Learning to Find Common Objects Across Few Image Collections

Amirreza Shaban*[1], Amir Rahimi*[2], Shray Bansal[1],
Stephen Gould[2], Byron Boots[1], and Richard Hartley[2]

[1]Georgia Tech, [2]ACRV, ANU Canberra

Abstract

Given a collection of bags where each bag is a set of images, our goal is to select one image from each bag such that the selected images are from the same object class. We model the selection as an energy minimization problem with unary and pairwise potential functions. Inspired by recent few-shot learning algorithms, we propose an approach to learn the potential functions directly from the data. Furthermore, we propose a fast greedy inference algorithm for energy minimization. We evaluate our approach on few-shot common object recognition as well as object co-localization tasks. Our experiments show that learning the pairwise and unary terms greatly improves the performance of the model over several well-known methods for these tasks. The proposed greedy optimization algorithm achieves performance comparable to state-of-the-art structured inference algorithms while being ∼10 times faster.

1. Introduction

We address the problem of finding images of a common object across bags of images. The input is a collection of bags, each containing several images from multiple classes. A bag is labelled as *positive* with respect to a given object class if it contains at least one image from that class and *negative* if none of the images in the bag are from the object class. The task is to find an instance of the common object in each positive bag. It is not assumed that objects of the common class have been seen previously during training.

Since collections of images may accidentally contain irrelevant common objects (for instance indoor images often contain person), the purpose of a negative bag is to indicate objects we are *not* looking for, but which may be common to the positive bags.

Several computer vision problems, including co-segmentation, co-localization, and unsupervised video object tracking and segmentation have been formulated in this way [48, 14, 23, 16, 3]. In the co-localization problem, Figure 1, each bag contains many cropped image regions (object proposals) from one image. The goal is to identify proposals, one per positive bag, that contain the common object. We design our approach to address the general problem of finding common objects from positive bags and evaluate it on two problems: few-shot common object recognition and object co-localization.

Figure 1. *Co-localization, shown here, is an instance of the general problem of finding common objects addressed in this paper. Each image in the top row generates a positive bag containing a set of cropped regions from that image. The task is to find a common object from the positive bags by selecting one region from each image (green bounding boxes). Cropped regions from the images in the bottom row form a negative bag as they do not contain the common object. The negative bag is optional here but can reduce ambiguity. For example, since a knife is present in the negative bag it can not be the desired common object.*

Weakly supervised classification methods like multiple-instance learning [29] have been used to address this type of problems, but they require many training bags to learn new concepts [24]. Meta-learning techniques [15, 39, 40] have been shown to reduce the need for training instances in few-shot learning, but these methods require full supervision for the new classes.

We model the problem of finding common objects

*Equal contribution. Contact at amirreza@gatech.edu or amir.rahimi@anu.edu.au.

as a minimum-energy graph labelling problem, otherwise known as a bidirectional graphical model or Markov Random Field. Each node of the graph corresponds to a positive bag and a graph labelling corresponds to choosing one image in each positive bag, the goal being to find a labelling that contains the common object. We use the word *selection* instead of *labelling* to refer to the process of selecting one image from each bag. The energy minimization problem uses unary and pairwise potential functions, where unary potentials reflect the relation of images in the positive bags to the images in the negative bag and the pairwise potentials derive from a similarity measure between pairs of images from two positive bags. The unary and pairwise potentials are computed using similar, but separately trained networks. We adapt the relation network [44], which has been successfully used in few-shot recognition to compute pairwise potentials, and propose a new algorithm that uses the relationship of an image to all of the images in the negative bag to provide unary potentials.

Once unary and pairwise potentials have been computed, a simple merge-and-prune inference heuristic is used to find a minimum-cost labelling. This provides a simple but effective solution to the NP-hard problem of optimal graph labelling.

Although graphical models have been used for Multiple Instance Learning (MIL) problems [12, 19], our method uses a learning-based approach, inspired by meta-learning, to increase the generalization power of potential functions to novel classes.

We make the following contributions:

1. We introduce a method to transfer knowledge from large-scale strongly supervised datasets by learning pairwise and unary potentials and demonstrate the superiority of this learned relation metric to earlier MIL approaches on two problems.

2. We propose a specialized algorithm for structured inference that achieves comparable performance to state-of-the-art inference methods while requiring less computational time.

2. Related Work

Multiple instance learning (MIL) [7, 33] methods have been used for learning weakly supervised tasks such as object localization (WSOL) [25, 8, 53, 41]. In a standard MIL framework, instance labels in each positive bag are treated as hidden variables with the constraint that at least one of them should be positive. MI-SVM and mi-SVM [2] are two popular methods for MIL, and have been widely adapted for many weakly supervised computer vision problems, achieving state-of-the-art results in many different applications [7, 13]. In these methods, images in each bag inherit the label of the bag and an SVM is trained to classify images. The trained SVM is used to relabel the instances and this process is repeated until the labels remain stable. While in MI-SVM only the image with the highest score in positive bags are labeled as positive, mi-SVM allows more than one positive label in each positive bag in the relabeling process.

Co-saliency [52, 23], co-segmentation [48, 14, 21], and co-localization [27] methods have the same kind of output as WSOL methods. Similar to standard MIL algorithms, some of these methods rely on a relatively large training set for learning novel classes [27, 45]. The main difference between these methods and WSOL methods is that they usually do not utilize negative examples [48, 27, 45]. Negative examples in our method are optional and could be used to improve the results of the co-localization task.

Our approach is related to weakly supervised methods that make use of auxiliary fully-labelled data to accelerate the learning of new categories [46, 22, 42, 37, 11]. Since visual classes share many visual characteristics, knowledge from fully-labelled source classes is used to learn from the weakly-labelled target classes. The general approach is to use the labelled dataset to learn an embedding function for images and use MI-SVM to classify instances of the weakly labelled dataset in this space [46, 22, 42]. We show that learning a scoring function to compare images in the embedded space significantly improves the performance of this approach, especially when few positive images are available. Rochan et al. [37] propose a method to transfer knowledge from a set of familiar objects to localize new objects in a collection of weakly supervised images. Their method uses semantic information encoded in word vectors for knowledge transfer. In contrast, our method uses the similarity between tasks in training and testing and does not rely solely on a given semantic relationship between the familiar and new classes. Deselaers et al. [11] transfer objectness scores from source classes and incorporate them into unary terms of a conditional random field formulation.

