth in the global apparel market and the rise of online consumption underscore the necessity for intelligent clothing recommendation systems that balance visual compatibility with personalized preferences, particularly price sensitivity. Existing recommendation systems often neglect nuanced consumer price behaviors, limiting their ability to deliver truly personalized suggestions. To address this gap, we propose DeepFMP, a multimodal deep learning framework that integrates visual, textual, and price features through an enhanced DeepFM architecture. Leveraging the IQON3000 dataset, our model employs ResNet-50 and BERT for image and text feature extraction, alongside a comprehensive price feature module capturing individual, statistical, and category-specific price patterns. An attention mechanism optimizes multimodal fusion, enabling robust modeling of user preferences. Comparative experiments demonstrate that DeepFMP outperforms state-of-the-art baselines (LR, FM, Wide & Deep, GP-BPR, and DeepFM), achieving AUC improvements of 1.6–12.2% and NDCG@10 gains of up to 3.2%. Case analyses further reveal that DeepFMP effectively improves the recommendation accuracy, offering actionable insights for personalized marketing. This work advances multimodal recommendation systems by emphasizing price sensitivity as a pivotal factor, providing a scalable solution for enhancing user satisfaction and commercial efficacy in fashion e-commerce.

Keywords: deep learning; multimodal fusion; consumer preference; garment recommendation; price sensitivity

Academic Editor: Andrea Prati

Received: 1 March 2025
Revised: 8 April 2025
Accepted: 17 April 2025
Published: 21 April 2025

Citation: Zhang, C.; Ji, X.; Cai, L. Clothing Recommendation with Multimodal Feature Fusion: Price Sensitivity and Personalization Optimization. *Appl. Sci.* **2025**, *15*, 4591. https://doi.org/10.3390/app15084591

1. Introduction

According to Statista's projections, the global apparel market is expected to reach USD 1.79 trillion by 2024, with an annual compound growth rate of 2.81% from 2024 to 2028. The apparel industry is a significant sector in economic development, particularly with the rapid growth in online consumption in recent years, making intelligent clothing shopping a key trend in the fashion retail industry [1]. Intelligent clothing shopping systems typically utilize personalized recommendation algorithms, virtual fitting technologies, and data-driven analytics to meet users' personalized needs. Especially in the era of big data, with the explosive growth in data, recommendation systems have become a crucial technology [2] in clothing consumption. E-commerce platforms can collect vast amounts of data on users' purchase histories, browsing records, search keywords, and product click-through rates, enabling businesses to effectively conduct personalized marketing and enhance user experiences.

Simultaneously, the rapid rise of international social platforms, such as Xiaohongshu, Instagram, and TikTok, has led to an increasing number of people sharing their outfit photos and clothing combinations on these platforms, showcasing personal styles. This trend has not only transformed shopping habits but also provided a rich data source for clothing brands and retailers. Supported by big data and recommendation algorithms, businesses can extract value from user-generated content (UGC), analyze similarities between users and clothing products, and thereby create more detailed user profiles, improve personalized recommendation accuracy, enhance user engagement, strengthen brand loyalty, and increase purchase rates.

In the context of the big data era, collaborative filtering algorithms recommend items to users by calculating item similarities, while content-based recommendation systems make recommendations by analyzing the match between item content features and users' historical preferences [3]. In the deep learning context, convolutional neural networks are used to extract information from images and texts for relevant recommendations. However, current recommendation systems often fail to account for consumers' price sensitivity and consumption preferences, which vary significantly across individuals. Some consumers prioritize cost-effectiveness [4], while others focus on fashion or brand appeal. This gap limits the ability to provide truly personalized recommendations that align with users' financial expectations and preferences.

To address these limitations, we propose DeepFMP, a novel multimodal recommendation framework that integrates textual descriptions, image features, and price information. By leveraging a deep learning-based approach, DeepFMP captures the complex interactions between these modalities, providing more accurate and personalized outfit recommendations. Our framework not only meets the visual requirements of fashion coordination but also aligns with consumers' reasonable price expectations, thereby enhancing the effectiveness and accuracy of recommendation systems [5]. The main contributions of this paper are summarized as follows:

1. From the perspective of price sensitivity, a price feature extraction module was designed to capture characteristics such as clothing brand, category, and user historical behavior, enabling the model to provide recommendations aligned with users' consumption preferences.
2. By leveraging transfer learning, pre-trained models were introduced into the domain of clothing outfit recommendation, enhancing the depth and breadth of feature representation in the model's initial input vectors.
3. In the context of multimodal recommendation, we propose an effective recommendation framework that integrates a multi-head attention mechanism into the multimodal recommendation algorithm. This approach explores optimized methods for interaction and fusion across different modalities, thereby improving the model's ability to learn from diverse modal features.

2. Related Work

With the advancement of machine learning and deep learning, recommendation algorithms have progressively evolved from linear models to deep neural networks, incorporating personalized ranking and probabilistic modeling approaches. As a representative of generalized linear models, logistic regression was widely adopted in early recommendation systems for click-through rate (CTR) prediction tasks. While logistic regression demonstrates advantages in computational efficiency and strong interpretability, its inherent linear nature inherently limits automatic capture of feature interaction effects, necessitating the manual construction of cross-features. To overcome the limitations of linear models, Steffen [6] proposed the Factorization Machine (FM) model in 2010, which

effectively models second-order feature interactions through latent vector inner products. This innovation significantly reduces manual feature engineering costs while maintaining robust generalization performance under sparse data conditions. Owing to its simplicity, efficiency, and scalability, the FM model has become one of the fundamental architectures in recommendation systems.

The hybrid recommendation system architecture Wide&Deep, proposed by the Google team [7] in 2016, marked the advent of nonlinear deep models in recommendation systems by simultaneously achieving memorization and generalization capabilities. However, its Wide component still requires manual feature engineering, and the dual-pathway joint training may induce gradient conflicts. To address these limitations, Guo et al. [8] developed the DeepFM model in 2017 through end-to-end learning optimization. This architecture integrates FM with deep neural networks (DNNs), sharing embedding layer parameters between both components to automatically capture feature interactions from second-order to higher-order levels. The DeepFM framework promotes the evolution of recommendation systems toward full automation while demonstrating exceptional architectural flexibility and extensibility.

The rapid development of the fashion industry has spurred research on apparel recommendation algorithms, particularly in garment compatibility modeling. In 2019, Song et al. [9] proposed GP-BPR, a compatibility modeling framework for personalized clothing outfit recommendations. This framework employs ResNet50 for visual feature extraction while adopting a hybrid approach combining Word2vec and textCNN for textual feature extraction, thereby pioneering a novel multimodal feature extraction methodology for fashion recommendation systems. Although this approach has been successfully applied to personalized outfit coordination scenarios, its relatively high computational complexity imposes limitations on large-scale industrial applications.

Multimodal feature extraction is critical for fashion recommendations, encompassing visual, textual, and price features. Early methods relied on manual feature engineering, but advancements in pre-trained models and transfer learning now enable direct extraction of high-dimensional dense features. In computer vision, convolutional neural networks (CNNs), particularly ResNet [10], have excelled in extracting visual features like material, color, and texture [11]. Miao et al. [12] tackled the cross-domain clothing retrieval problem by integrating the ResNet50 model with a quadruplet loss function. Xuan et al. [13] proposed integrating the Convolutional Block Attention Module (CBAM) into ResNet50 to improve feature extraction capabilities for garment regions, leading to a notable increase in retrieval accuracy. Recent work by Mu et al. [14] and Guo et al. [15] further enhanced feature extraction through multi-level fusion and attention mechanisms. In natural language processing (NLP), word embedding techniques have evolved from static representations like Word2Vec [16] to contextual models like BERT [7,17], which captures nuanced linguistic relationships. Gao et al. [18] proposed FashionBERT, which employs the BERT model as its backbone network and integrates textual descriptions with garment images for cross-modal retrieval, thereby improving matching accuracy. Li et al. [19] introduced an attribute-aware interpretable complementary clothing recommendation model based on the BERT framework, incorporating multimodal features and attention mechanisms to generate attribute-based outfit matching explanations, significantly enhancing both recommendation accuracy and user trust. For this study, we employ BERT-base-Japanese and ResNet-50 for text and image feature extraction, respectively.

