Journal Pre-proof

VICTOR: Visual incompatibility detection with transformers and fashion-specific contrastive pre-training

Stefanos-Iordanis Papadopoulos, Christos Koutlis,
Symeon Papadopoulos, Ioannis Kompatsiaris

PII: S1047-3203(22)00261-9
DOI: https://doi.org/10.1016/j.jvcir.2022.103741
Reference: YJVCI 103741

To appear in: *J. Vis. Commun. Image R.*

Received date : 2 September 2022
Revised date : 5 December 2022
Accepted date : 28 December 2022

Please cite this article as: S.-I. Papadopoulos, C. Koutlis, S. Papadopoulos et al., VICTOR: Visual incompatibility detection with transformers and fashion-specific contrastive pre-training, *J. Vis. Commun. Image R.* (2022), doi: https://doi.org/10.1016/j.jvcir.2022.103741.

Journal of Visual Communication and Image Representation (2022)

Contents lists available at ScienceDirect

Journal of Visual Communication and Image Representation

journal homepage: www.elsevier.com/locate/jvci

VICTOR: Visual incompatibility detection with transformers and fashion-specific contrastive pre-training

Stefanos-Iordanis Papadopoulos[*], Christos Koutlis, Symeon Papadopoulos, Ioannis Kompatsiaris

CERTH-ITI, 6th km Charilaou-Thermi, GR 57001 Thermi, Thessaloniki, Greece

ARTICLE INFO

Article history:
Received 2 September 2022

2000 MSC: 68T45, 68T50

Keywords: Recommendation System, Outfit Matching, Visual Compatibility, Computer Vision, Deep Learning

ABSTRACT

For fashion outfits to be considered aesthetically pleasing, the garments that constitute them need to be compatible in terms of visual aspects, such as style, category and color. Previous works have defined visual compatibility as a binary classification task with items in a garment being considered as fully compatible or fully incompatible. However, this is not applicable to Outfit Maker applications where users create their own outfits and need to know which specific items may be incompatible with the rest of the outfit. To address this, we propose the Visual InCompatibility TransfORmer (VICTOR) that is optimized for two tasks: 1) overall compatibility as regression and 2) the detection of mismatching items and utilize fashion-specific contrastive language-image pre-training for fine tuning computer vision neural networks on fashion imagery. We build upon the Polyvore outfit benchmark to generate partially mismatching outfits, creating a new dataset termed Polyvore-MISFITs, that is used to train VICTOR. A series of ablation and comparative analyses show that the proposed architecture can compete and even surpass the current state-of-the-art on Polyvore datasets while reducing the instance-wise floating operations by 88%, striking a balance between high performance and efficiency. We release our code at `https://github.com/stevejpapad/Visual-InCompatibility-Transformer`

1. Introduction

Fashion products do not exist in a vacuum. When customers consider buying a new garment they may contemplate its subjective appeal, price, quality or trendiness but also think of ways to match it with other pieces and how compatible it is with other items in their wardrobe. To help customers in their endeavours, contemporary e-commerce applications usually provide outfit recommendations and suggestions of how to "complete the look" based on an item of interest. Outfit compatibility is a rather challenging task: not only is it highly subjective but it also involves numerous variables such as the style, color, fit, patterns, proportions, textures

[*] Corresponding author
e-mail: `stefpapad@iti.gr` (Stefanos-Iordanis Papadopoulos), `ckoutlis@iti.gr` (Christos Koutlis), `papadop@iti.gr` (Symeon Papadopoulos), `ikom@iti.gr` (Ioannis Kompatsiaris)

of numerous garments and how these aspects interrelate. To this end, researchers have recently utilized computer vision neural networks, that learn to produce informative representations from fashion images, along with pairwise-based [1, 2], graph-based [3, 4] or attention-based neural networks [5, 6, 7] that learn to predict the compatibility of outfits.

However, previous studies define outfit compatibility prediction as a binary (OC_b) classification task. An outfit is either fully compatible or fully incompatible. This is a reasonable assumption for e-commerce applications that recommend fully compatible outfits to their customers. It is not as applicable to *Outfit Maker* applications[1], where users combine garments to create their own outfits. Instead, it would be more useful to offer an overall compatibility score and detect specific mismatching garments in order to inform users which items are not compatible with the rest of the outfit. This would give a sense of how aesthetically pleasing an outfit is and help users identify garments with clashing colors or patterns, select more suitable alternatives and generally fine-tune their outfits.

In this study we define outfit compatibility as a regression (OC_r) problem and also address the task of mismatching item detection (MID) in fashion outfits. To the best of the authors' knowledge, no previous study has addressed these tasks. We use the Polyvore outfit dataset [2] which consists of fully compatible and incompatible outfits to generate partially mismatching outfits (MISFITs). We propose the **V**isual **I**n**C**ompatibility **T**ransf**OR**mer, or VICTOR, a multi-tasking, Transformer-based architecture that is trained to predict the overall OC_r score and detect mismatching garments in an outfit. Previous works on OC_b either rely on feature extraction from computer vision models pre-trained on ImageNet[7, 8] or end-to-end (E2E) fine-tuning [9, 2, 10, 1, 11]. While E2E fine-tuning tends to significantly outperform feature extraction, it is notably more resource intensive. Instead, we utilize *fashion-specific contrastive language image pre-training* (FLIP) to fine-tune computer vision models for fashion imagery and then use the extracted visual features for OC_r and MID. The ablation study showed that multi-tasking outperforms the single-tasking and that multi-modality improves upon the visual-only versions on VICTOR while the comparative analysis showed that VICTOR with FLIP are capable of competing and even surpassing, the current state-of-the-art on Polyvore datasets for OC_b while reducing instance-wise floating point operations (FLOPs) by an impressive 88%. Notably, multi-tasking VICTOR improved upon the previous state-of-the-art (SotA) on the Polyvore-Disjoint dataset by 2.38%.

The main contributions of our work are:

- We define two new sub-tasks around visual compatibility, namely: outfit compatibility prediction as regression (OC_r) and mismatching item detection (MID) and examine them in the domain of Fashion.
- We propose VICTOR, a multi-tasking Transformer-based neural network that is optimized for both tasks and utilize fashion-specific contrastive language image pre-training (FLIP) for fine-tuning computer vision neural networks on fashion imagery. Moreover, we generate and experiment on the Polyvore-MISFITs which can be used as a benchmark dataset for OC_r and MID.
- We experiment with four computer vision backbone networks and perform an extensive ablation and comparative analysis that shows VICTOR with FLIP to be capable of competing and even surpassing the current state-of-the-art on Polyvore datasets (2.38% improvement on Polyvore-Disjoint) while reducing instance-wise floating point operations by 88% and total study-wise operations by up to 98%.

[1] Examples of outfit maker applications include: ShopLook, Smart Closet, Stylebook, Pureple and Combyne

2. Related Work

In recent years, researchers have shown increased interest in applying deep learning and computer vision neural networks [12] in order to address numerous tasks relevant to the Fashion domain including category and attribute classification [13, 14], trend forecasting [15, 16], popularity prediction [17, 18], fashion recommendations systems [19, 20] and among them, the task of outfit recommendations. In order to recommend complete outfits it is first necessary to understand which garments go well together and can create compatible and cohesive outfits.

The first studies to address the task, considered outfit compatibility as a series of pairwise comparisons between all comprising garments [1, 2]. Pairwise-based approaches have utilized Siamese [21] and triplet loss networks with either type-aware embeddings [2] or similarity-aware embeddings [1]. Other works, instead of aggregating garment-level relations attempted to capture global outfit-level representations with the use of bidirectional LSTMs [9] or graph neural networks [3, 4]. In practice, outfits are not ordered sequences; the order of the garments should not affect the model's predictions. Thus, recurrent neural networks are not the most suitable architecture for the task. On the other hand, graph-based approaches tend to require large "neighborhoods" of compatible garment-nodes as input in order to reach optimal performance which is problematic for new items that lack neighbor information and may straggle from the cold start problem [10].