Our approach inspired by methods that use the meta-learning paradigm for few-shot classification. These methods simulate the few-shot learning task during the training phase in which the model learns to optimize over a batch of sampled tasks. The meta-learned method is later used to optimize over similar tasks during testing. Optimization-based methods [36, 15], feature and metric learning methods [49, 43, 44], and memory augmented-based methods [39] are just a few examples of modern few-shot learning. While our work is inspired by these methods, it is different in the sense that we do not assume strong supervision for the tasks. In relation networks [44] a similarity function is learned between image pairs and used to classify images from unseen classes. We adopt this method to learn the unary and pairwise potential functions in our graphical model.

3. Problem description

We consider a set $\mathcal{I}$ with a binary relation R. The elements of the set are called *images* in our work for simplicity of exposition. A *relation* R is simply a subset of $\mathcal{I} \times \mathcal{I}$:

$$R(e, e') = \begin{cases} +1, & \text{if } (e, e') \in R \text{ (inputs are related)} \\ -1, & \text{otherwise.} \end{cases} \quad (1)$$

A *bag* is a set of images, thus, a subset of $\mathcal{I}$. We will be concerned with collections of bags, $\mathcal{V} = \{v_1, v_2, \ldots, v_N\}$. We say that a collection $\mathcal{V} = \{v_1, \ldots, v_N\}$ is *consistent* if it is possible to select images, one from each bag, so that they are all related in pairs. These are known as *positive* bags.

Given a consistent collection, $\mathcal{V}$ and an optional additional bag $\bar{v}$ that we designate as *negative*[1], the task is to output a *selection* of images, namely an ordered set $O = (e_1, \ldots, e_N)$ where e_i is from positive bag v_i, such that the images are pairwise related, $R(e_i, e_j) = 1$, and that not all images are pairwise related to any image in the negative bag, i.e., $\exists e_i \in O$ such that $\forall \bar{e} \in \bar{v}, R(e_i, \bar{e}) = -1$.

The situation of most interest is where each of the images $e \in \mathcal{I}$ has a single latent (unknown) label $c_e \in \{c_\varnothing\} \cup \mathscr{C}$ where $c_\varnothing$ is a *background class* and $\mathscr{C}$ is a set of *foreground classes*. Two images e_1 and e_2 are related if their labels are the same and belong to a foreground class, i.e., $c_{e_1} = c_{e_2} \in \mathscr{C}$. For example, (cropped) images may be labelled according to the foreground object they contain. In this case, two images (e_1, e_2), both containing a "cake" are related, $R(e_1, e_2) = 1$. Whereas two images (e_3, e_4) that are not of the same foreground class are unrelated, $R(e_3, e_4) = -1$. In this case, R is an equivalence relation.

Energy function. We pose the problem of finding the common object as finding a selection O that minimizes an energy function. Our energy function is defined as sum of potential functions as follows:

$$E(O \mid \bar{v}) = \sum_{\substack{e_i, e_j \in O \\ i > j}} \psi^{\mathrm{P}}_\theta(e_i, e_j) + \eta \sum_{e_i \in O} \psi^{\mathrm{U}}_\beta(e_i \mid \bar{v}), \quad (2)$$

in which $\psi^{\mathrm{P}}_\theta(\cdot, \cdot)$ and $\psi^{\mathrm{U}}_\beta(\cdot \mid \bar{v})$ are pairwise and unary potential functions with trained parameters θ and β, and hyperparameter $\eta \geq 0$ controls the importance of the unary terms. Both unary and pairwise potential functions are learned by neural networks, which will be described in Section 4. The pairwise potential function is learned so that it encourages choosing pairs that are related to each other. The unary potential is chosen so it is minimized when its input is not related to the images in the negative bag. In this way, the overall energy is minimized when images in O are related to each other and unrelated to images in the negative bag.

[1] There is no point in having more than one negative bag in a collection since its purpose is simply to provide a set of images that are not compatible with the positive bags, in the sense described.

3.1. Training and Test Splits

For a dataset $\mathcal{D} \subseteq \mathcal{I}$, we use the notation $\mathcal{W} \sim \mathcal{D}$ to indicate that a random collection $\mathcal{W} = (\mathcal{V}, \bar{v})$ is drawn from the dataset. We define the sampling strategy in the implementation details for each dataset. During training, algorithms have access to a dataset $\mathcal{D}_{\text{train}}$ and corresponding ground-truth relation. We construct the relation for the training dataset based on a set of foreground classes $\mathscr{C}_{\text{train}}$ as described above.

Methods are evaluated on samples from a test dataset $\mathcal{W} \sim \mathcal{D}_{\text{test}}$. There are no image in common between the training and test datasets. Moreover, the set of foreground classes $\mathscr{C}_{\text{test}}$ used for the test dataset is disjoint from the set of foreground classes used during training, i.e., $\mathscr{C}_{\text{test}} \cap \mathscr{C}_{\text{train}} = \varnothing$. At test time we only know whether a bag is positive or negative with respect to some foreground class. The ground-truth (which foreground class is common to the positive bags) is unknown to the algorithm and is only used for evaluating performance.

4. Learning the potential functions

We now present the method for learning the pairwise and unary potential functions. The proposed method relies on an algorithm to estimate a similarity measure of an input image pair (e, e'). One common approach is to learn an embedding function and use a fixed distance metric to compare the input pairs in the embedded space. In this approach, the learning is used only to determine the embedding function. The relation network [44] extends this by jointly learning the embedding function and a comparator. The network consists of embedding and relation modules. The *embedding module* learns a joint feature embedding (into $\mathbb{R}^d$) for the input pair of images $\mathcal{C}(e, e')$ and the *relation module* learns a mapping $g : \mathbb{R}^d \to \mathbb{R}$, mapping the embedded feature to a relation score $r_\phi(e, e') = g(\mathcal{C}(e, e'))$ where ϕ denotes the parameters of the embedding and scoring functions combined.[2] We adopt the relation module from the Relation Network due to its simplicity and success in few-shot learning. However, any other method which computes the relationship between a pair of images could be used in our method.

