Decorrelating Semantic Visual Attributes by Resisting the Urge to Share

Dinesh Jayaraman
UT Austin
dineshj@cs.utexas.edu

Fei Sha
USC
feisha@usc.edu

Kristen Grauman
UT Austin
grauman@cs.utexas.edu

Abstract

Existing methods to learn visual attributes are prone to learning the wrong thing—namely, properties that are correlated with the attribute of interest among training samples. Yet, many proposed applications of attributes rely on being able to learn the correct semantic concept corresponding to each attribute. We propose to resolve such confusions by jointly learning decorrelated, discriminative attribute models. Leveraging side information about semantic relatedness, we develop a multi-task learning approach that uses structured sparsity to encourage feature competition among unrelated attributes and feature sharing among related attributes. On three challenging datasets, we show that accounting for structure in the visual attribute space is key to learning attribute models that preserve semantics, yielding improved generalizability that helps in the recognition and discovery of unseen object categories.

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". Recent research demonstrates that attributes provide a useful bridge between low-level image features and high-level entities like object or scene categories [5, 14, 17]. Methods for attribute learning typically follow the standard discriminative learning pipeline that has been successful in other visual recognition problems. 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 [14, 5, 17, 3, 22].

The problem is that this standard approach is prone to learning image properties that are *correlated* with the attribute of interest, rather than the attribute itself. Fig 1 helps illustrate why. Suppose you are tasked with learning the attribute present in the first three images, but absent in the others. Even if you restrict yourself to "nameable" properties, there are many plausible hypotheses for the attribute: brown? furry? has-ears? land-dwelling?

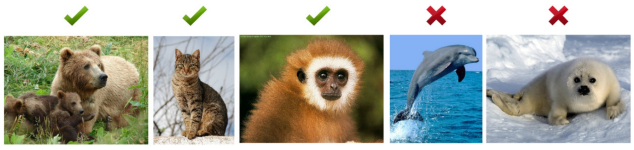


Fig 1: What attribute is present in the first three images, but not the last two? Standard methods attempting to learn "furry" from such images are prone to learn "brown" instead—or some combination of correlated properties. We propose a multi-task attribute learning approach that resists the urge to share features between attributes that are semantically distinct yet often co-occur.

A key underlying challenge is that the hypothesis space for attribute learning is very large. A 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 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 Fig 1.

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 [14, 12, 22], zero-shot learning [17], and textual description generation [5]—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 is to decorrelate attributes at the time of learning. 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. 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 properties. In the example in Fig 1, dimensions corresponding to color histogram bins might be used to detect "brown", whereas those corresponding to texture in the center of the image might be reserved to detect "furry". Moreover, since some attributes naturally *should* share features, we leverage side information about the attributes' semantic relatedness to encourage feature sharing among closely related properties (e.g., reflecting that "red" and "brown" are likely to share).

Our method takes as input images labeled according to the presence/absence of each attribute, as well as a set of attribute "groups" reflecting those that are mutually semantically related. As output, it produces one binary classifier for each attribute. Attributes in the same group are encouraged to share low-level feature dimensions, while unrelated attributes compete for them. We formulate these preferences using structured sparsity regularization on a multi-task classification learning objective for principled feature selection.

We show that our approach helps disambiguate attributes and thus preserves semantics better—through standard tests such as attribute localization and zero-shot category recognition, as well as through a new application of semantic visual attributes for category discovery. Our results on three datasets consistently show that the proposed approach helps "learn the right thing."

2. Related Work

Attributes as semantic features A visual attribute is a binary predicate for an image that indicates whether or not a property is present [14, 5, 17]. 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") [14, 12, 22]; 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.) [17, 3, 23]; deviations from an expected configuration of attributes may be used to generate textual descriptions of what humans would find remarkable [5, 21]. In all such applications, inadvertently learning correlated visual properties 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.

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") [30, 25, 17, 24]. Our goal 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.

Differentiating attributes To our knowledge, the only previous work that attempts to explicitly decorrelate semantic attributes is [5]. For each attribute, their 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. However, this method is susceptible to learning chance correlations among the reduced number of samples of individual classes and moreover requires expensive instance-wise attribute annotations. Our approach overcomes these issues, as we demonstrate with extensive comparisons to [5] in results.

