This ICCV paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the version available on IEEE Xplore.

Semantic Jitter: Dense Supervision for Visual Comparisons via Synthetic Images

Aron Yu
University of Texas at Austin
aron.yu@utexas.edu

Kristen Grauman
University of Texas at Austin
grauman@cs.utexas.edu

Abstract

Distinguishing subtle differences in attributes is valuable, yet learning to make visual comparisons remains non-trivial. Not only is the number of possible comparisons quadratic in the number of training images, but also access to images adequately spanning the space of fine-grained visual differences is limited. We propose to overcome the sparsity of supervision *problem via synthetically generated images. Building on a state-of-the-art image generation engine, we sample pairs of training images exhibiting slight modifications of individual attributes. Augmenting real training image pairs with these examples, we then train attribute ranking models to predict the relative strength of an attribute in novel pairs of real images. Our results on datasets of faces and fashion images show the great promise of bootstrapping imperfect image generators to counteract sample sparsity for learning to rank.*

1. Introduction

Fine-grained analysis of images often entails making *visual comparisons*. For example, given two products in a fashion catalog, a shopper may judge which shoe appears more pointy at the toe. Given two selfies, a teen may gauge in which one he is smiling more. Given two photos of houses for sale on a real estate website, a home buyer may analyze which facade looks better maintained.

In these and many other such cases, we are interested in inferring how a pair of images compares in terms of a particular property, or "attribute". That is, which is more *pointy*, *smiling*, *well-maintained*, etc. Importantly, the distinctions of interest are often quite subtle. Subtle comparisons arise both in image pairs that are very similar in almost every regard (e.g., two photos of the same individual wearing the same clothing, yet smiling more in one photo than the other), as well as image pairs that are holistically different yet exhibit only slight differences in the attribute in question (e.g., two individuals different in appearance, and one is smiling slightly more than the other).

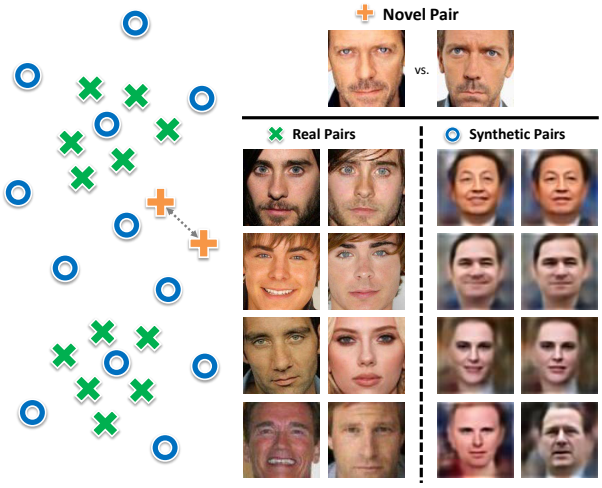

Figure 1: Our method "densifies" supervision for training ranking functions to make visual comparisons, by generating ordered pairs of synthetic images. Here, when learning the attribute *smiling*, real training images need not be representative of the entire attribute space (e.g., Web photos may cluster around commonly photographed expressions, like toothy smiles). Our idea "fills in" the sparsely sampled regions to enable fine-grained supervision. Given a novel pair (top), the nearest synthetic pairs (right) may present better training data than the nearest real pairs (left).

A growing body of work explores computational models for visual comparisons [6, 19, 24, 28, 33, 36, 38, 39, 43, 46, 47, 48]. In particular, ranking models for "relative attributes" [19, 24, 28, 33, 43, 47] use human-ordered pairs of images to train a system to predict the relative ordering in novel image pairs.

A major challenge in training a ranking model is the *sparsity of supervision*. That sparsity stems from two factors: label availability and image availability. Because training instances consist of pairs of images—together with the ground truth human judgment about which exhibits the property more or less—the space of all possible comparisons is quadratic in the number of potential training images. This quickly makes it intractable to label an image collection exhaustively for its comparative properties. At the same time, attribute comparisons entail a greater cog-

nitive load than, for example, object category labeling. Indeed, the largest existing relative attribute datasets sample only less than 0.1% of all image pairs for ground truth labels [47], and there is a major size gap between standard datasets labeled for classification (now in the millions [7]) and those for comparisons (at best in the thousands [47]). A popular shortcut is to propagate category-level comparisons down to image instances [2, 28]—e.g., deem all ocean scenes as "more open" than all forest scenes—but this introduces substantial label noise and in practice underperforms training with instance-level comparisons [19].

More insidious than the annotation cost, however, is the problem of even *curating* training images that sufficiently illustrate fine-grained differences. Critically, sparse supervision arises not simply because 1) we lack resources to get enough image pairs labeled, but also because 2) *we lack a direct way to curate photos demonstrating all sorts of subtle attribute changes*. For example, how might we gather unlabeled image pairs depicting all subtle differences in "sportiness" in clothing images or "surprisedness" in faces? As a result, even today's best datasets contain only partial representations of an attribute. See Figure 1.

We propose to use *synthetic image pairs* to overcome the sparsity of supervision problem when learning to compare images. The main idea is to synthesize plausible photos exhibiting variations along a given attribute from a generative model, thereby recovering samples in regions of the attribute space that are underrepresented among the real training images. After (optionally) verifying the comparative labels with human annotators, we train a discriminative ranking model using the synthetic training pairs in conjunction with real image pairs. The resulting model predicts attribute comparisons between novel pairs of real images.

Our idea can be seen as semantic "jittering" of the data to augment real image training sets with nearby variations. The systemic perturbation of images through label-preserving transforms like mirroring/scaling is now common practice in training deep networks for classification [9, 37, 41]. Whereas such low-level image manipulations are performed independent of the semantic content of the training instance, the variations introduced by our approach are high-level changes that affect the very meaning of the image, e.g., facial shape changes as the expression changes. In other words, our jitter has a semantic basis rather than a purely geometric/photometric basis. See Figure 2.

We demonstrate our approach in domains where subtle visual comparisons are often relevant: faces and fashion. To support our experiments, we crowdsource a lexicon of fine-grained attributes that people naturally use to describe subtle differences, and we gather new comparison annotations. In both domains—and for two distinct popular ranking models [38, 47]—we show that artificially "densifying" comparative supervision improves attribute predictions.

