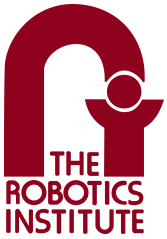
Visual Representation and Recognition without Human Supervision

Technical Report Number: CMU-RI-TR-22-21

Senthil Purushwalkam

May 2022



The Robotics Institute
School of Computer Science
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

Thesis Committee

Abhinav Gupta	Carnegie Mellon University (*chair*)
Deva Ramanan	Carnegie Mellon University
David Held	Carnegie Mellon University
Kristen Grauman	University of Texas at Austin
Alexei Efros	University of California, Berkeley

Thesis submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Robotics

© Senthil Purushwalkam, 2022

Abstract

The advent of deep learning based artificial perception models has revolutionized the field of computer vision. These methods take advantage of the ever-growing computational capacity of machines and the abundance of human-annotated data to build supervised learners for a wide-range of visual tasks. However, the reliance on human-annotated is also a bottleneck for the scalability and generalizability of these methods. We argue that in order to build more general learners (akin to an infant), it is crucial to develop methods that learn without human-supervision. In this thesis, we present our research on minimizing the role of human-supervision for two key problems: *Representation* and *Recognition*.

Recent self-supervised representation learning (SSL) methods have demonstrated impressive generalization capabilities on numerous downstream tasks. In this thesis, we investigate these approaches and demonstrate that they still heavily rely on the availability of clean, curated and structured datasets. We experimentally demonstrate that these learning capabilities fail to extend to data collected "in-the-wild" and hence, expose the need for better benchmarks in self-supervised learning. We also propose novel SSL approaches that minimize this dependence on curated data.

Since exhaustively collecting annotations for all visual concepts is infeasible, methods that generalize beyond the available supervision are crucial for building scalable recognition models. We present a novel neural network architecture that takes advantage of the compositional nature of visual concepts to construct image classifiers for unseen concepts. For domains where collecting dense annotations is infeasible, we present an "understanding via associations" paradigm which reformulates the recognition problem as identification of correspondences. We apply this to videos and show that we can densely describe videos by identifying dense spatio-temporal correspondences to other similar videos. Finally, to explore the human ability of generalizing beyond semantic categories, we introduce the "Functional Correspondence Problem" and demonstrate that representations that encode functional properties of objects can be used to recognize novel objects more efficiently.

Acknowledgements

My PhD journey towards producing this thesis has been far from a solitary endeavor. I have had the privilege of receiving help from some of the best research minds, and the support of friends and family.

I am very grateful for the mentorship of my advisor, Abhinav Gupta. He has guided me through 7 years of research, through numerous succesful and failed projects with an unwavering faith in my abilities. His ability to cut through the chaos, think about the big picture and constructively critique any work is something I strive to emulate. He has taught me to set higher standards for research and more importantly, to be "bored" of trivial research. I am grateful for having the opportunity to learn from him.

Over the course of my graduate school, I have been lucky to have several other amazing mentors. I am grateful to Marc'Aurelio Ranzato for giving me the opportunity to work with him during an internship. He taught me the value of having clear goals, structuring your thoughts and conducting thorough experiments. Bryan Russell for giving me the confidence and opportunity to explore crazy ideas (like writing a paper on bouncing balls). Thank you for introducing me to GTD which has become an invaluable part of my life and has signficantly reduced the stress of graduate school. Kristen Grauman for showing me that progress in research can be very systematic and well planned. She also taught me that writing papers early makes a huge difference in the quality of content you produce. I'm also grateful to Dhruv Batra who introduced me to research in Computer Vision. His infectious passion for research is responsible for my motivation to pursue graduate school. I would like to thank my thesis committee - Deva Ramanan, David Held, Alyosha Efros and Kristen Grauman for their time and valuable discussions. I am also thankful for all my amazing collaborators for their contributions - Saurabh Gupta, Pedro Morgado, Zihang Lai, Tian Ye, Maximilian Nickel, Sebastian Vicenc Amengual Gari, Vamsi Krishna Ithapu, Carl Schissler, Philip Robinson, Danny Kaufman. A lot of my research has heavily relied on drawing from their expertise. I would like to thank the administrative staff at CMU for their support, including Suzanne Muth, Christine Downey, Jess Butterbaugh and Alison Day.

Members of my lab and Smith Hall have always been a source of guidance, feedback, constructive criticism and a lot of fun! I'm grateful for all the discussions about research and more with Achal Dave, Kenneth Marino, Nadine Chang, Aayush Bansal, Debidatta Dwibedi, Shubham Tulsiani, Deepak Pathak, Nilesh Kulkarni, David Fouhey, Gaurav Pathak, Allie Del Giorno, Rohit Girdhar, Lerrel Pinto, Adithya Murali, Jacob Walker, Xiaolong Wang, Sudeep Dasari, Shikhar Bahl, Yufei Ye, Dinesh Reddy, Tanmay Shankar, Arjun Sharma, Khushi Gupta, Gunnar Atli Sigurdsson, Pavel Tokmakov, Nick Rhinehart, Dhiraj Gandhi, Sam Powers, Victoria Dean, Helen Jiang, Yin Li, Xinlei Chen, Olga Russakovsky, Martin Li, Xiaofang Wang, Jason Zhang, Peiyun Hu, Chen-Hsuan Lin, Minyoung Huh, Pratyusha Sharma, Mohit Sharma,

Jianren Wang, Tao Chen and Wenxuan Zhou.

I have also made some great friends over the last 7 years who have helped me enjoy life outside of work in Pittsburgh. I'm grateful for therapy dinners with Achal Dave, Nadine Chang and Kenneth Marino. For super smash bros with Achal, Kenny, Nadine, Nick, Kit, Gunnar and others in Smith Hall. For rocket league with Rohit, Achal, Ankit and Dinesh which provided a great respite during the isolating pandemic. I would also like to thank the people who made living in Pittsburgh fun through parties, coffee chats, video games, skiing and trying out sketchy food joints: Anirudh Vemula, Pragna Mannam, Puneet Puri, Vishal Dugar, Ankit Bhatia, Adithya Murali, Alex Spitzer, Dhruv Saxena, Xuning Yang, Achal Dave, Debidatta Dwibedi, Shefali Umrania, Arjun Sharma, Rosario Scalise, Jaskaran Singh, Wony Hong, Mihir Hasabnis, Junichi Koganemaru, Shohin Mukherjee, Brian Okorn, Rohit Garg, Dinesh Reddy, Tanmay Shankar, Paloma Sodhi, Sanjiban Choudhury and Lerrel Pinto.

I am deeply grateful for my partner, Ananya Uppal, who has been through the highs and lows with me. She has helped me grow as a person, helped me set priorities in life and is still trying to teach me the important skill of "doing nothing". Our cooking sessions, road trips and exploring restaurants together (and the snooty critiquing) have made this journey more memorable than I could have imagined. I'm also thankful for the dogs we fostered together who brought enough joy in our lives to make the pandemic seem like just a few days. Last but far from the least, I would like to thank my family. My dad, Shiva Prakash Purushwalkam, who prioritized his education through extreme adversities has inspired me to pursue academic accomplishments. My mom, Kalpana Shiva Prakash, constantly impresses and inspires me with her creative skills. My interest in science and technology arose out of sibling competition with my brother, Sumanth Purushwalkam. He has always faced the brunt of exploring new opportunities and paved the easy path for me through his experience.

