Online Learning of Object Representations by Appearance Space Feature Alignment

Sören Pirk[1] and Mohi Khansari[2] and Yunfei Bai[2] and Corey Lynch[1] and Pierre Sermanet[1]

***Abstract*— We propose a self-supervised approach for learning representations of objects from monocular videos and demonstrate it is particularly useful for robotics. The main contributions of this paper are: 1) a self-supervised model called Object-Contrastive Network (OCN) that can discover and disentangle object attributes from video without using any labels; 2) we leverage self-supervision for online adaptation: the longer our online model looks at objects in a video, the lower the object identification error, while the offline baseline remains with a large fixed error; 3) we show the usefulness of our approach for a robotic pointing task; a robot can point to objects similar to the one presented in front of it. Videos illustrating online object adaptation and robotic pointing are provided as supplementary material.**

I. Introduction

One of the biggest challenges in real world robotics is robustness and adaptability to new situations. A robot deployed in the real world is likely to encounter a number of objects it has never seen before. Even if it can identify the class of an object, it may be useful to recognize a particular instance of it. Relying on human supervision in this context is unrealistic. Instead if a robot can self-supervise its understanding of objects, it can adapt to new situations when using online learning. Online self-supervision is key to robustness and adaptability and arguably a prerequisite to real-world deployment. Moreover, removing human supervision has the potential to enable learning richer and less biased continuous representations than those obtained by supervised training and a limited set of discrete labels. Unbiased representations can prove useful in unknown future environments different from the ones seen during supervision, a typical challenge for robotics. Furthermore, the ability to autonomously train to recognize and differentiate previously unseen objects as well as to infer general properties and attributes is an important skill for robotic agents.

In this work we focus on situated settings (i.e. an agent is embedded in an environment), which allows us to use temporal continuity as the basis for self-supervising correspondences between different views of objects. We present a self-supervised method that learns representations to disentangle perceptual and semantic object attributes such as class, function, and color. Assuming a pre-existing objectness detector, we extract objects from random frames of a scene containing the same objects, and let a metric learning system decide how to assign positive and negative pairs of embeddings. Representations that generalize across objects naturally emerge despite not being given groundtruth matches. Unlike previous methods, we abstain from employing additional self-supervisory training signals such as depth or those used for tracking. The only input to the system are monocular videos. This simplifies data collection and allows our embedding to integrate into existing end-to-end learning pipelines. We demonstrate that a trained Object-Contrastive Network (OCN) embedding allows us to reliably identify object instances based on their visual features such as color and shape. Moreover, we show that objects are also organized along their semantic or functional properties. For example, a cup might not only be associated with other cups, but also with other containers like bowls or vases.

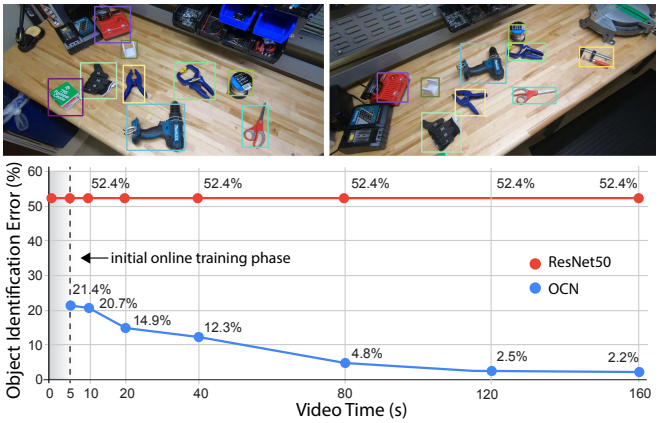

Fig. 1. **The longer our model looks at objects in a video, the lower the object identification error.** Left: example frames of a work bench video along with the detected objects. Right: result of online training on the same video. Our model self-supervises object representations as the video progresses and converges to 2% object identification error while the offline baseline remains at 52% error.

Fig. 1 shows the effectiveness of online training: we randomly selected frames of a continuous video sequence (top), OCN can adapt to the present objects and thereby lower the object identification error. The graph (bottom) shows the object identification error obtained by training on progressively longer sub-sequences of a 200 seconds video. While the supervised baseline remains at a high error rate (52.4%), OCN converges to a 2.2% error.

The key contributions of this work are: (1) a self-supervised objective trained with contrastive learning that can discover and disentangle object attributes from video without using any labels; (2) we leverage object self-supervision for online adaptation: the longer our model looks at objects in a video, the lower the object identification error, while the offline baseline remains with a large fixed

[1]Google Brain/Robotics at Google, 1600 Amphitheater Parkway, Mountain View, CA 94043
[2]X, 100 Mayfield Ave, Mountain View, CA 94043

error; (3) we let a robot collect data, then train on it with our self-supervised training scheme, and show the robot can point to objects similar to the one presented in front of it, demonstrating generalization of identifying object attributes.

