Fine-Grained Comparisons with Attributes

Aron Yu and Kristen Grauman

Abstract Given two images, we want to predict which exhibits a particular visual attribute more than the other—even when the two images are quite similar. For example, given two beach scenes, which looks *more calm*? Given two high-heeled shoes, which is *more ornate*? Existing relative attribute methods rely on global ranking functions. However, rarely will the visual cues relevant to a comparison be constant for all data, nor will humans' perception of the attribute necessarily permit a global ordering. At the same time, not every image pair is even orderable for a given attribute. Attempting to map relative attribute ranks to "equality" predictions is non-trivial, particularly since the span of indistinguishable pairs in attribute space may vary in different parts of the feature space. To address these issues, we introduce *local learning* approaches for fine-grained visual comparisons, where a predictive model is trained on the fly using only the data most relevant to the novel input. In particular, given a novel pair of images, we develop local learning methods to (1) infer their relative attribute ordering with a ranking function trained using only analogous labeled image pairs, (2) infer the optimal "neighborhood", i.e., the subset of the training instances most relevant for training a given local model, and (3) infer whether the pair is even distinguishable, based on a local model for *just noticeable differences* in attributes. Our methods outperform state-of-the-art methods for relative attribute prediction on challenging datasets, including a large newly curated shoe dataset for fine-grained comparisons. We find that for fine-grained comparisons, *more* labeled data is not necessarily preferable to isolating the *right* data.

Aron Yu
University of Texas at Austin, e-mail: aron.yu@utexas.edu

Kristen Grauman
University of Texas at Austin, e-mail: grauman@cs.utexas.edu

Fig. 1: A global ranking function may be suitable for *coarse* ranking tasks, but *fine-grained* ranking tasks require attention to subtle details—and which details are important may vary in different parts of the feature space. We propose a local learning approach to train comparative attributes based on fine-grained analogous pairs.

1 Introduction

Attributes are visual properties describable in words, capturing anything from material properties (*metallic*, *furry*), shapes (*flat*, *boxy*), expressions (*smiling*, *surprised*), to functions (*sittable*, *drinkable*). Since their introduction to the recognition community [19, 35, 37], attributes have inspired a number of useful applications in image search [32, 34, 35, 50], biometrics [11, 45], and language-based supervision for recognition [6, 37, 43, 49].

Existing attribute models come in one of two forms: categorical or relative. Whereas categorical attributes are suited only for clear-cut predicates, such as *male* or *wooden*, relative attributes can represent "real-valued" properties that inherently exhibit a spectrum of strengths, such as *serious* or *sporty*. These spectra allow a computer vision system to go beyond recognition into comparison. For example, with a model for the relative attribute *brightness*, a system could judge which of two images is *brighter* than the other, as opposed to simply labeling them as bright/not bright.

Attribute comparisons open up a number of interesting possibilities. In biometrics, the system could interpret descriptions like, "the suspect is *taller* than him" [45]. In image search, the user could supply semantic feedback to pinpoint his desired content: "the shoes I want to buy are like these but *more masculine*" [34], as discussed in Chapter XXXXX of this book. For object recognition, human supervisors could teach the system by relating new objects to previously learned ones, e.g., "a mule has a tail *longer than* a donkey's" [6, 43, 49]. For subjective visual tasks, users could teach the system their personal perception, e.g., about which human faces are *more attractive* than others [1].

One typically learns a relative attribute in a learning-to-rank setting; training data is ordered (e.g., we are told image A has it more than B), and a ranking function is optimized to preserve those orderings. Given a new image, the function returns a score conveying how strongly the attribute is present [1, 10, 14, 18, 34, 38, 41, 43, 46, 47]. While a promising direction, the standard ranking approach tends to

fail when faced with *fine-grained visual comparisons*. In particular, the standard approach falls short on two fronts: (1) it cannot reliably predict comparisons when the novel pair of images exhibits subtle visual attribute differences, and (2) it does not permit equality predictions, meaning it is unable to detect when a novel pair of images are so similar that their difference is indistinguishable.

Why do existing global ranking functions experience difficulties making fine-grained attribute comparisons? The problem is that while a single learned function tends to accommodate the gross visual differences that govern the attribute's spectrum, it cannot simultaneously account for the many fine-grained differences among closely related examples, each of which may be due to a distinct set of visual cues. For example, what makes a slipper appear *more comfortable* than a high heel is different than what makes one high heel appear more comfortable than another; what makes a mountain scene appear *more natural* than a highway is different than what makes a suburb more natural than a downtown skyscraper (Fig. 1).

Furthermore, at some point, fine-grained differences become so subtle that they become indistinguishable. However, existing attribute models assume that all images are orderable. In particular, they assume that *at test time*, the system can and should always distinguish which image in a pair exhibits the attribute more. Imagine you are given a pile of images of Barack Obama, and you must sort them according to where he looks most to least *serious*. Can you do it? Surely there will be some obvious ones where he is more serious or less serious. There will even be image pairs where the distinction is quite subtle, yet still perceptible, thus fine-grained. However, you are likely to conclude that forcing a *total* order is meaningless: while the images exhibit different degrees of the attribute seriousness, at some point the differences become indistinguishable. It is not that the pixel patterns in indistinguishable image pairs are literally the same—they just cannot be characterized consistently as anything other than "equally serious" (Fig. 2).

We contend that such fine-grained comparisons are critical to get right, since this is where modeling relative attributes ought to have great power. Otherwise, we could just learn coarse categories of appearance ("bright scenes", "dark scenes") and manually define their ordering. In particular, fine-grained visual comparisons are valuable for sophisticated image search and browsing applications, such as distinguishing subtle properties between products in an online catalog, as well as analysis tasks involving nuanced perception, such as detecting slight shades of human facial expressions or distinguishing the identifying traits between otherwise similar-looking people.

In light of these challenges, we introduce *local learning* algorithms for fine-grained visual comparisons. Local learning is an instance of "lazy learning", where one defers processing of the training data until test time. Rather than estimate a single global model from all training data, local learning methods instead focus on a subset of the data most relevant to the particular test instance. This helps learn fine-grained models tailored to the new input, and makes it possible to adjust the capacity of the learning algorithm to the local properties of the data [7]. Local methods include classic nearest neighbor classification as well as various novel formulations

Fig. 2: At what point is the strength of an attribute indistinguishable between two images? While existing relative attribute methods are restricted to inferring a total order, in reality there are images that look different but where the attribute is nonetheless perceived as "equally strong". For example, in the fourth and fifth images of Obama, is the difference in *seriousness* noticeable enough to warrant a relative comparison?

that use only nearby points to either train a model [2, 3, 7, 24, 57] or learn a feature transformation [16, 17, 25, 51] that caters to the novel input.

The local learning methods we develop in this chapter address the questions of (1) how to compare an attribute in highly similar images as well as (2) how to determine when such a comparison is not possible. To learn fine-grained ranking functions for attributes, given a novel test pair of images, we first identify *analogous* training pairs using a learned attribute-specific metric. Then we train a ranking function on the fly using only those pairs [54]. Building on this framework, we further explore how to predict the local *neighborhood* itself—essentially answering the "how local" question. Whereas existing local learning work assumes a fixed number of proximal training instances are most relevant, our approach infers the relevant set as a whole, both in terms of its size and composition [55]. Finally, to decide when a novel pair is indistinguishable in terms of a given attribute, we develop a Bayesian approach that relies on local statistics of orderability to learn a model of *just noticeable difference* (JND) [56].

Roadmap The rest of the chapter proceeds as follows. In Section 2, we discuss related work in the areas of relative attributes, local learning, and fine-grained visual learning. In Section 3, we provide a brief overview of the relative attributes ranking framework. In Sections 4 and 5, we discuss in detail our proposed approaches for fine-grained visual comparisons and equality prediction using JND. Finally, we conclude in Sections 6 and 7 with further discussion and future work. The work described in this chapter originally was presented in our previous conference papers [54, 55, 56].

2 Related Work

Attribute Comparison Attribute comparison has gained attention in the last several years. The original "relative attributes" approach learns a global linear ranking

function for each attribute [43]. Pairwise supervision is used for training: a set of pairs ordered according to their perceived attribute strength is obtained from human annotators, and a ranking function that preserves those orderings is learned. Given a novel pair of images, the ranker indicates which image has the attribute more. It is extended to non-linear ranking functions in [38] by training a hierarchy of rankers with different subsets of data, then normalizing predictions at the leaf nodes. In [14], rankers trained for each feature descriptor (color, shape, texture) are combined to produce a single global ranking function. In [47], part-based representations weighted specifically for each attribute are used instead of global features.

Aside from learning to rank formulations, researchers have applied the Elo rating system for biometrics [45], and regression over "cumulative attributes" for age and crowd density estimation [11].

All the prior methods produce a single global function for each attribute, whereas we propose to learn local functions tailored to the comparison at hand. While some implementations (including [43]) augment the training pool with "equal" pairs to facilitate learning, notably no existing work attempts to discern distinguishable from indistinguishable pairs at test time. As we will see below, doing so is non-trivial.

Fine-Grained Visual Tasks Work on fine-grained visual *categorization* aims to recognize objects in a single domain, e.g., bird species [9, 20]. While such problems also require making distinctions among visually close instances, our goal is to compare attributes, not categorize objects.

In the facial attractiveness ranking method of [10], the authors train a hierarchy of SVM classifiers to recursively push a image into buckets of more/less attractive faces. The leaf nodes contain images "unrankable" by the human subject, which can be seen as indistinguishability for the specific attribute of human attractiveness. Nonetheless, the proposed method is not applicable to our problem. It learns a ranking model specific to a single human subject, whereas we learn a subject-independent model. Furthermore, the training procedure [10] has limited scalability, since the subject must rank *all* training images into a partial order; the results focus on training sets of 24 images for this reason. In our domains of interest, where thousands or more training instances are standard, getting a reliable global partial order on all images remains an open challenge.

Variability in Visual Perception The fact that humans exhibit inconsistencies in their comparisons is well known in social choice theory and preference learning [8]. In existing global models [1, 10, 14, 18, 34, 38, 41, 43, 47], intransitive constraints would be unaccounted for and treated as noise. While the HodgeRank algorithm [28] also takes a global ranking approach, it estimates how much it suffers from cyclic inconsistencies, which is valuable to know how much to trust the final ranking function. However, that approach does not address the fact that the features relevant to a comparison are not uniform across a dataset, which we find is critical for fine-grained comparisons.

We are interested in modeling attributes where there *is* consensus about comparisons, only they are subtle. Rather than personalize a model towards an observer [1, 10, 31], we want to discover the (implicit) map of where the consensus for

JND boundaries in attributes exists. The attribute calibration method of [48] post-processes attribute classifier outputs so they can be fused for multi-attribute search. Our method is also conscious that differences in attribute outputs taken at "face value" can be misleading, but our goal and approach are entirely different.

Local Learning In terms of learning algorithms, lazy local learning methods are relevant to our work. Existing methods primarily vary in how they exploit the labeled instances nearest to a test point. One strategy is to identify a fixed number of neighbors most similar to the test point, then train a model with only those examples (e.g., a neural network [7], SVM [57], ranking function [3, 24], or linear regression [2]). Alternatively, the nearest training points can be used to learn a transformation of the feature space (e.g., Linear Discriminant Analysis); after projecting the data into the new space, the model is better tailored to the query's neighborhood properties [16, 17, 25, 51]. In *local selection* methods, strictly the subset of nearby data is used, whereas in *locally weighted* methods, all training points are used but weighted according to their distance [2]. For all these prior methods, a test case is a new data point, and its neighboring examples are identified by nearest neighbor search (e.g., with Euclidean distance). In contrast, we propose to learn local ranking functions for comparisons, which requires identifying analogous neighbor *pairs* in the training data. Furthermore, we also explore how to *predict* the variable-size set of training instances that will produce an effective discriminative model for a given test instance.

