

Effectively Leveraging Attributes for Visual Similarity

Samarth Mishra^{*1} Zhongping Zhang^{*1} Yuan Shen² Ranjitha Kumar²
 Venkatesh Saligrama¹ Bryan A. Plummer¹
¹Boston University ²University of Illinois, Urbana-Champaign
¹{samarthm, zpzhang, srv, bplum}@bu.edu ²{yshen47, ranjitha}@illinois.edu

Abstract

Measuring similarity between two images often requires performing complex reasoning along different axes (e.g., color, texture, or shape). Insights into what might be important for measuring similarity can be provided by annotated attributes. Prior work tends to view these annotations as complete, resulting in them using a simplistic approach of predicting attributes on single images, which are, in turn, used to measure similarity. However, it is impractical for a dataset to fully annotate every attribute that may be important. Thus, only representing images based on these incomplete annotations may miss out on key information. To address this issue, we propose the Pairwise Attribute-informed similarity Network (PAN), which breaks similarity learning into capturing similarity conditions and relevance scores from a joint representation of two images. This enables our model to identify that two images contain the same attribute, but can have it deemed irrelevant (e.g., due to fine-grained differences between them) and ignored for measuring similarity between the two images. Notably, while prior methods of using attribute annotations are often unable to outperform prior art, PAN obtains a 4-9% improvement on compatibility prediction between clothing items on Polyvore Outfits, a 5% gain on few shot classification of images using Caltech-UCSD Birds (CUB), and over 1% boost to Recall@1 on In-Shop Clothes Retrieval. Implementation available at <https://github.com/samarth4149/PAN>

1. Introduction

Learning similarity metrics between images is a central problem in computer vision with wide-ranging applications such as face recognition [21, 35], image retrieval [9, 25, 49], prototype based few shot image classification [16, 36, 39, 44], continual learning of image classification [2, 32, 37], and fashion compatibility or recommen-

^{*}Indicates equal contribution

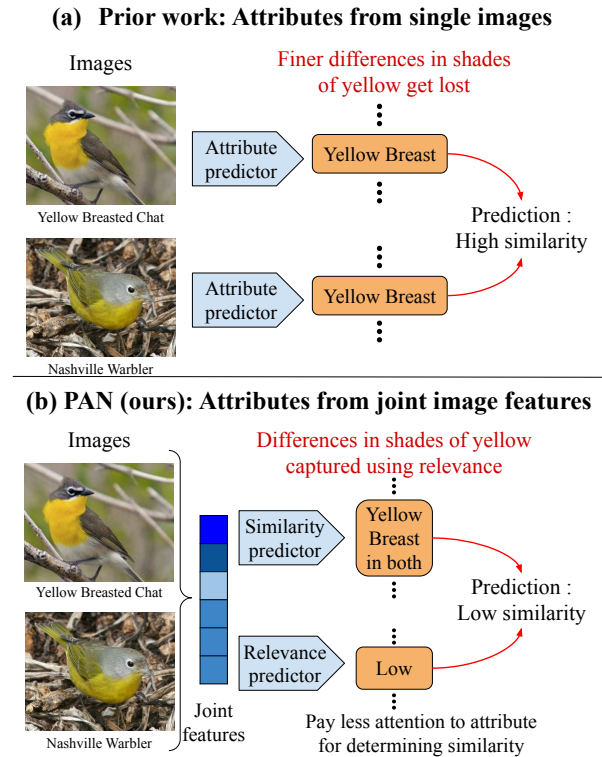


Figure 1: In prior work (e.g. [17, 28, 41, 50]), shown in (a), attributes used for image similarity are predicted for each image and then are used as input to the image similarity model. However, this can result in loss of important information about how attributes are expressed (e.g., different shades of the attribute *yellow breast*). Thus, in our work, shown in (b), we avoid this loss of information by using a joint representation of the two images to compute multiple disentangled similarity scores, each corresponding to an attribute, and relevances of each similarity score in the final similarity prediction. This allows for more fine-grained reasoning about different attribute manifestations, boosting performance.

dation [6, 40, 41, 42, 43, 52]. There has been a recent trend of learning these metrics by decomposing the prob-

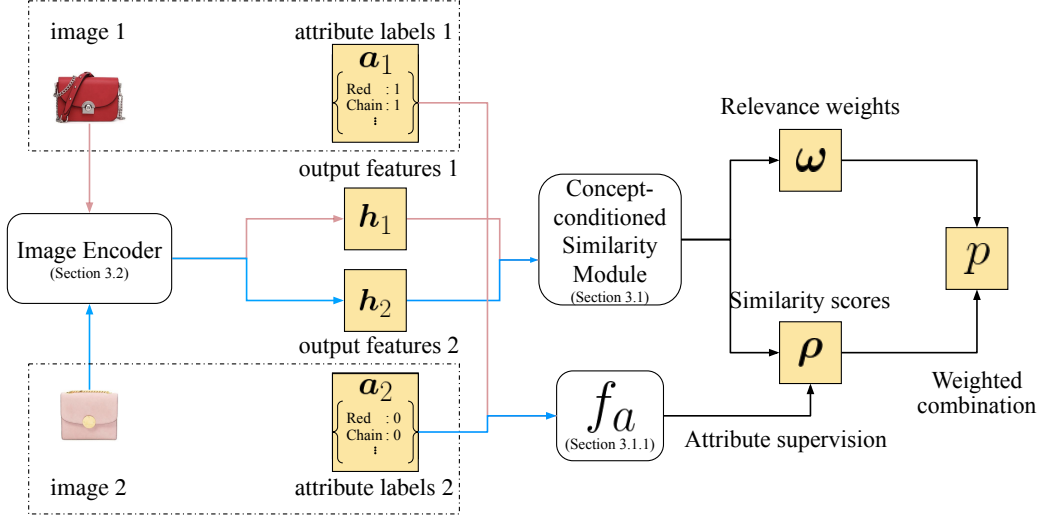


Figure 2: **PAN overview.** Given a pair of images, the goal of PAN is produce its similarity score. We begin by using the image encoder to generate feature vectors for input images. The image features are then fed into the Concept-conditioned Similarity Module (CSM) that uses these features to generate a set of similarity scores with corresponding relevance weights. This enables PAN to identify that two images do contain the same attribute, but that they are not relevant to the similarity score since they are different manifestations of the attribute (see Fig 1 for an example). The final similarity score $p \in [0, 1]$ is produced using a weighted combination of the similarity conditions and their relevance. Note that the different colored lines (blue, pink) represent information flow pertaining to individual images.

