

Fine-Grained Visual Comparisons with Local Learning

Aron Yu and Kristen Grauman
University of Texas at Austin

aron.yu@utexas.edu, grauman@cs.utexas.edu

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. Existing relative attribute methods rely on global ranking functions; yet 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. To address these issues, we propose a local learning approach for fine-grained visual comparisons. Given a novel pair of images, we learn a local ranking model on the fly, using only analogous training comparisons. We show how to identify these analogous pairs using learned metrics. With results on three challenging datasets—including a large newly curated dataset for fine-grained comparisons—our method outperforms state-of-the-art methods for relative attribute prediction.

1. Introduction

Beyond recognizing objects (or activities, scenes, emotions, etc.), a computer vision system ought to be able to *compare* them. A promising way to represent visual comparisons is through *attributes*, which are mid-level properties that appear across category boundaries and often vary in terms of their perceived strengths. For example, with a model for the relative attribute “brightness”, the 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” [29]. 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*” [21]. 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” [5, 28, 30]. In texture recognition, relative attributes could capture the strength of base properties [26]. For subjective visual tasks, users could teach the system their personal perception, e.g., about which human faces are *more attractive* than others [1].



Figure 1: A global ranker 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.

While a promising direction, the standard ranking approach tends to fail when faced with *fine-grained visual comparisons*, in which the novel pair of images exhibits subtle visual attribute differences. While the 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 (Figure 1). Furthermore, by learning a single global function to rank all data, existing methods ignore the reality that visual comparisons need not be transitive; if human viewers perceive $A \succ B \succ C \succ A$, one global function cannot adequately capture the perceived ordering (Figure 2).

We contend that the fine-grained comparisons are actually 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.

We propose a local learning approach to the fine-grained visual comparison problem. Rather than learn a single global function to predict how pairs of examples relate, we learn local functions that tailor the comparisons to the



Figure 2: Visual comparisons need not be transitive. An observer rates A as *more comfortable* than B , and $B > C$, but $A < C$.

neighborhood statistics of the data. In particular, given a novel test pair of images and the attribute along which they should be compared, we first identify analogous training pairs using a learned metric. We then train a ranking function on the fly using only those nearby pairs, and apply it to the test case. 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. At the same time, by not insisting on a single global function to relate all images, we mitigate the impact of inconsistencies in visual comparisons.

To our knowledge, no prior work specifically explores fine-grained visual comparisons, and all prior methods assume a single global function is sufficient [9, 11, 21, 23, 28]. Furthermore, while local learning methods have been explored for classification [2, 6, 17, 31, 33] and information retrieval problems [3, 13, 16, 24], our idea for learning local ranking functions with comparisons is new. A key question is how to identify neighboring training pairs; we show how to learn pairs that appear analogous to the input, accounting for the attribute-specific visual similarities.

On three challenging datasets from distinct domains, our approach improves the state of the art in relative attribute predictions. We also introduce a new large dataset of 50,000 Zappos shoe images that focuses on fine-grained attribute comparisons. Our results indicate that *more* labeled data is not necessarily preferable to isolating the *right* data.

2. Related Work

Comparing attributes has gained attention in the last several years. The “relative attributes” approach learns a global linear ranking function for each attribute [28]. It is extended to non-linear ranking functions in [23] by training a hierarchy of rankers with different subsets of data, then normalizing predictions at the leaf nodes. In [11], rankers trained for each feature descriptor (color, shape, texture) are combined to produce a single global ranking function. Aside from learning to rank formulations, researchers have applied the Elo rating system for biometrics [29], and regression over “cumulative attributes” for age and crowd density estimation [9]. 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.