Contents

Chapter 1

Introduction

Visual perception is arguably the least understood component of human intelligence. Humans (and most animals) possess the remarkable ability to generate meaningful interpretations from the images sensed by the eyes. We unconsciously identify low level cues like depth and object boundaries, and also instantaneously perform complex reasoning tasks like recognizing semantics of objects and scenes, anticipating hidden 3D structure, anticipating motion of objects (including other humans), etc. In order to perform these tasks, we rely on knowledge and experience gathered through our exploration of the world (see [84]). Perhaps a more impressive attribute is the efficiency in amount of additional knowledge required for learning novel concepts. For example, we can generally interact with unseen objects without having observed it in the past. We can even identify previously unseen objects simply based on a description. Understanding and imitating these learning capabilities has been an elusive goal of artificial intelligence research for many decades.

Nevertheless, the field of Computer Vision has made tremendous strides towards building artificial models to perceive and understand the visual world. Approaches for detecting objects, recognizing faces and recognizing human actions have been successfully deployed around us in our everyday lives. The success can be largely attributed to the ever-increasing computational power available to us in conjunction with the development of artificial perception models using deep learning. However, there is another (sometimes) under-appreciated actor behind the success stories in AI: *human supervised data*.

In [119], deep neural networks were brought to the forefront of AI research by demonstrating an impressive leap in image classification performance. Prior to this work, the well-established approaches for solving computer vision tasks relied on expert designed features to extract the relevant information in images. In contrast, in [119] it was shown that deep neural networks can be directly optimized on large-scale datasets to learn extremely powerful and discriminative representations. Following the success of [119] on the large-scale human supervised dataset ImageNet[37], similar datasets have emerged for most visual recognition

tasks like Object Detection[136], Action Recognition[57, 111, 214] and even robotic grasping[181]. Most of the recent success in Computer Vision has been a result of exploring deep learning based architectures on these supervised datasets.

More broadly, human supervised data has been extensively used in Computer Vision to address two important related problems:

- ***Representation****: The problem of encoding visual signals by extracting relevant information such that downstream reasoning tasks can be made simpler.*
  Convolutional neural networks (CNNs) optimized on large-scale human-annotated datasets simultaneously learn to represent visual signals in the intermediate layers. These representations have been shown to possess remarkable generalization capabilities[27, 71]. For almost a decade now, state-of-the-art approaches for most computer vision tasks have relied on a representation constructed by optimizing a CNN for classifying the ImageNet dataset.
- ***Recognition****: The problem of reasoning about the encoded visual signals to identify visual concepts based on a desired semantic categorization.*
  Some common recognition problems addressed in computer vision research are image classification, object detection/segmentation and action recognition. Several large-scale supervised datasets have been collected for each of these problems, depicting images/videos with human annotations from a predefined vocabulary. State-of-the-art recognition models[27, 71, 72, 92] generally involve neural network architectures that are desiged for the specific task, whose parameters optimized using the supervision from the human annotations.

Despite this progress, the capabilities of our current representation and recognition models are significantly inferior compared to human perception systems. One question that naturally arises is: can we simply collect larger human-annotated datasets to make progress towards more robust and general visual perception models?

Limited Scale of Supervised Datasets
The scale of the datasets we currently use in Computer Vision is negligible compared to the amount of visual data available on the web or the amount of visual data observed in the life of an infant. The primary reason for this is the cost (in time and money) of annotating images and videos. For reference, collecting the outline of a single object takes between 54 seconds and 79 seconds[14]. Recent studies show that we upload 400 million pictures every day and about 300 hours of video content every minute. Based on these observations, it is clearly impractical to attempt to annotate any significant fraction of this data. This is also evident from the fact that over the last decade, we have made limited progress towards a dataset larger than ImageNet[37]. Therefore, purely relying on human-supervised datasets significantly constrains the amount of knowledge that can be leveraged by our learning algorithms.

Limited Taxonomy of Supervised Datasets:
Annotating samples also requires curating a collection of images or videos drawn from all the data available to us. Since visual concepts follow a long-tailed distribution[234],

these sampled sets do not cover a large number of infrequent concepts. Furthermore, curation of datasets has generally required predefining a vocabulary of relevant concepts (either for curating samples or for usage as target labels). This focus on small vocabularies restricts the ability of our models to develop any general understanding of the concepts in the real world. For example, the most common object detection models are built on the MS-COCO[136] allowing us to detect only 80 object categories.

Learning beyond Supervisable Concepts
Current deep learning methods demonstrate impressive performance for learning from supervised datasets and generalizing to unseen instances of the supervised concepts. However, as alluded to earlier, human perception capabilities stretch far beyond identifying learned concepts. We learn to identify most visual concepts without any external supervision[108]. We learn to imagine/identify unseen concepts described as a composition of learned concepts (for example, you can imagine a red elephant). We can identify concepts that have no associated names. We can learn new concepts with as few as a single observation. At the moment, it is unclear how these capabilities can be learned in a supervised manner.

Due to these challenges, it has become clear that we can rely on humans only for limited amounts of supervision. In this thesis, we present our research on minimizing the role of human supervision for the *representation* and *recognition* problems.

1.1 Overview of the Thesis

This thesis is divided into three parts: (I) Learning Representations without Supervision, (II) Semantic Recognition beyond Supervision and (III) Representation Beyond Semantics.

Part I: Learning Representations without Supervision

We first study the problem of constructing better representations without human-supervision. This problem is addressed in literature as self-supervised representation learning, a learning paradigm where supervision is automatically obtained from the data without the need for human annotation. Research in this direction has explored innovative approaches for obtaining supervision like predicting relative locations of patches[47], colorizing images[271], matching temporally tracked patches[247] and predicting motion cues[177, 183]. More recently, the self-supervised representations learned through instance classification[29–31, 91, 97, 173, 184, 256] have demonstrated great success, even outperforming ImageNet-based representations on some tasks[23].

In Chapter 2, we investigate the efficacy of the representations learned by contrastive learning methods to develop a deeper understanding. We present a framework to quantify the invariances learned in these representations. We show that

the artificial augmentations used in these approaches, lead to improved occlusion invariance compared to an ImageNet-based representation. However, they suffer at learning other invariances. Based on these inferences, we propose an approach to improve invariances in these representations by using the naturally occurring transformations in videos.

In Chapter 3, we further investigate the efficacy of representations learned using contrastive learning in more challenging training setups. We observe that while these methods produce strong representations, they rely on the clean curated structure of the ImageNet dataset. In realistic in-the-wild setups, contrastive learning methods are faced with three challenges: data and computational inefficiency, non-IID data and non-stationary semantic distributions. As a first step to remedy these issues, we propose simple replay buffer based approaches that demonstrate superior performance on downstream tasks compared to off-the-shelf contrastive learning methods.

Part II: Semantic Recognition beyond Supervision

Recognizing semantic concepts like objects and actions is an important problem in computer vision with innumerable applications. However, there are practically infinite number of concepts that occur in our visual world. The traditional method of collecting supervised data for the relevant concepts is not scalable to capture the diversity of concepts. In Chapter 4, we propose an approach to build models to recognize a large number of visual concepts beyond those concepts where supervision is available. We begin by observing that the visual world is highly compositional. For example, an "old tree" is the composition of the "old" and "tree" concepts and an "old car" is the composition of the same "old" concept with the "car" concept. We present an approach for image classification in the zero-shot learning setting that leverages this compositional nature of object-attribute concepts to build classifiers for unseen concepts. We present a modular neural network that captures the rich interactions between instances, objects and attributes.

