Test-Time Training for Improved Object Affordance Prediction

Jitendra Malik, Ed.
Ren Ng, Ed.

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2020-115
http://www2.eecs.berkeley.edu/Pubs/TechRpts/2020/EECS-2020-115.html

May 29, 2020

Copyright © 2020, by the author(s).
All rights reserved.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission.

Test-Time Training for Improved Object Affordance Prediction

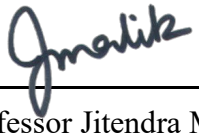
by Gefen Kohavi

Research Project

Submitted to the Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, in partial satisfaction of the requirements for the degree of **Master of Science, Plan II**.

Approval for the Report and Comprehensive Examination:

Committee:



Professor Jitendra Malik
Research Advisor

May 28, 2020

(Date)

* * * * * * *

Professor Ren Ng
Second Reader

May 29, 2020

(Date)

Abstract

Learning affordances of unseen objects is an important aspect of learning how to interact with and understand the world. However, current research on this subject is restricted to small datasets that are limited in variety. Recent efforts developing weakly-supervised approaches show progress in making affordance prediction more generalized, but performance gaps remain from supervised methods. This paper attempts to close this gap by proposing that affordance prediction is primarily a representation learning task between inactive images and active videos. As a result, this paper proposes using an unsupervised representation learning task that can be be trained on images not in the training data. The final model improves robustness of learning Grounded Interactions Hotspots to changes in model types and to out-of-domain objects and further bridges the gap between weak and strong supervision approaches all while requiring no extra parameters.

Thank you to friends and family for their love and support.

Contents

1. Introduction

In computer vision, most modern research on objects deals with detection, recognition, and localization. More generally, most frameworks attempt to understand what components exist in a scene, not how components in a scene can interact. Learning how objects interact is both non-trivial to understand and yet vital for scene understanding. When we understand how interaction affects the environment, we can further understand how to improve embodied and intelligent behavior and interaction in a scene.

The theory of affordances as suggested in [14, 13] by J. J. Gibson and later elaborated in a book by Donald Norman [32] poses that a major component of how the world is perceived is through the action possibilities of objects. An object is said to *afford* some action if that action can be achieved or completed with that object. Further, an object can afford multiple different actions at once; a fork can be held, eaten with, flung, or washed, among other actions. This suggests that objects also have different parts that afford each action. Therefore, the problem of understanding affordances can also be thought of as an object segmentation problem.

Understanding object affordances are an important part of both video and scene understanding. Popular deep learning models that are made for video understanding focus on what is happening in a scene [46, 48, 27, 44]. More recent work has explored the questions of where something is happening in a scene [30, 31, 12]. Implicitly, learning object affordances both tell the what and where of a scene through the context of objects. This extends past videos too. Being able to understand and predict where to interact with an inactive object given an action query is an important step for intelligent agents to interact with their environments.

The theory of affordances is also strongly rooted in neuroscience. A study by Michael Land [25] recorded eye fixations across different tasks. The resulting findings show that eye fixations are strongly associated with the task being attempted. Even though some tasks are seen as trivial, the eyes still lead motor function every step of the way. Implicitly then, humans perceive

and understand object affordances when interacting with the environment around them. Further understanding how object affordances are perceived and emerge from object design could improve our understanding of how people also perceive and interact with the world.

The task of predicting affordances for object interaction is highly non-trivial. There consists multiple problems with current research regarding object interaction. First, shape-based understanding of objects for learning interactions pose potential limitations to unseen objects with new shapes or ways of interaction. Second, current datasets for deep learning approaches are small and consist of limited domains. This poses an issue to supervised approaches that require large amounts of data for high performance. It also prevents generalization to new domains as well as robustness to hard examples within the domain trained.

Weakly supervised approaches propose exciting and strong results for object affordance prediction. Using limited labels, these methods can potentially be trained on larger and richer datasets than their supervised counterparts and as a result be more generalizable and robust. However, one primary issue with these approaches is caused by the fact that the optimization objective may not always coincide with the true objective. Lower test loss does not necessarily mean a better final metric.

Similar to how humans and animals learn, visual systems can be taught through visual demonstrations of an expert. By grounding the predictions of visual systems to these demonstrations, [30] suggests that affordances can be learned through the weakly-supervised task of video action classification. Implicitly a classifier learns to attend to afforded regions of an object in order to understand the action occurring. Using a modified version of Grad-CAM [36], [30] has to ability to visualize this attention. These attention maps can thus be effectively used as heatmaps for visual object affordance.

We follow [30] in this paper. While this method still suffers from the same afflictions as other weakly supervised, a key insight is that this method is fundamentally a contrastive learning problem. The goal of the method is to learn features of a static non-active image in a way that is similar to the features of an active frame or video.

Through reframing the problem of affordance prediction to contrastive learning, we propose the following solution to the framework. Given the

unsupervised similarity learning task in [30] we show that as long as novel input has video, the model parameters for the task can be updated, even after training.

Using this approach, we show that this method can improve performance across different size model architectures. We show across three backbone networks - ResNet 18, ResNet 34, and ResNet 50 [18] - performance benefits to test-time training across multiple evaluation metrics. We also show that this method improves performance in out-of-distribution generalization. More specifically, we show that this method generalizes well to objects not seen during training.

