Divide, Share, and Conquer: Multi-task Attribute Learning with Selective Sharing

Chao-Yeh Chen*, Dinesh Jayaraman*, Fei Sha, and Kristen Grauman

Abstract Existing methods to learn visual attributes are plagued by two common issues: (i) they are prone to confusion by properties that are correlated with the attribute of interest among training samples, and (ii) they often learn generic, imprecise "lowest common denominator" attribute models in an attempt to generalize across classes where a single attribute may have very different visual manifestations. Yet, many proposed applications of attributes rely on being able to learn the precise and correct semantic concept corresponding to each attribute. We argue that these issues are both largely due to indiscriminate "oversharing" amongst attribute classifiers along two axes — (i) visual features and (ii) classifier parameters. To address both these issues, we introduce the general idea of *selective sharing* during multi-task learning of attributes. First, we show how selective sharing helps learn decorrelated models for each attribute in a vocabulary. Second, we show how selective sharing permits a new form of transfer learning between attributes, yielding a specialized attribute model for each individual object category. We validate both these instantiations of our selective sharing idea through extensive experiments on multiple datasets. We show how they help preserve semantics in learned attribute models, benefitting various downstream applications such as image retrieval or zero-shot learning.

Chao-Yeh Chen*
e-mail: chaoyeh@cs.utexas.edu

Dinesh Jayaraman*
e-mail: dineshj@cs.utexas.edu

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

* indicates equal contribution

1 Introduction

Visual attributes are human-nameable mid-level semantic properties. They include both holistic descriptors, such as "furry", "dark", or "metallic", as well as localized parts, such as "has-wheels", or "has-snout". Because attributes describe object and scene categories in natural language terms, they can be used to describe an unfamiliar object class [9], teach a system to recognize new classes by zero-shot learning [25, 32, 36], learn mid-level cues from cross-category images [23], or provide a useful bridge between low-level image features and high-level entities like object or scene categories [9, 22, 25].[1]

All these applications stem from one crucial property of attributes—the fact that they are *shared across object categories*. Typically, the idea is that a system can learn about an attribute from image examples drawn from arbitrary objects, e.g., learning "furry" from bunnies, dogs, and bears alike. In fact, attributes are usually shared among not only among some limited set of "seen" categories present in the training data, but among other "unseen" categories too. Thus, it is particularly important to be able to correctly recognize each attribute manifested in diverse configurations that may or may not have been previously observed.

The intent to share features and classifiers raises important challenges specific to attribute learning. On the one hand, as we will soon see, spurious correlated factors (including other attributes) in training data may easily be mistaken for the attribute of interest by a learner, which would prevent generalization, especially to instances of the attribute manifested in unseen classes. Further, even among seen classes, attributes may have different visual manifestations in each category, making it difficult for one shared generic attribute classifier to work well on all classes.

Existing methods follow the same standard discriminative learning pipeline that has been successful in other visual recognition problems, particularly object recognition. Using training images labeled by the attributes they exhibit, low-level image descriptors are extracted, and used to *independently* train a discriminative classifier for each attribute in isolation [5, 9, 22, 23, 25, 32, 33, 36, 38]. A single monolithic model is trained per attribute, which is shared across all object categories. For example, classifiers for "furry" and "dark" attributes may be trained independently with color, texture, and shape features. Each of these classifiers is expected now to apply to new instances, agnostic to the category that each instance belongs to, such as "cat", "human" or "tower". In short, the status quo approach thus uniformly shares both the low-level features across all attributes as well as the attribute classifier across all categories.

In this chapter, we explore the following question: when and to what extent is sharing useful for attribute learning? We show that the standard attribute learning approach suffers from a problem of indiscriminate sharing along two axes: (i) it "overshares" features across distinct attribute classifiers and (ii) it overshares classifier parameters for each attribute across distinct categories. See Figure 1 for a visual

[1] Throughout, we use the term "category" to refer to an object or scene class, whereas an "attribute" is a visual property describing some such category.

depiction of this problem. We contend that this oversharing approach ignores inter-category and inter-attribute distinctions during attribute learning and thus does not optimally exploit training data.

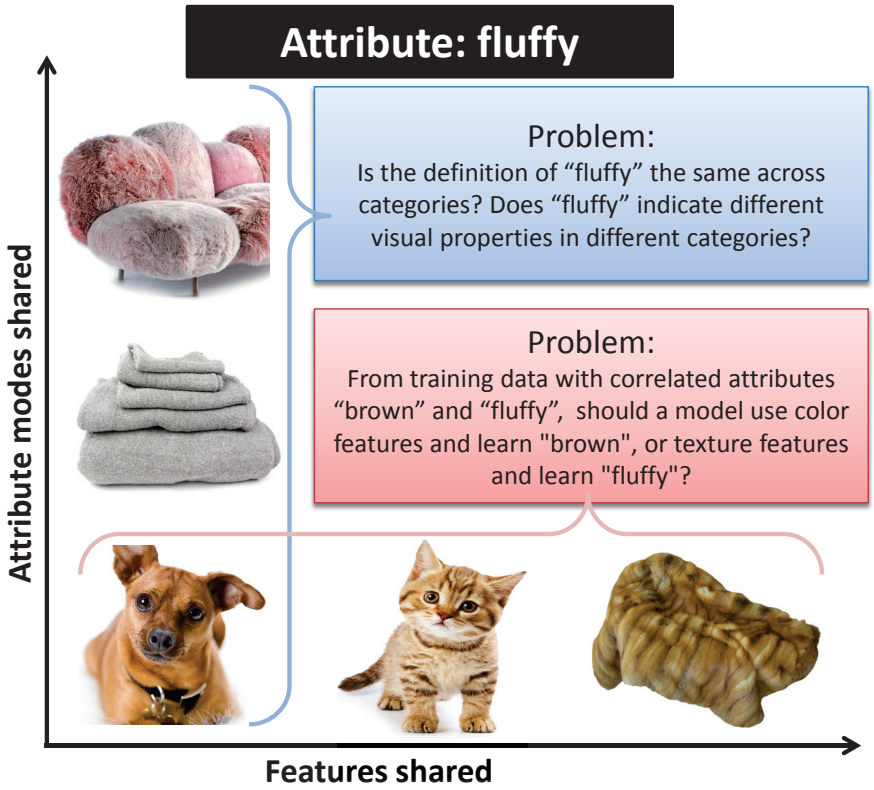


Fig. 1: Two problems caused by oversharing the features and attribute modes in attribute learning framework. (i) On the one hand, when attribute models overshare feature supports, it is hard to disambiguate correlated attributes that are semantically very different, such as “brown” and ‘fluffy” in the example depicted on the horizontal axis. (ii) On the other hand, when attribute classifiers are overshared across object categories, we ignore the fact that the same semantic attribute could have very different visual appearances in different categories.

We propose methods to actively account for the semantic information presented by these distinctions, which allow the learning of better attribute classifiers using the same attribute-labeled training data. Our key idea for improving upon existing attribute learning methods is to make the system “learn the right thing” by avoiding oversharing, using semantic knowledge to decide what to share and what not to share during learning. We implement this general idea in two separate *multi-task learning* (MTL) schemes to address each of the two problems enumerated above. Multi-task learning methods aim to jointly learn multiple tasks. Whereas typically a multi-task

learner strives for greater sharing between tasks, we propose new forms of MTL where the algorithm is intentionally selective about where to share. We show how the concept of selective sharing helps eliminate two major problems that plague the standard attribute recognition approach—namely, (i) disambiguating each attribute from its spurious correlated image properties (Section 2), and (ii) specializing individual attribute classifiers to fit differences in visual manifestations of the same attribute across different object categories (Section 3).

Problem #1: Oversharing image features across categories conflates pair(s) of attributes. In the first main contribution of this chapter, we reconsider the standard approach of using the same feature representation for all attributes. Even standard multi-task learning approaches encourage the sharing of features across attributes. This defect makes these approaches especially prone to learning image properties that are *correlated* with the attribute of interest, rather than the attribute itself. In Section 2, we propose a multi-task learning method informed by attribute semantics to disambiguate correlated attributes while learning attribute vocabularies, by encouraging different classifiers to rely on signals from disjoint sets of dimensions in the visual feature space [17].

Problem #2: Oversharing attributes across categories conflates diverse "modes" of same-named attributes. In the second main contribution of this chapter, we reconsider the standard approach of learning one monolithic attribute classifier from training images pooled from all categories. While the notion of a category-independent attribute has certain appeal—are attributes really category-independent? For instance, does fluffiness on a dog look the same as fluffiness on a towel? Are the features that make a high heeled shoe look formal the same as those that make a sandal look formal? In such examples (and many others), while the *linguistic* semantics are preserved across categories, the *visual* appearance of the property is transformed to some degree. That is, some attributes are specialized to the category. This suggests that simply pooling a bunch of training images of any object/scene with the named attribute and learning a discriminative classifier—the status quo approach—will weaken the learned model to account for the "least common denominator" of the attribute's appearance, and, in some cases, completely fail to generalize. In Section 3, we present a method to learn category-sensitive *analogous attributes*, by exploring the correlations between different attributes and object categories [6].

Thus, both of these approaches implement our key idea of *selective* sharing (of features and models respectively) when treating attribute learning as a multi-task learning problem. In both approaches, we pursue joint learning of a vocabulary of attributes. Whereas the first approach produces a single attribute model per attribute word, the second approach further formulates the learning of each attribute itself as multiple related tasks corresponding to specialized models of the attribute for each category. In both cases, easily available semantic information (attribute semantics and category labels respectively) is exploited to help guide the selective sharing.

Roadmap In the rest of this chapter, we will first zoom in, one by one, to study the two above-listed instantiations of our general idea of selective sharing (as opposed

to indiscriminate "oversharing") during attribute learning, delving into their technical approaches and experimental results validating their usefulness. Specifically, in Section 2, we will focus on our method for learning decorrelated models for a vocabulary of visual attributes as described above, and Section 3, we will focus on our method for learning category-specific attribute classifiers. In Section 4, we will zoom back out to look at previous work that is relevant to the ideas discussed in this chapter. Finally, in Section 5, we will summarize our findings and outline areas for future work that may build on our ideas.

2 Learning Decorrelated Attributes

Many applications of visual attributes such as image search and zero-shot learning build on learned models for a *vocabulary* of multiple, diverse attributes, e.g., a detailed textual query in image search might describe various attributes of the desired target image. A key underlying challenge in learning discriminative models of multiple attributes is that the hypothesis space is very large. The standard discriminative model can associate an attribute with any direction in the feature space that happens to separate positive and negative instances in the training dataset, resulting very often in the learning of properties that are spuriously correlated with the attribute of interest. The issue is exacerbated by the fact that many nameable visual properties will occupy the same spatial region in an image. For example, a "brown" object might very well also be "round" and "shiny". In contrast, when learning object categories, each pixel is occupied by just one object of interest, decreasing the possibility of learning incidental classes. Furthermore, even if we attempt stronger training annotations, spatial extent annotation for attributes is harder and more ambiguous than it is for objects. Consider, for example, how one might mark the spatial extent of "pointiness" in the images in Figure 2.