II. Related Work

Object discovery from visual media. Identifying objects and their attributes has a long history in computer vision and robotics [39]. Traditionally, approaches focus on identifying regions in unlabeled images to locate and identify objects [36], [2]. Discovering objects based on the notion of 'objectness' instead of specific categories enables more principled strategies for object recognition [40], [32]. Several methods address the challenge to discover, track, and segment objects in videos based on supervised [42] or unsupervised [18], [34], [11] techniques. The spatio-temporal signal present in videos can also help to reveal additional cues that allow to identify objects [43], [16]. In the context of robotics, methods also focus on exploiting depth to discover objects and their properties [22], [17].

Many recent approaches exploit the effectiveness of convolutional deep neural networks to detect objects [31], [20], [12]. While the detection efficiency of these methods is unparalleled, they rely on supervised training procedures and therefore require large amounts of labeled data. Self-supervised methods for the discovery of object attributes mostly focus on learning representations by identifying features in multi-view imagery [6], [19] and videos [43], or by stabilizing the training signal through domain randomization [7]. Some methods not only operate on RGB images but also employ additional signals, such as depth [9], [29] or egomotion [1] to self-supervise the learning process. It has been recognized, that contrasting observations from multiple views can provide a view-invariant training signal allowing to even differentiate subtle cues as relevant features that can be leveraged for instance categorization and imitation learning tasks [35].

Unsupervised representation learning. Unlike supervised learning techniques, unsupervised methods focus on learning representations directly from data to enable image retrieval [27], transfer learning [47], image denoising [41], learning dense representations [33], [9], [38] and other tasks [8], [44]. Using data from multiple modalities, such as imagery of multiple views [35], sound [24], [3], or other sensory inputs [5], along with the often inherent spatio-temporal coherence [7], [30], can facilitate the unsupervised learning of representations and embeddings. For example, [46] explore multiple architectures to compare image patches and [26] exploit temporal coherence to learn object-centric features. [10] rely of spatial proximity of detected objects to determine attraction in metric learning, OCN operates similarly but does not require spatial proximity for positive matches, it does however take advantage of the likely presence of a same object in any pair of frames within a video. [48] also take a similar unsupervised metric learning approach for tracking specific faces, using tracking trajectories and heuristics for matching trajectories and obtain richer positive matches. While our approach is simpler in that it does not require tracking or 3D matching, it could be augmented with extra matching signals.

In robotics and other real-world scenarios where agents are often only able obtain sparse signals from their environment, self-learned embeddings can serve as an efficient representation to optimize learning objectives. [25] introduce a curiosity-driven approach to obtain a reward signal from visual inputs; other methods use similar strategies to enable grasping [28] and manipulation tasks [35], or to be pose and background agnostic [14]. [23] jointly uses 3D synthetic and real data to learn a representation to detect objects and estimate their pose, even for cluttered configurations. [15] learn semantic classes of objects in videos by integrating clustering into a convolutional neural network.

III. Learning of Object Representations

We propose a model called Object-Contrastive Network (OCN) trained with a metric learning loss based on the following steps: 1) we randomly extract two frames of a video sequences, 2) we detect objects in these frames by using an off-the-shelf objectness detector [31], 3) we use a standard ConvNet (ResNet50) and individually embed each object, 4) we use the embeddings to compute a distance matrix of the objects of one frame against the objects of the other frame and find the closest matching pairs of objects; objects of one frame are selected as anchors and their closest match from the other frame as positives, 5) we train our OCN model with a metric learning loss (n-pairs loss [37]); nearest neighbors in the embedding space are pulled together while being pushed away from dissimilar objects. This training scheme does not rely on knowing the true correspondence between objects and therefore does not require any labels. Fig. 2 shows the steps of our setup.

The fact that this works despite not using any labels might be counter-intuitive. One of the main findings of this paper is that given a limited set of objects, object correspondences will naturally emerge when using metric learning. One advantage of the self-supervised learning of object representations is that objects are organized in a continuous and multi-dimensional (e.g. shape, color, function, etc.) way; object properties are not biased by or limited to a discrete set of labels determined by human annotators. We show these embeddings allow us to discover and disentangle object attributes and that they generalize to previously unseen environments. Fig. 3 illustrates how objects of one frame (anchors) are matched to the objects of another frame after 20K training iterations.

We propose a self-supervised approach to learn object representations for the following reasons: (1) make data collection simple and scalable, (2) increase autonomy in robotics by continuously learning about new objects without assistance, (3) discover continuous representations that are richer and more nuanced than the discrete set of attributes that humans might provide as supervision, which may not match future and new environments. All these objectives

