Learning Shared Body Plans

Ian Endres, Vivek Srikumar, Ming-Wei Chang, Derek Hoiem
University of Illinois at Urbana-Champaign
{iendres2,vsrikum2,mchang21,dhoiem}@uiuc.edu

Abstract

We cast the problem of recognizing related categories as a unified learning and structured prediction problem with shared body plans. When provided with detailed annotations of objects and their parts, these body plans model objects in terms of shared parts and layouts, simultaneously capturing a variety of categories in varied poses. We can use these body plans to jointly train many detectors in a shared framework with structured learning, leading to significant gains for each supervised task. Using our model, we can provide detailed predictions of objects and their parts for both familiar and unfamiliar categories.

1. Introduction

Many important applications require visual systems to make sensible predictions about *every* object that they encounter. An automated vehicle must respond appropriately whenever an object crosses its path, whether that object is a cement block, a cow, or a child on a tricycle. The vehicle needs to localize the object and predict its movement. When confronted with a cement block, the vehicle should not wait for the block to move, but when facing a child on a tricycle, the vehicle should brake or give the tricycle wide berth, moving behind it. When viewed from the lens of basic categorization, the problem seems insurmountable — there are thousands of potential categories, and it is difficult to identify the relevant ones in advance. Instead, we believe it important to explicitly model objects in terms of parts and multiple levels of categorization, so that novel objects can sometimes be related to known ones. Some knowledge of materials, shapes, animals, and wheeled vehicles should lead to reasonable behavior for the scenarios described above, even if the designers did not build in detectors for "cement block", "cow", or "tricycle". Our main challenge is, how can we represent multiple related categories in a way that leads to more efficient learning, more accurate recognition, and generalization to novel objects?

In this paper, we propose to model objects in terms of shared parts and layouts, so that learning of several related categories can be treated as a single, unified recognition problem. Our representation is organized as a mixture of *body plans*, shown in Figures 1 and 2, that predict the categories and spatial arrangement of parts. Left-facing, standing dogs, cats, horses and cows, all have the same visible parts in roughly the same configuration, and they can be modeled with the same body plan. One body plan can be shared by many categories, and a single category may be represented by several plans that correspond to different viewpoints or poses. Likewise, we model each part's appearance with a mixture of models that is shared across categories, encoding, for example, that frontal views of horse heads and cow heads have similar appearance. Our approach is to learn these models from bounding boxes of objects and their parts. The part annotations provide explicit correspondence within and across categories, allowing us to construct a shared representation with more flexible layout models and detailed prediction. Because parts may be difficult to detect in isolation, we use structured learning to jointly model the appearance of parts and categories and body plans, and use structured prediction for inference.

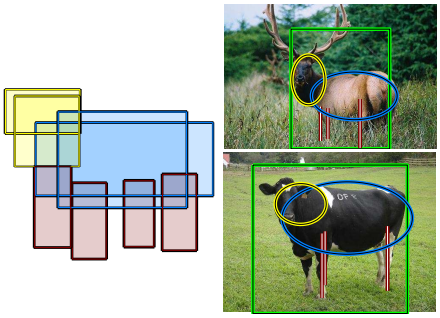


Figure 1. Our shared body plans enable joint training of detailed structured representations of related categories. In the results above, the body plan (**left**) is able to detect and localize the parts of both the familiar elk and the unfamiliar cow (**right**).

1.1 Background and Contributions

In this paper, we propose: 1) a representation of related object categories in terms of shared appearance and part layout; 2) a structured learning method to jointly train the parameters of these models; and 3) an efficient inference procedure to jointly localize objects and their parts. When provided with detailed annotation (bounding boxes of entire objects and their parts), we show how to learn structured representations of objects that are shared across categories and discriminatively trained to localize objects and

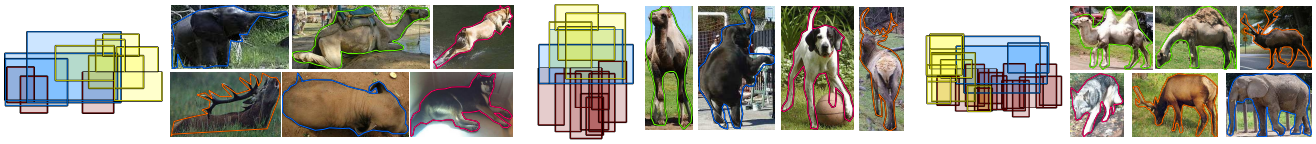


Figure 2. **Body plans:** Shown are three of the nine body plans used to represent four-legged animals. Each body plan represents a cluster of coherent poses from one or many categories. (**left**) Each box represents an anchor point which encodes an expected position and scale of a particular object or part detector. (**right**) Examples of a wide range of animals and poses that are captured by each body plan.

their parts. We also show how our structured representation can learn from objects that have only bounding box annotations, so that the computer can better recognize them and find their parts.

We show that by jointly learning appearance and layout of both parts and categories from fully supervised examples we can better recognize objects than if training from only whole-object bounding boxes. Intuitively, part annotations should help because they provide a more detailed correspondence that reveals the internal object structure. Yet, as Felzenszwalb et al. [9] note, it has been difficult to show that part-based models [2, 3, 6, 11, 12, 14, 16, 15] can outperform simpler rigid template [19, 20, 24, 4] or bag of feature [27, 23] models. The deformable parts model of Felzenszwalb et al. [9] succeeds by jointly learning appearance parameters of latent parts. The parts are entirely in the service of the category detection; they are trained to aid object detection, not to be individually detectable. Latent parts are attractive for their discriminative potential and minimal annotation requirements, and many in the recognition community are wary of explicitly annotated parts.