Relation network. As we need to evaluate the relation of many image pairs, we adapt the original relation network architecture [44] in order to make the embedding and scoring functions as computationally efficient as possible. The

[2] We adopt the notation used in the relation network paper [44]

feature embedding function $\mathcal{C}(\cdot,\cdot) : \mathcal{I} \times \mathcal{I} \to \mathbb{R}^d$ consists of feature concatenation and a single linear layer with gated activation [47] and skip connections. Let f and f' be features in $\mathbb{R}^d$ extracted from images e and e' by a CNN *feature extraction module* and $[f, f']$ be the concatenation of feature pairs. The embedding function is defined as:

$$\mathcal{C}(e, e') = \tanh(W_1[f, f'] + b_1)\sigma(W_2[f, f'] + b_2) + \frac{f + f'}{2}$$

where $W_1, W_2 \in \mathbb{R}^{d \times 2d}$ and vectors $b_1, b_2 \in \mathbb{R}^d$ are the parameters of the feature embedding module and $\tanh(\cdot)$ and $\sigma(\cdot)$ are hyperbolic tangent and sigmoid activation functions respectively, applied componentwise to vectors in $\mathbb{R}^d$. Then, we use a linear layer to map this features into relation score

$$r_\phi(e, e') = w^\top \mathcal{C}(e, e') + b$$

where $w \in \mathbb{R}^d$ and $b \in \mathbb{R}$. We found in practice that using gated activation in the embedding module improves the performance over a simple ReLU, whereas adding more layers does not affect the performance. We note that the effectiveness of gated activation has also been shown in other work [35].

Pairwise potentials. The pairwise potential function is defined as the negative of the output of the relation module: $\psi^{\mathrm{P}}_\theta(e_i, e_j) = -r_\theta(e_i, e_j)$ so it has a lower energy for related pairs. For a sampled collection $\mathcal{V}$ the episode loss is written as a binary logistic regression loss

$$\mathcal{L}^{\mathrm{P}} = \frac{1}{N_P} \sum_{(e_i,e_j)\sim\mathcal{V}} \log\Big(1 + \exp(-R(e_i, e_j) r_\theta(e_i, e_j))\Big)$$

where the sum is over all the pairs in the collection, N_P is the total number of such pairs, and relation $R(.,.)$ defined in Eq (1) provides the ground-truth labels.

Note that image pairs are sampled from $\mathcal{V}$, so that the loss function reflects prior distributions of image pairs from consistent image collections.

Unary potentials. The unary potential $\psi^{\mathrm{U}}(e \mid \bar{v})$ is constructed by comparing image e with images in the negative bag $\bar{v}$. Let the vector $u(e, \bar{v})$ be the estimated relation between image e and all the images in $\bar{v}$, that is, $u(e, \bar{v})_j = r_\beta(e, \bar{e}_j)$ where $\bar{e}_j$ is the j-th image in the negative bag and β is the (new) set of parameters for the relation network. By definition, the unary energy for an image e should be high if at least one of the values in $u(e, \bar{v})$ is high. In other words, e is related to $\bar{v}$ if it is related to at least one image in $\bar{v}$. This suggests the use of $\max_j(u(e, \bar{v})_j)$ as the unary energy potential. However, depending on the class distribution of images in the negative bag, an image e which is not from the common object class could be related to more than just one image from the negative bag. In this case, using the average relation to the few mostly related elements in $u(e, \bar{v})$ helps to reduce the noise in the estimation and works better than a simple max operator. This motivates us to use a form of exponential-weighted average of the relations so that higher values get a higher weight

$$\psi^{\mathrm{U}}_{\beta,\nu}(e \mid \bar{v}) = \frac{\sum_{k=1}^{\bar{B}} u(e, \bar{v})_k \exp\big(\nu\, u(e, \bar{v})_k\big)}{\sum_{k=1}^{\bar{B}} \exp(\nu\, u(e, \bar{v})_k)} \,. \qquad (3)$$

Here, $\bar{B}$ is the total number of images in the negative bag and ν is the temperature parameter. Observe that for $\nu = 0$ we have the mean value of $u(e, \bar{v})$ and it converges to the max operator as $\nu \to +\infty$. We let the algorithm learn a balanced temperature value in a data-driven way.

For a sampled collection $\mathcal{W} = (\mathcal{V}, \bar{v})$, the episode loss for the unary potential is defined as a binary logistic regression loss

$$\mathcal{L}^{\mathrm{U}} = \frac{1}{N_U} \sum_{v\in\mathcal{V}} \sum_{e\in v} \log\Big(1 + \exp(-R(e, \bar{v})\psi^{\mathrm{U}}_{\beta,\nu}(e, \bar{v}))\Big) \qquad (4)$$

where we use an extended definition of the relation function where $R(e, \bar{v}) = \max_{\bar{e}\in\bar{v}} R(e, \bar{e})$ and N_U is the total number of images in all positive bags in the collection. Through training, this loss is minimized over choices of parameters β of the relation network, and the weight parameter ν. By optimizing this loss, we learn a potential function that has higher value if e is related to one example in the negative bag. Note that in Eq (2) selection of unary potentials with high values are discouraged.

As before, training samples are chosen from collections $\mathcal{W}$ to reflect the prior distributions of related and unrelated pairs.

Parameters of the unary and pairwise potential functions are learned separately by optimizing the respective loss functions over randomly sampled problems from the training set. Although both unary and pairwise potential functions use the relation network with an identical architecture, their input class distributions are different, since one is comparing images in positive bags and one is comparing images in positive and negative bags. Thus, sharing their parameters decreases overall performance.

4.1. Inference

Finding an optimal selection O that minimizes the energy function defined in Eq (2) is NP-hard and thus not feasible to compute exactly, except in small cases. Loopy belief propagation [50], TRWS [26], and AStar [4], are among the many algorithms used for approximate energy minimization. We propose an alternative approach specifically designed for solving our optimization problem.

Our approach is designed to decompose the overall problem into smaller subproblems, solve them, and combine

their solutions to find a solution to the overall problem. This is based on the observation that a solution to the overall problem will also be a valid solution to any of the subproblems. Let $\mathcal{V}(p,q) = \{v_p, v_{p+1}, ..., v_q\}$ be a subset of $\mathcal{V}$. Then, a subproblem refers to finding a set of common object proposals $\mathcal{B}$ for $\mathcal{V}(p,q)$ with low energy values; $\mathcal{B}$ represents a collection of proposed selections of images from the set of bags $\mathcal{V}(p,q)$. The energy value for a selection $O_{p,q} \in \mathcal{B}$ is defined as sum of all pairwise and unary potentials in the subproblem, similar to how the energy function is defined for the overall problem in Eq (2).