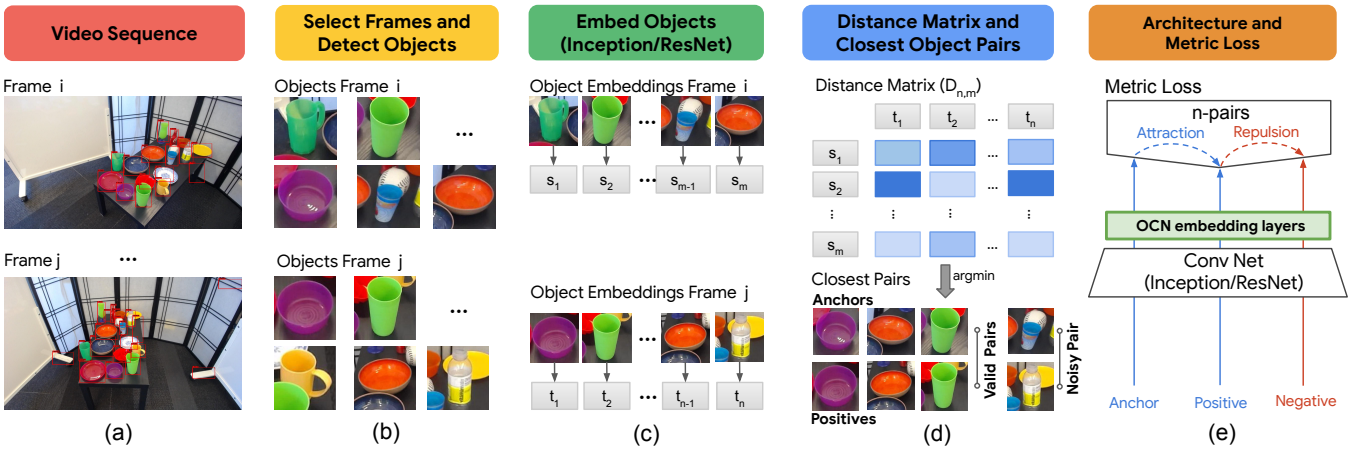

Fig. 2. **Object-Contrastive Networks (OCN)**: we use two randomly selected frames of a video sequence (a) to detect objects based on their objectness (b). We embed the detected objects with a ConvNet (c) and compute a distance matrix of the objects in one frame against the objects of the other frame (d). We select the objects of one frame as anchors and the closest objects of the other frame as positives (d). We use n-pairs loss to train an OCN embedding without any labels (e). Each object is attracted to its closest neighbor while being pushed away from all dissimilar objects. Object pairs may be wrong (e.g. the same object in two different frames is not matched with itself), however the training still converges toward disentangled object representations.

require a method that can learn about objects and differentiate them without supervision. To bootstrap our learning signal we leverage two assumptions: (1) we are provided with a general objectness model so that we can attend to individual objects in a scene, (2) during an observation sequence the same objects will be present in most frames. Given a video sequence of a scene containing multiple objects, we randomly select two frames I and $\hat{I}$ in the sequence and detect the objects present in each image. Let us assume the objects N and M are detected in images I and $\hat{I}$, respectively. Each of the n-th and m-th cropped object images are embedded in a low dimensional space, organized by a metric learning objective. Unlike traditional methods, which rely on human-provided similarity labels to drive metric learning, we use a self-supervised approach to mine similarity labels (Fig. 2).

Objectness Detection: To detect objects, we use Faster-RCNN [31] trained on the COCO object detection dataset [21]. Faster-RCNN detects objects in two stages: first generate class-agnostic bounding box proposals of all objects present in an image (Fig. 2, a, b), second associate detected objects with class labels. We use OCN to discover object attributes, and only rely on the first *objectness* stage of Faster-R-CNN to detect object candidates.

A. Metric Loss for Object Disentanglement

We denote a cropped object image by $x \in \mathcal{X}$ and compute its embedding based on a convolutional neural network $f(x) : \mathcal{X} \rightarrow K$. Note that for simplicity we may omit x from $f(x)$ while f inherits all superscripts and subscripts. Let us consider two pairs of images I and $\hat{I}$ that are taken at random from the same contiguous observation sequence. Let us also assume there are n and m objects detected in I and $\hat{I}$ respectively. We denote the n-th and m-th objects in the images I and $\hat{I}$ as x_n^I and $x_m^{\hat{I}}$, respectively. We compute the distance matrix $D_{n,m} = \sqrt{(f_n^I - f_m^{\hat{I}})^2},\ n \in 1..N,\ m \in 1..M$. For every embedded *anchor* $f_n^I,\ n \in 1..N$, we select a *positive* embedding $f_m^{\hat{I}}$ with minimum distance as *positive*: $f_{n+}^{\hat{I}} = argmin(D_{n,m})$. Given a batch of (*anchor*, *positive*) pairs $\{(x_i, x_i^+)\}_{i=1}^N$, the n-pair loss is defined as follows [37]:

$$\mathcal{L}_{N-pair}\big(\{(x_i, x_i^+)\}_{i=1}^N; f\big) = \frac{1}{N}\sum_{i=1}^{N} log\Big(1 + \sum_{j \neq i} exp(f_i^\intercal f_j^+ - f_i^\intercal f_i^+)\Big)$$

The loss learns embeddings that identify ground truth (anchor, positive)-pairs from all other (anchor, negative)-pairs in the same batch. It is formulated as a sum of softmax multi-class cross-entropy losses over a batch, encouraging the inner product of each (anchor, positive)-pair $(f_i,\ f_i^+)$ to be larger than all (anchor, negative)-pairs $(f_i,\ f_{j\neq i}^+)$. The final OCN training objective over a sequence is the sum of npairs losses over all pairs of individual frames:

$$\mathcal{L}_{OCN} = \mathcal{L}_{N-pair}\big(\{(x_n^I, x_{n+}^{\hat{I}})\}_{n=1}^N; f\big) + \mathcal{L}_{N-pair}\big(\{(x_m^{\hat{I}}, x_{m+}^I)\}_{m=1}^M; f\big).$$