Our method is evaluated across a diverse video-based dataset: Online Product Reviews for Affordances (OPRA) [12]. OPRA, a third-person product review dataset, contains a wide variety of videos as well as numerous different objects and actions.

The contributions of this paper are as follows:

- We propose an unsupervised test-time training framework based on [30] that can be trained on individual examples.
- We discuss the benefits and detriments of the framework and show the performance gains of test-time training across different strength backbone models.
- We show improvement in performance to out-of-distribution examples such as object classes that were not seen during training.

2. Related Work

Object Affordance Prediction

A common approach to affordance prediction is through the task of predicting gaze or visual saliency. Works [5, 6] provide popular datasets for visual saliency comparison while [7] suggests metrics to evaluate saliency prediction. Alongside the popularity of these datasets, there has also been a rise of a multitude of methods that attempt to predict gaze and visual salience [34, 24, 20, 26]. Other methods focus on how object shape can hint at specific affordances [29, 19]. Hand pose [8, 42] can also give information about how and where an object is used.

Other endeavors in affordance prediction emphasises focus on human demonstration for affordances [22, 11, 1, 23]. Demo2Vec [12] provides OPRA, a YouTube-sourced product review dataset containing videos of human interaction as well as affordance heatmaps for each product. In addition, they also propose a novel framework to encode action for use in affordance prediction. More recently, [30] focuses on learning interaction hotspots grounded by human demonstration. Using grounded affordances has also been used to segment the enviroment and map it to distinct activity-centric zones [31].

Self-Supervised and Representation Learning

Self-supervised learning is a popular method to do representation learning without the need for any labels. Commonly, methods use implicit schemas and properties of the input data itself to learn representations. Image rotation prediction [15] and Jigsaw puzzle solving [33] are simple but popular approaches utilizing implicit assumptions about the placement of environmental attributes in a scene. Methods such as [47] show how time can provide a strong signal for deep learning. Using image color has also been utilized as

a powerful method for some self-supervised and representation learning tasks [45, 43].

To achieve strong metric learning performance on faces, triplet loss [35] has been used. More recently, more sophisticated methods have been employed to get better feature representations through unsupervised learning. Contrastive multiview coding [43] splits data into implicitly paired parts such as LAB images, NYU depth+images, or different video frames and flow, and learns to ensure the different views have the same feature representation. Momentum Contrast [17] ensures similar representations with a moving average encoder model. SimCLR [9] tries to use augmentations between the same data instance to learn a representation that is both meaningful and robust to transformations. Other work has taken these recent methods and applied them to videos [16, 21].

Test-Time Training

There has been a plethora of work focusing on only training on single examples [4, 37, 38, 39]. Some methods such as [2] adapt a trained model to new instanced. Others use self-supervised methods to allow test-data to be brought into the training process [3]. [37] and [38] show that individual images contain enough self-similarity and pattern repetitions for meaningful deep learning applications to be used such as scene-aware image resizing and still-image animation. [39, 4] attempt to solve the task of super-resolution on single instances. Finally, online learning has gained popularity with methods like [28] highlighting how these methods could potentially improve inference efficiency.

More recent work has focused on applications of test-time training to supervised tasks. [40] adds the simple self-supervised task of image rotation and tile location prediction to improve the domain adaptability of a supervised classifier. Follow-up work [41] explores test-time training for out-of-distribution robustness and generalizability as well as makes theoretical arguments for why test-time training should work for classification.

3. Method

3.1 Data

3.1.1 OPRA

OPRA is an affordance-specific dataset made by [12] to study the role of using videos to predict affordances on still images. This dataset consists of 11.5k interaction demonstrations across 2.5k objects. Overall there are 7 possible actions and 32 object categories. This dataset covers numerous YouTube videos of appliance reviews pulled from 6 different review channels. Each train instance consists of a short video demonstration of a person using the appliance, as well as a clean, static image with a white background and no interaction visible (almost like an amazon product image). Along with this information comes a noun and action label for the action done in the video. This dataset was built so that a given product may have multiple actions associated with it. Finally, each static image has associated points collected via Amazon Mechanical Turk which are human annotations on the product. Each point represents a label where each annotator thinks a part of the object affords a specific action.

3.2 Data Processing

Video frames for the OPRA dataset are sampled at 5fps. Depending on the experiment, between 2-8 frames are sampled. If a video has less frames than needed at a given sample rate, the rest of the needed frames are padded with black images.

Each data instance also contains point annotations for each of the static images. While this is useful, it is not the proper format needed for to evaluate against predictions using KL-Divergence, Similarity, and Area under the ROC

curve. As a result, all points for a given image are placed spatially and convolved with a Gaussian Kernel to generate a heatmap.

The start of the video frames are randomly sampled from each train instance. Each image (both in the video and the static images) are then resized to 256×256 and then randomly cropped to a size of 224×224. Each video and static image are flipped horizontally with probability $\frac{1}{2}$.