The decomposition method starts at the root (i.e., full problem) and divides the problem into two disjoint subproblems and recursively continues dividing each into two subproblems until each subproblem only contains a single bag v_i. If $N = 2^Z$, then this can be represented as a full binary tree[3] where each node represents a subproblem. Let $\mathcal{N}_i^l$ be the i-th node at level l. Then root node $\mathcal{N}_1^Z$ represents the full problem, nodes $\mathcal{N}_i^l$ at any given level l represent disjoint subproblems of the same size, and the leaf nodes, $\mathcal{N}_i^0$, at level 0 of the tree each represent a subproblem with only one positive bag v_i.

Each level in the tree maintains a set of partial solutions to the root problem. Computation starts at the lowest level (leaf nodes) where each partial solution is simply one of the images for all images in the bag. At the next level, each node combines the partial solutions from its child nodes and prunes the resulting set to form a new set of partial solutions for its own subproblem, which in turn is used as input to nodes at the next level in the tree and so on until we reach the root node, which is the output for the optimization. The joining procedure used to combine the partial solutions from two child nodes is described next.

Joining: Node i at level l receives as input solution proposals $\mathcal{B}_{2i-1}^{l-1}$ and $\mathcal{B}_{2i}^{l-1}$ from its child nodes $\mathcal{N}_{2i-1}^{l-1}$ and $\mathcal{N}_{2i}^{l-1}$. The joining operation simply concatenates every possible selection from the first set with every possible selection in the second set and forms a set of selection proposals $\mathcal{X}_i^l$ for the subproblem

$$\mathcal{X}_i^l = \{[O^l, O^r] \mid O^l \in \mathcal{B}_{2i-1}^{l-1}, O^r \in \mathcal{B}_{2i}^{l-1}\}$$

where $[\cdot,\cdot]$ concatenates two selection sequences. We denote the joining operation by the Cartesian product notation, i.e., $\mathcal{X}_i^l = \mathcal{B}_{2i-1}^{l-1} \times \mathcal{B}_{2i}^{l-1}$.

Pruning: Since combining the partial solutions from two nodes results in a quadratic increase in the number of partial solutions, the number of potential solutions grows exponentially as we ascend the tree. Also, not all the generated partial solutions contain a common object. Therefore, we use a pruning algorithm $\mathcal{B}_j^l = \text{prune}(\mathcal{X}_j^l; k)$ that picks the k selections with the lowest energy values. The energy values for each subproblem can also be efficiently computed from bottom to top. At the lowest level, the energy for each selection is the unary potential from Eq (3),

$$E_i^0(O_{i,i}) = \eta\psi_\beta^{\text{U}}(e_i \mid \bar{v}) \quad \forall e_i \in \mathcal{B}_i^0 = v_i, \tag{5}$$

Note that selection $O_{i,i} = (e_i)$ consists of only one image. Starting at the leaves, energy in all nodes can be computed recursively. Let $O \in \mathcal{X}_i^l$ be formed by joining two selection proposals $O^l \in \mathcal{B}_{2i-1}^{l-1}$ and $O^r \in \mathcal{B}_{2i}^{l-1}$. The energy function $E_i^l(O)$ can be factored as

$$E_i^l(O) = E_{2i-1}^{l-1}(O^l) + E_{2i}^{l-1}(O^r) + P(O^l, O^r) \tag{6}$$

where $P(\cdot,\cdot)$ is the sum of all pairwise potentials on edges joining the two subproblems and is computed on the fly. Algorithm 1 summarizes the method. A good value of k in the pruning method depends on the ambiguity of the task. It is possible to construct an adversarial example that needs *all* possible proposals at the root node to find the optimal solution. However, in practice, we found that k does not need to be large to achieve good performance. Importantly, unlike other methods, this algorithm does not necessarily compute all of the pairwise potentials. For example, if an object class only appears in a small subproblem, the images of that class will get removed by nodes whose subproblem size is large enough. Thus, in the next level of the tree, the pairwise potentials between those images and other images is no longer required. In general, the number of pairwise potentials computed depends on both the value of k and the dataset. We observed that only a small fraction of the total pairwise potentials were required in our experiments.

Algorithm 1: Greedy Optimization Algorithm

Input: $\mathcal{V} = \{v_1, ..., v_N\}$, $\bar{v}$, and $N = 2^Z$.
Output: Selection $O = (e_1, \ldots, e_N)$
$\mathcal{B}_i^0 = v_i \quad \forall i \in [1, \ldots N]$
$E_i^0(O_{i,i}) = \eta\psi_\beta^{\text{U}}(e_i \mid \bar{v}) \quad \forall e_i \in \mathcal{B}_0^i,\ i \in [1, \ldots, N]$
for $l \leftarrow 1$ ***to*** Z **do**
  for $i \leftarrow 1$ ***to*** 2^{Z-l} **do**
    $\mathcal{X}_i^l \leftarrow \mathcal{B}_{2i-1}^{l-1} \times \mathcal{B}_{2i}^{l-1}$ (joining)
    Compute $\mathcal{X}_i^l$ Energies According to Eq (6)
    $\mathcal{B}_i^l \leftarrow \texttt{prune}(\mathcal{X}_i^l; k)$ (pruning)
return $O \in \mathcal{B}_1^Z$ *with the minimum energy*

5. Experiments

We evaluate the proposed algorithm on few-shot common object recognition and co-localization tasks. For each task, we first pre-train a CNN feature extractor module to perform classification on the seen categories from the training dataset. We then use the learned CNN to compute a

[3]This is without loss of generality since zero padding could be used if the number of positive bags is not a power of 2.

feature descriptor of each image. This ensures a consistent image representation for all methods under consideration.

For learning pairwise and unary potentials, stochastic gradient descent with gradual learning rate decay schedule is used. The complete framework ("Ours" in the tables) uses greedy optimization method described in Algorithm 1. The optimal value of η in Eq (2) is found using grid search. In all experiments, a maximum of $k = 300$ top selection proposals are kept in the greedy algorithm.

All experiments are done on a single Nvidia GTX 2080 GPU and 4GHz AMD Ryzen Threadripper 1920X CPU with 12 Cores[4].

5.1. Baseline Methods

We compare the greedy optimization algorithm to AStar [4] which is used for object co-segmentation [48] and the faster TRWS [26] which is used for inference on MIL problems [12, 11]. We use a highly efficient parallel implementation of these algorithms [1]. The proposed method is compared to SVM based and attention based MIL baselines described below.