lem into multiple axes of similarity or *similarity conditions*, which has improved performance on a variety of tasks [12, 20, 26, 27, 28, 40, 41, 42]. Generally speaking, methods that automatically learn what these conditions represent [27, 40] have reported better performance than those that predefine this knowledge using information like labeled image attributes and item categories [20, 42, 41, 28]. We argue this is primarily due to prior work using attributes to predict their presence on single images (e.g. [17, 50, 41, 28]), and subsequently using these predictions for predicting similarity (Fig 1 (a)). This incurs a loss in information about the different manifestations of an attribute, differences that could affect similarity prediction, but may not be distinguishable in attribute annotations. While this could be addressed by collecting a complete set of annotations of every possible attribute and their different manifestations that could affect similarity, such a collection would be expensive. In addition, it is often impossible to articulate every fine-grained attribute that may affect similarity.

In this paper, we introduce a Pairwise Attribute-informed similarity Network (PAN) that effectively learns to use supervisory information in the form of attribute labels, avoiding information loss, to create a powerful image similarity model that performs well on a range of diverse tasks. To illustrate how we do this we refer to the example in Fig 1(b). The figure shows two birds (of different categories) from the Caltech-UCSD Birds (CUB) dataset [45],

where they are both labeled positively for the binary attribute `has_breast_color::yellow`, indicating that they have yellow breasts. Prior work (e.g. [17, 28, 41, 50]) directly predicts attributes for each image, which tends to lose information about subtle differences in the manifestations of the attributes, like the shades of the color yellow. Our PAN model avoids this issue by first comparing images in a feature space rather than attribute space, as illustrated in Fig 1(b). Using the joint image features it then predicts both a similarity score and a relevance for different similarity conditions defined by the attributes. Even when the similarity score may coarsely indicate that the two images are similar since they have the same attribute, the model can pick up on finer attribute differences and decide that the mere presence of the same attribute is of low relevance to a positive similarity prediction. As our experiments will show, this difference can make a dramatic impact on the performance of the learned image similarity model.

One major challenge we face is the considerable difference in how attributes relate to the similarity functions that arise in different tasks. For example, in few-shot classification, where we use the labeled support images in a nearest-neighbors classifier, the goal of a visual similarity classifier would be to simply measure similarity by matching attributes between the test and support images. In contrast, for tasks like fashion compatibility, where two images are deemed similar if they complement each other when worn together

in an outfit, image pairs with different attributes (*e.g.*, black and orange) can indicate they are highly compatible. Furthermore, simply modeling which attribute pairs indicate compatibility is insufficient, since two attributes which often result in compatible pairs could be deemed incompatible depending on the other attributes that are present. For example, *black* and *orange* items are often compatible, except when some other attributes like *red* are also present. Thus, visual similarity models must learn a far more complex set of relationships between attributes when learning fashion compatibility. These differences mean that methods that do well on few-shot classification often perform poorly on fashion compatibility and vice-versa. PAN, however, can take this into account via the method we use to convert the incomplete attribute labels for single images into supervisory signals for image pairs and improve performance across diverse tasks.

As we will discuss in Sec 3 (and illustrated in Fig 2), PAN naturally allows for training and automatically learning similarity conditions in the absence of any additional data like attributes. We also find that PAN can improve performance even in cases where only sparse attribute labels are available.

Summarizing our contributions:

- We propose a Pairwise Attribute-informed similarity Network (PAN), which incorporates fine-grained attribute information during training based on a joint representation of two images enabling us to avoid the loss of information suffered by prior work.
- While prior methods of incorporating attribute information under-perform prior art, PAN outperforms them on three diverse tasks—by 4-9% on fashion item compatibility prediction on Polyvore Outfits [41], 5% on few shot classification on CUB [45] and over 1% Recall@1 on In-Shop Clothing Retrieval [22], demonstrating PAN’s generality. In comparison to prior approaches of incorporating attribute supervision, PAN is better by a wider margin, *e.g.*, it outperforms them by a sizeable 6-17 % on Polyvore Outfits.
- We propose different methods of using attributes for supervising predictions along similarity conditions, delving into the interpretations of each, providing insights for their applicability in different tasks.
- Our analysis also outlines the contributions of the training procedures, specifically training batch-size. This has commonly been overlooked in prior work, but could have significant impact on final model performance. In doing so, we factor out contribution of the training procedure in demonstrating PAN’s benefits.

2. Related Work

Visual Similarity Learning. Learning visual similarity can be used for a wide range of visual tasks. A couple widely

used evaluation tasks for similarity learning are face recognition [5, 21, 24, 35] and image retrieval [25, 9, 49], the latter itself subcategorized multiple ways depending on type of images involved, *e.g.* fashion images, natural images. Image retrieval has a fairly direct application in e-commerce. A major portion of this industry consists of websites selling clothing and accessories, giving rise to the challenging task of predicting fashion compatibility [43, 14, 41], which, as discussed in the introduction, is a form of visual similarity different from the conventional notion. However, we use similarity and fashion compatibility interchangeably as they are addressed in the same manner in our paper.

