

# SEMI-SUPERVISED COMPATIBILITY LEARNING ACROSS CATEGORIES FOR CLOTHING MATCHING

Zekun Li<sup>1,3</sup>, Zeyu Cui<sup>2,3</sup>, Shu Wu<sup>2,3,4</sup>, Xiaoyu Zhang<sup>1</sup> and Liang Wang<sup>2,3</sup>

<sup>1</sup>Institute of Information Engineering, Chinese Academy of Sciences

<sup>2</sup>Institute of Automation, Chinese Academy of Sciences

<sup>3</sup>University of Chinese Academy of Sciences

<sup>4</sup>Jiaozhou Artificial Intelligence Research, Chinese Academy of Sciences

lizekunlee@gmail.com, {zeyu.cui, shu.wu}@nlpr.ia.ac.cn,

zhangxiaoyu@iie.ac.cn, wangliang@nlpr.ia.ac.cn

## ABSTRACT

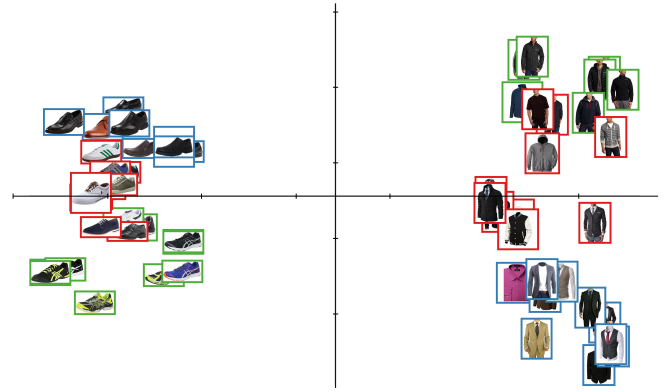
Learning the compatibility between fashion items across categories is a key task in fashion analysis, which can decode the secret of clothing matching. The main idea of this task is to map items into a latent style space where compatible items stay close. Previous works try to build such a transformation by minimizing the distances between annotated compatible items, which require massive item-level supervision. However, these annotated data are expensive to obtain and hard to cover the numerous items with various styles in real applications. In such cases, these supervised methods fail to achieve satisfactory performances. In this work, we propose a semi-supervised method to learn the compatibility across categories. We observe that the distributions of different categories have intrinsic similar structures. Accordingly, the better distributions align, the closer compatible items across these categories become. To achieve the alignment, we minimize the distances between distributions with unsupervised adversarial learning, and also the distances between some annotated compatible items which play the role of *anchor points* to help align. Experimental results on two real-world datasets demonstrate the effectiveness of our method.

**Index Terms**— Semi-supervised compatibility learning, Clothing matching, Adversarial learning

## 1. INTRODUCTION

Nowadays, clothes with various styles are increasing quickly, broadening the range of people’s choices. “Which pair of shoes should I select to match the jeans?”, such a problem has become a daily headache for many people. Solving this problem requires learning the compatibility between fashion items across categories. As a matter of fact, many existing

The first two authors Zekun Li and Zeyu Cui are listed as joint first authors. Shu Wu and Xiaoyu Zhang are both corresponding authors.



**Fig. 1.** 2-D visualization of the distributions of tops and shoes, which have similar structures despite different orientations and scales. The tops and shoes in frames with the same color have same styles and are compatible.

efforts have been dedicated to the task of compatibility learning. The key idea is to map the items into a latent style space where compatible items would stay close. Previous works [1, 2] try to learn the transformation by minimizing the distance between annotated compatible items in the style space, which requires massive supervision to be general. However, these annotated data are expensive to obtain and hard to cover the numerous and increasing clothing items in real applications, and so these supervised methods often fail to achieve satisfactory performance. How to learn such a general transformation for numerous items of various styles with a limited amount of supervision has become a demanding problem.

The key to this task is to make compatible items close in the learned style space. Previous works consider it only at the item level (i.e., minimize the distances between annotated compatible items), which inevitably require enough item-level supervision to be general. In fact, some information at a higher level is ignored. Here we look into the distribution level. We randomly select some tops and shoes and visualize their distributions respectively in Figure 1, with raw

image features reduced to 2-D vectors by Principal Component Analysis (PCA). Different colors of frames represent different styles. The tops and shoes in frames of the same color have same styles and are compatible. As can be seen, in spite of different orientations and scales, these distributions have intrinsic similar structures according to styles. Accordingly, there may exist a desired transformation to align the distributions of different categories in the style space, which can make compatible items with same styles close in return.