SVM based MIL. We report the results of the three well-known approaches: MI-SVM [22], mi-SVM [2] and sb-MIL [6] using publicly available source code [13]. The sb-MIL method is specially designed to deal with sparse positive bags. The RBF and linear kernel are chosen as they work better on few-shot common object recognition and co-localization respectively. Grid search is performed in order to select the hyperparameters.

Attention based deep MIL. Along with the SVM based methods, the results of the more recent attention based deep learning MIL method [24] (ATNMIL) is presented on our benchmarks. After training the model, we select the image proposal with the maximum attention weight from each positive bag.

5.2. Few-shot Common Object Recognition

In this task we make use of the *mini*ImageNet dataset [49]. The advantage of *mini*ImageNet is that we can compare many different design choices without requiring large scale training and performance evaluations. The dataset contains $60,000$ images of size 84×84 from 100 classes. We experiment on the standard split for this task of 64, 16 and 20 classes for training, validation and testing, respectively [36].

For the CNN feature extractor module, a Wide Residual Network (WRN) [51] with depth 28 and width factor 10 is pre-trained on the training split. The $d = 640$ dimensional output of global average pooling layer of the pre-trained network is provided as input to all the methods.

To construct bags, we first randomly select M classes out of all the possible classes $\mathscr{C}$. One of these is selected to be the target and the rest are considered non-target classes. Then, each positive bag is constructed by randomly sampling one image from the target class and $B-1$ images from the target and non-target classes. The negative bag is built by sampling $\bar{B}$ examples from non-target classes. For output selection O, we measure the *success rate* which is equal to the percentage of $e \in O$ that belong to the target class. We compute the expected value of success rate for 1000 randomly sampled problems and report the mean and 95% confidence interval of the evaluation metric.

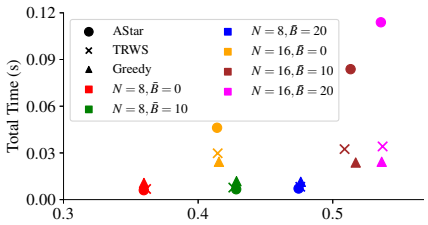

Figure 2. *Average runtime vs. accuracy of different inference algorithms on miniImageNet for $N \in \{8, 16\}$, $\bar{B} \in \{0, 10, 20\}$, and $B = 10$. Each setting is shown with a distinct color.*

We vary the number of bags as well as their sizes. We select the number of positive bags $N \in \{4, 8, 16\}$, the size of each positive bag $B \in \{5, 10\}$, and the size of negative bag $\bar{B} \in \{10, 20\}$. The number of classes M to sample from in each episode changes the difficulty of the task. We randomly choose M between 5 and 15 when $B = 5$, and between 10 and 20 when $B = 10$ for each problem.

The results in Table 1 show our method outperforms ATNMIL and SVM based approaches for all versions of the problem. To test the importance of learning the unary and pairwise potentials, we construct a baseline that uses cosine similarity to compute the relation between pairs[5] while keeping the rest of the algorithm identical. The performance gap between our method and the baseline shows that the relation learning method, apart from structured inference formulation, plays an important role in boosting the performance.

Average total (potentials computation and inference) runtime versus accuracy plot of different energy minimization methods on different settings is shown in Figure 2. Even on this small scale problem, the greedy optimization is faster on average while its accuracy is on par with other inference methods. See the supplementary material for complete numerical results.

5.3. Co-Localization

We evaluate on the co-localization problem to illustrate the benefits of the methods discussed in the paper on a real world and large scale dataset. In this task, we train the algorithm on a split of COCO 2017 [28] dataset with 63 seen classes and evaluate on the remaining 17 unseen classes.

[4]The code is publicly available here.

[5]We also use negative of Euclidean distance measure for the relation but it shows inferior performance.

	N	4		8		16	
	$\bar{B}$	10	20	10	20	10	20
Bag Size = 5	Ours	$\mathbf{63.83 \pm 1.49}$	$\mathbf{65.48 \pm 1.47}$	$\mathbf{72.49 \pm 0.98}$	$\mathbf{73.99 \pm 0.96}$	$\mathbf{78.60 \pm 0.64}$	$\mathbf{79.93 \pm 0.62}$
	Baseline	60.88 ± 1.51	63.83 ± 1.49	64.46 ± 1.05	68.08 ± 1.02	66.78 ± 0.73	70.39 ± 0.77
	MI-SVM [22]	56.25 ± 1.54	59.03 ± 1.52	62.75 ± 1.06	63.76 ± 1.05	67.91 ± 0.72	73.33 ± 0.69
	sbMIL [6]	54.55 ± 1.54	59.93 ± 1.52	58.25 ± 1.08	64.68 ± 1.05	61.35 ± 0.75	65.55 ± 0.74
	mi-SVM [2]	54.23 ± 1.54	59.43 ± 1.52	60.43 ± 1.07	66.08 ± 1.04	64.49 ± 0.74	69.69 ± 0.71
	ATNMIL [24]	50.35 ± 1.55	60.33 ± 1.52	56.05 ± 1.09	63.29 ± 1.06	58.97 ± 0.76	67.26 ± 0.73
Bag Size = 10	Ours	$\mathbf{37.42 \pm 1.50}$	38.50 ± 1.51	$\mathbf{42.85 \pm 1.08}$	$\mathbf{47.63 \pm 1.09}$	$\mathbf{51.70 \pm 0.77}$	$\mathbf{53.63 \pm 0.77}$
	Baseline	35.73 ± 1.49	$\mathbf{40.40 \pm 1.52}$	38.01 ± 1.06	43.95 ± 1.09	41.08 ± 0.76	47.83 ± 0.77
	MI-SVM [22]	29.53 ± 1.41	35.05 ± 1.48	35.25 ± 1.05	39.94 ± 1.07	41.21 ± 0.76	46.63 ± 0.77
	sbMIL [6]	31.55 ± 1.44	31.50 ± 1.44	34.10 ± 1.04	39.86 ± 1.07	28.80 ± 0.70	43.63 ± 0.77
	mi-SVM [2]	31.55 ± 1.44	35.33 ± 1.48	34.10 ± 1.04	39.86 ± 1.07	39.48 ± 0.76	45.16 ± 0.77
	ATNMIL [24]	26.58 ± 1.37	33.10 ± 1.46	28.48 ± 0.99	35.11 ± 1.05	31.56 ± 0.72	38.14 ± 0.75