In information retrieval, local learning methods have been developed to sort documents by their relevance to query keywords [3, 17, 24, 39]. They take strategies quite similar to the above, e.g., building a local model for each cluster in the training data [39], projecting training data onto a subspace determined by the test data distribution [17], or building a model with only the query's neighbors [3, 24]. Though a form of ranking, the problem setting in all these methods is quite different from ours. There, the training examples consist of queries and their respective sets of ground truth "relevant" and "irrelevant" documents, and the goal is to learn a function to rank a keyword query's relevant documents higher than its irrelevant ones. In contrast, we have training data comprised of paired comparisons, and the goal is to learn a function to compare a novel query pair.

Metric Learning The question "what is relevant to a test point?" also brings to mind the metric learning problem. Metric learning methods optimize the parameters of a distance function so as to best satisfy known (dis)similarity constraints between training data [4]. Most relevant to our work are those that learn *local* metrics; rather than learn a single global parameterization, the metric varies in different regions of the feature space. For example, to improve nearest neighbor classification, in [22] a set of feature weights is learned for each individual training example, while in [52, 53] separate metrics are trained for clusters discovered in the training data. Such methods are valuable when the data is multi-modal and thus ill-suited by a single global metric. In contrast to our approach, however, they learn local models offline on the basis of the fixed training set, whereas our approaches dynamically train new models as a function of the novel queries.

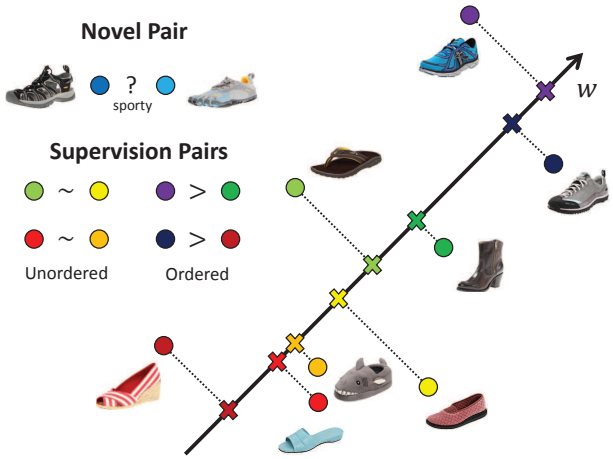


Fig. 3: Illustration of a learned linear ranking function trained from ordered pairs. The goal is to learn a ranking function $R_{\mathcal{A}}(x)$ that satisfies both the ordered and unordered pairwise constraints. Given a novel test pair, the real-valued ranking scores of the images are compared to determine their relative ordering.

3 Ranking Functions for Relative Attributes

First we describe how attribute comparisons can be addressed with a learning to rank approach, as originally proposed by Parikh and Grauman [43]. Ranking functions will also play a role in our solution, and the specific model we introduce next will further serve as the representative traditional "global" approach in our experiments.

Our approach addresses the relative comparison problem on a per attribute basis.[1] As training data for the attribute of interest $\mathcal{A}$ (e.g., *comfortable*), we are given a pool of ground truth comparisons on pairs of images. Then, given a novel pair of images, our method predicts which exhibits the attribute more, that is, which of the two images appears *more comfortable*, or if the images are equal, or in other words, *totally indistinguishable*. We first present a brief overview of Relative Attributes [43] as it sets the foundation as a baseline global ranking approach.

The Relative Attributes approach treats the attribute comparison task as a learning-to-rank problem. The idea is to use ordered pairs (and optionally "equal" pairs) of training images to train a ranking function that will generalize to new images. Compared to learning a regression function, the ranking framework has the advantage that training instances are themselves expressed comparatively, as opposed to requiring a rating of the absolute strength of the attribute per training image.

For each attribute $\mathcal{A}$ to be learned, we take as input two sets of annotated training image pairs. The first set consists of ordered pairs, $\mathcal{P}_o = \{(i, j)\}$, for which humans perceive image i to have the attribute more than image j. That is, each pair in $\mathcal{P}_o$ has a "noticeable difference". The second set consists of unordered, or "equal" pairs,

[1] See Chapter XXXXX for discussion on methods for jointly training multiple attributes.

$\mathcal{P}_e = \{(m,n)\}$, for which humans cannot perceive a difference in attribute strength. See Section 4.3 for discussion on how such human-annotated data can be reliably collected.

Let $x_i \in \mathbb{R}^d$ denote the d-dimensional image descriptor for image i, such as a GIST descriptor or a color histogram, and let $R_\mathcal{A}$ be a linear ranking function:

$$R_\mathcal{A}(x) = w_\mathcal{A}^T x. \tag{1}$$

Using a large-margin approach based on the SVM-Rank framework [29], the goal for a global relative attribute is to learn the parameters $w_\mathcal{A} \in \mathbb{R}^d$ that optimize the rank function parameters to preserve the orderings in $\mathcal{P}_o$, maintaining a margin between them in the 1D output space, while also minimizing the separation between the unordered pairs in $\mathcal{P}_e$. By itself, the problem is NP-hard, but [29] introduces slack variables and a large-margin regularizer to approximately solve it. The learning objective is:

$$\begin{aligned} \text{minimize} \quad & \left(\tfrac{1}{2}||w_\mathcal{A}||_2^2 + C\left(\sum \xi_{ij}^2 + \sum \gamma_{m,n}^2\right)\right) \\ s.t. \quad & w_\mathcal{A}^T(x_i - x_j) \geq 1 - \xi_{ij}; \forall (i,j) \in \mathcal{P}_o \\ & |w_\mathcal{A}^T(x_m - x_n)| \leq \gamma_{pq}; \forall (m,n) \in \mathcal{P}_e \\ & \xi_{ij} \geq 0; \gamma_{mn} \geq 0, \end{aligned} \tag{2}$$

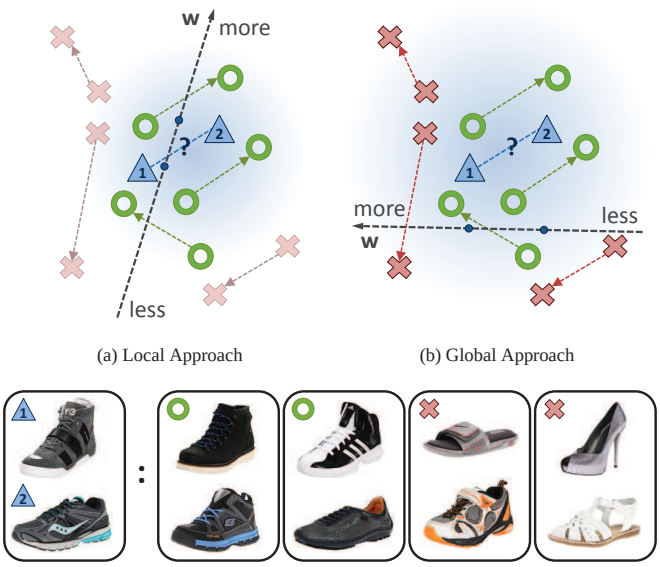
where the constant C balances the regularizer and ordering constraints, and γ_{pq} and ξ_{ij} denote slack variables. By projecting images onto the resulting hyperplane $w_\mathcal{A}$, we obtain a 1D global ranking for that attribute, e.g., from least to most *comfortable*.

Given a test pair (x_r, x_s), if $R_\mathcal{A}(x_r) > R_\mathcal{A}(x_s)$, then image r exhibits the attribute more than image s, and vice versa. While [43] uses this linear formulation, it is also kernelizable and so can produce non-linear ranking functions.

Our local approach defined next draws on this particular ranking formulation, which is also used in both [43] and in the hierarchy of [38] to produce state-of-the-art results. Note however that our local learning idea would apply similarly to alternative ranking methods.

4 Fine-Grained Visual Comparisons

Existing methods train a global ranking function using all available constraints $\mathcal{P}_o$ (and sometimes $\mathcal{P}_e$), with the implicit assumption that more training data should only help better learn the target concept. While such an approach tends to capture the coarse visual comparisons, it can be difficult to derive a single set of model parameters that adequately represents both these big-picture contrasts *and* more subtle fine-grained comparisons (recall Fig. 1). For example, for a dataset of shoes, it will map all the sneakers on one end of the *formal* spectrum, and all the high heels on the other, but the ordering among closely related high heels will not show a clear pattern. This suggests there is an interplay between the model capacity and the density of available training examples, prompting us to explore local learning solutions.



(a) Local Approach (b) Global Approach

Fig. 4: Given a novel test pair (blue △) in a learned metric space, our local approach (a) selects only the most relevant neighbors (green ○) for training, which leads to ranking test image 2 over 1 in terms of *sporty*. In contrast, the standard global approach defined in Sect. 3 (b) uses all training data (green ○ & red ×) for training; the unrelated training pairs dilute the training data. As a result, the global model accounts largely for the coarse-grained differences, and incorrectly ranks test image 1 over 2. The end of each arrow points to the image with *more* of the attribute (sporty). Note that the rank of each point is determined by its *projection* onto w.

In the following, we next introduce our local ranking approach (Sect. 4.1) and the mechanism to selecting fine-grained neighboring pairs with attribute-specific metric learning (Sect. 4.2). On three challenging datasets from distinct domains, including a newly curated large dataset of 50,000 Zappos shoe images that focuses on fine-grained attribute comparisons (Sect. 4.3), we show our approach improves the state-of-the-art in relative attribute predictions (Sect. 4.4). After the results, we briefly overview an extension of the local attribute learning idea that learns the *neighborhood* of relevant training data that ought to be used to train a model on the fly (Sect. 4.5).

4.1 Local Learning for Visual Comparisons

The solution to overcoming the shortcomings of existing methods discussed above is not simply a matter of using a higher capacity learning algorithm. While a low capacity model can perform poorly in well-sampled areas, unable to sufficiently exploit the dense training data, a high capacity model can produce unreliable (yet highly confident) decisions in poorly sampled areas of the feature space [7]. Different properties are required in different areas of the feature space. Furthermore, in our visual ranking domain, we can expect that as the amount of available train-

ing data increases, more human subjectiveness and ordering inconsistencies will emerge, further straining the validity of a single global function.

Our idea is to explore a local learning approach for attribute ranking. The idea is to train a ranking function tailored to each novel pair of images $X_q = (x_r, x_s)$ that we wish to compare. We train the custom function using only a subset of all labeled training pairs, exploiting the data statistics in the neighborhood of the test pair. In particular, we sort all training pairs $\mathcal{P}_\mathcal{A}$ by their similarity to (x_r, x_s), then compose a local training set $\mathcal{P}'_\mathcal{A}$ consisting of the top K neighboring pairs, $\mathcal{P}'_\mathcal{A} = \{(x_{k1}, x_{k2})\}_{k=1}^{K}$. We explain in the next section how we define similarity between pairs. Then, we train a ranking function using Equation 2 on the fly, and apply it to compare the test images.

While simple, our framework directly addresses the flaws that hinder existing methods. By restricting training pairs to those visually similar to the test pair, the learner can zero in on features most important for that kind of comparison. Such a fine-grained approach helps to eliminate ordering constraints that are irrelevant to the test pair. For instance, when evaluating whether a high-topped athletic shoe is more or less *sporty* than a similar looking low-topped one, our method will exploit pairs with similar visual differences, as opposed to trying to accommodate in a single global function the contrasting sportiness of sneakers, high heels, and sandals (Fig. 4).