For some recognition problems, collecting annotations is not merely an expensive endeavor, but could also be infeasible. For example, consider the problem of building a dense "understanding" of videos. Most related computer vision methods deal with identifying coarse high-level semantics like action labels, temporal location of actions, language descriptions, etc. However, a dense understanding would involve explaining every pixel, patch, object part, etc. Clearly, all such concepts to be recognized can not even be enumerated here. In Chapter 5, we present a solution for dense video understanding using a "understanding via associations" paradigm. The key insight is that videos can be described by finding dense spatial and temporal correspondences to other videos. We present a weakly supervised cycle-consistency loss that only uses video level category labels. We show that our method can improve on ImageNet-based representations and identify the correspondences, thereby eliminating the need for any dense human annotations.

Part III: Representation beyond Semantics

Visual recognition problems in computer vision have primarily focused on identifying semantic categories. One way to interpret this goal is to find similarities or correspondences between instances of a category. However, humans possess the ability to reason beyond semantic categories. For example, if we have to, we can use a shoe to hammer a nail. Here, we are able to identify correspondences between a shoe and a hammer - the corresponding grasp location, the corresponding head used to hit and so on. In Chapter 6, we formulate the learning of this ability in Computer Vision as the *Functional Correspondence Problem*. We present the FunKPointdataset comprising on task-dependent keypoints for various objects. We leverage to learn a task-driven representation for objects and demonstrate that we can successfully identify functional correspondences between objects. Furthermore, we show that such functional representations generalize better for image-classification tasks in a few-shot learning setting.

Chapter 7: Conclusion and Future Research

Finally, in Chapter 7 we present a summary of the thesis. We also identify a few crucial future directions of research for making progress towards more general computer vision models that learn with minimal human guidance.

Part I

Learning Representations without Supervision

Chapter 2

Demystifying Contrastive Self-Supervised Learning: Invariances, Augmentations and Dataset Biases

2.1 Introduction

Inspired by biological agents and necessitated by the manual annotation bottleneck, there has been growing interest in self-supervised visual representation learning. Early work in self-supervised learning focused on using "pretext" tasks for which ground-truth is free and can be procured through an automated process [48, 247]. Most pretext tasks include prediction of some hidden portion of input data (e.g., predicting future frames [173] or color of a grayscale image [271]). However, the performance of the learned representations have been far from their supervised counterparts.

The past six months have been revolutionary in the field of self-supervised learning. Several recent works [29, 91, 97, 158, 173] have reported significant improvements in self-supervised learning performance and now surpassing supervised learning seems like a foregone conclusion. So, what has changed dramatically? The common theme across recent works is the focus on the instance discrimination task [52] – treating every instance as a class of its own. The image and its augmentations are positive examples of this class; all other images are treated as negatives. The contrastive loss[97, 173] has proven to be a useful objective function for instance discrimination, but requires gathering pairs of samples belonging to the same class (or instance in this case). To achieve this, all recent works employ an "aggressive" data augmentation strategy where numerous samples can be generated from a single image. Instance discrimination, contrastive loss and aggressive augmentation are

the three key ingredients underlying these new gains.

While there have been substantial gains reported on object recognition tasks, the reason behind the gains is still unclear. Our work attempts to demystify these gains and unravel the hidden story behind this success. The utility of a visual representation can be understood by investigating the invariances (see Section 2.4.1 for definition) it encodes. First, we identify the different invariances that are crucial for object recognition tasks and then evaluate two state of the art contrastive self-supervised approaches [91, 158] against their supervised counterparts. Our results indicate that a large portion of the recent gains come from occlusion invariances. The occlusion invariance is an obvious byproduct of the aggressive data augmentation which involves cropping and treating small portions of images as belonging to the same class as the full image. When it comes to viewpoint and category instance invariance there is still a gap between the supervised and self-supervised approaches.

Occlusion invariance is a critical attribute for useful representations, but is artificially cropping images the right way to achieve it? The contrastive loss explicitly encourages minimizing the feature distance between positive pairs. In this case, the pair would consist of two possibly non-overlapping cropped regions of an image. For example, in the case of an indoor scene image, one sample could depict a chair and another could depict a table. Here the representation would be forced to be bad at differentiating these chairs and tables - which is intuitively the wrong objective! So why do these approaches work? We hypothesize two possible reasons: (a) The underlying biases of pre-training dataset - Imagenet is an object-centric dataset which ensures that different crops correspond to different parts of same object; (b) the representation function is not strong enough to achieve this faulty objective, leading to a sub-optimal representation which works well in practice. We demonstrate through diagnostic experiments that indeed the success of these approaches originates from the object-centric bias of the training dataset. This suggests that the idea of employing aggressive synthetic augmentations must be rethought and improved in future work to ensure scalability.

As a step in this direction, in this paper, we argue for usage of a more natural form of data for the instance discrimination task: videos. We present a simple method for leveraging transformations occurring naturally in videos to learn representations. We demonstrate that leveraging this form of data leads to higher viewpoint invariance when compared to image-based learning. We also show that the learned representation outperforms MoCo-v2 [30] trained on the same data in terms of viewpoint invariance, category instance invariance, occlusion invariance and also demonstrates improved performance on object recognition tasks.

2.2 Related Work

A large body of research in Computer Vision is dedicated to training feature extraction models, particularly deep neural networks, without the use of human-annotated

Figure 2.1: Aggressive Augmentation Constrastive self-supervised learning methods employ an aggressive cropping strategy to generate positive pairs. Through this strategy, an image (left) yields many non-overlapping crops (right) as samples. We can observe that the crops do not necessarily depict objects of the same category. Therefore, a representation that matches features of these crops would be detrimental for downstream object recognition tasks.

data. These learned representations are intended to be useful for a wide range of downstream tasks. Research in this domain can be coarsely classified into generative modeling [46, 117, 132, 147, 223, 236] and self-supervised representation learning[48, 50, 68, 247].

Pretext Tasks Self-supervised learning involves training deep neural networks by constructing "pretext" tasks for which data can be automatically gathered without human intervention. Numerous such pretext tasks have been proposed in recent literature including predicting relative location of patches in images[48], learning to match tracked patches[247], predicting the angle of rotation in an artificially rotated image[68], predicting the colors in a grayscale image[271] and filling in missing parts of images[178]. These tasks are manually designed by experts to ensure that the learned representations are useful for downstream tasks like object detection, image classification and semantic segmentation. However, the intuitions behind the design are generally not verified experimentally due to the lack of a proper evaluation framework beyond the metrics of the downstream tasks. While we do not study these methods in our work, our proposed framework to understand representations (Section 2.4) can directly be applied to any representation. In many cases, it can be used to verify the motivations for the pretext tasks.

Instance Discrimination Most recent approaches that demonstrate impressive performances on downstream tasks involve training for *Instance Discrimination*. Dating back to [52], the task of instance discrimination involves treating an image and it's transformed versions as one single class. However, the computational costs of performing instance discrimination on large datasets had impeded it's applicability to larger deep neural networks. In NPID[256], the computational expense was

avoided using a non-parametric classification method leveraging a memory bank of instance representations. MOCO[91], MOCO-v2[30] adopted the contrastive learning framework(see Section 2.3) and maintain a queue of negative features which is updated at each iteration. PIRL[158] proposes learning of features which are invariant to the transformations proposed in "pretext" tasks and also uses the memory bank proposed in [256]. At the core, these approaches employ a common mechanism of generating samples for an instance's class - aggressively augmenting the initial image[29, 97, 173, 226].