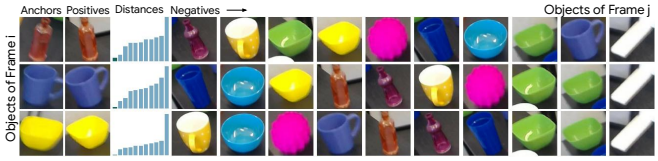

Fig. 3. View-to-view object correspondences: the first column shows all objects detected in one frame (anchors). Each object is associated to the objects found in the other frame, objects in the second column are the nearest neighbors (positives). The third column shows the embedding space distance of objects. The remaining objects (negatives) are shown from left to right in descending order according to their distances to the anchor (not all objects shown).

B. Network Architecture and Embedding Space

OCN uses a standard ResNet50 architecture until layer *global_pool* (which can be initialized with ImageNet pre-trained weights). We then add three additional convolutional layers and a fully connected layer to produce the final

embedding. The network is trained with the n-pairs metric learning loss as discussed in Sec. III-A; our architecture is depicted in 2 (e).

Object-centric Embeding Space: By using multiple views of the same scene and by attending to individual objects, our architecture allows us to differentiate subtle variations of object attributes. Observing the same object across different views facilitates learning invariance to scene-specific properties, such as scale, occlusion, lighting, and background, as each frame exhibits variations of these factors. The network solves the metric loss by representing object-centric attributes, such as shape, function, or color, as these are consistent for (anchor, positive)-pairs, and dissimilar for (anchor, negative)-pairs.

C. Discussion

One might expect that this approach may only work if it is given an initialization so that matching the same object across multiple frames is more likely than random chance. While ImageNet pretraining certainly helps convergence as shown in Tab. II, it is not a requirement to learn meaningful representations as shown in Tab. III. When all weights are random and no labels are provided, we estimate that the co-occurrence of the following hypotheses drives this convergence: (1) objects often remain visually similar to themselves across multiple viewpoints, (2) limiting the possible object matches within a scene increases the likelihood of a positive match, (3) the low-dimensionality of the embedding space forces the model to generalize by sharing abstract features across objects, (4) the smoothness of embeddings learned with metric learning facilitates convergence when supervision signals are weak, and (5) occasional true-positive matches (even by chance) yield more coherent gradients than false-positive matches which produce inconsistent gradients and dissipate as noise, leading over time to an acceleration of consistent gradients and stronger initial supervision signal.

D. Training

OCN is trained based on the detected objects of two views of the same synthetic or real scene. We randomly pick two frames of a video sequence and detect objects to produce two sets of cropped images. The distance matrix $D_{n,m}$ (Sec. III-A) is constructed based on the individually detected objects for each of the two frames. The object detector was not specifically trained on any of our datasets. As the number of detected objects per view varies, we reciprocally use both frames to find anchors and their corresponding positives as discussed in Sec. III-A. Across our experiments, we observed an embeddings size of 32-64 provides optimal results; training converged after 600k-1.2M iterations.

IV. Experimental Results

We evaluated the effectiveness of OCN embeddings on identifying objects through self-supervised online training, a real robotics pointing tasks, and large-scale synthetic data.

Fig. 4. The environments we used for our self-supervised online experiment. Top: living room, office, kitchen. Bottom: one of our more challenging scenes, and two examples of the Epic-Kitchens [4] dataset.

Fig. 5. We use 187 unique object instance in the real world experiments: 110 object for training (left), 43 objects for test (center), and 34 objects for validation (right). The degree of similarity makes it harder to differentiate these objects.

A. Online Object Identification

Our online training scheme enables to train and to evaluate on unseen objects and scenes. This is of utmost importance for robotic agents to ensure adaptability and robustness in real world scenes. To show the potential of our method for these situations we use OCN embeddings to identify instances of objects in multiple views and over time.

We quantitatively evaluate the online adaptation capabilities of our model through the object identification error of novel objects. We use sequences of videos showing objects in random configurations in different environments and train an OCN on the first 5, 10, 20, 40, 80, and 160 seconds of a 200 seconds video. Our dataset provides object bounding boxes and unique identifiers for each object as well as reference objects and their identifiers. The goal of this experiment is to assign the identifier of a reference object to the matching object detected in a video frame. We evaluate the identification error (ground truth index vs. assigned index) of objects present in the last 40 seconds of each video and for each training phase to compare our results to a ResNet50 (2048-dimensional vectors) baseline.

We train an OCN for each video individually. Therefore, we only split our dataset into validation and testing data. We used 42 videos of the six categories kids room, kitchen, living room, office, work bench, and Epic-Kitchens [4] (Fig. 4). For each category we used 4 videos for validation and 3 for testing. We jointly train on the validation videos to find meaningful hyperparameters across the categories and use the same hyperparameters for the test videos.