In order to address the aforementioned challenges, more recent works have employed attention-based methods [5, 6, 7]. Attention mechanisms have been used in pairwise-based approaches [10, 22] but the Transformer architecture has been successfully used for personalised outfit recommendations [7] and complementary item retrieval [11]. With the use of multi-head attention, the Transformer is suitable for learning relations between multiple items, in this case the compatibility between all garments in an outfit. Additionally, by removing the positional encoding [23, 24] it can capture unordered relations between all garments.

However, all aforementioned studies have defined outfit compatibility as a binary classification problem. An outfit is treated as either fully compatible or fully incompatible. To the best of our knowledge, this is the first study to tackle the task of mismatching item detection (MID) and treat compatibility prediction as a regression (OC_r) instead of a binary task (OC_b).

Previous works have relied on visual, textual information and fashion categories for creating representations of garments in outfits. Transfer learning is generally being used for extracting visual information from the garment's images, either with feature extraction (FX) from ImageNet pretrained models [7, 8] or by end-to-end fine-tuning (E2E) for OC_b [9, 2, 10, 1, 11]. E2E tends to outperform FX-ImageNet since the visual features are trained to specialize on the target domain and task. Nevertheless, E2E is a highly resource intensive process since the gradients of a - usually large - network backbone need to be updated on top of the outfit matching neural network. In this study, we attempt to find the middle ground between the efficiency of FX and the high accuracy of E2E by utilizing contrastive language-image pre-training - inspired by [25] - with a focus on fashion imagery.

3. Methodology

3.1. Problem Formulation

In this study, we address the task of mismatching item detection (MID) in fashion outfits. Moreover, we define visual outfit compatibility prediction as a regression task (OC_r) - allowing for partially mismatching outfits - in contrast to previous studies that define it as a binary classification task (OC_b). Let a fashion outfit $O = \{g_1, g_2, \ldots, g_n\}$ consist of n garments g_i. Our architecture after processing the outfit images, $\mathcal{I} = \{I(g_1), I(g_2), \ldots, I(g_n)\}$, produces $n + 1$ outputs, one for the OC_r task denoted $Y_{OC_r} \in (0, 1) \subset \mathbb{R}$ and n for the MID task denoted $Y_{MID} \in (0, 1)^n \subset \mathbb{R}^n$, which are optimized to comply with the corresponding target variables, T_{OC_r} and T_{MID}. First, $T_{OC_r} \in (0, 1) \subset \mathbb{R}$ denotes the compatibility of the garments, where 0 means that all garments are incompatible, 1 that all are compatible and in-between values denote partial compatibility. Second, a list of binary values $T_{MID} = [x_{g_1}, x_{g_2}, \dots, x_{g_n}]$, with $x_{g_i} \in \{0, 1\}$ and $i = 1, \dots, n$, where 1 denotes the mismatching garments in outfit O. OC_r is defined as a regression task and MID as a multi-label classification task.

3.2. Generating Mismatching Outfits

Existing outfit datasets, e.g. Polyvore [2], provide annotations for fully compatible or fully incompatible outfits. In this study we attempt to address partial incompatibility and the detection of specific mismatching items within an outfit. To this end, we generate partially mismatching outfits (MISFITs) with the following method. For every matching outfit O, with $n > 2$ we generate m number of MISFITs by randomly selecting (i) the number of garments $1 \leq r \leq n - 2$ that will be replaced and (ii) their positions $\mathcal{P}$. The garments in positions $\mathcal{P}$ are then replaced with randomly selected items of the same category. We do not sample garments from different categories because this would result in very easy negative samples and the model would learn less useful relations [2], for example that an outfit can not consist of two dresses or two pairs of shoes. By relying on random negative sampling from the same category, our method generates a mix of hard and easy negative samples. We deem it important to have a mix of 'hard' and 'easy' negative samples since either could reflect the choices of different users in an outfit maker application. Newer fashion enthusiasts may make bigger mistakes that would be considered 'easy negative samples', while more advanced users could make more subtle mistakes that would be covered by 'hard negative samples'.

For O with $n = 3$ we only allow $r = 1$ because having outfits with only 1 compatible item is invalid. The target compatibility score is calculated as $T_{OC_r} = 1 - r/n$ and the mismatching items target is defined as a list T_{MID} of binary values with 1 in $\mathcal{P}$ positions denoting the incompatible garments and 0 in other positions denoting the compatible garments. The fully compatible outfits retain $T_{OC_r} = 1$ and $T_{MID} = [0, 0, \dots, 0]$, while the fully incompatible ones $T_{OC_r} = 0$ and $T_{MID} = [1, 1, \dots, 1]$, respectively.

3.3. VICTOR

The proposed pipeline of the Visual InCompatibility TransfORmer (VICTOR) is illustrated in Fig. 1. First, the images $\mathcal{I} = \{I(g_1), I(g_2), \dots, I(g_n)\}$ of all garments in an outfit O are passed through a visual encoder $\mathrm{E_V}(\cdot)$ that produces the corresponding vector representations $\mathbf{v}_{g_i} \in \mathbb{R}^{e \times 1}$, where e is the encoder's embedding dimension. Then, following Dosovitskiy et al. [24] that makes use of a classification token (CLS) we similarly consider a regression token (REG), a trainable vector that learns a global representation incorporating information about the relations of all garments in an outfit, and pass $\{< \mathrm{REG} >\} \cup \{\mathbf{v}_{g_i}\}_{i=1}^{n}$ through a Transformer decoder[2] $\mathrm{D}(\cdot)$. Outfits are not sequential objects thus we do not make use of positional encodings [23, 24] so as to capture the unordered relations between garments. The Transformer decoder $\mathrm{D}(\cdot)$ consists of L layers that have h attention heads of embedding dimension d. Finally, the OC_r score Y_{OC_r} and the MID scores list Y_{MID} of the outfit are calculated as:

$$\mathbf{v}_{g_i} = \mathrm{E_V}(I(g_i)) \tag{1}$$

$$[\mathbf{d}_{<\mathrm{REG}>}, \mathbf{d}_{g_1}, \dots, \mathbf{d}_{g_n}] = \mathrm{D}([< \mathrm{REG} >, \mathbf{v}_{g_1}, \dots, \mathbf{v}_{g_n}]) \tag{2}$$

$$Y_{OC_r} = \mathbf{W}_1 \cdot \mathrm{GELU}(\mathbf{W}_0 \cdot \mathrm{LN}(\mathbf{d}_{<\mathrm{REG}>})) \tag{3}$$

[2] $\mathrm{D}(\cdot)$ is actually structured as the encoder part of the original Transformer architecture but we use it to decode the image embeddings in our model thus we call it a decoder herein.

$$Y_{MID}[i] = \mathbf{W}_i \cdot \mathrm{GELU}(\mathrm{LN}(\mathbf{d}_{g_i})) \tag{4}$$

where $\mathbf{d}_{g_i} \in \mathbb{R}^{d\times 1}$ and $\mathbf{d}_{\text{<REG>}} \in \mathbb{R}^{d\times 1}$ are the Transformer's outputs, $\mathbf{W}_0 \in \mathbb{R}^{\frac{e}{2}\times d}$, $\mathbf{W}_1 \in \mathbb{R}^{1\times \frac{e}{2}}$ and $\mathbf{W}_i \in \mathbb{R}^{1\times d}$ are sigmoid activated dense projection layers (learnable bias terms are considered but omitted here for clarity), LN stands for Layer Normalization and GELU is the activation function. Zero padding is also considered for D(·) input in outfits with less than 19 items; that being the largest outfit size in the Polyvore dataset.

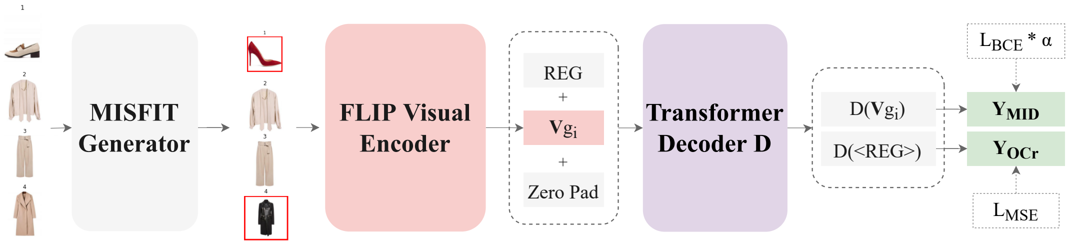


Fig. 1: Workflow of the VICTOR architecture.