Price sensitivity is another crucial factor in fashion recommendations, yet traditional systems often overlook its nuanced role [20]. Early approaches treated price as a static attribute, while FM-based methods [6,8] integrated it with other features but struggled with category-dependent effects and implicit user sensitivity [21]. Recent graph-based models,

such as Price-aware User Preference-modeling (PUP) [20], use Graph Convolutional Networks (GCNs) to model price interactions across users and categories, capturing both global and category-specific effects. Techniques like uniform or rank-based discretization [22] further enhance price-aware systems, enabling seamless integration with visual and textual features. Lin et al. [23] proposed the GPCTR model, which integrates collaborative filtering with topic modeling to effectively fuse multimodal data comprising textual reviews and numerical information (e.g., salary data). This hybrid approach significantly enhances both predictive accuracy and interpretability in employer brand attractiveness assessment. These advancements provide a robust foundation for multimodal recommendation systems.

3. The Proposed Model

The purpose of this study is to develop an advanced clothing recommendation model (DeepFMP) that leverages multimodal feature fusion to enhance user satisfaction by integrating price sensitivity and personalization optimization. The proposed recommendation system mainly includes two modules: feature extraction and recommendation model, as seen in Figure 1. The core idea and significant innovation of this research proposal lie in introducing an improved DeepFM architecture to extract price information, text, and image features, followed by the effective fusion of multimodal features.

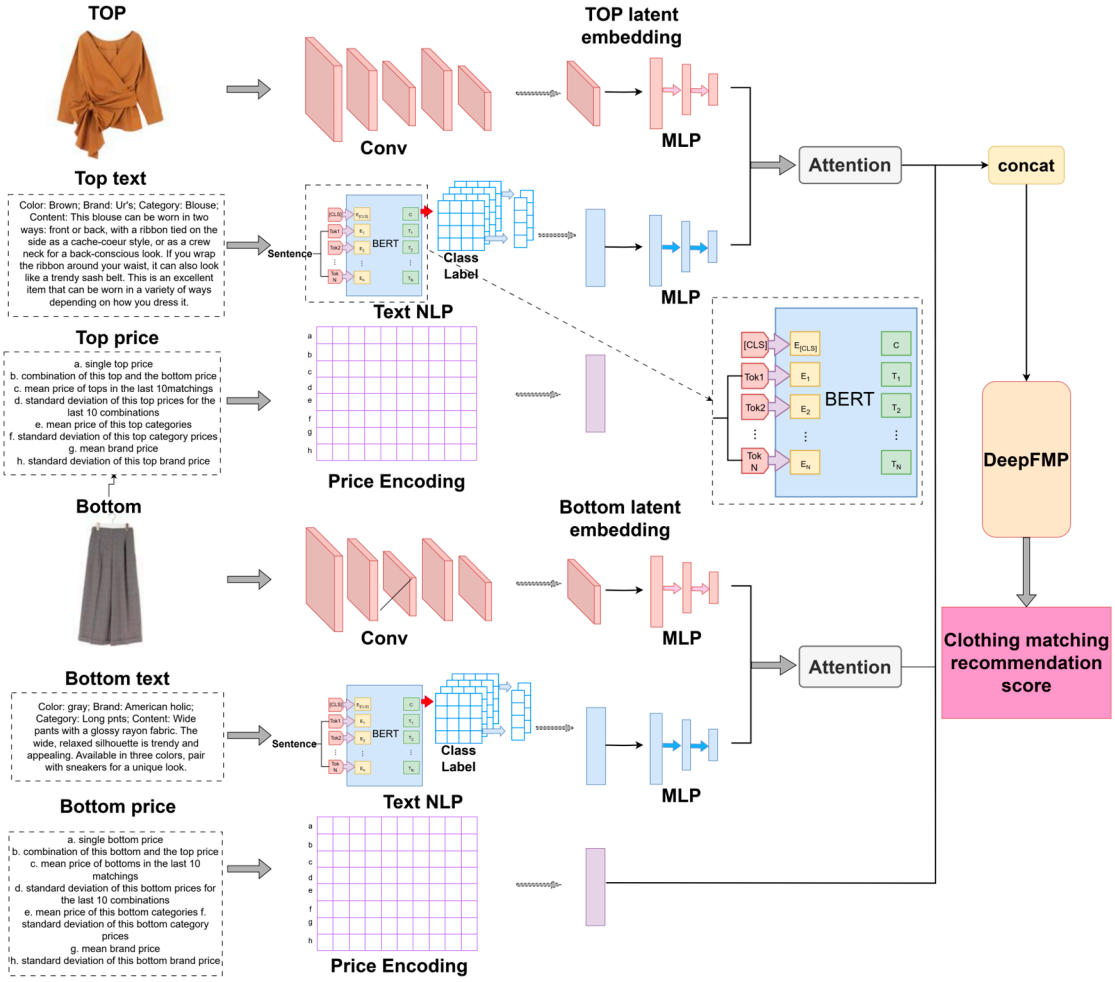

Figure 1. Overall design of clothing recommendation system.

3.1. Feature Extraction and Embedding

In this study, the input to the DeepFMP model is composed of three key modalities: textual features, visual features, and price feature. In order to fully explore the deep

information in clothing matching data, this study uses a pre-trained model-based approach to extract text and visual features, while price features are extracted through data mining methods [24], and a variety of cross-features are designed to deeply explore the potential impact of price on clothing matching.

3.1.1. Image Feature Extraction and Embedding

In this study, the goal is to transform apparel images into a format that can efficiently support continuous learning models for predicting apparel characteristics. To achieve this, we applied a transfer learning strategy, utilizing the ResNet50 model pre-trained on the ImageNet dataset to extract visual features. The ResNet50, based on the residual neural network architecture (ResNet), offers strong performance and the ability to capture both low-level and high-level features in the images [25].

To standardize the data, we first performed a pre-processing operation on the apparel images. We rescaled the original image $I_{orig} \in \mathbb{R}^{H \times W \times C}$, where H is the height of the image, W is the width, and C represents the color channels [26]. To meet the input requirements of ResNet50, we resized the image to 224 × 224 pixels, and stored the RGB channels, representing the input $I_{input} \in \mathbb{R}^{224 \times 224 \times 3}$, where the RGB channels are mapped to the values in the range [0, 255]. The image processing is as follows:

For efficient neural network training and speed, the pixel values of the image are first normalized from the range [0, 255] to [0, 1], with the original pixel value $I_{input}(i, j, k)$ being transformed to:

$$I_{norm}(i, j, k) = \frac{I_{input}(i, j, k)}{255}$$

For the image values to be more consistent, the images were standardized using ImageNet data. The mean values for the ImageNet dataset are [0.485, 0.456, 0.406], and the standard deviations for the ImageNet dataset are [0.229, 0.224, 0.225]. Thus, the normalization process was performed as follows:

$$I_{std}(i, j, k) = \frac{I_{norm}(i, j, k) - \mu_k}{\sigma_k}$$

Next, we used ResNet50 for feature extraction. ResNet50 performed convolution operations to extract high-level features from the input images and produced a feature vector $F_{resnet} \in \mathbb{R}^{H' \times W' \times D}$, where D is the depth of the convolution (i.e., the number of channels) [26]. The feature extraction process can be expressed as:

$$F_{resnet} = \text{Conv2D}(I_{std}, W_{conv}, b_{conv})$$

where W_{conv} and b_{conv} are the weights and biases of the convolution layers. In this layer, ResNet50 extracts both low-level features such as garment textures and high-level features such as clothing styles and color patterns.

The extracted feature map was subsequently passed through a fully connected layer, which utilized the GELU activation function to introduce non-linearity and enhance the model's ability to capture complex relationships. The flattening operation was applied to reduce the high-dimensional feature map into a one-dimensional vector:

$$F_1 = \text{GELU}(W_1 \cdot \text{Flatten}(F_{resnet}) + b_1)$$

where $W_1 \in \mathbb{R}^{dm1 \times D}$ is the weight matrix and $b_1 \in \mathbb{R}^{dm1}$ is the bias vector, the GELU function serves as the activation function [2], and Flatten denotes the operation of flattening the feature map into a one-dimensional vector. To mitigate overfitting, a dropout layer is

incorporated after the fully connected layer [27], with a dropout rate set to p. During each training iteration, a random proportion p of neurons are discarded.