In fact, the method of distribution alignment has been proven to be effective on the bilingual lexicon induction task in the domain of natural language processing (NLP), which is highly analogous to our task. The studies on the bilingual lexicon induction task follow the idea to map words into a word embedding space. There is evidence that different languages represent semantic concepts with similar structure leading to the structural isomorphism across word embedding spaces of different languages [3]. Viewing word embedding spaces as distributions, Zhang *et al.* proposed to build the cross-lingual connection by minimizing their earth movers distance [4]. Differently, they achieve the alignment only by minimizing the distance between distributions while we find the combination of distribution-level and item-level distance minimization can better align the distributions.

In this paper, we propose Semi-supervised Compatibility learning Generative Adversarial Networks (**SC-GANs**) to learn the compatibility across categories, not requiring massive supervision. In our semi-supervised method, the unsupervised distribution-level distance minimization is combined with the supervised item-level one to build a general transformation, which can align the distributions of different categories and make compatible items with same styles close in the style space. An adversarial learning strategy is adopted to minimize both the Wasserstein distance between distributions and Euclidean distance between compatible items. We conduct experiments to evaluate the performance of our method on two real-world datasets. Our semi-supervised method shows superiority over other supervised methods when lacking massive supervision, which is effective in real applications. The code and data has been released<sup>1</sup>.

Our main contributions can be summarized in threefold:

- We first propose that aligning distributions of different categories in the style space can make compatible items with same styles close, which can be achieved by minimizing the distances between distributions and also some annotated compatible items.
- We propose a semi-supervised method **SC-GANs** to learn compatibility across categories not requiring massive supervision, which is potential in real applications.
- Experimental results on two real-world datasets demonstrate the effectiveness of our proposed method.

## 2. THE MODEL

The whole item set is denoted as  $\mathcal{I}$  while the set of category is  $\mathcal{C} = \{c_1, c_2, \dots\}$ . The set of items in category  $c_i$  is  $\mathcal{I}_{c_i}$ . The set of compatible pairs is  $\mathcal{P}$ . The compatibility of two items  $x \in \mathcal{I}_{c_j}, y \in \mathcal{I}_{c_j}$ , is  $r(x, y)$ . Our goal is to estimate the value of  $r(x, y)$ .  $\mathbf{v}_x$  is a high-dimensional feature vector of item  $x$  extracted from its image. The distribution of category  $c$  in the style space is  $\mathbb{P}_c$ . We aim to find the exact style transformation to map items (their feature vectors) of different categories into one style space, where the distributions of different categories align and the compatible items are close as well. In the style space, the distance between items  $x$  and  $y$  is  $d(x, y)$ , which can indicate  $r(x, y)$ . The lower  $d(x, y)$ , the higher  $r(x, y)$  is, and the more compatible  $x$  and  $y$  are.

### 2.1. Preliminaries

**Feature Extraction.** The visual features of items are extracted from their images using deep convolution networks, VGG-16 [5], which is widely used for image representation learning [1, 2, 6]. It has been pre-trained on large-scale ImageNet images. We adopt the output of the second fully connected layer, a 4096-dimensional feature vector.

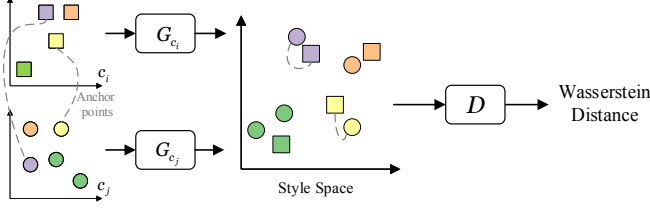
**Style Space Transformation.** Compatible items usually have similar styles. Previous works assume that there exists a style space where compatible items stay close. Veit *et al.* [6] use Siamese CNNs to learn a feature transformation from the image space to the style space. McAuley *et al.* [1] use Low-rank Mahalanobis Transformation (LMT) to map compatible items to close positions in the style space. He *et al.* [2] map items into several style spaces to compute a weighted sum of the  $K$  distances between two items, which can deal with diversity across different query items. Liu *et al.* [7] proposed to map items into a style space where the categorical information are eliminated. The above methods build such a transformation only by minimizing the distance between annotated compatible items. Nevertheless, we build such a transformation by aligning the distributions, i.e., minimizing the distance between the distributions of different categories as well as annotated compatible items.