4.2 Selecting Fine-Grained Neighboring Pairs

A key factor to the success of the local rank learning approach is how we judge similarity between pairs. Intuitively, we would like to gather training pairs that are somehow *analogous* to the test pair, so that the ranker focuses on the fine-grained visual differences that dictate their comparison. This means that not only should individual members of the pairs have visual similarity, but also the visual contrasts between the two test pair images should mimic the visual contrasts between the two training pair images. In addition, we must account for the fact that we seek comparisons along a particular attribute, which means only certain aspects of the image appearance are relevant; in other words, Euclidean distance between their global image descriptors is likely inadequate.

To fulfill these desiderata, we define a paired distance function that incorporates attribute-specific metric learning. Let $X_q = (x_r, x_s)$ be the test pair, and let $X_t = (x_u, x_v)$ be a labeled training pair for which $(u, v) \in \mathcal{P}_\mathcal{A}$. We define their distance as:

$$D_\mathcal{A}(X_q, X_t) = \min\left(D'_\mathcal{A}((x_r, x_s), (x_u, x_v)), D'_\mathcal{A}((x_r, x_s), (x_v, x_u))\right), \tag{3}$$

where $D'_\mathcal{A}$ is the product of the two items' distances:

$$D'_\mathcal{A}((x_r, x_s), (x_u, x_v)) = d_\mathcal{A}(x_r, x_u) \times d_\mathcal{A}(x_s, x_v). \tag{4}$$

The product reflects that we are looking for pairs where each image is visually similar to one of those in the novel pair. It also ensures that the constraint

pairs are evaluated for distance as a pair instead of as individual images.[2] If both query-training couplings are similar, the distance is low. If some image coupling is highly dissimilar, the distance is greatly increased. The minimum in Equation 3 and the swapping of $(x_u, x_v) \rightarrow (x_v, x_u)$ in the second term ensure that we account for the unknown ordering of the test pair; while all training pairs are ordered with $R_{\mathcal{A}}(x_u) > R_{\mathcal{A}}(x_v)$, the first or second argument of X_q may exhibit the attribute more. When learning a local ranking function for attribute $\mathcal{A}$, we sort neighbor pairs for X_q according to $D_{\mathcal{A}}$, then take the top K to form $\mathcal{P}'_{\mathcal{A}}$.

When identifying neighbor pairs, rather than judge image distance $d_{\mathcal{A}}$ by the usual Euclidean distance on global descriptors, we want to specialize the function to the particular attribute at hand. That's because often a visual attribute does not rely equally on each dimension of the feature space, whether due to the features' locations or modality. For example, if judging image distance for the attribute *smiling*, the localized region by the mouth is likely most important; if judging distance for *comfort* the features describing color may be irrelevant. In short, it is not enough to find images that are globally visually similar. For fine-grained comparisons we need to focus on those that are similar in terms of the property of interest.

To this end, we learn a Mahalanobis metric:

$$d_{\mathcal{A}}(x_i, x_j) = (x_i - x_j)^T \mathbf{M}_{\mathcal{A}} (x_i - x_j), \tag{5}$$

parameterized by the $d \times d$ positive definite matrix $\mathbf{M}_{\mathcal{A}}$. We employ the information-theoretic metric learning (ITML) algorithm [15], due to its efficiency and kernelizability. Given an initial $d \times d$ matrix $\mathbf{M}_{\mathcal{A}_0}$ specifying any prior knowledge about how the data should be compared, ITML produces the $\mathbf{M}_{\mathcal{A}}$ that minimizes the LogDet divergence $D_{\ell \mathrm{d}}$ from that initial matrix, subject to constraints that similar data points be close and dissimilar points be far:

$$\begin{aligned} \min_{M_{\mathcal{A}} \succeq 0} \quad & D_{\ell \mathrm{d}}(\mathbf{M}_{\mathcal{A}}, \mathbf{M}_{\mathcal{A}_0}) \\ s.t. \quad & d_{\mathcal{A}}(x_i, x_j) \leq c \quad (i,j) \in \mathcal{S}_{\mathcal{A}} \\ & d_{\mathcal{A}}(x_i, x_j) \geq \ell \quad (i,j) \in \mathcal{D}_{\mathcal{A}}. \end{aligned} \tag{6}$$

The sets $\mathcal{S}_{\mathcal{A}}$ and $\mathcal{D}_{\mathcal{A}}$ consist of pairs of points constrained to be similar and dissimilar, and ℓ and c are large and small values, respectively, determined by the distribution of original distances. We set $\mathbf{M}_{\mathcal{A}_0} = \mathbf{\Sigma}^{-1}$, the inverse covariance matrix for the training images. To compose $\mathcal{S}_{\mathcal{A}}$ and $\mathcal{D}_{\mathcal{A}}$, we use image pairs for which human annotators found the images similar (or dissimilar) *according to the attribute* $\mathcal{A}$. While metric learning is usually used to enhance nearest neighbor classification (e.g., [23, 27]), we employ it to gauge perceived similarity along an attribute.

[2] A more strict definition of "analogous pair" would further constrain that there be low distortion between the vectors connecting the query pair and training pair, respectively, i.e., forming a parallelogram in the metric space. This is similarly efficient to implement. However, in practice, we found the stricter definition is slightly less effective than the product distance. This indicates that some variation in the intra-pair visual differences are useful to the learner.

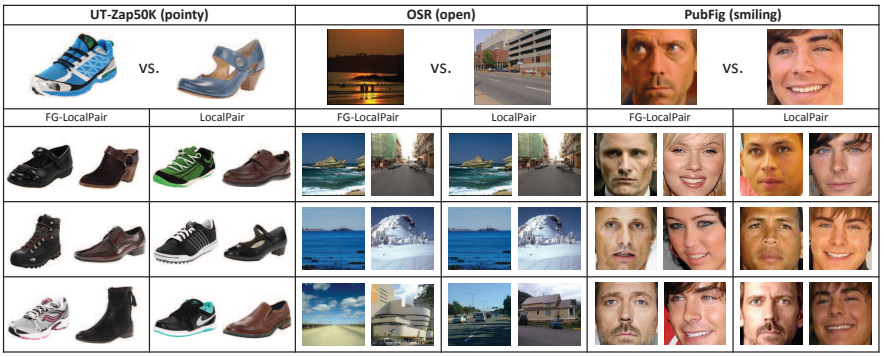


Fig. 5: Example fine-grained neighbor pairs for three test pairs (top row) from the datasets tested in this chapter. We display the top 3 pairs per query. FG-LocalPair and LocalPair denote results with and without metric learning (ML), respectively. **UT-Zap50K pointy**: ML puts the comparison focus on the tip of the shoe, caring less about the look of the shoe as a whole. **OSR open**: ML has less impact, as openness in these scenes relates to their whole texture. **PubFig smiling**: ML learns to focus on the mouth/lip region instead of the entire face. For example, while the LocalPair (non-learned) metric retrieves face pairs that more often contain the same people as the top pair, those instances are nonetheless less relevant for the fine-grained smiling distinction it requires. In contrast, our FG-LocalPair learned metric retrieves nearby pairs that may contain different people, yet are instances where the degree of smiling is most useful as a basis for predicting the relative smiling level in the novel query pair.

Figure 5 shows example neighbor pairs. They illustrate how our method finds training pairs analogous to the test pair, so the local learner can isolate the informative visual features for that comparison. Note how holistically, the neighbors found with metric learning (FG-LocalPair) may actually look less similar than those found without (LocalPair). However, in terms of the specific attribute, they better isolate the features that are relevant. For example, images of the same exact person need not be most useful to predict the degree of *smiling*, if others better matched to the test pair's expressions are available (last example). In practice, the local rankers trained with learned neighbors are substantially more accurate.

4.3 Fine-Grained Attribute Zappos Dataset

Having explained the basic approach, we now describe a new dataset amenable to fine-grained attributes. We collected a new UT Zappos50K dataset (**UT-Zap50K**[3]) specifically targeting the fine-grained attribute comparison task. The dataset is fine-grained due to two factors: 1) it focuses on a narrow domain of content, and 2) we develop a two-stage annotation procedure to isolate those comparisons that humans find perceptually very close.

[3] UT-Zap50K dataset and all related data are publicly available for download at `vision.cs.utexas.edu/projects/finegrained`

Fig. 6: Sample images from each of the high-level shoe categories of UT-Zap50K.

The image collection is created in the context of an online shopping task, with 50,000 catalog shoe images from Zappos.com. For online shopping, users care about precise visual differences between items. For instance, it is more likely that a shopper is deciding between two pairs of similar men's running shoes instead of between a woman's high heel and a man's slipper. The images are roughly 150×100 pixels and shoes are pictured in the same orientation for convenient analysis. For each image, we also collect its meta-data (shoe type, materials, manufacturer, gender, etc.) that are used to filter the shoes on Zappos.com.

Using Mechanical Turk (mTurk), we collect ground truth comparisons for 4 relative attributes: *open*, *pointy at the toe*, *sporty*, and *comfortable*. The attributes are selected for their potential to exhibit fine-grained differences. A worker is shown two images and an attribute name, and must make a relative decision (more, less, equal) and report the confidence of his decision (high, mid, low). We repeat the same comparison for 5 workers in order to vote on the final ground truth. We collect 12,000 total pairs, 3,000 per attribute. After removing the low confidence or agreement pairs, and "equal" pairs, each attribute has between 1,500 to 1,800 total ordered pairs.

Of all the possible $50,000^2$ pairs we could get annotated, we want to prioritize the fine-grained pairs. To this end, first, we sampled pairs with a strong bias (80%) towards intra-category and -gender images (based on the meta-data). We call this collection **UT-Zap50K-1**. We found $\sim$40% of the pairs came back labeled as "equal" for each attribute. While the "equal" label can indicate that there's no perceivable difference in the attribute, we also suspected that it was an easy fallback response for cases that required a little more thought—that is, those showing fine-grained differences. Thus, we next posted the pairs rated as "equal" (4,612 of them) back onto mTurk as new tasks, but *without* the "equal" option. We asked the workers to look closely, pick one image over the other, and give a one sentence rationale for their decisions. We call this set **UT-Zap50K-2**.

Interestingly, the workers are quite consistent on these pairs, despite their difficulty. Out of all 4,612 pairs, only 278 pairs had low confidence or agreement (and so were pruned). Overall, 63% of the fine-grained pairs (and 66% of the coarser pairs) had at least 4 out of 5 workers agree on the same answer with above average confidence. This consistency ensures we have a dataset that is both fine-grained as well as reliably ground truthed.

Compared to an existing Shoes attribute dataset [5] with relative attributes [34], UT-Zap50K is about $3.5\times$ larger, offers meta-data and $10\times$ more comparative labels, and most importantly, specifically targets fine-grained tasks. Compared to ex-

isting popular relative attribute datasets like PubFig [36] and Outdoor Scenes [42], which contain only category-level comparisons (e.g., "Viggo *smiles* less than Miley") that are propagated down uniformly to all image instances, UT-Zap50K is distinct in that annotators have made *image-level* comparisons (e.g., "this particular shoe image is *more pointy* than that particular shoe"). The latter is more costly to obtain but essential for testing fine-grained attributes thoroughly.

In the next section we use UT-Zap50K as well as other existing datasets to test our approach. Later in Section 5 we will discuss extensions to the UT-Zap50K annotations that make it suitable for the just noticeable difference task as well.