SSL from Videos Self-supervised learning research has also involved leveraging videos for supervision [177, 247, 249, 250]. Specifically, approaches such as [247] and [249] attempt to encode viewpoint and deformation invariances by tracking objects in videos. [177] uses an off-the-shelf motion segmentation as the ground truth for training a segmentation model. Inspired by these works, we propose an approach that tracks regions using weaker self-supervised learning features and uses the tracks to learn better representations within the contrastive learning framework.

Understanding Self-Supervised Representations Self-supervised learning methods are evaluated by using the learned representations (either by finetuning or training an additional neural network) to perform numerous downstream tasks[79]. This evaluation framework provides a utilitarian understanding of the representations and fails to provide any insights about why a self-supervised learning approach works for a specific downstream task. There has been some research on developing a more fundamental understanding of the representations learned by deep neural networks in supervised settings [12, 76, 208, 209, 275].

We focus on representations learned by constrastive self-supervised learning methods. In [227], empirical evidence is provided showing that reducing the mutual information between the augmented samples, while keeping task-relevant information intact improves representations. In the context of object recognition, this implies that the category of the augmented sample (task-relevant information) should not change. In our work, we show that the common augmentation methods used in MOCO, MOCOv2, SimCLR, do not explicitly enforce this and instead rely on a object-centric training dataset bias (see Section 2.4.2). In [242], the contrastive loss is analyzed to show that it promotes two properties 'alignment' (closeness of features of positive pairs) and 'uniformity' (in the distribution of features on a hypersphere). In our work, we focus on understanding why the learned representations are useful for object recognition tasks. We study two aspects of the representations: 1) invariances encoded in the representations and their relation to the augmentations performed on images and 2) the role of the dataset used for training.

2.3 Contrastive Representation Learning

Contrastive learning [97, 173] is general framework for learning representations that encode similarities according to pre-deterimined criteria. Consider a dataset $\mathcal{D} = \{x_i | x_i \in \mathbb{R}^n, i \in [N]\}$. Let us assume that we have a way to sample positive

pairs $(x_i, x_i^+) \in \mathcal{D} \times \mathcal{D}$ for which we desire to have similar representations. We denote the set of all such positive pairs by $\mathcal{D}^+ \subset \mathcal{D} \times \mathcal{D}$. The contrastive learning framework learns a normalized feature embedding f by optimizing the following objective function:

$$\mathcal{L}(D, D^+) = - \sum_{(x,x^+) \in \mathcal{D}^+} \frac{\exp[\, f(x)^\intercal \, f(x^+)/\tau \,]}{\exp[\, f(x)^\intercal \, f(x^+)/\tau \,] + \sum\limits_{\substack{x^- \in \mathcal{D} \\ (x,x^-) \notin D^+}} \exp[\, f(x)^\intercal \, f(x^-)/\tau \,]} \tag{2.1}$$

Here τ is a hyperparameter called temperature. The denominator encourages discriminating negative pairs that are not in the positive set $\mathcal{D}^+$. In practice, this summation is expensive to compute for large datasets $\mathcal{D}$ and is performed over K randomly chosen negative pairs for each x. Recent works have proposed approaches to scale up the number of negative samples considered while retaining efficiency (see Section 5.2). In our experiments, we adopt the approach proposed in [30].

The contrastive learning framework relies on the ability to sample positive pairs (x_i, x_i^+). Self-supervised approaches have leveraged a common mechanism: each sample x is transformed using various transformation functions $t \in \mathrm{T}$ to generate new samples. The set of positive pairs is then considered as $\mathcal{D}^+ = \{(t_i(x), t_j(x)) \,|\, t_i, t_j \in \mathrm{T}, x \in \mathcal{D}\}$ and any pair $(t_i(x), t_k(x'))$ is considered a negative pair if $x \neq x'$.

The choice of transformation functions T controls the properties of the learned representation. Most successful self-supervised approaches [29, 30, 91, 256] have used: 1) cropping sub-regions of images (with areas in the range 20%-100% of the original image), 2) flipping the image horizontally, 3) jittering the color of the image by varying brightness, contrast, saturation and hue, 4) converting to grayscale and 5) applying gaussian blur. By composing these functions and varying their parameters, infinitely many transformations can be constructed.

2.4 Demystifying Contrastive SSL

The goal of self-supervised learning in Computer Vision is to learn visual representations. But what is a good visual representation? The current answer [79] seems to be: a representation that is useful for downstream tasks like object detection, image classification, etc. Therefore, self-supervised representations are evaluated by directly measuring the performance on the downstream tasks. However, this only provides a very utilitarian analysis of the the learned representations. It does not provide any feedback as to why an approach works better or insights into the generalization of the representation to other tasks. Most self-supervised learning approaches[30, 48, 91, 247] provide intuitions and conjectures for the efficacy of the learned representations. However, in order to systematically understand and improve self-supervised learning methods, a more fundamental analysis of these representations is essential.

2.4.1 Measuring Invariances

Invariance to transformations is a crucial component of representations in order to be deployable in downstream tasks. A representation function $h(x)$ defined on domain $\mathcal{X}$ is said to be invariant to a transformation $t : \mathcal{X} \to \mathcal{X}$ if $h(t(x)) = h(x)$. An important question to ask is what invariances do we need?

An ideal representation would be invariant to all the transformations that do not change the target/ground-truth label for a task. Consider a ground-truth labeling mechanism $y = \mathsf{Y}(x)$ (where $x \in \mathcal{X}, y \in \mathcal{Y}$ such that $\mathcal{Y}$ is the set of all labels). An ideal representation $h^*(x)$ would be invariant to all the transformations $t : \mathcal{X} \to \mathcal{X}$ that do not change the target i.e. if $\mathsf{Y}(t(x)) = \mathsf{Y}(x)$, then $h^*(t(x)) = h^*(x)$. In object recognition tasks, a few important transformations that do not change the target are viewpoint change, deformations, illumination change, occlusion and category instance invariance. We seek representations that do not change too much when these factors are varied for the same object.

We formulate an approach to measure task-relevant invariances in representations. We adopt the approach proposed in [76] with some modifications to incorporate dependence on the task labels. Consider a representation $h(x) \in R^n$ where each dimension is the output of a hidden unit. According to [76], the i-th hidden unit is said to fire when $s_i h_i(x) > t_i$ where the threshold t_i is chosen according to a heuristic described next and $s_i \in \{-1, 1\}$ allows a hidden unit to use either low or high activation values to fire. For each hidden unit, s_i is selected to maximize the considered invariance. Using this definition, a *firing representation* $f(x) \in R^n$ can be constructed where each dimension is the indicator of the corresponding hidden unit firing *i.e.* $f_i(x) = \mathbb{1}(s_i h_i(x) > t_i)$.

The *global firing rate* of each hidden unit is defined as $G(i) = \mathbb{E}(f_i(x))$. This is controlled by the chosen threshold t_i. In this work, we choose the thresholds such that $G(i) = 1/|\mathcal{Y}|$. Intuitively, we choose a threshold such that the number of samples the hidden unit fires on is equal to (or close to) the number of samples in each class[1].