**Wasserstein Distance.** In this work, we adopt the Wasserstein distance as the measure of distance between distributions. Wasserstein distance is a measure of distance between probability distributions, which can be formulated as,

$$W(\mathbb{P}_1, \mathbb{P}_2) = \inf_{\gamma \in \Gamma(\mathbb{P}_1, \mathbb{P}_2)} \mathbb{E}_{(x, y) \sim \gamma} [c(x, y)], \quad (1)$$

where  $\Gamma(\mathbb{P}_1, \mathbb{P}_2)$  denotes the set of all joint distributions  $\gamma(x, y)$  with marginals  $\mathbb{P}_1$  and  $\mathbb{P}_2$ . It can be considered as the continuous case of the Earth Mover’s Distance, a powerful tool widely used in computer vision and natural language processing [4, 8]. Intuitively, if each distribution is viewed as a unit amount of “dirt”, earth mover’s distance is the minimum “cost” of turning one pile into the other, which is assumed to

<sup>1</sup><https://github.com/CRIPAC-DIG/SCGAN>



**Fig. 2.** The framework of SC-GANs. Items of each category (we only show two categories here briefly) are mapped into one style space with category-specific generators. The critic  $D$  estimates the Wasserstein distance, which will be passed to the generators and guide them towards minimizing the Wasserstein estimate. The generators also try to minimize the distance between annotated compatible items, which play the role of *anchor points* to help align.

be the amount of dirt that needs to be moved multiplies the distance it has to be moved. This conforms to the nature of our task, i.e., put an item close to its compatible item of another category in the latent style space. In order to minimize the Wasserstein distance between distributions, we adopt an adversarial learning strategy similar with WGANs.

**Wasserstein GANs.** Goodfellow et al. [9] first propose GANs to generate distribution similar with the target distribution. But the original GANs are difficult to train. Many efforts have been devoted to solve the problem. Arjovsky *et al.* propose WGANs [10] to deal with this training problem. WGANs can be viewed as an adversarial game to minimize the Wasserstein distance between the generated distribution  $\mathbb{P}_g$  and the real distribution  $\mathbb{P}_r$ . With the ground distance  $c$  being the Euclidean distance L2, Eq.(1) can be cast to the following equation according to the Kantorovich-Rubinstein duality,

$$W(\mathbb{P}_g, \mathbb{P}_r) = \frac{1}{K} \sup_{\|f\|_L \leq K} \mathbb{E}_{x \sim \mathbb{P}_g}[f(x)] - \mathbb{E}_{x \sim \mathbb{P}_r}[f(x)], \quad (2)$$

where the supremum is over all  $K$ -Lipschitz functions  $f$ .

WGANs consist of two components, critic  $D$  and generator  $G$ . The critic  $D$  is a neural network to approximate  $f$  in Eq.(2) with weight clipping to ensure that the function family is  $K$ -Lipschitz. It manages to distinguish the critic real distribution and the generated distribution, so as to maximize their Wasserstein distance. The objective of  $D$  is

$$\max_D \mathbb{E}_{x \sim \mathbb{P}_r}[D(x)] - \mathbb{E}_{x \sim \mathbb{P}_g}[D(x)]. \quad (3)$$

When the objective (3) is trained until optimality, it approximates the Wasserstein distance. The generator  $G$  then aims to minimize the approximate Wasserstein distance, which leads to

$$\min_G -\mathbb{E}_{x \sim \mathbb{P}_g}[D(x)] \quad (4)$$

## 2.2. Model Architecture

The framework of our model is shown in Figure 2. Our model consists of two components, generator and critic. The part of generator plays the role of style space transformation to map items into the latent style space. The part of critic estimates the Wasserstein distance between the transformed distributions. In our model, we have one critic  $D$  and category-specific generators  $G = \{G_{c_1}, G_{c_2}, \dots\}$ , since each category has its own unique characteristics. Accordingly, the style vector  $\mathbf{s}_x$  of item  $x$  in the style space can be calculated as,

$$\mathbf{s}_x = \mathbf{G}_{c_i} \mathbf{v}_x, \quad (5)$$

where  $x \in \mathcal{I}_{c_i}$ ,  $\mathbf{G}_{c_i}$  is the transformation matrix corresponding to the category  $c_i$ .