Some prior visual similarity learning approaches focused on learning a single similarity space [43, 14, 10]. More recent work [42, 26, 41, 40, 20] found that learning multiple *similarity conditions*, each capturing a different concept, performs better. Concurrent to some of these, the role of using contextual information in image encodings was discussed by Cucurull *et al.* [6]. They used a graph convolutional network (GCN) on images, with similarity links defining the graph structure, to achieve state of the art performance in fashion compatibility prediction.

Up until now, methods using predefined similarity conditions [26, 42, 41] have underperformed methods that automatically learned these conditions [40, 20]. Via PAN, we show a method of incorporating additional attribute annotations to supervised similarity conditions while improving final task performance, breaking this trend.

Few Shot Learning. Given the cost associated with acquiring human-annotated labels, learning with few labelled examples is well sought after in computed vision, with a range of prior work exploring possible solutions [23, 19, 18, 16]. Given a few examples with labeled classes, [44] classified novel examples using attention weights to compute a probability distribution over known classes. They introduced an episodic training paradigm, later also adopted by [31, 36]. Each episode in an N-way K-shot classification task is a sample of N classes, with K images from each class available to the learning algorithm as “labeled” or “support” examples for learning few shot classification. Each episode is also accompanied by some query examples which the learner is supposed to predict class labels for. For training, the learning algorithm is provided episodes sampled from a base training dataset, and it is typically evaluated on test episodes sampled from a novel split of the dataset, containing classes different from those appearing in the base split. Both [44] and [36] adopted a strategy of minimizing a distance metric in feature space between query images and a prototypical support vector for training. Subsequent work [39] improved few shot classification performance by learning a parametric distance measure rather than using a closed-form metric without learnable parameters.

Another body of work uses meta-learning [8, 30, 38] to

initialize a classification model’s parameters allowing for fast adaptation to a new few-shot task using just a few parameter updates. [3] performed a thorough study of several recent few-shot learning methods and proposed a strong baseline classifier using cosine similarity and data augmentation. Some recent methods have also employed graph neural networks for few shot classification [33, 15, 34].

Like [42, 41, 40], PAN lies in the space of decomposing similarity prediction into multiple similarity conditions, but unlike them, PAN can use attribute supervision when available to supervise these conditions and improve performance. PAN is a general similarity learning approach and its benefits with multiple similarity condition spaces and attribute supervision can be seen with different image encoders, be it a simple CNN or a contextual GCN encoder similar to that used by [6], and on different tasks, like fashion compatibility and few shot classification.

3. Pairwise Attribute-informed similarity Network (PAN)

Given two images x_1 and x_2 , the goal of our PAN model is to output a score in $[0, 1]$ representing the probability that the two images are similar. The primary contribution of our approach lies in its Concept-conditioned Similarity Module (Sec 3.1), which takes features $h_1, h_2 \in \mathbb{R}^d$ representing the input images (computed using an image encoder from Sec 3.2), and predicts their similarity p using a weighted combination of predictions along multiple axes of similarity ρ and their relevance ω . As discussed in the introduction, learning relevance weights over the attribute defined similarity conditions can help us selectively ignore them when said attributes are present, but are not quite relevant in determining similarity between the images.

As our experiments demonstrate, the similarity conditions can be unsupervised as in prior work [27, 40], but the aim of PAN is to learn to supervise these conditions so they represent a particular concept (*i.e.*, a particular attribute). These attributes can represent any concept that has been annotated in a dataset. For example, for CUB [45] these represent parts of birds (*i.e.*, the type of beak or tail features). In Polyvore Outfits [41] and InShop Retrieval [22] these attributes can contain low-level concepts like colors and textures as well as high-level concepts like “formal” and “fashionable”. We discuss how we convert the attributes/concepts, which are annotated per image, into labels for image pairs in Sec 3.1.1. Fig 2 provides an overview of our approach.

3.1. Concept-conditioned Similarity Module (CSM)

Given features $h_i, h_j \in \mathbb{R}^d$ of two images, our Concept-conditioned Similarity Module (CSM) generates a set of M similarity scores $\rho = [\rho_1, \dots, \rho_M] \in \mathbb{R}^M$ and corresponding relevance weights $\omega = [\omega_1, \dots, \omega_M] \in \mathbb{R}^M$ which

represent the importance of each similarity condition to the final similarity score:

$$\rho = \sigma \left(W_1^\top |h_i - h_j| + b_1 \right) \quad (1)$$

$$\omega = \text{softmax} \left(W_2^\top |h_i - h_j| + b_2 \right) \quad (2)$$

where M is the number of distinct similarity conditions, $|\cdot|$ represents an element-wise absolute value, and $\sigma(\cdot)$ an element-wise sigmoid function. $W_1, W_2 \in \mathbb{R}^{d \times M}$, and $b_1, b_2 \in \mathbb{R}^M$ are learnable parameters. Note that ρ is supervised using attribute labels, but the relevance scores ω are treated as a latent variable and automatically learned. The final similarity score $p \in [0, 1]$ is calculated as the sum of similarity conditions weighed by their relevance, *i.e.*,

$$p = \sum_{m=1}^M \rho_m \omega_m = \rho^\top \omega. \quad (3)$$

Note that prior work that predicted multiple similarity conditions in both the supervised and unsupervised setting did so based off the features of a single image (*e.g.*, [20, 26, 41, 42]). In contrast, CSM predicts these conditions off a joint representation of both images. As we show in the supplementary, this results in a significant boost to performance when combined with relevance scores. We believe this is due, in part, to the fact that this joint representation makes it easier to identify differences in attribute manifestations (due to taking the difference of features for the two images). Thus, our approach can more accurately identify when to ignore attribute predictions.

3.1.1 Defining Similarity Conditions

Depending on the availability of labelled attributes for the images, we can choose to supervise similarity conditions to give them semantic meaning. This choice results in two kinds of similarity conditions as described below:

Unsupervised similarity conditions. Similarity conditions are treated as latent variables as done in [40]. The benefit of this approach is that it requires no additional annotations. Note that all the conditions we predict are based on a joint representation of the two images, unlike in [40] where they were computed per-image.