4.4 Experiments and Results

To validate our method, we compare it to two state-of-the-art methods as well as informative baselines.

4.4.1 Experimental Setup

Datasets We evaluate on three datasets: **UT-Zap50K**, as defined above, with concatenated GIST and color histogram features; the Outdoor Scene Recognition dataset [42] (**OSR**); and a subset of the Public Figures faces dataset [36] (**PubFig**). OSR contains 2,688 images (GIST features) with 6 attributes, while PubFig contains 772 images (GIST + Color features) with 11 attributes. We use the exact same attributes, features, and train/test splits as [38, 43]. Our choice of features is based on the intent to capture spatially localized textures (GIST) as well as global color distributions, though of course alternative feature types could easily be employed in our framework.

Setup We run for 10 random train/test splits, setting aside 300 ground truth pairs for testing and the rest for training. We cross-validate C for all experiments, and adopt the same C selected by the global baseline for our approach. We use no "equal" pairs for training or testing rankers. We report accuracy in terms of the percentage of correctly ordered pairs, following [38]. We present results using the same labeled data for all methods.

For learning to rank, our *total* training pairs $\mathcal{P}_\mathcal{A}$ consist of only ordered pairs $\mathcal{P}_o$. For ITML, we use the ordered pairs $\mathcal{P}_\mathcal{A}$ for rank training to compose the set of dissimilar pairs $\mathcal{D}_\mathcal{A}$, and the set of "equal" pairs to compose the similar pairs $\mathcal{S}_\mathcal{A}$. We use the default settings for c and ℓ in the authors' code [15]. The setting of K determines "how local" the learner is; its optimal setting depends on the training data and query. As in prior work [7, 57], we simply fix it for all queries at $K = 100$ (though see Sect. 4.5 for a proposed generalization that learns the neighborhood size as well). Values of $K = 50$ to 200 give similar results.

Baselines We compare the following methods:

Table 1: Results for the UT-Zap50K dataset.

	Open	Pointy	Sporty	Comfort
Global [43]	87.77	89.37	91.20	89.93
RandPair	82.53	83.70	86.30	84.77
LocalPair	88.53	88.87	92.20	90.90
FG-LocalPair	**90.67**	**90.83**	**92.67**	**92.37**

(a) UT-Zap50K-1 with ***coarser*** pairs.

	Open	Pointy	Sporty	Comfort
Global [43]	60.18	59.56	62.70	**64.04**
RandPair	61.00	53.41	58.26	59.24
LocalPair	71.64	59.56	61.22	59.75
FG-LocalPair	**74.91**	**63.74**	**64.54**	62.51

(b) UT-Zap50K-2 with ***fine-grained*** pairs.

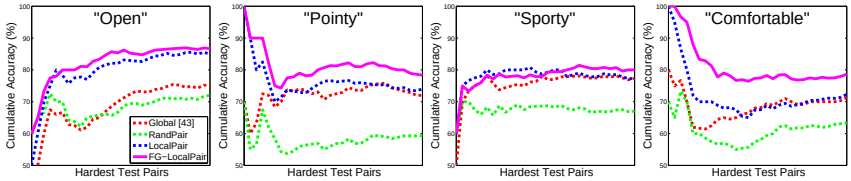


Fig. 7: Accuracy for the 30 hardest test pairs on UT-Zap50K-1.

- **FG-LocalPair**: the proposed fine-grained approach.
- **LocalPair**: our approach without the learned metric (i.e., $\mathbf{M}_{\mathcal{A}} = \mathbb{I}$). This baseline isolates the impact of tailoring the search for neighboring pairs to the attribute.
- **RandPair**: a local approach that selects its neighbors randomly. This baseline demonstrates the importance of selecting relevant neighbors.
- **Global**: a global ranker trained with all available labeled pairs, using Equation 2. This is the Relative Attributes method [43]. We use the authors' public code.
- **RelTree**: the non-linear relative attributes approach of [38], which learns a hierarchy of functions, each trained with successively smaller subsets of the data. Code is not available, so we rely on the authors' reported numbers (available for OSR and PubFig).

4.4.2 Zappos Results

Table 1a shows the accuracy on UT-Zap50K-1. Our method outperforms all baselines for all attributes. To isolate the more difficult pairs in UT-Zap50K-1, we sort the test pairs by their intra-pair distance using the learned metric; those that are close will be visually similar for the attribute, and hence more challenging. Figure 7 shows the results, plotting cumulative accuracy for the 30 hardest test pairs per split. We see that our method has substantial gains over the baselines (about 20%), demonstrating its strong advantage for detecting subtle differences. Figure 8 shows some qualitative results.

We proceed to test on even more difficult pairs. Whereas Figure 7 focuses on pairs difficult according to the learned metric, next we focus on pairs difficult according to our human annotators. Table 1b shows the results for UT-Zap50K-2. We use the original ordered pairs for training and all 4,612 fine-grained pairs for testing (Sect. 4.3). We outperform all methods for 3 of the 4 attributes. For the two more

Fig. 8: Example pairs contrasting our predictions to the Global baseline's. In each pair, the top item is *more sporty* than the bottom item according to ground truth from human annotators. (1) We predict correctly, Global is wrong. We detect subtle changes, while Global relies only on overall shape and color. (2) We predict incorrectly, Global is right. These coarser differences are sufficiently captured by a global model. (3) Both methods predict incorrectly. Such pairs are so fine-grained, they are difficult even for humans to make a firm decision.

objective attributes, *open* and *pointy*, our gains are sizeable—14% over Global for *open*. We attribute this to their localized nature, which is accurately captured by our learned metrics. No matter how fine-grained the difference is, it usually comes down to the top of the shoe (*open*) or the tip of the shoe (*pointy*). On the other hand, the subjective attributes are much less localized. The most challenging one is *comfort*, where our method performs slightly worse than Global, in spite of being better on the coarser pairs (Table 1a). We think this is because the locations of the subtleties vary greatly per pair.

4.4.3 Scenes and PubFig Results

We now shift our attention to OSR and PubFig, two commonly used datasets for relative attributes [34, 38, 43]. The paired supervision for these datasets originates from category-wise comparisons [43], and as such there are many more training pairs—on average over 20,000 per attribute.

Tables 2 and 3 show the accuracy for PubFig and OSR, respectively. See [54] for attribute-specific precision recall curves. On both datasets, our method outperforms all the baselines. Most notably, it outperforms RelTree [38], which to our knowledge is the very best accuracy reported to date on these datasets. This particular result is compelling not only because we improve the state-of-the-art, but also because RelTree is a non-linear ranking function. Hence, we see that local learning with linear models is performing better than global learning with a non-linear model. With a lower capacity model, but the "right" training examples, the comparison is better learned. Our advantage over the global Relative Attributes linear model [43] is even greater.

On OSR, RandPair comes close to Global. One possible cause is the weak supervision from the category-wise constraints. While there are 20,000 pairs, they are less diverse. Therefore, a random sampling of 100 neighbors seems to reasonably

Table 2: Accuracy comparison for the OSR dataset. FG-LocalPair denotes the proposed approach.

	Natural	**Open**	**Perspective**	**LargeSize**	**Diagonal**	**CloseDepth**
RelTree [38]	95.24	92.39	87.58	88.34	89.34	89.54
Global [43]	95.03	90.77	86.73	86.23	86.50	87.53
RandPair	92.97	89.40	84.80	84.67	84.27	85.47
LocalPair	94.63	93.27	88.33	89.40	90.70	89.53
FG-LocalPair	**95.70**	**94.10**	**90.43**	**91.10**	**92.43**	**90.47**

Table 3: Accuracy comparison for the PubFig dataset.

	Male	**White**	**Young**	**Smiling**	**Chubby**	**Forehead**	**Eyebrow**	**Eye**	**Nose**	**Lip**	**Face**
RelTree [38]	85.33	82.59	84.41	83.36	78.97	88.83	81.84	83.15	80.43	81.87	86.31
Global [43]	81.80	76.97	83.20	79.90	76.27	87.60	79.87	81.67	77.40	79.17	82.33
RandPair	74.43	65.17	74.93	73.57	69.00	84.00	70.90	73.70	66.13	71.77	73.50
LocalPair	81.53	77.13	83.53	82.60	78.70	89.40	80.63	82.40	78.17	79.77	82.13
FG-LocalPair	**91.77**	**87.43**	**91.87**	**87.00**	**87.37**	**94.00**	**89.83**	**91.40**	**89.07**	**90.43**	**86.70**

mimic the performance when using all pairs. In contrast, our method is consistently stronger, showing the value of our learned neighborhood pairs.

While metric learning (ML) is valuable across the board (FG-LocalPair $>$ LocalPair), it has more impact on PubFig than OSR. We attribute this to PubFig's more localized attributes. Subtle differences are what makes fine-grained comparison tasks hard. ML discovers the features behind those subtleties *with respect to each attribute*. Those features could be spatially localized regions or particular image cues (GIST vs. color). Indeed, our biggest gains compared to LocalPair (9% or more) are on *white*, where we learn to emphasize color bins, or *eye*/*nose*, where we learn to emphasize the GIST cells for the part regions. In contrast, the LocalPair method compares the face images as a whole, and is liable to find images of the same person as more relevant, regardless of their properties in that image (Fig. 5).

4.4.4 Runtime Evaluation

Learning local models on the fly, though more accurate for fine-grained attributes, does come at a computational cost. The main online costs are finding the nearest neighbor pairs and training the local ranking function. For our datasets, with $K = 100$ and 20,000 total labeled pairs, this amounts to about 3 seconds. There are straightforward ways to improve the run-time. The neighbor finding can be done rapidly using well known hashing techniques, which are applicable to learned metrics [27]. Furthermore, we could pre-compute a set of representative local models. For example, we could cluster the training pairs, build a local model for each cluster, and invoke the suitable model based on a test pair's similarity to the cluster representatives. We leave such implementation extensions as future work.

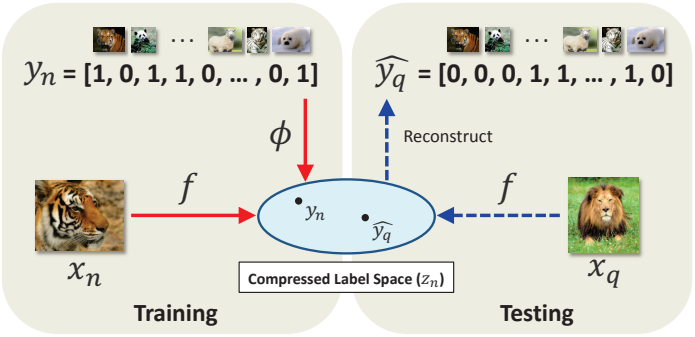


Fig. 9: Overview of our compressed sensing based approach. y_n and $\hat{y_q}$ represent the M-dimensional neighborhood indicator vectors for a training and testing instance, respectively. ϕ is a $D \times M$ random matrix where D denotes the compressed indicators' dimensionality. f is the learned regression function used to map the original image feature space to the compressed label space. By reconstructing back to the full label space, we get an estimate of $\hat{y_q}$ indicating which labeled training instances together will form a good neighborhood for the test instance x_q.

4.5 Predicting Useful Neighborhoods

This section expands on the neighbor selection approach described in Section 4.2, briefly summarizing our NIPS 2014 paper [55]. Please see that paper for more details and results.

As we have seen above, the goal of local learning is to tailor the model to the properties of the data surrounding the test instance. However, so far, like other prior work in local learning we have made an important core assumption: that the instances most *useful* for building a local model are those that are *nearest* to the test example. This assumption is well-motivated by the factors discussed above, in terms of data density and intra-class variation. Furthermore, as we saw above, identifying training examples solely based on proximity has the appeal of permitting specialized similarity functions (whether learned or engineered for the problem domain), which can be valuable for good results, especially in structured input spaces.