A *local trajectory* $T(x) = \{t(x, \gamma) \mid \forall \gamma\}$ is a set a transformed versions of a reference input $x \in \mathcal{X}$ under the parametric transformation t. For example, for measuring viewpoint invariance, $T(x)$ would contain different viewpoints of x. The *local firing rate* for target y, is defined as:

$$L_y(i) = \frac{1}{|\mathcal{X}_y|} \sum_{z \in \mathcal{X}_y} \frac{1}{|T(z)|} \sum_{x \in T(z)} f_i(x) \quad \text{where} \quad \mathcal{X}_y = \{x | x \in \mathcal{X}, \mathsf{Y}(x) = y\} \tag{2.2}$$

Intuitively, $L_y(i)$ measures the fraction of transformed inputs (of target y) on which the i-th neuron fires. Normalizing the local firing rate by the global firing rate gives us the *target conditioned invariance* for the i-th hidden unit as $I_y(i) = \frac{L_y(i)}{G(i)}$.

[1]Note that this heuristic is only applicable for datasets with uniformly distributed targets and has been presented to simplify notation. See supplementary material Appendix 2.A for a more general formulation of this heuristic.

The final *Top-K Representation Invariance Score* (*RIS*) can be computed by averaging target conditioned invariance for top-K neurons (selected to maximize RIS) and computing the mean over all targets. We convert the Top-K RIS to a percentage of the maximum possible value (i.e. for all neurons $L_y(i) = 1 \ \forall y \in \mathcal{Y}$). For discussion on differences from [76], please see supplementary material Appendix 2.A.

We can now investigate the invariances encoded in the constrastive self-supervised representations and their dependence on the training data. Since we wish to study the properties relevant for object recognition tasks, we focus on invariances to viewpoint, occlusion, illumination direction, illumination color, instance and a combination of instance and viewpoint changes. We now describe the datasets used to evaluate these invariances and will publicly release the code to reproduce the invariance evaluation metrics on these datasets.

Occlusion: We use the training set of the GOT-10K tracking dataset[100] which consists of videos, every frame annotated with object bounding boxes and the amount of occlusion (0-100% occlusion discretized into 8 bins). We crop each bounding box to create a separate image. *Local trajectories* consisting of varying occlusions are constructed for each video by using one sample for each unique level of occlusion.

Viewpoint+Instance and Instance We use the PASCAL3D+ dataset[259] which consists of images depicting objects from 12 categories, annotated with bounding boxes and the viewpoint angle with respect to reference CAD models. We again crop each bounding box to create a separate image. *Local trajectories* consisting of objects from the same category, but different viewpoints are collected by ensuring that each trajectory only contains one image for each unique viewpoint. Additionally, we can construct local trajectories containing objects belonging to the same category and depicted in the same viewpoint, restricting the transformation to instance changes only.

Viewpoint, Illumination Direction and Illumination Color The ALOI dataset[66] contains images of 1000 objects taken on a turntable by varying viewpoint, illumination direction and illumination color separately. Therefore, the dataset directly provides 1000 local trajectories for each of the annotated properties.

Discussion The aggressive cropping in MOCO and PIRL creates pairs of images that depict parts of objects, thereby simulating occluded objects. Therefore, learning to match features of these pairs should induce occlusion invariance. From our results, we do observe that the self-supervised approaches MOCO and PIRL have significantly higher occlusion invariance compared to an Imagenet supervised model. PIRL has slightly better occlusion invariance compared to MOCO which be attributed to the more aggressive cropping transformation used by PIRL. However, the self-supervised approaches are inferior at capturing viewpoint invariance, and significantly inferior at instance and instance+viewpoint invariance. This can be attributed to the fact that instance discrimination explicitly forces the self-supervised models to minimize instance invariance.

Table 2.1: Invariances learned from Imagenet: We compare invariances encoded in supervised and self-supervised representations learned on the Imagenet dataset. We consider invariances that are useful for object recognition tasks. See text for details about the datasets used. We observe that compared to the supervised model, the contrastive self-supervised approaches are better only at occlusion invariance.

Dataset	Method	Occlusion		Viewpoint		Illumination Dir.		Illumination Color		Instance		Instance+Viewpoint	
		Top-10	Top-25	Top-10	Top-25	Top-10	Top-25	Top-10	Top-25	Top-10	Top-25	Top-10	Top-25
Imagenet	Sup. R50	80.89	74.21	89.54	82.62	94.63	89.08	99.88	99.38	66.11	59.44	70.17	63.47
Imagenet	MOCOv2	84.19	77.88	85.15	75.08	90.28	80.76	99.66	97.11	62.49	55.01	67.4	60.52
Imagenet	PIRL	84.46	78.38	85.8	76.08	87.7	78.45	99.68	97.19	52.97	46.79	57.01	51.03

2.4.2 Augmentation and Dataset Biases

The results above raise an interesting question: how do self-supervised approaches outperform even supervised approaches on occlusion invariances. As discussed above, the answer lies in how contrastive self-supervised learning construct positive examples. Most approaches treat random crops (from 20% to 100% of original image) of images as the positive pairs which essentially is matching features of partially visible (or occluded) images. Note that PIRL[158] follows an even more agressive strategy: dividing a random crop further into a 3x3 grid.

But this aggressive augmentation comes at a cost. Consider the example of an indoor scene shown in the Figure 2.1(left). Random cropping leads to samples like those shown in Figure 2.1(right). Contrastive learning on such positive pairs effectively forces the couch, dining table, refridgerator and the window to have similar representations. Such a representation is clearly not beneficial for object discriminating tasks. However, the learned approaches still demonstrate strong results for image classification. We hypothesize that this could be due to two reasons: (a) Bias: The pre-training datasets and downstream tasks are biased; (b) Capacity: the capacity of current representation function is low. While the objective being optimized is incorrect, current networks can only provide sub-optimal optimization which in practice is effective. In this paper, we focus on the first hypothesis.

Biases: Contrastive self-supervised approaches are most commonly trained on the ImageNet dataset. Images in this dataset have an object-centric bias: single object is depicted, generally in the center of the image. This dataset bias is highly advantageous for constrastive self-supervised learning approaches since the random crops always include a portion of an object and not include objects from other categories. While PIRL [158] has also used YFCC[224] which are less biased, the evaluation framework does not effectively evaluate the discriminative power. For example, in image classification, if test images very frequently contain both couches and television, representations that do not differentiate them can still achieve seemingly impressive performances. Furthermore, background features are generally strongly tied with the objects depicted. We believe that these biases exist in the standard classification benchmark - Pascal VOC[56].

In order to verify the hypothesis of pre-training dataset bias, we first construct a new pre-training and downstream image classification task. We pretrain self-

supervised models on the MSCOCO dataset[136] which is more scene-centric and does not suffer the object-centric bias like Imagenet. Instead of using the standard VOC classification benchmark for evaluation, we crop the annotated bounding boxes in this dataset to include only one object per image (referred to as **Pascal Cropped Boxes**). This allows us to focus on the model's discriminative power.