3.3 Model

The primary model is based off of [30] which proposes a weakly supervised model to predict affordance. This model can be split into two parts: supervised action detection, which trains multiple networks, and gradient-based class activation mapping, which is used in conjunction with the model to generate affordances.

The model proposed by [30] contains four neural networks. First, features are extracted from both the video $\mathcal{V}$ and the static image $\mathcal{I}$ using a frozen Imagenet-pretrained ResNet model (this model will vary as described in later sections). The size of the ResNet output is enlarged in increase final heatmap resolution to 28×28. When fed into the LSTM, video frames are pooled using L2-pooling to flatten the spatial dimension. The video frames are next input into an LSTM encoder for each time step. The final hidden layer of the LSTM is then fed into an action classifier to predict the action of the video. Separately from this, the static image $\mathcal{I}$ is featureized $x_{\mathcal{I}}$ using the same pretrained backbone. $x_{\mathcal{I}}$ can be described as an "inactive" featurization since no actions are present in the image. To convert the features to an "active" featurization $\hat{x}_{\mathcal{I}} = \mathcal{F}_{ant}(x_{\mathcal{I}})$, an anticipation network is used. These active features, are also fed into the same action classifier network to compute an auxiliary loss.

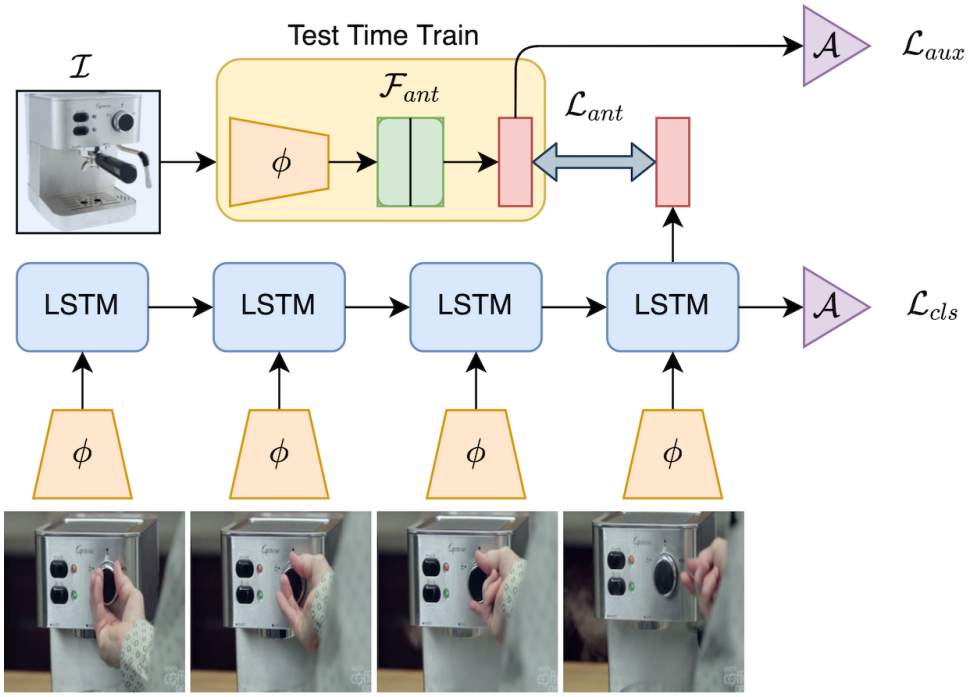


Figure 3.1: The primary hotspots model. Each video frame is fed into a ResNet backbone model ϕ. The the last hidden feature from the LSTM is used by action classifier network $\mathcal{A}$ to predict an action for the video and to compute the classification loss $\mathcal{L}_{cls}$. Features for the static image $\mathcal{I}$ are also computed using ϕ. $\mathcal{F}_{ant}$ projects the static features into new "active" features (as denoted as red boxes). The most "active" video frame's features are used to compare against the projected features using the anterior loss $\mathcal{L}_{ant}$. Finally an auxilary classification loss $\mathcal{L}_{aux}$ is also computed for the projected features.

3.4 Training

The primary loss for the network is a cross entropy loss for the action classifier on the video, denoted $\mathcal{L}_{cls}$. To train the anticipation network we minimize it's difference from the most active frame features in the video. To find the most "active" frame, [30] suggests taking the hidden cell from the LSTM that has the lowest classification loss. Alternatively, we use the hidden cell that has the highest class confidence. Given the active image feature $\hat{x}_{\mathcal{I}}$ and the most

active video frame instance x_t^* we compute a comparison loss between the two features. Since OPRA static images are visually similar to the objects in the video, a simple L2 loss can be used.

$$\mathcal{L}_{ant} = ||\hat{x}_{\mathcal{I}} - x_t^*||_2 \tag{3.1}$$

Finally, an action classification loss for the action predictor prediction given $\hat{x}_{\mathcal{I}}$ is used as an auxiliary loss $\mathcal{L}_{aux}$. The total loss is a weighted average of the three losses mentioned before.

$$\mathcal{L}(\mathcal{V}, \mathcal{I}, a) = \lambda_{cls}\mathcal{L}_{cls} + \lambda_{ant}\mathcal{L}_{ant} + \lambda_{aux}\mathcal{L}_{aux} \tag{3.2}$$

For both datasets, $\lambda_{cls} = \lambda_{aux} = 1$ and $\lambda_{ant} = 0.1$. All experiments are trained for 20 epochs using an Adam optimizer with 1e-4 learning rate and batch size 128.