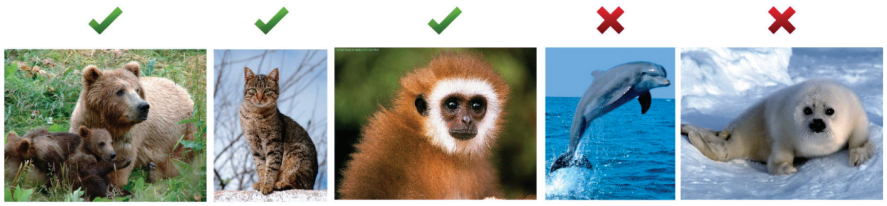


Fig. 2: What attribute is present in the first three images, but not the last two? Standard methods attempting to learn "furry" from such images are prone to learn "brown" instead—or some combination of correlated properties. We propose a multi-task attribute learning approach that resists the urge to share features between attributes that are semantically distinct yet often co-occur.

But does it even matter if we inadvertently learn a correlated attribute? After all, weakly supervised object recognition systems have long been known to exploit correlated background features appearing outside the object of interest that serve as

"context". For attribute learning, however, it is a problem, on two fronts. First of all, with the large number of possible combinations of attributes (up to 2^k for k binary attributes), we may see only a fraction of plausible ones during training, making it risky to treat correlated cues as a useful signal. In fact, semantic attributes are touted for their extendability to novel object categories, where correlation patterns may easily deviate from those observed in training data. Secondly, many attribute applications—such as image search [20,22,38], zero-shot learning [25], and textual description generation [9]—demand that the named property align meaningfully with the image content. For example, an image search user querying for "pointy-toed" shoes would be frustrated if the system (wrongly) conflates pointiness with blackness due to training data correlations. We contrast this with the object recognition setting, where object categories themselves may be thought of as co-occurring, correlated bundles of attributes. Learning to recognize an object thus implicitly involves learning these correlations.

Given these issues, our goal for the rest of this section is to decorrelate attributes at the time of learning, thus learning attribute classifiers that fire only when the correct semantic property is present. In particular, we want our classifiers to generalize to test images where the attribute co-occurrence patterns may differ from those observed in training. To this end, we propose a multi-task learning framework that encourages each attribute classifier to use a disjoint set of image features to make its predictions. This idea of feature *competition* is central to our approach.

As discussed in Section 1, whereas conventional models train each attribute classifier independently, and therefore are prone to re-using image features for correlated attributes, our multi-task approach *resists the urge to share*. Instead, it aims to isolate distinct low-level features for distinct attributes in a vocabulary by enforcing a structured sparsity prior over the attributes. We design this prior to leverage side information about the attributes' semantic relatedness, aligning feature *sharing* patterns with semantically close attributes and feature *competition* with semantically distant ones. In the example in Figure 2, the algorithm might discover that dimensions corresponding to color histogram bins should be used to detect "brown", whereas those corresponding to texture in the center of the image might be reserved to detect "furry".

2.1 Approach

In the following, we first describe the inputs to our algorithm: the semantic relationships among attributes (Section 2.1.1) and the low-level image descriptors (Section 2.1.2). Then we introduce our learning objective and optimization framework (Section 2.1.3), which outputs a classifier for each attribute in the vocabulary.

2.1.1 Semantic Attribute Groups

Suppose we are learning attribute classifiers[2] for a vocabulary of M nameable attributes, indexed by $\{1,2,\ldots,M\}$. To represent the attributes' semantic relationships, we use L attribute *groups*, encoded as L sets of indices $S_1,\ldots,S_L$, where each $S_l = \{m_1, m_2, m_3, \ldots\}$ contains the indices of the specific attributes in that group, and $1 \leq m_i \leq M$. While nothing in our approach restricts attribute groups to be disjoint, for simplicity in our experiments each attribute appears in one group only.

If two attributes are in the same group, this reflects that they have some semantic tie. For instance, in Figure 3, S_1 and S_2 correspond to texture and shape attributes respectively. For attributes describing fine-grained categories, like bird species, a group can focus on domain-specific aspects inherent to the taxonomy—for example, one group for beak shape (hooked, curved, dagger, etc.) and another group for belly color (red belly, yellow belly, etc.). While such groups could conceivably be mined automatically (from text data, WordNet, or other sources), we rely on existing manually defined groups [25, 48] in our experiments (see Figure 6).

As we will see below, group co-membership signals to our learning algorithm that the attributes are more likely to share features. For spatially localized attribute groups (e.g., beak shape), this could guide the algorithm to concentrate on descriptors originating from the same object part; for global attribute groups (e.g., colors), this could guide the algorithm to focus on a subset of relevant feature channels. There might be no such thing as a single "optimal" grouping; rather, we expect such partial side information about semantics to help intelligently decide when to allow sharing.

Our use of attribute label dimension-grouping to exploit relationships among tasks is distinct from and not to be confused with descriptor dimension grouping to represent *feature* space structure, as in the single-task "group lasso" [55]. While simultaneously exploiting feature space structure could conceivably further improve our method's results, we restrict our focus in this paper to modeling and exploiting *task relationships*.

2.1.2 Image Feature Representation

When designating the low-level image feature space where the classifiers will be learned, we are mindful of one main criterion: we want to expose to the learning algorithm *spatially localized* and *channel localized* features. By spatially localized, we mean that the image content within different local regions of the image should appear as different dimensions in an image's feature vector. Similarly, by channel localized, we mean that different types of descriptors (color, texture, etc.) should occupy different dimensions. This way, the learner can pick and choose a sparse set of both spatial regions and descriptor types that best discriminate attributes in one semantic group from another.

[2] We use "attribute", "classifier" and "task" interchangeably in this section.

To this end, we extract a series of histogram features for multiple feature channels pooled within grid cells at multiple scales. We reduce the dimension of each component histogram (corresponding to a specific window+feature type) using Principal Component Analysis (PCA). This alleviates gains from trivially discarding low-variance dimensions and isolates the effect of attribute-specific feature selection. Since we perform PCA *per channel*, we retain the desired localized modality and location associations in the final representation. More dataset-specific details are in the experiments below in Section 2.2.

2.1.3 Joint Attribute Learning with Feature Sharing and Competition

The input to our learning scheme is (i) the descriptors for N training images, each represented as a D-dimensional vector $\mathbf{x}_n$, (ii) the corresponding (binary) attribute labels for all attributes, which are indexed by $a = 1,\ldots,M$, and (iii) the semantic attribute groups $S_1,\ldots,S_L$. Let $\mathbf{X}_{N\times D}$ be the matrix composed by stacking the training image descriptors. We denote the n^{th} row of $\mathbf{X}$ as the row vector $\mathbf{x}_n$ and the d^{th} column of $\mathbf{X}$ as the column vector $\mathbf{x}^d$. The scalar x_n^d denotes the $(n,d)^{th}$ entry of $\mathbf{X}$. Similarly, the training attribute labels are represented as a matrix $\mathbf{Y}_{N\times M}$ with all entries $\in \{0,1\}$. The rows and columns of $\mathbf{Y}$ are denoted $\mathbf{y}_n$ and $\mathbf{y}^m$ respectively.

Because we wish to impose constraints on relationships between attribute models, we learn all attributes simultaneously in a multi-task learning setting, where each "task" corresponds to an attribute. The learning method outputs a parameter matrix $\mathbf{W}_{D\times M}$ whose columns encode the classifiers corresponding to the M attributes. We use logistic regression classifiers, with the loss function

$$L(\mathbf{X},\mathbf{Y};\mathbf{W}) = \sum_{m,n} \log(1+\exp((1-2y_n^m)\,\mathbf{x}_n^T\mathbf{w}^m)). \quad (1)$$

Each classifier has an entry corresponding to the "weight" of each feature dimension for detecting that attribute.

Note that a row $\mathbf{w}_d$ of $\mathbf{W}$ represents the usage of feature dimension d across all attributes; a zero in w_d^m means that feature d is not used for attribute m.

Formulation

Our method operates on the premise that semantically related attributes tend to be determined by (some of) the same image features, and that semantically distant attributes tend to rely on (at least some) distinct features. In this way, the support of an attribute in the feature space—that is, the set of dimensions with non-zero weight—is strongly tied to its semantic associations. Our goal is to effectively exploit the supplied semantic grouping by inducing (i) in-group feature sharing (ii) between-group competition for features. We encode this as a structured sparsity problem, where structure in the output attribute space is represented by the grouping. Figure 3 illustrates the envisioned effect of our approach.

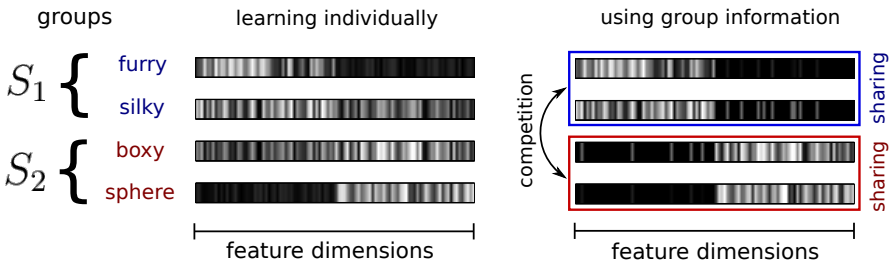


Fig. 3: Sketch of our idea. We show weight vectors (absolute value) for attributes learnt by standard (left) and proposed (right) approaches. The higher the weight (lighter colors) assigned to a feature dimension, the more the attribute relies on that feature. In this instance, our approach would help resolve "silky" and "boxy", which are highly correlated in training data and consequently conflated by standard learning approaches.

To set the stage for our method, we next discuss two existing sparse feature selection approaches, both of which we will use as baselines in Section 2.2. The first is a simple adaptation of the single-task lasso method [43]. The original lasso regularizer applied to learning a single attribute m in our setting would be $\|\mathbf{w}^m\|_1$. As is well known, this convex regularizer yields solutions that are a good approximation to sparse solutions that would have been generated by the count of non-zero entries, $\|\mathbf{w}^m\|_0$.

By summing over all tasks, we can extend single-task lasso [43] to the multi-task setting to yield an "all-competing" lasso minimization objective:

$$\mathbf{W}^* = \arg\min_{\mathbf{W}} L(\mathbf{X},\mathbf{Y};\mathbf{W}) + \lambda \sum_m \|\mathbf{w}^m\|_1, \tag{2}$$

where $\lambda \in \mathbb{R}$ is a scalar regularization parameter balancing sparsity against classification loss. Note that the regularizing second term may be rewritten $\sum_m \|\mathbf{w}^m\|_1 = \sum_d \|\mathbf{w}_d\|_1 = \|\mathbf{W}\|_1$. This highlights how the regularizer is symmetric with respect to the two dimensions of $\mathbf{W}$, and may be thought of, respectively, as (i) encouraging sparsity on each task column $\mathbf{w}^m$, and (ii) imposing sparsity on each feature row $\mathbf{w}_d$. The latter effectively creates competition among all tasks for the feature dimension d.

In contrast, the "all-sharing" ℓ_{21} multi-task lasso approach for joint feature selection [1] promotes sharing among all tasks, by minimizing the following objective function:

$$\mathbf{W}^* = \arg\min_{\mathbf{W}} L(\mathbf{X},\mathbf{Y};\mathbf{W}) + \lambda \sum_d \|\mathbf{w}_d\|_2. \tag{3}$$

To see that this encourages feature sharing among *all* attributes, note that the regularizer may be written as the ℓ_1 norm $\|\mathbf{V}\|_1 = \sum_d \|\mathbf{w}_d\|_2$, where the single-column matrix $\mathbf{V}$ is formed by collapsing the columns of $\mathbf{W}$ with the ℓ_2 operator, i.e. its d^{th} entry $v_d = \|\mathbf{w}_d\|_2$. The ℓ_1 norm of $\mathbf{V}$ prefers sparse-$\mathbf{V}$ solutions, which in turn means

the individual classifiers must only select features that also are helpful to other classifiers. That is, $\mathbf{W}$ should tend to have rows that are either all-zero or all-nonzero.