In Fig. 1 we show that a model observing objects for a few minutes from different angles can self-teach to identify them almost perfectly while the offline supervised approach cannot. The supervised offline baseline stays at a 52.4% error, while OCN improves down to 2% error after 80s, a 25x error reduction. Fig. 6 shows the same video frames of two scenes from our dataset. Objects with wrongly matched in-

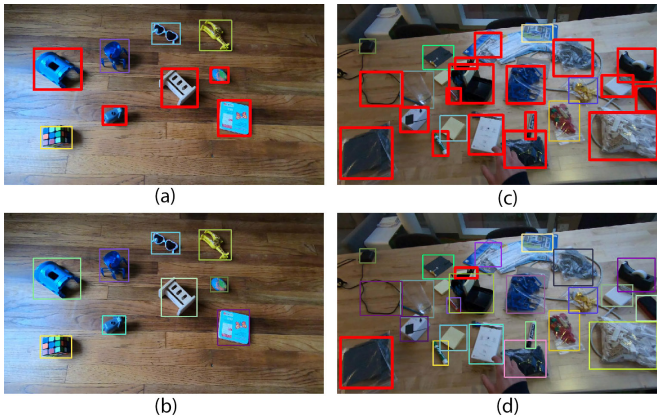

Fig. 6. Comparison of identifying objects with ResNet50 (a, c) and OCN (b, d) embeddings for the environments kids room and challenging. Red bounding boxes indicate a mismatch of ground truth and associated index

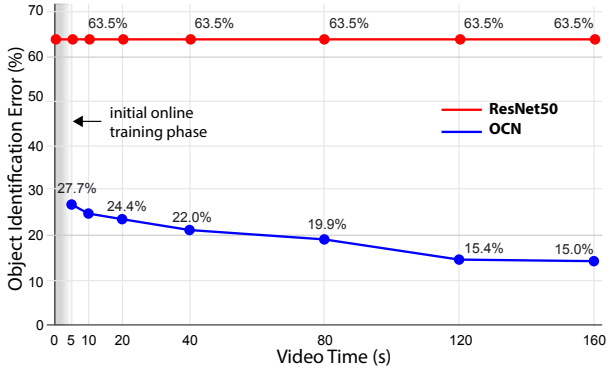

Fig. 7. Evaluation of online adaptation: we train an OCN on the first 5, 10, 20, 40, 80, and 160 seconds of each 200 second test video and then evaluate on the remaining 40 seconds. Here we report the lowest average error of all videos (over 1000K iterations) of online adaptation.

dices are shown with a red bounding box, correctly matched objects are shown with random colors. In Fig. 7 we report the average error of OCN object identification across our videos compared to the ResNet50 baseline. As the supervised model cannot adapt to unknown objects without providing labels, OCN outperforms this baseline by a large margin. Furthermore, the optimal result among the first 50K training iterations closely follows the overall optimum obtained after 1000K iterations. Fig 10 shows a t-SNE plot of the generated embeddings for one of the EpicKitchens scenes.

B. Robotic Pointing

To evaluate OCN for real world robotics scenarios we defined a robotics pointing task. The goal of the task is to enable a robot to point to an object that it deems most similar to the object directly in front of it (Fig. 8). The objects on the rear table are randomly selected from the object categories (Fig. 5). We consider two sets of these target objects. The quantitative experiment in Tab. I uses three query objects per category and is ran three times for each combination of query and target objects ($3 \times 2 \times 18 = 108$ experiments performed).

We collect data with a real robot by looking at a table from multiple angles and then train OCN. The robot is then tasked to point to objects similar to the one presented in front of it. Objects can be similar in terms of shape, color or class. If able to perform that task, the robot has learned to

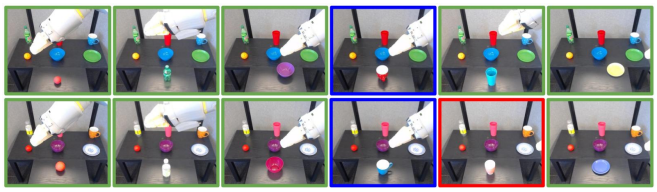

Fig. 8. The robot experiment of pointing to the best match of a query object (placed in front of the robot on the small table). The closest match is selected from two sets of target objects, placed on the table behind the query object. The first and the second row correspond to the experiment for the first and second target set. Images with green frame indicate cases where both the 'class' and 'container' attributes are matched correctly. Blue frames show where only the 'container' attribute is matched correctly and red frames indicate neither attribute is matched.

distinguish and recognize these attributes. The robot is able to perform the pointing task with 72% recognition accuracy of 5 classes, and 89% recognition accuracy of the binary is-container attribute.

We report errors related to 'class' and 'container' attributes. While the trained OCN model is performing well on the most categories, it has difficulty on the object classes 'cups & mugs' and 'glasses'. These categories are generally mistaken with the category 'bowls'. As a result the network performs much better in the attribute 'container' since all the three categories 'bowls', 'bottles & cans', and 'glasses' refer to the same attribute. At the beginning of each experiment the robot captures a snapshot of the scene. We then split the captured image into two images: the upper portion of the image that contains the target object set and the lower portion of the image that only contains the query object. We detect the objects and find the nearest neighbor of the query object in the embedding space to find the closest match.