The basic idea in local learning [2, 6] is to concentrate the learning algorithm on training instances that are most

similar to the test example. Primarily two formulations have been studied. In the first, the system identifies the K training examples nearest to the test input, trains a model with only those examples, and applies it to the test case. For example, this paradigm is employed for neural network classification [6], linear regression [2], and SVM classification [33]. In the second strategy, the system learns a feature space mapping (e.g., with LDA) with only those instances close to the test example [17, 31], thereby tailoring the representation to the input. In a similar spirit, local metric learning methods use example-specific weights [15, 25] or a cluster-specific feature transformation [32], then apply nearest neighbor classification. 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.

In information retrieval, local learning methods have been developed to sort documents by their relevance to query keywords [3, 13, 16, 24]. They take strategies quite similar to the above, e.g., building a local model for each cluster in the training data [24], projecting training data onto a subspace determined by the test data distribution [13], or building a model with only the query’s neighbors [3, 16]. 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.

The fact that humans exhibit inconsistencies in their comparisons is well known in social choice theory and preference learning [7]. In all the global models above, intransitive constraints would be unaccounted for and treated as noise. While the HodgeRank algorithm [19] 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.

Work on fine-grained visual categorization aims to recognize objects in a single domain, e.g., bird species [8, 14]. While such problems also require making distinctions among visually close instances, our goal is to compare attributes, not categorize objects.

3. Approach

Our local learning approach addresses the relative comparison problem on a per attribute basis. 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*.

In the following, we first present a brief overview of Relative Attributes [28] (Section 3.1), as it sets the foundation as a state-of-the-art global ranking approach. Then we introduce our local ranking approach (Section 3.2), followed by our idea to select fine-grained neighboring pairs with metric learning (Section 3.3).

3.1. Ranking for Relative Attributes

The Relative Attributes approach [28] treats the attribute comparison task as a learning-to-rank problem. The idea is to use ordered 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. Each attribute is learned independently.

Let $\mathbf{x}_i \in \mathbb{R}^d$ denote the image descriptor for image i , such as a GIST descriptor or a color histogram. The algorithm is given a set of training image pairs $\mathcal{O}_{\mathcal{A}} = \{(i, j)\}$, in which each ordered pair (i, j) denotes that image i displays the attribute \mathcal{A} more than image j . Let $R_{\mathcal{A}}$ be a linear ranking function:

$$R_{\mathcal{A}}(\mathbf{x}) = \mathbf{w}_{\mathcal{A}}^T \mathbf{x}. \quad (1)$$

The goal is to learn the parameters $\mathbf{w}_{\mathcal{A}} \in \mathbb{R}^d$ so that the ordering $R_{\mathcal{A}}$ assigns to the training pairs agrees with $\mathcal{O}_{\mathcal{A}}$ as much as possible. That is, $\forall (i, j) \in \mathcal{O}_{\mathcal{A}} : \mathbf{w}_{\mathcal{A}}^T \mathbf{x}_i > \mathbf{w}_{\mathcal{A}}^T \mathbf{x}_j$. By itself, the problem is NP-hard, but [20] introduces slack variables and a large-margin regularizer to approximately solve it. The learning objective is:

$$\begin{aligned} \text{minimize} \quad & \left(\frac{1}{2} \|\mathbf{w}_{\mathcal{A}}^T\|_2^2 + C \sum \xi_{ij}^2 \right) \\ \text{s.t.} \quad & \mathbf{w}_{\mathcal{A}}^T (\mathbf{x}_i - \mathbf{x}_j) \geq 1 - \xi_{ij}; \forall (i, j) \in \mathcal{O}_{\mathcal{A}} \\ & \xi_{ij} \geq 0 \end{aligned} \quad (2)$$

where the constant C balances the regularizer and ordering constraints. The objective can also be seen as a paired classification problem, where, rather than predict the class label of an individual \mathbf{x}_i , we want to predict the label “more” or “less” for a pair (i, j) based on the difference in their visual features. The margin one wants to maximize is the distance between the nearest ranked points. While [28] uses this linear formulation, it is also kernelizable and so can produce non-linear ranking functions.¹

¹The objective in [28] further adds a set of “similar” training pairs that should receive similar ranks. We found they did not impact results for either global or local methods on all our datasets, and so we omit them.