Subsequently, the resulting $F_1 \in \mathbb{R}^{dm1}$ is compressed into the final visual feature vector through a second fully connected layer, with an output dimensionality of $dm2 = 300$. The second fully connected layer is defined as:

$$E_1 = W_2 \cdot F_1 + b_2$$

where $W_2 \in \mathbb{R}^{dm2 \times dm1}$ is the weight matrix and $b_2 \in \mathbb{R}^{dm2}$ is the bias term. The resulting feature vector $E_1 \in \mathbb{R}^{300}$ serves as the visual feature representation of the clothing image and is passed to the subsequent recommendation algorithm model. This image feature vector encapsulates both low-level features of the clothing image, such as edges and textures, and high-level semantic features, such as shape, category, and color. The recommendation system leverages these feature vectors in conjunction with user consumption preference data and product attributes to achieve personalized clothing recommendations.

3.1.2. Text Feature Extraction and Embedding

The objective of this phase is to transform raw textual data into high-dimensional dense vectors by employing a pre-trained language model [28]. In this context, we developed a text feature extractor based on the principles of transfer learning, utilizing the pre-trained Bidirectional Encoder Representations from Transformers (BERT) model, which is grounded in the Transformer architecture. The raw textual information utilized in this experiment comprises fashion item titles and detailed descriptions. Given that the IQON3000 dataset was scraped from web sources, it contains certain special characters and advertising terminology. Upon extracting vectors directly from the raw text and computing the cosine similarity between various fashion items, we observed that the presence of similar advertising terminology significantly inflated the overall similarity scores [29]. To achieve a more precise representation of fashion item texts and to more accurately and independently describe the attribute characteristics of fashion items, this experiment employed regular expression matching for data cleansing. This process involved removing special characters used to denote spaces in web data and filtering out high-similarity advertising terms across product text representations. The cleansed text data were subsequently fine-tuned using the BERT model.

BERT, a model based on the Transformer architecture, is typically utilized in two primary modes: unsupervised pre-training and supervised fine-tuning [30]. For our pre-trained BERT model, we employed BERT_base_japanese. This model architecture comprises 12 layers, a hidden layer dimension of 768, and 12 multi-head self-attention heads. The pre-training data were derived from the Japanese Wikipedia corpus, and fine-tuning was conducted using the supervised fine-tuning (SFT) approach. Given the unique characteristics of Japanese text, we initially processed the input text using the MeCab morphological analyzer with the IPA dictionary for token-level segmentation. Subsequently, the text was subdivided into subwords via the WordPiece algorithm. Considering the average length of the text data and the computational efficiency of the model, we truncated the input text length to 256 tokens. Notably, to effectively capture the textual vector representations of fashion items, we applied average pooling to both the first and last layers of the BERT output.

$$T = \frac{1}{2}\Big(Pool(BERT_{first}) + Pool(BERT_{last})\Big)$$

where $\mathrm{Pool}(\cdot)$ denotes average pooling. The first layer retains surface-level textual features, while the last layer preserves semantic information [31]. The intermediate vectors obtained

from these two layers were concatenated and then subjected to average pooling, resulting in a 768-dimensional raw text vector for each fashion item. Finally, a fully connected layer followed by a ReLU layer was employed to reduce the output dimensionality to $d_n = 300$, yielding the ultimate textual feature vector representation.

$$E_2 = ReLU(W_t \cdot T + b_t)$$

3.1.3. Price Feature Extraction and Embedding

In the process of mining clothing price data, to enhance the precision of the analysis, this study constructed a series of multi-dimensional price features. These features include the price of individual clothing items, such as the price of tops, bottoms, and the combined price of top and bottom sets, as well as the mean and standard deviation of the prices of tops, bottoms, and combined sets from the user's last 10 historical outfit records [32]. Additionally, the mean and standard deviation of prices related to clothing categories were introduced to further delineate the impact of clothing categories on price distribution [33]. To delve deeper into users' consumption preferences, we also extracted historical average price features for brands and categories, thereby capturing users' price preference patterns across different brands and categories.

In the specific implementation process, the extraction of price features is shown in Figure 2:

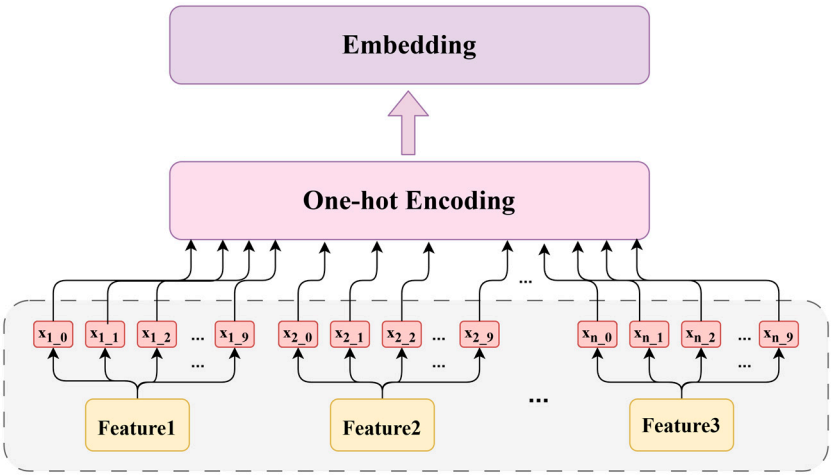

Figure 2. Price feature extraction.

(1) Single item price feature

$$P_{top} = \text{price}(\text{top}), P_{bottom} = \text{price}(\text{bottom})$$

where P_{top} and P_{bottom} represent the prices of the top and bottom clothing items, respectively.

(2) Combination Price Feature

$$P_{combo} = P_{top} + P_{bottom}$$

where P_{combo} denotes the combined price of the top and bottom clothing items.

(3) Price Statistical Features from Historical Outfit Records

For the user's last 10 historical outfit records, the mean and standard deviation of the prices for tops, bottoms, and combinations are calculated as follows:

$$\mu_{\text{top}} = \frac{1}{N}\sum_{i=1}^{N} P_{\text{top}}^{(i)}, \sigma_{\text{top}} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(P_{\text{top}}^{(i)} - \mu_{\text{top}}\right)^2}$$

where μ_{top} and σ_{top} represent the mean and standard deviation of the top prices, respectively [5], $P_{\text{top}}^{(i)}$ denotes the price of the top in the i-th outfit record, and N is the number of historical outfit records. The same formulas apply to the bottom and combination prices. These features capture the user's recent price preferences and acceptance levels in clothing selection, while the standard deviation reflects the user's budget consistency or interest in different price ranges across various outfits.

(4) Category Price Mean and Standard Deviation

For each clothing category, the mean and standard deviation of prices for all items within that category are calculated as follows:

$$\mu_{\text{category}} = \frac{1}{M}\sum_{i=1}^{M} P_{\text{category}}^{(i)}, \sigma_{\text{category}} = \sqrt{\frac{1}{M}\sum_{i=1}^{M}\left(P_{\text{category}}^{(i)} - \mu_{\text{category}}\right)^2}$$

where μ_{category} and σ_{category} represent the mean and standard deviation of prices within the category, respectively, $P_{\text{category}}^{(i)}$ denotes the price of the i-th item in the category, and M is the total number of items in the category. These metrics reflect the overall pricing level and price stability within the category, providing insights into the general pricing trends and variability for the specific clothing category.

(5) Brand Price Mean and Standard Deviation

$$\mu_{\text{brand}}(b) = \frac{1}{K_b}\sum_{i=1}^{K_b} P_{\text{brand}}^{(i)}, \sigma_{\text{brand}}(b) = \sqrt{\frac{1}{K_b}\sum_{i=1}^{K_b}\left(P_{\text{brand}}^{(i)} - \mu_{\text{brand}}(b)\right)^2}$$

where $\mu_{\text{brand}}(b)$ and $\sigma_{\text{brand}}(b)$ represent the mean and standard deviation of prices for brand b, respectively, $P_{\text{brand}}^{(i)}$ denotes the price of the i-th clothing item under brand b, and K_b is the total number of clothing items under brand b. Due to variations in pricing strategies among different brands, some brands are positioned in the high-end market with relatively higher prices, while others target the mass market with a focus on high cost-performance, resulting in relatively lower prices. Additionally, certain brands may cater to multiple market segments, leading to significant price fluctuations. By extracting these features, it is possible to effectively capture the pricing strategy characteristics across different brands.

