Personalised Outfit Recommendation via History-aware Transformers

Myong Chol Jung*
Amazon Machine Learning
Melbourne, Australia
davidmcjung@gmail.com

Julien Monteil
Amazon Machine Learning
Brisbane, Australia
jul@amazon.com

Philip Schulz
Amazon Machine Learning
Sydney, Australia
phschulz@amazon.com

Volodymyr Vaskovych
Amazon Machine Learning
Melbourne, Australia
vaskovyc@amazon.com

Abstract

We present the history-aware transformer (HAT), a transformer-based model that uses shoppers' purchase history to personalise outfit predictions. The aim of this work is to recommend outfits that are internally coherent while matching an individual shopper's style and taste. To achieve this, we stack two transformer models, one that produces outfit representations and another one that processes the history of purchased outfits for a given shopper. We use these models to score an outfit's compatibility in the context of a shopper's preferences as inferred from their previous purchases. During training, the model learns to discriminate between purchased and random outfits using 3 losses: the focal loss for outfit compatibility typically used in the literature, a contrastive loss to bring closer learned outfit embeddings from a shopper's history, and an adaptive margin loss to facilitate learning from weak negatives. Together, these losses enable the model to make personalised recommendations based on a shopper's purchase history.

Our experiments on the IQON3000 and Polyvore datasets show that HAT outperforms strong baselines on the outfit Compatibility Prediction (CP) and the Fill In The Blank (FITB) tasks. The model improves AUC for the CP hard task by 15.7% (IQON3000) and 19.4% (Polyvore) compared to previous SOTA results. It further improves accuracy on the FITB hard task by 6.5% and 9.7%, respectively. We provide ablation studies on the personalisation, constrastive loss, and adaptive margin loss that highlight the importance of these modelling choices.

CCS Concepts

• **Computing methodologies** → **Learning latent representations**; *Search methodologies*; **Neural networks**; **Supervised learning**; • **Applied computing** → **Online shopping**.

*Work done while at Amazon.

WSDM '25, March 10–14, 2025, Hannover, Germany

ACM ISBN 979-8-4007-1329-3/25/03
https://doi.org/10.1145/3701551.3703545

Keywords

Personalisation, Recommendation, Transformer, Constrastive Learning

ACM Reference Format:
Myong Chol Jung, Julien Monteil, Philip Schulz, and Volodymyr Vaskovych. 2025. Personalised Outfit Recommendation via History-aware Transformers. In *Proceedings of the Eighteenth ACM International Conference on Web Search and Data Mining (WSDM '25), March 10–14, 2025, Hannover, Germany.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3701551.3703545

1 Introduction

Fashion shoppers want stylistically matching outfits that look good. In a brick-and-mortar store, a sales assistant can help coordinating an outfit. In an online setting, shoppers rely on outfit recommendations provided on a website. To showcase relevant outfits from millions of individual fashion items, e-commerce retailers need to automate their recommendations. Consequently, outfit recommendation is a task that has been approached in several ways in the research literature, e.g., [12, 13, 17, 26, 28, 38]. However, these recommendations are generic and reflect outfit preferences aggregated across shoppers. To match the physical store experience, shoppers need personalised outfit recommendations that reflect their individual taste and style.

Existing studies like [13, 26] have addressed non-personliased outfit recommendation problems by optimising two distinct tasks which are: (1) the outfit Compatibility Prediction (CP) task that predicts whether items in an outfit are compatible or not, and (2) the Fill in the Blank (FITB) task that selects the most compatible item for an incomplete outfit given a set of candidate choices. However, these approaches do not relate the outfit to the preferences of the shopper who is looking at it. This can lead to recommendations that fit well for shoppers with mainstream taste but are sub-optimal for all other customers.

To overcome this limitation, we propose the history-aware transformer (HAT), a personalised stacked transformer model that incorporates a shopper's view or purchase history. By leveraging the transformer [34], HAT is invariant to the order of items of each outfit. In order to enable personalisation, we introduce a contrastive loss to bring the embeddings of outfits that have been bought by the same shopper close to each other. We further introduce weak negative outfits which are purchased or curated outfits in which single item has been randomly replaced. These weak negatives make the

discrimination task harder, meaning that bought-together outfits form tighter clusters in latent space. This form of data augmentation is inspired by the intuition that most shoppers are looking to buy an item that is compatible with their existing selection instead of buying a whole outfit.

Finally, we introduce an adaptive margin ranking loss that forces the compatibility score of a weak negative outfit to be lower than the compatibility score of a corresponding positive outfit by a certain margin. This margin should be larger if a stylistically important item is replaced in the weak negative and smaller if a less important item is switched. The importance of each item in an outfit is computed by a cross-attention mechanism with learnable queries, inspired by the Q-former [11]. The full HAT model is trained with positive, negative and weak negative labels (ratio 1:1:1) on a weighted combination of the focal loss, contrastive loss, and adaptive margin ranking loss.

In summary, our contributions are four-fold:

(1) We propose HAT, a stacked transformer architecture that jointly learns individual outfit embeddings while also summarising a shopper's purchase history.
(2) We enable personalisation based on purchase history by utilising a contrastive loss to outfit embeddings bought by the same shopper closer to each other.
(3) We explicitly model the fact that shoppers desire to buy items that are compatible with their existing selection, by introducing weak negatives, that consists of bought outfits with single item randomly replaced, and an adaptive margin ranking loss to adapt to the importance of the replaced items.
(4) We show that our approach outperforms competitive baselines on the IQON3000 and Polyvore datasets by 15.7% and 19.4% in terms of AUC on the outfit CP-hard task, and by 6.5% and 9.7% in terms of accuracy on FITB-hard task.

2 Related Work

2.1 Outfit Recommendation

2.1.1 Non-personalised outfits. Fashion recommenders is a subset of recommendation systems that involve a particular domain market: garments and fashion items. The most straightforward approach to obtain outfit-level representations is to use multi-instance pooling on individual item embeddings [12, 31]. While this is an easy and efficient technique, it cannot capture complex outfit-item interactions. To fix this shortcoming, sequence, graph, and attention models have been proposed. In sequence modelling [4, 6, 7, 12, 17], outfits are presented as an ordered sequence of items and variations of RNNs are used to obtain a final outfit representation. However, this approach makes strong assumptions on the item order while outfits are inherently unordered as conceded in [17].