While this is the only prior work on decorrelating *semantic* attributes, some unsupervised approaches attempt to diversify discovered (un-named/non-semantic) "attributes" [31, 18, 5]—for example by designing object class splits that yield uncorrelated features [31] or converting redundant semantic attributes into discriminative ones [18]. In contrast, we jointly learn a specified vocabulary of *semantic* attributes.

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, 11]; e.g., fea-

ture sharing between objects can yield faster detectors [27], and sharing between objects and their attributes can isolate features suitable for both tasks [29, 8]. A few works have begun to explore the value of modeling *negative* correlations [33, 15, 7, 20]. 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 [15, 7]. These methods exploit the inherent mutual exclusivity among object labels, which does not hold in our attributes setting. Unlike any of these approaches, 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 [9, 10, 13]. Our method involves grouped tasks, but with two crucial differences: (1) we explicitly model between-group *competition* along with in-group sharing to achieve inter-group decorrelation, and (2) we treat external knowledge about semantic groups as supervision to be exploited during learning. In contrast, the prior methods [9, 10, 13] discover task groups from data, which is prone to suffer from correlations in the same way as a single-task learner.

3. Approach

Our goal is to learn attribute classifiers that fire only when the correct semantic property is present. In particular, we want them to generalize to test images where the attribute co-occurrence patterns may differ from what is observed in training. The key to our approach is to jointly learn all attributes in a vocabulary, while enforcing a structured sparsity prior that aligns feature sharing patterns with semantically close attributes and feature competition with semantically distant ones.

In the following, we first describe the inputs to our algorithm: the semantic relationships among attributes (Sec. 3.1) and the low-level image descriptors (Sec. 3.2). Then we introduce our learning objective and optimization framework (Sec. 3.3), which outputs a classifier for each attribute in the vocabulary.

3.1. Semantic Attribute Groups

Suppose we are learning attribute classifiers[1] 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.

[1]We use "attribute", "classifier" and "task" interchangeably.

If two attributes are in the same group, this reflects that they have some semantic tie. For instance, in Fig 2, 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 [17, 28] in our experiments.

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. We do not claim there exists 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" [32]. 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*.

3.2. 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.

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 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 Sec. 4.

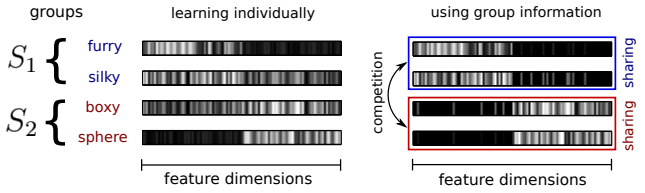


Fig 2: Sketch of our idea. We show weight vectors (absolute value) for attributes learnt by standard (left) and proposed (right) approaches. The higher the weight (lighter colors) assigned to a feature dimension, the more the attribute relies on that feature. In this instance, our approach would help resolve "silky" and "boxy", which are highly correlated in training data and consequently conflated by standard learning approaches.

3.3. Joint Attribute Learning with Feature Sharing and Competition

The input to our learning scheme is (1) the descriptors for N training images, each represented as a D-dimensional vector $\mathbf{x}_n$, (2) the corresponding (binary) attribute labels for all attributes, which are indexed by $a = 1, \ldots, M$, and (3) 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 rows $\mathbf{y}_n$ and columns $\mathbf{y}^m$.

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 (1) in-group feature sharing (2) between-group competition for features. We encode this as a structured sparsity

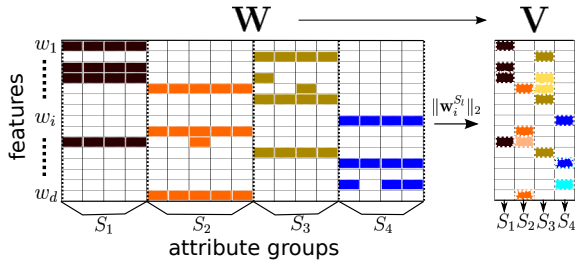


Fig 3: "Collapsing" of grouped columns of the feature selection matrix $\mathbf{W}$ prior to applying the lasso penalty $\sum_l \|\mathbf{v}^l\|_1$. Non-zero entries in W and V are shaded. Darkness of shading in $\mathbf{V}$ represents how many attributes in that group selected that feature.

problem, where structure in the output attribute space is represented by the grouping. Fig 2 illustrates the envisioned effect of our approach.