3.5 Affordance Generation

Following [30], we use their modification of Grad-CAM to generate heatmaps of the static objects. Heatmaps for an image $\mathcal{I}$ are generated by creating a weighted sum of the features $x_{\mathcal{I}}$. This weighted sum is a element-wise product of the backpropaged gradient, given some target action, and the L2-pooled features $x_{\mathcal{I}}$. Like Grad-CAM, this allows attention maps to be generated anywhere along the architecture as well as for all possible actions.

4. Experiments

4.1 Setup

Experimental results for this paper are split into two parts: Model Robustness, and Object-based Domain Adaptation.

4.2 Test-Time Training

After training a model, all parameters are frozen except for the anticipation module. The goal of the anticipation network, $\mathcal{F}_{ant}$, is to project the output features from the trained backbone on the static, inactive, image to an "active" featurization. This module is a two-layer nerual network with $28 \times 28 \times f$ size input where f is the number of channels output by the backbone (in ResNet 18, ResNet 34 it is 512 and in ResNet 50 it is 2048). The output of the network represents the same feature width and height.

As shown in Table 4.1 video time length has little impact on the final performance of the model. This may be because the location of a person's hand on an object is a simple predictor of the action occurring or the model only needs to consider small motions to accurately predict the action. Consequentially, both experiments are trained with only two video frames.

Video Length	KLD ↓	SIM ↑	AUC-J ↑
Max Length 8	1.427	0.360	0.806
Max Length 4	1.443	0.358	0.802
Max Length 2	1.434	0.360	0.805

Table 4.1: All models are run with ResNet 50 backbones. ↓ and ↑ represent lower and higher is better, respectively. Max video length refers to the maximum number of frames that are fed into the LSTM. Videos with less than the max number of frames are padded with zeros. Videos with more than the max number of frames are randomly sampled for a contiguous sequence with the max length.

Given the hallucinated "active" features, the features are compared to the LSTM features of the most active video frame. Given the ground truth action, simply choosing the frame that has the highest action classification accuracy works well. Without the ground truth action, choosing the frame with the most confident action prediction works roughly the same.

As described in Section 3.4, an anticipation loss is used to ensure similarity between the two image embeddings. For OPRA, a simple L2 loss is used on the L2-pooled embeddings. This can be done as the static images have no background, perspective transformation, or occlusion.

Given a new video-image pair, the trained LSTM predicts the action occurring and the most active frame is chosen. The features for the static image are chosen and then projected. Next, only the anticipation loss is computed. Given this fully unsupervised representation loss on the single new example the anticipation network's parameters can be updated. With these new parameters, the anticipation module can predict a more "active" embedding. Consequentially, a more refined object affordance heatmap is predicted.

For both experiments, each individual test example updates the anticipation network 3 times before using the last prediction for the heatmaps. The network is updated with a learning rate of 1e-4 with an Adam optimizer.

4.2.1 Backbone Model Robustness

Three ResNet backbones are trained: ResNet18, ResNet34, and ResNet50. Each model is trained for 20 epochs. It is important to note that due to the size of the output channels, the anticipation network for Res18 and Res34 is approximately 5.2 million parameters. For Res50, the anticipation network is 16 times larger at 83.8 million parameters. As a result, we expect the greatest performance gain to come from the two models with the small anticipation network, ResNet18 and ResNet34.

As noted in Section 4.2 the anticipation network is updated three times for each test example using the anticipation loss before the final heatmap is computed.

4.2.2 Out of distribution generalization

The training and testing process are similar to the previous section. In this experiment, generalization to novel objects is tested. Only ResNet18 is used for the analysis.

For the OPRA dataset, which consists of 26 different types of objects (denoted nouns), training is split across three subsets of $\frac{13}{26}$ randomly chosen objects. Performance is then tested and compared for all test images, only the objects seen during training, and only the objects not seen during training. Results are averaged across the three splits.

4.3 Evaluation

Each static image generates a $28 \times 28 \times 1$ normalized heatmap for each possible action in the dataset. For each train and test instance, included is a list of points collected from humans each picking a point on the image where they think the object in the image affords a specific action. Given these points, Section 3.2 describes a processing step to convert them into a single normalized $28 \times 28 \times 1$ heatmap.

There are three evaluation metrics used to compare the predicted heatmap $\hat{h}$ and the ground truth heatmap h: KL-Divergence (KLD), Similarity or

2D histogram intersection (SIM), and Area under the ROC curve (AUC-Judd/AUC).

- KL-Divergence treats both histograms as probability distributions. It is defined as:

$$KL(h\|\hat{h}) = \sum_i h_i \log\left(\frac{h_i}{\hat{h}_i}\right) \tag{4.1}$$

  For this metric, lower values are better where two matching distributions have a KL-divergence of 0. KL-Divergence is sensitive to false negatives. It is bounded by [0, ∞].