As an improvement to sequence modelling, attention-based approaches have gained popularity in the recent years. Content Attentive Neural Networks (CANN) were proposed in [13], where attention blocks are used to find representations responsible for compositional coherence between global contents of outfit images and coherence from semantic-focal contents. Mixed Category Attention Net (MCAN) was introduced in [36] which leverages item category information in its attention networks to find better recommendation. OutfitTransformer [26] feeds an item's text and image embeddings into the transformer block together with a learnable

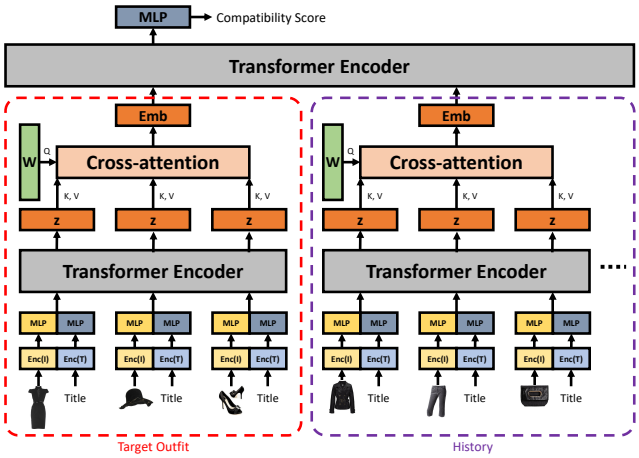

Figure 1: Model diagram of HAT. The target outfit is an outfit whose compatibility score we wish to compute. It consists of images and titles for each item. K, V, and Q represent key, value, and query respectively for the cross-attention. The bottom-level transformer encoder shares weights.

outfit token. The output of the transformer encoder serves as the global outfit representation. The outfit representation is fed into an MLP used to judge the compatibility of the outfit. While the Outfit-Transformer does not make personalised predictions, we leverage part of this architecture for the personalised outfit tasks (see Section 3.2).

2.1.2 Personalised outfits. A notable drawback of non-personalised outfit recommendation systems is that the shopper's individual preferences are not considered when recommendations are made. Thus, many studies have proposed additional components to the outfit recommendation systems to incorporate personal preference signals.

Similar to non-personalised recommendation, there were attempts [4, 19, 20, 37] to use pairwise item-item compatibility together with shopper-item compatibility. Lin et al. [15] proposed OutfitNet, a two-stage approach that first learns item compatibility of embeddings which then applies attention-based pooling to shopper and item embeddings to compute outfit relevancy scores. The model uses a triplet loss to maximise the difference between the relevancy scores of outfits that shoppers bought and ignored.

Similarly, Personalised Outfit Generation (POG) was proposed in [2] for tackling the outfit generation problem. They use transformer blocks on embeddings of all items in the outfit and a masked embedding for the missing item. The output embedding on the masked slot is used to find the best matching item out of the alternatives. Contrary to our work, personalised outfit generations is achieved by encoding the entire purchase history of a shopper as one outfit with no masks. This embedding is then used in the masked transformer decoder blocks to generate outfit items, one at a time.

2.2 Sequential Recommendation

While the focus of this paper is personalised outfit recommendation, the personalisation mechanism can be viewed as a form of sequential recommendation, an active field of recommendation research. Before reviewing the relvevant literature, let us emphasise that all works known to us focus on next-item prediction and do not address the problem of recommending outfits or, more generally, sets of products.

Many sequential approaches are based on Wide and Deep [3] in that they process the history of user-product interactions in separate sub-network whose output is later combined with output based on other features. Prominent sequential recommenders include Deep Interest Networks [39] and their successor, the Behavior Sequence Transformer [1]. In general, there has been vivid interest in using transformer models for sequential recommendation [8, 30]. The most similar work to our approach is SSE-PT [35] which uses a transformer to encode a user history and predict the last item in that history. Contrary to SSE-PT, our work focuses on scoring the consistency of outfits, does not use non-standard regulariasation schemes and does not rely on user embeddings, making it more likely to perform well on cold customers.

Finally, using pre- trained LLMs for sequential recommendation is alluring but has proven to be challenging [16]. However, recent work that combines purpose-trained transformers with LLMs is showing the feasibility of combining recommendation knowledge with the strong reasoning abilities of LLMs [21, 23]. It remains to be seen if the ideas presented in those works carry over to set/outfit recommendation.

2.3 Contrastive Learning

Constrastive learning was first used in [5] to estimate intractable distributions by turning the density estimation problem into a classification problem. Since this seminal work, contrastive learning has been further developed by e.g. [32] to extend to a setting where each positive sample is paired with multiple negative samples. This makes the classification problem harder and thus leads to better representation learning, and has since been successfully applied in several representation learning frameworks such as CLIP [24]. Supervised contrastive learning [9] allows for more than one positive sample by using class labels. Each item in a batch serves as an anchor; all other items with the same class label are treated as positive while all remaining samples are treated as negative. Supervised constrastive learning seeks to maximise the inner product between the vector representation of the anchor and all positive samples. It has the effect that normalised embeddings from the same class are pulled closer together than embeddings from different classes. In practice, negative samples are typically drawn from the same mini-batch instead of the entire dataset.

3 Methodology

In this section, we present the details of the proposed architecture for personalised outfit recommendation and the corresponding training procedure. We also present a data augmentation strategy that uses "weak negative" outfits in training (Sec 3.3.3). Following previous work [27], we represent an outfit through the titles and images of the individual items it contains. To add personalisation to our model, we represent a shopper's style through the outfits they have purchased in the past. Technically, we aim to predict the style consistency of the target outfit conditioned on the outfits in a shopper's interaction history.