We now define our objective, which is a semantics-informed intermediate approach that lies between the extremes in Equation (2) and 3 above. Our minimization objective retains the competition-inducing ℓ_1 norm of the conventional lasso across groups, while also applying the ℓ_{21}-type sharing regularizer within every semantic group:

$$\mathbf{W}^* = \arg\min_{\mathbf{W}} L(\mathbf{X}, \mathbf{Y}; \mathbf{W}) + \lambda \sum_{d=1}^{D} \sum_{l=1}^{L} \|\mathbf{w}_d^{S_l}\|_2, \tag{4}$$

where $\mathbf{w}_d^{S_l}$ is a row vector containing a subset of the entries in row $\mathbf{w}_d$, namely, those specified by the indices in semantic group S_l. This regularizer restricts the column-collapsing effect of the ℓ_2 norm to within the semantic groups, so that $\mathbf{V}$ is no longer a single column vector but a matrix with L columns, one corresponding to each group. Figure 4 visualizes the idea. Note how sparsity on this $\mathbf{V}$ corresponds to promoting feature competition across unrelated attributes, while allowing sharing among semantically grouped attributes.

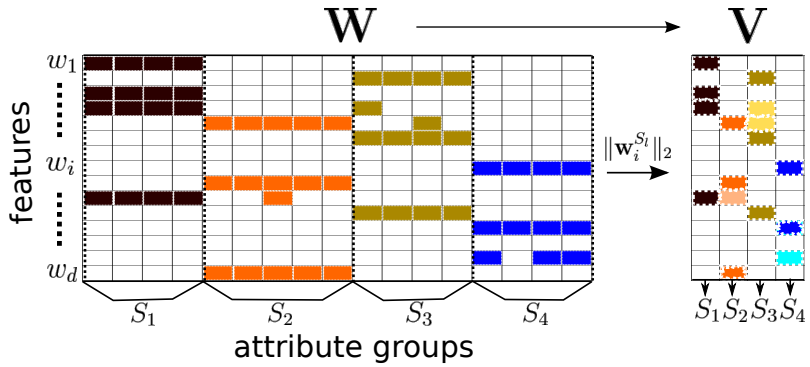
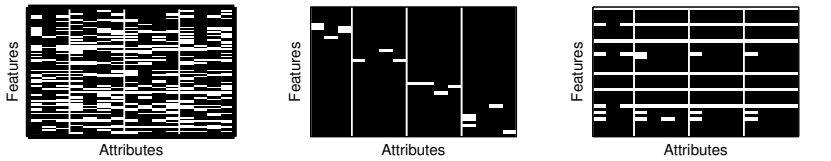


Fig. 4: "Collapsing" of grouped columns of the feature selection matrix $\mathbf{W}$ prior to applying the lasso penalty $\sum_l \|\mathbf{v}^l\|_1$. Non-zero entries in W and V are shaded. Darkness of shading in $\mathbf{V}$ represents how many attributes in that group selected that feature.

Our model unifies the previous formulations and represents an intermediate point between them. With only one group $S_1 = \{1, 2, \ldots, M\}$ containing all attributes, Equation (4) simplifies to Equation (3). Similarly, setting each attribute to belong to its own singleton group $S_m = \{m\}$ produces the lasso formulation of Equation (2). Figure 5 illustrates their respective differences in structured sparsity. While standard lasso aims to drop as many features as possible across all tasks, standard "all-sharing" aims to use only features that can be shared by multiple tasks. In contrast, the proposed method seeks features shareable among related attributes, while it resists feature sharing among less related attributes.

As we will show in results, this mitigates the impact of incidentally correlated attributes. Pushing attribute group supports away from one another helps decorre-



(a) Lasso (Equation (2)) (b) Proposed (Equation (4)) (c) All-sharing (Equation (3))

Fig. 5: A part of the **W** matrix (thresholded, absolute value) learned by the different structured sparsity approaches on CUB data. The thin white vertical lines separate attribute groups.

late unrelated attributes *within* the vocabulary. Even if "brown" and "furry" always co-occur at training time, there is pressure to select distinct features in their classifiers. Meanwhile, feature sharing within the group essentially pools in-group labels together for feature selection, mitigating the risk of chance correlations—not only within the vocabulary, but also with visual properties (nameable or otherwise) that are not captured in the vocabulary. For example, suppose "hooked beak" and "brown belly" are attributes that often co-occur; if "brown belly" shares a group with the easier-to-learn "yellow belly", the pressure to latch onto feature dimensions shareable between brown and yellow belly indirectly leads "hooked beak" towards disjoint features.

We stress, however, that the groups are only a prior. While our method prefers sharing for semantically related attributes, it is not a hard constraint, and misclassification loss also plays an important role in deciding which features are relevant.

2.1.4 Optimization

Mixed norm regularizations of the form of Equation (4), while convex, are non-smooth and non-trivial to optimize. Such norms appear frequently in the structured learning literature [1, 3, 19, 55]. As in [19], we reformulate the objective by representing the 2-norm in the regularizer in its dual form, before applying the smoothing proximal gradient descent [7] method to optimize a smooth approximation of the resulting objective. More details are in [17].

2.2 Experiments and Results

2.2.1 Datasets

We use three datasets with 422 total attributes: (i) CUB-200-2011 ("CUB") [48], (ii) Animals with Attributes ("AwA") [25], and (iii) aPascal/aYahoo ("aPY") [9]. Dataset statistics are summarized in Table 1. Following common practice, we separate the datasets into "seen" and "unseen" classes. The idea is to learn attributes on one set of seen object classes, and apply them to new unseen objects at test time.

Table 1: Summary of dataset statistics

	Categories		Attributes		Features	
Datasets	seen	unseen	num (M)	groups (L)	# windows	D
CUB-200-2011 (CUB) [48]	100	100	312	28	15	375
Animals with Attributes (AwA) [25]	40	10	85	9	1,21	290
aPascal/aYahoo-restricted (aPY-25) [9]	20	12	25	3	7	105

This stress-tests the generalization power, since correlation patterns will naturally deviate in novel objects. The seen and unseen classes for AwA and aPY come pre-specified. For CUB, we randomly select 100 of the 200 classes to be "seen".

2.2.2 Features

Section 2.1.2 defines the basic feature extraction process. On AwA, we use the features provided with the dataset (global bag-of-words on 4 channels, 3-level pyramid with $4\times4+2\times2+1=21$ windows on 2 channels). For CUB and aPY, we compute features with the authors' code [9]. On aPY, we use a one-level pyramid with $3\times2+1=7$ windows on four channels, following [9]. On CUB, we extract features at the provided annotated part locations. To avoid occluded parts, we restrict the dataset to instances that have the most common part visibility configuration (all parts visible except "left leg" and "left eye").

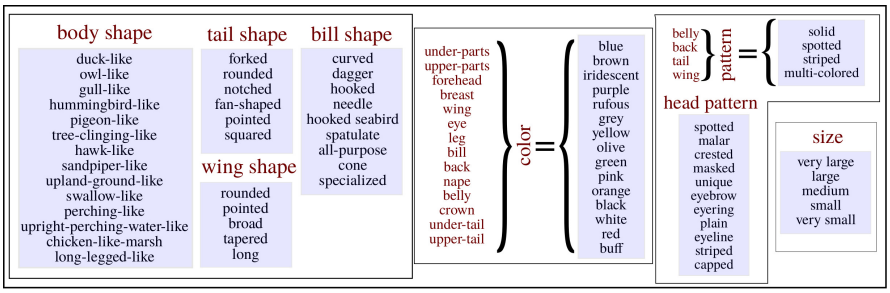
2.2.3 Semantic Groups

To define the semantic groups, we rely largely on existing data. CUB specifies 28 attribute groups [48] (head color, back pattern etc.). For AwA, the authors suggest 9 groups in [24] (color, texture, shape etc.). For aPY, which does not have pre-specified attribute groups, we group 25 attributes (of the 64 total) into shape, material and facial attribute groups guided by suggestions in [24] ("aPY-25"). The full groups are shown in Figure 6.

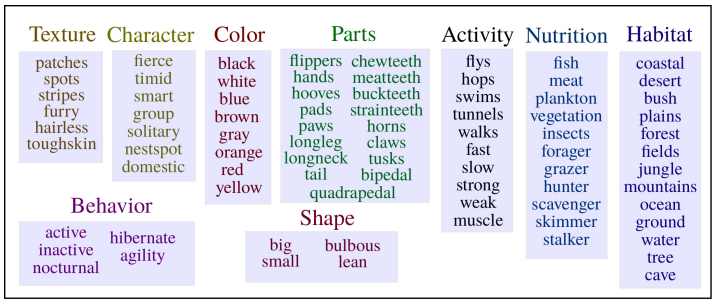
As discussed in Section 2.1.2, our method requires attribute groups and image descriptors to be mutually compatible. For example, grouping attributes based on their locations would not be useful if combined with a bag-of-words description that captures no spatial ordering. However, our results suggest that this compatibility is easy to satisfy. Our approach successfully exploits pre-specified attribute groups with independently pre-specified feature representations.

2.2.4 Baselines

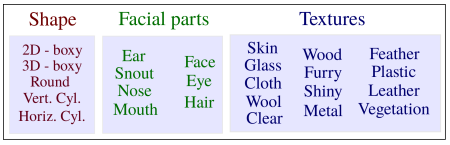
We compare to four methods throughout. Two are single-task learning baselines, in which each attribute is learned separately: (i) "standard": ℓ_2-regularized logistic regression, and (ii) "classwise": the object class-label based feature selection scheme



(a) Caltech-UCSD Birds (CUB) groups



(b) Animals with Attributes (AwA) groups



(c) aPascal (aPY-25) groups

Fig. 6: Semantic attribute groups on **(a)** CUB, **(b)** AwA and **(c)** aPY-25 datasets, as used in Section 2.2. Attribute groups are enclosed in shaded boxes, and phrases in larger font labeling the boxes indicate the rationale for the grouping. Additionally, in (a), the color and pattern groups, condensed above, are to be interpreted as follows. Each part on the left, coupled with the term in the middle (color/pattern) represents the title of an attribute group. The predicates on the right applied to each part constitute the attributes in the its group, *e.g.*, the "belly-color" attribute group has attributes "belly-color-blue", "belly-color-brown" *etc.*

proposed in [9]. The "classwise" method is, to our knowledge, the only previous work that attempts to explicitly decorrelate semantic attributes. For each attribute, the classwise method selects discriminative image features *for each object class*, then pools the selected features to learn the attribute classifier. For example, it first finds features good for distinguishing cars with and without "wheel", then buses with and without "wheel", etc. The idea is that examples from the same class help isolate the attribute of interest. For this baseline, we use logistic regression in the final stage replacing the SVM, for uniformity with the others. The other two baselines are the sparse multi-task methods in Section 2.1: (iii) "lasso" (Eq 2), and (iv)

"all-sharing" (Eq 3). All methods produce logistic regression classifiers and use the same input features. All parameters (λ for all methods, plus a second parameter for [9]) are validated with held out unseen class data.