(6) User Brand Preference Price Features

For the historical purchase or outfit records of user u, we further extract price preference features at the brand level. These features include the mean and standard deviation of the user's prices for brand b:

$$\mu_{\text{user-brand}}(u, b) = \frac{1}{N_b}\sum_{i=1}^{N_b} P_{\text{user-brand}}^{(i)}$$

$$\sigma_{\text{user-brand}}(u, b) = \sqrt{\frac{1}{N_b}\sum_{i=1}^{N_b}\left(P_{\text{user-brand}}^{(i)} - \mu_{\text{user-brand}}(u, b)\right)^2}$$

where $\mu_{\text{user-brand}}(u, b)$ and $\sigma_{\text{user-brand}}(u, b)$ represent the mean and standard deviation of prices for user u under brand b, respectively, $P^{(i)}_{\text{user-brand}}$ denotes the price of the i-th clothing item purchased or matched by user u under brand b, and N_b is the number of historical records of user u for brand b. By statistically analyzing the user's historical behavior in selecting clothing items across different brands, the recommendation system can infer the user's price tolerance for various brands, thereby enhancing the personalization of the recommendation results [34,35].

(7) Discretization of Price Features

To enhance the model's ability to handle continuous variables, all continuous price features undergo discretization. Specifically, an equal-frequency binning method is employed to divide each continuous feature into 10 groups, which are then transformed into a sparse feature matrix using one-hot encoding. The implementation process is as follows:

Assume that the continuous price feature P_{price} is discretized into K intervals, each representing a discrete category $\{C_1, C_2, \ldots, C_K\}$. If the price feature value P_{price} of a data-point falls into the j-th interval, the discretized category label for that data point is C_j.

(8) One-Hot Encoding

For each discretized category label C_j, its corresponding one-hot encoding is represented as a K-dimensional vector. Given N samples, the feature vector after discretization is expressed as:

$$E_3 = \begin{bmatrix} x_1^T \\ x_2^T \\ \vdots \\ x_N^T \end{bmatrix} \quad (1)$$

where x_i denotes the one-hot encoded vector for the i-th sample, with a dimensionality of K. Through this method, a sparse matrix E_3 is obtained, where each row corresponds to the one-hot encoding of the discretized price feature for a sample.

After one-hot encoding transformation, each price feature is represented as a sparse vector, with a non-zero value (1) only at the position corresponding to its interval and zeros elsewhere. This approach helps reduce the nonlinear complexity of the model when processing continuous features and facilitates efficient matrix computations by the computer.

3.2. Recommendation Model

Through the above feature extraction modules, we obtained feature vectors from three distinct dimensions. For the textual and visual features, we designed a multi-head self-attention module [30]. The multi-head self-attention mechanism aids in capturing subtle feature information within single-modal features, with each head representing an independent subspace that can apply feature weighting to different regions of the input feature vector. By employing the same multi-head self-attention module for both textual and visual features, and leveraging the weight-sharing mechanism of the self-attention module, we not only perform weighted scaling on the feature vectors by MLP [36], but also facilitate information interaction between the two modalities through the self-attention layer [37]. This approach helps establish connections between textual and visual features.

For the base model of our recommendation algorithm, we considered the need to integrate multiple dimensions and types of feature data, and thus decided to reference the DeepFM model structure. Starting with the multi-modal feature combination input, we designed a series of embedding layers and fully connected layers to map and transform the input feature vectors. For the sparse feature matrix input [38], we designed two sets of

embedding layers to convert discrete, high-dimensional sparse vectors into continuous, low-dimensional dense vectors. The first embedding layer maps each sparse feature into a vector of dimension $d_{s1} = 1$, denoted as $embed_{s1}$, which is used for the first-order summation in the FM module. The second embedding layer projects each sparse feature into a vector of dimension $d_{s2} = n$, denoted as $embed_{s2}$, which is passed to the second-order summation module and the DNN module of the FM.

For the dense textual and visual features of the clothing data, we also designed two sets of fully connected layers. The first set of fully connected layers concatenates all received dense features and maps them into a vector of dimension d_{d1}, denoted as $embed_{d1}$, which is used for the first-order summation in the FM module. The second set of fully connected layers maps each received dense feature individually, generating new low-dimensional dense feature representations $embed_{d2}$ with dimension $d_{d2} = m$, which are passed to the second-order summation module and the DNN module of the FM. To balance computational performance and accuracy, we adjusted the hyperparameters $d_{s2} = n$ and $d_{d2} = m$. For the sparse features represented by $d_{s2} = n$, we categorized them into ID-type features and statistical-type features based on their data type and statistical quantity. For ID-type features, we set the vector dimension $d_{s2\_1} = 8$ to obtain the vector $embed_{s2\_1}$, while for statistical-type features, we set the vector dimension $d_{s2\_2} = 3$ to obtain the vector $embed_{s2\_2}$.

Following the above steps, we obtained four sets of vectors computed through different modules. By concatenating $embed_{s2\_1}$, $embed_{s2\_2}$ and $embed_{d2}$ along the first dimension, we derived a new vector $embed_2$. In the DeepFM model, the computation module is primarily divided into two components: FM and DNN.

In the FM module, we first perform a statistical summation on $embed_{s1}$ (obtained by mapping sparse features through the embedding layer) and $embed_{d1}$ (obtained by mapping dense features through the fully connected layer) to compute the first-order summation result $score1$:

$$linear_{part} = \sum_{i=1}^{d} v_i$$

In the second-order interaction module of the FM, we first compute the square of the sum of the newly concatenated vector $embed_2$ to capture the interactions between all feature pairs. Then, we compute the sum of the squares of $embed_2$ to eliminate the influence of interactions between features and themselves. By calculating the difference between these two results and multiplying by 0.5, we obtain the second-order interaction result $score2$:

$$interaction_{part} = 0.5\left(\left(\sum_{i=1}^{d} v_i\right)^2 - \sum_{i=1}^{d} v_i^2\right)$$

In the DNN module, the input vector is also the newly concatenated vector $embed_2$. We designed a four-layer perceptron model, which consists of four fully connected layers, three activation function layers, and three dropout layers. The dimensionality scaling between the fully connected layers is set to [128, 64, 32, 1]. The ReLU activation function is employed to accelerate computations and reduce the computational burden. Dropout layers with parameters [0.4, 0.3, 0.2] are used to mitigate overfitting. Finally, the DNN module computes the result $score3$.

We concatenate the three scores ($score1$, $score2$, $and\ score3$) obtained from the above modules to form the score list $score\_all$. A fully connected layer is initialized to scale the vector $score\_all$ to a length of 1, and the final recommendation score $score\_final$ is obtained by applying a sigmoid activation function.

To optimize the recommendation task, we employ the binary cross-entropy loss function, which is widely adopted for binary classification problems in recommendation systems. Given the implicit feedback nature of the dataset, the training objective is to distinguish between positive samples (actual outfit combinations) and negative samples (randomly sampled mismatched pairs). The loss function is formulated as:

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N}[y_i log(\hat{y}_i) + (1-y_i)log(1-\hat{y}_i)] + \lambda \parallel \Theta \parallel^2$$

where $y_i \in \{0, 1\}$ denotes the ground-truth label of the i-th sample, $\hat{y}_i \in [0, 1]$ represents the predicted probability generated by the sigmoid-activated final score $score\_final$, and N is the total number of training samples. An L_2-regularization term $\lambda \parallel \Theta \parallel^2$ is incorporated to mitigate overfitting, with λ controlling the regularization strength and Θ denoting all trainable parameters in the model.

4. Results and Discussions

In this section, all experiments conducted using the proposed architecture will be presented and thoroughly discussed. Both qualitative and quantitative results will be evaluated, accompanied by detailed analysis and commentary.