- Similarity (SIM) measures the intersection between two histograms. To compute for normalized heatmaps h and $\hat{h}$, we simply need to take the sum of the minimum value between each pixel of h and $\hat{h}$. Like KL-Divergence, SIM is also sensitive to false negatives, moreso than other metrics. SIM is bounded on [0, 1].

- AUC-J measures the area under an ROC curve. Since the heatmap is normalized by range, each pixel value in the predicted heatmap can be thought of as a predictor of if that pixel is on in the ground truth. At multiple thresholds, the Receiver Operating Characteristic (ROC) measures the true and false positive rate. AUC takes the area under that curve. AUC tends to not penalize predictions with many low-valued false positives. AUC is also bounded on [0, 1].

5. Results and Analysis

5.1 Model Robustness

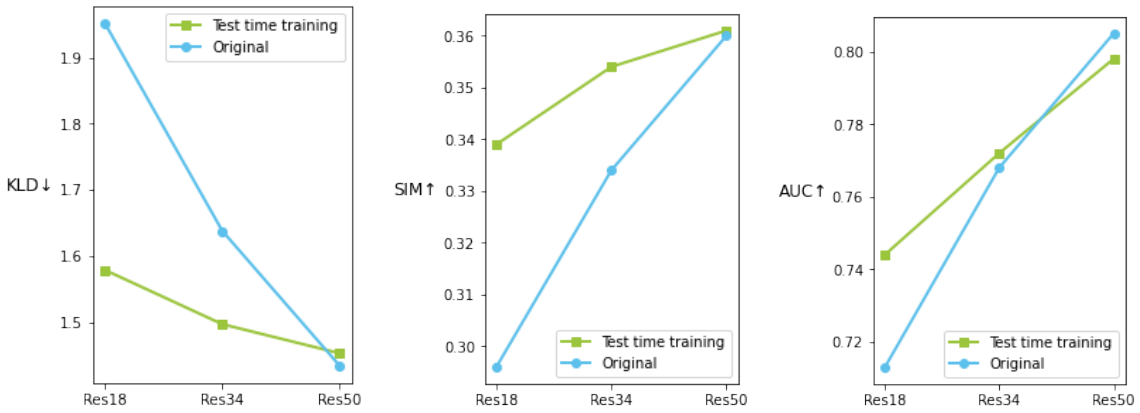


Figure 5.1: OPRA dataset comparison. Blue is the baseline model and green is with test-time training.

Given smaller backbone architecture sizes, we see a clear decrease in performance across all metrics. Although a decrease in performance is expected with decreasing model size, the magnitude of degradation is surprisingly large. We theorize the reason for this stark decrease in performance is due to the the change in quality of the featurization of the images. Since each backbone is frozen during training, the features of the ResNets are fixed and the other networks (LSTM and anticipation network) simply transform those fixed features. Larger models in this case will have better featurization when pretrained on ImageNet, and we see this is the case in the results. Since the number of output features is also greater in ResNet50, there are more fixed features to transform. In addition, due to ResNet50 having more features, the size of the anticipation network and LSTM are significantly bigger as well. All of these factors contribute to better performance of the ResNet50 model despite being similar in parameter count to ResNet34.

Models	KLD ↓	SIM ↑	AUC-J ↑
ResNet 18	1.952	0.296	0.713
ResNet 18T	**1.579**	**0.339**	**0.744**
ResNet 34	1.638	0.334	0.766
ResNet 34T	**1.497**	**0.354**	**0.772**
ResNet 50	**1.434**	0.360	**0.805**
ResNet 50T	1.453	**0.361**	0.798

Table 5.1: Comparison of performance for different backbone models. Each model was trained for 20 epochs. T at the end of the model denotes Test-Time Training performance where the anticipation network is updated for each new example. ↓ and ↑ represent where lower and higher is better, respectively.

Both Figure 5.2 and Table 5.1 show the performance improvements to each of the three models with the use of test-time training of the anticipation network. As expected, the biggest gains come from the smallest model - with ResNet 18T having better metrics in KLD and SIM than ResNet34 whilst having no additional parameters. ResNet 34T also shows modest improvement over the baseline ResNet 34. Surprisingly the performance of ResNet 50T slightly degrades across the metrics in comparison for ResNet 50. Overall, it seems that this method experiences diminishing returns with better model backbones. Since featurization is worse in smaller models, the anticipation network seems to compensate the performance.

An important note is that the trend of model sizes yielding improvements seems to break down after ResNet 50. Training with a ResNet 101 backbone has worse performance across all metrics (KLD: 1.580, SIM: 0.334, AUC-J: 0.768). As a result, experiments were only run for models of smaller than size ResNet101.

5.2 Qualitative Improvements

Figure 5.2: The result of updating the anticipation network on individual examples. Red, blue, and green heatmaps are for actions hold, push, and rotate, respectively. The left-most image is the original prediction, each successive image towards the right are the computed heatmaps after each additional update to the anticipation network.

5.3 Out of Distribution Generalization

The goal of this section is to measure the performance of the method to adapt to different domains than the data that was trained on. For OPRA, the domains can be split in multiple different ways such as by object type, action performed, and video domain. Primarily we focus on splitting domains by noun.