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 Sec. 4. The first is a simple adaptation of the single-task lasso method [26]. 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 [26] to the multi-task setting to yield an "all-competing" lasso minimization objective:

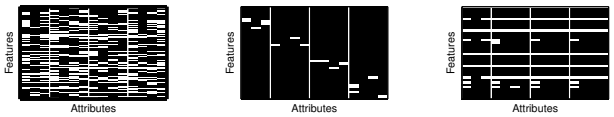
$$\mathbf{W}^* = \arg\min_{\mathbf{W}} L(\mathbf{X}, \mathbf{Y}; \mathbf{W}) + \lambda \sum_m \|\mathbf{w}^m\|_1, \quad (2)$$

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 (1) encouraging sparsity on each task column $\mathbf{w}^m$, and (2) 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:

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

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



Lasso (Eq 2) Proposed (Eq 4) All-sharing (Eq 3)

Fig 4: A part of the $\mathbf{W}$ matrix (thresholded, absolute value) learned by the different structured sparsity approaches on CUB data. The thin white vertical lines separate attribute groups.

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 Eqs 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:

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

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. Fig 3 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.

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, Eq 4 simplifies to Eq 3. Similarly, setting each attribute to belong to its own singleton group $S_m = \{m\}$ produces the lasso formulation of Eq 2. Fig 4 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 decorrelate 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.

	Categories		Attributes		Features	
Datasets	seen	unseen	num (M)	groups (L)	# win	D
CUB	100	100	312	28	15	375
AwA	40	10	85	9	1,21	290
aPY-25	20	12	25	3	7	105

Table 5: Summary of dataset statistics

Optimization Mixed norm regularizations of the form of Eq 4, while convex, are non-smooth and non-trivial to optimize. Such norms appear frequently in the structured learning literature [32, 2, 1, 11]. As in [11], we reformulate the objective by representing the 2-norm in the regularizer in its dual form, before applying the smoothing proximal gradient descent [4] method to optimize a smooth approximation of the resulting objective. See supp.

4. Experiments and results

Datasets We use three datasets with 422 total attributes: (1) the CUB-200-2011 Birds ("CUB") [28], (2) Animals with Attributes ("AwA") [17] (3) aPascal/aYahoo ("aPY") [5]. Dataset statistics are summarized in Table 5. 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. 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".

Features Sec. 3.2 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 [5]. On aPY, we use a one-level pyramid with $3\times2+1=7$ windows on four channels, following [5]. 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"). See supp. for details.

Semantic groups To define the semantic groups, we rely largely on existing data. CUB specifies 28 attribute

Tasks	Attribute detection scores (mean average precision)								Zero-shot DAP acc.(%)		
Datasets	CUB			AwA		aPY-25			CUB	AwA	aPY-25
Methods	U	H	S	U	S	U	H	S	[100 cl]	[10 cl]	[12 cl]
lasso	0.1783	0.2552	0.2219	0.5274	0.6175	0.2713	0.2925	0.3184	7.345	25.32	9.88
all-sharing [1]	0.1778	0.2546	0.2217	0.5378	0.6021	0.2601	0.2934	0.2560	7.339	19.40	6.95
classwise [5]	0.1909	0.2756	0.2406	N/A	N/A	0.2729	0.2776	0.3595	9.149	N/A	20.00
standard	0.1836	0.2706	0.2369	0.5366	**0.6687**	0.2727	0.2845	**0.3772**	9.665	26.29	**20.09**
proposed	**0.2114**	**0.2962**	**0.2654**	**0.5497**	0.6480	**0.2989**	**0.3318**	0.3021	**10.696**	**30.64**	19.43

Table 6: Scores on attribute detection (left, AP) and zero-shot object recognition (right, accuracy). Higher is better. U, H and S refer respectively to *unseen*, *hard-seen* and *all-seen* test sets (Sec. 4.1). Our approach generally outperforms existing methods, and especially shines when attribute correlations differ between train and test data (i.e., the U, H, and zero-shot (Sec. 4.2) scenarios).

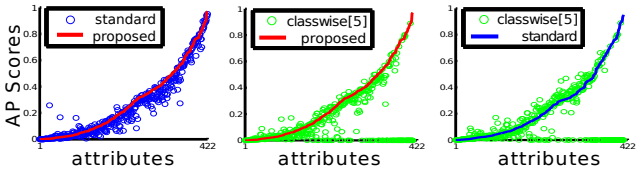


Fig 7: Attribute detection results across all datasets (Sec 4.1)

groups [28] (head color, back pattern *etc*.). For AwA, the authors suggest 9 groups in [16] (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 [16] ("aPY-25"). See supp. for full groupings.