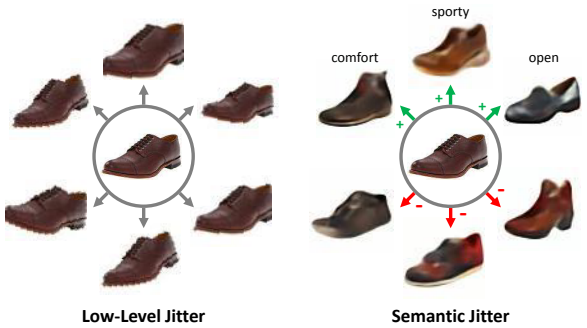

Figure 2: Whereas standard data augmentation with low-level "jitter" (left) expands training data with image-space alterations (mirroring, scaling, etc.), our *semantic jitter* (right) expands training data with high-level alterations, tweaking semantic properties in a controlled manner.

2. Related Work

Attribute Comparisons Since the introduction of relative attributes [28], the task of attribute comparisons has gained attention for its variety of applications, such as online shopping [19], biometrics [32], novel forms of low-supervision learning [2, 36], and font selection [26].

The original approach [28] adopts a learning-to-rank framework [15]. Pairwise supervision is used to train a linear ranking function for each attribute. More recently, non-linear ranking functions [24], combining feature-specific rankers [6], multi-task learning [5], fusing pointwise and pairwise labels [42], and training local rankers on the fly [47, 48] are all promising ways to improve accuracy. Other work investigates features tailored for attribute comparisons, such as facial landmark detectors [33] and *visual chains* to discover relevant parts [43]. The success of deep networks has motivated end-to-end frameworks for learning features and attribute ranking functions simultaneously [38, 39, 46]. Unlike any of the above, the novelty of our idea rests in the source data for training, not the learning algorithm. We evaluate its benefits for two popular ranking frameworks—RankSVM [2, 15, 19, 28, 47, 48] and a Siamese deep convolutional neural network (CNN) [38].

Attributes and Image Synthesis Our approach relies on a generative model for image synthesis that can progressively modify a target attribute. Attribute-specific alterations have been considered in several recent methods, primarily for face images. Some target a specific domain and attribute, such as methods to enhance the "memorability" [17] or age [16] of facial photos, or to edit outdoor scenes with transient attributes like weather [22]. Alternatively, the success of deep neural networks for image generation (i.e., GAN [10] or VAE [11, 18, 20]) opens the door to learning how to generate images conditioned on desired properties [8, 23, 27, 44, 45]. For example, a conditional multimodal auto-encoder can generate faces from attribute descriptions [27], and focus on identity-preserving

changes [23]. We employ the state-of-the-art model of [44] due to its generality. Whereas the above methods aim to produce an image for human inspection, we aim to generate dense supervision for learning algorithms.

Training Recognition Models with Synthetic Images The use of synthetic images as training data has been explored to a limited extent, primarily for human bodies. Taking advantage of high quality graphics models for humanoids, rendered images of people from various viewpoints and body poses provide free data to train pose estimators [29, 34, 35] or person detectors [31]. For objects beyond people, recent work considers how to exploit non-photorealistic images generated from 3D CAD models to augment training sets for object detection [30], or words rendered in different fonts for text recognition [13].

While these methods share our concern about the sparsity of supervision, our focus on attributes and ranking is unique. Furthermore, most methods assume a graphics engine and 3D model to render new views with desired parameters (pose, viewpoint, etc.). In contrast, we investigate images generated from a 2D image synthesis engine in which the modes of variation are controlled by a *learned* model. Being data-driven can offer greater flexibility, allowing tasks beyond those requiring a 3D model, and variability beyond camera pose and lighting parameters.

3. Approach

Our idea is to "densify" supervision for learning to make visual comparisons, by leveraging images sampled from an attribute-conditioned generative model. First we overview the visual comparison task (Sec. 3.1). Then, we describe the generative model and how we elicit dense supervision pairs from it (Sec. 3.2). Finally, we integrate synthetic and real images to train rankers for attribute comparisons (Sec. 3.3).

3.1. Visual Comparison Predictor

Let $\boldsymbol{x}_i \in \mathbb{R}^{N_x}$ denote an image with N_x pixels and let $\phi(\boldsymbol{x}_i) \in \mathbb{R}^D$ denote its D-dimensional descriptor (e.g., Gist, color, CNN feature, or simply raw pixels). Given a target attribute $\mathcal{A}$ and two images $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, the goal of visual comparison is to determine which of the images contains "more" of the specified attribute. The supervision paradigm widely adopted in ranking models for attribute comparisons [24, 28, 33, 38, 39, 43, 46, 47, 48] consists of *ordered pairs* of images. Specifically, the learning algorithm is provided with ordered pairs $\mathcal{P}_\mathcal{A} = \{(\boldsymbol{x}_i, \boldsymbol{x}_j)\}$ for which human annotators perceive image i to have the attribute $\mathcal{A}$ more than image j. The idea is to learn a ranking function $R_\mathcal{A}(\phi(\boldsymbol{x}))$ that satisfies the specified orderings as well as possible:

$$\forall (i,j) \in \mathcal{P}_\mathcal{A} : R_\mathcal{A}(\phi(\boldsymbol{x_i})) > R_\mathcal{A}(\phi(\boldsymbol{x_j})). \quad (1)$$

Precisely what defines "as well as possible" depends on the specifics of the model, such as a RankNet objective [4, 38] or paired classification objective with wide margin regularization [15, 28].

Given a novel image pair $(\boldsymbol{x}_m, \boldsymbol{x}_n)$, the ranker compares them to determine which exhibits the attribute more. If $R_\mathcal{A}(\phi(\boldsymbol{x}_m)) > R_\mathcal{A}(\phi(\boldsymbol{x}_n))$, then image m exhibits attribute $\mathcal{A}$ more than image n, and vice versa.