3.1 Problem Formulation

We define the o^{th} outfit in a shopper's history as $S_o = \{x_i^o\}_{i=1}^{N^o} = \{I_i^o, T_i^o\}_{i=1}^{N^o}$ where x_i^o is an individual fashion item represented by an image I_i^o and a title T_i^o with N^o being the number of items in the outfit. We denote the u^{th} shopper's purchase history by $H_u = \{S_o\}_{o=1}^{M^u}$ where M^u is the number of outfits purchased by the shopper. Our task is to predict a compatibility score (of whether the items in an outfit match together) $p^o \in [0, 1]$ for a target outfit S_t. During training, the ground truth for the target is either $y^o = 1$ for an outfit judged as compatible (e.g. via annotation) or $y^o = 0$ for an incompatible outfit (e.g. a random outfit). The score p^0 represents the compatibility of an outfit as predicted by our model.

The different layers in our model architecture are denoted by lower case letters. We use greek subscripts to denote model parameters. For example, e_θ and e_ϕ are the image and title encoders.

3.2 History-aware Transformer (HAT)

We propose the History-aware Transformer (HAT) model to personalise outfit compatibility scoring, shown in Fig. 1. HAT is a two-level stack of transformer encoders [34]. The bottom-level transformer encoder generates an outfit embedding for the purpose of understanding the internal compatibility of that outfit. The top-level transformer encoder enables personalisation. It computes the compatibility score of the target outfit in the context of a shoppers's previously purchased outfits. By incorporating information about the purchased outfits as inputs to the top-level encoder, the model can learn not only whether a given outfit is compatible on its own but also if the given outfit matches well with the other outfits that the shopper has purchased. This allows the model to match the current outfit to the shopper's fashion style as inferred from their historical purchases.

3.2.1 Bottom-level Encoder. For a given outfit S_o, the bottom-level transformer encoder t_ψ processes item embeddings which are concatenations of an encoded image $e_\theta(I_i^o)$ and an encoded title $e_\phi(T_i^o)$. In practice, we use the pre-trained CLIP model [24] to embed both the images and titles. The outputs of the bottom-level transformer encoder, representing each item as z_i^o, are fed into a cross-attention layer. We introduce a learnable weight (W) as a query which is shared between all outfits. This query lets the model attend to important items in an outfit. Since W is learned, we make no prior assumptions about what item types (e.g. trousers vs. shirts) are most important for the compatibility of an outfit. We call the resulting outfit embedding E^o. Formally, we define the outfit embedding of the o^{th} outfit as:

$$E^o = A^o Z^o = \text{Softmax}\left(\frac{W Z^{oT}}{d}\right) Z^o \tag{1}$$

$$W \in \mathbb{R}^d, \quad Z^o \in \mathbb{R}^{N^o \times d}$$

$$Z^o = \left[z_1^o, \ldots, z_{N^o}^o\right] = t_\psi\left(\left[e_\theta(I_i^o); e_\phi(T_i^o)\right]\right]_{i=1}^{N_{item}^o}\Big) \tag{2}$$

where A^o is the cross-attention matrix, d is the latent dimension, $t_\psi(\cdot)$ is the bottom-level transformer encoder with learnable parameters ψ, and $[\cdot;\cdot]$ is concatenation along the feature dimension. The bottom-level transformer encoder t_ψ captures the relationship between the item embedding within an outfit. It is largely inspired by the OutfitTransformer [27]; the addition of the attention mechanism to weigh an item's importance within the outfit via a learnable query is novel. As we show in Tables 2 and 4, it improves upon the original non-personalised Outfit Transformer.

Conceptually, there are two advantages of introducing the learnable outfit representation using cross-attention. Firstly, the resulting outfit representation is a single vector independently of the number and the order of items. Secondly, the attention matrix approximates the importance of each item in an outfit (e.g., a top may have higher contribution to the compatibility of an outfit than an earring) without making prior assumptions. We show how to use the attention weights in the adaptive margin loss we introduce in Section 3.3.3.

3.2.2 Top-level Encoder. The inputs of the top-level transfomer encoder t_ρ are outfit representations $\{E^i\}_{i=1}^{M_u}$ from a user's purchase history H_u generated by the bottom-level encoder t_ψ. In particular, we concatenate a representation of the target outfit (i.e. the outfit whose compatibility we want to estimate) and the representations from outfits in a shopper's the purchase history. We then use an MLP f_λ to predict the compatibility score. We use the sigmoid function σ to bring the prediction onto the unit scale.

$$p^o = \sigma\left(f_\lambda\left(t_\rho\left([E_t; E_1, \ldots, E_{M_u}]\right)\right)\right) \tag{3}$$

By leveraging the shopper's purchase history, the model is able to estimate the compatibility of the target outfit S_t (whose outfit represenation is E_t) in the context of the shoppers's personal style. Notice that the target outfit is not included in the history; it can be varied to score different target candidates.

3.3 Model Training

In this section, we discuss how to train HAT for personalised outfit recommendation. The training procedure contains three losses which address different aspects of the personalised recommendation problem. We define all losses over a mini-batch B of size $|B| = K$.

3.3.1 Outfit Compatibility. We follow [27] in using the focal loss [14] for compatibility prediction:

$$\mathcal{L}_{FL}(B) = \sum_{k \in B} F(p^o); \text{ where}$$
$$F = \begin{cases} -\alpha(1-p^o)^\gamma \log(p^o), & y^o = 1 \\ -(1-\alpha)(p^o)^\gamma \log(1-p^o), & y^o = 0 \end{cases} \tag{4}$$

where α and γ are hyperparameters that balance classes and difficult samples respectively.

We limit the maximum number of history outfits since long histories are rare but take up a lot of GPU memory. If the total number of outfits purchased by a shopper exceeds the threshold, we randomly sample outfits. Refer to Section 4.2.4 for the impact of the number of history outfits on performance.

3.3.2 Personalisation by Contrastive Learning. In order to fully utilise the top-level transformer, we found that it is important to bring the embeddings of outfits that have been bought by the same shopper close to each other. This produces outfit representations E^o for which the compatibility score is dependent on the outfits in the history. To achieve this, we leverage supervised contrastive learning [10]. Each history outfit embedding in a mini-batch is an anchor, outfit embeddings from the same shopper are positives, and outfit embeddings from different shoppers are negatives. In other words, we leverage shoppers as labels for supervised contrastive learning.