As discussed in Sec 3.2, 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.

Baselines We compare to four methods throughout. Two are single-task learning baselines, in which each attribute is learned separately: (1) "standard": ℓ_2-regularized logistic regression, and (2) "classwise": the object class-label based feature selection scheme proposed in [5] described in Sec. 2 (with logistic regression in the final stage replacing the SVM, for uniformity). The other two are the sparse multi-task methods in Sec. 3: (3) "lasso" (Eq 2), and (4) "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 [5]) are validated with held out unseen class data.

4.1. 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: (1) *unseen*: unseen class instances (2) *all-seen*: other instances of seen classes and (3) *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).

Overall results Table 6 (left) shows the mean AP scores over all attributes, per dataset.[2] On all three datasets, our method generalizes significantly better than all baselines to unseen classes and hard seen data.

While the "classwise" technique of [5] 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 correlated 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 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 [5], and standard. For each plot, attributes are ar-

[2] AwA has only class-level attribute annotations, so (i) the classwise baseline [5] is not applicable and (ii) the "hard-seen" test set is not defined.

Fig 8: Success cases: Annotations shown are our method's attribute predictions, which match ground truth. The logistic regression baseline ("standard") fails on all these cases.

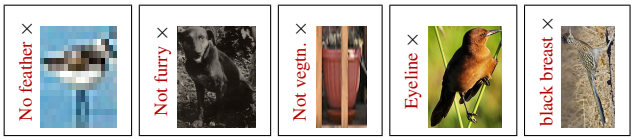


Fig 9: Failure cases: Cases where our predictions (shown) are incorrect and the "standard" baseline succeeds.

ranged in order of increasing detectability for one method.[3] 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).

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 Fig 8, 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 1, end) is correctly marked not "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.

Fig 9 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 (see supp). Fig 10 shows how 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.

[3]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 10: 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.

4.2. Zero-shot Object Recognition

Next we show the impact of retaining attribute semantics for zero-shot object recognition. Closely following the setting in [17], the goal is to learn object categories from textual descriptions, but no training images (*e.g.*, "zebras are striped and four-legged"), making attribute correctness crucial. We input attribute probabilities from each method's models to the Direct Attribute Prediction (DAP) framework for zero-shot learning [17] (see supp for details). Table 6 (right) 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 (Sec 4.1). 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).

4.3. Category Discovery with Semantic Attributes

Finally, we demonstrate the impact on category discovery. Cognitive scientists propose that natural categories are

Methods / Datasets	CUB-s	AwA	aPY-25	CUB-f
lasso	0.5485	0.1891	0.1915	0.3503
all-sharing [1]	0.5482	0.1881	0.1717	0.3508
classwise [5]	0.5746	N/A	0.1973	0.3862
standard	0.5697	0.2239	0.1761	0.3719
proposed	**0.5944**	**0.2411**	**0.2476**	**0.4281**
GT annotations	0.6489	1.0000	0.6429	0.4937

Table 11: NMI scores for discovery of unseen categories (Sec 4.3). Higher is better.

convex regions in *conceptual spaces* whose axes correspond to "psychological quality dimensions" [6]. 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 11 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 (1) visual attributes indeed constitute a plausible "conceptual space" for discovery and (2) improved attribute learning models could yield large gains for high-level visual tasks.

5. Conclusions

We introduced a method for using semantics to guide attribute learning. Our extensive experiments across three datasets support our two major claims: (1) our approach overcomes misleading training data correlations to successfully learn semantic visual attributes, and (2) preserving semantics in learned attributes is beneficial as an intermediate step in high-level tasks. In future work, we plan to investigate the effect of overlapping attribute groups and explore methods to automatically mine semantic information.

Acknowledgements: We would like to thank Sung Ju Hwang for helpful discussions. This research is supported in part by NSF IIS-1065390 and NSF IIS-1065243.

References

[1] A. Argyriou, T. Evgeniou, and M. Pontil. Multi-Task Feature Learning. In *NIPS*, 2007.

[2] F. Bach. Consistency of the group lasso and multiple kernel learning. In *JMLR*, 2008.