However, several recent works show that additional supervision can improve detection accuracy. For example, Farhadi et al. [7] use part and attribute labels to improve superordinate category detection and Bourdev and Malik [1] use labeled joint positions to improve human detection. However, both procedures add spatial models on top of pretrained appearance models, which we show in this work can hinder the efficacy of the additional annotation. Instead, we treat the spatial model as an integral component of our learning procedure, allowing detectors to learn to rely on each other, giving greater gains. Further, our model aims to improve a number of related tasks simultaneously, rather than pooling many supervised detectors for a single task.

Recently, Sun and Savarese [21] train a fully supervised part based model to improve detection of individual categories while localizing their parts. In contrast, our model attacks the more general problem of not only recognizing individual categories but also jointly representing several related categories, requiring that our model address the larger variation in part size, appearance, and spatial configuration. Further, our model can encode missing parts and multiple occurrences of a single part type, such as legs, while parts in their model are distinct and must always be detected.

In addition to Farhadi et al. [7], there have been several other works that train shared representations across categories [22, 17, 18] for detecting objects. However, each of these works relies on latent parts to build the shared representation and are unable to localize named parts. One recent work of this nature from Ott and Everingham [18] is complementary to ours, as it extends the deformable part model to share latent (rather than supervised) parts across multiple categories. In fact, with the constraint construction and latent structure parameterization of our model, latent parts of their form can be directly incorporated into our model.

2. Multicategory Object Representations

We aim to improve prediction accuracy and learning efficiency by sharing representations among related categories. We can take advantage of explicitly labeled parts and broad categories to better correspond objects within and across categories. This additional supervision enables more flexible layout models that handle deformation in scale, multiple parts, and occlusion, and it also facilitates sharing of detector and spatial layout parameters. Our multi-category representation is motivated by the following intuitions:

1. Objects from related categories, viewed in similar poses, have similar overall shapes, appearance of parts, and layout of parts. We model objects with broad category detectors, part detectors, and body plans, whose parameters are shared across basic categories.
2. Objects from the same category vary dramatically in appearance and layout, due to change in pose and viewpoint. We employ mixture models for part and category appearance and train a mixture of body plans.
3. For different instances or categories, the same part may vary in relative size and position. We learn *anchor points* that set the expected positions and sizes of a part and allow detections to vary in both position and size with some deformation cost.
4. One object may have multiple copies of the same part, though some may be occluded. We define multiple anchor points for each named part, so that one part can be detected multiple times; in structured prediction, we infer which were actually detected, potentially encoding that several or no instances of a part were observed.
5. A particular detector should work well in the context of other detectors, not necessarily in isolation. We employ structured learning and prediction to learn parameters that work well together.

2.1 Body Plans

The body plans are viewpoint-dependent models of the spatial layout of parts, and they are shared across basic categories. One body plan may correspond to four-legged ani-

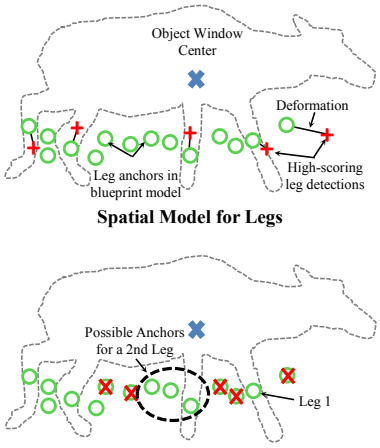


Figure 3. **Spatial Model:** Each body plan uses a number of anchor points to represent possible locations of each detector type, such as legs. (**Top**) For each leg anchor point, we search for a detection window that maximizes the tradeoff between appearance score and deformation cost. (**Bottom**) We iteratively choose the anchor with highest score (Leg 1), and apply the exclusion constraints to narrow the search for the next possible anchor. We repeat until we reach the maximum number of allowed detections.

mals that are standing and facing right, while another might correspond to flying birds, and another to non-objects.

The body plan regularizes the set of part/category detections in three ways. First, the plan provides a set of anchor positions where the detections are likely to occur (detections are penalized for drifting from these anchor positions), see Figure 3. Second, the plan encodes pairwise constraints between anchor points. For example, an object cannot be both a "horse" and a "dog", an animal cannot have two "head" detections, and an "eye" cannot be detected on distant anchor points. Third, the plan provides a prior (through a bias term) on the likelihood of observing particular categories and parts. When encoding the appearance of parts with a mixture of viewpoint-dependent models (as in our model), these bias terms also provide a prior on viewpoint.

2.2 Details of Parameterization

We represent a range of object categories (e.g., all four-legged animals) with shared body plans and appearance-based detectors. The object model consists of a set of detectors of type t with appearance parameters $\mathbf{w}_t^A$ and a set of body plans. One body plan b is parameterized by: an object root position and scale; a set of anchor points ($\{\hat{\mathbf{l}}_{bi}\}$) defining the expected position and scale of a detector of type t_{bi} relative to the root; weights for deformation in position and scale $\mathbf{w}_b^D$; a bias w_{bt}^B for each detector; constraints between anchor points $\mathcal{H}_b$; and a bias term w_{b0}^B for the body plan.