Our goal is to address the sparsity issue in $\mathcal{P}_\mathcal{A}$ through the addition of synthetic image pairs, such that the training pairs are more representative of subtle differences in $\mathcal{A}$. Our approach does not interfere with the formulation of the specific ranking model used. So, improvements in densifying supervision are orthogonal to improvements in the relative attribute prediction model. To demonstrate this versatility, in experiments we explore two successful learning-to-rank models from the attributes literature (see Sec. 3.3).

3.2. Synthesizing Dense Supervision

The key to improving *coverage* in the attribute space is the ability to generate images exhibiting subtle differences—with respect to the given attribute—while keeping the others constant. In other words, we want to walk semantically in the high-level attribute space.

3.2.1 Attribute-Conditioned Image Generator

We adopt an existing state-of-the-art image generation system, Attribute2Image, recently introduced by Yan et al. [44, 45], which can generate images that exhibit a given set of attributes and latent factors.

Suppose we have a lexicon of N_a attributes, $\{\mathcal{A}_1, \ldots, \mathcal{A}_{N_a}\}$. Let $\boldsymbol{y} \in \mathbb{R}^{N_a}$ be a vector containing the strength of each attribute, and let $\boldsymbol{z} \in \mathbb{R}^{N_z}$ be the latent variables. The Attribute2Image approach constructs a generative model for $p_\theta(\boldsymbol{x}|\boldsymbol{y}, \boldsymbol{z})$ that produces realistic images $\boldsymbol{x} \in \mathbb{R}^{N_x}$ conditioned on $\boldsymbol{y}$ and $\boldsymbol{z}$. The authors maximize the variational lower bound of the log-likelihood $\log\ p_\theta(\boldsymbol{x}|\boldsymbol{y})$ in order to obtain the model parameters θ. The model is implemented with a Conditional Variational Auto-Encoder (CVAE). The network architecture generates the entangled hidden representation of the attributes and latent factors with multilayer perceptrons, then generates the image pixels with a coarse-to-fine convolutional decoder. The authors apply their approach for attribute progression, image completion, and image retrieval. See [44, 45] for more details.

3.2.2 Generating Dense Synthetic Image Pairs

We propose to leverage the Attribute2Image [44] engine to supply realistic synthetic training images that "fill in" underrepresented regions of image space, which we show helps train a model to infer attribute comparisons.

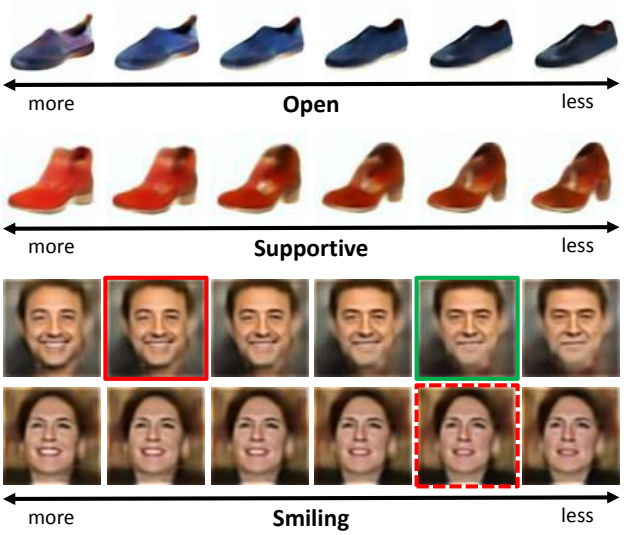

Figure 3: Spectra of generated images given an identity and an attribute. We form two types of image pairs: The two solid boxes represent an *intra-identity pair*, whereas the two red boxes represent an *inter-identity pair*.

The next key step is to generate a series of synthetic *identities*, then sample images for those identities that are close by in a desired semantic attribute space.[1] The resulting images will comprise a set of synthetic image pairs $\mathcal{S}_\mathcal{A}$. We explore two cases for using the generated pairs: one where their putative ordering is verified by human annotators, and another where the ordering implied by the generation engine is taken as their (noisy) label. Section 3.3 describes how we use the hybrid real and synthetic image pairs to train specific attribute predictors.

Each identity is defined by an entangled set of latent factors and attributes. Let $p(\boldsymbol{y})$ denote a prior over the attribute occurrences in the domain of interest. We model this prior with a multivariate Gaussian whose mean and covariance are learned from the attribute strengths observed in real training images: $p(\boldsymbol{y}) = \mathcal{N}(\mu, \Sigma)$. This distribution captures the joint interactions between attributes, such that a sample from the prior reflects the co-occurrence behavior of different pairs of attributes (e.g., shoes that are very pointy are often also uncomfortable, faces that have facial hair are often masculine, etc.).[2] The prior over latent factors $p(\boldsymbol{z})$, captures all non-attribute properties like pose, background, and illumination. Following [45], we represent $p(\boldsymbol{z})$ with an isotropic multivariate Gaussian.

To sample an identity

$$\mathcal{I}_j = (\boldsymbol{y}_j, \boldsymbol{z}_j) \tag{2}$$

we sample $\boldsymbol{y}_j$ and $\boldsymbol{z}_j$ from their respective priors. Then, using an Attribute2Image model trained for the domain of interest, we sample from $p_\theta(\boldsymbol{x}|\boldsymbol{y}_j, \boldsymbol{z}_j)$ to generate an image $\hat{\boldsymbol{x}}_j \in \mathbb{R}^{N_x}$ for this identity. Alternatively, we could sample an identity from a single real image, after inferring its latent variables through the generative model [46]. However, doing so requires having access to attribute labels for that image. More importantly, sampling novel identities from the prior (vs. an individual image) supports our goal to densify supervision, since we can draw nearby instances that need not have been exactly observed in the real training images. In experiments, we generate thousands of identities.