5.3.1 Object Generalization

Baseline	KLD ↓	SIM ↑	AUC-J ↑
Full Test Set	2.176	0.301	0.755
Seen Objects	2.070	0.302	0.766
Unseen Objects	2.255	0.280	0.749
Test-Time Training	**KLD ↓**	**SIM ↑**	**AUC-J ↑**
Full Test Set	1.557	0.336	0.757
Seen Objects	1.518	0.346	0.764
Unseen Objects	1.611	0.329	0.754

Table 5.2: Performance of the baseline ResNet18 model versus the ResNet18T model which uses test-time training. ↓ and ↑ represent where lower and higher is better, respectively.

To view generalization performance, we opt to use the ResNet18 as the backbone to test since its performance degraded the most after removing half of the object classes from training.

We see significant performance decrease across the whole test-set when removing half of the objects from test, including objects *seen* during training. As expected, unseen objects during test perform the worst.

The model was next evaluated for test-time training. Model parameters are updated the same as in Section 5.1 and described in Section 4.2. As shown in Table 5.2, the use of test-time training sees significant benefits to the performance of the model across all metrics with KLD improving the most and AUC-J improving the least. We attribute the domain adaptation performance to the anticipation network due to the action classifier still having an even split for action verbs. Effectively the model still performs well on classifying the action of a novel video, but the features between the projected static image and the active video frame may be very different due to the shallow size of the anticipation network. Improving the performance of the anticipation network per-example allows the model to effectively keep its

action classification performance while providing a better model for heatmap generation.

5.4 Failure Modes

There are a few ways that this method can fail. Misattribution of what is causing an action is by far the most prevalent failure. As shown in Figure 5.3, to grind coffee, a person must both hold the handle as well as the side of then grinder to rotate it. While the handle is the only part being rotated, all demonstrations also require holding. This entanglement between actions consequently will pose challenges deducing which part of an object really affords the action.

Cluttered scenes are another way in which this method can fail. Without one main object to project features, the anticipation network will assign affordances across many objects. Due to OPRA being human-annotated, the inherent ambiguity of which of many objects afford an action could cause a drop in performance.

Figure 5.3: Coffee grinder. The model incorrectly attributes the side of the grinder to both the actions hold and rotate.

Finally, object affordance predicted heatmaps can also be incorrect or incomplete due to inherent limitations in grounded human demonstration. Over the OPRA dataset, most actions involve what the demonstrator's hands do. As a result, parts of an object that may afford an action can be occluded by the hand. Fine-tuning on a demonstration that occludes key parts of an object may in fact result in decreased performance. Differences in object perspective between video and static images also can cause a drop in performance.

6. Future Work

Despite the benefits of the proposed method, performance gaps between weak and strong supervised approaches still exist. Further work in weakly supervised approaches possibly could close this performance gap between the two approaches. From experimentation, it is apparent that temporal information of the video frames are not fully leveraged. Further exploration in this realm may either show better methods to utilize time or highlight that time is not necessary to learn affordances.

Next, the anticipation network of this framework represents a significant portion of the total parameters of the model. More exploration on the performance trade offs between the number of parameters in each part of the frame work may prove to increase overall efficiency of the framework.

While built off the current state-of-the-art weakly-supervised method, this method still struggles with dichotomy between the training objective and the final evaluation metrics. For example, training the current model *including* the backbone results in significantly lower train and test loss but significantly worse final performance compared to training with a frozen backbone.

While the current evaluation metrics are good, they are not extensive. For example, while KL is not highly correlated with SIM and AUC, SIM and AUC are moderately correlated. Furthermore, more metrics focusing on precision and recall of the method could highlight further strengths and failure modes of the method.

While both datasets provide a strong foundation for affordance prediction, they also have some limitations. Both fully-annotated datasets are relatively small in comparison to common image or video-based datasets. More actions, nouns, or environments would also allow methods to be shown that they are generalizable across object types, interactions, and scenes.

Learning affordances can also lead to efficiencies in other deep learning tasks. For example, the task of action classification is very computationally heavy due to the use of a time dimension as well as the use of 3D model architectures. Since actions happen around interactions with objects and the

environment, narrowing the spatial scope that a model looks at to just around afforded regions poses potential opportunities for efficiency improvements.

Object affordances could also influence robotic interaction with it's environment. Novel object grasping is a popular task in vision-based reinforcement learning and ties in well with the idea of object affordances. An agent must implicitly learn what parts an object afford being held as well as other actions if needed. Having model-based affordance prediction may yield improvements in learning efficiency and overall generalization.

7. Conclusion

This method proposes an alteration of an existing framework to benefit from test-time training. We claim that object affordance prediction can be thought as primarily a similarity learning problem between a single "inactive" image containing an object and an "active" video of that object being interacted with.

As a result of this thought paradigm, we can construct an unsupervised similarity learning objective that allows for the proposed model to be trained on examples during test-time while still yielding performance benefits.

We show performance benefits to applying this approach to weaker backbone and anticipation models, showing this method can be an alternative to simply increasing model size. As a result, more efficient affordance prediction models are possible and potentially open the door to mobile and embedded applications.