2.2.5 Attribute Detection Accuracy

First, we test basic attribute detection accuracy. For this task, every test image is to be labeled with a binary label for each attribute in the vocabulary. Attribute models are trained on a randomly chosen 60% of the "seen" class data and tested on three test sets: (i) *unseen*: unseen class instances (ii) *all-seen*: other instances of seen classes and (iii) *hard-seen*: a subset of the all-seen set that is designed to consist of outliers within the seen-class distribution. To create the hard-seen set, we first compute a binary class-attribute association matrix as the thresholded mean of attribute labels for instances of each seen class. Then hard sets for each attribute are composed of instances that violate their class-level label for that attribute in the matrix, e.g. albino elephants (gray), cats with occluded ears (ear).

2.2.6 Overall Results

Table 2: Accuracy scores for attribute detection (AP×100). Higher is better. U, H and S refer respectively to *unseen*, *hard-seen* and *all-seen* test sets (Section 2.2.5). Our approach generally outperforms existing methods, and especially shines when attribute correlations differ between train and test data (i.e., the U and H scenarios).

Datasets	CUB			AwA		aPY-25		
Methods	U	H	S	U	S	U	H	S
lasso	17.83	25.52	22.19	52.74	61.75	27.13	29.25	31.84
all-sharing [1]	17.78	25.46	22.17	53.78	60.21	26.01	29.34	25.60
classwise [9]	19.09	27.56	24.06	N/A	N/A	27.29	27.76	35.95
standard	18.36	27.06	23.69	53.66	**66.87**	27.27	28.45	**37.72**
proposed	**21.14**	**29.62**	**26.54**	**54.97**	64.80	**29.89**	**33.18**	30.21

Table 2 shows the mean AP scores over all attributes, per dataset.[3] On all three datasets, our method generalizes better than all baselines to unseen classes and hard seen data.

While the "classwise" technique of [9] helps decorrelate attributes to some extent, improving over "standard" on aPY-25 and CUB, it is substantially weaker than the proposed method. That method assumes that same-object examples help isolate the attribute; yet, if two attributes always co-vary in the same-object examples (e.g., if cars with wheels are always metallic) then the method is still prone to exploit cor-

[3] AwA has only class-level attribute annotations, so (i) the classwise baseline [9] is not applicable and (ii) the "hard-seen" test set is not defined.

related features. Furthermore, the need for sufficient positive and negative attribute examples within each object class can be a practical burden (and makes it inapplicable to AwA). In contrast, our idea to jointly learn attributes and diffuse features between them is less susceptible to same-object correlations and does not make such label requirements. Our method outperforms this state-of-the-art approach on each dataset.

The two multi-task baselines (lasso and all-sharing) are typically weakest of all, verifying that semantics play an important role in deciding when to share. In fact, we found that the all-sharing/all-competing regularization generally hurt the models, leading the validated regularization weights λ to remain quite low.

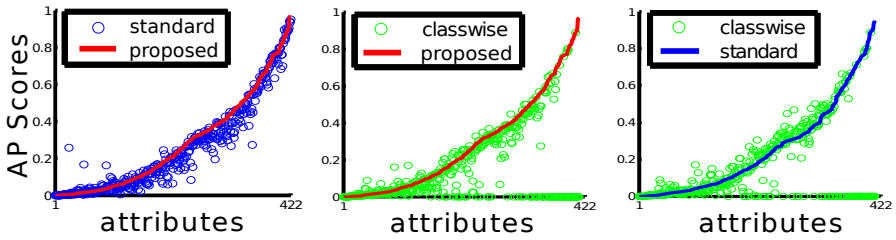


Fig. 7: Attribute detection results across all datasets (Section 2.2.5)

Figure 7 plots the unseen set results for the individual 422 attributes from all datasets. Here we show paired comparisons of the three best performing methods: proposed, classwise [9], and standard. For each plot, attributes are arranged in order of increasing detectability for one method.[4] For nearly all of the 422 attributes, our method outperforms both the standard learning approach (first plot) and state-of-the-art classwise method (second plot).

2.2.7 Evidence of "Learning the Right Thing"

Comparing results between the all-seen and hard-seen cases, we see evidence that our method's gains are due to its ability to preserve attribute semantics. On aPY-25 and AwA, our method *underperforms* the standard baseline on the all-seen set, whereas it *improves* performance on the unseen and hard-seen sets. This matches the behavior we would expect from a method that successfully resolves correlations in the training data: it generalizes better on novel test sets, sometimes at the cost of mild performance losses on test sets that have similar correlations (where a learner would benefit by learning the correlations).

In Figure 9(a), we present qualitative evidence in the form of cases that were mislabeled by the standard baseline but correctly labeled by our approach, e.g., the wedge-shaped "Flatiron" building (row 2, fourth from left) is correctly marked not

[4] Since "classwise" is inapplicable to AwA, its scores are set to 0 for that dataset (hence the circles along the x-axis in plots 2 and 3).

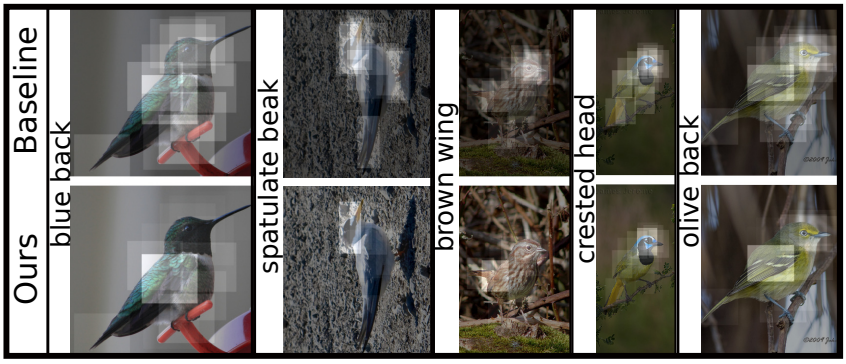
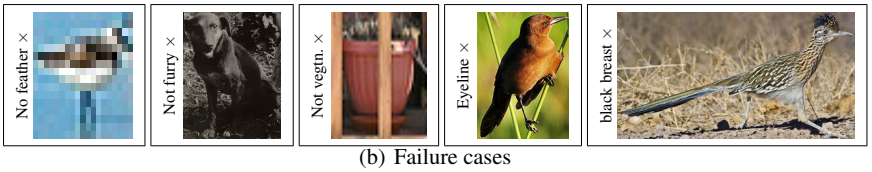


Fig. 8: Contributions of bird parts (shown as highlights) to the correct detection of specific attributes. Our method looks in the right places more often than the standard single-task baseline.

(a) Sucess cases



(b) Failure cases

Fig. 9: (a) Success cases: Annotations shown are our method's attribute predictions, which match ground truth. The logistic regression baseline ("standard") fails on all these cases. **(b) Failure cases:** Cases where our predictions (shown) are incorrect and the "standard" baseline succeeds.

"3D boxy" and the bird in the muck (row 2, end) is correctly marked as not having "brown underparts" because of the black grime sticking to it. In contrast, the baseline predicts the attribute based on correlated cues (e.g., city scenes are usually boxy, not wedge-shaped) and fails on these images.

Figure 9(b) shows some failure cases. Common failure cases for our method are when the image is blurred, the object is very small or information is otherwise deficient—cases where learning context from co-occurring aspects helps. In the low-resolution "feather" case, for instance, recognizing bird parts might have helped to correctly identify "feather".

Still more qualitative evidence that we preserve semantics comes from studying the features that influence the decisions of different methods. The part-based representation for CUB allows us to visualize the contributions of different bird parts to determine any given attribute. To find locations on instance number n that contribute to positive detection of attribute m, we take the absolute value of the element-wise product of descriptor $\mathbf{x}_n$ with the attribute weight vector $\mathbf{w}^m$—denote this $\mathbf{h}$. Each feature dimension is mapped onto the bird part it was computed from, in a mapping f. For each part p, we then compute its weight as $l_p = \sum_{f(i)=p} |h_i|$. These part weights are visualized as highlights in Fig 8.

Our method focuses on the proper spatial regions associated with the bird parts, whereas the baseline picks up on correlated features. For example, on the "brown wing" image, while the baseline focuses on the head, our approach almost exclusively highlights the wing.

2.2.8 Zero-shot Object Recognition

Table 3: Scores on zero-shot object recognition (accuracy). Higher is better.

Datasets	CUB	AwA	aPY-25
Methods	[100 cl]	[10 cl]	[12 cl]
lasso	7.35	25.32	9.88
all-sharing [1]	7.34	19.40	6.95
classwise [9]	9.15	N/A	20.00
standard	9.67	26.29	**20.09**
proposed	**10.70**	**30.64**	19.43

Next we show the impact of retaining attribute semantics for zero-shot object recognition. Closely following the setting in [25], the goal is to learn object categories from textual descriptions (e.g., "zebras are striped and four-legged"), but no training images, making attribute correctness crucial. We input attribute probabilities from each method's models to the Direct Attribute Prediction (DAP) framework for zero-shot learning [25].

Table 3 shows the results. Our method yields substantial gains in multi-class accuracy on the two large datasets (CUB and AwA). It is marginally worse than "standard" and "classwise" on the aPY-25 dataset, despite our significantly better attribute detection (Section 2.2.5). We believe that this may be due to recognition with DAP being less reliable when working with fewer attributes, as in aPY-25 (25 attributes).

2.2.9 Category Discovery with Semantic Attributes

Finally, we demonstrate the impact on category discovery. Cognitive scientists propose that natural categories are convex regions in *conceptual spaces* whose axes correspond to "psychological quality dimensions" [12]. This motivates us to perform category discovery with attributes. Treating semantic visual attributes as a conceptual space for visual categorization, we cluster each method's attribute presence probabilities (on unseen class instances) using k-means to discover the convex clusters. We set k to the true number of classes. We compare each method's clusters with the true unseen classes on all three datasets. For CUB, we test against both the 100 species (CUB-s) as well as the taxonomic families (CUB-f). Performance is measured using the normalized mutual information (NMI) score which measures the information shared between a given clustering and the true classes without requiring hard assignments of clusters to classes.

Table 4 shows the results. Our method performs significantly better than the baselines on all tasks. If we were to instead cluster the ground truth attribute signatures, we get a sense of the upper bound (last row). This shows that (i) visual attributes indeed constitute a plausible "conceptual space" for discovery and (ii) improved attribute learning models could yield large gains for high-level visual tasks.

Table 4: NMI scores for discovery of unseen categories (Section 2.2.9). Higher is better (0-100).

Methods / Datasets	CUB-s	AwA	aPY-25	CUB-f
lasso	54.85	18.91	19.15	35.03
all-sharing [1]	54.82	18.81	17.17	35.08
classwise [9]	57.46	N/A	19.73	38.62
standard	56.97	22.39	17.61	37.19
proposed	**59.44**	**24.11**	**24.76**	**42.81**
GT annotations	64.89	100.00	64.29	49.37

Before moving on to the second instantiation of our general idea for multi-task learning of attributes without oversharing, here is a summary of what we have learned so far. We have shown how to use semantics to guide attribute learning without oversharing across attributes. Through extensive experiments across multiple datasets, we have verified that: (i) our approach overcomes misleading training data correlations to successfully learn semantic visual attributes, and (ii) preserving semantics in learned attributes is beneficial as an intermediate step in high-level tasks.