4.1. Experimental Settings

4.1.1. Dataset

Currently, most existing datasets for clothing outfit composition are constructed by collecting data from outfit-sharing websites, such as FashionVC, ExpFashion, Polyvore Dataset, and IQON3000. The specific research problem addressed in this study involves utilizing users' historical behavior and garment information for top–bottom outfit matching and recommendation. However, the FashionVC dataset is too small in scale, the ExpFashion dataset primarily contains user review texts about complete outfits rather than individual items, and the Polyvore Dataset contains an excessive number of items per outfit composition. Furthermore, none of these three datasets provide user information, which does not align with the requirements of this research. To better analyze the performance of clothing outfit matching models, this study ultimately selects the IQON3000 dataset.

This study utilizes the IQON3000 dataset, which was collected by Song et al. from the popular fashion website IQON [9]. On this platform, users upload images of their outfits, which can be liked and commented on by others, creating an interactive mechanism based on user behavior. This provides rich information for analyzing user preferences in the dataset. The IQON3000 dataset comprises 308,747 outfit records uploaded by 3568 users, involving 672,335 fashion items [9], as shown in Table 1.

Table 1. Statistics of dataset IQON3000.

Item Category	Quantity
Outwear	35,765
Tops	119,895
Bottom	77,813
Dresses	25,816
Shoes	106,598
Accessories	306,448

Each outfit not only includes tops and bottoms but also accessories, footwear, and other components, comprehensively showcasing the diversity and complexity of clothing combinations. Most importantly, the IQON3000 dataset provides price information for

individual fashion items, offering a data foundation for training the model in terms of price sensitivity. The composition of an outfit in the dataset is illustrated in Figure 3.

Figure 3. One outfit in the IQON3000 dataset.

In this study, we focus exclusively on the combination of tops and bottoms. Therefore, based on the category labels in the dataset, we filtered out items that belong to the top and bottom categories, discarding data from other unused categories, and retained the data as the original clothing dataset. Additionally, to facilitate the data mining of price-related information, we preserved the brand, color, price, and category labels of the clothing items in the dataset. To extract textual and visual features of the clothing, we also retained the image information and product descriptions provided in the dataset.

The proposed clothing outfit recommendation model employs a training approach based on implicit feedback. We constructed triplets for training purposes. The construction process is illustrated in Figure 4:

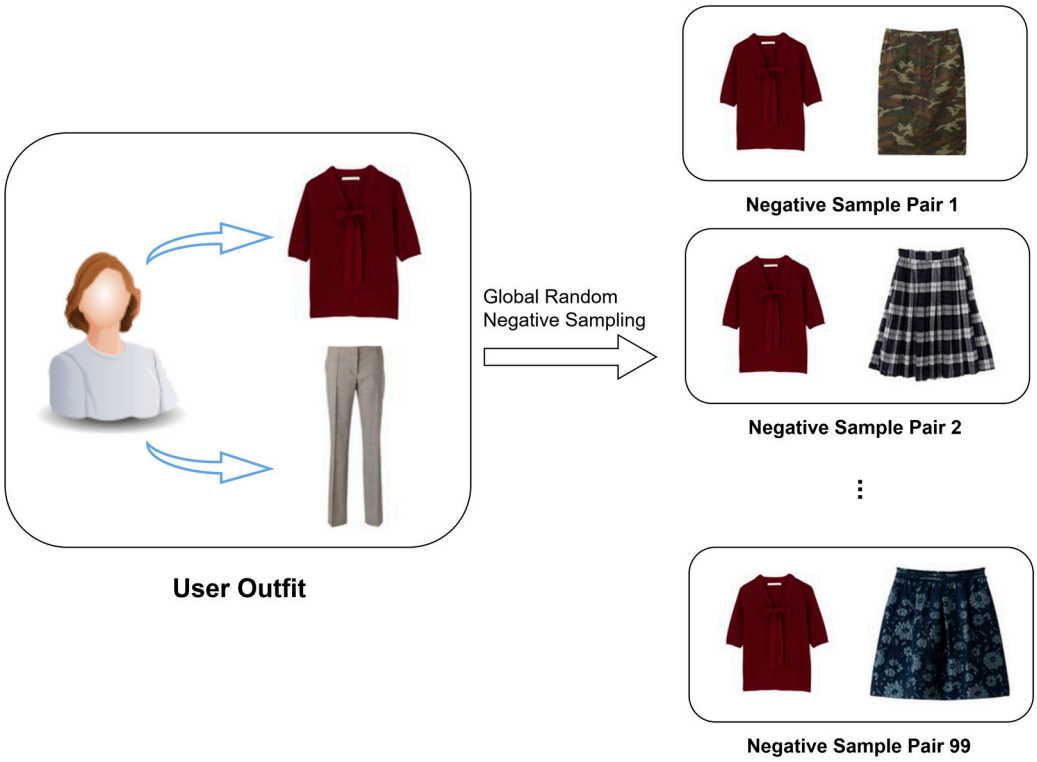

Figure 4. Construction of negative samples for the training dataset.

In the dataset, each user possesses a set of top–bottom outfit combinations $O_u = \{(t_1, b_1), (t_2, b_2), \ldots, (t_m, b_n)\}$. By aggregating the outfit combinations of all users, we obtain a comprehensive set of tops $T = \{t_1, t_2, \ldots, t_M\}$ and a set of bottoms $B = \{b_1, b_2, \ldots, b_n\}$.

For each user u's actual outfit combination (t_m, b_n), we perform global random negative sampling for both tops and bottoms. Specifically, we randomly select negative samples t_j and b_k from the complete sets T and B, respectively. These negative samples

are then paired with the original positive samples to generate two negative sample pairs: $\{(t_m, b_k), (t_j, b_n)\}$. The negative sample pairs are labeled as 0, while the positive sample pairs are labeled as 1, thereby creating the dataset required for training [39]. Subsequently, the dataset is randomly partitioned, with 80% allocated as the training set, 10% as the validation set, and 10% as the test set.

For each user u's actual outfit combination (t_m, b_n), we perform global random negative sampling on the bottoms. Specifically, we sample negative samples b_{k_n} from the set of bottoms B that the user has not historically selected. These negative samples are then paired with the original positive sample to generate a set of 99 negative sample pairs $\{(t_m, b_{k_1}), (t_m, b_{k_2}), \ldots, (t_m, b_{k_{99}})\}$. The negative sample pairs are labeled as 0, while the positive sample pairs are labeled as 1, thereby constructing the dataset required for training. Additionally, based on the chronological order of each user's outfit combinations, we retain users with more than 20 outfit combinations and truncate the dataset after the last 8 combinations for training purposes. Among the truncated last eight combinations, the first two are used as the validation set, and the last two are used as the test set.

4.1.2. Evaluation Metrics

In evaluating clothing recommendation systems, selecting appropriate metrics is essential for a comprehensive performance assessment. This study employs NDCG@5, NDCG@10, Recall@5, Recall@10, and AUC (Area Under the Curve) to evaluate the model's performance from multiple perspectives.

NDCG (Normalized Discounted Cumulative Gain) measures ranking quality by assessing the relevance and position of recommended clothing outfits [40]. The formula is as follows:

$$DCG = \sum_{i=1}^{k} \frac{rel_i}{log_2(i+1)}$$

$$IDCG = \sum_{i=1}^{k} \frac{rel_i^{ideal}}{log_2(i+1)}$$

$$nDCG = \frac{DCG}{IDCG}$$

Recall assesses coverage by measuring the proportion of user-preferred items correctly recommended [41]. The formula is as follows:

$$recall = \frac{TP}{TP + FN}$$

AUC evaluates classification performance by distinguishing between preferred and non-preferred items [42]. The AUC formula is as follows:

$$AUC = \frac{\sum_{i=1}^{N_+} \sum_{j=1}^{N_-} \mathbb{I}(p_i > p_j)}{N_+ \cdot N_-}$$

By integrating these metrics, this study provides a comprehensive evaluation of ranking quality, coverage, and classification performance, supporting the development of personalized clothing recommendation systems [43].

4.2. Result Analysis and Performance Comparison

After completing the feature extraction of the original data and constructing the clothing recommendation model, the extracted textual, visual, and price features were integrated and fed into the designed model. This section validates the effectiveness of our proposed recommendation model through comparative experiments with different models.