We also show that test-time training allows the model to be more generalizable and be robust to domain shifts. Through holding out a large percentage of objects in training, we show that the framework strongly improves results.

While requiring no-extra parameters, this method improves among existing methods in a simple and intuitive way. Allowing for greater robustness across backbone models and improving generalizability gives this method more applicability in areas where compute is more limited or where there is limited training data, and further closes the gap between weak and strong supervised approaches.

Bibliography

[1] Jean-Baptiste Alayrac, Josev Sivic, Ivan Laptev, and Simon Lacoste-Julien. *Joint Discovery of Object States and Manipulation Actions*. 2017. eprint: arXiv:1702.02738.

[2] David Bau, Hendrik Strobelt, William Peebles, Jonas, Bolei Zhou, Jun-Yan Zhu, and Antonio Torralba. "Semantic Photo Manipulation with a Generative Image Prior". In: (2020). DOI: 10.1145/3306346.3323023. eprint: arXiv:2005.07727.

[3] Roman Beliy, Guy Gaziv, Assaf Hoogi, Francesca Strappini, Tal Golan, and Michal Irani. *From voxels to pixels and back: Self-supervision in natural-image reconstruction from fMRI*. 2019. eprint: arXiv:1907.02431.

[4] Sefi Bell-Kligler, Assaf Shocher, and Michal Irani. *Blind Super-Resolution Kernel Estimation using an Internal-GAN*. 2019. eprint: arXiv:1909.06581.

[5] Ali Borji and Laurent Itti. "CAT2000: A Large Scale Fixation Dataset for Boosting Saliency Research". In: *CVPR 2015 workshop on "Future of Datasets"* (2015). arXiv preprint arXiv:1505.03581.

[6] Zoya Bylinskii, Tilke Judd, Ali Borji, Laurent Itti, Frédo Durand, Aude Oliva, and Antonio Torralba. *MIT Saliency Benchmark*. saliency.mit.edu.

[7] Zoya Bylinskii, Tilke Judd, Aude Oliva, Antonio Torralba, and Frédo Durand. "What do different evaluation metrics tell us about saliency models?" In: *arXiv preprint arXiv:1604.03605* (2016).

[8] Claudio Castellini, Tatiana Tommasi, Nicoletta Noceti, Francesca Odone, and Barbara Caputo. "Using object affordances to improve object recognition". In: *IEEE transactions on autonomous mental development* 3.3 (2011), pp. 207–215.

[9] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. *A Simple Framework for Contrastive Learning of Visual Representations*. 2020. eprint: arXiv:2002.05709.

[10] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. "Scaling Egocentric Vision: The EPIC-KITCHENS Dataset". In: *European Conference on Computer Vision (ECCV)*. 2018.

[11] Dima Damen, Teesid Leelasawassuk, Osian Haines, Andrew Calway, and Walterio Mayol-Cuevas. "You-Do, I-Learn: Discovering Task Relevant Objects and their Modes of Interaction from Multi-User Egocentric Video". In: *Proceedings of the British Machine Vision Conference*. BMVA Press, 2014.

[12] Kuan Fang, Te-Lin Wu, Daniel Yang, Silvio Savarese, and Joseph J. Lim. "Demo2Vec: Reasoning Object Affordances From Online Videos". In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2018.

[13] James J Gibson. *Ecological approach to visual perception*. Houghton Mifflin, 1985.

[14] James J Gibson. *The Senses Considered as Perceptual Systems*. Allen and Unwin, London, 1966.

[15] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. *Unsupervised Representation Learning by Predicting Image Rotations*. 2018. eprint: `arXiv:1803.07728`.

[16] Daniel Gordon, Kiana Ehsani, Dieter Fox, and Ali Farhadi. *Watching the World Go By: Representation Learning from Unlabeled Videos*. 2020. eprint: `arXiv:2003.07990`.

[17] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. *Momentum Contrast for Unsupervised Visual Representation Learning*. 2019. eprint: `arXiv:1911.05722`.

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. *Deep Residual Learning for Image Recognition*. 2015. eprint: `arXiv:1512.03385`.

[19] Tucker Hermans, James M Rehg, and Aaron Bobick. "Affordance prediction via learned object attributes". In: Citeseer.

[20] Yifei Huang, Minjie Cai, Zhenqiang Li, and Yoichi Sato. *Predicting Gaze in Egocentric Video by Learning Task-dependent Attention Transition*. 2018. eprint: arXiv:1803.09125.

[21] Joshua Knights, Anthony Vanderkop, Daniel Ward, Olivia Mackenzie-Ross, and Peyman Moghadam. *Temporally Coherent Embeddings for Self-Supervised Video Representation Learning*. 2020. eprint: arXiv:2004.02753.

[22] Hema S Koppula and Ashutosh Saxena. "Physically grounded spatio-temporal object affordances". In: *European Conference on Computer Vision*. Springer. 2014, pp. 831–847.

[23] Hema Swetha Koppula, Rudhir Gupta, and Ashutosh Saxena. "Learning human activities and object affordances from rgb-d videos". In: *The International Journal of Robotics Research* 32.8 (2013), pp. 951–970.