3 Learning Analogous Category-Sensitive Attributes

In the previous section, we showed how to avoid oversharing features across different attributes by our proposed multi-task learning approach. In this section, we will

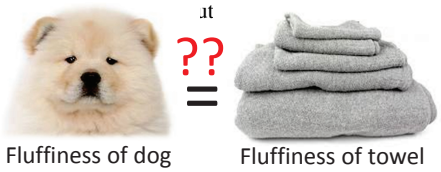


Fig. 10: Is fluffiness on a dog the same as fluffiness on a towel? Existing approaches assume an attribute such as "fluffy" can be used across different categories. However, as seen in here, in reality the same attribute name may refer to different visual properties for different categories.

move to a different instantiation of our idea for multi-task learning with selective sharing. Specifically, we are going to show how to learn *analogous category sensitive attributes*. These analogous attributes aim to prevent another aspect of oversharing: using a single universal attribute model across all object categories.

Intuitively, the conventional approach of universal attribute learning is an oversimplification. For example, as shown in Figure 10, fluffiness on a dog does not look the same as fluffiness on a towel. In this case, the attribute "fluffy" refers to different visual properties in different categories. Whereas above we encourage some features to be shared within certain attributes and keep some features disjoint between certain attributes, here we want to distinctively build a category-sensitive attribute for each category. Instead of sharing the attribute across categories, we utilize the correlation between attributes and categories during training.

What would it mean to have category-sensitive attribute predictions? At a glance it sounds like the other extreme from the current norm: rather than a single attribute model for all categories, one would train a single attribute model for each and every category. Furthermore, to learn accurate category-sensitive attributes, it seems to require category-sensitive training. For example, we could gather positive exemplar images for each category+attribute combination (e.g., separate sets of fluffy dog images, fluffy towel images). If so, this is a disappointment. Not only would learning attributes in this manner be quite costly in terms of annotations, but it would also fail to leverage the common semantics of the attributes that remain in spite of their visual distinctions.

To resolve this problem, we introduce a novel form of transfer learning to infer category-sensitive attribute models. Intuitively, even though an attribute's appearance may be specialized for a particular object, there likely are latent variables connecting it to other objects' manifestations of the property. Plus, some attributes *are* quite similar across some class boundaries (e.g., spots look similar on Dalmatian dogs and Pinto horses). Having learned some category-sensitive attributes, then, we ought to be able to predict how the attribute might look on a new object, *even without labeled examples depicting that object with the attribute*. For example, in Figure 11, suppose we want to recognize striped dogs, but we have no separate curated set of striped-dog exemplars. Having learned "spotted", "brown", etc. classifiers for dogs, cats, and equines, the system should leverage those models to infer what "striped" looks like on a dog. For example, it might infer that stripes on a dog look somewhat like stripes on a zebra but with shading influenced by the shape dogs share with cats.

Based on this intuition, we show how to infer an *analogous attribute*—an attribute classifier that is tailored to a category, even though we lack annotated exam-

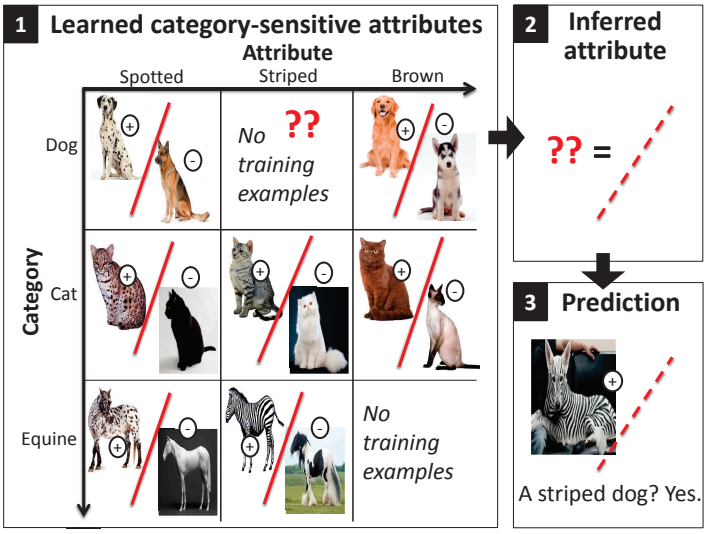


Fig. 11: Having learned a sparse set of object-specific attribute classifiers, our approach infers *analogous attribute classifiers*. The inferred models are object-sensitive, despite having no object-specific labeled images of that attribute during training.

ples of that category exhibiting that attribute. Given a sparse set of category-sensitive attribute classifiers, our approach first discovers the latent structure that connects them, by factorizing a tensor indexed by categories, attributes, and classifier dimensions. Then, we use the resulting latent factors to complete the tensor, inferring the "missing" classifier parameters for any object+attribute pairings unobserved during training. As a result, we can create category-sensitive attributes with only partial category-sensitive labeled data. Our solution offers a middle ground between completely category-independent training (the norm today [9, 23, 25, 32, 33, 36]) and completely category-sensitive training. We do not need to observe all attributes isolated on each category, and we capitalize on the fact that some categories and some of their attributes share common parameters.

Analogous attributes can be seen as a form of transfer learning. Existing transfer learning approaches for object recognition [2, 4, 10, 27, 30, 34, 44, 50, 53] aim to learn a new object category with few labeled instances by exploiting its similarity to previously learned class(es). While often the source and target classes must be manually specified [2, 4, 50], some techniques automatically determine which classes will benefit from transfer [16, 27, 44]. [30] uses class co-occurrence statistics to infer classifier weights for a given concept from those of related visual concepts. Different from them, our goal is to reduce labeled data requirements. More importantly, our idea for transfer learning jointly in two label spaces is new, and, unlike the prior work, we can infer new classifiers without training examples. See Section 4 for further discussion of related work.

3.1 Approach

Given training images labeled by their category and one or more attributes, our method produces a series of category-sensitive attribute classifiers. Some of those classifiers are explicitly trained with the labeled data, while the rest are inferred by our method. We show how to create these analogous attribute classifiers via tensor completion.

3.1.1 Learning Category-Sensitive Attributes

In existing systems, attributes are trained in a category-independent manner [5, 9, 22, 23, 25, 32, 33, 36, 38]. Positive exemplars consist of images from various object categories, and they are used to train a discriminative model to detect the attribute in novel images. We will refer to such attributes as *universal*.

Here we challenge the convention of learning attributes in a completely category-independent manner. As discussed above, while attributes' visual cues are often shared among *some* objects, the sharing is not universal. It can dilute the learning process to pool cross-category exemplars indiscriminately.

The naive solution to instead train *category-sensitive* attributes would be to partition training exemplars by their category labels, and train one attribute per category. Were labeled examples of all possible attribute+object combinations abundantly available, such a strategy might be sufficient. However, in initial experiments with large-scale datasets, we found that this approach is actually inferior to training a single universal attribute. We attribute this to two things: (i) even in large-scale collections, the long-tailed distribution of object/scene/attribute occurrences in the real world means that some label pairs will be undersampled, leaving inadequate exemplars to build a statistically sound model, and (ii) this naive approach completely ignores attributes' inter-class semantic ties.

To overcome these shortcomings, we instead use an importance-weighted support vector machine (SVM) to train each category-sensitive attribute. Let each training example $(\mathbf{x}_i, y_i)$ consist of an image descriptor $\mathbf{x}_i \in \Re^D$ and its binary attribute label $y_i \in \{-1, 1\}$. Suppose we are learning "furriness" for dogs. We use examples from all categories (dogs, cats, etc.), but place a higher penalty on violating attribute label constraints for the same category (the dog instances). This amounts to an SVM objective for the hyperplane $\mathbf{w}$:

$$\begin{aligned} \text{minimize} \quad & \left(\frac{1}{2}||\mathbf{w}||^2 + C_s \sum_i \xi_i + C_o \sum_j \gamma_j \right) \\ s.t. \quad & y_i \mathbf{w}^T \mathbf{x}_i \geq 1 - \xi_i; \quad \forall i \in \mathscr{S} \\ & y_j \mathbf{w}^T \mathbf{x}_j \geq 1 - \gamma_j; \quad \forall j \in \mathscr{O} \\ & \xi_i \geq 0; \gamma_j \geq 0, \end{aligned} \tag{5}$$

where the sets $\mathscr{S}$ and $\mathscr{O}$ denote those training instances in the same-class (dog) and other classes (non-dogs), respectively, and C_s and C_o are slack penalty constants. Note, $\mathscr{S}$ and $\mathscr{O}$ contain both positive and negative examples for the attribute in consideration.

Instance re-weighting is commonly used, e.g., to account for label imbalance between positives and negatives. Here, by setting $C_o < C_s$, the out-of-class examples of the attribute serve as a simple prior for which features are relevant. This way we benefit from more training examples when there are few category-specific examples of the attribute, but we are inclined to ignore those that deviate too far from the category-sensitive definition of the property.

3.1.2 Object-Attribute Classifier Tensor

Next we define a tensor to capture the structure underlying many such category-sensitive models. Let $m = 1, \ldots, M$ index the M possible attributes in the vocabulary, and let $t = 1, \ldots, T$ index the T possible object/scene categories. Let $\mathbf{w}(t,m)$ denote a category-sensitive SVM weight vector trained for the t-th object and m-th attribute using Equation (5).

We construct a 3D tensor $\mathsf{W} \in \Re^{T \times M \times D}$ using all available category-sensitive models. Each entry w_d^{tm} contains the value of the d-th dimension of the classifier $\mathbf{w}(t,m)$. For a linear SVM, this value reflects the impact of the d-th dimension of the feature descriptor $\mathbf{x}$ for determining the presence/absence of attribute m for the object class t.

The resulting tensor is quite sparse. We can only fill entries for which we have class-specific positive and negative training examples for the attribute of interest. In today's most comprehensive attribute datasets [33, 36], this means only $\sim 25\%$ of the possible object-attribute combinations can be trained in a category-sensitive manner. Rather than resort to universal models for those "missing" combinations, we propose to use the latent factors for the observed classifiers to synthesize analogous models for the unobserved classifiers, as we explain next.

3.1.3 Inferring Analogous Attributes

Having learned how certain attributes look for certain object categories, our goal is to transfer that knowledge to hypothesize how the same attributes will look for other object categories. In this way, we aim to infer analogous attributes: category-sensitive attribute classifiers for objects that lack attribute-labeled data. We pose the "missing classifier" problem as a tensor completion problem.

Matrix (tensor) completion techniques have been used in vision, from bi-linear models for separating style and content [11], to multi-linear models separating the modes of face image formation (e.g., identity vs. expression vs. pose) [46,47]. While often applied for visualization, the discovered factors can also be used to impute

missing data—for example, to generate images of novel fonts [11] or infer missing pixels for in-painting tasks [28].

Different from the existing work, we want to use tensor factorization to infer *classifiers*, not data instances or labels. This enables a new "zero-shot" transfer protocol: we leverage the latent factors underlying previously trained models to create new analogous ones without any labeled instances. Our goal is to recover the latent factors for the 3D object-attribute tensor W, and use them to impute the unobserved classifier parameters.