In this experiment, we train three MOCOv2 models: trained on 118K MSCOCO images, trained on a randomly sampled 10% subset of ImageNet (similar number of images as MSCOCO) and trained on a dataset of 118K cropped bounding boxes from the MSCOCO dataset. The results are shown in Table 2.2. We observe that MOCOv2 trained on MSCOCO outperforms the model trained on MSCOCO Boxes on the standard Pascal dataset (Column 1). This could be due to two reasons: 1) due to the co-occurrence and background biases of Pascal (discussed above) which is favorable for models trained on full MSCOCO images or 2) MSCOCO Cropped boxes represent a significantly smaller number of pixels and diversity of samples compared to the full MSCOCO. On the other hand, the trend is reversed when tested on Pascal cropped boxes (Column 2). In this setting, the MOCOv2 model trained on full COCO images cannot rely on co-occurrence statistics and background. However, The object-centric bias of the MSCOCO cropped boxes leads to higher discrimination power. A similar trend is observed in comparison to the MOCOv2 model trained on the Imagenet 10% (which also possesses a strong object-centric bias) [2]. This indicates that the aggressive cropping is harmful in object discrimination and does not lead to right representation learning objective unless trained on an object-centric dataset.

2.5 Learning from Videos

Since our analysis suggests that aggressive cropping is detrimental, we aim to explore an alternative in order to improve the visual representation learned by MOCOv2. Specifically, we would like to focus on improving invariance to viewpoint and deformation since they are not captured by the MOCOv2 augmentation strategy. One obvious source of data is videos since objects naturally undergo deformations, viewpoint changes, illumination changes and are frequently occluded. We refer to these transformations collectively as *Temporal Transformations*. Since we seek representations that are invariant to these transformations, such videos provide the ideal training data. Consider the dataset of videos $v \in \mathcal{V}$ where each video $v = (v_i)_{i=1}^{\mathsf{N}(v)}$ is a sequence of $\mathsf{N}(v)$ frames.

Baseline The naive approach for learning representations from this dataset

[2] An alternative explanation for the drop in performance could be the domain change *i.e.* full scene images are shown during training, but cropped boxes are used for testing. In order to discredit this explanation, we create a separate test-dataset consisting of the subset of Pascal VOC07 test images which depict either table or chair in the image, but not both. We observe that on the table vs chair full image classification task, the representation trained on COCO-Boxes outperforms full COCO-image pre-training. Specifically, COCO-R50 has mAP of 73.64 and COCO-Boxes has mAP of 74.92

Table 2.2: Discriminative power of representations: We compare representations trained on different datasets, in supervised and self-supervised settings, on the task of image classification. We observe that representations trained on object-centric datasets, like Imagenet and cropped boxes from MSCOCO, are better at discriminating objects. We also demonstrate that the standard classification setting of Pascal VOC is not an ideal testbed for self-supervised representations since it does not test the ability to discriminate frequently co-occurring objects.

Dataset	**Method**	**Pascal** Mean AP	**Pascal Cropped Boxes** Mean AP	**ImageNet** Top-1 Acc
ImageNet	Supervised	87.5	90.13	76.5
ImageNet	MOCOv2	83.3	90.03	67.5
ImageNet	PIRL	81.1	84.82	63.6
ImageNet 10%	MOCOv2	62.32	73.85	38.53
MSCOCO	MOCOv2	64.39	71.94	33.64
MSCOCO Boxes	MOCOv2	59.6	75.29	34.24

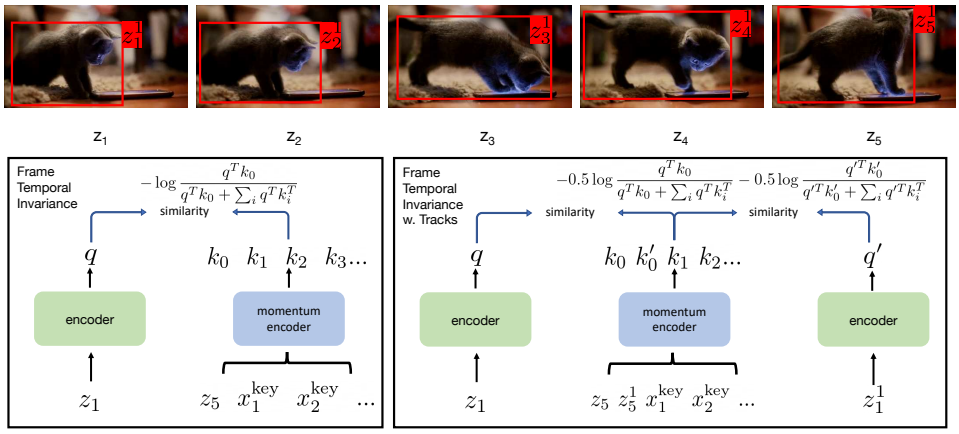


Figure 2.2: Leveraging Temporal Transformations: We propose an approach to leverage the naturally occurring transformations in videos and learn representations in the MOCOv2 framework. The Frame Temporal Invariance model uses full frames and tracked region proposals separated in time as the query and key. See supplementary material.

would be to consider the set of all frames $\{z_i | z \in \mathcal{V}, i \in \mathsf{N}(z)\}$ and apply a self-supervised contrastive learning method. We evaluate this baseline by training MOCOv2 on frames extracted from TrackingNet[165] videos. Note that in practice we extract 3 frames per video (giving 118K frames in total) uniformly spaced apart in time.

Frame Temporal Invariance The baseline approach ignores the natural transformations occurring in videos. We therefore propose an alternative approach to

leverage these temporal transformations to learn viewpoint invariant representations. We first construct a dataset of pairs of frames: $\mathcal{V}_{\text{pairs}} = \{(z_i, z_{i+k}) \mid z \in \mathcal{V}, i \in \mathsf{N}(z), i \bmod k = 0\}$. In each of these pairs, z_{i+k} captures a naturally transformed version of z_i and vice versa. For training under contrastive learning, we can create a set of 118K positive pairs by applying the standard transformations on these frames separately. Following the notation in Section 2.3, $\mathcal{D}^+ = \{(t_i(z_i), t_j(z_{i+k})) \mid t_i, t_j \in \mathrm{T}, (z_i, z_{i+\Delta}) \in \mathcal{V}_{\text{pairs}}\}$ where T is the set of transformations used in MOCO-v2.

While this captures the temporal transformations occurring in the frames, the learned features focus on scene representations i.e. the whole frame. In order to be effective for object recognition tasks, we desire representations that encode *objects* robustly. As also demonstrated in Section 2.4.2, training on images that are not object-centric decreases the robustness of the representations. Therefore, we propose an extension to the Frame Temporal Invariance model.

Region Tracker Each frame z_i is further divided into R regions $\{z_i^r\}_{r=1}^R$ using an off-the-shelf unsupervised region proposal method[231]. In order to find temporally transformed versions of each region, we track the region in time through the video. This is done by matching each region z_i^r to a region in a subsequent frame $z_{i+\Delta}^s$ by choosing the minimum distance between the region features *i.e.* $s = \arg\min_{r'} d(z_i^r, z_{i+\Delta}^{r'})$. While any unsupervised feature representation can be used for this, we use the baseline model described above and pool features at `layer3` of the ResNet using ROI-Pooling[71]. By recursively matching regions between $\{z_i^r\}_{r=1}^R, \{z_{i+\Delta}^r\}_{r=1}^R, \{z_{i+2\Delta}^r\}_{r=1}^R, ...$, we can generate tracks of the form (z_i^r, z_{i+k}^s) that lie above a certain threshold of cumulative match scores. These tracks can be used as positive pairs for contrastive learning. We employ a similar training approach as the Frame Temporal Invariance model, but with an additional contrastive loss to match positive region pairs and discriminate negative region pairs. We provide more concrete implementation details in the supplementary material.