**Distribution-level Distance Minimization.** This is an **unsupervised** part, not requiring supervision of compatible pairs. Different from WGANs minimizing the Wasserstein distance between the generated distribution and real distribution, SC-GANs minimize the Wasserstein distance between dyadic transformed category distributions in the style space as shown in Figure 2. We here take two categories  $c_i, c_j$  for example. The Wasserstein distance between transformed distribution in the latent style space  $\mathbb{P}_{c_i}$  and  $\mathbb{P}_{c_j}$  of categories  $c_i, c_j$  can be estimated as:

$$W(\mathbb{P}_{c_i}, \mathbb{P}_{c_j}) = \frac{1}{K} \sup_{\|f\|_L \leq K} \mathbb{E}_{x \sim \mathcal{I}_{c_i}}[f(\mathbf{G}_{c_i} \mathbf{v}_x)] - \mathbb{E}_{x \sim \mathcal{I}_{c_j}}[f(\mathbf{G}_{c_j} \mathbf{v}_x)]. \quad (6)$$

Following WGANs, we approximate  $f$  with the critic  $D$ , whose objective is,

$$\max_D \mathbb{E}_{x \sim \mathcal{I}_{c_i}}[D(\mathbf{G}_{c_i} \mathbf{v}_x)] - \mathbb{E}_{x \sim \mathcal{I}_{c_j}}[D(\mathbf{G}_{c_j} \mathbf{v}_x)]. \quad (7)$$

When the objective (7) is trained until optimality, it approximates the Wasserstein distance between distributions  $\mathbb{P}_{c_i}$  and  $\mathbb{P}_{c_j}$ . The generators aim to minimize this distance as,

$$\min_{\mathbf{G}_{c_i}, \mathbf{G}_{c_j}} \mathbb{E}_{x \sim \mathcal{I}_{c_i}}[D(\mathbf{G}_{c_i} \mathbf{v}_x)] - \mathbb{E}_{x \sim \mathcal{I}_{c_j}}[D(\mathbf{G}_{c_j} \mathbf{v}_x)]. \quad (8)$$

**Item-level Distance Minimization.** This is a **supervised** part. The generators should also keep the compatible pairs close, which play the role of *anchor points* to help align the distributions. In addition, we impose an orthogonal constraint on the transformation matrices to keep structural information, according to [4]. Overall, we have the following objective for generators:

$$\begin{aligned} & \min_{\mathbf{G}_{c_i}, \mathbf{G}_{c_j}} \mathbb{E}_{x \sim \mathcal{I}_{c_i}}[D(\mathbf{G}_{c_i} \mathbf{v}_x)] - \mathbb{E}_{x \sim \mathcal{I}_{c_j}}[D(\mathbf{G}_{c_j} \mathbf{v}_x)] \\ & + \eta \sum_{x \in \mathcal{I}_{c_i}, y \in \mathcal{I}_{c_j}, (x, y) \in \mathcal{P}} \|\mathbf{G}_{c_i} \mathbf{v}_x - \mathbf{G}_{c_j} \mathbf{v}_y\|_2^2 \\ & + \lambda \|\mathbf{G}_{c_i} \mathbf{G}_{c_i}^T - \mathbf{E}\|_F + \lambda \|\mathbf{G}_{c_j} \mathbf{G}_{c_j}^T - \mathbf{E}\|_F, \end{aligned} \quad (9)$$

---

**Algorithm 1** SC-GANs

---

**Require:**  $G = \{\mathbf{G}_{c_1}, \mathbf{G}_{c_2}, \dots\}$ : generators.  $D$ : critic.  $m$ : batch size.  $l$ : the gradient clip bound.  $\lambda, \eta$ : coefficients.  $n_{critic}$ : the number of critic iterations per generator iteration.