By projecting images onto the resulting hyperplane $\mathbf{w}_{\mathcal{A}}$, we obtain a 1D global ranking for that attribute, e.g., from least to most “comfortable”. Given a test pair $(\mathbf{x}_p, \mathbf{x}_q)$, if $R_{\mathcal{A}}(\mathbf{x}_p) > R_{\mathcal{A}}(\mathbf{x}_q)$, the method predicts image p has the attribute “more” than image q , and “less” otherwise.

Our local approach draws on this particular ranking formulation, which is also used in both [28] and in the hierarchy of [23] to produce state-of-the-art results. However, we note that our local learning idea would apply similarly to alternative ranking methods.

3.2. Local Learning for Visual Comparisons

Existing methods train a global ranking function using all available constraints $\mathcal{O}_{\mathcal{A}}$, 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 Figure 1). Indeed, in our early exploration applying Relative Attributes [28], we were impressed by the qualitative results at either end of an attribute’s spectrum, but we could not make sense of its finer-grained predictions. For example, for a dataset of shoes, it would 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 did not show a clear pattern.

The solution 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 [6]. 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 training data increases, more human subjectiveness and ordering inconsistencies will emerge, further straining the validity of a single global function.

Thus, we propose a local learning approach for attribute ranking. The idea is to train a custom ranking function tailored to each novel pair of images $X_q = (\mathbf{x}_r, \mathbf{x}_s)$ that we wish to compare. We train the custom function using just 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 $\mathcal{O}_{\mathcal{A}}$ by their similarity to $(\mathbf{x}_r, \mathbf{x}_s)$, then compose a local training set $\mathcal{O}'_{\mathcal{A}}$ consisting of the top K neighboring pairs, $\mathcal{O}'_{\mathcal{A}} = \{(\mathbf{x}_{k1}, \mathbf{x}_{k2})\}_{k=1}^K$. (We explain in the next section how we define similarity between pairs.) Then, we train a ranking function using Eq. 2 on the fly, and apply it to compare the test images.

Such a fine-grained approach helps to eliminate ordering constraints that are irrelevant to the test pair. For in-

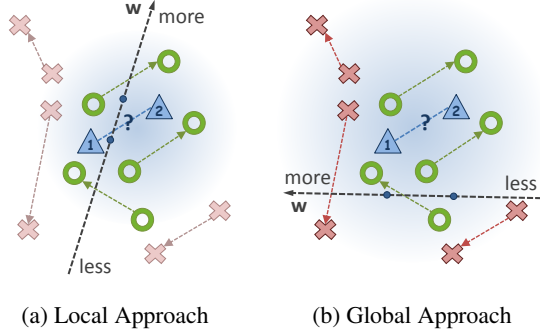


Figure 3: Given a novel test pair (blue Δ) in a learned metric space, our local approach (a) selects only the most relevant neighbors (green \circ) for training, which leads to ranking test image 2 over 1 in terms of “sporty”. In contrast, the standard global approach (b) uses all training data (green \circ & red \times) 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 .

stance, 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 (Figure 3).

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., in the example above. This would be another local learning strategy, but it is much too restrictive. First of all, it would require category-labeled examples (in addition to the orderings \mathcal{O}_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.

3.3. 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 $X_q = (\mathbf{x}_r, \mathbf{x}_s)$ be the test pair, and let $X_t = (\mathbf{x}_u, \mathbf{x}_v)$ be a labeled training pair for which $(u, v) \in \mathcal{O}_A$. We define their distance as:

$$D_A(X_q, X_t) = \min(D'_A((\mathbf{x}_r, \mathbf{x}_s), (\mathbf{x}_u, \mathbf{x}_v)), D'_A((\mathbf{x}_r, \mathbf{x}_s), (\mathbf{x}_v, \mathbf{x}_u))), \quad (3)$$

where D'_A is the product of the two items’ distances:

$$D'_A((\mathbf{x}_r, \mathbf{x}_s), (\mathbf{x}_u, \mathbf{x}_v)) = d_A(\mathbf{x}_r, \mathbf{x}_u) \times d_A(\mathbf{x}_s, \mathbf{x}_v). \quad (4)$$