4.2.1. Baseline

LR [44]: This model is a linear classifier that transforms weighted feature combinations into probabilities through a sigmoid function. It relies on feature engineering to extract meaningful information and possesses the advantages of high efficiency and strong interpretability, making it a fundamental baseline model.

FM [6]: This model captures second-order feature interactions through latent vector inner products, effectively addressing feature combination problems in high-dimensional sparse data. It demonstrates computational efficiency and strong scalability.

Wide&Deep [7]: This model combines the memorization capability of linear models with the generalization advantages of deep networks. The wide component captures explicit feature interactions, while the deep component explores implicit correlations, jointly improving CTR prediction performance.

GP-BPR [45]: This model integrates Gaussian processes with Bayesian personalized ranking to probabilistically model users' implicit preferences and optimize item ranking.

DeepFM [8]: This model unifies factorization machines and deep neural networks to jointly capture both low-order feature interactions and high-order implicit patterns. Its end-to-end training eliminates the need for manual feature engineering, significantly improving recommendation accuracy and generalization capability.

4.2.2. Comprehensive Performance of Comparative Experiments

In this section, the proposed DeepFMP model is compared with several classical models using the extracted textual and visual features as the data foundation first. Then, to evaluate the impact of price features, the price feature vectors generated by the designed price feature module are incorporated. The overall performance and experimental results of all models on the constructed training set are presented in Table 2. For a comprehensive comparison, representative models for first-order, second-order, and high-order feature interactions are included.

Table 2. Comparative experimental results of various recommendation models.

Dataset	Model	AUC	Recall@5	Recall@10	NDCG@5	NDCG@10
Image, text	LR	0.680	0.231	0.267	0.298	0.419
	FM	0.759	0.415	0.516	0.388	0.483
	Wide&Deep	0.763	0.461	0.549	0.405	0.501
	GP-BPR	0.779	0.486	0.586	0.438	0.530
	DeepFM	0.786	0.519	0.601	0.455	0.553
	DeepFMP	0.802	0.536	0.622	0.476	0.585
Image, text and price	LR	0.709	0.239	0.278	0.311	0.436
	FM	0.786	0.432	0.538	0.405	0.503
	Wide&Deep	0.794	0.479	0.572	0.422	0.526
	GP-BPR	0.811	0.506	0.610	0.456	0.552
	DeepFM	0.816	0.545	0.635	0.483	0.596
	DeepFMP	0.833	0.558	0.648	0.495	0.609

The experimental results presented in the table demonstrate that the DeepFMP model outperforms other baseline models across all evaluation metrics when utilizing only textual and visual feature vectors. Specifically, the AUC value of DeepFMP shows an improvement ranging from 1.6% to 12.2% compared to the baseline models, with the most significant enhancement observed in the NDCG@10 metric, where it achieves a 3.2% improvement over the second-best model, DeepFM. These results indicate that the DeepFM model, enhanced with a multi-head self-attention mechanism, effectively improves the quality and accuracy of recommendation outcomes. Specifically, the DeepFMP model introduces a

multi-head self-attention mechanism with weight sharing, which enhances the interaction quality between input textual and visual feature vectors and improves the model's ability to uncover latent factors within the feature space.

Furthermore, it is evident that traditional models with simple interactions, such as LR, significantly lag behind other models in performance. In contrast, models incorporating complex high-order interactions consistently outperform traditional models with simple interactions across all evaluation metrics, particularly in the Recall metric, where the performance gap is nearly 30%. This demonstrates that second-order or high-order interactions effectively enhance the predictive performance of recommendation models, validating that models with high-order interactions generally outperform those with low-order interactions. This aligns with the observation that high-order interactions optimize the robustness and accuracy of the model.

After incorporating price features, the experimental results show that nearly all models exhibit improvements across various metrics. These findings suggest that the designed price feature extraction module significantly enhances the personalization and accuracy of clothing recommendations.

Figure 5 illustrates the AUC curve of the validation set after incorporating the feature extraction module, while Figure 6 presents the loss curve of the validation set. It can be observed that the proposed DeepFMP model exhibits relatively faster convergence and a smoother decline during the training process.

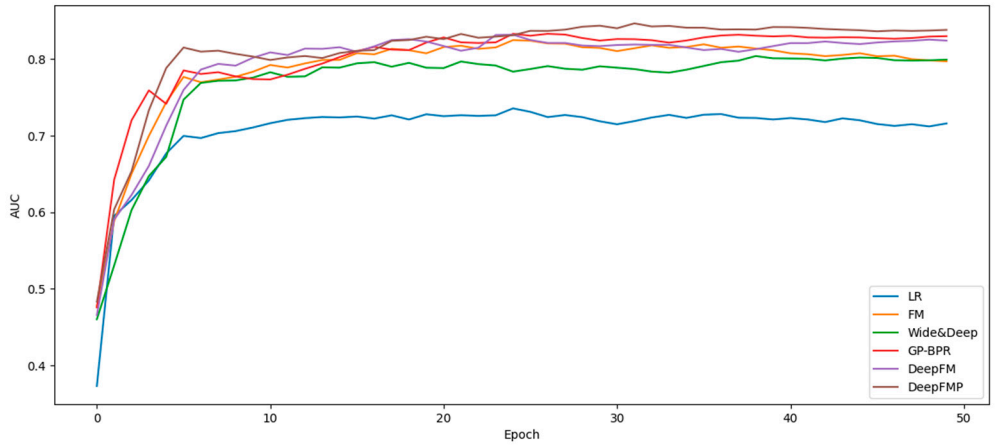

Figure 5. AUC of proposed model.

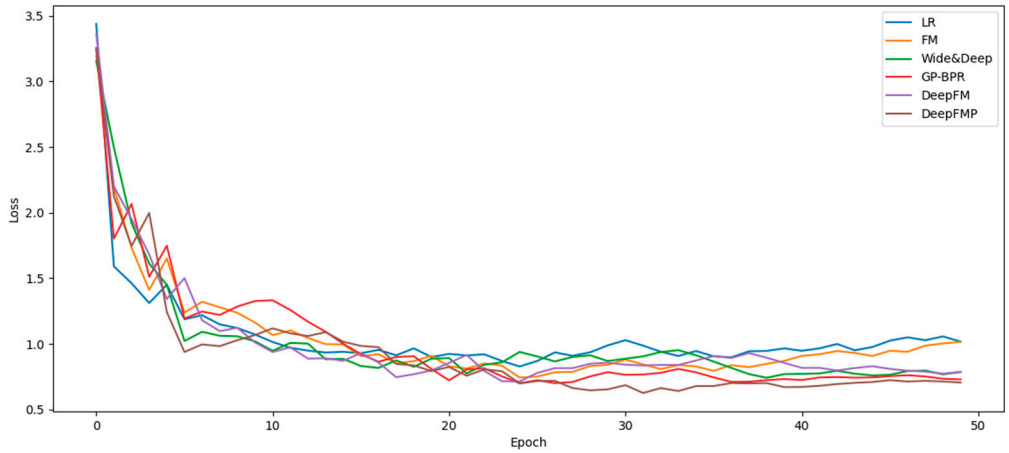

Figure 6. Loss of proposed model.

4.2.3. Comparative Test Case Study

To facilitate a clearer comprehension of the tasks involved in this study, enable a more intuitive analysis of the final recommendation outcomes of our model, and compare the performance of our model with that of others, this section presents the top 10 successfully retrieved samples randomly selected from the test set, displaying the recommended list of apparel images. The comparative Top-10 recommendation results of the DeepFMP model with the added price feature module are illustrated in Figure 7, where the garment highlighted by a red box represents the actual matching sample chosen by the user for the respective top.

Figure 7. Comparative experiment: visualization of ranking results.

As illustrated in Figure 7, for the same upper garment query, the proposed model ranks positive sample lower garments in higher positions within the recommendation sequence. The red bounding box in the figure highlights the positive sample lower garment. It can be observed that the price ranges of the tops and bottoms actually chosen by users are relatively close, with the total cost remaining within 1000. In the recommendation list generated by the model incorporating the price feature module, the prices of the recommended items are quite similar, aligning well with the user's current consumption habits. In contrast, the Top-10 recommendation list obtained after removing the price module from the model, although not significantly different in terms of garment appearance, exhibits a

considerable fluctuation in price range. Some individual items far exceed the price range of the actual matching, potentially surpassing the user's actual price tolerance range. This demonstrates the effectiveness of the proposed price feature extraction module in the actual recommendation results and its significant impact on the personalization and accuracy of the recommendations.