TABLE I

EVALUATION OF ROBOTIC POINTING

Objects	Class Error	Container Error
Balls	11.1 ±7.9%	11.1 ±7.9%
Bottles & Cans	0.0 ±0.0%	0.0 ±0.0%
Bowls	22.2 ±15.7%	16.7 ±0.0%
Cups & Mugs	88.9 ±7.9%	16.7 ±13.6%
Glasses	38.9 ±7.9%	5.6 ±7.9%
Plates	5.6 ±7.9%	11.1 ±2.3%
Total	27.8 ±3.9%	11.1 ±2.3%

C. Object Attribute Classification and Offline Analysis

To analyze what our model is able to disentangle, we quantitatively evaluate performance on a large-scale synthetic dataset. We used 12k object models of the ModelNet40 dataset [45] to generate 100K object arrangements (Fig. 9) and use a 80-20-20 split for training, validation, and testing data. In Tab. II we find that our self-supervised model closely follows its supervised equivalent baseline when trained with metric learning. The cross-entropy/softmax supervised baseline approach performs best and establishes the upper-bound error while the ResNet50 baseline is the lower-bound.

One way to evaluate the quality of unsupervised embeddings is to train attribute classifiers on top of the embedding using labeled data. Note however, that this may not entirely reflect the quality of an embedding because classification is only measuring a discrete and small number of attributes

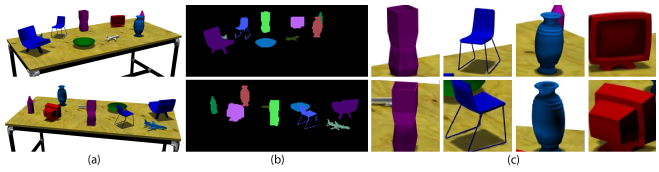

Fig. 9. Synthetic data: two frames of a synthetically generated scene of table-top objects (a) and a subset of the detected objects (c). To validate our method against a supervised baseline, we additionally render color masks (b) that allow us to identify objects across the views and to associate them with their semantic attributes after object detection. Note that objects have the same color id across different views. The color id's allow us to supervise the OCN during training.

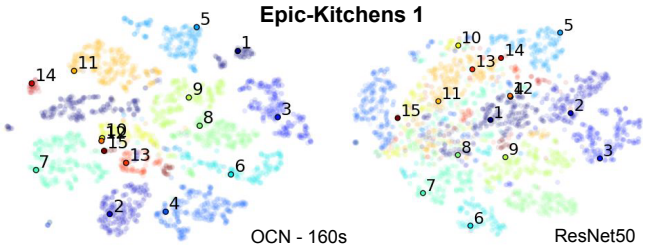

Fig. 10. t-SNE plots of Epic-Kitchens object embeddings. The plots show each object of the 600 frames used for evaluation with their ground truth index as color. Compared to the ResNet50 baseline, OCN trained on 160 seconds of video produces a cleaner seperation of clusters, which indicates an improved disentanglement of object features.

while an embedding may capture more continuous and larger number of visual object features.

Classifiers: we consider two types of classifiers to be applied on top of existing embeddings in this experiment: linear and nearest-neighbor classifiers. The linear classifier consists of a single linear layer going from embedding space to the 1-hot encoding of the target label for each attribute. It is trained with a range of learning rates and the best model is retained for each attribute. The nearest-neighbor classifier consists of embedding objects of an entire 'training' set. For each object embedding of the evaluation set we then assign the labels of the nearest sample from the training set. Nearest-neighbor classification is not an ideal approach because it does not necessarily measure generalization as linear classification does and results may vary significantly depending on how many nearest neighbors are available. It is also less subject to data imbalances. We still report this metric to get a sense of its performance because in an unsupervised inference context, the models might be used in a nearest-neighbor fashion (e.g. as in Sec. IV-B).

Baselines: we compare multiple baselines (BL) in Tab. II. The 'Softmax' baseline refers to the exact same architecture as for OCN except that the model is trained with a supervised cross-entropy/softmax loss. The 'ResNet50' baseline refers to using the unmodified outputs of the ResNet50 model [13] (2048-dimensional vectors) as embeddings and training a nearest-neighbor classifier as defined above. We consider 'Softmax' and 'ResNet50' baselines as the lower and upper error-bounds for standard approaches to a classification task. The 'OCN supervised' baseline refers to an OCN trained with ground truth matches that provided rather than discovered automatically. 'OCN supervised' represents the metric learning upper bound for classification. Finally we indicate the error rates for random classification.