Let $\mathbf{O} \in \Re^{K \times T}$, $\mathbf{A} \in \Re^{K \times M}$, and $\mathbf{C} \in \Re^{K \times D}$ denote matrices whose columns are the K-dimensional latent feature vectors for each object, attribute, and classifier dimension, respectively. We assume that w_d^{tm} can be expressed as an inner product of latent factors,

$$w_d^{tm} \approx \langle O_t, A_m, C_d \rangle, \tag{6}$$

where a subscript denotes a column of the matrix. In matrix form, we have $\mathsf{W} \approx \sum_{k=1}^{K} O^k \circ A^k \circ C^k$, where a superscript denotes the row in the matrix, and $\circ$ denotes the vector outer product.

The latent factors of the tensor W are what affect how the various attributes, objects, and image descriptors covary. What might they correspond to? We expect some will capture mixtures of two or more attributes, e.g., factors distinguishing how "spots" appear on something "flat" vs. how they appear on something "bumpy". The latent factors can also capture useful clusters of objects, or supercategories, that exhibit attributes in common ways. Some might capture other attributes beyond the M portrayed in the training images—namely, those that help explain structure in the objects and other attributes we have observed.

We use Bayesian probabilistic tensor factorization [52] to recover the latent factors. Using this model, the likelihood for the explicitly trained classifiers (Section 3.1.1) is

$$p(\mathsf{W}|\mathbf{O},\mathbf{A},\mathbf{C},\alpha) = \Pi_{t=1}^{T}\Pi_{m=1}^{M}\Pi_{d=1}^{D}\left[\mathcal{N}(w_d^{tm}|\langle O_t, A_m, C_d\rangle, \alpha^{-1})\right]^{I_{tm}},$$

where $\mathcal{N}(w|\mu,\alpha)$ denotes a Gaussian with mean μ and precision α, and $I_{tm} = 1$ if object t has an explicit category-sensitive model for attribute m, and $I_{tm} = 0$ otherwise. For each of the latent factors O_t, A_m, and C_d, we use Gaussian priors. Let Θ represent all their means and covariances. Following [52], we compute a distribution for each missing tensor value by integrating out all model parameters and hyper-parameters, given all the observed attribute classifiers:

$$p(\hat{w}_d^{tm}|\mathsf{W}) = \int p(\hat{w}_d^{tm}|O_t, A_m, C_d, \alpha) p(\mathbf{O},\mathbf{A},\mathbf{C},\alpha,\Theta|\mathsf{W})\ d\{\mathbf{O},\mathbf{A},\mathbf{C},\alpha,\Theta\}.$$

After initializing with the MAP estimates of the three factor matrices, this distribution is approximated using Markov chain Monte Carlo (MCMC) sampling:

$$p(\hat{w}_d^{tm}|\mathsf{W}) \approx \sum_{l=1}^{L} p(\hat{w}_d^{tm}|O_n^{(l)}, A_m^{(l)}, C_d^{(l)}, \alpha^{(l)}). \tag{7}$$

Each of the L samples $\{O_t^{(l)}, A_m^{(l)}, C_d^{(l)}, \alpha^{(l)}\}$ is generated with Gibbs sampling on a Markov chain whose stationary distribution is the posterior over the model parameters and hyper-parameters. We use conjugate distributions as priors for all the Gaussian hyper-parameters to facilitate sampling. See [52] for details.

We use these factors to generate analogous attributes. Suppose we have no labeled examples showing an object of category t with attribute m (or, as is often the case, we have so few that training a category-sensitive model is problematic). Despite having no training examples, we can use the tensor to directly infer the classifier parameters

$$\hat{\mathbf{w}}(t,m) = [\hat{w}_1^{tm}, \ldots, \hat{w}_D^{tm}], \tag{8}$$

where each $\hat{w}_d^{tm}$ is the mean of the distribution in Equation (7).

3.1.4 Discussion

In this approach, we use factorization to infer *classifiers* within a tensor representing two inter-related label spaces. Our idea has two key useful implications. First, it leverages the interplay of both label spaces to generate new classifiers without seeing any labeled instances. This is a novel form of transfer learning. Second, by working directly in the classifier space, we have the advantage of first isolating the low-level image features that are informative for the observed attributes. This means the input training images can contain realistic (un-annotated) variations. In comparison, existing data tensor approaches often assume a strict level of alignment; e.g., for faces, examples are curated under t specific lighting conditions, m specific expressions, etc. [46, 47].

Our design also means that the analogous attributes can transfer information from multiple objects and/or attributes simultaneously. That means, for example, our model is not restricted to transferring the fluffiness of a dog from the fluffiness of a cat; rather, its analogous model for dog fluffiness might just as well result from transferring a mixture of cues from carpet fluffiness, dog spottedness, and cat shape.

In general, transfer learning can only succeed if the source and target classes are related. Similarly, we will only find an accurate low-dimensional set of factors if some common structure exists among the explicitly trained category-sensitive models. Nonetheless, a nice property of our formulation is that even if the tensor is populated with a variety of classes—some with no ties—analogous attribute inference can still succeed. Distinct latent factors can cover the different clusters in the observed classifiers. For similar reasons, our approach naturally handles the question of "where to transfer": sources and targets are never manually specified. Below, we consider the impact of building the tensor with a large number of semantically diverse categories versus a smaller number of closely related categories.

3.2 Experiments and Results

We evaluate our approach on two datasets: the attribute-labeled portion of ImageNet [36] and SUN Attributes [33]. See Figure 12 for example images of these two datasets. The datasets do not contain data for all possible category-attribute pairings. Figure 13 shows which are available: there are 1,498 and 6,118 pairs in ImageNet and SUN, respectively. The sparsity of these matrices actually underscores the need for our approach, if one wants to learn category-sensitive attributes.

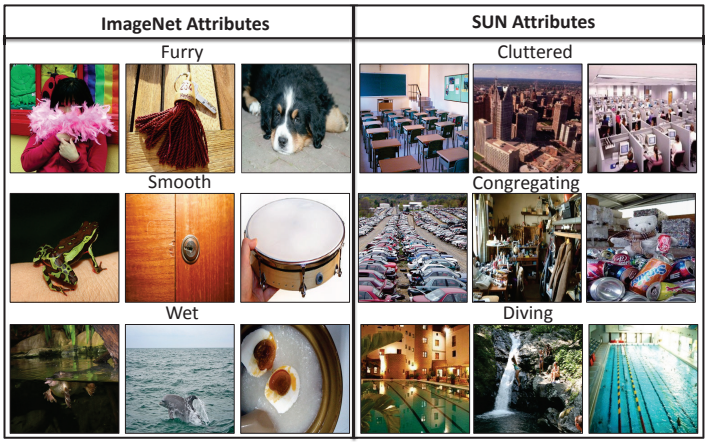


Fig. 12: Example images of ImageNet [36] and SUN Attributes [33] dataset.

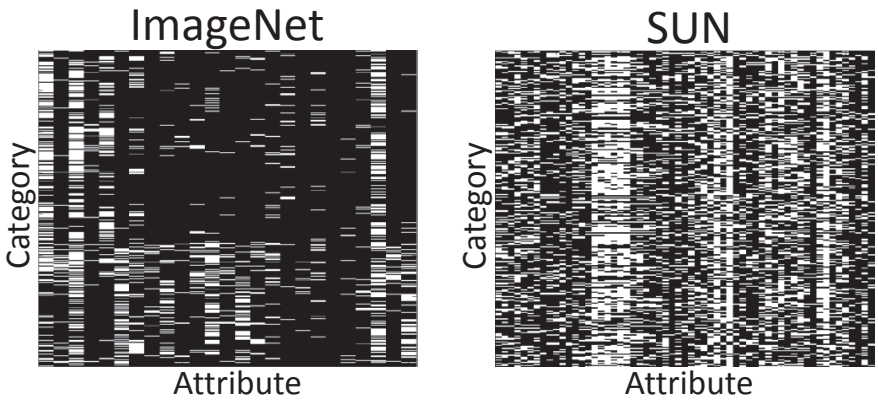


Fig. 13: Data availability: white entries denote category-attribute pairs that have positive and negative image exemplars. In ImageNet, most vertical stripes are color attributes, and most horizontal stripes are man-made objects. In SUN, most vertical stripes are attributes that appear across different scenes, such as vacationing or playing, while horizontal stripes come from scenes with varied properties, such as airport and park.

3.2.1 Category-Sensitive vs. Universal Attributes

First we test whether category-sensitive attributes are even beneficial. We explicitly train category-sensitive attribute classifiers using importance-weighted SVMs, as described in Section 3.1.1. This yields 1,498 and 6,118 classifiers for ImageNet and SUN, respectively. We compare their predictions to those of universal attributes, where we train one model for each attribute. When learning an attribute, both models have access to the exact same images; the universal method ignores the category labels, while the category-sensitive method puts more emphasis on the in-category examples.

Table 5: Accuracy (mAP) of attribute prediction. Category-sensitive models improve over standard universal models, and our inferred classifiers nearly match their accuracy with no training image examples. Traditional forms of transfer (rightmost two columns) fall short, showing the advantage of exploiting the 2D label space for transfer, as we propose.

	Datasets		Trained explicitly		Trained via transfer			
	# Categ (N)	# Attr (M)	Category-sens.	Universal	Inferred (Ours)	Adopt similar	One-shot	Chance
ImageNet	384	25	**0.7304**	0.7143	**0.7259**	0.6194	0.6309	0.5183
SUN	280	59	**0.6505**	0.6343	**0.6429**	N/A	N/A	0.5408

Table 5 (columns 4 and 5) shows the results, in terms of mean average precision across all 84 attributes and 664 categories. Among those, our category-sensitive models meet or exceed the universal approach 76% of the time. This indicates that the status quo [9, 23, 25, 32, 33, 36] pooling of training images across categories is indeed detrimental.

3.2.2 Inferring Analogous Attributes

The results so far establish that category-sensitive attributes are desirable. However, the explicit models above are *impossible to train for 18,000 of the ∼26,000 possible attributes in these datasets.* This is where our method comes in. It can infer all remaining 18,000 attribute models even without class-specific labeled training examples.

We perform leave-one-out testing: in each round, we remove one observed classifier (a white entry in Figure 13), and infer it with our tensor factorization approach. Note that even though we are removing one at a time, the full tensor is always quite sparse due to the available data. Namely, only 16% (in ImageNet) and 37% (in SUN) of all possible category-sensitive classifiers can be explicitly trained.

Table 5 (columns 4 to 6) shows this key result. In this experiment, the explicitly trained category-sensitive result is the "upper bound"; it shows how well the model trained with real category-specific images can do. We see that our inferred analogous attributes (column 6) are nearly as accurate, yet use zero category-specific labeled images. They approximate the explicitly trained models well. Most importantly, our inferred models remain more accurate than the universal approach. Our inferred attributes again meet or exceed the universal model's accuracy 79% of the time.

We stress that our method infers models for *all* missing attributes. That is, using the explicitly trained attributes, it infers another $8,064$ and $10,407$ classifiers on ImageNet and SUN, respectively. While the category-sensitive method would require approximately 20 labeled examples per classifier to train those models, our method uses zero.

Table 5 also compares our approach to conventional transfer learning. The first transfer baseline infers the missing classifier simply by adopting the category-sensitive attribute of the category that is semantically closest to it, where semantic distance is measured via WordNet using [8] (not available for SUN). For example, if there are no furry-dog exemplars, we adopt the wolf's "furriness" classifier. The second transfer baseline additionally uses one category-specific image example to perform "one-shot" transfer (e.g., it trains with both the furry-wolf images plus a furry-dog example). Unlike the transfer baselines, our method uses neither prior knowledge about semantic distances nor labeled class-specific examples. We see that our approach is substantially more accurate than both transfer methods. This result highlights the benefit of our novel approach to transfer, which leverages both label spaces (categories and their attributes) simultaneously.