- 1: Randomly initialize  $G$  and  $D$ , set  $t = 0$ ;
- 2: **while**  $(G, D)$  not converged **do**
- 3:   Sample  $c_i, c_j \in \mathcal{C}$ ;  $t \leftarrow t + 1$ ;
- 4:   Sample  $\{x^{(i)}\}_{i=1}^m \in \mathcal{I}_{c_i}$ , a batch of items from  $\mathcal{I}_{c_i}$ ;
- 5:   Sample  $\{y^{(i)}\}_{i=1}^m \in \mathcal{I}_{c_j}$ , a batch of items from  $\mathcal{I}_{c_j}$ ;
- 6:   Sample  $(x, y) \in \mathcal{P}, x \in \mathcal{I}_{c_i}, y \in \mathcal{I}_{c_j}$ , a batch of compatible pairs from  $\mathcal{P}$ ;
- 7:   **if**  $t \bmod n_{critic} = 0$  **then**
- 8:     Update  $\mathbf{G}_{c_i}$  and  $\mathbf{G}_{c_j}$  by descending:  
       $\frac{1}{m} \sum_{i=1}^m D(\mathbf{G}_{c_i} x_i) - \frac{1}{m} \sum_{i=1}^m D(\mathbf{G}_{c_j} y_i)$   
       $+ \eta \sum_{x \in \mathcal{I}_{c_i}, y \in \mathcal{I}_{c_j}, (x, y) \in \mathcal{P}} \|\mathbf{G}_{c_i} \mathbf{v}_x - \mathbf{G}_{c_j} \mathbf{v}_y\|_2^2$   
       $+ \lambda \|\mathbf{G}_{c_i} \mathbf{G}_{c_i}^T - \mathbf{E}\|_F + \lambda \|\mathbf{G}_{c_j} \mathbf{G}_{c_j}^T - \mathbf{E}\|_F$ ;
- 9:     **end if**
- 10:    Update  $D$  by ascending:  
       $\frac{1}{m} \sum_{i=1}^m D(\mathbf{G}_{c_1} x_i) - \frac{1}{m} \sum_{i=1}^m D(\mathbf{G}_{c_2} y_i)$ ;
- 11:     $D \leftarrow clip(D, -l, l)$ ;
- 12: **end while**

---

where  $\lambda, \eta$  are coefficients and  $\mathbf{E}$  is the identity matrix.

**Learning the Model.** The training process of our model is shown in Algorithm 1. Since there are many categories in the training dataset, we only train the critic and two randomly selected category-specific generators each time. The generators and the critic are trained in a settled proportion  $n_{critic}$  (i.e., train the critic  $n_{critic}$  iterations per generator iteration), until the critic and all the generators converge.

**Distance and Compatibility.** After training, the compatible items of different categories are close in the learned style space. Therefore, we can calculate the distance between items  $x \in \mathcal{I}_{c_i}, y \in \mathcal{I}_{c_j}$  in the style space as,

$$d(x, y) = \|\mathbf{G}_{c_i} \mathbf{v}_x - \mathbf{G}_{c_j} \mathbf{v}_y\|_2^2. \quad (10)$$

Following [1], their compatibility is related to the distance as,

$$r(x, y) = \sigma(-d(x, y)) = \frac{1}{1 + e^{d(x, y)}}. \quad (11)$$

### 3. EXPERIMENT

#### 3.1. Datasets

We conduct experiments on two datasets: the Amazon “also-bought” dataset and the Taobao dataset.

**Amazon “also-bought” dataset.** Amazon dataset was collected by McAuley et al. [1]. Following previous work, the “also-bought” relationships are used as compatibility in the five clothing categories, “Women”, “Men”, “Girls”, “Boys” and “Baby”. There are totally 1101118 items from 263 sub-categories and 3457219 relationships covering all the items.

Although it is commonly used by previous works [1, 2, 6], the relationship “also-bought” is not totally equal to compatibility. To compare our method with others in case of lacking enough supervision covering all the items, we randomly select 0.5, 1, 2 permillage of compatible pairs in the Amazon dataset as seeds to form the training dataset and 20% to form the testing set.