On the other hand, there is a problem with this core assumption. By treating the individual nearness of training points as a metric of their utility for local training, existing methods fail to model how those training points will actually be employed. Namely, the relative success of a locally trained model is a function of the entire *set* or *distribution* of the selected data points—not simply the individual pointwise nearness of each one against the query. In other words, the ideal target subset consists of a set of instances that together yield a good predictive model for the test instance. Thus, local neighborhood selection ought to be considered jointly among training points.

Based on this observation, we have explored ways to *learn* the properties of a "good neighborhood". We cast the problem in terms of large-scale multi-label classification, where we learn a mapping from an individual instance to an indicator vector over the entire training set that specifies which instances are jointly useful to

the query. The approach maintains an inherent bias towards neighborhoods that are local, yet makes it possible to discover subsets that (i) deviate from a strict nearest-neighbor ranking and (ii) vary in size. We stress that learning what a good *neighbor* looks like (metric learning's goal) is distinct from learning what a good *neighborhood* looks like (our goal). Whereas a metric can be trained with pairwise constraints indicating what should be near or far, jointly predicting the instances that ought to compose a neighborhood requires a distinct form of learning.

The overall pipeline includes three main phases, shown in Figure 9. (1) First, we devise an empirical approach to generate ground truth training neighborhoods (x_n, y_n) that consist of an individual instance x_n paired with a set of training instance indices capturing its target "neighbors", the latter being represented as a M-dimensional indicator vector y_n, where M is the number of labeled training instances. (2) Next, using the Bayesian compressed sensing approach of [30], we project y_n to a lower-dimensional compressed label space z_n using a random matrix ϕ. Then, we learn regression functions $f_1(x_n), ..., f_D(x_n)$ to map the original features x_n to the compressed label space. (3) Finally, given a test instance x_q, we predicts its neighborhood indicator vector $\hat{y_q}$ using ϕ and the learned regression functions f. We use this neighborhood of points to train a classifier on the fly, which in turn is used to categorize x_q.[4]

In [55] we show substantial advantages over existing local learning strategies, particularly when attributes are multi-modal and/or its similar instances are difficult to match based on global feature distances alone. Our results illustrate the value in estimating the size and composition of discriminative neighborhoods, rather than relying on proximity alone. See our paper for the full details [55].

5 Just Noticeable Differences

Having established the strength of local learning for fine-grained attribute comparisons, we now turn to task of predicting when a comparison is even possible. In other words, given a pair of images, the output may be one of "more", "less", or "equal".

While some pairs of images have a clear ordering for an attribute (recall Fig. 2), for others the difference may be indistinguishable to human observers. Attempting to map relative attribute ranks to equality predictions is non-trivial, particularly since the span of indistinguishable pairs in an attribute space may vary in different parts of the feature space. In fact, as discussed above, despite the occasional use of unordered pairs for training[5], it is assumed in prior work that all test images will be orderable. However, the real-valued output of a ranking function as trained in Section 3 will virtually never be equal for two distinct inputs. Therefore, even though existing

[4] Note that the neighborhood learning idea has been tested thus far only for classification tasks, though in principle applies similarly to ranking tasks.

[5] Empirically, we found the inclusion of unordered pairs during training in [43] to have negligible impact at test time.

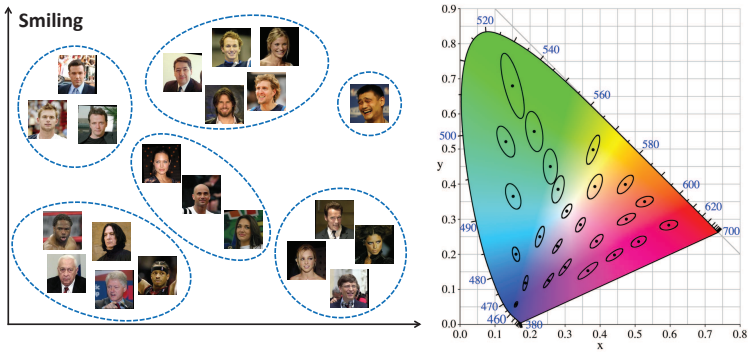


Fig. 10: Analogous to the MacAdam ellipses in the CIE x,y color space (right) [21], relative attribute space is likely not uniform (left). That is, the regions within which attribute differences are indistinguishable may vary in size and orientation across the high-dimensional visual feature space. Here we see the faces within each "equally *smiling*" cluster exhibit varying qualities for differentiating smiles—such as age, gender, and visibility of the teeth—but are still difficult or impossible to order in terms of *smiling-ness*. As a result, simple metrics and thresholds on attribute differences are insufficient to detect just noticeable differences.

methods may learn to produce similar rank scores for equal pairs, it is unclear how to determine when a novel pair is "close enough" to be considered un-orderable.

We argue that this situation calls for a model of *just noticeable difference* among attributes. Just noticeable difference (JND) is a concept from psychophysics. It refers to the amount a stimulus has to be changed in order for it to be detectable by human observers at least half the time. For example, JND is of interest in color perception (which light sources are perceived as the same color?) and image quality assessment (up to what level of compression do the images look ok?). JNDs are determined empirically through tests of human perception. For example, JND in color can be determined by gradually altering the light source just until the human subject detects that the color has changed [21].

Why is it challenging to develop a computational model of JND for relative attributes? At a glance, one might think it amounts to learning an optimal threshold on the difference of predicted attribute strengths. However, this begs the question of how one might properly and densely sample real images of a complex attribute (like *seriousness*) to gradually walk along the spectrum, so as to discover the right threshold with human input. More importantly, an attribute space need not be *uniform*. That is, depending on where we look in the feature space, the magnitude of attribute difference required to register a perceptible change may vary. Therefore, the simplistic "global threshold" idea falls short. Analogous issues also arise in color spaces, e.g., the famous MacAdam ellipses spanning indistinguishable colors in the CIE x,y color space vary markedly in their size and orientation depending on where in the feature space one looks (leading to the crafting of color spaces like CIE Lab that are more uniform). See Figure 10.

We next introduce a solution to infer when two images are indistinguishable for a given attribute. Continuing with the theme of local learning, we develop a Bayesian

approach that relies on *local* statistics of orderability. Our approach leverages both a low-level visual descriptor space, within which image pair proximity is learned, as well as a mid-level visual attribute space, within which attribute distinguishability is represented (Fig. 11). Whereas past ranking models have attempted to integrate equality into *training*, none attempt to distinguish between orderable and un-orderable pairs at test time.

Our method works as follows. First, we construct a predicted attribute space using the standard relative attribute framework (Sect. 3). Then, on top of that model, we combine a likelihood computed in the predicted attribute space (Sect. 5.1.1) with a local prior computed in the original image feature space (Sect. 5.1.2). We show our approach's superior performance compared to various baselines for detecting noticeable differences, as well as demonstrate how attribute JND has potential benefits for an image search application (Sect. 5.2).

5.1 Local Bayesian Model of Distinguishability

The most straightforward approach to infer whether a novel image pair is distinguishable would be to impose a threshold on their rank differences, i.e., to predict "indistinguishable" if $|R_{\mathcal{A}}(x_r) - R_{\mathcal{A}}(x_s)| \leq \varepsilon$. The problem is that unless the rank space is uniform, a global threshold ε is inadequate. In other words, the rank margin for indistinguishable pairs need not be constant across the entire feature space. By testing multiple variants of this basic idea, our empirical results confirm this is indeed an issue, as we will see in Section 5.2.

Our key insight is to formulate distinguishability prediction in a probabilistic, local learning manner. Mindful of the non-uniformity of relative attribute space, our approach uses distributions tailored to the data in the proximity of a novel test pair. Furthermore, we treat the relative attribute ranks as an imperfect mid-level representation on top of which we can learn to target the actual (sparse) human judgments about distinguishability.

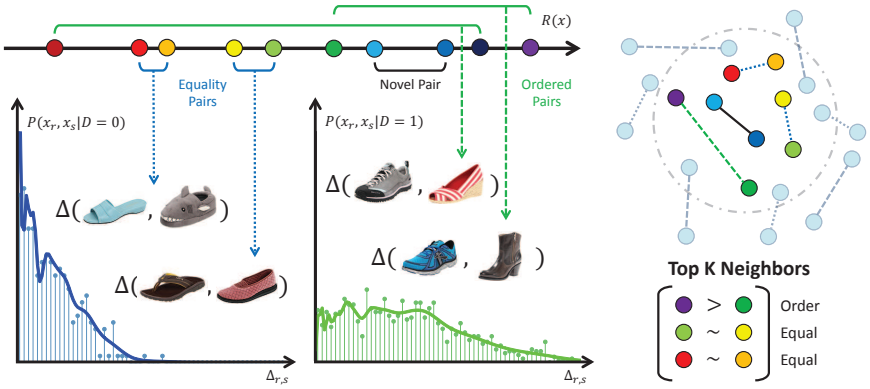
Let $D \in \{0,1\}$ be a binary random variable representing the distinguishability of an image pair. For a distinguishable pair, $D = 1$. Given a novel test pair (x_r, x_s), we are interested in the posterior:

$$P(D|x_r, x_s) \propto P(x_r, x_s|D)P(D), \tag{7}$$

to estimate how likely two images are distinguishable. To make a hard decision we take the maximum a posteriori estimate over the two classes:

$$d^* = \arg\max_d P(D = d|x_r, x_s). \tag{8}$$

At test time, our method performs a two-stage cascade. If the test pair appears distinguishable, we return the response "more" or "less" according to whether $R_{\mathcal{A}}(x_r) < R_{\mathcal{A}}(x_s)$ (where R is trained in either a global or local manner). Otherwise,



(a) Likelihood Term (b) Prior Term

Fig. 11: Overview of our Bayesian approach. (1) Learn a ranking function $R_{\mathcal{A}}$ using all annotated training pairs (Sect. 3), as depicted in Figure 3. (2) Estimate the likelihood densities of the equal and ordered pairs, respectively, using the pairwise distances in relative attribute space. (3) Determine the local prior by counting the labels of the analogous pairs in the image descriptor space. (4) Combine the results to predict whether the novel pair is distinguishable (not depicted). Best viewed in color.

we say the test pair is indistinguishable. In this way we unify relative attributes with JND, generating partially ordered predictions in spite of the ranker's inherent totally ordered outputs.

Next, we derive models for the likelihood and prior in Equation 7, accounting for the challenges described above.

5.1.1 Likelihood Model

We use a kernel density estimator (KDE) to represent the distinguishability likelihood over image pairs. The likelihood captures the link between the observed rank differences and the human-judged just noticeable differences.

Let $\Delta_{r,s}$ denote the difference in attribute ranks for images r and s:

$$\Delta_{r,s} = |R_{\mathcal{A}}(x_r) - R_{\mathcal{A}}(x_s)|. \tag{9}$$

Recall that $\mathcal{P}_o$ and $\mathcal{P}_e$ refer to the sets of ordered and equal training image pairs, respectively. We compute the rank differences for all training pairs in $\mathcal{P}_o$ and $\mathcal{P}_e$, and fit a non-parametric Parzen density:

$$P(x_r, x_s|D) = \frac{1}{|\mathcal{P}|} \sum_{i,j \in \mathcal{P}} K_h\left(\Delta_{i,j} - \Delta_{r,s}\right), \tag{10}$$

for each set in turn. Here $\mathcal{P}$ refers to the ordered pairs $\mathcal{P}_o$ when representing distinguishability ($D = 1$), and the equal pairs $\mathcal{P}_e$ when representing indistinguishability ($D = 0$). The Parzen density estimator [44] superimposes a kernel function K_h at each data pair. In our implementation, we use Gaussian kernels. It integrates local

estimates of the distribution and resists overfitting. The KDE has a smoothing parameter h that controls the model complexity. To ensure that all density is contained within the positive absolute margins, we apply a positive support to the estimator. Namely, we transform $\Delta_{i,j}$ using a log function, estimate the density of the transformed values, and then transform back to the original scale. See (a) in Figure 11.