Table 1. *Success rate on miniImageNet for different positive bags N, and total number of negative images $\bar{B}$. The first and the second part of the table show the results for bag size 5 and 10 respectively.*

Method	COCO	ImageNet
MI-SVM [22]	60.74 ± 1.07	49.44 ± 1.10
ATNMIL [24]	60.00 ± 1.07	49.35 ± 1.10
Ours	$\mathbf{65.34 \pm 1.04}$	$\mathbf{55.18 \pm 1.09}$
TRWS [26]	65.04 ± 1.05	54.20 ± 1.09
AStar [4]	64.99 ± 1.05	54.23 ± 1.09
Unary Only	59.24 ± 1.08	50.29 ± 1.10
TRWS Pairwise Only	64.53 ± 1.05	52.95 ± 1.09
AStar Pairwise Only	64.54 ± 1.05	52.89 ± 1.09
Ours Pairwise Only	64.65 ± 1.05	53.00 ± 1.10

Table 2. *CorLoc(%) on COCO and ImageNet with 8 positive and 8 negative images.*

The resulting dataset contains $111,085$ and $8,245$ images in the training and test set respectively. To evaluate the performance of the trained algorithm on a larger set of unseen classes we also test on validation set of ILSVRC2013 detection [38]. This dataset has originally 200 classes but only 148 classes do not have overlap with the classes that were used for training. The final dataset, after removing coco seen classes, contains $12,544$ images from 148 unseen classes. The dataset creation method is explained in the supplementary material in more detail.

For the CNN feature extractor module, we pre-train a Faster-RCNN detector [18] with ResNet-50 [20] backbone on the COCO training dataset which has only seen classes. For each image, region proposals with the highest objectness scores are kept. The output of the second stage feature extractor is used in all methods.

For this task, each bag is constructed by extracting top $B = 300$ region proposals from one image and a selection O represents one bounding box from each image. To select images of each problem, we first randomly select one class as the target. Then, N images which have at least one object from the target class are sampled as positive bags. The negative bag is composed of images which do not contain the target class. The success rate metric used in few-shot common object detection is used to evaluate the performance of different algorithms. A region proposal is considered successful if it has IoU overlap greater than 0.5 with the ground-truth target bounding box. Note that for the co-localization task, this metric is equivalent to class agnostic CorLoc [10] measure which is widely used for localization problem evaluation [46, 42, 5, 9].

Table 2 illustrates the quantitative results on COCO and ImageNet datasets with 8 positive and 8 negative images[6]. Our method works considerably better than other strong MIL baselines. Qualitative results of our method compared with other MIL based approaches are illustrated in Figure 3. Our method selects the correct object even when the target object is not salient. More qualitative results are presented in the supplementary material.

[6]We skip the results for sbMIL and mi-SVM as they showed similar or inferior results to MI-SVM.

To see the effect of unary and pairwise potentials separately, we provide results for two new variants for structured inference based methods: (i) Unary Only: where the common object proposal in each bag is selected using only the information in negative bags without seeing the elements in other bags, and (ii) Pairwise Only: where the negative bag information is ignored in each problem. The results show that the pairwise potentials contribute more to the final results. This is not surprising since negative images only help when they contain an object which is also appearing in positive images which, given the number of classes we are sampling from, has a low chance. Interestingly, by using the learned unary potentials alone we could get comparable results to the MIL baselines. The results in Table 2 show that different inference algorithms have very similar performance. However, as it is shown in Figure 4, the greedy optimization algorithm is much faster. Note that our method requires to compute only 15% out of all pairwise potentials in average. One may argue that the pairs can be forwarded on multiple GPUs in parallel and this reduces the forward time. However, our greedy inference method can also take advantage of multiple GPUs since the nodes at each level are data independent.

Method	No Unary	MEAN	MAX	SOFTMAX
Accuracy (%)	64.48 ± 1.47	70.23 ± 1.00	71.76 ± 0.99	$\mathbf{72.49 \pm 0.98}$

Table 3. *Comparison of different unary potential functions in miniImageNet experiment with $N = 8$, $B = 5$ and $\bar{B} = 10$.*

Method	Accuracy (%)
adaResNet [31]	56.88 ± 0.62
SNAIL [30]	55.71 ± 0.99
Gidaris et al. [17]	55.45 ± 0.89
TADAM [32]	58.50 ± 0.30
Qiao et al. [34]	$\mathbf{59.60 \pm 0.41}$
Ours-ReLU	56.43 ± 0.79
Ours-Gated	57.80 ± 0.77

Table 4. *5-way, 1-shot, classification accuracy with 95% confidence interval on miniImageNet test set.*

5.4. Ablation Study

In order to evaluate the effectiveness of our proposed unary potential function, we devise the following experiment. In the few-shot common object recognition task with $N = 8$ positive bags, $\bar{B} = 10$ negative images, and $B = 5$,

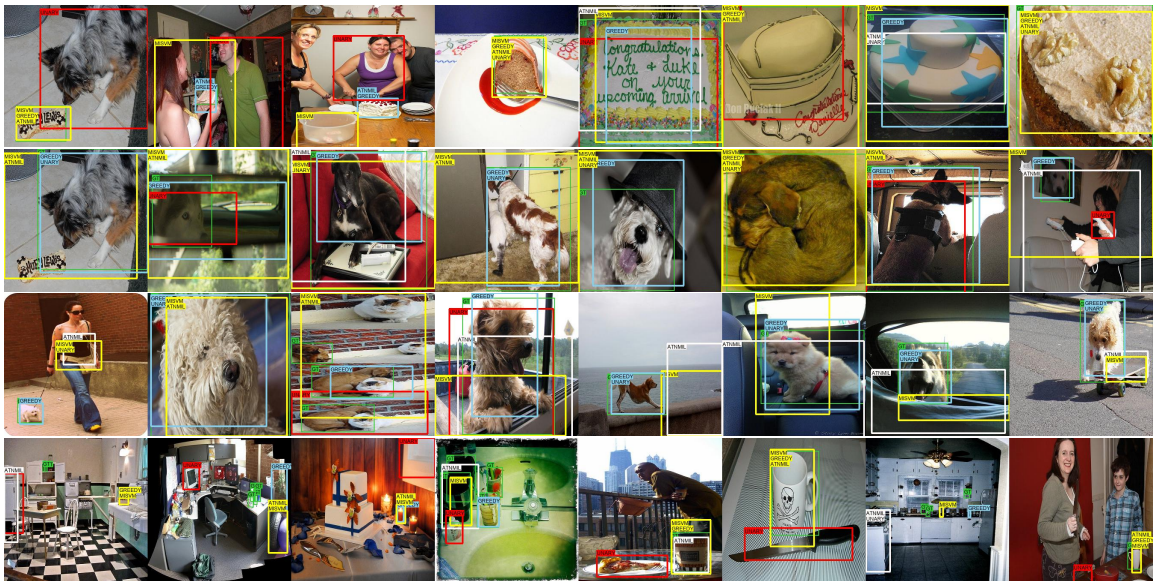

Figure 3. *Qualitative results on COCO dataset. Each row shows positive bags of a sampled collection. Negative bags are not shown. Note that the first image in the first two rows are identical but the target object is different. Last row shows a failure case of our algorithm. While cup is the target object, our method finds plant in the second image. This might be due to the fact that pot (which has visual similarities to cup) and plant are labelled as one class in the training dataset. Note that "dog", "cake" and "cup" are samples from unseen classes. Selected regions are tagged with method names. Ground-truth target bounding box is shown in green with tag "GT".*

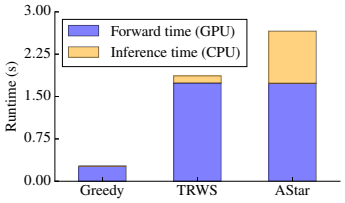

Figure 4. *Forward and inference time (in sec.) on COCO.*

we train the unary potentials with four different settings: (1) SOFTMAX: Unary potential function with the learned ν described in Section 4, (2) MAX: $\nu \to +\infty$, (3) MEAN: $\nu = 0$, and (4) No Unary: the model without using negative bag information. The pairwise potential function is kept identical in all the methods. The performance of our methods on the described settings are presented in Table 3. The results show the superiority of the learned weighted similarity to other strategies.

Next, we evaluate the quality of the learned pairwise relations $r(e, e')$ by using them for the task of one-shot image recognition [49] on *mini*ImageNet and compare it to the other state-of-the-art methods. In each episode of a one-shot 5-way problem, 5 classes are randomly chosen from the set of possible classes and one image is sampled from each class. This mini-training set is used to predict the label of a new query image which is sampled from one of the 5 classes. The performance is the accuracy of the method to predict the correct label averaged over many sampled episodes. All of these models are trained with a variant of deep residual networks [51, 20]. Note that unlike other methods, the model in [34] is trained on validation+training meta-sets.

At each episode, we use the learned relation function to score the similarity between the query image and all the images in the mini training set. The predicted label for the query image is simply the label of the image in mini training set which has the highest relation value to the query image. We compute the accuracy of the predictions of our pairwise potentials on test classes of *mini*ImageNet and compare it with current state-of-the-art few-shot methods in Table 4. We also provide comparison of gated activation function and a simplified ReLU activation in our architecture. Although our method is not trained directly for the task of one-shot learning, it achieves competitive results to the previous methods which are specifically trained for the task. Also, the results show the advantage of using gated activation over ReLU.

6. Conclusion

We introduce a method for learning to find images of a common object category across few bags of images which is constructed by learning unary and pairwise terms in an structured output prediction framework. Moreover, we propose an inference algorithm that uses the structure of the problem to solve the task at hand without requiring computation of all pairwise terms. Our experiments on two challenging tasks in the low data regime illustrate the advantage of our knowledge transfer method to several MIL weakly supervised algorithms. In addition, our inference algorithm performs comparable to the well-known structured inference algorithms for this task while being faster.

References

[1] Bjoern Andres, Thorsten Beier, and Joerg H. Kappes. OpenGM: A C++ library for discrete graphical models. *CoRR*, abs/1206.0111, 2012. 6

[2] Stuart Andrews, Ioannis Tsochantaridis, and Thomas Hofmann. Support vector machines for multiple-instance learning. In *NIPS*, pages 577–584, 2003. 2, 6, 7

[3] Boris Babenko, Ming-Hsuan Yang, and Serge Belongie. Visual tracking with online multiple instance learning. In *CVPR*, pages 983–990. IEEE, 2009. 1

[4] Martin Bergtholdt, Jörg Kappes, Stefan Schmidt, and Christoph Schnörr. A study of parts-based object class detection using complete graphs. *IJCV*, 87(1-2):93, 2010. 4, 6, 7

[5] Hakan Bilen, Marco Pedersoli, and Tinne Tuytelaars. Weakly supervised object detection with convex clustering. In *CVPR*, pages 1081–1089, 2015. 7

[6] Razvan C. Bunescu and Raymond J. Mooney. Multiple instance learning for sparse positive bags. In *ICML*, pages 105–112, 2007. 6, 7

[7] Marc-André Carbonneau, Veronika Cheplygina, Eric Granger, and Ghyslain Gagnon. Multiple instance learning: A survey of problem characteristics and applications. *Pattern Recognition*, 77:329–353, 2018. 2

[8] Kai Chen, Hang Song, Chen Change Loy, and Dahua Lin. Discover and learn new objects from documentaries. In *CVPR*, 2017. 2

[9] Ramazan Gokberk Cinbis, Jakob Verbeek, and Cordelia Schmid. Weakly supervised object localization with multi-fold multiple instance learning. *TPAMI*, 39(1):189–203, 2017. 7

[10] Thomas Deselaers, Bogdan Alexe, and Vittorio Ferrari. Localizing objects while learning their appearance. In *ECCV*, pages 452–466. Springer, 2010. 7

[11] Thomas Deselaers, Bogdan Alexe, and Vittorio Ferrari. Weakly supervised localization and learning with generic knowledge. *IJCV*, 100(3):275–293, 2012. 2, 6

[12] Thomas Deselaers and Vittorio Ferrari. A conditional random field for multiple-instance learning. In *ICML*, pages 287–294, 2010. 2, 6

[13] Gary Doran and Soumya Ray. A theoretical and empirical analysis of support vector machine methods for multiple-instance classification. *Machine Learning*, pages 79–102, 2014. 2, 6

[14] Alon Faktor and Michal Irani. Co-segmentation by composition. In *ICCV*, pages 1297–1304, 2013. 1, 2