**Taobao dataset.** Taobao dataset is a collection of outfits of women clothing on Taobao.com released by Alibaba Group<sup>2</sup>, in which compatibility was manually labelled by fashion experts. The taobao dataset consists of 499983 items from 71 categories. There are 407152 compatibility relationships covering 60767 items. Since the compatibility doesn’t cover all the items, it’s suitable to test the performances of these methods in the real scenario. We thus conduct experiments on the *whole* dataset with 80% for training and 20% for testing. For the two datasets (Amazon and Tabao), we denote the training set as  $\mathcal{P}_{train}$  and testing set as  $\mathcal{P}_{test}$ .

#### 3.2. Compared Methods

**Nearest Neighborhood (NN)** is a traditional unsupervised method. The dimensionality of raw item features is reduced to  $d$  by PCA, and then their Euclidean distance are used to measure the compatibility. **Category Tree (CT)** measures the compatibility between two items using the co-occurrences between their categories. **Low-rank Mahalanobis Transform (LMT)** models the relationships between items in the style space via a single low-rank Mahalanobis embedding matrix [1]. **Mixtures of Non-Metric Embeddings for Recommendation (Monomer)** is proposed by He [2]. This method maps the compatible items into  $K$  latent style spaces to compute weighted sum of the  $K$  distances between the two items. **UC-GANs** is the unsupervised version of **SC-GANs**, with only distribution-level distance minimization.

#### 3.3. Experimental Settings

We set the dimensionality of style vectors  $d$  in all compared methods as 128, the orthogonal regularization coefficient  $\lambda$  as 0.01,  $\eta$  as 0.1. RMSProp is adopted for gradient descent, with the learning rate 0.001. The gradient clip bound  $l$  is  $-1$  and the batch size  $m$  is 30.  $n_{critic}$  is 5, i.e., we train the critic 5 iterations per generator iteration. When testing, for each compatible pair  $(x, y) \in \mathcal{P}_{test}$  we randomly select an item to replace  $y$  to generate a negative pair  $(x, y^-)$ . We adopt the widely used AUC (Area Under the ROC curve) as metric,

$$AUC = \frac{1}{|\mathcal{P}_{test}|} \sum_{(x, y) \in \mathcal{P}_{test}} \delta(r(x, y) > r(x, y^-)), \quad (12)$$

where  $\delta(a)$  is an indicator function that returns one if the argument  $a$  is *true* and zero otherwise.

<sup>2</sup> <https://tianchi.aliyun.com/datalab/index.html>.

**Table 1.** Performance comparison on the Amazon “also-bought” dataset evaluated by AUC. Seed refers to the permillage of compatible pairs in the whole dataset.

Method	Seed	Women	Men	Girls	Boys	Baby
NN	0	0.5674	0.5900	0.5229	0.5799	0.5340
CT	0.5	0.5813	0.5943	0.5100	0.5181	0.5206
	1	0.6159	0.6235	0.5510	0.5796	0.5859
	2	0.6373	0.6341	0.6073	0.6182	0.6049
LMT	0.5	0.6802	0.6722	0.6162	0.6062	0.6301
	1	0.6892	0.6812	0.6616	0.6822	0.6518
	2	0.7270	0.7111	0.6707	0.6984	0.7391
Monomer	0.5	0.6897	0.6892	0.6613	0.6691	0.6307
	1	0.6911	0.6923	0.6977	0.6742	0.6422
	2	0.7311	0.7403	0.7210	0.7301	0.6506
UC-GANs	0	0.7369	0.7248	0.6916	0.7307	0.6565
SC-GANs	0.5	0.7554	0.7620	0.7183	0.7311	0.6997
	1	0.7634	0.7702	0.7682	0.7391	0.7297
	2	<b>0.7899</b>	<b>0.7906</b>	<b>0.7778</b>	<b>0.7574</b>	<b>0.7434</b>

**Table 2.** Performance comparison evaluated by AUC on the *whole* Taobao Dataset, in which the annotated compatibility relationships don’t cover all the items.