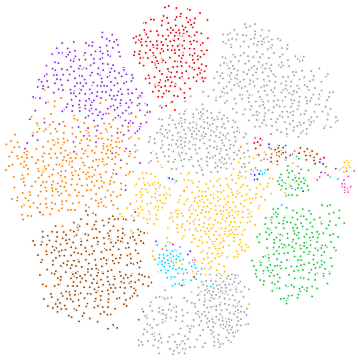

Figure 2: Exemplary outfit embeddings projected to 2D space by t-SNE [33] in IQON3000. Colours indicate different shoppers.

Let E^k and E^p be outfit representations within batch B. During training, a mini-batch contains purchase histories of various shoppers. Let $ID(k)$ be the shopper ID associated with the k^{th} outfit. We define the positive samples for the k^{th} outfit in the batch as $P(k) \equiv \{p \in B^{-k} : ID(p) = ID(k)\}$, i.e. those outfits from the same shopper's purchase history. We further define $B^{-k} = B \setminus \{k\}$ to be the batch without the k^{th} outfit. The supervised contrastive loss is defined as:

$$\mathcal{L}_{CL}(B) = \sum_{k \in B} \frac{-1}{|P(k)|} \sum_{p \in P(k)} \log\left(\frac{\exp(E^k \cdot E^p/\tau)}{\sum_{n \in B^{-k}} \exp(E^k \cdot E^n/\tau)}\right) \tag{5}$$

where τ is a temperature scale parameter. The effect of the supervised contrastive loss is that outfit representation from the same shopper's history are close to each other. This is illustrated in Fig. 2. Refer to Section 4.2.3 for an ablation study on the effectiveness of the supervised contrastive loss.

3.3.3 Adaptive Margin Loss. Previous studies such as [26] have created negative outfits by randomly switching all items of an outfit as shown in Fig. 3a and Fig. 3b to train outfit recommendation models. While this helps the model distinguish an annotated outfit and a random outfit, the signal is weak since random assortments of items are likely to be non-compatible and hence easy to distinguish from curated outfits. A more challenging task is to distinguish the positive outfit and a "weak negative" sample in which only one of items is randomly switched as shown in Fig. 3c. This is also closer to the choice that shoppers face when buying clothes. Most of the time, shoppers are looking to buy an item that is compatible with their existing wardrobe; a shopper may look for a pair of shoes that match well to a top and a bottom that the shopper already

(a) Positive outfit.

(b) Negative outfit.

(c) Weak negative outfit.

Figure 3: Examples of positive outfit (a), negative outfit (b), and weak negative outfit (c). Every item of the negative outfit is randomly selected item in place of the positive items within the same item category. On the other hand, a weak negative outfit only has a single item switched.

owns. Thus, we augment our training data with weak negative outfits while training the model to learn fine-grained details of a compatible outfit.

When using weak negative outfits for model training, one challenge is the lack of ground truth compatibility for these outfits. It would be misleading to label them as either 0 (fully incompatible) or 1 (fully compatible). To address this issue, we propose a margin ranking loss which requires the compatibility score of a weak negative sample to be lower than compatibility score of a corresponding positive sample by a certain margin. This margin ranking loss is defined as:

$$\mathcal{L}_M(B) = \sum_{k \in B} (0, -p_p^k + p_w^k + m) \tag{6}$$

where (x, y) denotes a maximum value between x and y, p_p^k is the compatibility score for a positive (original) version of outfit o^k and p_w^k is the compatibility score for a weak negative version of outfit o_i. We use m to denote the fixed margin.

Although Eq. (6) allows the model to learn the compatibility score of the weak negative sample, the margin is fixed for all weak negative samples. This neglects each items contribution to the overall compatibility of an outfit. For instance, the margin is fixed whether a jacket among three items or an earring among ten items is switched. Intuitively, the margin should be larger if a more important item is switched and smaller if a less important item is switched. We incorporate this intuition into the loss by using the learned attention weight in Eq. (1) to represents the importance of each item in computing the compatibility score at no additional computation cost. We define the **adaptive margin loss** as:

$$\mathcal{L}_{AM}(B) = \sum_{k \in B} (0, -p_p^k + p_w^k + m \cdot A_i^p) \tag{7}$$

where i is the index of the switched item in the weak negative sample and A_i^p is that item's attention weight. We demonstrate this in Fig. 4.

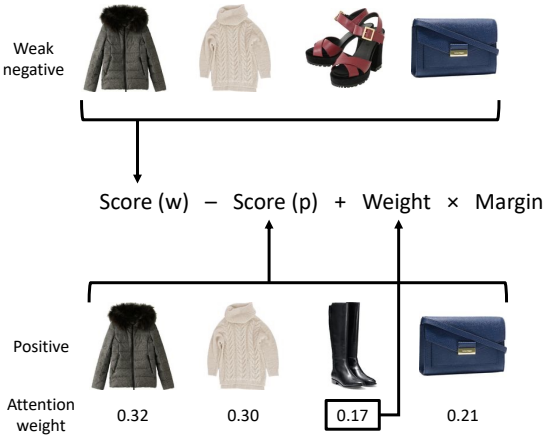

Figure 4: Illustration of the adaptive margin loss. Score (w) indicates the compatibility score of a weak negative outfit, and Score (p) indicates the compatibility score of a positive outfit. The margin is weighted by the learned attention weight of the switched item representing its importance in determining the compatibility of an outfit.

The overall training loss is defined as follows:

$$\mathcal{L} = \sum_B [\mathcal{L}_{FL}(B) + c_{CL} \cdot \mathcal{L}_{CL}(B) + c_{AM} \cdot \mathcal{L}_{AM}(B)] \tag{8}$$

where c_{CL} and c_{AM} are balancing coefficients.

4 Experiments

Our experiments aim to address the following questions:

- **Q1**: How does our proposed method perform on personalised outfit recommendation compared to baselines?
- **Q2**: How does contrastive learning improve the model's performance on the outfit recommendation?
- **Q3**: How does the number of history outfits affect the model's performance on the outfit recommendation?
- **Q4**: How does our proposed method perform on item recommendation given a partial outfit?
- **Q5**: How effective is the adaptive margin loss on the item recommendation?