D(·) utilizes multi-head attention, thus each token contains information about a garment's interrelations with all other garments. In our case this translates to an item being mismatching with the rest of the items in the outfit. Sarkar et al. [11] proposed the use of the CLS token for predicting the overall compatibility of the outfit. However, after experimentation, we found this to be sub-optimal for the MID task and our architectural approach to perform consistently better. VICTOR is optimized based on two different loss functions. Y_{OC_r} - being a regression task - is optimized based on the mean squared error loss function (L_{MSE}), while Y_{MID} - being a multi-label classification task - is optimized based on the binary cross entropy (L_{BCE}) loss, ignoring the zero padded items. However, the two loss functions do not necessarily have balanced values. We therefore introduce α, a hyper-parameter for weighted combination of the two loss functions as a standard multi-objective optimization practice. The final loss for VICTOR is calculated as $L = L_{MSE} + L_{BCE} \cdot \alpha$.

In this study, our focus is mainly centered around visual features. However, we also experiment with text in order to be comparable with the current state of the art. For the experiments that also use text, we pass the text descriptions $\mathcal{T} = \{T(g_1), T(g_2), \dots, T(g_n)\}$ of outfit O through a text encoder $\mathrm{E_T}(\cdot)$ that produces the corresponding vector representations $\mathbf{t}_{g_i} \in \mathbb{R}^{e\times 1}$. $\mathbf{t}_{g_i}$ are concatenated with $\mathbf{v}_{g_i}$ and passed through the transformer decoder D(·). Thereafter, the following steps are identical with the ones described for the image-only experiments.

3.4. Fashion-specific language image pre-training (FLIP)

Analysing the visual compatibility of fashion items requires the use of computer vision neural networks for producing informative representations of said items. Unlike previous works that have utilized feature extraction from ImageNet-pretrained models or end-to-end fine-tuning, we propose the use of contrastive language-image pre-training for fashion imagery (FLIP). FLIP's workflow is illustrated in Fig. 2 and is following the training procedure proposed by Radford et al. [25]. FLIP consists of one visual $\mathrm{E_V}(\cdot)$ and one textual $\mathrm{E_T}(\cdot)$ encoder. Image-text pairs $(I(g_i), T(g_i))$ are passed through their respective encoders and the resulting embeddings are projected onto the joint embedding space with the use of two fully connected layers of the same size one for each encoder, as shown below:

$$\mathrm{F_V(i)} = \mathbf{W_V} \cdot \mathrm{E_V}(I(g_i)) \tag{5}$$

$$\mathrm{F_T(i)} = \mathbf{W_T} \cdot \mathrm{E_T}(T(g_i)) \tag{6}$$

The dot product between image and text projection embeddings are calculated and the loss function is defined as the mean cross entropy between the predicted and the target image-text pairs, the latter being reflected by the main diagonal. Essentially, FLIP compares a collection of image-text pairs and creates a joint embedding space between image and language pairs where actual pairs are projected closer in the embedding space. In this sense, it can be considered to be a contrastive method.

Our rationale for utilizing FLIP is that it balances performance and efficiency. Training computer vision models end-to-end for outfit compatibility can yield a high performance but is a rather resource intensive process. On the other hand, ImageNet-pretrained models do not specialise on fashion imagery and can only produce a general visual representation. In contrast, the visual encoder of FLIP will learn to produce fashion-specific features. FLIP does not rely on the concept of classes and as a result it does not require annotated fashion datasets, which are expensive and time consuming to produce, instead it relies on image-text pairs of existing fashion products which are easier to attain from retailers and e-commerce applications. Moreover, we may train a single FLIP model, extract the visual features from fashion imagery and textual features from product descriptions and re-use them for numerous experiments on outfit compatibility, such as hyper-parameter tuning and ablation analyses, without requiring E2E fine-tuning. Thus, significantly reducing floating point operations (FLOPs) and by extension computational costs and training time.

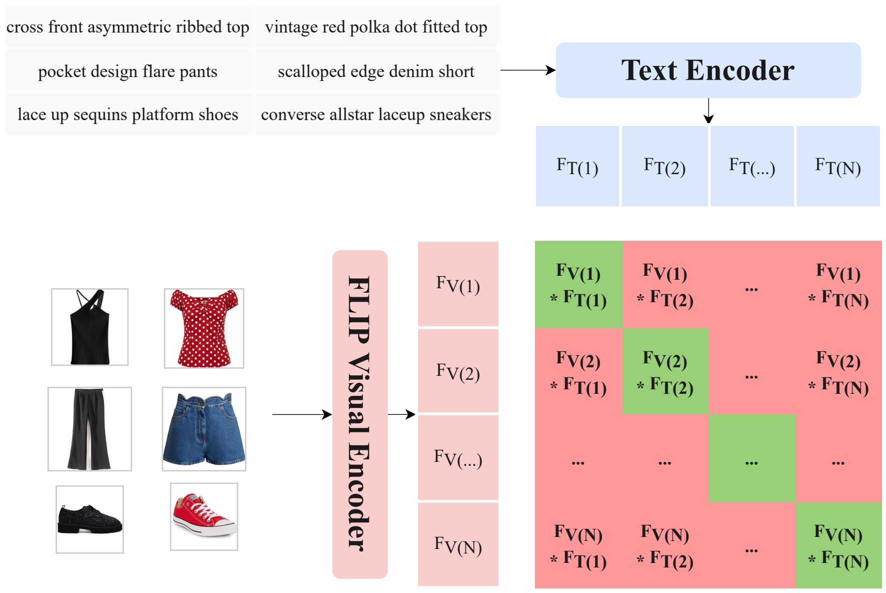


Fig. 2: Workflow of fashion language-image pre-training (FLIP). FLIP consists of a visual and a textual encoder that are trained contrastively to predict the correct image-text pair which are placed in the main diagonal. Images and texts are selected with in-batch sampling.

4. Experimental Setup

4.1. Polyvore Dataset

The Polyvore dataset is a widely used benchmark dataset for outfit recommendation that was collected by Vasileva et al. [2]. The dataset provides 68,306 matching outfits comprising 251,008 unique garments. Each garment comes with multi-modal information including an image, product name, description and associated fashion categories consisting of 14 *types* and 142 categories including *bottoms*: “skirt”, “long skirt”, *tops*: “sweater”, “turtleneck sweater”, *shoes*: “boots”, “flat sandals” but also *hats*, *jewelry* and other *accessories*. For every matching outfit the authors have generated an equal amount of fully incompatible outfits by randomly replacing each garments with items of the same category. The dataset comes in two versions that have fixed training, validation and testing splits. The first version of Polyvore consists of 106,612, 10,000, 20,000 outfits for training, validation and testing respectively. There are no overlapping outfits between the different splits but garments can overlap between the splits. The second version, Polyvore-Disjoint, consists of 33,990, 6,000, 30,290 outfits for training, validation and testing but there are no overlapping garments between the splits. Each outfit has at least 2, a maximum of 19 and a median value of 5 garments. As the target variable T_{OC_b}, fully compatible outfits have a score of 1 while fully incompatible have 0.

Outfits with more than 10 garments make up less than 0.5% of all outfits and could therefore be considered outliers. Moreover, we found that outfits with more than 10 garments often have more than one garment of the same category e.g. two pants or two jackets, which is infeasible. However, we do not filter anything out so as to ensure comparability with previous works.

4.2. Polyvore-MISFITs Dataset

We apply the MISFIT generation process described in section 3.2 on the Polyvore dataset for $m = 2$ and $m = 4$. $m = 2$ creates a balanced dataset between the initial and the generated outfits, with 133,944 MISFITs out of the 270,556 in total which are distributed into 104,498, 9,794, 19,652 for training, validation and testing. This means that 25% of the dataset consists of fully compatible, 50% generated MISFITs and 25% fully incompatible outfits for $m = 2$. $m = 4$ generates 267,888 MISFITs with a total of 404,500 outfits which are split into 315,608, 29,588, 59,304 with 16.88% of the dataset being fully compatible, 66.22% generated MISFITs and 16.88% fully incompatible outfits. The distribution of the compatibility scores for the Polyvore-MISFITs dataset are illustrated in Fig. 3.