An instance of an object hypothesis of body plan b is defined by the structure $\mathbf{h}$, where each element $h_i = (\delta_i, \mathbf{l}_i)$ defines the state of anchor point i. The indicator δ_i is 1 if the detection for anchor point i is visible and 0 otherwise. $\mathbf{l}_i$ gives the position of this detection in location and scale.

Objective Function. At each object root position and scale $\mathbf{l}_0 = (x, y, s)$, we search for the highest scoring body plan b and structure $\mathbf{h}$:

$$\{b^*, \mathbf{h}^*\} = \underset{b, \mathbf{h} \in \mathcal{H}_b}{\operatorname{argmax}} f_b(\mathbf{h}; \mathbf{x}, \mathbf{l}_0, \mathbf{w})$$

$$f_b(\mathbf{h}; \mathbf{x}, \mathbf{l}_0, \mathbf{w}) = w_{b0}^B + \sum_{i=1}^{N_b} \delta_i \cdot S_b(\mathbf{l}_i; \mathbf{x}, \mathbf{l}_0, \mathbf{w}) \quad (1)$$

$$S_b(\mathbf{l}_i; \mathbf{x}, \mathbf{l}_0, \mathbf{w}) = w_{bt_i}^B + S_A(\mathbf{l}_i; \mathbf{x}, \mathbf{w}_{t_i}^A) - S_D(\mathbf{l}_i; \hat{\mathbf{l}}_{bi}, \mathbf{l}_0, \mathbf{w}_{bt_i}^D)$$

For each body plan b and i^{th} anchor point $\hat{\mathbf{l}}_{bi}$, we find the detection position $\mathbf{l}_i$ that maximizes the overall body plan score S_b. This score is composed of the appearance score S_A, a deformation penalty S_D, and a per-type bias $w_{bt_i}^B$. If the overall score is greater than zero, then $\delta_i = 1$, subject to the constraints given by set $\mathcal{H}_b$; otherwise, $\delta_i = 0$. The most likely object structure $\mathbf{h}^*$, therefore, is composed of a set of detections that are consistent in appearance and joint configuration with an object in the given domain at position $\mathbf{l}_0$, and its score is the sum of the individual detection scores. Our inference procedure is further explained in Section 5.

Appearance Models. The appearance score S_A uses the HOG-based, deformable latent part models of Felzenszwalb et al. [9] for categories and parts. Each detector models an object or part in terms of a whole-window appearance template (grid of oriented gradient histograms) and a set of latent part templates, anchor points, and deformation costs. Modal variations within category are partially captured by modeling the appearance as a mixture of components, leading to a set of detectors for a given object part or category label. We treat each component as a different detector type, allowing body plans to select which aspect of a detector to use. Given the locations of the latent parts, the appearance model for detector type t can be written as linear classifier $\mathbf{w}_t^A$ over the structured features ϕ_t. Note that, though our appearance models are parameterized as in [9], ours are shared across categories and jointly trained using structured learning.

Deformation Costs. Like [9], deformation costs are a linear combination of linear and quadratic penalties for deforming from the expected position. However, we also allow deformations in scale in addition to position.

$$S_D(\mathbf{l}_i; \hat{\mathbf{l}}_{bi}, \mathbf{l}_o, \mathbf{w}_{bt_i}^D) = \mathbf{w}_{bt_i}^D \cdot \psi_t(\mathbf{l}_i, \hat{\mathbf{l}}_{bi} + \mathbf{l}_0)$$

$$\psi_t(\mathbf{l}, \hat{\mathbf{l}}) = \left(dx, dx^2, dy, dy^2, ds, ds^2\right)^T \quad (2)$$

$$dx = \frac{(l_x - \hat{l}_x)}{2^{l_s}},\ dy = \frac{(l_y - \hat{l}_y)}{2^{l_s}},\ ds = l_s - \hat{l}_s$$

Deformations in position are normalized by the scale of the detection to maintain consistent score across scales.

.

Constraints. Our model prevents unlikely detections with count constraints and exclusion constraints. Count constraints can be used to prevent an animal from having two

heads or more than four legs. Exclusion constraints avoid unusual combinations of anchor points, such as all four legs being detected on the same side of the animal. See Figure 3 for an example. In the form of a linear binary program, they can be written as follows:

$$\delta_i + \delta_j \leq 1, \qquad \forall (i,j) \in S_e \qquad \text{(Exclusion)}$$
$$\sum_{i \in S_c} \delta_i \leq \tau_c, \qquad \forall c \qquad \text{(Count)}.$$

The set S_e over pairs of examples defines the exclusion constraints, where only one element in each pair can be active. Each count constraint is defined by the set S_c, indicating the elements constrained, and τ_c, the maximum number of anchor points in S_c that can be active.

3. Initialization

Before training the model, we must first initialize the spatial model by partitioning the data into separate body plans and finding appropriate anchor points.

Body Plan Selection. Since we have detailed annotations for each object and its parts, we can establish direct correspondences between examples and cluster them into body-plans with coherent layouts. These body plans help reduce the variation in appearance and simplify their spatial models. To cluster the examples, we compute distance d_{ij} between each pair of objects and use kernel K-means [13] to generate K body plan clusters.

The distance d_{ij} measures how many parts the objects share and the agreement between their spatial configuration. First, the objects are rectified so their bounding boxes have unit diagonal and are centered at $(0, 0)$. Then, for each part type, we compute the distance between the centers of the parts. If there are multiple parts of a type, such as legs, we compute the matching with minimal average distance. Parts that do not have a match or have a matched distance greater than 25% the size of the object are given a penalty of 1. Examples of the resulting clusters can be found in Figure 2.