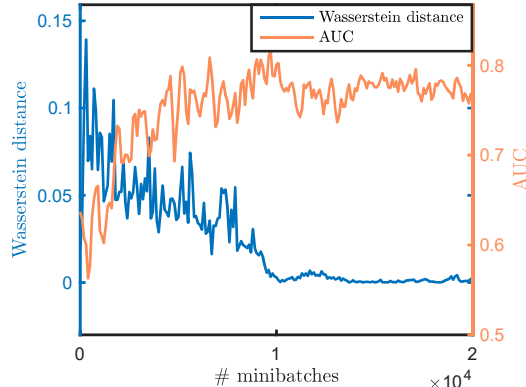
Method	NN	CT	LMT	Monomer	UC-GANs	SC-GANs
TaoBao	0.5183	0.6168	0.7335	0.7897	0.8239	<b>0.8421</b>

### 3.4. Performance Comparison

The performances on Amazon dataset and Taobao dataset are shown in Table 1 and 2 respectively. On both datasets, CT achieves better performance than NN, which indicates that categorical information is necessary for learning compatibility. As can be seen, the performances of supervised models improve with supervision increasing on Amazon dataset, which demonstrates that these supervised methods need massive supervision to achieve considerable performance. Compared with the supervised methods LMT and Monomer, UC-GANs achieve highly competitive performance on Amazon dataset and much better performance on Taobao dataset, which may due to the higher quality of annotated compatible relations in Taobao dataset. In a word, this finding confirms the effectiveness of distribution alignment on the task of compatibility learning across categories. SC-GANs outperform UC-GANs with item-level supervision, suggesting the combination of item-level distance minimization and distribution-level can better align distributions indeed. Overall, we can see that our semi-supervised method SC-GANs outperform other supervised methods in case of lacking massive supervision covering enough items, which is effective in real applications.

### 3.5. Model Analysis

We first verify that aligning the distributions can make the compatible items close. The Wasserstein distance between



**Fig. 3.** The change of AUC and Wasserstein distance along with the training process on a toy dataset consisting of two categories in the Taobao dataset.

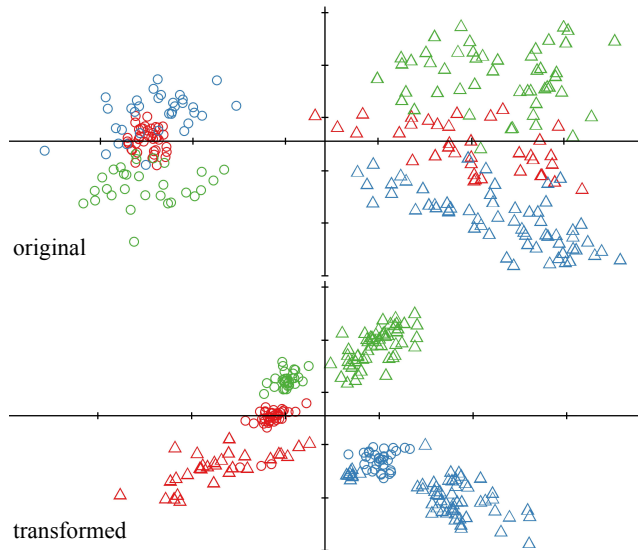
**Table 3.** The AUC results of ablation study on SC-GANs

Method	TaoBao	Amazon				
		Women	Men	Girls	Boys	Baby
SC-GANs(-O)	0.5947	0.5412	0.5318	0.5612	0.5042	0.5528
SC-GANs(-A)	0.7119	0.6608	0.6463	0.6174	0.6007	0.5519
SC-GANs(1G)	0.6856	0.7187	0.7278	0.6373	0.6815	0.6555
SC-GANs	<b>0.8421</b>	<b>0.7899</b>	<b>0.7906</b>	<b>0.7778</b>	<b>0.7574</b>	<b>0.7434</b>

two distributions indicates the degree of their alignment. We train the model on a toy dataset consisting of two randomly selected categories in the Taobao dataset. The change of AUC and Wasserstein distance along with the training process is shown in Figure 3. It is obvious that the Wasserstein distance is strongly correlated with AUC, which suggests that it’s effective to minimize the Wasserstein distance (aligning the distributions) for shortening the distances between compatible items.