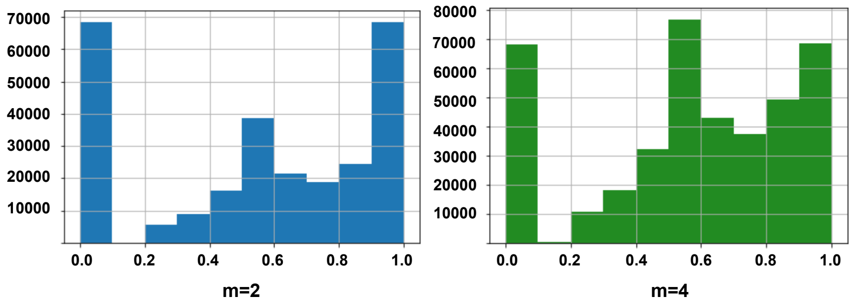


Fig. 3: Distribution of compatibility scores for Polyvore-MISFITs with m=2 and m=4.

Fig. 4 presents two indicative examples of generated MISFITs. On top there are two women’s outfits of different styles which are annotated as matching. On the left, a classic monochromatic look with a loose fit and on the right a casual look with black pieces and blue jeans. The MISFIT generation process has randomly replaced certain garments of the original outfit with items of the same category. For example, the beige pair of wide-fitting pants is replaced with leopard-print leggings (item 3, row 1) and the leather jacket (right outfit) is replaced with a colorful Aztec-pattern jacket (item 6, row 1). These, like most replaced garments, are not matching the aesthetic and style of the initial outfit. Thus, they are correctly categorised as mismatching items. On the other hand, the beige loafers are replaced with a beige pair of heels (item 1, row 4). This could be considered a 'hard negative' sample because the items are quite similar. Actually, some users could even consider them to be interchangeable and thus not mismatching. Hard negative samples like this forces the model to recognize and focus on fine-grained characteristics of the garments and their interrelations. Furthermore, the same pair of loafers is replaced with a brown heel with a pink tassel (item 1, row 2) which could be considered to be an 'easy negative' sample because it breaks the level of formality and the color consistency of the rest of the outfit. It is realistic to assume that different users would make mistakes on either end of the spectrum.

To ensure reproducibility and in order to encourage further research in the field, we provide the code[3] that generates the Polyvore-MISFITs dataset.

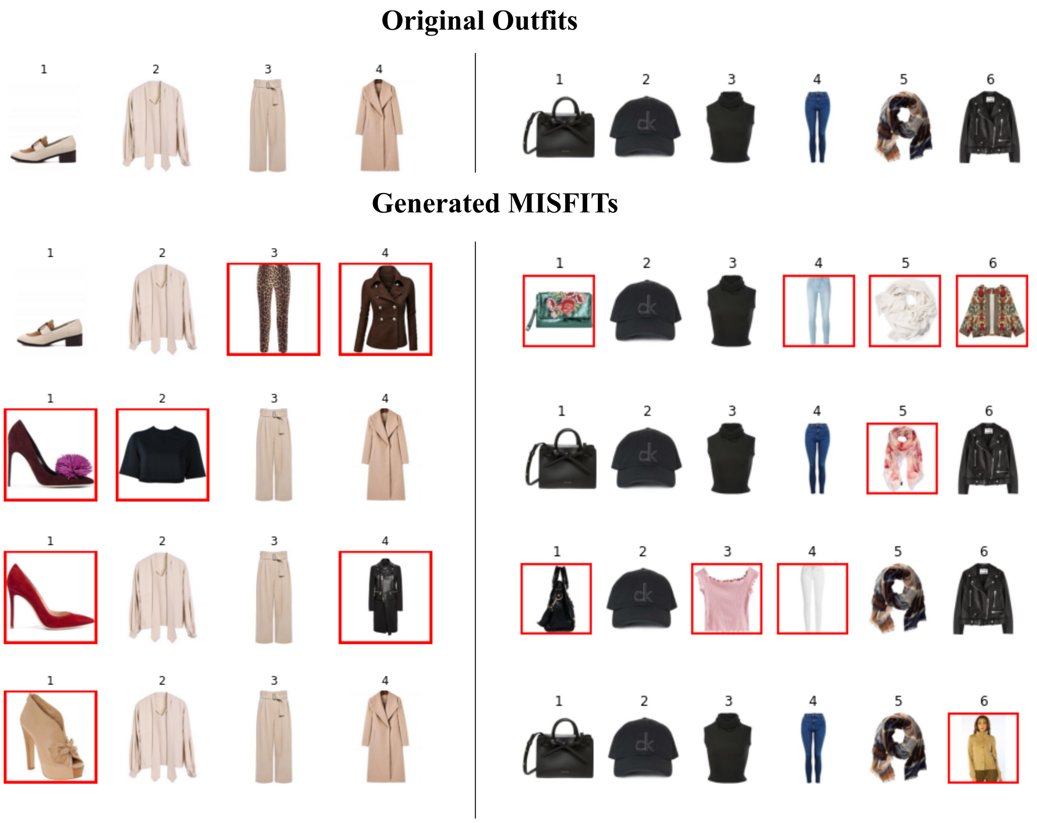


Fig. 4: Examples of generated MISFITs from fully compatible outfits. Red frames denote the mismatching items.

[3] https://github.com/stevejpapad/Visual-InCompatibility-Transformer

4.3. Implementation Details

We perform an ablation and comparative analysis and in order to distinguish different versions of VICTOR, we denote the training task in square brackets. The proposed multi-tasking learning (MTL) model optimized both for OC_r and MID is referred to as VICTOR[MTL]. Furthermore, we define: (1) VICTOR[OC_b] trained only for binary outfit compatibility, optimized based on the binary cross entropy loss function, (2) VICTOR[OC_r] trained only for compatibility as regression optimized based on on the MSE loss function and (3) VICTOR[MID] trained only for mismatching item detection, optimized based on the multi-label binary cross entropy loss function. For all versions of VICTOR, we select $L = 8$ transformer layers of $d = 64$ dimensions, $h = 16$ attention heads, a dropout rate of 0.2 and a batch size of 512. We train VICTOR[MTL] four times with $\alpha \in [0.2, 0.5, 1, 2]$ and denote different values of α as VICTOR[MTL;α]. Wherever required, we also denote the version of Polyvore-MISFITs that was used to train VICTOR as VICTOR[MTL;α;m].

We use the image-text pairs from the Polyvore-Disjoint dataset for training FLIP since there is no overlap between training, validation and testing sets. For FLIP's visual encoder $\mathrm{E_V}$, we experiment with four models: 1) ResNet18 [26], 2) EfficientNetV2-B3 [27], 3) MLP-Mixer B/16 [28] and ViT B/32 [24]. The aformentioned models are taken from the timm library[4] and are initially pre-trained on ImageNet. The input image sizes are 224 for all models expect EfficientNetV2-B3 which is 300. For FLIP's textual encoder $\mathrm{E_T}$, we use CLIP's Transformer text encoder and do not fine-tune it any further. We select a projection layer of 512 and a batch size of 32 for FLIP.

We train both FLIP and VICTOR for 20 epochs with the Adam optimizer and a learning rate scheduler with an initial learning rate of 1e-4 that reduces by a rate of 0.1 at 10 epochs.

Regarding the evaluation protocol, we follow all previous works that use the area under the roc curve (AUC) as the evaluation metrics for OC_b. For OC_r we report the mean absolute error (MAE) and for MID the binary accuracy and exact match. We use the training, validation and testing sets as provided by the Polyvore dataset in order to ensure fair comparability. We checkpoint the network's parameters with TOPSIS [29] based on the validation MAE, binary accuracy and exact match.

5. Results

5.1. FLIP and FLOPs

Table 1: Performance of computer vision models fine-tuned with FLIP in terms of the cross entropy loss.