Next we modify the strength of a single attribute in $\boldsymbol{y}$ while keeping all other variables constant. This yields two "tweaked" identities $\mathcal{I}_j^{(-)}$ and $\mathcal{I}_j^{(+)}$ that look much like $\mathcal{I}_j$, only with a bit less or more of the attribute, respectively. Specifically, let $\sigma_\mathcal{A}$ denote the standard deviation of attribute scores observed in real training images for attribute $\mathcal{A}$. We revise the attribute vector for identity $\mathcal{I}_j$ by replacing the dimension for attribute $\mathcal{A}$ according to

$$\begin{aligned} \boldsymbol{y}_j^{(-)}(\mathcal{A}) &= \boldsymbol{y}_j(\mathcal{A}) - 2\sigma_\mathcal{A} \text{ and} \\ \boldsymbol{y}_j^{(+)}(\mathcal{A}) &= \boldsymbol{y}_j(\mathcal{A}) + 2\sigma_\mathcal{A}, \end{aligned} \tag{3}$$

and $\boldsymbol{y}_j^{(-)}(a) = \boldsymbol{y}_j^{(+)}(a) = \boldsymbol{y}_j(a), \forall a \neq \mathcal{A}$. Finally, we sample from $p_\theta(\boldsymbol{x}|\boldsymbol{y}_j^{(-)}, \boldsymbol{z}_j)$ and $p_\theta(\boldsymbol{x}|\boldsymbol{y}_j^{(+)}, \boldsymbol{z}_j)$ to obtain images $\hat{\boldsymbol{x}}_j^{(-)}$ and $\hat{\boldsymbol{x}}_j^{(+)}$. Recall that our identity sampling accounts for inter-attribute co-occurrences. Slightly altering a *single* attribute recovers plausible but yet-unseen instances.

Figure 3 shows examples of synthetic images generated for a sampled identity, varying only in one attribute. The generated images form a smooth progression in the attribute space. This is exactly what allows us to curate fine-grained pairs of images that are very similar in attribute strength. Crucially, such pairs are rarely possible to curate systematically among real images. The exception is special "hands-on" scenarios, e.g., for faces, asking subjects in a lab to slowly exhibit different facial expressions, or systematically varying lighting or head pose (cf. PIE, Yale face datasets). The hands-on protocol is not only expensive, it is inapplicable in most domains outside of faces and for rich attribute vocabularies. For example, how would one physically modify the pointiness of a shoe's toe, while leaving all other properties the same? Furthermore, the generation process allows us to collect in a controlled manner subtle visual changes *across* identities as well.

Next we pair up the synthetic images to form the set $\mathcal{S}_\mathcal{A}$, which, once (optionally) verified and pruned by human annotators, will augment the real training image pairs $\mathcal{P}_\mathcal{A}$. In order to maximize our coverage of the attribute space, we sample two types of synthetic image pairs: *intra-identity pairs*, which are images sampled from the same identity's spectrum and *inter-identity pairs*, which

[1] Note that here the word identity means an instance for some domain, not necessarily a human identity; in experiments we apply our idea both for human faces as well as fashion images of shoes.

[2] The training image attribute strengths originate from the raw decision outputs of a preliminary binary attribute classifier trained on disjoint data labeled for the presence/absence of the attribute (see Sec. 5 and Supp).

are images sampled from different spectrums (see Fig. 3). Specifically, for every identity j, $\mathcal{S}_\mathcal{A}$ receives intra pairs $\{(\hat{\boldsymbol{x}}_j^{(-)}, \hat{\boldsymbol{x}}_j), (\hat{\boldsymbol{x}}_j, \hat{\boldsymbol{x}}_j^{(+)})\}$ and for every pair of identities (j, k), $\mathcal{S}_\mathcal{A}$ receives inter pairs $\{(\hat{\boldsymbol{x}}_j, \hat{\boldsymbol{x}}_k^{(+)}), (\hat{\boldsymbol{x}}_k^{(-)}, \hat{\boldsymbol{x}}_j)\}$.

We expect many of the generated pairs to be valid, meaning that both images are realistic and that the pair exhibits a slight difference in the attribute of interest. However, this need not always be true. In some cases the generator will create images that do not appear to manipulate the attribute of interest, or where the pair is close enough in the attribute to be indistinguishable, or where the images simply do not look realistic enough to tell. Our experiments indicate this happens about 15% of the time.

To correct erroneous pairs, we collect order labels from 5 crowdworkers per pair. However, while human-verified pairs are most trustworthy for a learning algorithm, we suspect that even noisy (unverified) pairs could be beneficial too, provided the learning algorithm 1) has high enough capacity to accept a lot of them and/or 2) is label-noise resistant. Unverified pairs are attractive because they are free to generate in mass quantities. We examine both cases below.

3.3. Learning to Rank with Hybrid Comparisons

In principle any learning algorithm for visual comparisons could exploit the newly generated synthetic image pairs. We consider two common ones from the attribute literature: RankSVMs with local learning and a deep Siamese RankNet with a spatial transformer network (STN).

RankSVM+Local Learning RankSVM is a learning-to-rank solution that optimizes $R_\mathcal{A}(\phi(\boldsymbol{x}))$ to preserve orderings of training pairs while maximizing the rank margin, subject to slack [15]. In the linear case,

$$R_\mathcal{A}^{(svm)}(\phi(\boldsymbol{x})) = \boldsymbol{w}_\mathcal{A}^T \phi(\boldsymbol{x}), \tag{4}$$

where $\boldsymbol{w}$ is the ranking model parameters. The formulation is kernelizable, which allows non-linear ranking functions. It is widely used for attributes [2, 19, 28, 47, 48].

We employ RankSVM with a *local learning* model. In local learning, one trains with only those labeled instances nearest to the test input [1, 3]. Given a hybrid set of sparse real pairs and dense synthetic pairs, $\{\mathcal{P}_\mathcal{A} \bigcup \mathcal{S}_\mathcal{A}\}$, we use a local model to select the most relevant *mix* of real and synthetic pairs (see Fig. 4). Just as bare bones nearest neighbors relies on adequate density of labeled exemplars to succeed, in general local learning is expected to flourish when the space of training examples is more densely populated. Thus, local learning is congruent with our hypothesis that data *density* is at least as important as data *quantity* for learning subtle visual comparisons. See Figure 1.