4.1 Experimental Settings

4.1.1 Datasets. We used two popular fashion datasets, IQON3000 [29] and Polyvore-630 [20], for the evaluation. In both datasets, outfits were created by shoppers, which clearly reflects each shopper's preferred outfit styles. We consider outfits created by the same shopper as the shopper's purchase history. The number of items per outfit varies in IQON3000, while Polyvore-630 contains exactly three items per outfit. A summary of the datasets is shown in Table 1.

4.1.2 Compared Methods. In both outfit and item recommendation tasks, we compared our model with five baselines:

(1) Bi-LSTM [6]: Bi-LSTM for fashion compatibility is a model which learns compatibility of an outfit by sequentially predicting each item conditioned on the previous items.

Table 1: The number of shoppers, outfits, and fashion items in IQON3000 and Polyvore-630.

Dataset	# Shoppers	# Outfits	# Items
IQON3000 [29]	3,568	308,747	672,335
Polyvore-630 [20]	630	150,380	205,234

(2) FHN [20]: Fashion hashing network (FHN) is a composite model of hashing modules which learn binary codes of shoppers and items to score item-item compatibility and item-shopper compatibility.
(3) LPAE [19]: Learnable personalised anchor embedding (LPAE) is a model which encodes anchors specific to each shopper and general anchors by using only images of items.
(4) LPAE-T [19]: LPAE-T is a variant of LPAE which leverages both images and title of items by concatenating image features and text features to form item features.
(5) OutfitTransformer [27]: OutfitTransformer is the SOTA non-personalised outfit recommendation model which leverages transformer encoders to learn the internal compatibility of an outfit.

As baselines, we selected all the competitive models with published code. All reported numbers are obtained on the original test sets. To assess the significance of observed performance differences between models, we used sampled 10,000 bootstrapped test sets and used them to compute confidence intervals for HAT's metrics. We consider the difference between a baseline and HAT significant if the baseline's metric lies outside HAT's 95% confidence interval. Confidence intervals (CIs) are shown in Tables 2 and 5.

4.1.3 Implementation Details. For all the baselines and our model, we used frozen CLIP [25] ViT-B/32 weights (θ, ϕ) for the image encoder and the text encoder without fine-tuning. We stacked a learnable MLP on top of the CLIP embeddings to adjust the embeddings without having to fine-tune the larger CLIP models. We set the maximum number of history outfits as ten for all the experiments and the hyperparameters as $\alpha = 0.5$, $\gamma = 2$, $\tau = 1$, $m = 5$, $c_{CL} = 1$, and $c_{AM} = 0.5$. We used AdamW [18] optimiser and all experiments were conducted in PyTorch framework [22].

4.2 Personalised Outfit Recommendation

We evaluated the models' performance on personalised outfit recommendation with the **compatibility prediction (CP)** task. CP evaluates whether a model can determine compatibility of an outfit. It measures the area under the receiver operating characteristic curve (AUC) of compatibility scores given positive samples ($y^o = 1$) and negative samples ($y^o = 0$).

4.2.1 Negative Sampling Methods. For every positive outfit in both datasets, we created negative outfits in two different ways. Firstly, similar to the previous studies on outfit recommendation [26], we randomly switched each item in an outfit to another item within the same item category, resulting in the ratio of positive and negative outfits as 1:1. We call this dataset **CP-Random**.

We reiterate our argument from Section 3.3.3 that while CP-Random is useful to evaluate the model's ability of distinguishing a random outfit from a compatible outfit, the task is rather easy. In particular, it is difficult to evaluate a model's personalisation capability since compatible outfits can be incompatible with a shopper's individual taste, something that CP-Random does not reflect. A personalised recommendation model should avoid recommending outfits that do not align with the shopper's personal style even if the outfits are internally compatible.

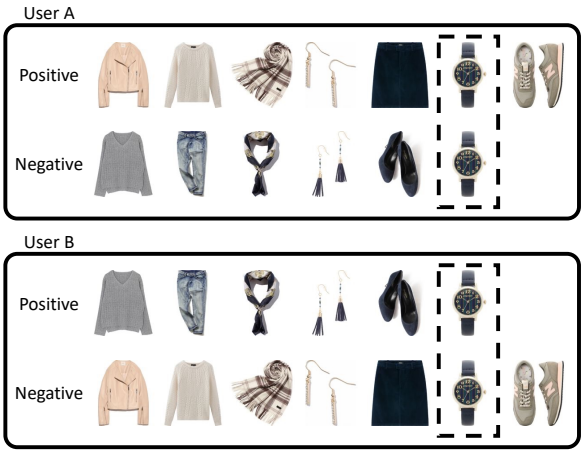

Figure 5: An example of a pair of positive outfit and negative outfit in CP-Hard. Shopper A and Shopper B have different positive outfits that share the same watch. For Shopper A, we consider the positive outfit of Shopper B as a negative outfit, and for Shopper B, we consider the positive outfit of Shopper A as a negative outfit.

In order to evaluate compatibility of an outfit with a shopper's taste, we treated positive outfits from one shopper as negative outfits for all other shoppers. To avoid cases where a shopper did not purchase outfits because he had not seen them, we only consider those positive outfits from other shoppers that have at least one overlapping item with each outfit. Intuitively, this indicates the two shoppers were exposed to the same item but purchased other outfits, which implies different fashion styles. We call this dataset **CP-Hard**. An example is shown in Fig. 5. Note that we trained models only with CP-Random and evaluated them on both CP-Random and CP-Hard. HAT was also trained on weak negative samples describes in Section 3.3.3.

4.2.2 Experimental Results (Q1). Table 2 shows the experimental results of personalised outfit recommendation. For IQON3000, our method significantly outperforms all baselines. The biggest performance gap between HAT and the second best model is observed in the CP-Hard task where HAT outperforms FHN by 8.9%. While LPAE-T is the second best model with performance difference of 0.37% from HAT in CP-Random, the performance difference is much larger in CP-Hard, which is 11.16%. This highlights HAT's personalisation capability by leveraging shopper purchase history. It also shows that CP-Random is inadequate for evaluating personalised outfit recommendation.