Supervised similarity conditions. Unsupervised similarity conditions need no attribute annotations. However, we would expect that with some expert knowledge of what might be important in the image, as provided with attributes, we can improve performance using these attributes effectively. Hence, rather than treating each similarity condition as a latent variable, supervised similarity conditions are trained to reflect a specific concept. Since attribute annotations are defined per image, and we predict attributes based off a joint

representation, we convert these labels to represent both images, as described next.

Suppose the images have M labeled binary attributes. Each image i is then accompanied with an M dimensional vector $\mathbf{a}_i \in \{0, 1\}^M$. For a pair of images i and j , we can use a function $f_a : \{0, 1\} \times \{0, 1\} \rightarrow [0, 1]$ to get an M dimensional vector $\mathbf{a}_{i,j} = f_a(\mathbf{a}_i, \mathbf{a}_j)$. Elements of $\mathbf{a}_{i,j}$ can then be used as labels for supervising the similarity conditions in the model output scores ρ . Note that if there are missing entries in $\mathbf{a}_{i,j}$ because of missing attribute labels, these can be handled by zeroing out the loss resulting from them using a binary mask over the indices of $\mathbf{a}_{i,j}$.

In Section 5.3, we experiment with common logical functions as f_a , which map to clear semantic meanings. For instance, using $f_a = \text{logical AND}$, with the similarity score ρ , the model is asked to predict whether a given attribute appears in both images. Similarly with *OR*, model predicts whether the attribute is in either image, with *XOR* it predicts whether it is exclusively in one image, and with *XNOR* it predicts whether the attribute is in both images or in neither. A more detailed discussion of opting for these 4 choices for f_a is in the supplementary. In practice for a given dataset, a logical function can be selected using some prior knowledge about how the attributes relate to the similarity score, or can be selected empirically using held-out data.

3.2. Types of Image Encoder

As mentioned previously, the image encoder generates a lower dimensional feature representation $\mathbf{h}$ for an image $\mathbf{x}$. We experiment with three different image encoders.

Convolutional Network. Unless specified otherwise, we use a simple convolutional neural network (CNN), specifically a ResNet [11] to obtain our image feature representation (details in the supplementary).

Graph Encoder (GE) [6]. For some image similarity tasks like fashion compatibility, context can be an important cue in determining how similar two items are. Thus, the second encoder we explore is a graph convolutional network (GCN) that operates on features extracted by a CNN. The GE (composed of the CNN and GCN) takes in as input both images from a dataset and an adjacency matrix over them and simultaneously generates features for all images. This encoder was also used by Cucurull *et al.* [6] and we refer readers to Section 3.1 of their paper for complete details.

ProxyNCA++ [48]. Many tasks may also find context unhelpful or that GCNs may be too computationally expensive to use [20]. For example, in retrieval tasks a particular emphasis is placed on speed as methods may have to search through millions of images in order to locate the desired item. As such, for our last encoder we use a state-of-the-art retrieval method when evaluating on the In-Shop Retrieval task [22]. ProxyNCA++ at its core learns a distance metric

between images based on learning proxy feature representations for each class. Consequently, it relies on annotated categories for images, and cannot directly be applied for similarity metric learning where no such annotation is available. We refer readers to [48] for complete details of this encoder.

3.3. Model Objective and Training

The final objective function on a pair of images $\mathbf{x}_i$ and $\mathbf{x}_j$ is then defined as:

$$\mathcal{L}(\mathbf{x}_i, \mathbf{x}_j, e_{i,j}, \mathbf{a}_{i,j}) = \mathcal{L}^{BCE}(e_{i,j}, p) + \lambda \mathcal{L}_{el}^{BCE}(\mathbf{a}_{i,j}, \rho), \quad (4)$$

where λ is a tunable hyperparameter, $e_{i,j} \in \{0, 1\}$ is the ground truth similarity label between images $\mathbf{x}_i$ and $\mathbf{x}_j$, $\mathcal{L}^{BCE}$ is the binary cross-entropy loss and $\mathcal{L}_{el}^{BCE}$ is the mean element-wise binary cross-entropy. Note that when there are no supervision attributes, the second term in Equation 4 is 0. For training, an equal number of positive and negative pairs are sampled randomly from the training split and the model is trained to predict similarity between them. Details regarding the exact process for each encoder are in the supplementary.

4. Datasets and Tasks

Polyvore Outfits [41] contains 53K outfits (sets of fashion items) for training, 5K for validation and 10K for testing. It also provides fine-grained category information and text descriptions of items. We use the 205 sparsely annotated attributes from [29] as labels for supervising similarity conditions. Evaluation involves two tasks. First, in the fill-in-the-blank (FITB) outfit completion task a model is given a partial outfit and has to select from four possible answers what item would best complete it. Performance is measured by how often the answer was correct. Second, in outfit compatibility a model is asked to discriminate between good and bad outfits. Performance is measured using area under a receiver operating characteristic curve (AUC). Following [41, 6], outfit compatibility scores are computed by averaging the similarity prediction over all pairs of items in the outfit. There are 10K FITB questions and 10K each positive and negative samples for outfit compatibility (20K total) in the test split.

Since current methods get almost perfect performance on the original outfit compatibility task, we created a more challenging testing set of the same size by modifying the procedure outlined in [41], that we refer to as the *resampled* set. For outfit compatibility we collected new negative outfits by replacing only part of a ground truth outfit, unlike the original split which replaced all items. We randomly selected the number of items to replace, and each item is replaced with another of the same type in the same split (*i.e.*, a top could only be replaced with another top). Similarly, we made a more challenging FITB task, where a model must select

between 10 candidate answers (the original test had 4). As with the original sampling, we ensured any replaced items and candidate answers were of the same type.