TABLE II

ATTRIBUTES CLASSIFICATION ERRORS

Method	Class (12) Attribute Error	Color (8) Attribute Error	Binary Attributes Error	Embedding Size
[BL] Softmax	2.98%	0.80%	7.18%	-
[BL] OCN sup (linear)	7.49%	3.01%	12.77%	32
[BL] OCN sup (NN)	9.59%	3.66%	12.75%	32
[ours] OCN unsup. (linear)	10.70%	5.84%	13.76%	24
[ours] OCN unsup. (NN)	12.35%	8.21%	13.75%	24
[BL] ResNet50 embed. (NN)	14.82%	64.01%	13.33%	2048
[BL] Random Chance	91.68%	87.50%	50.00%	-

TABLE III

RESULTS WITH RANDOM WEIGHTS (NO IMAGENET PRE-TRAINING)

Method	Class (12) Attribute Error	Color (8) Attribute Error	Binary Attributes Error	Finetuning
[BL] Softmax	23.18%	10.72%	13.56%	yes
[BL] OCN sup. (NN)	29.99%	2.23%	20.25%	yes
[BL] OCN sup. (linear)	34.17%	2.63%	27.37%	yes
[ours] OCN unsup. (NN)	35.51%	2.93%	22.59%	yes
[ours] OCN unsup. (linear)	47.64%	4.43%	35.73%	yes
[BL] Softmax	27.28%	5.48%	20.40%	no
[BL] OCN sup. (NN)	37.90%	4.00%	23.97%	no
[BL] OCN sup. (linear)	39.98%	4.68%	32.74%	no
[ours] OCN unsup. (NN)	43.01%	5.56%	26.29%	no
[ours] OCN unsup. (linear)	48.26%	6.15%	37.05%	no
[BL] ResNet50 embed. (NN)	59.65%	21.14%	34.94%	no
[BL] Random Chance	91.68%	87.50%	50.00%	-

Results: we quantitatively evaluate our unsupervised models against supervised baselines on the labeled synthetic datasets. Note that there is no overlap in object instances between the training and the evaluation set. The take-away is that unsupervised performance closely follows its supervised baseline when trained with metric learning. As expected the cross-entropy/softmax approach performs best and establishes the error lower bound while the ResNet50 baseline are upper-bound results.

V. CONCLUSION AND FUTURE WORK

We introduced a self-supervised objective for object representations that is able to disentangle object attributes, such as color, shape, and function. We showed this objective can be used in online settings which is particularly useful for robotics to increase robustness and adaptability to unseen objects. We demonstrated a robot is able to discover similarities between objects and pick an object that matches the visual features to one presented to it. In summary, we find that within a single scene with novel objects, the more our model looks at these objects, the more it can recognize them and understand their visual attributes, despite never receiving any labels for them. Current limitations include relying on all objects to be present in all frames of a video. Relaxing this limitation would allow to use the model in unconstrained settings. Additionally, the online training is currently not real-time as we first set out to demonstrate the usefulness of online-learning in non-real-time. Real-time training requires additional engineering that is beyond the scope of this research.

REFERENCES

[1] P. Agrawal, J. Carreira, and J. Malik. Learning to see by moving. In *ICCV*, 2015.

[2] H. Arora, N. Loeff, D. A. Forsyth, and N. Ahuja. Unsupervised segmentation of objects using efficient learning. In *CVPR*, pages 1–7, June 2007.

[3] Y. Aytar, C. Vondrick, and A. Torralba. Soundnet: Learning sound representations from unlabeled video. In *NIPS*, 2016.

[4] D. Damen, H. Doughty, G. M. Farinella, S. Fidler, A. Furnari, E. Kazakos, D. Moltisanti, J. Munro, T. Perrett, W. Price, and M. Wray. Scaling egocentric vision: The epic-kitchens dataset. In *ECCV*, 2018.

[5] O. Dehzangi, M. Taherisadr, and R. ChangalVala. Imu-based gait recognition using convolutional neural networks and multi-sensor fusion. *Sensors*, 17(12), 2017.

[6] D. DeTone, T. Malisiewicz, and A. Rabinovich. Superpoint: Self-supervised interest point detection and description. *CoRR*, abs/1712.07629, 2017.

[7] C. Doersch, A. Gupta, and A. A. Efros. Unsupervised visual representation learning by context prediction. In *ICCV*, 2015.

[8] V. Dumoulin, I. Belghazi, B. Poole, A. Lamb, M. Arjovsky, O. Mastropietro, and A. Courville. Adversarially learned inference. *CoRR*, abs/1606.00704, 2016.

[9] P. R. Florence, L. Manuelli, and R. Tedrake. Dense object nets: Learning dense visual object descriptors by and for robotic manipulation. *CoRL*, 2018.

[10] R. Gao, D. Jayaraman, and K. Grauman. Object-centric representation learning from unlabeled videos. *CoRR*, abs/1612.00500, 2016.

[11] E. Haller and M. Leordeanu. Unsupervised object segmentation in video by efficient selection of highly probable positive features. In *ICCV*, 2017.

[12] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask R-CNN. In *ICCV*, 2017.

[13] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *CVPR*, pages 770–778, 2016.

[14] D. Held, S. Thrun, and S. Savarese. Deep learning for single-view instance recognition. *CoRR*, abs/1507.08286, 2015.

[15] S. Hickson, A. Angelova, I. A. Essa, and R. Sukthankar. Object category learning and retrieval with weak supervision. *CoRR*, abs/1801.08985, 2018.