The likelihood reflects how well the equal and ordered pairs are separated in the attribute space. However, critically, $P(x_r, x_s|D = 1)$ need not decrease monotonically as a function of rank differences. In other words, the model permits returning a higher likelihood for certain pairs separated by smaller margins. This is a direct consequence of our choice of the non-parametric KDE, which preserves local models of the original training data. This is valuable for our problem setting because in principle it means our method can correct imperfections in the original learned ranks and account for the non-uniformity of the space.

5.1.2 Prior Model

Finally, we need to represent the prior over distinguishability. The prior could simply count the training pairs, i.e., let $P(D = 1)$ be the fraction of all training pairs that were distinguishable. However, we again aim to account for the non-uniformity of the visual feature space. Thus, we estimate the prior based only on a subset of data near the input images. Intuitively, this achieves a simple prior for the label distribution in multiple pockets of the feature space:

$$P(D = 1) = \frac{1}{K}|\mathcal{P}'_o|, \tag{11}$$

where $\mathcal{P}'_o \subset \mathcal{P}_o$ denotes the set of K neighboring ordered training pairs. $P(D = 0)$ is defined similarly for the indistinguishable pairs $\mathcal{P}_e$. Note that while the likelihood is computed over the pair's rank difference, the locality of the prior is with respect to the image descriptor space. See (b) in Figure 11.

To localize the relevant pocket of the image space, we adopt the metric learning strategy detailed in Section 4.2. Using the learned metric, pairs analogous to the novel input (x_r, x_s) are retrieved based on a product of their individual Mahalanobis distances, so as to find pairs whose members both align.

5.2 Experiments and Results

We present results on the core JND detection task (Sect. 5.2.2) on two challenging datasets and demonstrate its impact for an image search application (Sect. 5.2.3).

5.2.1 Experimental Setup

Datasets and Ground Truth Our task requires attribute datasets that (1) have instance-level relative supervision, meaning annotators were asked to judge attribute

comparisons on individual pairs of images, not object categories as a whole and (2) have pairs labeled as "equal" and "more/less". To our knowledge, our UT-Zap50K and LFW-10 [47] are the only existing datasets satisfying those conditions.

To train and evaluate just noticeable differences, we must have annotations of utmost precision. Therefore, we take extra care in establishing the (in)distinguishable ground truth for both datasets. We perform pre-processing steps to discard unreliable pairs, as we explain next. This decreases the total volume of available data, but it is essential to have trustworthy results.

The **UT-Zap50K** dataset is detailed in Section 4.3. As ordered pairs $\mathcal{P}_o$, we use all coarse and fine-grained pairs for which all 5 workers agreed and had high confidence. Even though the fine-grained pairs might be visually similar, if all 5 workers could come to agreement with high confidence, then the images are most likely distinguishable. As equal pairs $\mathcal{P}_e$, we use all fine-grained pairs with 3 or 4 workers in agreement and only medium confidence. Since the fine-grained pairs have already been presented to the workers twice, if the workers are still unable to come to an consensus with high confidence, then the images are most likely indistinguishable. The resulting dataset has 4,778 total annotated pairs, consisting of on average 800 ordered and 350 indistinguishable (equal) pairs per attribute.

The **LFW-10** dataset [47] consists of 2,000 face images, taken from the Labeled Faces in the Wild [26] dataset.[6] It contains 10 relative attributes, like *smiling*, *big eyes*, etc., with 1,000 labeled pairs each. Each pair was labeled by 5 people. As ordered pairs $\mathcal{P}_o$, we use all pairs labeled "more" or "less" by at least 4 workers. As equal pairs $\mathcal{P}_e$, we use pairs where at least 4 workers said "equal", as well as pairs with the same number of "more" and "less" votes. The latter reflects that a split in decision signals indistinguishability. Due to the smaller scale of LFW-10, we could not perform as strict of a pre-processing step as in UT-Zap50K; requiring full agreement on ordered pairs would eliminate most of the labeled data. The resulting dataset has 5,543 total annotated pairs, on average 230 ordered and 320 indistinguishable pairs per attribute.

Baselines We are the first to address the attribute JND task. No prior methods infer indistinguishability at test time [32, 38, 43, 46, 47]. Therefore, we develop multiple baselines to compare to our approach:

- **Rank Margin**: Use the magnitude of $\Delta_{r,s}$ as a confidence measure that the pair r,s is distinguishable. This baseline assumes the learned rank function produces a uniform feature space, such that a *global threshold* on rank margins would be sufficient to identify indistinguishable pairs. To compute a hard decision for this method (for F1-scores), we threshold the Parzen window likelihood estimated from the training pairs by ε, the mid-point of the likelihood means.
- **Logistic Classifier** [32]: Train a logistic regression classifier to distinguish training pairs in $\mathcal{P}_o$ from those in $\mathcal{P}_e$, where the pairs are represented by their rank differences $\Delta_{i,j}$. To compute a hard decision, we threshold the posterior at 0.5. This is the method used in [32] to obtain a probabilistic measure of attribute

[6] `cvit.iiit.ac.in/projects/relativeParts`

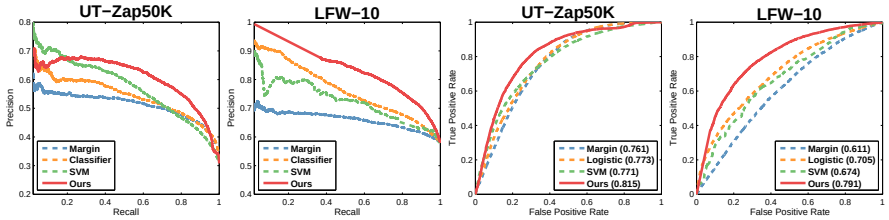


Fig. 12: Just noticeable difference detection accuracy for all attributes. We show the precision-recall (top row) and ROC curves (bottom row) for the shoes (left) and faces (right) datasets. Legends show AUC values for ROC curves. Note that the Mean Shift baseline does not appear here, since it does not produce confidence values.

equality. It is the closest attempt we can find in the literature to represent equality predictions, though the authors do not evaluate its accuracy. This baseline also maintains a global view of attribute space.

- **SVM Classifier**: Train a nonlinear SVM classifier with a RBF kernel to distinguish ordered and equal pairs. We encode pairs of images as single points by concatenating their image descriptors. To ensure symmetry, we include training instances with the two images in either order.[7]
- **Mean Shift**: Perform mean shift clustering on the predicted attribute scores $R_{\mathcal{A}}(x_i)$ for all training images. Images falling in the same cluster are deemed indistinguishable. Since mean shift clusters can vary in size, this baseline does *not* assume a uniform space. Though unlike our method, it fails to leverage distinguishability supervision as it processes the ranker outputs.

Implementation Details For UT-Zap50K, we use 960-dim GIST and 30-bin Lab color histograms as image descriptors. For LFW-10, they are 8,300-dim part-based features learned on top of dense SIFT bag of words features (provided by the authors). We reduce their dimensionality to 100 with PCA to prevent overfitting. The part-based features [47] isolate localized regions of the face (e.g., exposing cues specific to the eyes vs. hair). We experimented with both linear and RBF kernels for $R_{\mathcal{A}}$. Since initial results were similar, we use linear kernels for efficiency. We use Gaussian kernels for the Parzen windows. We set all hyperparameters (h for the KDE, bandwidth for Mean Shift, K for the prior) on held-out validation data. To maximize the use of training data, in all results below, we use leave-one-out evaluation and report results over 4 folds of random training-validation splits.

Table 4: JND detection on UT-Zap50K (F1 scores).

	Open	**Pointy**	**Sporty**	**Comf.**	**All Attributes**
Margin	48.95	67.48	66.93	57.09	60.11 ± 1.89
Logistic	10.49	62.95	63.04	45.76	45.56 ± 4.13
SVM	48.82	50.97	47.60	40.12	46.88 ± 5.73
M. Shift	54.14	58.23	60.76	**61.60**	58.68 ± 8.01
Ours	**62.02**	**69.45**	**68.89**	54.63	$\mathbf{63.75 \pm 3.02}$

Table 5: JND detection on LFW-10 (F1 scores). NaN occurs when recall=0 and precision=inf.

	Bald	**DarkHair**	**BigEyes**	**GdLook**	**Masc.**	**Mouth**	**Smile**	**Teeth**	**Forehead**	**Young**	**All Attributes**
Margin	71.10	55.81	74.16	61.36	82.38	62.89	60.56	65.26	67.49	**34.20**	63.52 ± 2.67
Logistic	75.77	53.26	86.71	64.27	87.29	63.41	59.66	64.83	75.00	NaN	63.02 ± 1.84
SVM	79.06	32.43	89.70	70.98	87.35	70.27	55.01	39.09	79.74	NaN	60.36 ± 9.81
M. Shift	66.37	56.69	54.50	51.29	69.73	68.38	61.34	65.73	73.99	23.19	59.12 ± 10.51
Ours	**81.75**	**69.03**	**89.59**	**75.79**	**89.86**	**72.69**	**73.30**	**74.80**	**80.49**	32.89	$\mathbf{74.02 \pm 1.66}$

5.2.2 Just Noticeable Difference Detection

We evaluate just noticeable difference detection accuracy for all methods on both datasets. Figure 12 shows the precision-recall curves and ROC curves, where we pool the results from all 4 and 10 attributes in UT-Zap50K and LFW-10, respectively. Tables 4 and 5 report the summary F1-scores and standard deviations for each individual attribute. The F1-score is a useful summary statistic for our data due to the unbalanced nature of the test set: 25% of the shoe pairs and 80% of the face pairs are indistinguishable for some attribute.

Overall, our method outperforms all baselines. We obtain sizeable gains—roughly 4-18% on UT-Zap50K and 10-15% on LFW-10. This clearly demonstrates the advantages of our local learning approach, which accounts for the non-uniformity of attribute space. The "global approaches", Rank Margin and Logistic Classifier, reveal that a uniform mapping of the relative attribute predictions is insufficient. In spite of the fact that they include equal pairs during training, simply assigning similar scores to indistinguishable pairs is inadequate. Their weakness is likely due both to noise in those mid-level predictions as well as the existence of JND regions that vary in scale. Furthermore, the results show that even for challenging, realistic image data, we can identify just noticeable differences at a high precision and recall, up to nearly 90% in some cases.

The SVM baseline is much weaker than our approach, indicating that discriminatively learning what indistinguishable image pairs look like is insufficient. This result underscores the difficulty of learning subtle differences in a high-dimensional image descriptor space, and supports our use of the compact rank space for our likelihood model.

Looking at the per-attribute results (Tables 4 and 5), we see that our method also outperforms the Mean Shift baseline. While Mean Shift captures dominant clusters in the spectrum of predicted attribute ranks for certain attributes, for others (like *pointy* or *masculine*) we find that the distribution of output predictions are more

[7] We also implemented other encoding variants, such as taking the difference of the image descriptors or using the predicted attribute scores $R_{\mathcal{A}}(x_i)$ as features, and they performed similarly or worse.