Specifically, following [47], we train a local model for each novel image pair (at test time) using only the most relevant image pairs. Relevance is captured by the inter-pair image distance: for a test pair $(\boldsymbol{x}_m, \boldsymbol{x}_n)$, one gathers the K nearest pairs according to the product of element-wise distances between $(\boldsymbol{x}_m, \boldsymbol{x}_n)$ and each training pair. Only those K pairs are used to train a ranking function (Eqn(4)) to predict the order of $(\boldsymbol{x}_m, \boldsymbol{x}_n)$. See [47] for details.

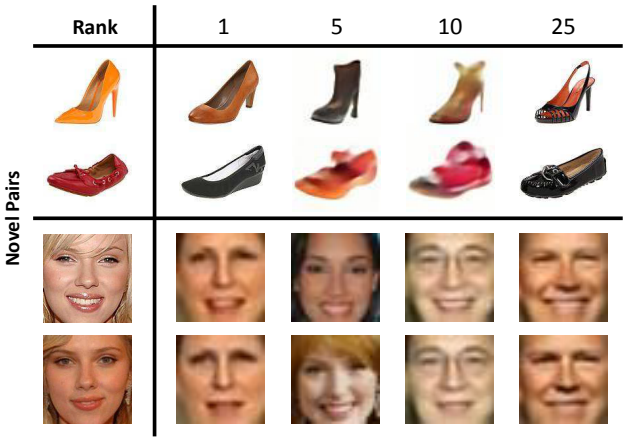

Figure 4: Examples of nearest neighbor image pairs given novel test pairs (left). Both real and synthetic image pairs appear in the top neighbors, suggesting their combined importance in the local learning algorithm.

DeepCNN+Spatial Transformer Our choice for the second ranker is motivated both by its leading empirical performance [38] as well as its high capacity, which makes it data hungry.

This deep learning to rank method combines a CNN optimized for a paired ranking loss [4] together with a spatial transformer network (STN) [14]. In particular,

$$R_\mathcal{A}^{(cnn)}(\phi(\boldsymbol{x})) = \text{RankNet}_\mathcal{A}(\text{STN}(\phi(\boldsymbol{x}))), \tag{5}$$

where RankNet denotes a Siamese network with duplicate stacks. During training these stacks process ordered pairs, learning filters that map the images to scalars that preserve the desired orderings in $\mathcal{P}_\mathcal{A}$. The STN is trained simultaneously to discover the localized patch per image that is most useful for ranking the given attribute (e.g., it may focus on the mouth for *smiling*). Given a single novel image, either stack can be used to assign a ranking score. See [38] for details. As above, our approach trains this CNN with all pairs in $\{\mathcal{P}_\mathcal{A} \bigcup \mathcal{S}_\mathcal{A}\}$.

Generator vs. Ranker A natural question to ask is why not feed back the synthetic image pairs into the same generative model that produced them, to try and enhance its training? We avoid doing so for two important reasons. First, this would lead to a circularity bias where the system would essentially be trying to exploit new data that it has already learned to capture well (and hence could generate already). Second, the particular image generator we employ is not

equipped to learn from relative supervision nor make relative *comparisons* on novel data. Rather, it learns from individual images with absolute attribute strengths. Thus, we use the synthetic data to train a distinct model capable of learning relative visual concepts.

Curating Images vs. Curating Supervision While traditional data collection methods lack a direct way to curate image pairs covering the full space of attribute variations, our approach addresses exactly this *sparsity*. It densifies the attribute space via plausible synthetic images that venture into potentially undersampled regions of the attribute spectra. Our approach does not expect to get "something for nothing". Indeed, the synthesized examples are still annotated by humans. The idea is to expose the learner to realistic images that are critical for fine-grained visual learning yet difficult to attain in traditional data collection pipelines.

4. A Lexicon of Fine-Grained Attributes

While there are numerous large datasets for single image tasks like object detection, datasets for visual comparison with *instance-level pairwise supervision* are more modest. In addition, the *lexicon* of attributes used in existing relative attributes datasets is selected based on intuitions, i.e., words that seem domain relevant [19] or words that seem to exhibit the most subtle fine-grained differences [47].

Towards addressing both limitations, we construct a new fine-grained relative attribute dataset. We 1) use crowdsourcing to mine for an attribute lexicon that is explicitly fine-grained, and 2) collect a large number of pairwise orderings for each attribute in that lexicon. We focus on fashion images of shoes from the UT-Zap50K dataset [47].

Given a pair of images, we ask Turkers to complete the sentence, "Shoe A is a *little more* ⟨insert word⟩ than Shoe B" using a single word. They are instructed to identify subtle differences between the images and provide a short rationale. The goal is to find out how people differentiate fine-grained differences between shoe images. Over 1,000 workers participated in the study, yielding a total of 350+ distinct word suggestions across 4,000 image pairs viewed. This approach to lexicon generation takes inspiration from [25], but fine-tuned towards eliciting "almost indistinguishable" visual changes rather than arbitrary attribute differences.

After post-processing based on the rationales and merging of synonyms, we select the 10 most frequent words as the new **fine-grained relative attribute lexicon** for shoes: *comfort*, *casual*, *simple*, *sporty*, *colorful*, *durable*, *supportive*, *bold*, *sleek*, and *open*. See Supp File for more details.

Using this new lexicon, we collect pairwise supervision for about 4,000 pairs for each of the 10 attributes, using images from UT-Zap50K [47]. This is a step towards denser supervision on real images—more than three times the comparison labels provided in the original dataset. Still, as we will see in results, the greater density offered by synthetic training instances is needed for best results.

5. Experiments

We conduct fine-grained visual comparison experiments to validate the benefit of our dense supervision idea, for both rankers described in Section 3.3.

Datasets Our experiments rely on the following existing and newly collected datasets. To our knowledge there exist no other instance-labeled relative attribute datasets.

Zap50K+New Lexicon The UT-Zap50K dataset [47] consists of 50,025 catalog shoe images from Zappos.com. It contains 2,800 pairwise labels on average for each of 4 attributes: *open*, *pointy*, *sporty*, and *comfort*. The labels are divided into coarse (UT-Zap50K-1) and fine-grained pairs (UT-Zap50K-2). We augment it with the crowd-mined lexicon (cf. Sec. 4) for 10 additional attributes.