Table 2.3: Evaluating Video representations: We evaluate our proposed approach to learn representations by leveraging *temporal transformations* in the contrastive learning framework. We observe that leveraging frame-level and region-level temporal transformations improves the discriminative power of the representations. We present results on four datasets - Pascal, Pascal Cropped Boxes, Imagenet (image classification) and ADE20K (semantic segmentation).

Dataset	**Pascal**	**Pascal Cropped Boxes**	**ImageNet**	**ADE20K**	
	Mean AP	Mean AP	Top-1	Mean IOU	Pixel Acc.
Baseline MOCOv2	61.8	70.91	30.33	14.69	61.78
Frame Temp. Invariance	63.89	72.17	29.34	14.41	61.85
Ground Truth Tracks	66.21	76.16	37.45	14.69	61.78
Region Tracker	66.47	75.86	36.51	15.28	63.29

We now evaluate the representations learned from videos using the proposed approaches. First, we perform a quantitative evaluation of the approaches on down-

Table 2.4: Invariances of Video representations: We evaluate the invariances in the representations learned by our proposed approach that leverages frame-level (row 2) and region-level (row 3, 4) temporal transformations. We observe compared to the Baseline MOCOv2 model, the models that leverage temporal transformations demonstrate higher viewpoint invariance, illumination invariance, category instance invariance and instance+viewpoint invariance.

Method	Occlusion		Viewpoint		Illumination Dir.		Illumination Col.		Instance		Instance+Viewpoint	
	Top-10	Top-25	Top-10	Top-25	Top-10	Top-25	Top-10	Top-25	Top-10	Top-25	Top-10	Top-25
Baseline MOCOv2	81.73	75.35	81.55	71.71	82.19	72.45	98.78	93.58	43.76	40.43	48.85	45.76
Frame Temp. Invariance	79.92	73.33	83.87	74.86	84.47	75.57	99.18	96.03	42.98	39.42	47.81	44.26
Ground Truth Tracks	81.52	74.6	84.82	75.3	88.28	78.51	99.92	98.31	47.51	42.93	53.47	48.63
Region Tracker	83.26	76.52	84.97	76.18	88.3	79.34	99.77	97.7	48.81	44.38	53.31	49.04
Imagenet 10% MOCOv2	84	78.26	80.42	70.42	81.9	72.27	98.29	92.71	46.23	42.65	48.54	45.46
Imagenet MOCOv2	84.19	77.88	85.15	75.08	90.28	80.76	99.66	97.11	62.49	55.01	67.4	60.52

stream tasks. We then analyze the invariances learned in this representation by following the framework established in Section 2.4.1.

2.5.1 Evaluating Temporal Invariance Models

We evaluate the learned representations for the task of image classification by training a Linear SVMs (for Pascal, Pascal Cropped boxes) and a linear softmax classifier (for Imagenet). We also evaluate on the task of semantic segmentation on ADE20K[276] by training a two-layered upsampling neural network[142]. In Table 2.3, we report the evaluation metrics to compare the three models presented in Section 2.5. The Ground Truth tracks model uses annotated tracks rather than unsupervised tracks. We observe that the Frame Temporal Invariance representation outperforms the Baseline MOCO model on the Pascal classification tasks. We additionally observe that the Region-Tracker achieves the best performance on these all tasks demonstrating stronger discriminative power.

2.5.2 Analyzing Temporal Invariance Models

The Frame Temporal Invariance and Region-Tracker representations were explicitly trained to be robust to the naturally occurring transformations in videos. Intuitively, we expect these representations to have higher viewpoint invariance compared to the Baseline MOCO. In Table 2.4, we report the Top-K RIS percentages for the three representations. Our analysis confirms that the two proposed representations indeed have significantly higher viewpoint invariance. Most importantly, we observe that the Region-Tracker model has significantly higher viewpoint and illumination dir. invarance compared to MOCOv2 trained on a 10% subset of Imagenet (same number of samples) and is comparable to the MOCOv2 model trained on full Imagenet (10x the number of samples).

2.6 Conclusion

The goal of this work is to demystify the efficacy of constrastive self-supervised representations on object recognition tasks. We present a framework to evaluate invariances in representations. Using this framework, we demonstrate that these self-supervised representations learn occlusion invariance by employing an aggressive cropping strategy which heavily relies on an object-centric dataset bias. We also demonstrate that compared to supervised models, these representations possess inferior viewpoint, illumination direction and category instance invariances. Finally, we propose an alternative strategy to improve invariances in these representations by leveraging naturally occurring temporal transformations in videos.

Appendices

Appendix 2.A Comparison of Invariance Measure to Goodfellow et. al[76]

In Section 2.4.1 of the main text, we presented an approach to measure invariances in representations. This approach was directly adopted from [76] with some minor modifications. In this section, we describe these differences and the motivation for these modifications.

In our work, we wish to measure invariances encoded in representations while accounting for the discriminative power of the representations. However, in [76], the focus is purely on measuring invariances which in many cases could assign higher scores to representations that are not discriminative. This is manifested in the following changes:

- **Chosen Thresholds** In [76], the threshold for each hidden unit is chosen to be a constant such that the global firing rate is 0.01 (i.e. the hidden unit fires on 1% of all samples). In contrast, in our work, we choose an adaptive threshold for each class in the dataset. For a specific class y, we choose the threshold such that the global firing rate is $G_y(i) = P(y)$ (i.e. the fraction of samples having label y). This allows each hidden unit the ability to fire on all samples having class y. In contrast, the threshold chosen in [76] could lead to a hidden unit firing on only a fraction of the samples of class y (if $p(y) > 0.01$). Consider a hidden unit that consistently has higher activations for samples of class y. Such a hidden unit is optimally invariant and discriminative, by could have lower invariance scores under the heuristic of [76] when a local trajectory contains a higher-scoring and a lower-scoring sample of y. Note that the heuristic presented in the main paper for simplicity of notation is only applicable for datasets with uniform distribution of labels where $G_y(i) = P(y) = 1/|\mathcal{Y}|$.
- **Local Firing Rate** Since in our work we choose thresholds that are class-dependent, we need to compute separate local firing rates considering the local trajectories for each class $L_y(i)$. This has the added benefit of assigning equal importance to samples of each class, especially in class-imbalanced datasets. This is in contrast to [76], where a single local firing rate is computed across all local trajectories of all classes (denoted by $L(i)$ in [76]). This assigns

higher weights to classes with larger number of samples, hence disregarding the discriminative power of representations.

- **Invariance Scores** Since in our work we compute class-dependent local firing rates, we first compute task-dependent invariance scores $I_y(i) = L_y(i)/G_y(i)$. The Top-K hidden units are chosen for each class separately and the mean task-dependent invariance score is computed.

$$I(f) = \frac{1}{|\mathcal{Y}|} \sum_{y \in \mathcal{Y}} \frac{1}{K} \Big[\max_{|C|=K} \sum_{\substack{i \in C \\ C \subseteq [n]}} I_y(i) \Big] \tag{2.3}$$

  In [76], the Top-K hidden units are chosen across all classes, again penalizing hidden units that are optimally discriminative and invariant for specific classes.

We believe that these modifications are essential to measure invariances in representations that are intended to be used in tasks that require discrimination of classes.