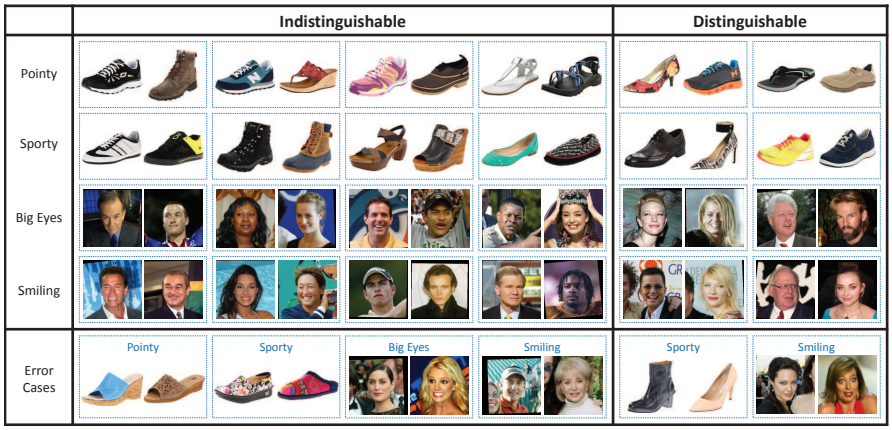


Fig. 13: Example predictions. The top four rows are pairs our method correctly classifies as indistinguishable (left panel) and distinguishable (right panel), whereas the Rank Margin baseline fails. Each row shows pairs for a particular attribute. The bottom row shows failure cases by our method; i.e., the bottom left pair is indistinguishable for pointiness, but we predict distinguishable.

Fig. 14: Example just noticeable differences. In each row, we take leftmost image as a starting point, then walk through nearest neighbors in relative attribute space until we hit an image that is distinguishable, as predicted by our method. For example, in row 2, our method finds the left block of images to be indistinguishable for *sportiness*; it flags the transition from the flat dress shoe to the pink "loafer-like sneaker" as being a noticeable difference.

evenly spread. Despite the fact that the rankers are optimized to minimize margins for equal pairs, simple post-processing of their outputs is inadequate.

We also see that that our method is nearly always best, except for two attributes: *comfort* in UT-Zap50K and *young* in LFW-10. Of the shoe attributes, *comfort* is perhaps the most subjective; we suspect that all methods may have suffered due to label noise for that attribute. While *young* would not appear to be subjective, it is clearly a more difficult attribute to learn. This makes sense, as youth would be a function of multiple subtle visual cues like face shape, skin texture, hair color, etc., whereas something like *baldness* or *smiling* has a better visual focus captured well by the part features of [47]. Indeed, upon inspection we find that the likelihoods insufficiently separate the equal and distinguishable pairs. For similar reasons, the Logistic Classifier baseline [32] fails dramatically on both *open* and *young*.

Figure 13 shows qualitative prediction examples. Here we see the subtleties of JND. Whereas past methods would be artificially forced to make a comparison for the left panel of image pairs, our method declares them indistinguishable. Pairs may look very different overall (e.g., different hair, race, headgear) yet still be indistin-

Fig. 15: The modified WhittleSearch framework. The user can now express an "equality" feedback, speeding up the process of finding his envisioned target.

guishable *in the context of a specific attribute*. Meanwhile, those that are distinguishable (right panel) may have only subtle differences.

Figure 14 illustrates examples of just noticeable difference "trajectories" computed by our method. We see how our method can correctly predict that various instances are indistinguishable, even though the raw images can be quite diverse (e.g., a strappy sandal and a flat dress shoe are equally *sporty*). Similarly, it can detect a difference even when the image pair is fairly similar (e.g., a lace-up sneaker and smooth-front sneaker are distinguishable for *openness* even though the shapes are close).

Figure 16 displays 2D t-SNE [40] embeddings for a subset of 5,000 shoe images based on the original image feature space and our learned attribute space for the attribute *pointy*. To compute the embeddings for our method, we represent each image x_i by its posterior probabilities of being indistinguishable to every other image. i.e. $v(x_i) = [P(D = 0|x_i, x_1), P(D = 0|x_i, x_2), ..., P(D = 0|x_i, x_N)]$ where N is the total number of images in the embedding. We see that while the former produces a rather evenly distributed mapping without distinct structures, the latter produces a mapping containing unique structures along with "pockets" of indistinguishable images. Such structures precisely reflect the non-uniformity we pointed out in Figure 10.

5.2.3 Image Search Application

Finally, we demonstrate how JND detection can enhance an image search application. Specifically, we incorporate our model into the WhittleSearch framework of Kovashka et al. [34], overviewed in Chapter XXXXX of this book. WhittleSearch is an interactive method that allows a user to provide relative attribute feedback, e.g., by telling the system that he wants images "more *sporty*" than some reference image. The method works by intersecting the relative attribute constraints, scoring database images by how many constraints they satisfy, then displaying the top scoring images for the user to review. See [34] for details.

We augment that pipeline such that the user can express not only "more/less" preferences, but also "equal" preferences (Fig. 15). For example, the user can now say, "I want images that are equally *sporty* as image *x*." Intuitively, enriching the feedback in this manner should help the user more quickly zero in on relevant im-

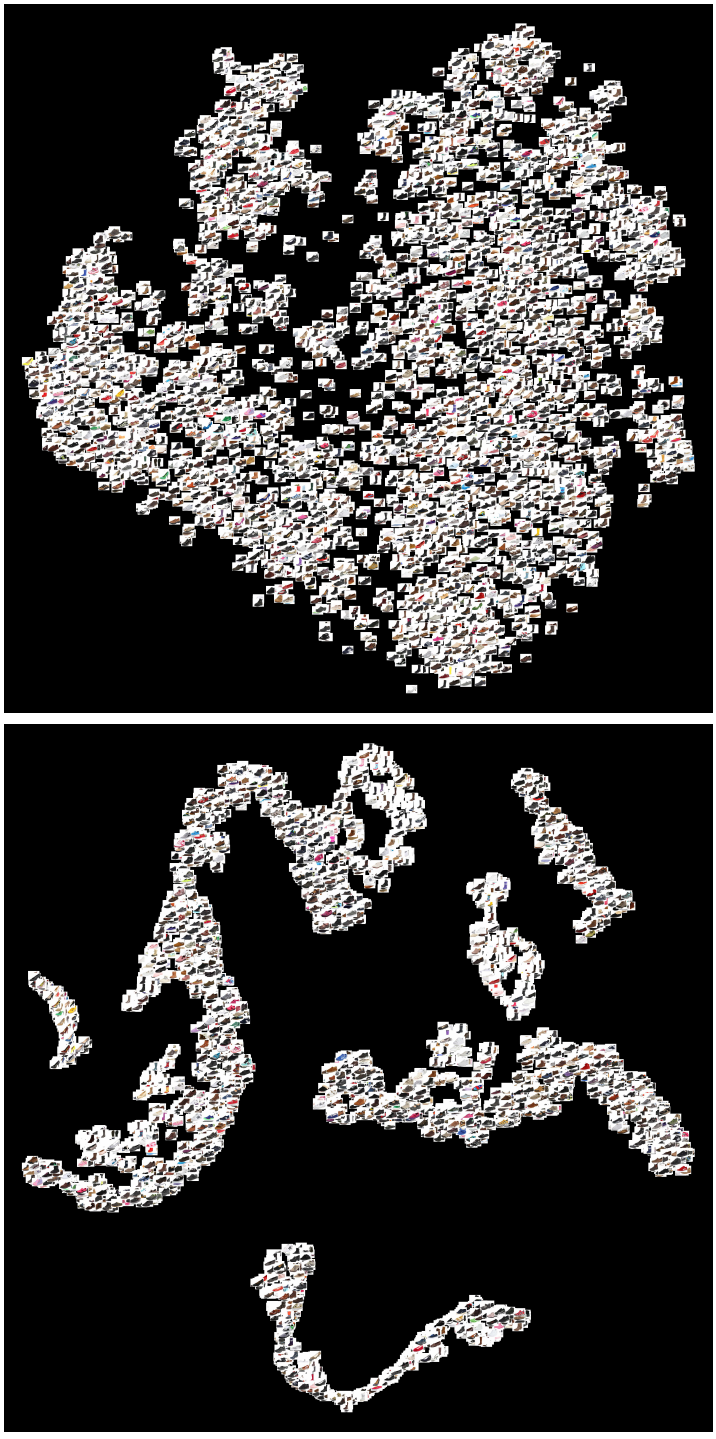


Fig. 16: t-SNE visualization of the original feature space (top) and our learned attribute space (bottom) for the attribute *pointy*. Shoes with similar level of *pointiness* are placed closer together in our learned space, forming loose "pockets" of indistinguishability. Best viewed on PDF.

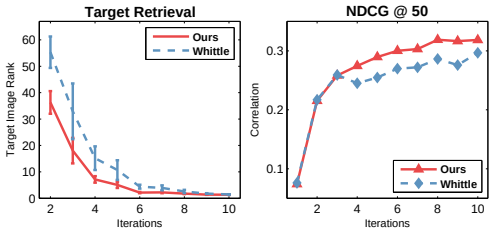


Fig. 17: Image search results. We enhance an existing relative attribute search technique called WhittleSearch [34] with our JND detection model. The resulting system finds target images more quickly (left) and produces a better overall ranking of the database images (right).

ages that match his envisioned target. To test this idea, we mimic the method and experimental setup of [34] as closely as possible, including their feedback generation simulator.

We evaluate a proof-of-concept experiment on UT-Zap50K, which is large enough to allow us to sequester disjoint data splits for training our method and performing the searches (LFW-10 is too small). We select 200 images at random to serve as the mental targets a user wants to find in the database, and reserve 5,000 images for the database. The user is shown 16 reference images and expresses 8 feedback constraints per iteration.

Figure 17 shows the results. Following [34], we measure the relevance rank of the target as a function of feedback iterations (left, lower is better), as well as the similarity of all top-ranked results compared to the target (right, higher is better). We see that JNDs substantially bolster the search task. In short, the user gets to the target in fewer iterations because he has a more complete way to express his preferences—*and* the system understands what "equally" means in terms of attribute perception.

6 Discussion

Our results show the promise of local models for addressing fine-grained visual comparisons. We saw how concentrating on the most closely related training instances is valuable for isolating the precise visual features responsible for the subtle distinctions. Our methods expand the viability of local learning beyond traditional classification tasks to include ranking. Furthermore, in an initial step towards eliminating the assumption of locality as the only relevant factor in local learning, we introduced a novel approach to learn the composition and size of the most effective neighborhood conditioned on the novel test input. Finally, we explored how local statistical models can address the "just noticeable difference" problem in attributes, successfully accounting for the non-uniformity of indistinguishable pairs in the feature space.

There are several interesting considerations worthy of further discussion and new research.

While global rankers produce comparable values for all test pairs, our local ranking method's predictions (Sect. 4) are test-pair specific. This is exactly what helps

accuracy for subtle, fine-grained comparisons, and, to some extent, mitigates the impact of inconsistent training comparisons. However, in some applications, it may be necessary to produce a full ordering of many images. In that case, one could try feeding our method's predictions to a rank aggregation technique [12], or apply a second layer of learning to normalize them, as in [11, 14, 38].