CUB-200-2011 [45] consists of 200 classes and a total of 11,768 images of birds. We use the split provided by [3] for our experiments, which contains 100 base classes, 50 validation and 50 novel classes. The CUB dataset also has 312 fine-grained binary attributes labeled for each image, with an accompanying score on a 4 point scale indicating the confidence of the assigned label. We drop all attribute labels that have a confidence score less than or equal to 2.

We use the 5-way 5-shot classification task for evaluation. Reported accuracies are averaged over 3 training runs from different random initializations accompanied by 95% confidence intervals. A test *episode* consists of a random sample of 5 classes and 5 support images from the 50 classes in the novel split of the dataset. 16 query images, distinct from support images, are also sampled for each of these 5 classes. The accuracy for an episode is the 5-way accuracy of a classifier over the $16 \times 5 = 80$ query images. A few shot learning model is evaluated using its average classification accuracy over 600 randomly generated test episodes.

In-Shop Retrieval [22] contains 52,712 images of clothes from 11,967 classes. There are 14,218 query images and 12,612 gallery images for testing. Given a query image, the task is to retrieve an image of the same item from the gallery set. Note that the query and gallery sets do not overlap with the training set. There are 463 attributes of clothes in total, we use these attribute labels for our PAN-Supervised model. Methods are ranked based on Recall@1.

5. Results

5.1. Comparison with prior work

Table 1, Table 2, and Table 3 compare the best settings (encoder, number of unsupervised similarity conditions, *etc.*) used by our model to representative state-of-the-art results reported in prior work on Polyvore Outfits, CUB, and In-Shop Retrieval, respectively. As shown in Table 1 we obtain a 4% better FITB accuracy and 9% AUC boost over the state-of-the-art on the fashion compatibility task using our more challenging resampled test set for both tasks, while also increasing FITB accuracy by 8% on the original split. Similarly, in Table 2 and Table 3 we observe a 5% and 1% performance improvement over the state-of-the-art on fine-grained few shot classification and In-Shop Retrieval, respectively. Improvement over the diverse set of tasks demonstrates PAN’s ability to generalize. Our PAN model can also be useful when no supervision is provided, as our PAN-Unsupervised model obtains a 3-4% gain over prior work on Polyvore Outfits and CUB, while also boosting performance on In-Shop Retrieval. Note that fashion compatibility benefited from using a graph image encoder (GE), while few-

	Method	Original		Resampled	
		FITB	AUC	FITB	AUC
(a)	TAN [41]	57.6	0.88	38.1	0.66
	SCE-Net [40]	61.6	0.91	43.4	0.68
	CSA-Net [20]	63.7	0.91	–	–
	CGAE [6]	74.1	0.99	60.8	0.67
(b)	X + Attr. Multitask-GE	73.8	0.99	57.6	0.65
	Attr. Similarity-GE	69.5	0.98	52.9	0.65
	PAN-Unsupervised-GE	78.4	0.99	64.1	0.70
	PAN-Supervised-GE	82.3	0.99	69.7	0.71

Table 1: Comparison of PAN on fashion compatibility on Polyvore Outfits to (a) results reported in prior work or reproduced with the author’s code and (b) other PAN and attribute supervision approaches.

shot classification reported best performance with a CNN encoder, which we shall discuss further in Section 5.2.

In addition to comparison with prior work, Table 1, Table 2, and Table 3 also provide two alternative methods of using attributes in an image similarity model. In “X + Attr. Multitask” we use a hard parameter sharing multitask approach [1], where we share an image encoder, but have separate output heads for each of the two tasks (one of them being attribute classification, the other similarity link prediction). In “Attr. Similarity” we use a standard framework where we predict attributes for each image and then learn a classifier implemented as a fully connected layer that takes both attributes as input and predicts similarity (the general framework used by [7]). Notably, both baseline methods that use attributes only improve performance on few-shot classification, but either make no difference, or are even harmful to performance on the other two datasets (*e.g.*, Attr. Similarity *reduces* FITB performance by 5-8% compared with the CGAE baseline). In contrast, our PAN-supervised model outperforms all other methods, including on the fashion compatibility task where we report a staggering 6-17% boost over the attribute baselines on the resampled test set.

5.2. Choice of image encoder and batch size

Prior work has been inconsistent in its training methods and controlling for hyperparameters like batch size, which can significantly affect performance. Table 4 remedies this by comparing training with the whole training split vs. using minibatches. It also compares the effect of using a graph encoder (GE) instead of a simple CNN. We see comparing the numbers of row 3 of Table 4 to those of prior methods in Tables 1 and 2 that training with the whole training set can significantly improve performance, making even a simple Siamese network trained using a triplet loss outperform most recent methods on both tasks. We note here that in our experiments with single batch training, we use a pre-trained CNN to extract image features, and do not finetune it.

	Method	Accuracy
(a)	Baseline++ [3]	83.58
	ProtoNet [36]	87.42
	TriNet [4]	84.10
	TEAM [30]	87.17
	CGAE [6]	88.00 $\pm$ 1.13
(b)	X + Attr. Multitask-GE	89.29 $\pm$ 0.57
	Attr. Similarity	92.21 $\pm$ 0.21
	PAN-Unsupervised	92.60 $\pm$ 0.10
	PAN-Supervised	92.77 $\pm$ 0.30

Table 2: Comparison of PAN on 5-way 5-shot classification on CUB-200-2011 to (a) results reported in prior work or reproduced with the author’s code and (b) other PAN and attribute supervision approaches. Intervals provided are 95% confidence intervals over 3 different runs with different random model initializations

	Method	Recall@1
(a)	MS [46]	89.7
	NormSoftMax[51]	89.4
	HORDE [13]	90.4
	Cont. w/M [47]	91.3
	ProxyNCA++[48]	90.9
(b)	ProxyNCA++ & Attr. Multitask	90.8
	ProxyNCA++ & Attr. Similarity	86.4
	ProxyNCA++ & PAN-Unsupervised	91.4
	ProxyNCA++ & PAN-Supervised	92.1