Following the integration of the price feature module, a comparison of the Top-10 recommendation results between DeepFMP and GP-BPR is illustrated in Figure 8. As shown in Figure 8, for the same upper garment query, the proposed model ranks positive sample lower garments higher in the recommendation sequence. The red box highlights the positive sample. It can be observed that, from the perspective of aesthetic coordination, the recommendation outcomes of both models are relatively aligned with the general public's aesthetic preferences, presenting reasonably coordinated ensembles. Notably, the Top-5 recommendation list generated by the proposed DeepFMP model successfully includes the actual matching samples chosen by users, with positive samples positioned more prominently in the recommendation list. This indicates that our model exhibits higher accuracy in apparel recommendation. Furthermore, from the standpoint of price features, the proposed model demonstrates a superior capability in uncovering latent factors related to pricing. Additionally, it achieves commendable recommendation efficacy for high-priced long-tail items, showcasing robust performance.

Figure 8. Price module comparative experiments: visualization of ranking results.

4.2.4. Comprehensive Performance of Ablation Experiments

In this study, we propose several optimization strategies for the multimodal fusion recommendation model based on consumer preferences. To verify the impact of different modules and features, as well as to evaluate the effectiveness of the model architecture design, we conduct ablation experiments in this section by progressively removing various components of the model. The variations in different metrics enable us to quantify their influence on the recommendation system's performance. The experimental results are presented in Table 3.

Table 3. Ablation experimental results of various recommendation models.

Dataset	Model	AUC	Recall@5	Recall@10	NDCG@5	NDCG@10
Image, text	Base	0.786	0.519	0.601	0.455	0.553
Image, text	Base + attention	0.802	0.536	0.622	0.476	0.585
Image, text, price	Base	0.816	0.545	0.635	0.483	0.596
Image, text, price	Base + attention	0.833	0.558	0.648	0.495	0.609

The experimental results demonstrate that price preference features and the multi-head self-attention mechanism significantly contribute to model performance improvement. When price features were excluded, the introduction of the multi-head attention mechanism increased the AUC from 0.786 to 0.802, with 3.2% improvement in NDCG@10, indicating that the attention mechanism enhances cross-modal feature interactions through dynamic allocation of modality weights. Upon incorporating price features, the base model's AUC improved from 0.786 to 0.816, representing 3.0% increase, while NDCG@10 showed 4.3% improvement, confirming the effectiveness of price features in capturing users' explicit preferences and mitigating semantic and demand biases. Notably, when both price features and the multi-head attention mechanism were simultaneously introduced, the model achieved optimal performance, with AUC and NDCG@10 improving by 4.7% and 5.6%, respectively, compared to the base model using only visual and textual features. This synergistic effect suggests that the multi-head self-attention mechanism and price preference feature mining exhibit complementarity in optimizing the rationality of ranking results for clothing outfit recommendations. These findings also validate the effectiveness of the proposed optimization strategies in this study.

5. Conclusions

This study focuses on the recommendation of fashion outfit combinations, specifically the pairing of tops and bottoms, by conducting in-depth data mining on price features and integrating image and textual information. We designed and implemented a multimodal deep learning recommendation model that incorporates price features, significantly enhancing the quality of outfit recommendations. The innovation of this study is reflected in two main aspects: first, by deeply exploring the impact of price features, especially when cross-referenced with other features, on user decision-making, and integrating these new features into the recommendation model; second, by combining the multi-head attention mechanism, the visual features, text features and price module features are effectively combined to combine the three modal features. The comparative experiments demonstrate that the DeepFMP model proposed in this study, along with the price feature extraction module, exhibit improvements across various evaluation metrics. This indicates that the two optimization points introduced in this paper are effective in the field of clothing outfit recommendation and provide a novel approach for personalized recommendation through multimodal fusion.

Building on this, we summarized the main research contributions and findings, analyzed the practical application value, and proposed future research directions. The experimental results demonstrate significant performance improvements, not only validating the importance of price features but also providing insights for real-world applications. Testing on real user outfit data shows that the proposed multimodal recommendation model better aligns with user preferences. Particularly in price-sensitive scenarios, the model's recommendations more closely match consumer expectations. Furthermore, by analyzing the top-10 recommendations from the full dataset on randomly sampled test data, we observed a notable improvement in price alignment, which aligns with user consumption habits and research expectations. This can help businesses optimize pricing strategies and more accurately target their customer base.

In future work, we plan to search for more diverse or larger datasets that can provide us with other dimensions of fashion clothing, such as fabric, material, color, and style, as well as factors like consumer age range, regional differences, cultural background, and social influences, which are also critical in outfit selection. These factors can be further mined and incorporated into feature design to enhance the personalization and accuracy of the model's recommendations. Additionally, clothing categories can be further subdivided. In this study, some categories were merged into two labels: tops and bottoms. While the recommendation results for popular top and bottom categories showed high accuracy and rationality, the results for niche categories or special styles need improvement. To address this, we aim to further refine clothing categories and design tailored feature extraction schemes to adapt to different clothing types, thereby improving the generalization capability of the recommendation system. Finally, for the pre-trained models used in the feature extraction module, with the development of deep learning and breakthroughs in hardware computing power, you can try to use more advanced pre-trained models [46] (such as Vit [47], Llama, etc.), or align different modal data using multimodal pre-trained models such as the clip model, so as to improve the accuracy and personalization of recommendations.

Author Contributions: Conceptualization, C.Z. and X.J.; Methodology, C.Z., X.J. and L.C.; Software, C.Z.; Validation, C.Z.; Formal analysis, C.Z.; Data curation, C.Z.; Writing—original draft, C.Z. and L.C.; Writing—review & editing, C.Z., X.J. and L.C.; Visualization, C.Z.; Supervision, X.J. and L.C.; Project administration, X.J. and L.C.; Funding acquisition, X.J. and L.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Philosophy and Social Science Planning Projects of Zhejiang Province (24NDJC170YB), Major Humanities and Social Sciences Research Projects in Zhejiang Universities (Grant No. 2024QN131).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available in IQON3000 at https://drive.google.com/file/d/1sTfUoNPid9zG_MgV--lWZTBP1XZpmcK8/view?usp=sharin (accessed on 28 February 2025), reference number [9].

Acknowledgments: The authors acknowledge and appreciate the support received from the Zhejiang Sci-Tech University.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Jo, J.; Lee, S.; Lee, C.; Lee, D.; Lim, H. Development of fashion product retrieval and recommendations model based on deep learning. *Electronics* **2020**, *9*, 508. [CrossRef]