Model	Cross entropy loss (↓)
ViT B/32	1.27
ResNet18	1.23
MLP-Mixer B/16	1.21
EfficientNetV2-B3	**1.07**

We fine-tune four computer vision neural networks for fashion imagery with the use of fashion language-image pre-training (FLIP). Their performance in terms of the cross entropy loss can be seen in Table 1. Lower values of cross entropy loss translates into fewer mistakes when matching the visual and textual projections of actual image-text pairs. However, lower cross entropy loss may not necessarily translate into better performance for VICTOR. Therefore, this issue should be examined empirically. Our rationale for employing FLIP was to fine-tune the models on fashion imagery while avoiding end-to-end (E2E) fine-tuning for outfit compatibility which can be considerably resource-intensive.

[4] `https://github.com/rwightman/pytorch-image-models`

Table 2: Floating-point operations (FLOPs) of VICTOR when trained with FLIP or end-to-end (E2E) fine-tuning with different computer vision models.

Model	**Model Parameters**	**FLIP**	**VICTOR**	**FLIP + VICTOR**	**VICTOR (E2E)**	**% ↓**
ResNet18	1.14E+07	5.36E+09	1.82E+08	5.54E+09	4.55E+10	87.8
EfficientNetV2-B3	1.30E+07	6.07E+09	1.55E+09	7.63E+09	6.02E+10	87.3
MLP-Mixer B/16	5.93E+07	7.31E+09	4.00E+08	7.71E+09	2.40E+11	96.8
ViT B/32	8.76E+07	1.56E+10	4.00E+08	1.60E+10	8.26E+10	80.62

To measure the efficiency gains of FLIP, we calculate the number of floating point operations (FLOPs) using Facebook's *fvcore*[5]. Table 2 presents the FLOPs of each computer vision model for a single instance of training. We observe that employing FLIP and then utilizing the extracted visual features to train VICTOR reduces the number of FLOPs by an average of 88.14% compared to E2E training. Moreover, if we not only consider instance-wise FLOPs but also epoch-wise FLOPs there is an average decrease of up to 94.86%. This is due to FLIP being trained on the Polyvore-Disjoint dataset (86,624 training+validation instances) - but is then also used for Polyvore - compared to the Polyvore's 202,446 training+validation instances. Furthermore, we should also consider the re-usability of FLIP, meaning that a FLIP model can be trained once but its extracted features can then be re-used with no additional cost. In our study, we run 12 experiments per computer vision model, for the ablation study and the tuning of α. Compared to using standard E2E training within the same experimental setup, we have actually reduced the number of FLOPs by an impressive average of 98.81%. Utilizing FLIP proved to be significantly more efficient than conventional E2E training for outfit compatibility prediction.

Table 3: Ablation analysis between VICTOR[OC_r], VICTOR[MID] and VICTOR[MTL] on Polyvore-MISFITs dataset with $m = 2$ and $m = 4$. For VICTOR[MTL] the comparison between single-modal (text-only or image-only) and multi-modal (text+images) inputs is also presented. The best performing $\alpha = a|b$ based on TOPSIS with a for $m = 2$ and b for $m = 4$ is reported. Textual features are extracted from FLIP's text encoder. Underline denotes the best performance among image-only models while **bold** denotes the best overall performance.

VICTOR	**FLIP Model**	**MAE (↓)**		**Exact Match (↑)**		**Accuracy (↑)**		***OC_b* AUC (↑)**	
		m=2	*m*=4	*m*=2	*m*=4	*m*=2	*m*=4	*m*=2	*m*=4
VICTOR[OC_r]	ResNet18	0.254	0.221	-	-	-	-	0.90	0.90
	EfficientNetV2-B3	0.255	0.226	-	-	-	-	0.91	0.88
	MLP-Mixer B/16	0.255	0.229	-	-	-	-	0.89	0.86
	ViT B/32	0.254	0.225	-	-	-	-	0.92	0.92
VICTOR[MID]	ResNet18	-	-	38.30	26.29	68.64	69.44	0.89	0.90
	EfficientNetV2-B3	-	-	38.50	27.52	71.99	70.29	0.91	0.90
	MLP-Mixer B/16	-	-	36.70	27.03	72.42	70.44	0.90	0.90
	ViT B/32	-	-	37.69	27.85	72.79	70.59	0.91	0.90
VICTOR[MTL]	ResNet18 (α = 0.2|0.2)	0.257	0.224	40.70	26.02	69.35	65.75	0.90	0.90
	EfficientNetV2-B3 (α = 0.2|1)	0.248	0.216	39.70	26.98	70.24	69.79	0.91	0.91
	MLP-Mixer B/16 (α = 0.2|0.2)	0.247	0.222	**41.55**	26.15	70.57	68.98	0.92	0.91
	ViT B/32 (α = 0.2|1)	0.250	0.214	40.65	27.91	70.38	70.68	0.92	0.92
	Text (α = 0.2|0.2)	0.293	0.243	33.91	20.94	63.98	62.15	0.80	0.80
	Text + ResNet18 (α = 0.2|0.2)	0.238	0.212	39.09	27.53	72.23	70.01	0.92	**0.93**
	Text + EfficientNetV2-B3 (α = 0.2|1)	0.248	0.221	38.60	23.68	70.44	66.34	0.91	0.90
	Text + MLP-Mixer B/16 (α = 0.2|0.2)	0.238	0.212	41.09	**28.97**	73.00	70.90	**0.93**	**0.93**
	Text + ViT B/32 (α = 0.2|1)	**0.230**	**0.204**	39.80	28.00	**73.29**	**71.28**	**0.93**	**0.93**

[5] https://github.com/facebookresearch/fvcore

5.2. Ablation Analysis

We perform an ablation analysis comparing the proposed multi-tasking VICTOR[MTL] with its two separate components, VICTOR[OC_r] and VICTOR[MID]. The results are shown in Table 3. For VICTOR we tune the α hyper-parameter - the weight that combines the two loss functions - and report the best performing based on TOPSIS which takes into account *MAE*, exact match and binary accuracy. VICTOR[OC_r] is only trained for compatibility prediction as regression and can not detect specific mismatching items. Being specialised on OC_r, it yields an average *MAE* of 0.255 for $m = 2$ and 0.225 for $m = 4$. VICTOR[MTL] performs marginally better with 0.250 for $m = 2$ and 0.219 for $m = 4$. The overall lowest, hence better, MAE scores are reached by VICTOR[MTL] with MLP-Mixer B/16 and ViT B/32 for $m = 2$ and $m = 4$ respectively.

VICTOR[MID] is trained on predicting mismatching items in outfits and yields on average a *binary accuracy* of 71.46% for $m = 2$ and 70.19% for $m = 4$, closely followed by VICTOR[MTL] which has 70.14% and 68.8% respectively. In terms of the *exact match* evaluation metric, the strictest evaluation metric for the MID task, we observe that VICTOR[MTL] significantly outperform VICTOR[MID] with 40.65% compared to 37.80% for $m = 2$ while they both perform similarly for $m = 2$, with 26.77% and 27.17% accordingly. The overall highest, hence better, exact match scores are reached by VICTOR[MTL] with MLP-Mixer B/16 and ViT B/32 for $m = 2$ and $m = 4$ respectively. Regarding binary outfit compatibility prediction (OC_b), which is evaluated in terms of *AUC*, we observe that VICTOR[MTL], with 0.91/0.91 AUC on average for $m = 2/m = 4$ respectively, slightly outperforming VICTOR[OC_r]: 0.91/0.89 and MID-only Transformer: 0.90/0.90. ViT B/32 reaches the highest OC_b AUC (0.92) for both m=2 and m=4 with either VICTOR[OC_r] or VICTOR[MTL].

Based on TOPSIS, the overall best performance, among image-only models, is reached by VICTOR[MTL;$\alpha = 0.2$;$m = 2$] with visual features from MLP-Mixer B/16. We observe that combining OC_r and MID in one model and tuning the hyper-parameter α, consistently performs well on both tasks with all computer vision models. Presumably, by addressing two closely related phenomena and tasks simultaneously and from different perspectives, the multi-tasking VICTOR learns to better recognize compatibility relations among various garments.

Finally, Table 3 also illustrates the performance of VICTOR[MTL] with different features as input, namely: text-only, image-only and the combination of text and image. We observe that multi-modality, combining textual and visual features, consistently exceeds the text-only and image-only versions of VICTOR. Based on TOPSIS, the overall best performance is reached by VICTOR[MTL;$\alpha = 0.2$;$m = 2$] with visual features from ViT B/32 and textual features from FLIP's text encoder.