We then look into each component in SC-GANs, the orthogonal constraint, category-specific style transformations and adversarial learning strategy. We compare our full model with the following models. **SC-GANs(-O)** doesn’t have the orthogonal constraint on generators. **SC-GANs(-A)** minimizes the Wasserstein distance without using adversarial learning strategy [11]. **SC-GANs(1G)** has only one transformation matrix for all categories. The experiments are conducted on the *whole* Taobao dataset and Amazon dataset with 2% annotated relationships.

Performance comparison is shown in Table 3. We notice that without the orthogonal constraint the performance is nearly equal to random guess. A possible explanation is that the transformation matrix may try to map all the items into a tiny area, in which case it’s hard to distinguish which are compatible. SC-GANs outperforms SC-GANs(-A), which implies that the adversarial learning strategy can better minimize the distances of distributions. Compared with SC-GANs(1G), the better performance of SC-GANs suggests that it’s hard to



**Fig. 4.** Visualization of the original and transformed distributions in the learned style space of tops and shoes. Circles represent shoes and triangles represent tops. The circles and triangles in same colors have same styles.

learn one general transformation matrix for all the categories. Therefore, it’s necessary to give different categories different transformation matrices.

### 3.6. Visualization

To intuitively illustrate that our proposed method can align the distributions and make compatible items with same styles close, we randomly select some tops and shoes from the Taobao dataset and show their original and transformed distributions in Figure 4, with their raw image features and style vectors reduced to 2-D by PCA. Circles represent shoes and triangles represent tops. The circles and triangles in same colors have same styles. Red represents casual style, blue represents formal style and green represents sports style. It can be seen the items in the original space cluster according to category and the compatible tops and shoes with same styles are distant. The distributions of tops and shoes have similar structures but different orientations and scales. After transformed into the learned style space, the two distributions become close. The items cluster according to style instead of category, which verifies the effectiveness of our method.

## 4. CONCLUSIONS

In this work, we first propose to consider the task of compatibility learning from item level to distribution level. We find that aligning distributions of different categories can make compatible items with same styles close. Achieving the alignment by minimizing the distance between distributions and also some annotated compatible items, we propose a semi-supervised method SC-GANs to learn compatibility across

categories for clothing matching. In fact, the item-level distance minimization part in our work can be replaced with any supervised clothing matching method, which can be improved in the future work.

## 5. ACKNOWLEDGMENTS

This work is supported by National Natural Science Foundation of China (61772528, 61871378) and National Key Research and Development Program (2016YFB1001000).

## References

- [1] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel, “Image-based recommendations on styles and substitutes,” in *SIGIR*, 2015.
- [2] Ruining He, Charles Packer, and Julian Mcauley, “Learning compatibility across categories for heterogeneous item recommendation,” in *ICDM*, 2017.
- [3] H Youn, L Sutton, E Smith, C Moore, J. F. Wilkins, I Maddieson, W Croft, and T Bhattacharya, “On the universal structure of human lexical semantics,” *Proceedings of the National Academy of Sciences of the United States of America*, 2016.
- [4] Meng Zhang, Yang Liu, Huanbo Luan, and Maosong Sun, “Earth mover’s distance minimization for unsupervised bilingual lexicon induction,” in *EMNLP*, 2017.
- [5] Karen Simonyan and Andrew Zisserman, “Very deep convolutional networks for large-scale image recognition,” *ICLR*, 2015.
- [6] Andreas Veit, Balazs Kovacs, Sean Bell, Julian McAuley, Kavita Bala, and Serge Belongie, “Learning visual clothing style with heterogeneous dyadic co-occurrences,” in *ICCV*, 2015.
- [7] Qiang Liu, Shu Wu, and Liang Wang, “Deepstyle: Learning user preferences for visual recommendation,” in *SIGIR*, 2017.
- [8] Yossi Rubner, Carlo Tomasi, and L. J. Guibas, “Metric for distributions with applications to image databases,” in *ICCV*, 1998.
- [9] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio, “Generative adversarial nets,” in *NIPS*, 2014.
- [10] Martin Arjovsky, Soumith Chintala, and Léon Bottou, “Wasserstein gan,” *ICML*, 2017.
- [11] Aude Genevay, Marco Cuturi, Gabriel Peyré, and Francis Bach, “Stochastic optimization for large-scale optimal transport,” in *NIPS*, 2016.