Zap50K-Synth A new synthetic shoe dataset with pairwise labels on the new 10-attribute lexicon. We train the generative model using a subset of UT-Zap50K and a superset of the above attributes (see Supp File for details). We generate 1,000 identities and each one is used to sample both an intra- and inter-identity pair, yielding $\sim$2,000 pair labels per attribute. The synthetic images are 64×64 pixels.

LFW-10 The LFW-10 dataset [33] consists of 2,000 face images from Labeled Faces in the Wild (LFW) [12]. It contains 10 attributes: *bald*, *dark hair*, *eyes open*, *good looking*, *masculine*, *mouth open*, *smile*, *visible teeth*, *visible forehead*, and *young*. After pruning pairs with less than 80% agreement from the workers, there are 600 pairwise labels on average per attribute.

PFSmile Face images from the Public Figures dataset (PubFigAttr) [21, 28]. 8 frontal images each of 8 random individuals are selected, with the frontal images showing different degrees of *smilingness* for the given individual (e.g., images of Zach Efron going from not smiling at all to fully smiling). We use smiling because it is the only PubFig attribute that manifests fine-grained changes on the same individual (e.g., it doesn't display Zach both as *bald* and *less bald*). This limitation of the data actually reinforces the difficulty of manually curating images with subtle differences for learning. We collect labels on all possible pairwise comparisons among images of the same individual. After pruning, there are 211 pairwise labels.

LFW-Synth A new synthetic face dataset with pairwise labels on the attribute *smiling*. We train the generative model on a subset of LFW images and the 73 attributes from [21, 44]. We generate 2,000 identities and sample a total of 4,000 intra pairs and 1,000 inter pairs. The synthetic images are 35×35 pixels, after zooming to a tight bounding box around the face region.

		Comfort	Casual	Simple	Sporty	Colorful	Durable	Supportive	Bold	Sleek	Open
	Classifier	72.69	79.32	82.20	81.21	17.87	80.05	78.62	18.35	17.60	27.15
RankSVM	Real	84.03	86.11	86.89	87.27	83.84	85.15	**87.75**	83.71	86.06	**84.41**
	Real+	82.41	87.04	86.18	87.58	84.79	84.69	**87.75**	81.44	88.02	81.18
	Jitter	84.49	87.35	88.52	83.36	85.36	86.77	86.86	85.36	86.31	82.53
	DSynth	**85.02**	**88.89**	85.56	**89.95**	**87.43**	84.32	87.29	**87.62**	86.40	81.05
	DSynth-Auto	84.72	87.35	**87.59**	86.06	85.74	**86.78**	83.74	85.36	**86.55**	83.87
DeepSTN	Real	84.95	87.04	89.46	88.79	94.30	83.29	85.75	87.42	85.82	84.68
	Real+	81.25	87.65	86.18	87.88	90.68	83.29	85.52	87.84	86.31	82.53
	Jitter	81.94	87.96	86.89	87.58	93.73	85.38	85.75	89.07	83.86	80.65
	DSynth	82.18	**89.81**	**89.70**	**90.30**	93.73	87.24	85.52	**89.28**	86.55	82.26
	DSynth-Auto	**87.27**	88.89	88.76	90.00	**95.44**	**88.86**	**87.75**	87.63	**86.80**	**86.29**

Table 1: Results on Zap50K for the new lexicon of 10 attributes most frequently used to distinguish fine-grained differences between shoe images. We experiment with two kinds of base training models: (top) local learning (RankSVM) [47] and (bottom) localized ranking (DeepSTN) [38].

Implementation Details For synthesis, we use the code shared by the authors for the Attribute2Image system [44], with all default parameters. Since we inherit the generator's 64×64 output image resolution, for apples-to-apples comparison, we downsize real images to match the resolution of the synthetic ones. Early experiments showed that a mix of inter and intra-identity pairs was most effective, so we use a 50-50 mix in all experiments. For RankSVM, we use Gist [40] and 30-bin Lab color histograms as the image features ϕ, following [28, 47][3], and validate K per method on held-out data. For DeepSTN, we use training parameters provided in [38] per dataset. The images used to train the generative model, to train the ranking functions, and to evaluate (test set) are kept strictly disjoint.

Baselines We compare the following methods:

- **Real**: Training pool consists of only real image pairs, labeled by human annotators.
- **Jitter**: Uses the same real training pairs, but augments them with pairs using traditional low-level jitter. Each real image is jittered following parameters in [9] in a combination of five changes: translation, scaling, rotation, contrast, and color. A jittered pair inherits the corresponding real pair's label.
- **DSynth**: Training pool consists of only *half* of Real's pairs, with the other half replaced with our dense synthetic image pairs, manually verified by annotators.
- **DSynth-Auto**: Training pool consists of all real image pairs and our automatically supervised synthetic image pairs, where *noisy* pairwise supervision is obtained (for free) based on the absolute attribute strength used to generate the respective images.
- **Classifier**: Predicts the attribute scores directly using the posterior $R_{\mathcal{A}}(\phi(\boldsymbol{x})) = p(\mathcal{A}|\boldsymbol{x})$ obtained from a binary classifier trained with the same images that train the image generator.
- **Real+**: Augments Real with additional pseudo real image pairs. The image generator [44] requires attribute strength values on its training images, which are obtained from outputs of an attribute classifier [21]. The Real+ baseline trains using the same real pairs used above, plus pseudo pairs of the equal size boostrapped from those strength values on individual images.

We stress that our DSynth methods use the same amount of human-annotated pairs as the Real and Jitter baselines.

[3]Pretrained CNN features with RankSVM proved inferior.

5.1. Fashion Images of Shoes

Fashion product images offer a great testbed for fine-grained comparisons. This experiment uses UT-Zap50K for real training and testing pairs, and Zap50K-Synth for synthetic training pairs. There are 10 attributes total. Since the real train and test pairs come from the same dataset, this presents a challenge for our approach—can synthetic images, despite their inherent domain shift, still help the algorithm learn a more reliable model?

Table 1 shows the results. The Classifier baseline underperforms both rankers, confirming that the generator's initial representation of attribute strengths is insufficient.