The product reflects that we are looking for pairs where each image is visually similar to one of those in the novel pair. 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 Eq. 3 and the swapping of $(\mathbf{x}_u, \mathbf{x}_v) \rightarrow (\mathbf{x}_v, \mathbf{x}_u)$ in the second term ensure that we account for the unknown ordering of the test pair; while all training pairs are ordered with $R_A(\mathbf{x}_u) > R_A(\mathbf{x}_v)$, 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_A , then take the top K to form \mathcal{O}'_A .

When identifying neighbor pairs, rather than judge image distance d_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:

$$d_A(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i - \mathbf{x}_j)^T \mathbf{M}_A (\mathbf{x}_i - \mathbf{x}_j), \quad (5)$$

parameterized by the $d \times d$ positive definite matrix \mathbf{M}_A . We employ the information-theoretic metric learning (ITML) algorithm [12], due to its efficiency and kernelizability. Given an initial $d \times d$ matrix \mathbf{M}_{A_0} specifying any prior knowledge about how the data should be compared, ITML produces the \mathbf{M}_A that minimizes the LogDet divergence

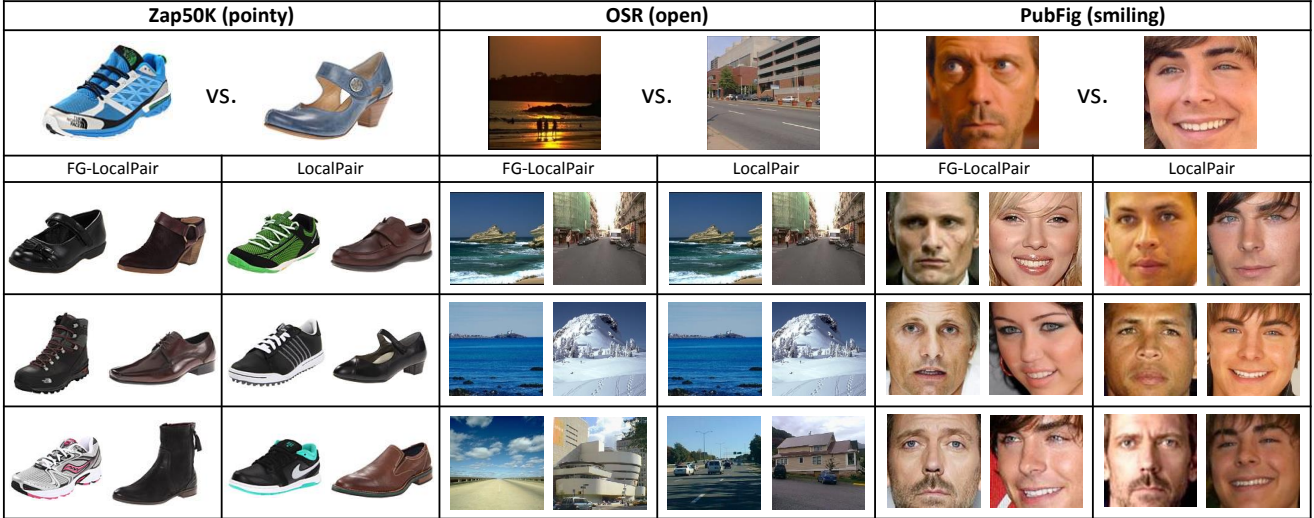


Figure 4: Example fine-grained neighbor pairs for three test pairs (top row) from the datasets tested in this paper. We display the top 3 pairs per query. FG-LocalPair and LocalPair denote results with and without metric learning (ML), respectively. **Zap50K pointy**: ML puts the comparison focus on 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.