A similar trend is observed in Polyvore-630. Our method outperforms the others in every metric, and the biggest performance gap

Table 2: AUC comparison of CP with IQON3000 and Polyvore-630.

	IQON3000		Polyvore-630	
Model	CP-Random (AUC)	CP-Hard (AUC)	CP-Random (AUC)	CP-Hard (AUC)
Bi-LSTM [6]	0.8909	0.4915	0.8051	0.5158
FHN [20]	0.9047	0.5675	0.7490	0.4815
LPAE [19]	0.9510	0.5605	0.7977	0.5127
LPAE-T [19]	0.9628	0.5449	0.7872	0.4881
OutfitTransformer [27]	0.9468	0.4650	0.7021	0.4528
HAT (ours)	**0.9665**	**0.6565**	**0.8569**	**0.6159**
HAT 95% CI	(0.9653, 0.9671)	(0.6521, 0.6585)	(0.8534, 0.8599)	(0.6109, 0.62310)
Gain	+0.4%	+15.7%	+6.4%	+19.4%

Table 3: Ablation experiment of contrastive learning.

Dataset	Variant	CP-Random (AUC)	CP-Hard (AUC)
IQON3000	Without $\mathcal{L}_{CL}$	0.9280	0.5350
	With $\mathcal{L}_{CL}$	**0.9665**	**0.6565**
Polyvore-630	Without $\mathcal{L}_{CL}$	0.7183	0.4888
	With $\mathcal{L}_{CL}$	**0.8569**	**0.6159**

between HAT and the second best model is seen in CP-Hard where HAT outperforms Bi-LSTM by 10.01%. Despite Bi-LSTM being the simplest baseline, it outperforms more complex models such as LPAE and FHN in CP-Random and CP-Hard. This difference might be due to the smaller dataset size and the simpler outfit configuration where every outfit in Polyvore-630 has exactly three items from three fixed categories while the number of items in an outfit in IQON3000 ranges from 2 to 20 from 59 different categories.

4.2.3 Effectiveness of Contrastive Learning (Q2). We conducted an ablation study on the contrastive loss where the personalised outfit recommendation performance with and without $\mathcal{L}_{CL}$ is shown in Table 3. The performance of both CP-Random and CP-Hard with $\mathcal{L}_{CL}$ is higher than without $\mathcal{L}_{CL}$ for both datasets. This illustrates that contrastive learning is essential in leveraging history outfits properly by aligning outfit representations from the same shopper's history.

4.2.4 Impact of Number of History Outfits (Q3). The experimental results in Section 4.2.2 and Section 4.2.3 clearly demonstrate HAT's personalisation capability when leveraging purchase history. We conducted ablation studies on how the number of history outfits affects the personalisation performance. Table 4 shows CP-Random and CP-Hard results with different numbers of history outfits. When the number of history outfits equals to zero, the model is not personalised. In CP-Hard, a higher number of history outfits correlates with better performance on both datasets, indicating the importance of using all of the purchase history for personalisation. The performance in CP-Random is initially higher with history outfits but decreases as the number of outfits exceeds ten. This shows that while incorporating more history outfits is critical in differentiating different shoppers' personal styles, it may not be as beneficial when distinguishing random outfits.

4.3 Fill-in-the-blank Task

The CP task evaluates how well a model can estimate human compatibility judgements. For real e-commerce applications, recommending the correct item out of a candidate pool is more important. We therefore evaluate the models' performance on the **fill-in-the-blank (FITB)** task. FITB evaluates whether a model can complete an outfit where one item is masked and four different choices are given (i.e. one correct item and three wrong items). It measures the accuracy of the model in selecting the correct item for the missing spot. We did not further train the models for this task. Instead, we estimated the CP scores of the four possible outfits, and if the score of the ground-truth outfit is highest, it was considered as a correct prediction.

Notice that our version of FITB requires the model to pick the correct item without any additional information. Previous work [27] has provided the title of the target outfit as input, making FITB an image retrieval task. Having access to the target title is not a realistic scenario in e-commerce applications and we thus do not provide the title in our evaluations. This explains why the FITB results we report for the Outfit Transformer on Polyvore are lower than those reported in [27].

4.3.1 Negative Sampling Methods. We used two methods to sample negative items for the masked position of an outfit. The first method is to randomly select an item regardless of the item category, and the second method is to randomly select an item in the same item category of the target item. We call the first dataset **FITB-Random** and the second one **FITB-Hard**, as we consider it a more difficult task.

4.3.2 Experimental Results (Q4). Table 5 shows test accuracy of FITB. In both datasets, HAT significantly outperforms the other models. Unsurprisingly, FITB-Random accuracy is higher than FITB-Hard accuracy for every model in IQON3000. However, in Polyvore-630, half of the models performed better in FITB-Random, and the other half performed better in FITB-Hard. This result might be caused by the limited number (3) of item categories in Polyvore-630, which results in the many of negative items in FITB-Random having the same item category with the missing item. On expectation, 33% of FITB-Random outfits are in fact FITB-Hard outfits in Polyvore. Accordingly, the difference in difficulty between FITB-Random and FITB-Hard in Polyvore-630 is less evident than IQON3000.

Table 4: Impact of the maximum number of history outfits on personalised outfit recommendation.

	IQON3000		Polyvore-630	
# History outfits	CP-Random (AUC)	CP-Hard (AUC)	CP-Random (AUC)	CP-Hard (AUC)
0	0.9511	0.4859	0.6772	0.4652
10	**0.9665**	0.6565	**0.8569**	0.6159
20	0.9652	0.6653	0.8485	0.6246
30	0.9647	**0.6701**	0.8456	**0.6249**

Table 5: Accuracy comparison of FITB with IQON3000 and Polyvore-630.