One might wonder if we could do as well by training one global ranking function per category—i.e., one for high heels, one for sneakers, etc. This would be another local learning strategy, but it appears much too restrictive. First of all, it would require category-labeled examples (in addition to the orderings $\mathcal{P}_A$), which may be expensive to obtain or simply not apropos for data lacking clear-cut category boundaries (e.g., is the storefront image an "inside city scene" or a "street scene"?). Furthermore, it would not permit cross-category comparison predictions; we want to be able to predict how images from different categories compare in their attributes, too.

As discussed in Section 4.4.4, straightforward implementations of lazy local learning come with noticeable runtime costs. In our approach, the main online costs are nearest neighbor search and rank function training. While still only seconds per test case, as larger labeled datasets become available these costs would need to be countered with more sophisticated (and possibly approximate) nearest neighbor search data structures, such as hashing or kd-trees. Another idea is to cache a set of representative models, pre-computing offline a model for each prototypical type of new input pair. Such an implementation could also be done in a hierarchical way, letting the system discover a fine-grained model in a coarse to fine manner.

An alternative approach to represent partial orders (and thus accommodate indistinguishable pairs) would be ordinal regression, where training data would consist of ordered equivalence classes of data. However, ordinal regression has severe shortcomings for our problem setting. First, it requires a consistent ordering of all training data (via the equivalence classes). This is less convenient for human annotators and more challenging to scale than the distributed approach offered by learning-to-rank, which pools any available paired comparisons. For similar reasons, learning-to-rank is much better suited to crowdsourcing annotations and learning universal (as opposed to person-specific [1, 10]) predictors. Finally, ordinal regression requires committing to a fixed number of buckets. This makes incremental supervision updates problematic. Furthermore, to represent very subtle differences, the number of buckets would need to be quite large.

Our work offers a way to learn a computational model for just noticeable differences. While we borrow the term JND from psychophysics to motivate our task, of course the analogy is not 100% faithful. In particular, psychophysical experiments to elicit JND often permit systematically varying a perceptual signal until a human detects a change, e.g., a color light source, a sound wave amplitude, or a compression factor. In contrast, the space of all visual attribute instantiations does not permit such a simple generative sampling. Instead, our method extrapolates from relatively few human-provided comparisons (fewer than 1,000 per attribute in our experiments) to obtain a statistical model for distinguishability, which generalizes to novel pairs

based on their visual properties. It remains interesting future work to explore the possibility of generative models for comparative attribute relationships.

Just noticeable difference models—and fine-grained attributes in general—appear most relevant for *category-specific* attributes. Within a category domain (e.g., faces, cars, handbags, etc.), attributes describe fine-grained properties, and it is valuable to represent any perceptible differences (or realize there are none). In contrast, comparative questions about very unrelated things or extra-domain attributes can be nonsensical. For example, do we need to model whether the shoes and the table are *equally ornate*? or whether the dog or the towel is *more fluffy*? Accordingly, we focused our experiments on domains with rich vocabularies of fine-grained attributes, faces and shoes.

Finally, we note that fine-grained differences, as addressed in this chapter, are a separate problem from *subjective* attributes. That is, our methods address the problem where there may be a subtle distinction, yet the distinction is non-controversial. Other work considers ways in which to personalize attribute models [31, 33] or discover which are subjective properties [13]. It would be interesting to investigate problems where both subjectivity and fine-grained distinctions interact.

7 Conclusion

Fine-grained visual comparisons have many compelling applications, yet traditional global learning methods can fail to capture their subtleties. We proposed several local learning-to-rank approaches based on analogous training comparisons, and we introduced a new dataset specialized to the problem. On multiple attribute datasets, we find our ideas improve the state-of-the-art.

Acknowledgements

We thank Mark Stephenson for his help creating the UT-Zap50K dataset, Naga Sandeep for providing the part-based features for LFW-10, and Ashish Kapoor for helpful discussions. This research is supported in part by NSF IIS-1065390 and ONR YIP Award N00014-12-1-0754.

References

[1] Altwaijry H, Belongie S (2012) Relative Ranking of Facial Attractiveness. In: WACV
[2] Atkeson C, Moore A, Schaal S (1997) Locally Weighted Learning. AI Review
[3] Banerjee S, Dubey A, Machchhar J, Chakrabarti S (2009) Efficient and Accurate Local Learning for Ranking. In: SIGIR Workshop
[4] Bellet A, Habrard A, Sebban M (2013) A survey on Metric Learning for Feature Vectors and Structured Data. CoRR abs/1306.6709

[5] Berg TL, Berg AC, Shih J (2010) Automatic Attribute Discovery and Characterization from Noisy Web Data. In: ECCV
[6] Biswas A, Parikh D (2013) Simultaneous Active Learning of Classifiers and Attributes via Relative Feedback. In: CVPR
[7] Bottou L, Vapnik V (1992) Local Learning Algorithms. Neural Computation
[8] Boutilier C (2011) Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems, Springer, chap Preference Elicitation and Preference Learning in Social Choice
[9] Branson S, Wah C, Schroff F, Babenko B, Welinder P, Perona P, Belongie S (2010) Visual Recognition with Humans in the Loop. In: ECCV
[10] Cao C, Kwak I, Belongie S, Kriegman D, Ai H (2014) Adaptive Ranking of Facial Attractiveness. In: ICME
[11] Chen K, Gong S, Xiang T, Loy C (2013) Cumulative Attribute Space for Age and Crowd Density Estimation. In: CVPR
[12] Conitzer V, Davenport A, Kalagnanam J (2006) Improved Bounds for Computing Kemeny Rankings. In: AAAI
[13] Curran W, Moore T, Kulesza T, Wong W, Todorovic S, Stumpf S, White R, Burnett M (2012) Towards Recognizing ”Cool”: Can End Users Help Computer Vision Recognize Subjective Attributes or Objects in Images? In: IUI
[14] Datta A, Feris R, Vaquero D (2011) Hierarchical Ranking of Facial Attributes. In: FG
[15] Davis JV, Kulis B, Jain P, Sra S, Dhillon I (2007) Information-Theoretic Metric Learning. In: ICML
[16] Domeniconi C, Gunopulos D (2001) Adaptive Nearest Neighbor Classification using Support Vector Machines. In: NIPS
[17] Duh K, Kirchhoff K (2008) Learning to Rank with Partially-Labeled Data. In: SIGIR
[18] Fan Q, Gabbur P, Pankanti S (2013) Relative Attributes for Large-Scale Abandoned Object Detection. In: ICCV
[19] Farhadi A, Endres I, Hoiem D, Forsyth D (2009) Describing Objects by their Attributes. In: CVPR
[20] Farrell R, Oza O, Zhang N, Morariu V, Darrell T, Davis L (2011) Birdlets: Subordinate Categorization using Volumetric Primitives and Pose-Normalized Appearance. In: ICCV
[21] Forsyth D, Ponce J (2002) Computer Vision: A Modern Approach. Prentice Hall
[22] Frome A, Singer Y, Malik J (2006) Image Retrieval and Classification Using Local Distance Functions. In: NIPS
[23] Frome A, Singer Y, Sha F, Malik J (2007) Learning Globally-Consistent Local Distance Functions for Shape-Based Image Retrieval and Classification. In: ICCV
[24] Geng X, Liu T, Qin T, Arnold A, Li H, Shum H (2008) Query Dependent Ranking using K-nearest Neighbor. In: SIGIR
[25] Hastie T, Tibshirani R (1996) Discriminant Adaptive Nearest Neighbor Classification. PAMI
[26] Huang GB, Ramesh M, Berg T, Learned-Miller E (2007) Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments. Tech. Rep. 07-49, University of Massachusetts, Amherst
[27] Jain P, Kulis B, Grauman K (2008) Fast Image Search for Learned Metrics. In: CVPR
[28] Jiang X, Lim L, Yao Y, Ye Y (2011) Statistical Ranking and Combinatorial Hodge Theory. Math Program
[29] Joachims T (2002) Optimizing Search Engines using Clickthrough Data. In: KDD
[30] Kapoor A, Jain P, Viswanathan R (2012) Multilabel Classification using Bayesian Compressed Sensing. In: NIPS
[31] Kovashka A, Grauman K (2013) Attribute Adaptation for Personalized Image Search. In: ICCV
[32] Kovashka A, Grauman K (2013) Attribute Pivots for Guiding Relevance Feedback in Image Search. In: ICCV
[33] Kovashka A, Grauman K (2015) Discovering Attribute Shades of Meaning with the Crowd. International Journal on Computer Vision (IJCV) 114(1):56–73

[34] Kovashka A, Parikh D, Grauman K (2012) WhittleSearch: Image Search with Relative Attribute Feedback. In: CVPR
[35] Kumar N, Belhumeur P, Nayar S (2008) FaceTracer: A Search Engine for Large Collections of Images with Faces. In: ECCV
[36] Kumar N, Berg A, Belhumeur P, Nayar SK (2009) Attribute and Simile Classifiers for Face Verification. In: ICCV
[37] Lampert C, Nickisch H, Harmeling S (2009) Learning to Detect Unseen Object Classes by Between-Class Attribute Transfer. In: CVPR
[38] Li S, Shan S, Chen X (2012) Relative Forest for Attribute Prediction. In: ACCV
[39] Lin H, Yu C, Chen H (2011) Query-Dependent Rank Aggregation with Local Models. In: AIRS
[40] Maaten L, Hinton G (2008) Visualizing High-Dimensional Data Using t-SNE. Journal of Machine Learning Research 9:2579–2605
[41] Matthews T, Nixon M, Niranjan M (2013) Enriching Texture Analysis with Semantic Data. In: CVPR
[42] Oliva A, Torralba A (2001) Modeling the Shape of the Scene: A Holistic Representation of the Spatial Envelope. IJCV
[43] Parikh D, Grauman K (2011) Relative Attributes. In: ICCV
[44] Parzen E (1962) On Estimation of a Probability Density Function and Mode. The Annuals of Mathematical Statistics 33(3):1065–1076
[45] Reid D, Nixon M (2011) Using Comparative Human Descriptions for Soft Biometrics. In: IJCB
[46] Sadovnik A, Gallagher A, Parikh D, Chen T (2013) Spoken Attributes: Mixing Binary and Relative Attributes to Say the Right Thing. In: ICCV
[47] Sandeep R, Verma Y, Jawahar C (2014) Relative Parts: Distinctive Parts for Learning Relative Attributes. In: CVPR
[48] Scheirer W, Kumar N, Belhumeur P, Boult T (2012) Multi-Attribute Spaces: Calibration for Attribute Fusion and Similarity Search. In: CVPR
[49] Shrivastava A, Singh S, Gupta A (2012) Constrained Semi-Supervised Learning Using Attributes and Comparative Attributes. In: ECCV
[50] Siddiquie B, Feris R, Davis L (2011) Image Ranking and Retrieval based on Multi-Attribute Queries. In: CVPR
[51] Vincent P, Bengio Y (2001) K-Local Hyperplane and Convex Distance Nearest Neighbor Algorithms. In: NIPS
[52] Weinberger K, Saul L (2009) Distance Metric Learning for Large Margin Nearest Neighbor Classification. JMLR
[53] Yang L, Jin R, Sukthankar R, Liu Y (2006) An Efficent Algorithm for Local Distance Metric Learning. In: AAAI
[54] Yu A, Grauman K (2014) Fine-Grained Visual Comparisons with Local Learning. In: CVPR
[55] Yu A, Grauman K (2014) Predicting Useful Neighborhoods for Lazy Local Learning. In: NIPS
[56] Yu A, Grauman K (2015) Just Noticeable Differences in Visual Attributes. In: ICCV
[57] Zhang H, Berg A, Maire M, Malik J (2006) SVM-KNN: Discriminative Nearest Neighbor Classification for Visual Category Recognition. In: CVPR