Keypoint Selection. For each body plan, we now need to select a small set of anchor points for each detector type. These anchor points should be chosen to minimize the expected distance to each example, allowing tightly tuned spatial models while remaining flexible enough to explain every example.

We use a bounding box clustering procedure similar to [7]. Each example is again scaled to a unit diagonal and centered at $(0, 0)$. For windows of each type, we incrementally build a set of clusters S to ensure that each example is sufficiently covered. At each iteration, we randomly select a box and check its overlap with elements already in S. If the overlap is below 40%, it is not well covered and is added to S. Otherwise, we add it to the existing cluster $s \in S$ that has the best overlap. Finally, we remove any cluster in S that has few examples assigned to it. After multiple independent trials of this clustering procedure, we choose the set S that minimizes the average deformation cost S_D (eq. 2) of the model. We add an exclusion constraint if a pair of anchors of a type are never active within the same example.

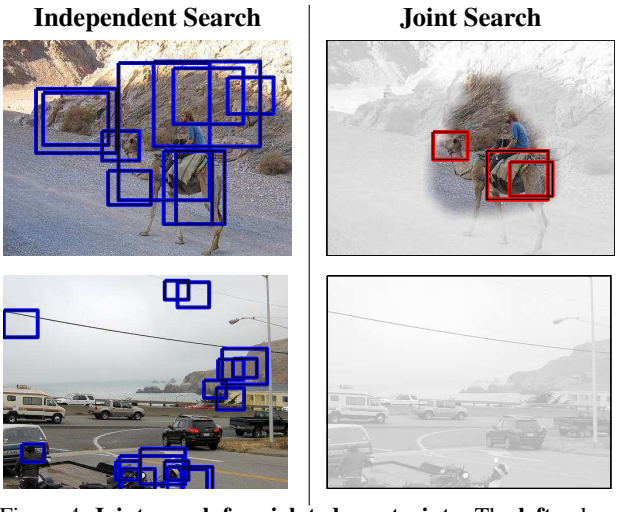


Figure 4. **Joint search for violated constraints:** The **left** column shows difficult negative examples when independently training a head detector. On the **right** are hard negative boxes for the jointly trained model. By only considering regions in the image where other detectors are also confident, indicated by the highlighted region, hard negatives from much (or all) of the image can be ignored, allowing the classifier to focus on a more constrained training and inference problem. See Section 4 for details on learning.

Initial Model Parameters. As observed in [9], it is important to provide good initializations when learning models with latent structures. We initialize the appearance models using detectors trained independently for each type t. Quadratic deformation costs are set to a small constant and linear deformation costs and biases are set to zero.

4. Structured Learning

We take a max-margin structured learning approach to train our model. This allows us to jointly train all of the appearance models and the spatial deformation parameters that tie them together. By jointly training all of the detectors, an individual detector can learn that it only needs to be correct when the other detectors provide sufficient evidence that an object is visible. Figure 4 illustrates this joint search. Further, by careful construction of a latent ground truth representation and structured loss, we can incorporate training examples with mixed levels of supervision, such as fully annotated examples with object and part boxes and partially annotated examples which are missing part annotations.

4.1 Structured Learning Objective

We begin by showing how the model can be parametrized as a linear weighting of features $\mathbf{\Phi}(\mathbf{h}; \mathbf{x}_i)$ induced by the inferred structure $\mathbf{h}$. Let ϕ_t and ψ_t be the detector and deformation features for each detector type. ϕ_t are the features

produced from the linear model of [9] for a detection of type t at location (x, y) and scale s. As shown in equation 2, the deformation costs are written as a weighted linear combination of the linear and quadratic deformations, giving the features $\psi_t(\mathbf{l}; \mathbf{p})$. A single feature vector $\mathbf{\Phi}$ is produced by concatenating the features from each detector type, with zeros for types not detected. To allow multiple detections of a single type, such as legs, we sum over the features of each detection.

Now, we can write the structured learning problem in the margin rescaled formulation of [25], written below in the unconstrained form:

$$\min_{\mathbf{w}} \frac{1}{2}||\mathbf{w}||_2^2 + \overbrace{C \sum_i \max_{\hat{\mathbf{h}} \in \mathcal{H}_{b_i}} \mathbf{w}^T \mathbf{\Phi}(\hat{\mathbf{h}}; \mathbf{x}_i) + \Delta(\mathbf{y}_i, \hat{\mathbf{h}})}^{F(\mathbf{w})} - \overbrace{C \sum_i \max_{\mathbf{h} \in \mathcal{H}_{GT}(\mathbf{y}_i)} \mathbf{w}^T \mathbf{\Phi}(\mathbf{h}; \mathbf{x}_i)}^{G(\mathbf{w})}. \quad (3)$$

While searching for weights $\mathbf{w}$, this objective imposes a penalty for each example where some $\mathbf{h} \in \mathcal{H}_b$ is within a margin (defined by the loss $\Delta(\mathbf{y}_i, \mathbf{h})$) from the highest scoring positive structure $\mathbf{h}_i^*$ chosen from the set of valid ground truth structures $\mathcal{H}_{GT}(\mathbf{y}_i)$.