Under the local RankSVM model, our approach outperforms the baselines in most attributes. Augmenting with traditional low-level jitter also provides a slight boost in performance, but not as much as ours. Looking at the composition of the local neighbors, we see that about 85% of the selected local neighbors are our synthetic pairs (15% real) while only 55% are jittered pairs (45% real). Thus, our synthetic pairs do indeed heavily influence the learning of the ranking models. Figure 4 shows examples of nearest neighbor image pairs retrieved for sample test pairs. The examples illustrate how 1) the synthetic images densify the supervision, providing perceptually closer instances for training, and 2) both real and synthetic image pairs play an important role in training. We conclude that semantic jitter densifies the space more effectively than low-level jitter.

Under the DeepSTN model (Table 1), our approach

		Open	Sporty	Comfort
Zap50K-1	RelAttr [28]	88.33	89.33	91.33
	FG-LP [47]	90.67	91.33	93.67
	DeepSTN [38]	93.00	93.67	94.33
	DSynth-Auto (Ours)	**95.00**	**96.33**	**95.00**
Zap50K-2	RelAttr [28]	60.36	65.65	62.82
	FG-LP [47]	69.36	66.39	63.84
	DeepSTN [38]	70.73	67.49	66.09
	DSynth-Auto (Ours)	**72.18**	**68.70**	**67.72**

Table 2: Results on UT-Zap50K-1 (coarse pairs) and UT-Zap50K-2 (fine-grained pairs) vs. prior methods. **All methods are trained and tested on 64×64 images for an apples-to-apples comparison.** All experimental setup details are kept the same except for the addition of dense synthetic pairs to the training pool for our approach.

outperforms the baselines in all attributes. Interestingly, DSynth-Auto often outperforms DSynth here. We believe the higher capacity of the DeepSTN model can better leverage the noisy auto-labeled pairs, compared to the RankSVM model, which more often benefits from the human-verification step. As one would expect, we notice that DSynth-Auto does best for attributes where the inferred labels agree most often with human provided labels. This is an exciting outcome; our model has potential to generate useful training data with "free" supervision. Low-level jitter on the other hand has limited benefit, even detrimental in some cases. Furthermore, the number of synthetic pairs used correlates positively with performance, e.g., halving the number of synthetic pairs to DSynth-Auto decreases accuracy by 4 points on average.

For both ranking models, our approach outperforms the Real baseline. This shows that simply collecting more annotations on real images is not enough: "Real" uses twice as many real training pairs as our method, yet is consistently less accurate. The finding holds even when we augment Real with [44]'s instance labels (Real+). Both baselines suffer from the *sparsity issue*, lacking the fine-grained comparisons needed to train a stronger model.

Overall, our gains are significant, considering they are achieved without any changes to the underlying ranking models, the features, or the experimental setup.

Comparison to Prior Relative Attribute Results Next, we take the best model from above (DeepSTN+DSynth-Auto), and compare its results to several existing methods. While authors have reported accuracies on this dataset, as-is comparisons to our model would not be apples-to-apples: due to the current limits of image synthesis, we work with low resolution data (64×64) whereas prior work uses full sized 150×100 [38, 39, 43]. Therefore, we use the authors' code to re-train existing methods from scratch with the same smaller real images we use. In particular, we train 1) Relative attributes (**RelAttr**) [28]; 2) Fine-grained local learning (**FG-LP**) [47]; and 3) End-to-end localization and ranking (**DeepSTN**). We compare them on UT-Zap50K, a primary benchmark for relative attributes [47].[4] We do *not* use the newly collected real labeled data for our method, to avoid an unfair advantage.

Tables 2 shows the results. Our approach does best, improving the state-of-the-art DeepSTN even for the difficult fine-grained pairs on UT-Zap50K-2 where attention to subtle details is necessary.

Smiling	Real	Real+	Jitter	DSynth	DSynth-Auto
Classifier	– – – – – – 62.35 – – – – – –				
RankSVM	69.29	68.95	74.29	73.88	**75.00**
DeepSTN	81.52	80.84	80.09	**85.78**	84.36

Table 3: Results on PFSmile dataset.

5.2. Human Faces

Next we consider the face domain. This experiment uses LFW-10 for real training pairs, LFW-Synth for synthetic training pairs, and PFSmile for real testing pairs. Since PFSmile only contains image pairs of the same individual, the comparison task is fine-grained by design. Here we have an additional domain shift, as the real train and test images are from different datasets with somewhat different properties.

Table 3 shows the results. Consistent with above, our approach outperforms all baselines. Even without human verification of our synthetic pairs (DSynth-Auto), our method secures a decent gain over the Real baseline: 75.00% vs. 69.29% and 84.36% vs. 81.52%. That amounts to a relative gain of 8% and 3.5%, respectively. The Classifier posterior baseline underperforms the rankers. Our semantic jitter strongly outperforms traditional low-level jitter for the DeepSTN rankers, with a 6 point accuracy boost.

6. Conclusion

Supervision sparsity hurts fine-grained attributes—closely related image pairs are exactly the ones the system must learn from. We presented a new approach to data augmentation, in which real training data mixes with realistic synthetic examples that vary slightly in their attributes. The generated training images more densely sample the space of images to illustrate fine-grained differences. We stress that sample *density* is distinct from sample *quantity*. As our experiments demonstrate, simply gathering more real images does not offer the same fine-grained density, due to the curation problem. Future work could explore ways to limit annotation effort to only the most questionable synthetic pairs.

Acknowledgements We thank Xinchen Yan for helpful discussions. This research is supported in part by NSF IIS-1065390 and ONR N00014-15-1-2291

[4] We test all attributes that overlap between UT-Zap50K and Zap50K-Synth, namely "open", "sporty", and "comfort".

References

[1] C. Atkeson, A. Moore, and S. Schaal. Locally weighted learning. *Artificial Intelligence Review*, 11:11–73, 1997.

[2] A. Biswas and D. Parikh. Simultaneous active learning of classifiers and attributes via relative feedback. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[3] L. Bottou and V. Vapnik. Local learning algorithms. *Neural Computation*, 1992.

[4] C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. Hullender. Learning to rank using gradient descent. In *Proceedings of International Conference on Machine Learning (ICML)*, 2005.