2. Duan, S.; Ouyang, M.; Wang, R.; Li, Q.; Xiao, Y. Let long-term interests talk: An disentangled learning model for recommendation based on short-term interests generation. *Inf. Process. Manag.* **2025**, *62*, 103997. [CrossRef]
3. Lops, P.; Jannach, D.; Musto, C.; Bogers, T.; Koolen, M. Trends in content-based recommendation: Preface to the special issue on Recommender systems based on rich item descriptions. *User Model. User-Adapt. Interact.* **2019**, *29*, 239–249. [CrossRef]
4. Köhler, S.; Wöhner, T.; Peters, R. The impact of consumer preferences on the accuracy of collaborative filtering recommender systems. *Electron. Mark.* **2016**, *26*, 369–379. [CrossRef]
5. Yin, R. *Enhanced Recommender Systems with Deep Neural Networks*; University of Technology Sydney: Sydney, NSW, Australia, 2022.
6. Rendle, S. Factorization machines. In Proceedings of the 2010 IEEE International Conference on Data Mining, Sydney, NSW, Australia, 13–17 December 2010; pp. 995–1000.
7. Cheng, H.-T.; Koc, L.; Harmsen, J.; Shaked, T.; Chandra, T.; Aradhye, H.; Anderson, G.; Corrado, G.; Chai, W.; Ispir, M. Wide & deep learning for recommender systems. In Proceedings of the 1st Workshop on Deep Learning for Recommender Systems, Boston, MA, USA, 15 September 2016; pp. 7–10.
8. Guo, H.; Tang, R.; Ye, Y.; Li, Z.; He, X. DeepFM: A factorization-machine based neural network for CTR prediction. *arXiv* **2017**, arXiv:1703.04247.
9. Song, X.; Han, X.; Li, Y.; Chen, J.; Xu, X.-S.; Nie, L. GP-BPR: Personalized compatibility modeling for clothing matching. In Proceedings of the 27th ACM International Conference on Multimedia, Nice, France, 21–25 October 2019; pp. 320–328.
10. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
11. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Adam, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv* **2017**, arXiv:1704.04861.
12. Miao, Y.; Li, G.; Bao, C. ClothingNet: Cross-domain clothing retrieval with feature fusion and quadruplet loss. *IEEE Access* **2020**, *8*, 142669–142679. [CrossRef]
13. Xuan, Y.; Liao, X.; Su, X. Clothing Image Retrieval Method Based on Convolutional Block Attention Model. *J. Comput. Sci. Appl.* **2022**, *12*, 1331–1340.
14. Mu, C.; Guo, Z.; Liu, Y. A multi-scale and multi-level spectral-spatial feature fusion network for hyperspectral image classification. *Remote Sens.* **2020**, *12*, 125. [CrossRef]
15. Guo, M.-H.; Xu, T.-X.; Liu, J.-J.; Liu, Z.-N.; Jiang, P.-T.; Mu, T.-J.; Zhang, S.-H.; Martin, R.R.; Cheng, M.-M.; Hu, S.-M. Attention mechanisms in computer vision: A survey. *Comput. Vis. Media* **2022**, *8*, 331–368. [CrossRef]
16. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient estimation of word representations in vector space. *arXiv* **2013**, arXiv:1301.3781.
17. Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Minneapolis, MN, USA, 2–7 June 2019; Volume 1, pp. 4171–4186.
18. Gao, D.; Jin, L.; Chen, B.; Qiu, M.; Li, P.; Wei, Y.; Hu, Y.; Wang, H. FashionBERT: Text and Image Matching with Adaptive Loss for Cross-modal Retrieval. In Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, Xi'an, China, 25–30 July 2020; pp. 2251–2260.
19. Li, Y.; Chen, T.; Huang, Z. Attribute-aware explainable complementary clothing recommendation. *World Wide Web* **2021**, *24*, 1885–1901. [CrossRef]
20. Zheng, Y.; Gao, C.; He, X.; Li, Y.; Jin, D. Price-aware recommendation with graph convolutional networks. In Proceedings of the 2020 IEEE 36th International Conference on Data Engineering (ICDE), Dallas, TX, USA, 20–24 April 2020; pp. 133–144.
21. Chen, J.; Jin, Q.; Zhao, S.; Bao, S.; Zhang, L.; Su, Z.; Yu, Y. Does product recommendation meet its Waterloo in unexplored categories? No, price comes to help. In Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval, Gold Coast, QLD, Australia, 3 July 2014; pp. 667–676.
22. Wang, J.; Zhang, Y. Utilizing marginal net utility for recommendation in e-commerce. In Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval, Beijing, China, 25–29 July 2011; pp. 1003–1012.
23. Lin, H.; Zhu, H.; Wu, J.; Zuo, Y.; Zhu, C.; Xiong, H. Enhancing employer brand evaluation with collaborative topic regression models. *ACM Trans. Inf. Syst. (TOIS)* **2020**, *38*, 1–33. [CrossRef]
24. Liu, H.; Li, L.; Yu, N.; Ma, K.; Peng, T.; Hu, X. Outfit compatibility model using fully connected self-adjusting graph neural network. *Vis. Comput.* **2024**, *40*, 8331–8343. [CrossRef]
25. Zafar, S.; Kumar, S.; Ahilan, A.; Cakir, G.K. *Industry 5.0 for Smart Healthcare Technologies: Utilizing Artificial Intelligence, Internet of Medical Things and Blockchain*; CRC Press: Boca Raton, FL, USA, 2024.
26. Ugail, H. *Deep Learning in Visual Computing: Explanations and Examples*; CRC Press: Boca Raton, FL, USA, 2022.
27. Su, J.; Her, P.; Clemens, E.; Yaz, E.; Schneider, S.; Medeiros, H. Violence detection using 3d convolutional neural networks. In Proceedings of the 2022 18th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Madrid, Spain, 24 November 2022; pp. 1–8.

28. Conia, S.; Li, M.; Lee, D.; Minhas, U.F.; Ilyas, I.; Li, Y. Increasing coverage and precision of textual information in multilingual knowledge graphs. *arXiv* **2023**, arXiv:2311.15781.
29. Branavan, S.; Silver, D.; Barzilay, R. Learning to win by reading manuals in a monte-carlo framework. *J. Artif. Intell. Res.* **2012**, *43*, 661–704. [CrossRef]
30. Peyton, K.; Unnikrishnan, S.; Mulligan, B. A review of university chatbots for student support: FAQs and beyond. *Discov. Educ.* **2025**, *4*, 21. [CrossRef]
31. Zhang, S.; Tay, Y.; Yao, L.; Sun, A.; Zhang, C. Deep learning for recommender systems. In *Recommender Systems Handbook*; Springer: Berlin/Heidelberg, Germany, 2021; pp. 173–210.
32. Gao, S.; Hu, Y.; Li, W. *Handbook of Geospatial Artificial Intelligence*; CRC Press: Boca Raton, FL, USA, 2023.
33. Singh, A. *Impacts of Adversarial Machine Learning Methods in Deep Learning Models Used in Iot Environments*; National University of Singapore: Singapore, 2023.
34. Hui, B.; Zhang, L.; Zhou, X.; Wen, X.; Nian, Y. Personalized recommendation system based on knowledge embedding and historical behavior. *Appl. Intell.* **2021**, *52*, 954–966. [CrossRef]
35. Wang, M.; Shi, X. Research on User Behavior Analysis in E-commerce Platforms Based on Personalized Recommendation Algorithms. In Proceedings of the International Conference on Decision Science & Management, Hong Kong, China, 26–28 April 2024; pp. 140–145.
36. Park, G.; Han, C.; Yoon, W.; Kim, D. MHSAN: Multi-head self-attention network for visual semantic embedding. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Snowmass Village, CO, USA, 1–5 March 2020; pp. 1518–1526.
37. Wang, X.; Li, Q.; Yu, D.; Huang, W.; Li, Q.; Xu, G. Neural causal graph collaborative filtering. *Inf. Sci.* **2024**, *677*, 120872. [CrossRef]
38. Fu, J.; Fu, Y.; Xue, H.; Xu, Z. TMFN: A text-based multimodal fusion network with multi-scale feature extraction and unsupervised contrastive learning for multimodal sentiment analysis. *Complex Intell. Syst.* **2025**, *11*, 133. [CrossRef]
39. Xie, Y.; Lin, J.; Dong, H.; Zhang, L.; Wu, Z. Survey of code search based on deep learning. *ACM Trans. Softw. Eng. Methodol.* **2023**, *33*, 1–42. [CrossRef]
40. Järvelin, K.; Kekäläinen, J. Cumulated gain-based evaluation of IR techniques. *ACM Trans. Inf. Syst. (TOIS)* **2002**, *20*, 422–446. [CrossRef]
41. Shani, G.; Gunawardana, A. Evaluating recommendation systems. In *Recommender Systems Handbook*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 257–297.
42. Bradley, A.P. The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognit.* **1997**, *30*, 1145–1159. [CrossRef]
43. Koren, Y.; Bell, R.; Volinsky, C. Matrix factorization techniques for recommender systems. *Computer* **2009**, *42*, 30–37. [CrossRef]
44. Cox, D.R. The regression analysis of binary sequences. *J. R. Stat. Soc. Ser. B Stat. Methodol.* **1958**, *20*, 215–232. [CrossRef]
45. Radford, A.; Kim, J.W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J. Learning transferable visual models from natural language supervision. In Proceedings of the International Conference on Machine Learning, Virtually, 18–24 July 2021; pp. 8748–8763.
46. Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; Stoyanov, V. Roberta: A robustly optimized bert pretraining approach. *arXiv* **2019**, arXiv:1907.11692.
47. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Houlsby, N. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv* **2020**, arXiv:2010.11929.