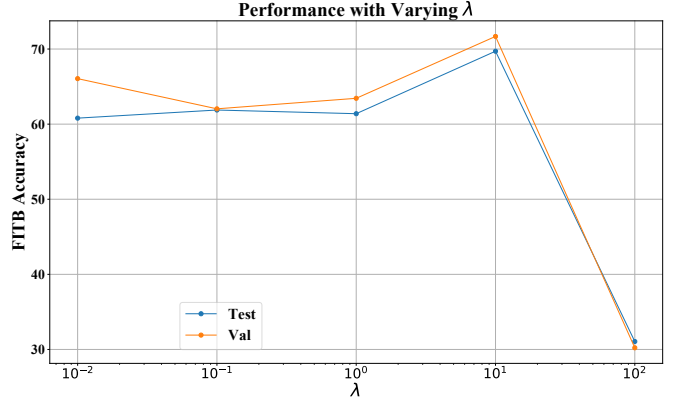
Table 3: Comparison of PAN on In-Shop Clothing Retrieval to (a) results reported in prior work and (b) other PAN and attribute supervision approaches.

It is also notable that using GE performs worse than the simpler Siamese Network baseline on CUB, which we believe is due to differences between tasks. Specifically, in fashion compatibility links exist between items that may be very different from each other. Thus, the additional context provided through the GE may be more important than in CUB, which has less variation between linked items since they all contain the same bird.

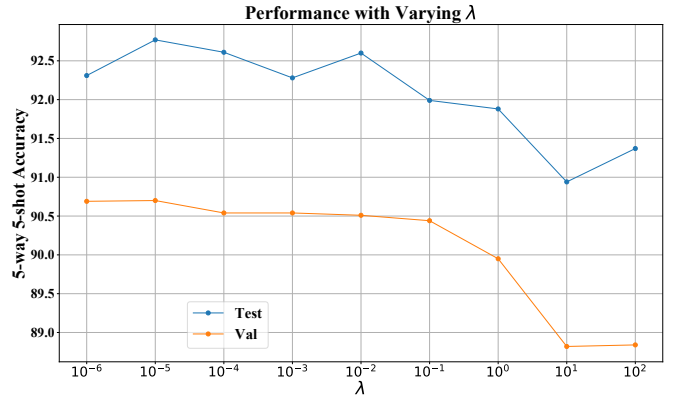
5.3. Choice of attribute combination f_a

As discussed in Section 3.1.1, choosing f_a as one of common logical functions can lead to different interpretations of the attribute supervision provided. Table 5 compares these functions that convert image attribute labels to pairwise labels for use in our PAN model.

At first glance, predicting 1 when either both images have an attribute or neither of them does (*i.e.* using $XNOR$) seems like an intuitive choice. This would indicate how many attributes match between two images. However, some functions like fashion compatibility, where images may match



(a) FITB accuracy of PAN-Supervised-GE on Polyvore Outfits.



(b) 5-way 5-shot classification accuracy of PAN-Supervised on CUB

Figure 3: Sensitivity to λ from Eq. (4).

because they contain different attributes, $XNOR$ would not be appropriate since any non-shared attributes would be ignored (*i.e.* have a supervision label 0). AND , which encourages models to predict 1 only when both images have the attribute has same issue. However OR would not face the issue since it can be 1 when either of the images has an attribute, allowing the model to use its relevance weights to decide whether a combination of attributes is relevant for similarity. In Table 5, we see OR perform the best on fashion compatibility on Polyvore Outfits.

Notably, we also see that OR is still competitive with $XNOR$ on the CUB dataset, where the goal is to determine similarity in the more conventional sense, *i.e.*, similar images should have matching attributes. This task seems like a good fit for $XNOR$. However, on CUB, many of the attributes are mutually exclusive (*e.g.*, a bird either has a “red beak” or a “black beak”, but not both). If a model can reason jointly about dissimilar attributes inferring that they should not coexist, it can correctly function on this task. Relevance weights in our model allow for this joint reasoning over different attributes. Thus, training the model to predict 1 when either image has an attribute (OR supervision), can

Method	Single Batch Training	Polyvore Outfits-Resampled			CUB-200-2011	
		$\bar{M}$	FITB	AUC	$\bar{M}$	Accuracy
Siamese Network	—	—	36.4	0.63	—	76.87 ± 0.72
X + Attr. Multitask	—	—	37.2	0.63	—	81.96 ± 1.01
Siamese Network	✓	—	44.7	0.69	—	89.01 ± 0.25
X + Attr. Multitask	✓	—	46.1	0.69	—	75.82 ± 0.42
Attr. Similarity	✓	—	31.1	0.63	—	92.21 ± 0.21
PAN-Unsupervised	✓	50	27.3	0.62	200	92.60 ± 0.10
PAN-Supervised	✓	206	28.2	0.62	312	92.77 ± 0.30
X + Attr. Multitask-GE	✓	—	57.6	0.65	—	89.29 ± 0.57
Attr. Similarity-GE	✓	—	52.9	0.65	—	87.02 ± 1.27
PAN-Unsupervised-GE	✓	50	64.1	0.70	200	89.55 ± 0.48
PAN-Supervised-GE	✓	206	69.7	0.71	312	90.16 ± 0.51

Table 4: Effect of batch size and image encoder on performance on the fashion compatibility and few-shot tasks. GE refers to the graph image encoder. Refer to Section 5.2 for discussion.

Attribute supervision label (f_a)	Polyvore Outfits				CUB-200-2011	In-Shop
	Original		Resampled		Accuracy	R@1
	FITB	AUC	FITB	AUC		
Present in either (<i>OR</i>)	82.4	0.99	69.7	0.71	92.77 ± 0.27	91.5
Present in both (<i>AND</i>)	76.3	0.98	62.4	0.62	92.61 ± 0.36	91.6
Present in both or neither (<i>XNOR</i>)	76.1	0.99	60.8	0.66	92.60 ± 0.41	92.1
Present exclusively in one (<i>XOR</i>)	69.0	0.98	51.8	0.65	92.61 ± 0.20	91.6
<i>AND</i> concat <i>XOR</i> *	78.9	0.99	64.7	0.71	92.39 ± 0.13	91.9

Table 5: Effect of different kinds of attribute supervision resulting from different functions f_a . *includes twice the number of similarity conditions as others

also perform well on this task, and from the empirical results in Table 5, we see that it does.