[5] L. Chen, Q. Zhang, and B. Li. Predicting Multiple Attributes via Relative Multi-task Learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[6] A. Datta, R. Feris, and D. Vaquero. Hierarchical ranking of facial attributes. In *Face and Gesture*, 2011.

[7] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A Large-Scale Hierarchical Image Database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[8] A. Dosovitskiy, J. Springenberg, and T. Brox. Learning to generate chairs with convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[9] A. Dosovitskiy, J. Springenberg, M. Riedmiller, and T. Brox. Discriminative unsupervised feature learning with convolutional neural networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2014.

[10] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems (NIPS)*, 2014.

[11] K. Gregor, I. Danihelka, A. Graves, and D. Wierstra. DRAW: A Recurrent Neural Network for Image Generation. In *Proceedings of International Conference on Machine Learning (ICML)*, 2015.

[12] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst, October 2007.

[13] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman. Synthetic data and artificial neural networks for natural scene text recognition. In *NIPS 14 Deep Learning workshop*, 2014.

[14] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu. Spatial transformer networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2015.

[15] T. Joachims. Optimizing search engines using clickthrough data. In *SIGKDD*, 2002.

[16] I. Kemelmacher-Shlizerman, S. Suwajanakorn, and S. Seitz. Illumination-aware age progression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[17] A. Khosla, W. A. Bainbridge, A. Torralba, and A. Oliva. Modifying the memorability of face photographs. In *International Conference on Computer Vision (ICCV)*, 2013.

[18] D. Kingma and M. Welling. Auto-encoding variational bayes. 2014.

[19] A. Kovashka, D. Parikh, and K. Grauman. WhittleSearch: Interactive image search with relative attribute feedback. *International Journal of Computer Vision (IJCV)*, 115(2):185–210, Nov 2015.

[20] T. Kulkarni, W. Whitney, P. Kohli, and J. Tenenbaum. Deep convolutional inverse graphics network. In *Advances in Neural Information Processing Systems (NIPS)*, 2015.

[21] N. Kumar, A. Berg, P. Belhumeur, and S. Nayar. Attribute and Simile Classifiers for Face Verification. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2009.

[22] P. Laffont, Z. Ren, X. Tao, C. Qian, and J. Hays. Transient attributes for high-level understanding and editing of outdoor scenes. In *SIGGRAPH*, 2014.

[23] M. Li, W. Zuo, and D. Zhang. Convolutional network for attribute-driven and identity-preserving human face generation. Technical Report arXiv:1608.06434, 2016.

[24] S. Li, S. Shan, and X. Chen. Relative forest for attribute prediction. In *Asian Conference on Computer Vision (ACCV)*, 2012.

[25] S. Maji. Discovering a lexicon of parts and attributes. In *Second International Workshop on Parts and Attributes, ECCV*, 2012.

[26] P. O'Donovan, J. Libeks, A. Agarwala, and A. Hertzmann. Exploratory font selection using crowdsourced attributes. In *SIGGRAPH*, 2014.

[27] G. Pandey and A. Dukkipati. Variational methods for conditional multimodal learning: Generating human faces from attributes. Technical Report arXiv:1603.01801, 2016.

[28] D. Parikh and K. Grauman. Relative Attributes. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2011.

[29] D. Park and D. Ramanan. Articulated pose estimation with tiny synthetic videos. In *ChaLearn Workshop, CVPR*, 2015.

[30] X. Peng, B. Sun, K. Ali, and K. Saenko. Learning deep object detectors from 3d models. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015.

[31] L. Pishchulin, A. Jain, C. Wojek, T. Thormahlen, and B. Schiele. In good shape: Robust people detection based on appearance and shape. In *British Machine Vision Conference (BMVC)*, 2011.

[32] D. Reid and M. Nixon. Human identification using facial comparative descriptions. In *ICB*, 2013.

[33] R. Sandeep, Y. Verma, and C. Jawahar. Relative parts: Distinctive parts for learning relative attributes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[34] G. Shakhnarovich, P. Viola, and T. Darrell. Fast Pose Estimation with Parameter-Sensitive Hashing. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2003.

[35] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake. Real-time human pose recognition in parts from single depth images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[36] A. Shrivastava, S. Singh, and A. Gupta. Constrained semi-supervised learning using attributes and comparative attributes. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2012.

[37] P. Simard, D. Steinkraus, and J. Platt. Best practices for convolutional neural networks applied to visual document analysis. In *ICDAR*, 2003.

[38] K. Singh and Y. J. Lee. End-to-end localization and ranking for relative attributes. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2016.

[39] Y. Souri, E. Noury, and E. Adeli. Deep relative attributes. In *Asian Conference on Computer Vision (ACCV)*, 2016.

[40] A. Torralba. Contextual Priming for Object Detection. *International Journal of Computer Vision (IJCV)*, 53(2):169–191, 2003.

[41] P. Vincent, H. Larochelle, Y. Bengio, and P. Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of International Conference on Machine Learning (ICML)*, 2008.

[42] Y. Wang, S. Wang, J. Tang, H. Liu, and B. Li. PPP: Joint Pointwise and Pairwise Image Label Prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[43] F. Xiao and Y. J. Lee. Discovering the spatial extent of relative attributes. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015.

[44] X. Yan, J. Yang, K. Sohn, and H. Lee. Attribute2Image: Conditional image generation from visual attributes. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2016.

[45] X. Yan, J. Yang, K. Sohn, and H. Lee. Attribute2Image: Conditional image generation from visual attributes. Technical Report arXiv:1512.00570, 2016.

[46] X. Yang, T. Zhang, C. Xu, S. Yan, M. Hossain, and A. Ghoneim. Deep relative attributes. *IEEE Trans. on Multimedia*, 18(9), Sept 2016.

[47] A. Yu and K. Grauman. Fine-Grained Visual Comparisons with Local Learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[48] A. Yu and K. Grauman. Just Noticeable Differences in Visual Attributes. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015.

[49] A. Yu and K. Grauman. Semantic Jitter: Dense Supervision for Visual Comparisons via Synthetic Images. Technical Report arXiv:1612.06341, 2016.