$D_{\ell d}$ from that initial matrix, subject to constraints that similar data points be close and dissimilar points be far:

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

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 $M_{\mathcal{A}_0} = \Sigma^{-1}$, 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}* .

Figure 4 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, as we will show in Section 5.

3.4. Discussion

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 [18]. 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.

While global rankers produce comparable values for all test pairs, our method’s predictions 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. For an application requiring a full ordering of many images, one could feed our predictions to a rank aggregation technique [10], or apply a second layer of learning to normalize them, as in [9, 11, 23].

4. Fine-Grained Attribute Zappos Dataset

We introduce a new Zappos50K dataset (**Zap50K²**) 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.

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

²Zap50K dataset and all related data are publicly available for download at vision.cs.utexas.edu/projects/finegrained



Figure 5: Example pairs contrasting our predictions to the Global baseline’s. In each pair, top item is *more sporty* than 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.

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 $50K^2$ 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 **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 detectable 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. The rationale functions as a speed bump to slow workers down so that they think more carefully about their decisions. We call this set **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 [4] with relative attributes [21], Zap50K is about $3.5 \times$ larger, offers meta-data and $10 \times$ more comparative labels, and most importantly, specifically targets fine-grained tasks.

5. Experiments

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

Datasets We evaluate on three datasets: **Zap50K**, as defined above, with concatenated GIST and color histogram features; the Outdoor Scene Recognition dataset [27] (**OSR**); and a subset of the Public Figures faces dataset [22] (**PubFig**). OSR contains 2,688 images (GIST features) with 6 attributes, while PubFig contains 772 images (GIST + Color features) with 11 attributes. See Supp File for more details. We use the exact same attributes, features, and train/test splits as [23, 28].

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 [23]. We present results using the same labeled data for all methods.

For ITML, we use the ordered pairs \mathcal{O}_A for rank training to compose the set of dissimilar pairs \mathcal{D}_A , and the set of “equal” pairs to compose the similar pairs \mathcal{S}_A . We use the default settings for c and l in the authors’ code [12]. The setting of K determines “how local” the learner is; its optimal setting depends on the training data and query. As in prior work [6, 33], we simply fix it for all queries at $K = 100$. Values of $K = 50$ to 200 give similar results. See Supp File for more details.

Baselines We compare the following methods:

- **FG-LocalPair**: the proposed fine-grained approach.
- **LocalPair**: our approach without the learned metric (i.e., $\mathbf{M}_A = \mathbb{I}$). This baseline isolates the impact of tailoring the search for neighboring pairs to the attribute.

	Open	Pointy	Sporty	Comfort
Global [28]	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

Table 1: Zap50K-1 dataset results for *coarser* pairs.

	Open	Pointy	Sporty	Comfort
Global [28]	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

Table 2: Zap50K-2 dataset results for *fine-grained* pairs.

- **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 Eq. 2. This is the Relative Attributes Method [28]. We use the authors’ public code.
- **RelTree**: the non-linear relative attributes approach of [23], 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).

Zappos Results Table 1 shows the accuracy on Zap50K-1. Our method outperforms all baselines for all attributes. To isolate the more difficult pairs in 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 6 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 5 shows the qualitative results.

We proceed to test on even more difficult pairs. Whereas Figure 6 focuses on pairs difficult according to the learned metric, next we focus on pairs difficult according to our human annotators. Table 2 shows the results for Zap50K-2. We use the original ordered pairs for training and all 4,612 fine-grained pairs for testing (Section 4). We outperform all methods for 3 of the 4 attributes. For the two more 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

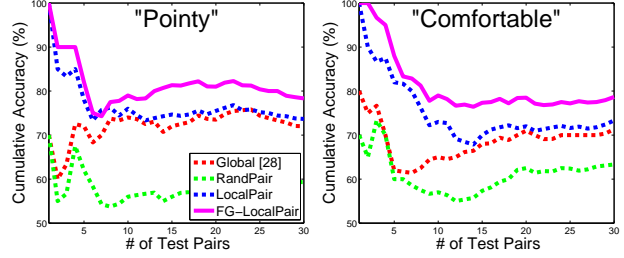


Figure 6: Accuracy for the 30 hardest test pairs on Zap50K-1.

pairs (Table 1). We think this is because the locations of the subtleties vary greatly per pair.