[24] Matthias Kümmerer, Thomas S. A. Wallis, and Matthias Bethge. *DeepGaze II: Reading fixations from deep features trained on object recognition*. 2016. eprint: arXiv:1610.01563.

[25] Michael Land, Neil Mennie, and Jennifer Rusted. "The Roles of Vision and Eye Movements in the Control of Activities of Daily Living". In: *Perception* 28.11 (1999). PMID: 10755142, pp. 1311–1328. DOI: 10.1068/p2935. eprint: https://doi.org/10.1068/p2935. URL: https://doi.org/10.1068/p2935.

[26] Yin Li, Alireza Fathi, and James M Rehg. "Learning to predict gaze in egocentric video". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2013, pp. 3216–3223.

[27] Ji Lin, Chuang Gan, and Song Han. *TSM: Temporal Shift Module for Efficient Video Understanding*. 2018. eprint: arXiv:1811.08383.

[28] Ravi Teja Mullapudi, Steven Chen, Keyi Zhang, Deva Ramanan, and Kayvon Fatahalian. "Online Model Distillation for Efficient Video Inference". In: (2018). eprint: arXiv:1812.02699.

[29] A. Myers, C. L. Teo, C. Fermüller, and Y. Aloimonos. "Affordance detection of tool parts from geometric features". In: *2015 IEEE International Conference on Robotics and Automation (ICRA)*. 2015.

[30] Tushar Nagarajan, Christoph Feichtenhofer, and Kristen Grauman. *Grounded Human-Object Interaction Hotspots from Video*. 2018. eprint: arXiv:1812.04558.

[31] Tushar Nagarajan, Yanghao Li, Christoph Feichtenhofer, and Kristen Grauman. *EGO-TOPO: Environment Affordances from Egocentric Video*. 2020. eprint: arXiv:2001.04583.

[32] Donald A. Norman. *The Design of Everyday Things*. USA, 1985.

[33] Mehdi Noroozi and Paolo Favaro. *Unsupervised Learning of Visual Representations by Solving Jigsaw Puzzles*. 2016. eprint: arXiv:1603.09246.

[34] Junting Pan, Cristian Canton Ferrer, Kevin McGuinness, Noel E. O'Connor, Jordi Torres, Elisa Sayrol, and Xavier Giro-i-Nieto. *SalGAN: Visual Saliency Prediction with Generative Adversarial Networks*. 2017. eprint: arXiv:1701.01081.

[35] Florian Schroff, Dmitry Kalenichenko, and James Philbin. "FaceNet: A Unified Embedding for Face Recognition and Clustering". In: (2015). DOI: 10.1109/CVPR.2015.7298682. eprint: arXiv:1503.03832.

[36] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. "Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization". In: (2016). DOI: 10.1007/s11263-019-01228-7. eprint: arXiv:1610.02391.

[37] Tamar Rott Shaham, Tali Dekel, and Tomer Michaeli. *SinGAN: Learning a Generative Model from a Single Natural Image*. 2019. eprint: arXiv:1905.01164.

[38] Assaf Shocher, Shai Bagon, Phillip Isola, and Michal Irani. *InGAN: Capturing and Remapping the "DNA" of a Natural Image*. 2018. eprint: arXiv:1812.00231.

[39] Assaf Shocher, Nadav Cohen, and Michal Irani. *"Zero-Shot" Super-Resolution using Deep Internal Learning*. 2017. eprint: arXiv:1712.06087.

[40] Yu Sun, Eric Tzeng, Trevor Darrell, and Alexei A. Efros. *Unsupervised Domain Adaptation through Self-Supervision*. 2019. eprint: `arXiv:1909.11825`.

[41] Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei A. Efros, and Moritz Hardt. *Test-Time Training for Out-of-Distribution Generalization*. 2019. eprint: `arXiv:1909.13231`.

[42] Spyridon Thermos, Georgios Th. Papadopoulos, Petros Daras, and Gerasimos Potamianos. *Deep Affordance-grounded Sensorimotor Object Recognition*. 2017. eprint: `arXiv:1704.02787`.

[43] Yonglong Tian, Dilip Krishnan, and Phillip Isola. *Contrastive Multiview Coding*. 2019. eprint: `arXiv:1906.05849`.

[44] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. *A Closer Look at Spatiotemporal Convolutions for Action Recognition*. 2017. eprint: `arXiv:1711.11248`.

[45] Carl Vondrick, Abhinav Shrivastava, Alireza Fathi, Sergio Guadarrama, and Kevin Murphy. *Tracking Emerges by Colorizing Videos*. 2018. eprint: `arXiv:1806.09594`.

[46] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. *Temporal Segment Networks: Towards Good Practices for Deep Action Recognition*. 2016. eprint: `arXiv:1608.00859`.

[47] Xiaolong Wang, Allan Jabri, and Alexei A. Efros. *Learning Correspondence from the Cycle-Consistency of Time*. 2019. eprint: `arXiv:1903.07593`.

[48] Bolei Zhou, Alex Andonian, Aude Oliva, and Antonio Torralba. *Temporal Relational Reasoning in Videos*. 2017. eprint: `arXiv:1711.08496`.