5.3. Comparative Analysis

The central focus of this study is the detection of mismatching items in outfits which can be considered a sub-task of visual compatibility. However, to the best of the authors knowledge, no previous works have addressed these tasks and there are no available models to compare VICTOR with. Instead, we compare the proposed VICTOR[MTL] with numerous state-of-the-art (SotA) models for binary outfit compatibility prediction (OC_b) which is closely related to OC_r. The current SotA for visual-based OC_b on the Polyvore dataset is held by CSA-Net [10] and OutfitTransformer [11] with 0.91 AUC. When category information are added the performance of OutfitTransformer increases to 0.92 and when texts are also added it yields 0.93 AUC.

Comparing the models that use pre-trained visual features on ImageNet, we observe that OutfitTransformer w/ ResNet18 (ImageNet) yields 0.82 AUC on Polyvore while our VICTOR[OC_b] w/ ResNet18 (ImageNet) outperforms it with 0.86 AUC. VICTOR[OC_b] exhibit a similar performance with the other three computer vision models, with an average AUC of 0.86. Furthermore, when employing $\mathbf{v}_{g_i}$ from FLIP, VICTOR[OC_b] w/ ResNet18 (FLIP) improves to 0.9 AUC similarly with all other computer vision models; that display an average AUC of 0.91 for Polyvore and 0.86 on Polyvore-D. The proposed VICTOR[MTL;α =

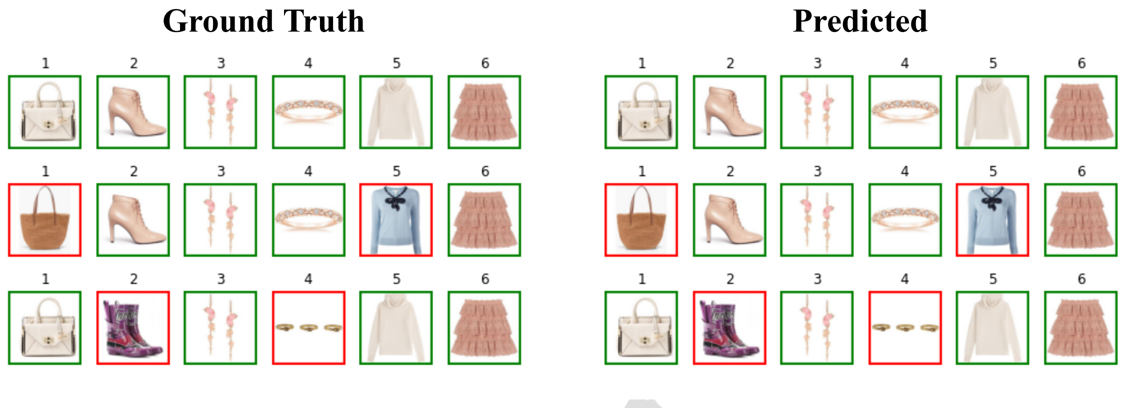
Table 4: Comparison with the state-of-the-art on binary outfit compatibility prediction (OC_b) in terms of AUC. For VICTOR, we report the best performing hyper-parameter combination. Underline denotes the best performance among image-only models while **bold** denotes the best overall performance.

Method	Input	Polyvore	Polyvore-D
BiLSTM + VSE [9]	ResNet18 (E2E) + Text	0.65	0.62
GCN (k=1) [30]	ResNet18 (E2E)	0.82	0.87
Li et al. [30]	ResNet18 (E2E)	0.90	0.85
SiameseNet [2]	ResNet18 (E2E)	0.81	0.81
Type-aware [2]	ResNet18 (E2E) + Text	0.86	0.84
SCE-Net [1]	ResNet18 (E2E) + Text	0.91	-
CSA-Net [10]	ResNet18 (E2E)	0.91	0.87
OutfitTranformer [11]	ResNet18 (ImageNet)	0.82	-
OutfitTranformer [11]	ResNet18 (E2E)	0.91	-
OutfitTranformer [11]	ResNet18 (E2E) + Text	**0.93**	0.88
VICTOR[OC_b]	ResNet18 (ImageNet)	0.86	0.78
	EfficientNetV2-B3 (ImageNet)	0.86	0.78
	MLP-Mixer B/16 (ImageNet)	0.84	0.73
	ViT B/32 (ImageNet)	0.86	0.80
	ResNet18 (FLIP)	0.90	0.85
	EfficientNetV2-B3 (FLIP)	0.91	0.86
	MLP-Mixer B/16 (FLIP)	0.91	0.86
	ViT B/32 (FLIP)	0.91	0.87
VICTOR[MTL]	ResNet18 (ImageNet)	0.86	0.77
	EfficientNetV2-B3 (ImageNet)	0.86	0.79
	MLP-Mixer B/16 (ImageNet)	0.84	0.71
	ViT B/32 (ImageNet)	0.86	0.78
	ResNet18 (FLIP)	0.90	0.85
	EfficientNetV2-B3 (FLIP)	0.91	0.87
	MLP-Mixer B/16 (FLIP)	0.92	0.87
	ViT B/32 (FLIP)	0.92	0.88
	ResNet18 + Text (FLIP)	**0.93**	0.88
	EfficientNetV2-B3 + Text (FLIP)	0.91	0.88
	MLP-Mixer B/16 + Text (FLIP)	**0.93**	0.89
	ViT B/32 + Text (FLIP)	**0.93**	**0.90**

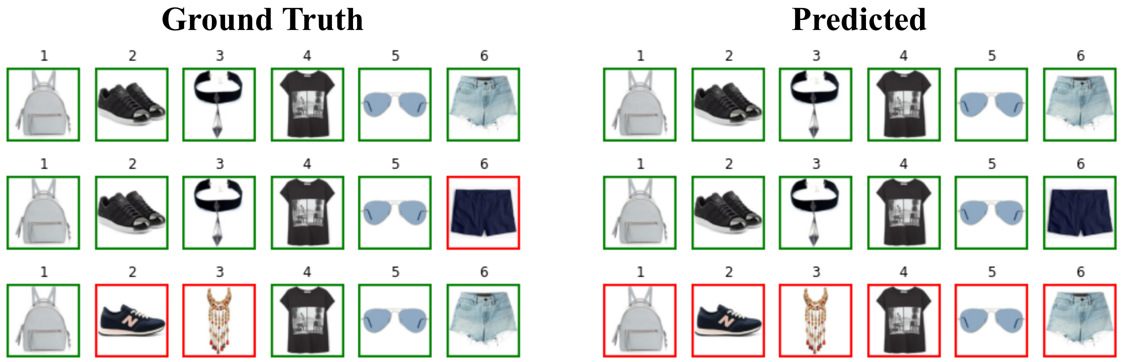
0.2;m = 2] further improves upon VICTOR[OC_b] with MLP-Mixer B/16 and ViT B/32 FLIP models. This slight improvement can be attributed to Y_{OCr} forcing VICTOR[MTL] to learn deeper and more complicated relations compared to the simple OC_b-based model. Notably, VICTOR[MTL] w/ ResNet18 (FLIP) performs at the same level as the SotA while being significantly faster and less resource-intensive to train; requiring 94.8% fewer FLOPs. VICTOR[MTL] w/ MLP-Mixer B/16 (FLIP) or ViT B/32 (FLIP) surpasses the vision-based SotA with 0.92 AUC on Polyvore while VICTOR[MTL] with ViT/B32 (FLIP) surpasses the SotA on Polyvore-D with 0.88 AUC.