Overall, local learning outperforms the state-of-the-art global approach [28] on the Zappos images.

Scenes and PubFig Results We now shift our attention to OSR and PubFig, two commonly used datasets for relative attributes [21, 23, 28]. The paired supervision for these datasets originates from category-wise comparisons [28], and as such there are many more training pairs—on average over 20,000 per attribute.

Tables 3 and 4 show the accuracy for PubFig and OSR, respectively. Figure 7 shows representative precision recall curves, using $|R(x_i) - R(x_j)|$ as a measure of confidence.

On both datasets, our method outperforms all the baselines. Most notably, it outperforms RelTree [23], 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 [28] 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 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,

	Male	White	Young	Smiling	Chubby	Forehead	Eyebrow	Eye	Nose	Lip	Face
RelTree [23]	85.33	82.59	84.41	83.36	78.97	88.83	81.84	83.15	80.43	81.87	86.31
Global [28]	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

Table 3: Accuracy comparison for the PubFig dataset. FG-LocalPair denotes the proposed approach.

	Natrl	Open	Persp.	LgSize	Diag	ClsDepth
RelTree [23]	95.24	92.39	87.58	88.34	89.34	89.54
Global [28]	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 4: Accuracy comparison for the OSR dataset.

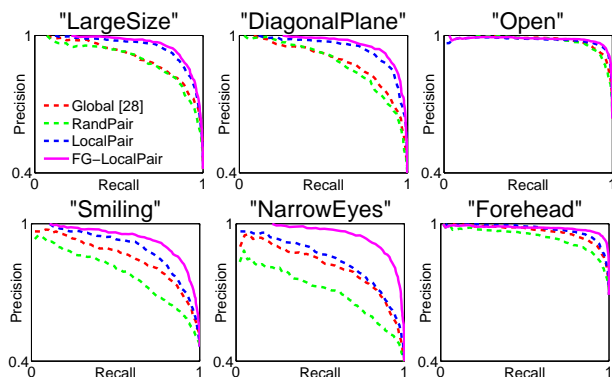


Figure 7: Precision-recall for OSR (top) and PubFig (bottom).

and is liable to find images of the same person as more relevant, regardless of their properties in that image (Figure 4).

6. Conclusion

Fine-grained visual comparisons have many compelling applications, yet traditional global learning methods can fail to capture their subtleties. We proposed a local learning-to-rank approach based on analogous training comparisons, and we introduced a new dataset specialized to the problem. With three attribute datasets, we find our idea improves the state of the art. In future work, we plan to explore ways to pre-compute local models to reduce run-time and investigate generalizations to higher-order comparisons.

Acknowledgements This research is supported in part by NSF IIS-1065390 and ONR YIP.

References

- [1] H. Altwaijry and S. Belongie. Relative ranking of facial attractiveness. In *WACV*, 2012.
- [2] C. Atkeson, A. Moore, and S. Schaal. Locally weighted learning. *AI Review*, 1997.
- [3] S. Banerjee, A. Dubey, J. Machchhar, and S. Chakrabarti. Efficient and accurate local learning for ranking. In *SIGIR Workshop*, 2009.
- [4] T. L. Berg, A. C. Berg, and J. Shih. Automatic Attribute Discovery and Characterization from Noisy Web Data. In *ECCV*, 2010.
- [5] A. Biswas and D. Parikh. Simultaneous active learning of classifiers & attributes via relative feedback. In *CVPR*, 2013.