Ground Truth Structure. The best ground truth structure $\mathbf{h}_i^*$ of each example $\mathbf{y}_i$ is a latent structure chosen from the set $\mathcal{H}_{GT}(\mathbf{y}_i)$. Any latent structure that has a highly overlapping detection for every ground truth box in $\mathbf{y}_i$ is considered correct and included in $\mathcal{H}_{GT}(\mathbf{y}_i)$. This allows us to incorporate structures with latent parts and mixed supervision. Missing annotations are simply treated as latent values that can be chosen freely. For an object labeled with object boxes, but not its parts, $\mathcal{H}_{GT}$ constrains the object detectors, while the parts can be detected or ignored based on their contribution to the overall score.

Loss. We use a Hamming loss $\Delta(\mathbf{y}, \mathbf{h})$ to measure the disagreement between a hypothesis $\mathbf{h}$ and the given ground truth $\mathbf{y}$. Here, a penalty of 1 is added for each false positive $\Delta_{\text{fp}}(\mathbf{y}, \mathbf{h})$ and false negative $\Delta_{\text{fn}}(\mathbf{y}, \mathbf{h})$. Each duplicate detection is counted as an additional false positive $\Delta_{\text{dd}}(\mathbf{y}, \mathbf{h})$:

$$\Delta(\mathbf{y}, \mathbf{h}) = \Delta_{\text{dd}}(\mathbf{y}, \mathbf{h}) + \Delta_{\text{fp}}(\mathbf{y}, \mathbf{h}) + \Delta_{\text{fn}}(\mathbf{y}, \mathbf{h}).$$

For examples which have not been fully labeled, there is no penalty for false positives, so $\Delta_{\text{fp}} = 0$.

4.2 Optimization

To minimize the learning objective in equation 3, we use the cutting-plane optimization procedure proposed in [25] for max-margin structured learning with latent variables. By writing the objective as the difference between two convex terms, $F(\mathbf{w})$ and $G(\mathbf{w})$, we can use the CCCP procedure [26] to find a local minimum (or saddle point). The CCCP algorithm iterates between computing a lower-bound

Algorithm 1 Optimization of Convex Subproblem

1: $\hat{H}_i = \emptyset$
2: **repeat**
3: t = 0
4: **for all** $\mathbf{x}_i$ **do**
5: $\hat{\mathbf{h}}_i = \text{argmax}_{\mathbf{h} \in \mathcal{H}_{b_i}} \mathbf{w}^T \mathbf{\Phi}(\mathbf{h}; \mathbf{x}_i) + \Delta(\mathbf{y}, \mathbf{h})$
6: $\hat{H}_i = \hat{H}_i \cup \{\hat{\mathbf{h}}_i\}$
7: **end for**
8: **repeat**
9: Randomly select example i, Let $\lambda = \frac{1}{c+t}$
10: $\hat{\mathbf{h}}_i = \text{argmax}_{\mathbf{h} \in \hat{H}_i} \left(\mathbf{w}^T \mathbf{\Phi}(\mathbf{h}; \mathbf{x}_i) + \Delta(\mathbf{y}_i, \mathbf{h})\right)$
11: $\mathbf{w}_{t+1} = \mathbf{w}_t - \lambda \left(\mathbf{w}_t + \mathbf{\Phi}(\hat{\mathbf{h}}_i) - \mathbf{\Phi}(\mathbf{h}_i^*)\right)$
12: $t = t + 1$
13: **until** Convergence
14: $\mathbf{w} = \mathbf{w}_t$
15: **until** Convergence

$G_t(\mathbf{w})$ of $G(\mathbf{w})$ and optimizing the resulting convex subproblem. At iteration t, we use the current model $\mathbf{w}_t$ to compute $G_t(\mathbf{w})$ by finding the highest scoring latent structures for each ground truth example:

$$\mathbf{h}_i^* = \underset{\mathbf{h} \in \mathcal{H}_{GT}(\mathbf{y}_i)}{\text{argmax}} \mathbf{w}_t^T \mathbf{\Phi}(\mathbf{h}; \mathbf{x}_i)$$
$$G_t(\mathbf{w}) = \sum_i \mathbf{w}^T \mathbf{\Phi}(\mathbf{h}_i^*; \mathbf{x}_i). \quad (4)$$

The resulting convex subproblem reduces to a non-latent structured learning problem, which we solve using cutting-plane based stochastic gradient descent in Algorithm 1. To avoid enumerating the exponential number of constraints, this algorithm incrementally builds up a set of violated constraints. At iteration t, we first update the constraint set $\hat{H}_i$ for each example with the new most violated constraint $\hat{\mathbf{h}}_i$ using loss augmented inference (Line 5) and then minimize the objective over these sets using stochastic gradient descent, continuing iteration until convergence.

4.3 Practical Considerations

Since this structured learning problem simultaneously trains multiple detectors, it requires dealing with large amounts of data, and there are several important practical tricks to speed up the optimization. **Constraint Generation:** Loss augmented inference requires running each detector at every iteration, making it important to minimize the number of iterations required to collect the entire constraint set. Since we compute the highest scoring structure for each root position to find the most violated constraint, we have easy access to a large number of additional violated examples. Therefore we choose the most violated structure along with a number of additional randomly sampled violated structures. This random sampling gives a diverse set of structures, giving a wider coverage of the constraint set. In practice, this inclusion of additional violated constraints greatly speeds convergence. **Constraint Caching:** One consequence of

including many extra structures is that irrelevant examples quickly accumulate. Therefore, we also employ a caching procedure similar to [9] which discards structures that have either remained outside the margin for a number of iterations or have consistently scored less than the most violated constraint. Waiting several iterations improves stability and avoids frequently discarding and adding the same constraints over time.