Which attributes does our method transfer? That is, which objects does it find to be analogous for an attribute? To examine this, we first take a category j and identify its neighboring categories in the latent feature space, i.e., in terms of Euclidean distance among the columns of $\mathbf{O} \in \Re^{K \times T}$. Then, for each neighbor i, we sort its attribute classifiers ($\mathbf{w}(i,:)$, real or inferred) by their maximal cosine similarity to any of category j's attributes $\mathbf{w}(j,:)$. The resulting shortlist helps illustrate which attribute+category pairs our method expects to transfer to category j.

Figure 14 shows 4 such examples, with one representative image for each category. We see neighboring categories in the latent space are often semantically related (e.g., syrup/bottle) or visually similar (e.g., airplane cabin/conference center); although our method receives no explicit side information on semantic distances, it discovers these ties through the observed attribute classifiers. Some semantically more distant neighbors (e.g., platypus/rorqual, courtroom/cardroom) are also discovered to be amenable to transfer. The words in Figure 14 are the neighboring categories' top 3 analogous attributes for the numbered category to their left (*not* attribute predictions for those images). It seems quite intuitive that these would be suited for transfer.

Next we look more closely at where our method succeeds and fails. Figure 15 shows the top (bottom) five category+attribute combinations for which our inferred classifiers most increase (decrease) the AP, per dataset. As expected, we see our method most helps when the visual appearance of the attribute on an object is quite different from the common case, such as "spots" on the killer whale. On the other hand, it can detract from the universal model when an attribute is more consistent in appearance, such as "black", or where more varied examples help capture a generic concept, such as "symmetrical".

Figure 16 shows qualitative examples that support these findings. We show the image for each method that was predicted to most confidently exhibit the named attribute. By inferring analogous attributes, we better capture object-specific prop-

Fig. 14: Analogous attribute examples for ImageNet (top) and SUN (bottom). Words above each neighbor indicate the 3 most similar attributes (learned or inferred) between leftmost query category and its neighboring categories in latent space. For these four examples, [Query category]:[Neighbor categories] = (1) [Bottle]:[filter, syrup, bullshot, gerenuk] (2) [Platypus]:[giraffe, ungulate, rorqual, patas] (3) [Airplane cabin]:[aquarium, boat deck, conference center, art studio] (4) [Courtroom]: [cardroom, florist shop, performance arena, beach house]

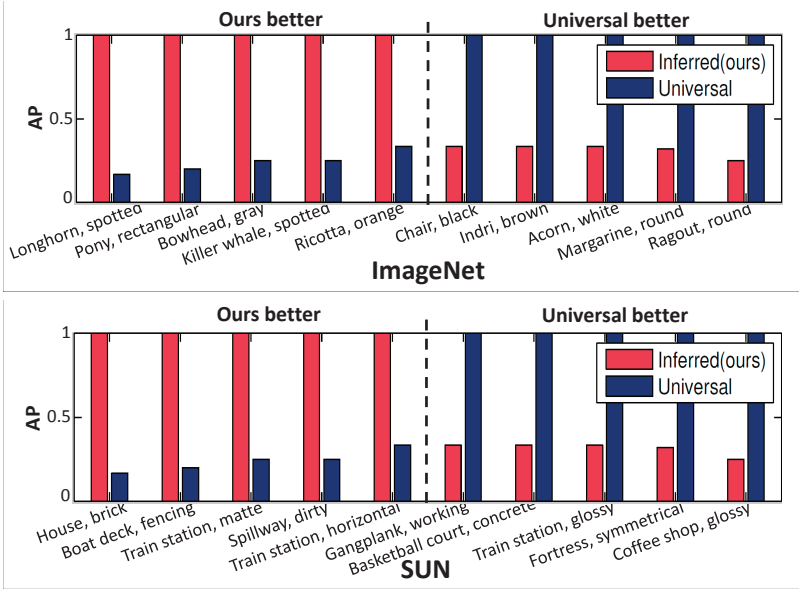


Fig. 15: *(Category,attribute)* pairs for which our inferred models most improve (left) or hurt (right) the universal baseline.

Fig. 16: Test images that our method (top row) and the universal method (bottom row) predicted most confidently as having the named attribute. (✓ = positive for the attribute, X = negative, according to ground truth.)

erties. For example, while our method correctly fires on a "smooth wheel", the universal model mistakes a Ferris Wheel as "smooth", likely due to the smoothness of the background, which might look like other classes' instantiations of smoothness.

3.2.3 Focusing on Semantically Close Data

In all results so far, we make no attempt to restrict the tensor to ensure semantic relatedness. The fact our method succeeds in this case indicates that it is capable of discovering clusters of classifiers for which transfer is possible, and is fairly resistant to negative transfer.

Still, we are curious whether restricting the tensor to classes that have tight semantic ties could enhance performance. We therefore test two variants: one where we restrict the tensor to closely related objects (i.e., downsampling the rows), and one where we restrict it to closely related attributes (i.e., downsampling the columns). To select a set of closely related objects, we use WordNet to extract sibling synsets for different types of dogs in ImageNet. This yields 42 categories, such as *puppy, courser, coonhound, corgi*. To select a set of closely related attributes, we extract only the color attributes.

Table 6: Attribute label prediction mAP when restricting the tensor to semantically close classes. The explicitly trained category-sensitive classifiers serve as an upper bound.

Subset	Category-sensitive	Inferred (subset)	Inferred (all)
Categories (dogs)	0.7478	0.7358	0.7173
Attributes (colors)	0.7665	0.7631	0.7628

Table 6 shows the results. We use the same leave-one-out protocol of Section 3.2.2, but during inference we only consider category-sensitive classifiers

among the selected categories/attributes. We see that the inferred attributes are stronger with the category-focused tensor, raising accuracy from 0.7173 to 0.7358, closer to the upper bound. This suggests that among the entire dataset, attributes for which categories differ can introduce some noise into the latent factors. On the other hand, when we ignore attributes unrelated to color, the mAP of the inferred classifiers remains similar. This may be because color attributes use such a distinct set of image features compared to others (like stripes, round) that the latent factors accounting for them are coherent with or without the other classifiers in the mix. From this preliminary test, we can conclude that when semantic side information is available, it could boost accuracy, yet our method achieves its main purpose even when it is not.

4 Related Work

In this section we describe related work in more detail and highlight contrasts and connections with the two main contributions described above.

4.1 Attributes as semantic features

A visual attribute is a binary predicate for an image that indicates whether or not a property is present and the standard approach to learn an attribute is to pool images regardless of their object category and train a discriminative classifier [5, 9, 22, 23, 25, 26, 32, 33, 36, 38].

While this design is well-motivated by the goal of having attributes that transcend category boundaries, it sacrifices accuracy in practice. We are not aware of any prior work that learns category-sensitive attributes, though class-specific attribute training is used as an intermediate feature generation procedure in [9, 51], prior to training class-independent models.

Recent research focuses on attributes as vehicles of semantics in human-machine communication. For example, using attributes for image search lets a user specify precise semantic queries ("find smiling Asian men") [20, 22, 38]; using them to augment standard training labels offers new ways to teach vision systems about objects ("zebras are striped", "this bird has a yellow belly", etc.) [5, 25, 26, 40]; deviations from an expected configuration of attributes may be used to generate textual descriptions of what humans would find remarkable [9, 37]. In all such applications, learning attributes incorrectly (such as by inadvertently learning correlated visual properties) or imprecisely (such as by learning a "lowest common denominator" model shared across all categories) is a real problem; the system and user's interpretations must align for their communication to be meaningful. However, despite all the attention to attribute applications, there is very little work on *how to learn attributes accurately*, preserving their semantics. The approaches presented in Sec-

tion 2 and Section 3 show promise for such applications that require "learning the right thing" when learning semantic attributes.

4.2 Attribute correlations

While most methods learn attributes independently, some initial steps have been taken towards modeling their relationships. Modeling co-occurrence between attributes helps ensure predictions follow usual correlations, even if image evidence for a certain attribute is lacking (e.g., "has-ear" usually implies "has-eye") [25, 41, 42, 51]. Our goal in decorrelating attributes (Section 2) is essentially the opposite of these approaches. Rather than equate co-occurrences with true semantic ties, we argue that it is often crucial that the learning algorithm avoid conflating pairs of attributes. This will prevent excessive biasing of the likelihood function towards the training data and thus deal better with unfamiliar configurations of attributes in novel settings.

While attribute learning is typically considered separately from object category learning, some recent work explores how to jointly learn attributes and objects, either to exploit attribute correlations [51], to promote feature sharing [15, 49], or to discover separable features [39, 54]. Our framework in Section 3 can be seen as a new way to jointly learn multiple attributes, leveraging structure in object-attribute relationships. Unlike any prior work, we use these ties to directly infer category-sensitive attribute models without labeled exemplars.

In [14], analogies between object categories are used to regularize a semantic label embedding. Our method also captures beyond-pairwise relationships, but the similarities end there. In [14], explicit analogies are given as input, and the goal is to enrich the features used for nearest neighbor object recognition. In contrast, our approach in Section 3 implicitly *discovers* analogical relationships among *object-sensitive attribute classifiers*, and our goal is to generate novel category-sensitive attribute classifiers.

4.3 Differentiating attributes

As discussed above, to our knowledge, the only previous work that attempts to explicitly decorrelate semantic attributes like we attempt in Section 2 is the classwise method of [9]. For each attribute, it selects discriminative image features *for each object class*, then pools the selected features to learn the attribute classifier. While the idea is that examples from the same class help isolate the attribute of interest, as seen above, this method is susceptible to learning chance correlations among the reduced number of samples of individual classes. Moreover, it requires expensive instance-wise attribute annotations. Our decorrelating attributes approach (Sec-

tion 2) overcomes these issues, as we demonstrate with experimental comparisons to [9] in Section 2.2.

While this is the only prior work on decorrelating *semantic* attributes, some unsupervised approaches attempt to diversify discovered (un-named/non-semantic) "attributes" [9, 29, 54]—for example by designing object class splits that yield uncorrelated features [54] or converting redundant semantic attributes into discriminative ones [29]. In contrast, our focus in Section 2 is on jointly learning a specified vocabulary of *semantic* attributes.

4.4 Multi-task learning (MTL)

Multi-task learning jointly trains predictive functions for multiple tasks, often by selecting the feature dimensions ("supports") each function should use to meet some criterion. Most methods emphasize feature *sharing* among all classes [1, 19, 31]; e.g., feature sharing between objects can yield faster detectors [45], and sharing between objects and their attributes can isolate features suitable for both tasks [15,49]. A few works have begun to explore the value of modeling *negative* correlations [13, 35, 56, 57]. For example, in a hierarchical classifier, feature competition is encouraged via disjoint sparsity or "orthogonal transfer", in order to remove redundancies between child and parent node classifiers [13, 56]. These methods exploit the inherent mutual exclusivity among object labels, which does not hold in our attributes setting. Unlike any of these approaches, in our decorrelating attributes method (Section 2), we model semantic structure in the target space using multiple task groups.