When textual features are added, we observe that VICTOR[MTL] with ResNet18 exhibits the same AUC (0.93) as OutfitTransformer on the Polyvore dataset. Moreover, VICTOR[MTL] with either MLP-Mixer B/16 and ViT B/32 surpass OutfitTransformer on the Polyvore-D dataset with 0.89 and 0.90 AUC scores respectively without requiring end-to-end fine-tuning. Therefore, the multi-modal VICTOR[MTL] has defined a new SotA score on Polyvore-D - the more challenging version of Polyvore - while maintaining very high efficiency. In order to verify this result, we design a replication experiment with VICTOR[MTL] utilizing textual and visual features from FLIP's ViT B/32. More specifically, we alter the Pytorch random seed between 0 or 1, alter the a parameter between 0.2 or 1 and alter the m parameter between 2 or 4; which translates into a total of 8 experiments. The experiment resulted in a highest score of 0.901[6], a lowest score of 0.889 and a mean value of 0.894 with standard deviation of 0.003.This translates into VICTOR surpassing the previous SotA with a relative improvement between 1.02% and 2.38% (lowest to highest VICTOR performance). This means that our method consistently outperforms the previous SotA on Polyvore-D.

5.4. Qualitative Analysis



(a)



(b)

Fig. 5: Inference examples from VICTOR on fully compatible outfits and their generated partially mismatching versions. Green frames denotes compatible items while red frames denote incompatible items.

Fig.5 illustrates two inference examples from VICTOR[MTL;$\alpha = 0.2$;$m = 2$] with MLP-Mixer B/16 since it exhibited the highest exact match score. We use samples from the Polyvore MISFITs $m = 2$ thus there are three fully compatible outfits and for each, there are two generated outfits containing at least one incompatible item. We observe that VICTOR is capable of correctly identifying the fully compatible outfits in both cases (row=1 of each outfit). There are also cases that correctly identifies all mismatching items,such as row 2 and 3 of Fig. 5a. VICTOR has presumably learned to “understand” which styles and colors of different garments can be matched together.

[6] In Table 4 we report the highest AUC score obtained after this experimentation, as hyper-parameter tuning is involved on top of minimal random seed selection.

Given the fact that VICTOR[MTL;$\alpha = 0.2$;$m = 2$] scores 41.55% in terms of exact match and 70.57% in terms of accuracy on Polvore-MISFITs, it is expected to also have some mistaken predictions. Row 3 of Fig. 5b the whole outfit is predicted to be incompatible while 4/6 items are annotated as compatible. As discussed in section 4.2, the number of mismatching items is balanced against the fully matching and fully mismatching outfits in Polyvore-MISFITs with $m = 2$. However, this being a regression problem, each individual score (e.g 4/6 in this case) is not balanced with each other; as illustrated in Fig. 3. This imbalance could potentially skew certain marginal cases towards the ends of the distribution; towards fully compatible or fully incompatible. This could also explain row 2 of Fig.5b where the pair of navy shorts is predicted to be matching despite being annotated as mismatching with the rest of the outfit; rendering the whole outfit to be fully compatible. On the other hand, some would consider this to be a mistaken annotation since gray, black and navy are often paired together. In that case, VICTOR would have learned to generalize well enough so as to ignore the few rare cases of annotations that match by mistake.

One general challenge for the task of visual compatibility is that there is always the element of subjectivity. Moreover, what is considered compatible differs from culture to culture and is time dependent; since fashion trends are in constant flux. In our case, the ground truth compatible outfits reflect the subjective opinions and biases of fashion stylists from Polyvore, creating a data-driven bias in our models. Despite this caveat, overall, VICTOR seems to produce reasonable predictions and we believe that a larger and more diverse dataset would further improve its performance.

Finally, VICTOR does not only predict the mismatching items in an outfit but has also learned to predict the overall compatibility of an outfit. As a result, it can also be used for outfit recommendation. Fig. 6 illustrates an example where VICTOR detects two mismatching items in an outfit and given a set of candidate garments, it selects the better suited alternatives, resulting in a more cohesive and aesthetically pleasing outfit. As candidates for replacing a mismatching item we use all available garments of the same category in our database (e.g. Polyvore dataset) and retrieve the item that results in the highest overall compatibility when added to the outfit.

6. Conclusions

In this study we define two new sub-tasks within the general task of visual compatibility prediction, namely compatibility prediction as regression (OC_r) and mismatching item detection (MID) and examine both in the Fashion domain. We use the Polyvore outfits dataset to generate partially mismatching outfits (MISFITs) and create the Polyvore-MISFITs dataset where we perform a series of ablative and comparative experiments. We propose a multi-tasking Transformer-based architecture, named VICTOR, and utilize visual features from multiple computer vision neural networks fine-tuned with *fashion-specific contrastive language-image pre-training* (FLIP).

The ablation study showed that addressing both tasks (OC_r and MID) in a single architecture performs better than single-task models and that multi-modality, combining both textual and visual features from FLIP, outperforms single-modality models. Furthermore, in the comparative analysis, VICTOR outperformed state-of-the-art models by 4.87% in terms of AUC on the Polyvore dataset when using visual features extracted from ImageNet-pretrained models; with no additional computational cost. By utilizing features from FLIP, VICTOR was not only capable of competing and even surpassing state-of-the-art methods on Polyvore datasets, but also to reduce instance-wise floating point operations by 88%. Notably, VICTOR improved upon the previous SotA on the Polyvore-Disjoint dataset by 2.38%.

One limitation of the current study is that when generating the Polyvore-MISFITs dataset, we use random alternative sampling. More intricate methods could theoretically be implemented, that take into account the rate of similarity between the ground truth

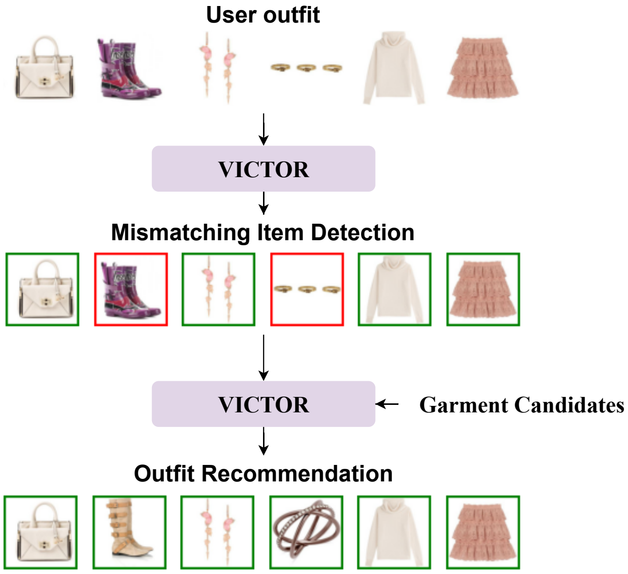


Fig. 6: Example of VICTOR detecting the mismatching items in an outfit and recommending more compatible alternatives. Green frames denotes compatible items while red frames denote incompatible items.

and the selected mismatching garment. However, it is difficult to select the appropriate threshold of similarity without input from professional stylists. Selecting too similar items - e.g a black pair of dress shoes with another - would not result in actual mismatching outfits. Conversely, selecting too dissimilar items - e.g dress shoes with a pair of snow boots - would lead to numerous easy-to-predict MISFITs and as a result, VICTOR would not have learned to discern more subtle cases of visual incompatibility. Secondly, as the target for the regression task, we define the ratio of mismatching items to all items in an outfit. However, this is not the only possible definition. For example, some may consider that a single or few completely irrelevant items could 'ruin' an entire outfit and that the whole outfit should be scored lower than the ratio of mismatching items. Such a definition would require feedback or manual annotation from fashion experts; which would be expensive, time consuming and would introduce a level of subjectivity. For these reasons we decided to proceed with the more straight-forward definition. Future works could expand upon this definition for the regression task and experiment with different methods of generating mismatching items. Another issue is that VICTOR has been trained on images from Polyvore dataset which depicts individual garments in a white background. This may limit its application to real-world fashion images worn by people "in the wild". However, VICTOR could very easily be integrated in a full system, similar to [14], that applies garment detection to real world fashion imagery and then extract the visual features of individual garments; given that the garments are fully or mostly visible. Finally, because our focus is centered on OC_r and MID tasks in this paper, we do not experiment with complementary item retrieval like [11] nor 'fill-in-the-blank' like [2]. However, VICTOR could be adjusted to accommodate both tasks. Moreover, methods like deep multi-view hashing [31] could be utilized in order to improve the efficiency of complementary item retrieval.