	IQON3000		Polyvore-630	
Model	FITB-Random (Acc)	FITB-Hard (Acc)	FITB-Random (Acc)	FITB-Hard (Acc)
Bi-LSTM [6]	0.7258	0.5627	0.4028	0.4554
FHN [20]	0.5887	0.5645	0.4252	0.4813
LPAE [19]	0.6739	0.5768	0.5164	0.4887
LPAE-T [19]	0.6960	0.6006	0.4889	0.4652
OutfitTransformer [27]	0.7195	0.5969	0.3989	0.3870
HAT (ours)	**0.7406**	**0.6394**	**0.5356**	**0.5363**
HAT 95% CI	(0.7370, 0.7443)	(0.6355, 0.6434)	(0.5317, 0.5448)	(0.5308, 0.5436)
Gain	+2.0%	+6.5%	+3.7%	+9.7%

Table 6: Ablation experiments of the adaptive margin loss.

Dataset	Variant	FITB-Random (Acc)	FITB-Hard (Acc)
IQON3000	Without $\mathcal{L}_{AM}$	0.6789	0.5822
	Without A_i^p	0.7257	0.6164
	With $\mathcal{L}_{AM}$, A_i^p	**0.7406**	**0.6394**
Polyvore-630	Without $\mathcal{L}_{AM}$	0.5139	0.5060
	Without A_i^p	0.5210	0.5160
	With $\mathcal{L}_{AM}$, A_i^p	**0.5356**	**0.5363**

4.3.3 Ablation Studies on Adaptive Margin Loss (Q5). We conducted two ablation studies on the adaptive margin loss. The first study is to test the effectiveness of introducing weak negative samples where we trained the model with and without $\mathcal{L}_{AM}$. The second ablation study is to examine whether using the attention weight A_i^p improves the performance compared to the non-adaptive margin loss (i.e., setting $A_i^p = 1$ in Eq. (7)). Table 6 shows that the model without $\mathcal{L}_{AM}$ performs the worst and the model with $\mathcal{L}_{AM}$ and A_i^p performs the best in both datasets. This highlights that having weak negatives in combination with an adaptive margin loss is crucial in FITB. This finding corroborates our intuition that weak negatives provide a stronger learning signal to the model since they are harder to distinguish from positive outfits.

5 Conclusion

In this work, we have introduced the History-Aware Transformer (HAT) model for personalised outfit compatibility prediction. Making compatibility models history-aware is a straightforward way of personalising outfit recommendations. We believe that this is highly relevant for practical use cases since online shoppers prefer personalised recommendations over generic ones.

We have also shown how to improve outfit recommendation models by introducing weak negative samples in the training process, i.e. outfits where we switch one item at random. This makes weak negatives harder to distinguish from outfits that are known to be compatible, thus providing a stronger learning signal to the model. The key to incorporating these weak negatives into model training is an adaptive margin loss whose margin increases with the attention weight for the switched item.

Our experiments show that the proposed model outperforms existing baselines in scoring outfits as well as in choosing a missing item to complete an outfit. In several ablation studies, we demonstrate the necessity of the weak negatives and the margin loss to the personalisation performance of HAT. The ablations also highlight the importance of aligning outfit representations via supervised contrastive training. Nonetheless, there are several shortcomings to the proposed model for future research.

First, our model scores an outfit instead of assembling it from scratch. The model being able to produce outfit representations can greatly benefit retrieval tasks to build an outfit. Second, the proposed model does not take the sequence of the customer history into account. A purchase made a year ago is arguably less informative of a shopper's current preferences than their most recent purchase. One simple modification to account for the diminishing informativeness of older items is to introduce an exponentially decaying weighting scheme on the importance of older outfits in a shopper's history. Finally, the way we represent fashion items is currently restricted to titles and images. We believe that incorporating more meta information such as material and customer reviews will help our model make more accurate compatibility judgements.

References

[1] Qiwei Chen, Huan Zhao, Wei Li, Pipei Huang, and Wenwu Ou. 2019. Behavior sequence transformer for e-commerce recommendation in Alibaba. In *Proceedings of the 1st International Workshop on Deep Learning Practice for High-Dimensional Sparse Data* (Anchorage, Alaska) *(DLP-KDD '19)*. 4 pages. https://doi.org/10.1145/3326937.3341261

[2] Wen Chen, Pipei Huang, Jiaming Xu, Xin Guo, Cheng Guo, Fei Sun, Chao Li, Andreas Pfadler, Huan Zhao, and Binqiang Zhao. 2019. POG: Personalized Outfit Generation for Fashion Recommendation at Alibaba IFashion. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (Anchorage, AK, USA) *(KDD '19)*. Association for Computing Machinery, New York, NY, USA, 2662–2670. https://doi.org/10.1145/3292500.3330652

[3] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, Rohan Anil, Zakaria Haque, Lichan Hong, Vihan Jain, Xiaobing Liu, and Hemal Shah. 2016. Wide & Deep Learning for Recommender Systems. In *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems (DLRS 2016)*. 7–10.

[4] Xue Dong, Xuemeng Song, Fuli Feng, Peiguang Jing, Xin-Shun Xu, and Liqiang Nie. 2019. Personalized capsule wardrobe creation with garment and user modeling. In *Proceedings of the 27th ACM international conference on multimedia*. 302–310.

[5] M. Gutmann and A. Hyvärinen. 2010. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *AISTATS*, Y.W. Teh and M. Titterington (Eds.), Vol. 9. 297–304.

[6] Xintong Han, Zuxuan Wu, Yu-Gang Jiang, and Larry S. Davis. 2017. Learning Fashion Compatibility with Bidirectional LSTMs. In *Proceedings of the 25th ACM International Conference on Multimedia* (Mountain View, California, USA) *(MM '17)*. Association for Computing Machinery, New York, NY, USA, 1078–1086. https://doi.org/10.1145/3123266.3123394

[7] Yangbangyan Jiang, XU Qianqian, and Xiaochun Cao. 2018. Outfit recommendation with deep sequence learning. In *2018 IEEE Fourth International Conference on Multimedia Big Data (BigMM)*. IEEE, 1–5.

[8] Wang-Cheng Kang and Julian J. McAuley. 2018. Self-Attentive Sequential Recommendation. In *ICDM*. 197–206.