5. Inference

Generating a hypothesis for each root location $\mathbf{o}$ requires computing $\operatorname{argmax}_{\{\mathbf{h},b\}} f_b(\mathbf{h}; \mathbf{x}, \mathbf{o}, \mathbf{w})$. We greedily compute an approximate solution which works well in practice. We begin by computing the score of placing a detection at each anchor point. This requires computing the appearance scores for each type using individual detectors followed by pre-computing the deformation costs using max convolutions. To construct a hypothesis for each root location $\mathbf{l}_0$, we incrementally choose the highest scoring anchor point that satisfies all active constraints and then update the constraint set to include any additional constraints. For n anchor points, this results in a computational complexity of $O(n \log n + n^2)$ per root position.

Future Extensions. Although the greedy solutions are found to be good in practice, high scoring structures can be refined using exact inference cast as a mixed integer linear program. To further speed inference, the cascaded detection approach from [8] could lead to significant speedups since the cascade would tie together many categories and their parts through the broad category detector.

5.1 Augmented Inference

For training, we need to modify the inference procedure from the previous section to find the highest scoring ground truth structure (Eq 4) and the most violated constraint (Algorithm 1, line 5).

Ground Truth. To find the highest scoring ground truth structure, we restrict our search to detections with sufficient overlap with each ground truth part. To ensure that every ground truth part is assigned to an anchor point, we greedily add the highest scoring anchor point that agrees with the ground truth and active constraints until all of the ground truth windows are covered. If the training example is partially labeled, we add any remaining anchor points that satisfy the constraints and increase the hypothesis score. Finally, if the model does not allow a zero loss solution, we instead choose the highest scoring solution with the smallest possible loss.

Loss Augmented Inference. To find the most violated constraint, the score of a hypothesis is augmented by adding its corresponding loss. Note that the false positive and false negative losses decompose over detection windows. Therefore, each window that has insufficient overlap with a ground truth part has its score increased by one to account for the false positive loss. Each correct window's score is decreased by one, since choosing it will remove the loss contributed by a false negative. Notice that this establishes a margin of two between each positive and negative detection, similar to the binary SVM. Finally, we can (approximately) account for duplicate detections by incrementing all correct detection scores of a given part once it is correctly detected with another window.

5.2 Detection Rescoring

Our inference procedure only provides a single score for the highest scoring structure $\mathbf{h}^*$, but we also need a score for each constituent detection. The score S_b (eq. 1) for each detection incorporates the appearance score and spatial agreement with the hypothesis, but does not include the evidence from the rest of the structure. Instead, we use a convex combination of the overall hypothesis score f_b and the detection score S_b.

$$S_{\text{resc}}(\mathbf{l}_i; \mathbf{h}) = (1 - \alpha_{t_i}) f_b(\mathbf{h}) + \alpha_{t_i} S_b(\mathbf{l}_i). \quad (5)$$

The weight α_t is chosen for each detector type t with a grid search on a held out set.

6. Experiments

In the following section we evaluate the benefits of our shared body plan model. We test using four-legged categories, with the goal of detecting objects and their parts, whether familiar or unfamiliar. Our results show that our method surpasses strong baselines when generalizing across many categories (Table 1), or specializing for specific categories (Table 2).

Baseline. For each part and object detection task, we train an independent deformable part detector from [10]. Each of these detectors has the same parameterization as the appearance models in our structured models. Throughout all experiments, we use two components for each model, with five latent parts for each object detector and three latent parts for each part detector. To demonstrate the benefits of our joint training, we also include a baseline which learns the same spatial model, but with independently trained appearance models. To do this, we replace the HOG appearance features with two features: the score of a pre-trained detector and a bias.

Datasets. We use the CORE dataset [5] to train our fully supervised models. We use 75 training examples from the following four-legged animals in CORE: camel, dog, elk, elephant. For testing, CORE includes 75 examples from the previous classes, and adds 150 fully labeled cats and cows to evaluate detection accuracy of unfamiliar objects and parts. We use the Pascal 2008 validation set for further evaluation, which contains the familiar dog category, along with four unfamiliar categories: cat, cow, horse, and sheep. To train a dog model with mixed supervision, we include 240 additional dog boxes from the Pascal 2008 train set.

	Basic Level				Superordinate	Parts			Pascal	
	Camel	Dog	Elephant	Elk	Fam/Unf	Head	Leg	Torso	Dog	Unf
Independent	25.5	4.2	55.7	50.7	26.7/13.6	**9.9/1.7**	10.5/3.8	28.3/12.1	2.6	9.6
Indep+Spatial–Parts	29.4	3.3	52.3	52.7	30.0/**14.0**	—	—	—	3.5	9.9
Indep+Spatial+Parts	26.6	3.2	53.5	55.9	31.3/13.5	3.6/0.8	4.3/1.5	30.1/13.2	2.4	10.6
Joint Spatial+App–Parts	**29.5**	**6.4**	54.9	**59.9**	33.2/13.6	—	—	—	2.3	11.4
Joint Spatial+App+Parts	29.0	6.0	**57.8**	57.8	**35.5/14.0**	9.0/**1.7**	**16.1/4.3**	**33.9/14.2**	**4.2**	**12.2**

Table 1. **Broad Category Model Results:** We compare results for the task of detecting four-legged animals, their basic level categories, and their parts on CORE and Pascal using the AP_N measure (see Section 6). **Independent** are independently trained deformable part detectors for each task. **Indep+Spatial** combine the pre-trained independent models and with our spatial model. **Joint Spatial+App** is our full model where all appearance and spatial parameters are jointly trained. **-parts**,**+parts** indicate whether the model uses only object level detectors (e.g. four-legged, dog) or also includes part detectors. Pairs of numbers indicate Familiar/Unfamiliar results. Our full model with parts is able to make a wide range of detailed predictions, outperforming many of the baselines.