While most MTL methods enforce joint learning on all tasks, a few explore ways to discover groups of tasks that can share features [16, 18, 21]. Our method for decorrelating attributes (Section 2) involves grouped tasks, but with two crucial differences: (i) we explicitly model between-group *competition* along with in-group sharing to achieve inter-group decorrelation, and (ii) we treat external knowledge about semantic groups as supervision to be exploited during learning. In contrast, the prior methods [16, 18, 21] discover task groups from data, which is prone to suffer from correlations in the same way as a single-task learner.

In Section 3, we argue for modeling even single attributes through multiple category-specific models, all learned in a multi-task learning framework. While the idea of inferring classifier weights for one task from those learned for other tasks is relatively unexplored, [30] recently estimates a classifier for a new class from weighted linear combinations of related class classifiers with the knowledge of co-occurrence statistics in images. Our approach can be seen as a new form transfer learning that leverages the interplay of both the category and attribute label spaces to generate new classifiers without seeing any labeled instances.

5 Conclusion

In this chapter, we have proposed and discussed two new methods to avoid the problem of "oversharing" in attribute learning.

First, we showed a method that exploits semantic relationships among attributes to guide attribute vocabulary learning by selectively sharing features among related attributes and encouraging disjoint supports for unrelated attributes. Our extensive experiments across three datasets validate two major claims for this method: (i) it overcomes misleading training data correlations to successfully learn semantic visual attributes, and (ii) preserving semantics in learned attributes is beneficial as an intermediate step in high-level tasks.

Next, we proposed a method to learn category-sensitive attributes rather than the standard monolithic attribute classifier over all categories. To do this, we developed a new form of transfer learning, in which analogous attributes are inferred using observed attributes organized according to two inter-related label spaces. Our tensor factorization approach solves the transfer problem, even when no training examples are available for the decision task of interest. Once again, our results confirm that our approach successfully addresses are the category-dependence of attributes and improves attribute recognition accuracy.

The work we have presented suggests a number of possible extensions. The decorrelating attributes approach of Section 2 may be extended to automatically mine attribute groups from web sources, or using distributed word representations etc. It may also be interesting to generalize the approach to settings where tasks cannot easily be clustered into discrete groups, but, say, pairwise semantic relationships among tasks are known. The analogous attributes approach would be interesting to consider in a one-shot or few-shot setting as well. While thus far we have tested it only in the case where no category-specific labeled examples are available for an attribute we wish to learn, it would be interesting to generalize the model to cases where some image instances are available. For example, such prior observations could be used to regularize the missing classifier parameter imputation step. In addition, we are interested in analyzing the impact of analogous attributes for learning relative properties.

Finally, a natural question is how the two "selective sharing" ideas presented in this chapter might be brought together. For instance, one might jointly train category-sensitive attribute classifiers with semantics-informed feature sharing between attributes, and then use the factorization method to infer classifiers for the category-attribute pairs for which we lack training examples. Our general idea of controlled sharing among tasks may also be applicable to many general multi-task learning problems that have additional sources of information on task relationships.

Acknowledgements We would like to thank Sung Ju Hwang for helpful discussions. This research was supported in part by NSF IIS-1065390 and ONR YIP N00014-12-1-0754.

References

1. Argyriou, A., Evgeniou, T., Pontil, M.: Multi-task feature learning. In: Conference on Neural Information Processing Systems (NIPS) (2007)
2. Aytar, Y., Zisserman, A.: Tabula rasa: Model transfer for object category detection. In: ICCV (2011)
3. Bach, F.: Consistency of the group lasso and multiple kernel learning. Journal of Machine Learning Research (JMLR) (2008)
4. Bart, E., Ullman, S.: Cross-Generalization: Learning Novel Classes from a Single Example by Feature Replacement. In: CVPR (2005)
5. Branson, S., Wah, C., Schroff, F., Babenko, B., Welinder, P., Perona, P., Belongie, S.: Visual recognition with humans in the loop. In: European Conference on Computer Vision (ECCV) (2010)
6. Chen, C.Y., Grauman, K.: Inferring analogous attributes. In: CVPR (2014)
7. Chen, X., Lin, Q., Kim, S., Carbonell, J.G., Xing, E.P.: Smoothing proximal gradient method for general structured sparse regression. Annals of Applied Statistics (AAS) (2012)
8. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: A Large-Scale Hierarchical Image Database. In: CVPR (2009)
9. Farhadi, A., Endres, I., Hoiem, D., Forsyth, D.: Describing Objects by Their Attributes. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2009)
10. Fei-Fei, L., Fergus, R., Perona, P.: A Bayesian approach to unsupervised one-shot learning of object categories. In: ICCV (2003)
11. Freeman, W.T., Tenenbaum, J.B.: Learning bilinear models for two-factor problems in vision. In: CVPR (1997)
12. Gardenfors, P.: Conceptual spaces as a framework for knowledge representation. In: Mind and Matter. The MIT Press
13. Hwang, S.J., Grauman, K., Sha, F.: Learning a Tree of Metrics with Disjoint Visual Features. In: Conference on Neural Information Processing Systems (NIPS) (2011)
14. Hwang, S.J., Grauman, K., Sha, F.: Analogy-preserving semantic embedding for visual object categorization. In: ICML (2013)
15. Hwang, S.J., Sha, F., Grauman, K.: Sharing features between objects and their attributes. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2011)
16. Jacob, L., Bach, F., Vert, J.: Clustered multi-task learning: a convex formulation. In: NIPS (2008)
17. Jayaraman, D., Sha, F., Grauman, K.: Decorrelating Semantic Visual Attributes by Resisting the Urge to Share. In: CVPR (2014)
18. Kang, Z., Grauman, K., Sha, F.: Learning with whom to share in multi-task feature learning. In: International Conference on Machine Learing (ICML) (2011)
19. Kim, S., Xing, E.: Tree-guided group lasso for multi-response regression with structured sparsity, with an application to eQTL mapping. Annals of Applied Statistics (AAS) (2012)
20. Kovashka, A., Parikh, D., Grauman, K.: WhittleSearch: Image Search with Relative Attribute Feedback. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2012)
21. Kumar, A., III, H.D.: Learning task grouping and overlap in multi-task learning. In: International Conference on Machine Learing (ICML) (2012)
22. Kumar, N., Belhumeur, P., Nayar, S.: Facetracer: A Search Engine for Large Collections of Images with Faces. In: European Conference on Computer Vision (ECCV) (2008)
23. Kumar, N., Berg, A., Belhumeur, P., Nayar, S.: Attribute and Simile Classifiers for Face Verification. In: ICCV (2009)
24. Lampert, C.: Semantic Attributes for Object Categorization (slides). http://ist.ac.at/ chl/talks/lampert-vrml2011b.pdf (2011)
25. Lampert, C., Nickisch, H., Harmeling, S.: Learning to detect unseen object classes by between-class attribute transfer. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2009)

26. Lampert, C., Nickisch, H., Harmeling, S.: Attribute-based classification for zero-shot visual object categorization. In: PAMI (2014)
27. Lim, J., Salakhutdinov, R., Torralba, A.: Transfer learning by borrowing examples for multiclass object detection. In: NIPS (2002)
28. Liu, J., Musialski, P., Wonka, P., Ye, J.: Tensor completion for estimating missing values in visual data. In: ICCV (2009)
29. Mahajan, D., Sellamanickam, S., Nair, V.: A joint learning framework for attribute models and object descriptions. In: International Conference on Computer Vision (ICCV) (2011)
30. Mensink, T.E.J., Gavves, E., Snoek, C.G.M.: Costa: Co-occurrence statistics for zero-shot classification. In: CVPR (2014)
31. Parameswaran, S., Weinberger, K.: Large margin multi-task metric learning. In: Conference on Neural Information Processing Systems (NIPS) (2010)
32. Parikh, D., Grauman, K.: Relative Attributes. In: ICCV (2011)
33. Patterson, G., Hays, J.: Sun attribute database: Discovering, annotating, and recognizing scene attributes. In: CVPR (2012)
34. Quattoni, A., Collins, M., Darrell, T.: Transfer learning for image classification with sparse prototype representations. In: CVPR (2008)
35. Romera-Paredes, B., Argyriou, A., Bianchi-Berthouze, N., Pontil, M.: Exploiting unrelated tasks in multi-task learning. In: Conference on Uncertainty in Artificial Intelligence (AISTATS) (2012)
36. Russakovsky, O., Fei-Fei, L.: Attribute learning in large-scale datasets. In: ECCV Workshop on Parts and Attributes (2010)
37. Saleh, B., Farhadi, A., Elgammal, A.: Object-Centric Anomaly Detection by Attribute-Based Reasoning. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2013)
38. Scheirer, W., Kumar, N., Belhumeur, P., Boult, T.: Multi-Attribute Spaces: Calibration for Attribute Fusion and Similarity Search. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2012)
39. Sharmanska, V., Quadrianto, N., Lampert, C.: Augmented attributes representations. In: ECCV (2012)
40. Shrivastava, A., Singh, S., Gupta, A.: Constrained semi-supervised learning using attributes and comparative attributes. In: European Conference on Computer Vision (ECCV) (2012)
41. Siddiquie, B., Feris, R., Davis, L.: Image Ranking and Retrieval Based on Multi-Attribute Queries. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2011)
42. Song, F., Tan, X., Chen, S.: Exploiting relationship between attributes for improved face verification (2011)
43. Tibshirani, R.: Regression shrinkage and selection via the lasso. In: RSS Series B (1996)
44. Tommasi, T., Orabona, F., Caputo, B.: Safety in numbers: learning categories from few examples with multi model knowledge transfer. In: CVPR (2010)
45. Torralba, A., Murphy, K., Freeman, W.: Sharing visual features for multiclass and multiview object detection. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) (2007)
46. Vasilescu, M.A.O., Terzopoulos, D.: Multilinear analysis of image ensembles: Tensorfaces. In: ECCV (2002)
47. Vlasic, D., Brand, M., Pfister, H., Popović, J.: Face transfer with multilinear models. ACM Trans Graphics **24**(3), 426–433 (2005)
48. Wah, C., Branson, S., Welinder, P., Perona, P., Belongie, S.: The Caltech-UCSD Birds-200-2011 Dataset (2011)
49. Wang, G., Forsyth, D.: Joint learning of visual attributes, object classes and visual saliency. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2009)
50. Wang, G., Forsyth, D., Hoiem, D.: Comparative object similarity for improved recognition with few or no examples. In: CVPR (2010)
51. Wang, Y., Mori, G.: A discriminative latent model of object classes and attributes. In: European Conference on Computer Vision (ECCV) (2010)
52. Xiong, L., Chen, X., Huang, T., Schneider, J., Carbonell, J.: Temporal collaborative filtering with Bayesian probabilistic tensor factorization. In: SDM (2010)

53. Yang, J., Yan, R., Hauptmann, A.: Cross-domain video concept detection using adaptive svms. In: ACM Multimedia (2007)
54. Yu, F., Cao, L., Feris, R., Smith, J., Chang, S.F.: Designing category-level attributes for discriminative visual recognition. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2013)
55. Yuan, M., Lin, Y.: Model selection and estimation in regression with grouped variables. In: RSS Series B (2006)
56. Zhou, D., Xiao, L., Wu, M.: Hierarchical Classification via Orthogonal Transfer. In: International Conference on Machine Learing (ICML) (2011)
57. Zhou, Y., Jin, R., Hoi, S.: Exclusive lasso for multi-task feature selection. In: Conference on Uncertainty in Artificial Intelligence (AISTATS) (2010)