Providing both the *AND* and *XOR* outputs (as a concatenation) seems lucrative since it seems more informative than *OR*, but we found that the model uses its additional capacity to overfit to training data. This is also challenging to use since attribute predictions are noisy (see supplementary for attribute recognition performance).

The In-Shop retrieval task involves fetching matches from a gallery of different views of an object. *XNOR* is ideal for matching different views in this case, since matching attributes can be directly translated to the inference that two views belong to the same object, therefore, are similar.

5.4. PAN sensitivity to λ

Figure 3 shows PAN’s sensitivity to different weights (λ) of the attribute supervision loss. Performance is plotted for both the testing and validation sets. On Polyvore Outfits, we see that the PAN-Supervised-GE model performs well when attribute supervision weight is relatively high (best accuracy at $\lambda = 10$). Model performance decreases on either side of this, with a heavier decrease when λ is increased significantly (to 100). On the CUB dataset, we see a somewhat different behavior where the best model performance is achieved at $\lambda = 10^{-5}$, which is much lower, indicating relatively lower

attribute supervision is optimal for the task. Note that on the In-Shop task we set $\lambda = 1$ and did not tune it on that task, demonstrating that PAN can be readily adapted to improve performance on other tasks/models.

6. Conclusion

We presented PAN, a method of incorporating additional attribute annotations in image datasets to learn a better similarity predictor. We saw that PAN’s method of decomposing similarity prediction into multiple conditions is general, functions with a range of different image encoders and is flexible in using attribute annotation, possibly sparse, when available. PAN outperformed state of the art on a diverse set of three tasks—by 4-9% on fashion item compatibility prediction on Polyvore Outfits, 5% on few shot classification on CUB and over 1% Recall@1 on In-Shop Clothing Retrieval—contrary to prior approaches of using attribute supervision, which were unable to outperform methods that automatically learned concepts in different similarity conditions. In showing these benefits of PAN we factored out contributions from training parameters like batch-sizes, hopefully informing future work with our analysis.

Acknowledgements: This work was funded in part by DARPA and by the Hariri Data Science Faculty Fellowship.

References

- [1] Rich Caruana. Multitask learning: A knowledge-based source of inductive bias. In *Proceedings of the Tenth International Conference on International Conference on Machine Learning*, 1993. 6
- [2] Francisco M Castro, Manuel J Marín-Jiménez, Nicolás Guil, Cordelia Schmid, and Karteek Alahari. End-to-end incremental learning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 233–248, 2018. 1
- [3] Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Frank Wang, and Jia-Bin Huang. A closer look at few-shot classification. *arXiv preprint arXiv:1904.04232*, 2019. 4, 6, 7
- [4] Zitian Chen, Yanwei Fu, Yinda Zhang, Yu-Gang Jiang, Xiangyang Xue, and Leonid Sigal. Multi-level semantic feature augmentation for one-shot learning. *IEEE Transactions on Image Processing*, pages 1–1, 2019. 7
- [5] Sumit Chopra, Raia Hadsell, and Yann LeCun. Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 539–546. IEEE, 2005. 3
- [6] Guillem Cucurull, Perouz Taslakian, and David Vazquez. Context-aware visual compatibility prediction. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12617–12626, 2019. 1, 3, 4, 5, 6, 7
- [7] Ali Farhadi, Ian Endres, Derek Hoiem, and David Forsyth. Describing objects by their attributes. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1778–1785. IEEE, 2009. 6
- [8] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1126–1135. JMLR. org, 2017. 3
- [9] Albert Gordo, Jon Almazán, Jerome Revaud, and Diane Larlus. Deep image retrieval: Learning global representations for image search. In *European conference on computer vision*, pages 241–257. Springer, 2016. 1, 3
- [10] Xintong Han, Zuxuan Wu, Yu-Gang Jiang, and Larry S Davis. Learning fashion compatibility with bidirectional lstms. In *Proceedings of the 25th ACM international conference on Multimedia*, pages 1078–1086. ACM, 2017. 3
- [11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 5
- [12] Ruining He, Charles Packer, and Julian McAuley. Learning compatibility across categories for heterogeneous item recommendation. In *International Conference on Data Mining (ICDM)*, 2016. 2
- [13] Pierre Jacob, David Picard, Aymeric Histace, and Edouard Klein. Metric learning with horde: High-order regularizer for deep embeddings. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6539–6548, 2019. 7
- [14] Wang-Cheng Kang, Chen Fang, Zhaowen Wang, and Julian McAuley. Visually-aware fashion recommendation and design with generative image models. In *2017 IEEE International Conference on Data Mining (ICDM)*, pages 207–216. IEEE, 2017. 3
- [15] Jongmin Kim, Taesup Kim, Sungwoong Kim, and Chang D. Yoo. Edge-labeling graph neural network for few-shot learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 4
- [16] Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. Siamese neural networks for one-shot image recognition. In *ICML deep learning workshop*, volume 2, 2015. 1, 3
- [17] Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept bottleneck models. In *International Conference on Machine Learning*, pages 5338–5348. PMLR, 2020. 1, 2
- [18] Brenden Lake, Ruslan Salakhutdinov, Jason Gross, and Joshua Tenenbaum. One shot learning of simple visual concepts. In *Proceedings of the annual meeting of the cognitive science society*, volume 33, 2011. 3
- [19] Fei-Fei Li, Rob Fergus, and Pietro Perona. One-shot learning of object categories. *IEEE transactions on pattern analysis and machine intelligence*, 28(4):594–611, 2006. 3
- [20] Yen-Liang Lin, Son Tran, and Larry S. Davis. Fashion outfit complementary item retrieval. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2, 3, 4, 5, 6
- [21] Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. Sphreface: Deep hypersphere embedding for face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 212–220, 2017. 1, 3
- [22] Ziwei Liu, Ping Luo, Shi Qiu, Xiaogang Wang, and Xiaoou Tang. Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 3, 4, 5, 6
- [23] Erik G Miller, Nicholas E Matsakis, and Paul A Viola. Learning from one example through shared densities on transforms. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No. PR00662)*, volume 1, pages 464–471. IEEE, 2000. 3
- [24] Hieu V Nguyen and Li Bai. Cosine similarity metric learning for face verification. In *Asian conference on computer vision*, pages 709–720. Springer, 2010. 3
- [25] Hyeonwoo Noh, Andre Araujo, Jack Sim, Tobias Weyand, and Bohyung Han. Large-scale image retrieval with attentive deep local features. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3456–3465, 2017. 1, 3
- [26] Bryan A. Plummer, M. Hadi Kiapour, Shuai Zheng, and Robinson Piramuthu. Give me a hint! Navigating Image Databases using Human-in-the-loop Feedback. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2019. 2, 3, 4
- [27] Bryan A Plummer, Paige Kordas, M Hadi Kiapour, Shuai Zheng, Robinson Piramuthu, and Svetlana Lazebnik. Conditional image-text embedding networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 249–264, 2018. 2, 4