Evaluation. We evaluate detection accuracy using a normalized version of the average precision measure used in the Pascal detection challenge, indicated by AP_N. To account for different numbers of test examples of parts and categories, the true positive count is normalized by $\frac{N}{N_j}$, where N is a constant factor and N_j the number of examples in the category j. N is chosen to be $0.15N_{\text{images}}$, approximately the average N_j on Pascal. Note that in our results, the relative ordering of the methods do not change, but the numbers are more directly comparable across tasks. As with Pascal, object detections are considered correct if the bounding box has at least 50% overlap with the ground truth. Part detections are correct with 25% overlap, as in [7].

Results. To test the benefits of a shared representation across four-legged animals, we first tie the four familiar basic level detectors together with a superordinate four-legged detector (**Joint Spatial+App-Parts**). Next, we create a second more detailed model by adding the shared supervised parts to the previous model (**Joint Spatial+App+Parts**). From Table 1 it is clear that jointly training all of the detectors yields a great benefit over tying together independently trained detectors. Second, although the joint models without parts are slightly better for basic level detection on CORE, the parts are important for improving accuracy for the broad category detection task, especially for familiar four-legged animals on CORE and for all objects on Pascal. By including part detectors we are also able to make more detailed predictions about both familiar and unfamiliar objects on both datasets. See Figure 5 for qualitative results.

Next, we construct a separate structured model for each individual category that jointly trains a basic level detector with the part detectors. Results are shown in Table 2. Again, the full joint model greatly improves object detection, while still providing detailed part predictions. These results further emphasize the importance of jointly training the appearance models. For many of the tasks, such as detecting elk parts, joint training is essential to get gains from our spatial model, indicating that some pre-trained detectors may not be well suited for use in the spatial model. Here we see more significant gains for object detection than with the broad model because the part models are allowed to specialize for each category. This suggests building a hierarchical model where each part detector in the broad model is supplemented by category specific part detectors. The broad model can retain its broad generalization across four-legged animals, while specializing for familiar categories.

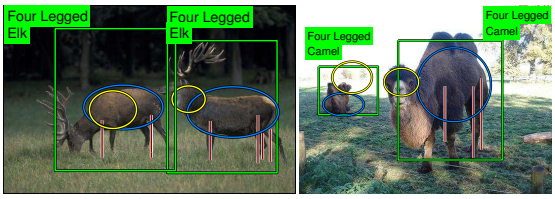
		Independent	Independent +Spatial	Joint Spatial+App
Camel	Object	25.5	27.9	**30.1**
	Head	22.3	13.3	**41.2**
	Leg	6.7	11.0	**19.4**
	Torso	30.0	35.4	**38.0**
Dog	Object	4.2	**20.8**	18.3
	Head	36.9	32.8	**40.8**
	Leg	1.4	4.2	**10.5**
	Torso	5.9	8.7	**14.6**
Elephant	Object	55.7	53.4	**62.3**
	Head	30.2	31.4	**46.0**
	Leg	13.7	32.0	**34.8**
	Torso	**53.2**	48.8	51.0
Elk	Object	50.7	50.0	**58.7**
	Head	37.2	4.6	**47.9**
	Leg	21.4	9.3	**31.7**
	Torso	48.2	56.1	**58.3**

Table 2. **Per-Category Model Results:** Part and Object Detection on CORE. For each category, we train a body plan based model for the object and its parts. We compare our **Joint** model to independently trained detectors without (**Independent**) and with a spatial model (**Independent+Spatial**). Jointly training appearance models with our spatial model again greatly improves performance for all but two tasks.

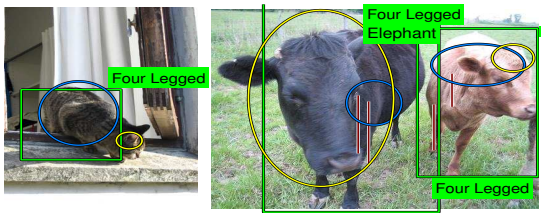
Finally, we consider the task of improving dog detection by adding additional examples with object-level boxes from Pascal. On CORE, dog detection AP_N increases to 21.4 and on Pascal increases to 7.9 from 4.2 for an independent dog detector trained on CORE and Pascal. A fully structured model trained only on CORE falls in between with AP_N of 6.0. These promising results show that our latent definition of the ground truth structure can be used for flexible learning and can lead to even greater gains.