[9] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised Contrastive Learning. In *NeurIPS*, H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (Eds.). 18661–18673. https://proceedings.neurips.cc/paper_files/paper/2020/file/d89a66c7c80a29b1bdbab0f2a1a94af8-Paper.pdf

[10] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised Contrastive Learning. In *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (Eds.), Vol. 33. Curran Associates, Inc., 18661–18673.

[11] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597* (2023).

[12] Yuncheng Li, Liangliang Cao, Jiang Zhu, and Jiebo Luo. 2017. Mining fashion outfit composition using an end-to-end deep learning approach on set data. *IEEE Transactions on Multimedia* 19, 8 (2017), 1946–1955.

[13] Zhi Li, Bo Wu, Qi Liu, Likang Wu, Hongke Zhao, and Tao Mei. 2020. Learning the compositional visual coherence for complementary recommendations. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*. 3536–3543.

[14] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollar. 2017. Focal Loss for Dense Object Detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

[15] Yusan Lin, Maryam Moosaei, and Hao Yang. 2020. OutfitNet: Fashion Outfit Recommendation with Attention-Based Multiple Instance Learning. In *Proceedings of The Web Conference 2020* (Taipei, Taiwan) *(WWW '20)*. Association for Computing Machinery, New York, NY, USA, 77–87. https://doi.org/10.1145/3366423.3380096

[16] Junling Liu, Chao Liu, Peilin Zhou, Renjie Lv, Kang Zhou, and Yan Zhang. 2023. Is ChatGPT a Good Recommender? A Preliminary Study. arXiv:2304.10149

[17] Alexander Lorbert, David Neiman, Arik Poznanski, Eduard Oks, and Larry Davis. 2021. Scalable and explainable outfit generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 3931–3934.

[18] Ilya Loshchilov and Frank Hutter. 2019. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations*.

[19] Zhi Lu, Yang Hu, Yan Chen, and Bing Zeng. 2021. Personalized Outfit Recommendation With Learnable Anchors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 12722–12731.

[20] Zhi Lu, Yang Hu, Yunchao Jiang, Yan Chen, and Bing Zeng. 2019. Learning Binary Code for Personalized Fashion Recommendation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

[21] Lin Ning, Luyang Liu, Jiaxing Wu, Neo Wu, Devora Berlowitz, Sushant Prakash, Bradley Green, Shawn O'Banion, and Jun Xie. 2024. User-LLM: Efficient LLM Contextualization with User Embeddings. arXiv:2402.13598

[22] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Eds.), Vol. 32. Curran Associates, Inc.

[23] Zekai Qu, Ruobing Xie, Chaojun Xiao, Xingwu Sun, and Zhanhui Kang. 2024. The Elephant in the Room: Rethinking the Usage of Pre-trained Language Model in Sequential Recommendation. arXiv:2404.08796

[24] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning Transferable Visual Models From Natural Language Supervision. (2021). https://arxiv.org/abs/2103.00020

[25] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning Transferable Visual Models From Natural Language Supervision. In *Proceedings of the 38th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 139)*, Marina Meila and Tong Zhang (Eds.). PMLR, 8748–8763.

[26] Rohan Sarkar, Navaneeth Bodla, Mariya Vasileva, Yen-Liang Lin, Anurag Beniwal, Alan Lu, and Gerard Medioni. 2022. Outfittransformer: Outfit representations for fashion recommendation. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*. 2263–2267.

[27] Rohan Sarkar, Navaneeth Bodla, Mariya I Vasileva, Yen-Liang Lin, Anurag Beniwal, Alan Lu, and Gerard Medioni. 2023. Outfittransformer: Learning outfit representations for fashion recommendation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 3601–3609.

[28] Dandan Sha, Daling Wang, Xiangmin Zhou, Shi Feng, Yifei Zhang, and Ge Yu. 2016. An approach for clothing recommendation based on multiple image attributes. In *Web-Age Information Management: 17th International Conference, WAIM 2016, Nanchang, China, June 3-5, 2016, Proceedings, Part I 17*. Springer, 272–285.

[29] Xuemeng Song, Xianjing Han, Yunkai Li, Jingyuan Chen, Xin-Shun Xu, and Liqiang Nie. 2019. GP-BPR: Personalized Compatibility Modeling for Clothing Matching. In *Proceedings of the 27th ACM International Conference on Multimedia* (Nice, France) *(MM '19)*. Association for Computing Machinery, New York, NY, USA, 320–328. https://doi.org/10.1145/3343031.3350956

[30] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 2019. BERT4Rec: Sequential Recommendation with Bidirectional Encoder Representations from Transformer *(CIKM)*. 1441–1450. https://doi.org/10.1145/3357384.3357895

[31] Pongsate Tangseng, Kota Yamaguchi, and Takayuki Okatani. 2017. Recommending outfits from personal closet. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*. 2275–2279.

[32] Aäron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation Learning with Contrastive Predictive Coding. (2018). http://arxiv.org/abs/1807.03748

[33] Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9, 11 (2008).

[34] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems*, I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.), Vol. 30. Curran Associates, Inc.

[35] Liwei Wu, Shuqing Li, Cho-Jui Hsieh, and James Sharpnack. 2020. SSE-PT: Sequential Recommendation Via Personalized Transformer *(RecSys)*. 328–337. https://doi.org/10.1145/3383313.3412258

[36] Xuewen Yang, Dongliang Xie, Xin Wang, Jiangbo Yuan, Wanying Ding, and Pengyun Yan. 2020. Learning tuple compatibility for conditional outfit recommendation. In *Proceedings of the 28th ACM International Conference on Multimedia*. 2636–2644.

[37] Huijing Zhan and Jie Lin. 2021. Pan: Personalized attention network for outfit recommendation. In *2021 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2663–2667.

[38] Na Zheng, Xuemeng Song, Qingying Niu, Xue Dong, Yibing Zhan, and Liqiang Nie. 2021. Collocation and try-on network: Whether an outfit is compatible. In *Proceedings of the 29th ACM International Conference on Multimedia*. 309–317.

[39] Guorui Zhou, Xiaoqiang Zhu, Chenru Song, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. 2018. Deep Interest Network for Click-Through Rate Prediction *(KDD '18)*. 1059–1068. https://doi.org/10.1145/3219819.3219823