[16] S. D. Jain, B. Xiong, and K. Grauman. Fusionseg: Learning to combine motion and appearance for fully automatic segmentation of generic objects in videos. In *CVPR*, pages 2117–2126, 2017.

[17] A. Karpathy, S. Miller, and L. Fei-Fei. Object discovery in 3d scenes via shape analysis. In *ICRA*, 2013.

[18] S. Kwak, M. Cho, I. Laptev, J. Ponce, and C. Schmid. Unsupervised object discovery and tracking in video collections. In *ICCV*, 2015.

[19] T. Lin, Y. Cui, S. Belongie, and J. Hays. Learning deep representations for ground-to-aerial geolocalization. In *CVPR*, pages 5007–5015, 2015.

[20] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017.

[21] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, editors, *ECCV*, pages 740–755, 2014.

[22] A. K. Mishra, A. Shrivastava, and Y. Aloimonos. Segmenting "simple" objects using rgb-d. In *ICRA*, pages 4406–4413, May 2012.

[23] C. Mitash, K. E. Bekris, and A. Boularias. A self-supervised learning system for object detection using physics simulation and multi-view pose estimation. In *IROS*, pages 545–551. IEEE, 2017.

[24] A. Owens, P. Isola, J. H. McDermott, A. Torralba, E. H. Adelson, and W. T. Freeman. Visually indicated sounds. In *CVPR*, pages 2405–2413, June 2016.

[25] D. Pathak, P. Agrawal, A. A. Efros, and T. Darrell. Curiosity-driven exploration by self-supervised prediction. In *ICML*, 2017.

[26] D. Pathak, R. Girshick, P. Dollár, T. Darrell, and B. Hariharan. Learning features by watching objects move. In *CVPR*, 2017.

[27] M. Paulin, M. Douze, Z. Harchaoui, J. Mairal, F. Perronin, and C. Schmid. Local convolutional features with unsupervised training for image retrieval. In *ICCV*, pages 91–99, Dec 2015.

[28] L. Pinto and A. Gupta. Supersizing self-supervision: Learning to grasp from 50k tries and 700 robot hours. In *ICRA*, pages 3406–3413, May 2016.

[29] E. Pot, A. Toshev, and J. Kosecka. Self-supervisory signals for object discovery and detection. *CoRR*, abs/1806.03370, 2018.

[30] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *CoRR*, abs/1511.06434, 2015.

[31] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, pages 91–99. 2015.

[32] A. C. Romea, M. M. Torres, and S. Srinivasa. The moped framework: Object recognition and pose estimation for manipulation. *IJRR*, 30(10):1284 – 1306, 2011.

[33] T. Schmidt, R. Newcombe, and D. Fox. Self-supervised visual descriptor learning for dense correspondence. *IEEE Robotics and Automation Letters*, 2(2):420–427, 2017.

[34] S. Schulter, C. Leistner, P. Roth, and H. Bischof. Unsupervised object discovery and segmentation in videos. In *BMVC*, 2013.

[35] P. Sermanet, C. Lynch, Y. Chebotar, J. Hsu, E. Jang, S. Schaal, and S. Levine. Time-contrastive networks: Self-supervised learning from video. In *ICRA*, 2018.

[36] J. Sivic, B. C. Russell, A. A. Efros, A. Zisserman, and W. T. Freeman. Discovering objects and their location in images. In *ICCV*, 2005.

[37] K. Sohn. Improved deep metric learning with multi-class n-pair loss objective. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *NIPS*, pages 1857–1865. 2016.

[38] J. Thewlis, H. Bilen, and A. Vedaldi. Unsupervised learning of object frames by dense equivariant image labelling. In *NIPS*, pages 1–12, 2017.

[39] T. Tuytelaars, C. H. Lampert, M. B. Blaschko, and W. Buntine. Unsupervised object discovery: A comparison. *IJCV*, 2009.

[40] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders. Selective search for object recognition. *IJCV*, 104(2):154–171, 2013.

[41] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol. Extracting and composing robust features with denoising autoencoders. In *ICML*, pages 1096–1103, 2008.

[42] L. Wang, G. Hua, R. Sukthankar, J. Xue, and N. Zheng. Video object discovery and co-segmentation with extremely weak supervision. In *ECCV*, 2014.

[43] X. Wang and A. Gupta. Unsupervised learning of visual representations using videos. In *ICCV*, 2015.

[44] X. Wang and A. Gupta. Unsupervised learning of visual representations using videos. pages 2794–2802, 2015.

[45] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao. 3d shapenets: A deep representation for volumetric shapes. In *CVPR*, pages 1912–1920. IEEE Computer Society, 2015.

[46] S. Zagoruyko and N. Komodakis. Learning to compare image patches via convolutional neural networks. In *CVPR*, June 2015.

[47] R. Zhang, P. Isola, and A. A. Efros. Split-brain autoencoders: Unsupervised learning by cross-channel prediction. In *CVPR*, 2017.

[48] S. Zhang, J. Huang, J. Lim, Y. Gong, J. Wang, N. Ahuja, and M. Yang. Tracking persons-of-interest via unsupervised representation adaptation. *CoRR*, abs/1710.02139, 2017.