7. Conclusions and Future Work

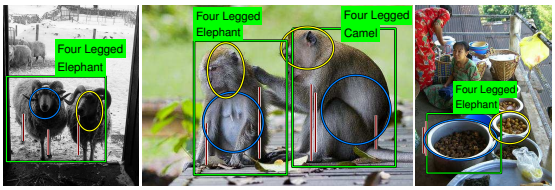
In this work, we treat recognition of many objects as a unified problem. When presented with many supervised detection tasks, our jointly trained detectors excel when compared to detectors that are tied together after being trained in isolation. By jointly training all of the appearance mod-



(a) Familiar Categories



(b) Unfamiliar Categories



(c) Common Mistakes

Figure 5. High scoring structures after non-maximum suppression (Yellow ellipse: head, blue ellipse: torso, red line: leg). (a) Familiar categories seen during training: our model can handle a variety of poses and missing parts (e.g., left camel). (b) Cats and cows were never seen during training, but we can still provide detailed predictions. Note the vast change in scale between the cat and cow head detections. (c) Typical mistakes include collecting parts from multiple nearby objects, predicting basic level labels for unfamiliar objects, and hallucinating parts in scenes with strong contours.

els with the spatial model, they can learn that they need only be confident in the presence of other strong object evidence. Our flexible definition of valid ground truth structures can be used to incorporate examples with incomplete annotations. For dogs, we show that adding training examples with only object level boxes can further improve accuracy.

Our results motivate several important future directions in representing and learning about objects. Models should aim to capture similarities between related categories, allowing better generalization for familiar and unfamiliar objects, while also specializing to capture the detailed differences between categories, giving better discrimination. Further, by including detailed annotations such as parts, we can inject high level knowledge that can improve recognition and give detailed predictions about objects we cannot name. By further exploring mixed supervision, we can include this detailed knowledge without requiring that we collect this more costly annotation every single object. The broad models with mixed supervision can also allow quick bootstrapping for learning about new related objects, requiring fewer detailed annotations. Code for learning and inference with our structured models can be found at http://vision.cs.uiuc.edu/bodyplans.

Acknowledgements

This research was supported in part by NSF CAREER Award 1053768, ONR MURI Award N000141010934, and a Google Research Award.

References

[1] L. Bourdev and J. Malik. Poselets: Body part detectors trained using 3d human pose annotations. In *ICCV*, 2009. 2

[2] M. Burl, M. Weber, and P. Perona. A probabilistic approach to object recognition using local photometry and global geometry. In *ECCV*. 1998. 2

[3] D. Crandall and D. Huttenlocher. Weakly supervised learning of part-based spatial models for visual object recognition. In *ECCV*. 2006. 2

[4] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005. 2

[5] I. Endres, A. Farhadi, D. Hoiem, and D. A. Forsyth. The benefits and challenges of collecting richer object annotations. In *ACVHL workshop (in conjunction with CVPR)*, 2010. 6

[6] B. Epshtein and S. Ullman. Semantic hierarchies for recognizing objects and parts. In *CVPR*, 2007. 2

[7] A. Farhadi, I. Endres, and D. Hoiem. Attribute-centric recognition for cross-category generalization. *CVPR*, 2010. 2, 4, 7

[8] P. Felzenszwalb, R. Girshick, and D. McAllester. Cascade object detection with deformable part models. In *CVPR*, 2010. 6

[9] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *PAMI*, 2010. 2, 3, 4, 5, 6

[10] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Discriminatively trained deformable part models, release 3. http://cs.brown.edu/ pff/latent-release3/. 6

[11] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *CVPR*, 2003. 2

[12] M. Fischler and R. Elschlager. The representation and matching of pictorial structures. *Computers, IEEE Transactions on*, 1973. 2

[13] M. Girolami. Mercer kernel-based clustering in feature space. *IEEE Transactions on Neural Networks*, 2002. 4

[14] S. Ioffe and D. Forsyth. Probabilistic methods for finding people. *IJCV*, 43, 2001. 2

[15] Y. Jin and S. Geman. Context and hierarchy in a probabilistic image model. In *CVPR*, 2006. 2

[16] K. Mikolajczyk, C. Schmid, and A. Zisserman. Human detection based on a probabilistic assembly of robust part detectors. In *ECCV*, 2004. 2

[17] A. Opelt, A. Pinz, and A. Zisserman. Learning an alphabet of shape and appearance for multi-class object detection. *IJCV*, 2008. 2

[18] P. Ott and M. Everingham. Shared parts for deformable part-based models. In *CVPR*, 2011. 2

[19] H. Rowley, S. Baluja, and T. Kanade. Neural network-based face detection. *PAMI*, 1998. 2

[20] H. Schneiderman and T. Kanade. A statistical model for 3-d object detection applied to faces and cars. In *CVPR*, 2000. 2

[21] M. Sun and S. Savarese. Articulated part-based model for joint object detection and pose estimation. In *ICCV*, 2011. 2

[22] A. Torralba, K. P. Murphy, and W. T. Freeman. Sharing features: efficient boosting procedures for multiclass object detection. In *CVPR*. 2004. 2

[23] A. Vedaldi, V. Gulshan, M. Varma, and A. Zisserman. Multiple kernels for object detection. In *ICCV*, 2009. 2

[24] P. Viola and M. J. Jones. Robust real-time face detection. *IJCV*, 2004. 2

[25] C.-N. J. Yu and T. Joachims. Learning structural svms with latent variables. In *ICML*, 2009. 5

[26] A. Yuille, A. Rangarajan, and A. L. Yuille. The concave-convex procedure (cccp). In *NIPS*. MIT Press, 2002. 5

[27] J. Zhang, S. Lazebnik, and C. Schmid. Local features and kernels for classification of texture and object categories: a comprehensive study. *IJCV*, 2007. 2