Our focus is centered around general visual compatibility in fashion. By relying on the Polyvore dataset VICTOR has learned to reflect the subjective opinions and biases of fashion stylists from Polyvore. It would be interesting for future works to re-create similar architectures that also take personalization into account [5]. Finally, the proposed VICTOR and FLIP fine-tuning are not limited to applications within the Fashion domain. Future works could experiment with other visually-driven domains such as exterior and interior architecture design [32].

7. Declarations

Competing interests. The authors have no competing interests to declare that are relevant to the content of this article.

Funding. This work is partially funded by the Horizon 2020 European project "eTryOn - virtual try-ons of garments enabling novel human fashion interactions" under grant agreement no. 951908.

Data availability. The Polyvore dataset is publicly available and has been properly referenced in the text. The code for generating the Polyvore-MISFITs dataset is also provided.

8. Author contributions

Stefanos-Iordanis Papadopoulos: Conceptualization, Methodology, Software, Data Curation, Investigation, Formal analysis, Validation, Writing - Original Draft, Visualization. **Christos Koutlis**: Methodology, Validation, Writing - Original Draft, Supervision. **Symeon Papadopoulos**: Writing - Review & Editing, Supervision, Project administration, Funding acquisition. **Ioannis Kompatsiaris**: Project administration, Funding acquisition.

References

[1] R. Tan, M. I. Vasileva, K. Saenko, B. A. Plummer, Learning similarity conditions without explicit supervision, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 10373–10382.

[2] M. I. Vasileva, B. A. Plummer, K. Dusad, S. Rajpal, R. Kumar, D. Forsyth, Learning type-aware embeddings for fashion compatibility, in: Proceedings of the European conference on computer vision (ECCV), 2018, pp. 390–405.

[3] G. Cucurull, P. Taslakian, D. Vazquez, Context-aware visual compatibility prediction, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2019, pp. 12617–12626.

[4] Z. Cui, Z. Li, S. Wu, X.-Y. Zhang, L. Wang, Dressing as a whole: Outfit compatibility learning based on node-wise graph neural networks, in: The World Wide Web Conference, 2019, pp. 307–317. doi:10.1145/3308558.3313444.

[5] H. Zhan, J. Lin, Pan: Personalized attention network for outfit recommendation, in: 2021 IEEE International Conference on Image Processing (ICIP), IEEE, 2021, pp. 2663–2667. doi:10.1109/ICIP42928.2021.9506344.

[6] H. Zhan, J. Lin, K. E. Ak, B. Shi, L.-Y. Duan, A. C. Kot, a^3-fkg: Attentive attribute-aware fashion knowledge graph for outfit preference prediction, IEEE Transactions on Multimedia 24 (2021) 819–831.

[7] W. Chen, P. Huang, J. Xu, X. Guo, C. Guo, F. Sun, C. Li, A. Pfadler, H. Zhao, B. Zhao, Pog: personalized outfit generation for fashion recommendation at alibaba ifashion, in: Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining, 2019, pp. 2662–2670. doi:10.1145/3292500.3330652.

[8] A. Lorbert, D. Neiman, A. Poznanski, E. Oks, L. Davis, Scalable and explainable outfit generation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 3931–3934.

[9] X. Han, Z. Wu, Y.-G. Jiang, L. S. Davis, Learning fashion compatibility with bidirectional lstms, in: Proceedings of the 25th ACM international conference on Multimedia, 2017, pp. 1078–1086. doi:10.1145/3123266.3123394.

[10] Y.-L. Lin, S. Tran, L. S. Davis, Fashion outfit complementary item retrieval, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 3311–3319.

[11] R. Sarkar, N. Bodla, M. Vasileva, Y.-L. Lin, A. Beniwal, A. Lu, G. Medioni, Outfittransformer: Outfit representations for fashion recommendation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 2263–2267.

[12] W.-H. Cheng, S. Song, C.-Y. Chen, S. C. Hidayati, J. Liu, Fashion meets computer vision: A survey, ACM Computing Surveys (CSUR) 54 (2021) 1–41.

[13] Z. Liu, P. Luo, S. Qiu, X. Wang, X. Tang, Deepfashion: Powering robust clothes recognition and retrieval with rich annotations, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 1096–1104.

[14] S.-I. Papadopoulos, C. Koutlis, M. Sudheer, M. Pugliese, D. Rabiller, S. Papadopoulos, I. Kompatsiaris, Attentive hierarchical label sharing for enhanced garment and attribute classification of fashion imagery, in: Recommender Systems in Fashion and Retail, Springer, 2022, pp. 95–115. doi:10.1007/978-3-030-94016-4_7.

[15] Z. Al-Halah, R. Stiefelhagen, K. Grauman, Fashion forward: Forecasting visual style in fashion, in: Proceedings of the IEEE international conference on computer vision, 2017, pp. 388–397.

[16] U. Mall, K. Matzen, B. Hariharan, N. Snavely, K. Bala, Geostyle: Discovering fashion trends and events, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 411–420. doi:10.1109/ICCV.2019.00050.

[17] G. Skenderi, C. Joppi, M. Denitto, M. Cristani, Well googled is half done: Multimodal forecasting of new fashion product sales with image-based google trends, arXiv preprint arXiv:2109.09824 (2021).

[18] S. I. Papadopoulos, C. Koutlis, S. Papadopoulos, I. Kompatsiaris, Multimodal quasi-autoregression: Forecasting the visual popularity of new fashion products, arXiv preprint arXiv:2204.04014 (2022).

[19] H. Hwangbo, Y. S. Kim, K. J. Cha, Recommendation system development for fashion retail e-commerce, Electronic Commerce Research and Applications 28 (2018) 94–101.

[20] M. A. Stefani, V. Stefanis, J. Garofalakis, Cfrs: a trends-driven collaborative fashion recommendation system, in: 2019 10th International Conference on Information, Intelligence, Systems and Applications (IISA), IEEE, 2019, pp. 1–4. doi:10.1109/IISA.2019.8900681.

[21] A. Veit, B. Kovacs, S. Bell, J. McAuley, K. Bala, S. Belongie, Learning visual clothing style with heterogeneous dyadic co-occurrences, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 4642–4650.

[22] M. Taraviya, A. Beniwal, Y.-L. Lin, L. Davis, Personalized compatibility metric learning, in: KDD Workshop, volume 1, 2021.

[23] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Advances in neural information processing systems 30 (2017).

[24] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al., An image is worth 16x16 words: Transformers for image recognition at scale, arXiv preprint arXiv:2010.11929 (2020).

[25] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al., Learning transferable visual models from natural language supervision, in: International Conference on Machine Learning, PMLR, 2021, pp. 8748–8763.

[26] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.

[27] M. Tan, Q. Le, Efficientnetv2: Smaller models and faster training, in: International Conference on Machine Learning, PMLR, 2021, pp. 10096–10106.

[28] I. O. Tolstikhin, N. Houlsby, A. Kolesnikov, L. Beyer, X. Zhai, T. Unterthiner, J. Yung, A. Steiner, D. Keysers, J. Uszkoreit, et al., Mlp-mixer: An all-mlp architecture for vision, Advances in Neural Information Processing Systems 34 (2021) 24261–24272.

[29] C.-L. Hwang, K. Yoon, Methods for multiple attribute decision making, in: Multiple attribute decision making, Springer, 1981, pp. 58–191. doi:10.1007/978-3-642-48318-9_3.

[30] K. Li, C. Liu, D. Forsyth, Coherent and controllable outfit generation, arXiv preprint arXiv:1906.07273 (2019).

[31] C. Yan, B. Gong, Y. Wei, Y. Gao, Deep multi-view enhancement hashing for image retrieval, IEEE Transactions on Pattern Analysis and Machine Intelligence 43 (2020) 1445–1451.

[32] D. Aggarwal, E. Valiyev, F. Sener, A. Yao, Learning style compatibility for furniture, in: German Conference on Pattern Recognition, Springer, 2018, pp. 552–566. doi:10.1007/978-3-030-12939-2_38.

Highlights

- Addressing outfit compatibility as regression and mismatching item detection
- Creation of the Polyvore-MISFITs dataset
- Proposal of fashion-specific contrastive image-language pre-training
- Reduction of the required computational resources by 88%
- Surpassing the state-of-the-art by 2.38% on the Polyvore-Disjoint dataset

Declaration of interests

☒ The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

☐ The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: