Learning Instance Segmentation by Interaction

Deepak Pathak*, Yide Shentu*, Dian Chen*, Pulkit Agrawal*,
Trevor Darrell, Sergey Levine, Jitendra Malik
University of California, Berkeley
{pathak,fredshentu,dianchen,pulkitag,trevor,svlevine,malik}@berkeley.edu

Abstract

We present an approach for building an active agent that learns to segment its visual observations into individual objects by interacting with its environment in a completely self-supervised manner. The agent uses its current segmentation model to infer pixels that constitute objects and refines the segmentation model by interacting with these pixels. The model learned from over 50K interactions generalizes to novel objects and backgrounds. To deal with noisy training signal for segmenting objects obtained by self-supervised interactions, we propose robust set loss. A dataset of robot's interactions along-with a few human labeled examples is provided as a benchmark for future research. We test the utility of the learned segmentation model by providing results on a downstream vision-based control task of rearranging multiple objects into target configurations from visual inputs alone. Videos, code, and robotic interaction dataset are available at https://pathak22.github.io/seg-by-interaction/.

1. Introduction

Objects are a fundamental component of visual perception. How are humans able to effortlessly reorganize their visual observations into a discrete set of objects is a question that has puzzled researchers for centuries. The Gestalt school of thought put forth the proposition that humans use similarity in color, texture and motion to group pixels into individual objects [1]. Various methods for object segmentation based on color and texture cues have been proposed [2–6]. These approaches are, however, known to over-segment multi-colored and textured objects.

The current state of the art overcomes these issues by making use of detailed class-specific segmentation annotations for a large number of objects in a massive dataset of web images [7–10]. A typical system first uses 1M human annotated Imagenet [11] images to pretrain a deep neural network. This network is then finetuned using over 700K object instances belonging to eighty semantic classes from the COCO dataset [12]. Such data is laborious and extremely time consuming to collect. Furthermore, current systems treat segmentation as an end goal, and do not provide a mechanism for correcting mistakes in downstream tasks. In contrast, one of the main challenges an active agent faces in the real world is adapting to previously unseen scenarios, where recovering from mistakes is critical to success.

Instead of treating segmentation as a passive process, in

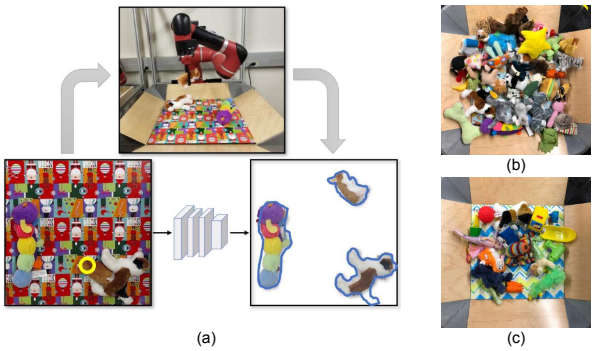

Figure 1. (a): Overview of our approach: a robotic agent conducts experiments in its environment to learn a model for segmenting its visual observation into individual object instances. Our agent maintains a belief about what groups of pixels might constitute an object and actively tests its belief by attempting to grasp this set of pixels (for e.g. attempts a grasp at the location shown by the yellow circle). Interaction with objects causes motion, whereas interaction with background results in no motion. This motion cue is utilized by the agent to train a deep neural network for segmenting objects. (b),(c): Visualization of the set of thirty six objects used for training (b) and sixteen objects used for testing (c). Validation objects can be seen in supp. materials. Separate sets of backgrounds were used for training, validation and testing.

*Denotes equal contribution

this work, our goal is to equip the learner with the ability to actively probe its environment and refine its segmentation model. One of the main findings in developmental psychology is that, very early on in development, infants have a notion of objects and they expect objects to move as wholes on connected paths, which in turn guides their perception of object boundaries [13, 14]. While at first entities merely separated by boundaries might all be the same for an infant, through interaction it is possible for the infant to learn about properties of individual entities and correlate these properties with visual appearance. For example, it is possible to learn that spherical objects roll, a smaller object can be contained inside a larger one, objects with rugged surfaces are harder to push etc. This progression of knowledge starting from delineating the visual space into discrete entities to learning about their detailed physical and material properties naturally paves the path for using this representation for control and eventually categorizing different segmented wholes into different "object classes."

In this work, we take the first step towards putting this developmental hypothesis to test and investigate if it is possible for an active agent to learn class agnostic instance segmentation of objects by starting off with two assumptions: (a) there are objects in the world; (b) principle of common fate [1], i.e. pixels that move together, group together. To that end, we set up an agent, shown in Figure 1, to interact with its environment and record the resulting RGB images. The agent maintains a belief about how images can be decomposed into objects, and actively tests its belief by attempting to grasp potential objects in the world. Through such self-supervised interaction, we show that it is possible to learn to segment novel objects kept on textured backgrounds into individual instances. We publicly release the collected data (i.e. over 50K interactions recorded from four different views) along with a set of 1700 human labelled images containing 9.3K object segments to serve as a benchmark for evaluating self-supervised, weakly supervised or unsupervised class agnostic instance segmentation method [1].

While interaction is a natural way for an agent to learn, it turns out that training signal for segmentation obtained via self-supervised interactions is very noisy as compared to object masks marked by human annotators. For example, in a single interaction, the agent might move two nearby objects, which would lead it to mistakenly think of these two objects as one. Dealing with such noise requires the training procedure to be robust, analogous to how in regression, we need to be robust to outliers in the data. However, direct application of pixel-wise robust loss is sub-optimal because we are interested in a set-level statistic such as the similarity between two sets of pixels (e.g. ground-truth and predicted masks) measured for instance using Jaccard index. Such a measurement depends on all the pixels and therefore requires one to define a robust loss over a set of pixels. In this work, we propose a technique, "robust set loss", to handle noisy segmentation training signal, with the general idea being that the segmenter is not required to predict exactly the pixels in the candidate object mask, rather that the predicted pixels as a set have a good Jaccard index overlap with the candidate mask. We show that robust set loss significantly improves segmentation performance and also reduces the variance in results.

We also demonstrate that the learned model of instance segmentation is useful for visuo-motor control by showing that our robot can successfully re-arrange objects kept on a table into a desired configuration using visual inputs alone. The utility of the learned segmentation method for control shows that it can guide further learning about properties of these segments in a manner similar to how human infants learn about physical and material object properties. An overview of our approach is shown in Figure 1.

2. Related Work

Our work draws upon the ideas from the active perception [15–18] to build a self-supervised object segmentation system. Closest to our work is [19] that makes use of optical flow to generate pseudo ground truth masks from passively observed videos. We discuss the similarities and differences from past work below.

Interactive Segmentation: Improving the result of segmentation by interaction has drawn a lot of interest [20–26]. However, most these works are concerned with using interaction to segment a specific scene. In contrast, our system uses interactions to actively gather supervision to train a segmentation system that can be used to segment objects in new images. The recent work on SE3 nets [27] learns to segment and model dynamics of rigid bodies in table-top environments containing boxes. As opposed to using depth data, we show object segmentation results from purely RGB images in visually more complex environment.

Self-Supervised Representation Learning: In this work, we use a self-supervised method to learn a model for segmentation. A number of recent works have studied self-supervision, using signals such as ego-motion [28, 29], audio [30], colorization [31], inpainting [32], context [33, 34], temporal continuity [35], temporal ordering [36], counting [37] and adversarial reconstruction [38] for learning visual features as an alternative to using human-provided labels. As far as we are aware, ours is the first work that aims to learn to segment objects using self-supervision from active robotic interaction.

Self-Supervised Robot Learning: Many recent papers

[1] Details at https://pathak22.github.io/seg-by-interaction/

Figure 2. Overview of experimental setup and method: (a) Sawyer robot's interactions with objects placed on an arena are recorded by four cameras. The arena is designed to allow easy modi cation of background texture. (b) From its visual observation (initial image) the robot hypothesizes what group of pixels constitute an object (intermediate segmentation hypothesis). It randomly chooses to interact with one such group by attempting to grasp and place it to a different location on the arena. If the grasped group indeed corresponds to an object, the mask of the object can be obtained by computing the difference image between the image after and before the interaction. The mask obtained from the difference image is used as pseudo ground truth for training a neural network to predict object segmentation masks. (c) Sometimes masks produced by this process are good (rst image), but they are often imperfect due to movement of multiple objects in the process of picking one object (second image) or creation of false masks due to lighting changes/shadows.

have investigated use of self-supervised learning for performing sensorimotor tasks. This includes self-supervised grasping [39–41], pushing [27, 42–44], navigation [45, 46] and rope-manipulation [45, 47]. However, the focus of these works was geared for an end task. Our goal is different – it is to a learn robust "segmentation" from noisy interaction signal. Such segmentation can be a building block for multiple robotic applications.

3. Experimental Setup

Our setup, shown in Figure 2, consists of a Sawyer robot interacting with objects kept on a at wooden arena. The arena is observed by four cameras placed at different locations around it. For diversifying the environment, we constructed the arena in a manner that the texture of the arena's surface could be easily modi ed. At any point in time, the arena contained 4 to 8 objects randomly sampled from a set of 36 training objects. We set up the agent to interact autonomously with objects without any human supervision. The agent made on average three interactions per minute using the *pick and place* primitive. We used the pick and place primitive as the primary mechanism for interaction as it leads to larger displacement of objects in comparison to say push actions and thereby provides more robust data for learning instance segmentation. We now describe each part of our methodology in detail.

Pick and Place Primitive: The pick action was parameterized by the location (a 2D point on the planar surface of the arena) and rotation of the agent's end effector (i.e., the gripper). The agent approached the pick location from the top with its gripper perpendicular to the arena, rotated by the desired angle and kept wide open. At the pick location, the gripper was closed by a pre- xed amount to grasp the object (if any). After the grasp, the gripper was moved to the place location and opened. If the gripper held an object, the place action caused the object to drop on the arena. This pick and place motion of the robot was enabled by calibrating the robot's internal coordinate system with the arena kept in front of it using Kinect sensing.

During the process of pick and place, three images (size 350x430 pixels) of the arena were captured: I_t before the pick action, I^0_{t+1} when the grasped object is picked but not placed on the arena and I_{t+1} taken after placing the object. All images were captured by positioning the agent's arm in a manner that did not obstruct the view of the arena from any of the four cameras. Note that every pick and place action did not lead to displacement of an object because: (a) either the pick operation was attempted at a location where no object was present (and in this case, I_t, I^0_{t+1}; I_{t+1} had objects in the same con guration) or (b) the grasping failed and the object was not picked, but was possibly displaced due to contact with the gripper. In case of (b), I_t and I^0_{t+1} typically had objects in slightly different configurations whereas I^0_{t+1} and I_{t+1} had objects in the same con guration.

Interaction Procedure: Let the agent's current observation be I_t and its belief about group of pixels that constitute an object be $\{s^t_1; s^t_2 :: s^t_K\}$, where s^t_i (a binary mask) indicates the set of pixels that belong to the ith group among a total of K groups. The agent interacts to verify if s^t_j ($j \in [1; K]$) constitutes an object by attempting to pick and then place s^t_j at a randomly chosen location on the arena. If pixels in s^t_j move, it con rms the agent's belief that s^t_j is an actual object. Otherwise, the agent revises its belief to the

Algorithm 1: Segmentation by Interaction

```
Pre-train network with passive unsupervised data
for iteration t = 1 to T do
    Record current observation I_t
    Generate object hypotheses {s_1^t, ..., s_K^t} = CNN(I_t)
    Randomly choose one hypothesis s_j^t ∈ {s_1^t, ..., s_K^t}
    Interact with hypothesized object: move(s_j^t)
    Record observation I_{t+1}
    mask = frame_difference(I_t, I_{t+1})
    if mask is empty then
        {(x,y), mask, I_t} is negative training example
    else
        {(x,y), mask, I_t} is positive training example
    end
    if t % update_interval == 0 then
        Update CNN using positive/negative examples
    end
end
```

contrary. Note that our goal is to show that we can obtain good instance segmentation by interaction, and hence we use standard motion planning procedure [48, 49] to simply hard-code the interaction pipeline. Details are in the supplementary.

Since objects were only moved by agent's interactions, the collected data is expected to be highly correlated. To prevent such correlations, the agent executed fully automatic reset after every 25 interactions without any human intervention. In such a reset procedure, the agent moved its gripper from eight different points uniformly sampled on the boundaries of the arena to the center to randomly displace objects. To further safeguard against correlations, the background was periodically changed. Overall, the agent performed more than 50,000 interactions.

4. Instance Segmentation by Interaction

The primary goal of this work is to investigate if it is possible for an active learner to separate its visual inputs into individual foreground objects (i.e., obtain instances) and background by self-supervised active interaction instead of human supervision. Broadly the agent moves hypothesized objects and this motion is used to generate (pseudo ground truth) object masks that are used to supervise learning of the segmentation model.

The major challenge in training a model with such self-generated masks is that they are far from perfect (Figure 2). Typical error modes include: (a) false negatives due to complete failure to grasp an object; (b) failure in grasping that slightly perturb the object resulting in incomplete masks; (c) in case two objects are located near each other, picking one object moves the other one, resulting in masks that span multiple objects; (d) erroneous masks due to variation in lighting, shadows and other nuisance factors. Any method attempting to learn object segmentation from interaction must deal with such imperfections in the self-generated pseudo ground truth masks.

When near-perfect human annotated masks are available, it is possible to directly optimize per-pixel loss determining whether the pixel belongs to background or foreground. With noisy masks it is desirable to optimize a robust loss function that only forces the predictions to approximately match the noisy ground truth. Since noise in segmentation masks is a global property of the image, it is non-trivial to employ pixel-wise robust loss. We discuss this challenge in more detail and a solution to it by proposing Robust Set Loss in section 4.2.

While there are many methods in the literature for making use of object masks for training instance segmentation systems, without any loss of generality in this work we use the state-of-art method known as DeepMask [8] to train a deep convolution neural network initialized with random weights (i.e. from scratch). Note that the use of this method for training CNN is complementary to our contribution of learning from active interaction and dealing with challenges of noisy training signal using robust set loss. The DeepMask framework produces class agnostic instance segmentation with the help of two sub-modules: a scoring network and a mask network. The basic idea is to scan image patches at multiple scales using the sliding window approach, and each patch is evaluated by the scoring network to determine whether the center pixel of the patch is part of foreground or background. If the center pixel of the image crop belongs to the foreground (i.e., the patch is believed to contain an object), it is passed into the mask network to output the segmentation mask. We use the active interaction data generated by the agent to train the mask and scoring networks.

4.1. Training Procedure

The training procedure is summarized in Algorithm 1. Let the current image observed by the agent be I_t. The image is first re-sized into seven different scales given by $2^{i/0.25 - 1.25}, i \in [0, 6]$. For each scale, the output of scoring network is computed for image patches of size 192x192 extracted at a stride of 16. All patches that are predicted by scoring network to contain object segment/masks $\{s_1^t, s_2^t, ..., s_K^t\}$.

The agent randomly decides to interact with one these object segment hypotheses (say s_j^t) using the pick and place primitive described in section 3. For ascertaining s_j^t if indeed corresponds to the object, we compute the difference image $d_t^0 = |I_{t+1}^0 - I_t|$. For increasing robustness to noise we only compute the difference in a square region of size

240x240 pixels around the point where robot attempted the pick action. Additional computations to increase robustness are described in the supplementary materials.

From the difference image, we extract a single mask of connected pixels (say m_j^t). If the number of non-zero pixels in this mask are greater than 1000, we regard the pick interaction to have found an object (i.e., the image patch s_j^t is considered to be a positive example for the scoring network). The correspond mask m_j^t is used as training data point for the mask network. Otherwise, we regard the s_j^t to be a part of the background (i.e., negative example for the scoring network). We generate additional training data by repeating the same process for image patches I_{t-1}^0, and I_{t+1} (see section 3).

Furthermore, to account for variance in object sizes, we augment the positive data points by randomly scaling images in the range of $2^{[-0.25, 2^{0.25}]}$ and obtain hard negatives by jittering the positive image patches by more than 64 pixels in L1 distance (i.e. combined jittering along x and y axes) and randomly jitter negative examples for data augmentation. We used a neural network with a ResNet-18 architecture to first extract a feature representation of the image. This feature representation is fed into two branches that predict the score and the mask each. We use a batch size of 32 and SGD with momentum for training.

4.2. Robust Set Loss

The masks computed by the agent's interaction are quite noisy to train the mask network using the standard cross entropy loss that forces the prediction to exactly match the noise in each training data point. Attempting to fit noise is adversarial for the learning process, as (a) overfitting to noise would hamper the ability to generalize to unseen examples, and (b) inability to fit noise would increase variance in the gradients and thereby make training unstable.

The principled approach of learning with noisy training data is to use a robust loss function for mitigating the effect of outliers. Robust loss functions have been extensively studied in statistics, in particular, Huber loss [50] applied to regression problems. However, such ideas have mostly been explored in the context of regression and classification for modeling independent outputs. Unfortunately, segmentation mask is a "set of pixels", where a statistic of interest such as the similarity between two sets of pixels (e.g., ground-truth and predicted masks) measured for instance using Jaccard Index (i.e., intersection over union (IOU)) depends on all the pixels. The dependence of the statistic on a set of pixels makes it non-trivial to generalize ideas such as Huber loss in a straightforward manner. We formulate Robust Set Loss to deal with "set-level" noise.

Before discussing the formulation, we describe the intuition behind our formulation using segmentation as an example. Our main insight is that, if the target segmentation mask is noisy, it is not desirable to force the per-pixel output of the model to exactly match the noisy target. Instead, we would like to impose a soft constraint for only matching a subset of target pixels while ensuring that some (potentially non-differentiable) metric of interest, such as IOU, between the prediction and the noisy target is greater than or equal to a certain threshold. In case the threshold is 1, it reduces to exactly fitting the target mask. If the threshold is less than 1, it amounts to allowing a margin between the predicted and the target mask.

The hope is that we can infer the actual (latent) ground-truth masks by only matching the network's prediction with the noisy target up to a margin measured by a metric of interest such as the IOU. Because the network parameters are optimized across multiple training examples, it is possible that the network will learn to ignore the noise (as it is hard to model) and predict the pattern that is common across examples and therefore easier to learn. The pattern "common" across examples is likely to correspond to the actual ground truth. We operationalize this idea behind Robust Set Loss (RSL) via a constrained optimization formulation over the output of the network and the noisy target mask.

Consider the pixel-wise labeling of an image I as a "set" of random variables $X = \{x_0, \ldots, x_n\}$ where $x_i \in L$, where n is total number of pixels and $L = \{0, 1\}$ is the set of possible labels that can be assigned to a pixel. Let the latent ground truth label corresponding to the image I be $P(X|I)$ and the noisy mask collected by interaction be $M_I = f(P(X|I))$, where f is an arbitrary non-linear function. Let the predicted mask be $Q(X|\theta, I)$, where θ are the parameters of the neural network. We want to minimize the distance between the prediction and latent ground truth measured using KL-divergence $D_{KL}(P(X)\|Q(X|\theta)) = \mathbb{E}_{X \sim P}[\log Q(X|\theta)]$. Assuming, the latent target mask is discrete, $P(X = \hat{X}) = \sum_i \mathbb{1}_{x_i = \hat{x}_i}$, where $\hat{X}$ is the prediction.

Given the network output $Q(X|\theta, I)$ and the noisy label set (i.e., mask collected by interaction) M_I, the goal is to optimize for the latent target $P(X|I)$ which is within a desired margin from the noisy mask M_I. The network prediction $Q(X|\theta, I)$ will then be trained to match $P(X|I)$.

We assume that the latent target is a mask with values in the label set L. Hence, we model it as a delta function $P(X) = [X = \hat{X}] = \sum_i \mathbb{1}_{x_i = \hat{x}_i}$. The distance of this latent target from the predicted distribution is measured via KL-divergence which in case of delta function reduces to $D_{KL}(P(X)\|Q(X|\theta)) = \mathbb{E}_{X \sim P}[\log Q(X|\theta)] = \sum_i \log q_i(\hat{x}_i)$. The final optimization problem is formu-

lated as follows:

$$\underset{;\hat{X};}{\text{minimize}} \quad \sum_i \log q_i(\hat{x}_i) + \quad {}^T$$

$$\text{subject to} \quad \text{IoU}(\hat{X}; M_I) \quad b \quad ; \quad 0 \qquad (1)$$

where is the slack variable. We optimize the above objective by approximate discrete optimization [51] in each iteration of training. Details of the optimization procedure are in the supplementary material. The approximate discrete optimization is fast and takes approximately 0.35 seconds for a batch of 32 examples.

Note that our formulation could be thought of as a generalization of the CCNN constrained formulation proposed in Pathak et. al. [52] with several key differences: (a) we handle non-linear, non-differentiable constraints compared to only linear ones in CCNN, (b) we propose a discrete formulation compared to a continous one in CNN, and (c) our main goal is to handle robustness in set data while CCNN's goal is to learn a pixel-wise ground truth from image level tags in weakly supervised segmentation setting. (d) Moreover, our optimization procedure is an approximate discrete solver while CCNN used projected gradient descent, which would be impractical with Jaccard Index like constraints.

4.3. Bootstrapping the Learning Process Using Passive Self-Supervision

Without any prior knowledge, the agent's initial beliefs about objects will be arbitrary, causing it to spend most of its time interacting with the background. This process would be very inef cient. We address this issue by assuming that initially our agent can passively observe objects moving in its environment. For this purpose we use a prior robotic pushing dataset [42] that was constructed by a robot randomly pushing objects in a tabletop environment. We apply the method of [19] to automatically extract masks from this data, which we use to pre-train our ResNet-18 network (initialized with random-weights). Note that this method of pre-training is completely self-supervised and does not rely on any human annotation, and it is quite natural to combine passive observation and active interaction for self-supervised learning. This initial model gives around 14% AP at IU of 0.3, while our nal model with robust set loss gives mean AP of 45.9%.

5. Baselines and Comparisons

We compare the performance of our method against a state-of-the-art bottom up segmentation method called Geodesic Object Proposals (GOP) [5], and a top-down instance segmentation method called DeepMask [8] which is pre-trained on 1M ImageNet and then netuned in a class agnostic manner using over 700K strongly supervised masks obtained from the COCO dataset. We incorporated NMS (non-max suppression) into DeepMask, which significantly boosted its performance on our dataset. In order to reduce the bias from the domain shift of transferring from web images to images recorded by the robot, we further removed very large masks that could not possibly correspond to individual objects from the outputs of both methods. Even after these modi cations, we found GOP to output proposals that corresponded to other smaller parts of the arena such as the corners. We explicitly removed these proposals and dubbed this method as GOP-Tuned.

Method	Supervision	AP at IU 0.3	AP at IU 0.5
GOP	Bottom up	10.9	04.1
GOP (tuned)	Bottom up	23.6	16.3
DeepMask	Strong Sup.	44.5	34.3
DeepMask (tuned)	Strong Sup.	61.8	47.3
Ours + Human	Semi-sup.	43.1 2.6	21.1 2.6
Ours	Self-sup.	41.1 2.4	16.0 2.6
Ours + Robust Set Loss	Self-sup.	45.9 2.1	22.5 1.3

Table 1. Quantitative comparison of our method with bottom-up (GOP [5]), learned top-down (DeepMask [8]) segmentation methods and optimization without robust set loss on the full test set. We report the mean and standard deviation for our approach. Note that our approach signi cantly outperforms GOP, but is outperformed by DeepMask that uses strong manual supervision of 700K+ COCO segments and 1M ImageNet images. Adding 1470 images (contains 7946 object instances) with clean segmentation masks labeled by humans improves performance of our base system. The robust set loss not only improves the mean performance over normal cross-entropy loss but also decreases the variance by handling noise across examples.

Training/Validation/Test Sets: We used 24 backgrounds for training, 6 for validation and 10 for testing. We used 36 different objects for training, 8 for validation and 15 for testing. The validation set consisted of 30 images (5 images per background), and the test set included 200 images (20 images per background). We manually annotated object masks in these images for the purpose of evaluation. In addition, we also provide instance segmentation masks for 1470 training images containing 7946 objects to promote research directions looking at combining small quantities of high-quality annotations along with larger amounts of potentially noisy data collected via self-supervision.

Metric: The performance of different segmentation systems is quanti ed using the standard mean average-precision (mAP; [53]) metric at different IoU (intersection over union) thresholds. Intuitively, the mAP at an IoU threshold ofth counts generated proposals with an IoU < th with the ground truth as a false positive and penalizes the system for each ground truth proposal that is not matched by a predicted proposal with IoU > th . Higher mAP indicates better performance.

(a) Performance vs. Interactions (b) Successes vs. Interactions (c) Precision vs. Recall

Figure 3. Quantitative evaluation of the segmentation model on the held-out test. (a) The performance of our system measured as mAP at IoU of 0.3 steadily increases with the amount of data. After 50K iterations our system signi cantly beats GOP tuned with domain knowledge (i.e. GOP-Tuned; section 5). (b) The ef cacy of experimentation performed by the robot is computed as the recall of ground truth objects that have IoU of more than 0.3 with the group of pixels that the robot believes to be objects. The steady increase in recall at different precision threshold shows that the robot learns to perform more ef cient experiments with time. (c) Precision-Recall curves re-con rm the results.

6. Results and Evaluations

We compare the performance of our system against GOP and DeepMask using the AP at IoU 0.3 metric on the held-out testing set as shown in Figure 3(a). Our system significantly outperforms GOP even when it is tuned with domain knowledge (GOP-tuned), is superior to DeepMask trained with strong human supervision, but is outperformed when non-max supression (NMS) thresholds and other domain speci c tunings are applied to DeepMask outputs (i.e. DeepMask-tuned). These results are re-con rmed by the precision-recall curves shown in Figure 3(c). These results indicate that our approach is able to easily outperform methods relying on hand-engineered bottom-up segmentation cues, but there is still a substantial way to go before matching the performance of a system trained using strong human-supervision (i.e. DeepMask). However, it is encouraging to see from Figure 3(a) that the performance of our system is steadily increasing with the amount of noisy self-supervised data collected via interactions.

Results in table 1 further reveal that adding a few images (i.e. 1470 images containing 7946 object instances) with clean segmentation masks during training (Ours + Human) helps improve the performance over our base system possibly due to reduction in noise in training signal. Finally, the robust set loss signi cantly improves performance.

While the curves in Figure 3(a) show an overall increase in performance with increasing amounts of interaction data, there are few intermediate downward de ections. This is not surprising in an active learning system because the data distribution encountered the agent is continuously changing. In our speci c case, when backgrounds that are signi cantly different from the existing training backgrounds are introduced, the existing model has potentially over- t to previous training backgrounds and the overall performance of the system dips. As the agent interacts in its new environment, it adapts its model and the performance eventually recovers and improves beyond what it was prior to the introduction of the new background. Note that passive learning systems also encounter easy and hard examples, but because the training is batched, in contrast to an active system these examples are uniformly sampled throughout the course of training and therefore such upward/downward uctuations in performance with increasing amount of data are almost never seen in practice.

Figure 4. The progression of segmentation masks produced by the method as number of experiments conducted by the report increase on held-out test dataset (from left to right). The number of false positives reduce and the quality of masks improve.

Qualitative Comparison Visualization of instance segmentation output of various methods on the test set in Figure 5 shows that our method generalizes and can segment novel objects on novel backgrounds. Qualitatively, our method performs similarly to DeepMask, despite receiving no human supervision. The performance of GOP is significantly worse due to low recall. While in most cases our method produces a connected segmentation mask, in some cases it produces a mask with disconnected jittered pixels (e.g., row-4 in Figure 5). The improvement in the quality of segmentation with agent's experience is visualized in Figure 4. Spurious segmentation of background reduces over time and the recall of objects increases.

Figure 5. Visualization of the object segmentation masks predicted by a bottom up segmentation method GOP with Domain Knowledge (c; GOP tuned), a top down segmentation method trained with strong supervision using more 700K human annotated ground truth object masks (d; DeepMask) and our method (e). The top three rows show representative examples of cases when our methods predicts good masks and the bottom two rows illustrate the failure mode of our method. In general GOP has low recall. The performance of our method and DeepMask is general. Our dominant failure mode is prediction of small disconnected masks (row 4).

6.1. Active Interactions v/s Passive Data Collection

An active agent continuously improves its model and therefore not only collects more data, but also higher quality data with time. This suggests that an active agent might re-quire fewer data points than a passive agent for learning. In case of object segmentation, higher quality of data collected by an agent would be re ected by generation of object hy-pothesis that have higher recalls at lower false positive rates. We tested if the quality of data generated by active agent im-proves over time by computing the recall of the ground truth objects using object hypothesis generated by our agent in novel environments at different precision thresholds. Fig-ure 3(b) shows that the recall increases over time indicat-ing that our agent learns to perform better experiments with time on the held-out backgrounds and objects.

6.2. Analyzing Generalization

Previous results have shown that the performance of our system increases with amount of data. A natural question to ask is, what kind of data would be more useful for agent to learn a segmentation model that will generalize better. In order to answer this question, we investigated whether our system generalized better to new objects or to new back-grounds. For our investigation we constructed four sets of images: (A) training objects on training backgrounds; (B) training objects on test backgrounds; (C) test objects on training backgrounds; and (D) test objects on test back-grounds. If our system generalizes to objects better than background, then changing from training to test objects (but keeping the training backgrounds) should lead to a smaller drop in performance as compared to changing from training

Figure 6. Fine-grained generalization analysis of our model. The y-axis denotes the drop in performance (i.e., lower the better) due to the change scenarios speci ed on the x-axis. When the performance is measured at IoU 0.3, the generalization of our model is better to novel objects as compared to novel backgrounds. However, when the quality of masks is more heavily penalized (i.e. IoU 0.5) the generalization is better to backgrounds as compared to novel objects.

to test backgrounds (but keeping the training objects).

When mAP is computed at IoU threshold of 0.3, we nd this indeed to be the case. However, when mAP is computed at a threshold of 0.5 we nd the reverse trend to hold true. These results suggest that if the quality of mask is not critical (i.e. IoU of 0.3 is suf cient), using larger number of backgrounds is likely to result in better generalization. Alternatively, in use cases where the mask quality is critical, (i.e. IoU of 0.5) using a larger set of objects is likely to result in better generalization.

6.3. Using Segmentation for Downstream Tasks

Until now, we have shown results on object segmentation. Next we evaluated if the segmentation returned by our system could be used for downstream tasks by perception or control systems. Towards this end we evaluated performance on the task rearranging objects kept on a table into a desired con guration.

6.3.1 Object Rearrangement

We tasked the system to rearrange the objects in its current visual observation into the con guration of objects show in a target image. We did not provide any other information to the agent. 1-3 objects were displaced between the current observation of the agent and the target image. Our robot is equipped with a pick and place primitive as described in section 3. If the quality of segmentation is good, it should be possible to match the objects between current and target image and use the pick/place primitive to displace these objects. Since our goal in this work is not to evaluate the matching system, we use off the shelf features extracted from AlexNet trained for classifying Imagenet images.

Our overall pipeline looks as following: (a) obtain a list of object segments produced by our method from current and target image; (b) crop a tight window around the object segments from the original image and pass it into AlexNet to compute feature representation per segment; (c) match the segments between current and target image to determine to what locations the objects in current should be moved to; (d) use the pick/place primitive to move the match objects one by one until the matched objects are within 15 pixels of each other. The robot is allowed a maximum of ten interactions. Qualitative results depicting the performance our system at the rearrangement task are shown in Figure 7. While our system is successful sometimes, it fails at many occasions. However most of these failures are a result of failures in feature matching or object grasping. Our system outperforms the inverse model method of [42] for rearranging objects that doesnot make use of explicit instance segmentation.

Figure 7. The agent was tasked to displace objects in the initial image to their con guration shown in the target image. The nal image obtained after manipulation performed by our system is shown in the middle column. Our system made use of the object segmentation learned using active interaction and a hand-designed controller described in section 6.3.1 to perform this task. The majority of failures in our system were due to failures in feature matching or object grasping.

7. Discussion

In this work, we presented a method for using active self-supervision to reorganize visual inputs into object instances. The performance of our system is likely to bene t from obtaining better pseudo ground truth masks by the use of better grasping techniques, use of other interaction primitives and joint learning of perceptual and control systems where the interaction mechanism also improves with time.

To build general purpose sensorimotor learning systems, it is critical to find ways to transfer knowledge across tasks. While one approach is to come up with better algorithms for transfer learning, the other is to make use of more structured representations of sensory data than obtained using vanilla feed-forward neural networks. This work, builds upon the second view, in proposing a method for segmenting an image into objects in the hope that object-centric representations might be an important aspect of future visuo-motor control systems. Our system is only the first step towards the grander goal of creating agents that can self-improve and continuously learn about their environment.

Acknowledgments

We would like to thank members of BAIR community for fruitful discussions. This work was supported in part by ONR MURI N00014-14-1-0671; DARPA; NSF Award IIS-1212798; Berkeley DeepDrive, Valrhona Reinforcement Learning Fellowship and an equipment grant from NVIDIA. DP is supported by the Facebook graduate fellowship.

References

[1] Wertheimer, M.: Laws of organization in perceptual forms. (1938) 1, 2

[2] Felzenszwalb, P.F., Huttenlocher, D.P.: Efficient graph-based image segmentation. IJCV (2004) 1

[3] Carreira, J., Sminchisescu, C.: Cpmc: Automatic object segmentation using constrained parametric min-cuts. PAMI (2012) 1

[4] Arbeláez, P., Pont-Tuset, J., Barron, J.T., Marques, F., Malik, J.: Multiscale combinatorial grouping. In: CVPR. (2014) 1

[5] Krähenbühl, P., Koltun, V.: Geodesic object proposals. In: ECCV. (2014) 1, 6

[6] Isola, P., Zoran, D., Krishnan, D., Adelson, E.H.: Crisp boundary detection using pointwise mutual information. In: European Conference on Computer Vision, Springer (2014) 799–814 1

[7] He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. arXiv preprint arXiv:1703.06870 (2017) 1

[8] Pinheiro, P.O., Collobert, R., Dollár, P.: Learning to segment object candidates. In: Advances in Neural Information Processing Systems. (2015) 1990–1998 1, 4, 6, 12

[9] Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. CVPR (2015) 1

[10] Hariharan, B., Arbeláez, P., Girshick, R., Malik, J.: Simultaneous detection and segmentation. In: ECCV. (2014) 1

[11] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: ImageNet Large Scale Visual Recognition Challenge. IJCV (2015) 1

[12] Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft COCO: Common objects in context. ECCV (2014) 1

[13] Spelke, E.S.: Principles of object perception. Cognitive science (1990) 2

[14] Spelke, E.S., Kinzler, K.D.: Core knowledge. Developmental science 10(1) (2007) 89–96 2

[15] Gibson, J.J.: The ecological approach to visual perception. Psychology Press (1979) 2

[16] Bajcsy, R.: Active perception. Proceedings of the IEEE 76(8) (1988) 966–1005 2

[17] Aloimonos, J., Weiss, I., Bandyopadhyay, A.: Active vision. International journal of computer vision 1(4) (1988) 333–356 2

[18] Bajcsy, R., Aloimonos, Y., Tsotsos, J.K.: Revisiting active perception. Autonomous Robots (2016) 1–20 2

[19] Pathak, D., Girshick, R., Dollár, P., Darrell, T., Hariharan, B.: Learning features by watching objects move. CVPR (2017) 2, 6

[20] Fitzpatrick, P.: First contact: an active vision approach to segmentation. In: IROS. (2003) 2

[21] Kenney, J., Buckley, T., Brock, O.: Interactive segmentation for manipulation in unstructured environments. In: ICRA. (2009) 2

[22] Hausman, K., Pangercic, D., Márton, Z.C., Bálint-Benczédi, F., Bersch, C., Gupta, M., Sukhatme, G., Beetz, M.: Interactive segmentation of textured and textureless objects. In: Handling Uncertainty and Networked Structure in Robot Control. Springer (2015) 2

[23] Van Hoof, H., Kroemer, O., Peters, J.: Probabilistic segmentation and targeted exploration of objects in cluttered environments. IEEE Transactions on Robotics 30(5) (2014) 1198–1209 2

[24] Pajarinen, J., Kyrki, V.: Decision making under uncertain segmentations. In: Robotics and Automation (ICRA), 2015 IEEE International Conference on, IEEE (2015) 1303–1309 2

[25] Björkman, M., Kragic, D.: Active 3d scene segmentation and detection of unknown objects. In: ICRA. (2010) 2

[26] Nalpantidis, L., Björkman, M., Kragic, D.: Yes-yet another object segmentation: exploiting camera movement. In: IROS. (2012) 2

[27] Byravan, A., Fox, D.: Se3-nets: Learning rigid body motion using deep neural networks. In: ICRA. (2017) 2, 3

[28] Agrawal, P., Carreira, J., Malik, J.: Learning to see by moving. ICCV (2015) 2

[29] Jayaraman, D., Grauman, K.: Learning image representations tied to ego-motion. ICCV (2015) 2

[30] Owens, A., Wu, J., McDermott, J.H., Freeman, W.T., Torralba, A.: Ambient sound provides supervision for visual learning. ECCV (2016) 2

[31] Zhang, R., Isola, P., Efros, A.A.: Colorful Image Colorization. ECCV (2016) 2

[32] Pathak, D., Krähenbühl, P., Donahue, J., Darrell, T., Efros, A.: Context Encoders: Feature Learning by Inpainting. CVPR (2016) 2

[33] Doersch, C., Gupta, A., Efros, A.A.: Context as supervisory signal: Discovering objects with predictable context. ECCV (2014) 2

[34] Noroozi, M., Favaro, P.: Unsupervised Learning of Visual Representations by Solving Jigsaw Puzzles. ECCV (2016) 2

[35] Wang, X., Gupta, A.: Unsupervised learning of visual representations using videos. ICCV (2015) 2

[36] Misra, I., Zitnick, C.L., Hebert, M.: Shuf e and Learn: Unsupervised Learning using Temporal Order Veri cation. ECCV (2016) 2

[37] Noroozi, M., Pirsiavash, H., Favaro, P.: Representation learning by learning to count. arXiv preprint arXiv:1708.06734 (2017) 2

[38] Donahue, J., Krähenbühl, P., Darrell, T.: Adversarial Feature Learning. ICLR (2017) 2

[39] Pinto, L., Gupta, A.: Supersizing self-supervision: Learning to grasp from 50k tries and 700 robot hours. ICRA (2016) 3

[40] Levine, S., Pastor, P., Krizhevsky, A., Quillen, D.: Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection. arXiv:1603.02199 (2016) 3

[41] Mahler, J., Liang, J., Niyaz, S., Laskey, M., Doan, R., Liu, X., Ojea, J.A., Goldberg, K.: Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics. arXiv preprint arXiv:1703.09312 (2017) 3

[42] Agrawal, P., Nair, A., Abbeel, P., Malik, J., Levine, S.: Learning to poke by poking: Experiential learning of intuitive physics. NIPS (2016) 3, 6, 9

[43] Finn, C., Levine, S.: Deep visual foresight for planning robot motion. In: Robotics and Automation (ICRA), 2017 IEEE International Conference on, IEEE (2017) 2786–2793 3

[44] Pinto, L., Gandhi, D., Han, Y., Park, Y.L., Gupta, A.: The curious robot: Learning visual representations via physical interactions. In: ECCV. (2016) 3

[45] Pathak, D., Mahmoudieh, P., Luo, G., Agrawal, P., Chen, D., Shentu, Y., Shelhamer, E., Malik, J., Efros, A.A., Darrell, T.: Zero-shot visual imitation. In: ICLR. (2018) 3

[46] Gandhi, D., Pinto, L., Gupta, A.: Learning to y by crashing. arXiv preprint arXiv:1704.05588 (2017) 3

[47] Nair, A., Chen, D., Agrawal, P., Isola, P., Abbeel, P., Malik, J., Levine, S.: Combining self-supervised learning and imitation for vision-based rope manipulation. arXiv preprint arXiv:1703.02018 (2017) 3

[48] Cowley, A., Cohen, B., Marshall, W., Taylor, C.J., Likhachev, M.: Perception and motion planning for pick-and-place of dynamic objects. In: IROS. (2013) 4

[49] Gong, H., Sim, J., Likhachev, M., Shi, J.: Multi-hypothesis motion planning for visual object tracking. In: ICCV. (2011) 4

[50] Huber, P.J.: Robust estimation of a location parameter. The annals of mathematical statistics (1964) 5

[51] Papandreou, G., Chen, L.C., Murphy, K.P., Yuille, A.L.: Weakly-and semi-supervised learning of a deep convolutional network for semantic image segmentation. In: ICCV. (2015) 6

[52] Pathak, D., Krähenbühl, P., Darrell, T.: Constrained convolutional neural networks for weakly supervised segmentation. In: ICCV. (2015) 6

[53] Everingham, M., Eslami, S.A., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The pascal visual object classes challenge: A retrospective. IJCV (2015) 6