- [28] Bryan A. Plummer, Arun Mallya, Christopher M. Cervantes, Julia Hockenmaier, and Svetlana Lazebnik. Phrase localization and visual relationship detection with comprehensive image-language cues. In *The IEEE International Conference on Computer Vision (ICCV)*, 2017. 1, 2
- [29] Bryan A. Plummer, Mariya I. Vasileva, Vitali Petsiuk, Kate Saenko, and David Forsyth. Why do these match? Explaining the behavior of image similarity models. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. 5
- [30] Limeng Qiao, Yemin Shi, Jia Li, Yaowei Wang, Tiejun Huang, and Yonghong Tian. Transductive episodic-wise adaptive metric for few-shot learning. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019. 3, 7
- [31] Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. In *ICLR*, 2016. 3
- [32] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 2001–2010, 2017. 1
- [33] Victor Garcia Satorras and Joan Bruna Estrach. Few-shot learning with graph neural networks. In *International Conference on Learning Representations*, 2018. 4
- [34] Edgar Schonfeld, Sayna Ebrahimi, Samarth Sinha, Trevor Darrell, and Zeynep Akata. Generalized zero- and few-shot learning via aligned variational autoencoders. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 4
- [35] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015. 1, 3
- [36] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, pages 4077–4087, 2017. 1, 3, 7
- [37] Stefan Stojanov, Samarth Mishra, Ngoc Anh Thai, Nikhil Dhanda, Ahmad Humayun, Chen Yu, Linda B Smith, and James M Rehg. Incremental object learning from contiguous views. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8777–8786, 2019. 1
- [38] Qianru Sun, Yaoyao Liu, Tat-Seng Chua, and Bernt Schiele. Meta-transfer learning for few-shot learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 3
- [39] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1199–1208, 2018. 1, 3
- [40] Reuben Tan, Mariya I Vasileva, Kate Saenko, and Bryan A Plummer. Learning similarity conditions without explicit supervision. In *The IEEE International Conference on Computer Vision (ICCV)*, pages 10373–10382, 2019. 1, 2, 3, 4, 6
- [41] Mariya I Vasileva, Bryan A Plummer, Krishna Dusad, Shreya Rajpal, Ranjitha Kumar, and David Forsyth. Learning type-aware embeddings for fashion compatibility. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 390–405, 2018. 1, 2, 3, 4, 5, 6
- [42] Andreas Veit, Serge Belongie, and Theofanis Karaletsos. Conditional similarity networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 830–838, 2017. 1, 2, 3, 4
- [43] Andreas Veit, Balazs Kovacs, Sean Bell, Julian McAuley, Kavita Bala, and Serge Belongie. Learning visual clothing style with heterogeneous dyadic co-occurrences. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4642–4650, 2015. 1, 3
- [44] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *Advances in neural information processing systems*, pages 3630–3638, 2016. 1, 3
- [45] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011. 2, 3, 4, 6
- [46] Xun Wang, Xintong Han, Weilin Huang, Dengke Dong, and Matthew R Scott. Multi-similarity loss with general pair weighting for deep metric learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5022–5030, 2019. 7
- [47] Xun Wang, Haozhi Zhang, Weilin Huang, and Matthew R Scott. Cross-batch memory for embedding learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 7
- [48] Eu Wern Teh, Terrance DeVries, and Graham W Taylor. Proxynca++: Revisiting and revitalizing proxy neighborhood component analysis. *arXiv*, pages arXiv–2004, 2020. 5, 7
- [49] Fan Yang, Ryota Hinami, Yusuke Matsui, Steven Ly, and Shin’ichi Satoh. Efficient image retrieval via decoupling diffusion into online and offline processing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 9087–9094, 2019. 1, 3
- [50] Xun Yang, Xiangnan He, Xiang Wang, Yunshan Ma, Fuli Feng, Meng Wang, and Tat-Seng Chua. Interpretable fashion matching with rich attributes. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 775–784, 2019. 1, 2
- [51] Andrew Zhai and Hao-Yu Wu. Classification is a strong baseline for deep metric learning. *arXiv preprint arXiv:1811.12649*, 2018. 7
- [52] Zhongping Zhang, Tianlang Chen, Zheng Zhou, Jiaxin Li, and Jiebo Luo. How to become instagram famous: Post popularity prediction with dual-attention. In *2018 IEEE International Conference on Big Data (Big Data)*, pages 2383–2392. IEEE, 2018. 1