- [6] L. Bottou and V. Vapnik. Local learning algorithms. *Neural Comp*, 1992.
- [7] C. Boutilier. *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*, chapter Preference Elicitation and Preference Learning in Social Choice. 2011.
- [8] S. Branson, C. Wah, F. Schroff, B. Babenko, P. Welinder, P. Perona, and S. Belongie. Visual recognition with humans in the loop. In *ECCV*, 2010.
- [9] K. Chen, S. Gong, T. Xiang, and C. Loy. Cumulative attribute space for age and crowd density estimation. In *CVPR*, 2013.
- [10] V. Conitzer, A. Davenport, and J. Kalagnanam. Improved bounds for computing kemeny rankings. In *AAAI*, 2006.
- [11] A. Datta, R. Feris, and D. Vaquero. Hierarchical ranking of facial attributes. In *FG*, 2011.
- [12] J. V. Davis, B. Kulis, P. Jain, S. Sra, and I. S. Dhillon. Information-Theoretic Metric Learning. In *ICML*, 2007.
- [13] K. Duh and K. Kirchhoff. Learning to rank with partially-labeled data. In *SIGIR*, 2008.
- [14] R. Farrell, O. Oza, N. Zhang, V. Morariu, T. Darrell, and L. Davis. Birdlets: Subordinate categorization using volumetric primitives and pose-normalized appearance. In *ICCV*, 2011.
- [15] A. Frome, Y. Singer, F. Sha, and J. Malik. Learning globally-consistent local distance functions for shape-based image retrieval and classification. In *ICCV*, 2007.
- [16] X. Geng, T. Liu, T. Qin, A. Arnold, H. Li, and H. Shum. Query dependent ranking using k-nearest neighbor. In *SIGIR*, 2008.
- [17] T. Hastie and R. Tibshirani. Discriminant adaptive nearest neighbor classification. *PAMI*, 1996.
- [18] P. Jain, B. Kulis, and K. Grauman. Fast image search for learned metrics. In *CVPR*, 2008.
- [19] X. Jiang, L. Lim, Y. Yao, and Y. Ye. Statistical Ranking and Combinatorial Hodge Theory. *Math. Program.*, 2011.
- [20] T. Joachims. Optimizing search engines using clickthrough data. In *KDD*, 2002.
- [21] A. Kovashka, D. Parikh, and K. Grauman. WhittleSearch: Image search with relative attribute feedback. In *CVPR*, 2012.
- [22] N. Kumar, A. Berg, P. Belhumeur, and S. K. Nayar. Attribute and simile classifiers for face verification. In *ICCV*, 2009.
- [23] S. Li, S. Shan, and X. Chen. Relative forest for attribute prediction. In *ACCV*, 2012.
- [24] H. Lin, C. Yu, and H. Chen. Query-dependent rank aggregation with local models. In *AIRS*, 2011.
- [25] T. Malisiewicz and A. Efros. Recognition by association via learning per-exemplar distance. In *CVPR*, 2008.
- [26] T. Matthews, M. Nixon, and M. Niranjan. Enriching texture analysis with semantic data. In *CVPR*, 2013.
- [27] A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *IJCV*, 2001.
- [28] D. Parikh and K. Grauman. Relative attributes. In *ICCV*, 2011.
- [29] D. Reid and M. Nixon. Using comparative human descriptions for soft biometrics. In *IJCB*, 2011.
- [30] A. Shrivastava, S. Singh, and A. Gupta. Constrained semi-supervised learning using attributes and comparative attributes. In *ECCV*, 2012.
- [31] P. Vincent and Y. Bengio. K-local hyperplane and convex distance nearest neighbor algorithms. In *NIPS*, 2001.
- [32] K. Weinberger and L. Saul. Distance metric learning for large margin nearest neighbor classification. *JMLR*, 2009.
- [33] H. Zhang, A. Berg, M. Maire, and J. Malik. SVM-KNN: Discriminative nearest neighbor classification for visual category recognition. In *CVPR*, 2006.