[15] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*, 2017. 1, 2

[16] Huazhu Fu, Dong Xu, Bao Zhang, and Stephen Lin. Object-based multiple foreground video co-segmentation. In *CVPR*, pages 3166–3173, 2014. 1

[17] Spyros Gidaris and Nikos Komodakis. Dynamic few-shot visual learning without forgetting. In *CVPR*, pages 4367–4375, 2018. 7

[18] Ross Girshick. Fast r-cnn. In *ICCV*, pages 1440–1448, 2015. 7

[19] Hossein Hajimirsadeghi, Jinling Li, Greg Mori, Mohamed Zaki, and Tarek Sayed. Multiple instance learning by discriminative training of markov networks. In *UAI*, pages 262–271, 2013. 2

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 7, 8

[21] Dorit S Hochbaum and Vikas Singh. An efficient algorithm for co-segmentation. In *ICCV*, pages 269–276, 2009. 2

[22] Judy Hoffman, Deepak Pathak, Trevor Darrell, and Kate Saenko. Detector discovery in the wild: Joint multiple instance and representation learning. *CVPR*, pages 2883–2891, 2015. 2, 6, 7

[23] Kuang-Jui Hsu, Chung-Chi Tsai, Yen-Yu Lin, Xiaoning Qian, and Yung-Yu Chuang. Unsupervised cnn-based co-saliency detection with graphical optimization. In *ECCV*, 2018. 1, 2

[24] Maximilian Ilse, Jakub Tomczak, and Max Welling. Attention-based deep multiple instance learning. In *ICML*, pages 2127–2136, 2018. 1, 6, 7

[25] Zequn Jie, Yunchao Wei, Xiaojie Jin, Jiashi Feng, and Wei Liu. Deep self-taught learning for weakly supervised object localization. *CVPR*, pages 4294–4302, 2017. 2

[26] Vladimir Kolmogorov. Convergent tree-reweighted message passing for energy minimization. *TPAMI*, 28(10):1568–1583, 2006. 4, 6, 7

[27] Yao Li, Lingqiao Liu, Chunhua Shen, and Anton van den Hengel. Image co-localization by mimicking a good detectors confidence score distribution. In *ECCV*, pages 19–34, 2016. 2

[28] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, pages 740–755, 2014. 6

[29] Oded Maron and Tomás Lozano-Pérez. A framework for multiple-instance learning. In *NIPS*, pages 570–576, 1998. 1

[30] Nikhil Mishra, Mostafa Rohaninejad, Xi Chen, and Pieter Abbeel. A simple neural attentive meta-learner. In *ICLR*, 2018. 7

[31] Tsendsuren Munkhdalai, Xingdi Yuan, Soroush Mehri, and Adam Trischler. Rapid adaptation with conditionally shifted neurons. In *ICML*, pages 3661–3670, 2018. 7

[32] Boris N Oreshkin, Alexandre Lacoste, and Pau Rodriguez. Tadam: Task dependent adaptive metric for improved few-shot learning. *arXiv preprint arXiv:1805.10123*, 2018. 7

[33] Deepak Pathak, Evan Shelhamer, Jonathan Long, and Trevor Darrell. Fully convolutional multi-class multiple instance learning. In *ICLR Workshop*, 2015. 2

[34] Siyuan Qiao, Chenxi Liu, Wei Shen, and Alan L. Yuille. Few-shot image recognition by predicting parameters from activations. In *CVPR*, 2018. 7, 8

[35] Prajit Ramachandran, Barret Zoph, and Quoc V. Le. Searching for activation functions. Technical report, Google Brain, 2017. 4

[36] Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. In *ICLR*, 2017. 2, 6

[37] Mrigank Rochan and Yang Wang. Weakly supervised localization of novel objects using appearance transfer. In *CVPR*, pages 4315–4324, 2015. 2

[38] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge. *IJCV*, 115(3):211–252, 2015. 7

[39] Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. Meta-learning with memory-augmented neural networks. In *ICML*, pages 1842–1850, 2016. 1, 2

[40] Amirreza Shaban, Ching-An Cheng, Nathan Hatch, and Byron Boots. Truncated back-propagation for bilevel optimization. *AISTATS*, 2019. 1

[41] Yunhan Shen, Rongrong Ji, Shengchuan Zhang, Wangmeng Zuo, Yan Wang, and F Huang. Generative adversarial learning towards fast weakly supervised detection. In *CVPR*, pages 5764–5773, 2018. 2

[42] Miaojing Shi, Holger Caesar, and Vittorio Ferrari. Weakly supervised object localization using things and stuff transfer. *ICCV*, pages 3401–3410, 2017. 2, 7

[43] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *NIPS*, pages 4077–4087, 2017. 2

[44] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *CVPR*, 2018. 2, 3

[45] Kevin Tang, Armand Joulin, Li-Jia Li, and Li Fei-Fei. Co-localization in real-world images. In *CVPR*, pages 1464–1471, 2014. 2

[46] Jasper Uijlings, Stefan Popov, and Vittorio Ferrari. Revisiting knowledge transfer for training object class detectors. In *CVPR*, 2018. 2, 7

[47] Aaron van den Oord, Nal Kalchbrenner, Lasse Espeholt, Oriol Vinyals, Alex Graves, et al. Conditional image generation with pixelcnn decoders. In *NIPS*, pages 4790–4798, 2016. 4

[48] Sara Vicente, Carsten Rother, and Vladimir Kolmogorov. Object cosegmentation. In *CVPR*, pages 2217–2224, 2011. 1, 2, 6

[49] Oriol Vinyals, Charles Blundell, Tim Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *NIPS*, pages 3630–3638, 2016. 2, 6, 8

[50] Yair Weiss and William T Freeman. On the optimality of solutions of the max-product belief-propagation algorithm in arbitrary graphs. *IEEE Transactions on Information Theory*, 47(2):736–744, 2001. 4

[51] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *BMVC*, pages 87.1–87.12, 2016. 6, 8

[52] Dingwen Zhang, Junwei Han, Chao Li, and Jingdong Wang. Co-saliency detection via looking deep and wide. In *CVPR*, pages 2994–3002, 2015. 2

[53] Bohan Zhuang, Lingqiao Liu, Yao Li, Chunhua Shen, and Ian D Reid. Attend in groups: a weakly-supervised deep learning framework for learning from web data. In *CVPR*, 2017. 2