[3] S. Branson, C. Wah, F. Schroff, B. Babenko, P. Welinder, P. Perona, and S. Belongie. Visual recognition with humans in the loop. In *ECCV*, 2010.

[4] X. Chen, Q. Lin, S. Kim, J. G. Carbonell, and E. Xing. Smoothing proximal gradient method for general structured sparse regression. In *AAS*, 2012.

[5] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth. Describing Objects by Their Attributes. In *CVPR*, 2009.

[6] P. Gardenfors. Conceptual spaces as a framework for knowledge representation. In *Mind and Matter*, 2004.

[7] S. J. Hwang, K. Grauman, and F. Sha. Learning a Tree of Metrics with Disjoint Visual Features. In *NIPS*, 2011.

[8] S. J. Hwang, F. Sha, and K. Grauman. Sharing features between objects and their attributes. In *CVPR*, 2011.

[9] L. Jacob, F. Bach, and J.P. Vert. Clustered Multi-Task Learning: A Convex Formulation. In *NIPS*, 2008.

[10] Z. Kang, K. Grauman, and F. Sha. Learning with whom to share in multi-task feature learning. In *ICML*.

[11] S. Kim and E. Xing. Tree-guided group lasso for multi-response regression with structured sparsity, with an application to eQTL mapping. In *AAS*, 2012.

[12] A. Kovashka, D. Parikh, and K. Grauman. WhittleSearch: Image Search with Relative Attribute Feedback. In *CVPR*, 2012.

[13] A. Kumar and H. Daume III. Learning task grouping and overlap in multi-task learning. In *ICML*, 2012.

[14] N. Kumar, P. N. Belhumeur, and S. K. Nayar. Facetracer: A Search Engine for Large Collections of Images with Faces. In *ECCV*, 2008.

[15] L. Xiao and D. Zhou and M. Wu. Hierarchical Classification via Orthogonal Transfer. In *ICML*, 2011.

[16] C. Lampert. Semantic Attributes for Object Categorization (slides). http://ist.ac.at/ chl/talks/lampert-vrml2011b.pdf, 2011.

[17] C. Lampert, H. Nickisch, and S. Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *CVPR*, 2009.

[18] D. Mahajan, S. Sellamanickam, and V. Nair. A joint learning framework for attribute models and object descriptions. In *ICCV*, 2011.

[19] S. Parameswaran and K. Weinberger. Large margin multi-task metric learning. In *NIPS*, 2010.

[20] B. Romera-Paredes, A. Argyriou, N. Bianchi-Berthouze, and M. Pontil. Exploiting unrelated tasks in multi-task learning. In *AISTATS*, 2012.

[21] B. Saleh, A. Farhadi, and A. Elgammal. Object-Centric Anomaly Detection by Attribute-Based Reasoning. In *CVPR*, 2013.

[22] W. Scheirer, N. Kumar, P.N. Belhumeur, and T.E. Boult. Multi-Attribute Spaces: Calibration for Attribute Fusion and Similarity Search. In *CVPR*, 2012.

[23] A. Shrivastava, S. Singh, and A. Gupta. Constrained semi-supervised learning using attributes and comparative attributes. In *ECCV*, 2012.

[24] B. Siddiquie, R. Feris, and L. Davis. Image Ranking and Retrieval Based on Multi-Attribute Queries. In *CVPR*, 2011.

[25] F. Song, X. Tan, and S. Chen. Exploiting relationship between attributes for improved face verification. In *BMVC*, 2011.

[26] R. Tibshirani. Regression shrinkage and selection via the lasso. In *RSS Series B*, 1996.

[27] A. Torralba, K. Murphy, and W. Freeman. Sharing visual features for multiclass and multiview object detection. In *PAMI*, 2007.

[28] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. 2011.

[29] G. Wang and D. Forsyth. Joint learning of visual attributes, object classes and visual saliency. In *CVPR*, 2009.

[30] Y. Wang and G. Mori. A discriminative latent model of object classes and attributes. In *ECCV*, 2010.

[31] F. Yu, L. Cao, R. Feris, J. Smith, and S. Chang. Designing category-level attributes for discriminative visual recognition. In *CVPR*, 2013.

[32] M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. In *RSS Series B*, 2006.

[33] Y. Zhou, R. Jin, and S.C.H. Hoi. Exclusive lasso for multi-task feature selection. In *AISTATS*, 2010.