Appendix 2.B Visualizing Local Trajectories

In this section, we visualize local trajectory examples for each of the invariance results in the main text. In Figure 2.B.1, we present local trajectories created to measure occlusion, instance and viewpoint+instance invariance. For invariances measured on the ALOI[66] dataset, please visit this webpage to visualize the dataset.

Appendix 2.C Intra-Instance Invariance to Sythetic Transforms

In this section, we analyze the invariance of MOCO-v2[30] and ImageNet-supervised representations to the synthetic transforms used in self-supervised contrastive representation learning methods. We consider two representations from MOCOv2 - one taken before the final MLP and one taken after the final MLP.

In order to compute the invariance score(from Section2.4.1), we consider 10000 images randomly from ImageNet. We create the "trajectories" by generating 10 instances for each image using the transforms defined in MOCO-v2[30]. In Figure 2.C.1, we present the invariance score of the three representations while considering different fractions of the representations for the top-K invariant neurons. We observe that the MOCO-v2 representation after the MLP layers is the most invariant. Additionally, we observe that small portions (¡50%) of the MOCO-v2 representation before the MLP layers is also more invariant than the supervised representation.

As explained in Section 2.4.1, the invariance score ensures that each neuron is class-discriminative by allowing it to fire on a limited number of samples. Without accounting for class-discrimination, we can also measure the cosine similarity of all pairs of samples in the instance trajectories. This gives us a measure of invariance of

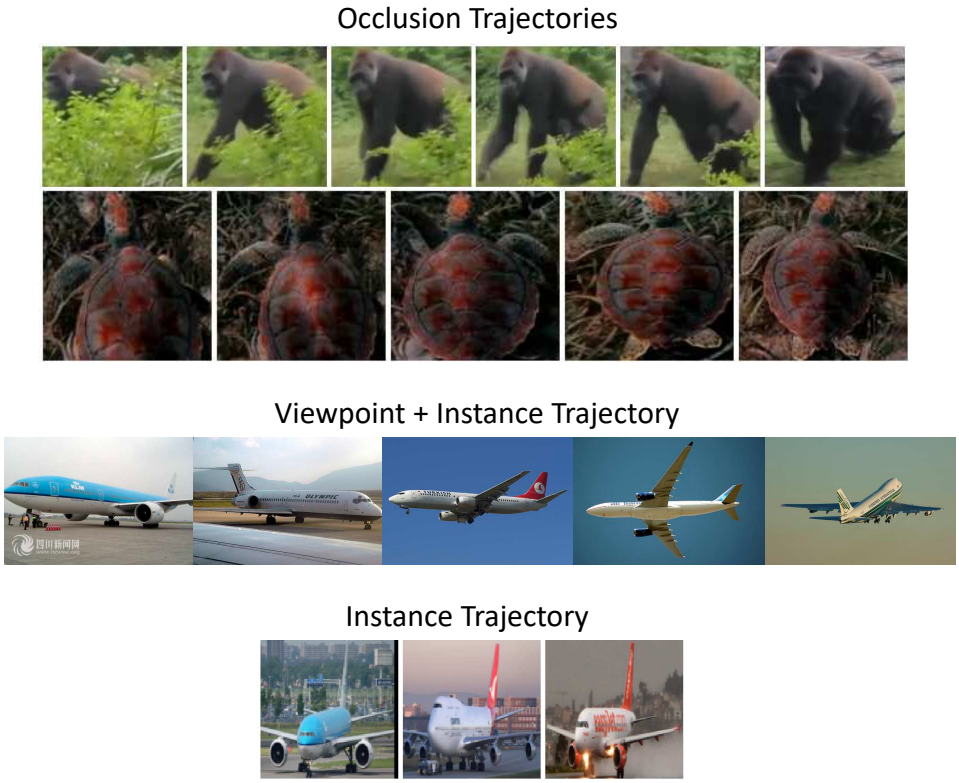


Figure 2.B.1: Local Trajectories: We present example local trajectories for measuring occlusion invariance, instance invariance and viewpoint+instance invariance.

the *full* representation by measuring average similarity between two transformed versions of an instance. In Table 2.C.1, we report these average similarity scores. We observe that both MOCO-v2 representations (before and after the MLP) demonstrate significantly higher invariance to the synthetic transforms.

Table 2.C.1: Intra-Instance Cosine Similarity: We measure the average similarity of two instances generated by synthetically transforming a single image. We compute the average similarity using 10000 images and 45 instance pairs per image.

	Supervised	MOCO-v2 (before MLP)	MOCO-v2 (after MLP)
Avg. Cosine Similarity	0.844	0.878	0.911

Appendix 2.D Implementation Details: Learning from Videos

In Section 2.5 of the main text, we present an approach to leverage naturally occurring *temporal transformations* to train models in the MOCOv2 framework[30]. In Algorithm 1, we provide pseudo-code to allow reproducibility of this method. In

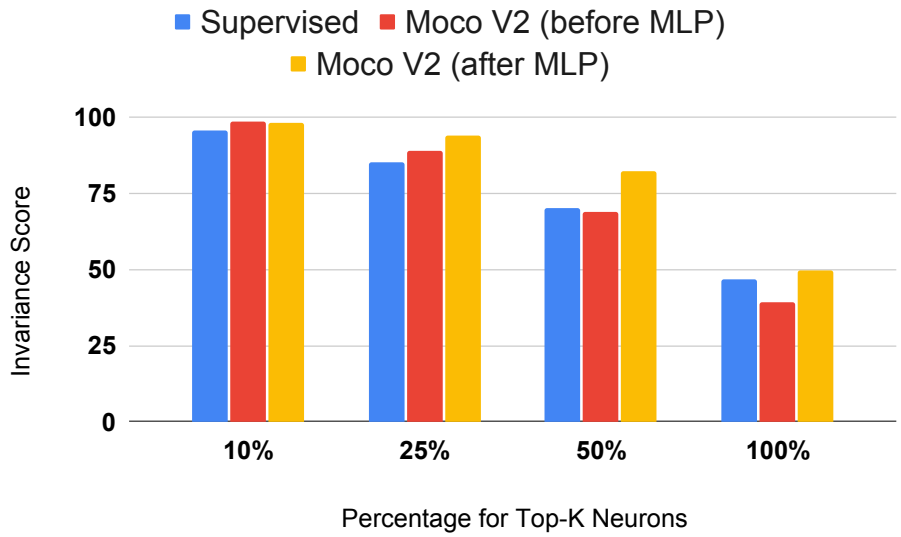


Figure 2.C.1: Intra-Instance Invariance: We measure the invariance to synthetic transforms used in self-supervised contrastive representation learning methods. We present the invariance score of different portions of supervised ImageNet-based representation and the MOCO-v2 representation taken before and after the final MLP layers.

this section, we also describe the dataset creation, unsupervised tracking method and other implementation details.

Dataset Creation For experiments in Section 5, we use the TrackingNet dataset[165] that consists of 30K video sequences. In order to increase the size of the dataset, from each video we extract 4 temporal chunks of 60 consecutive frames such that the chunks are maximally spaced apart in time. Each chunk is considered a separate video for all training purposes.

Generating Tracks For each frame, we extract region proposals using the unsupervised method - selective search[231]. We choose the top 300 region proposals for frames which produce more than 300 regions. Following the notation from the main text, each video $v = (v_i)_{i=1}^{\mathsf{N}(v)}$ is a sequence of $N(v)$ frames. Each frame consists of R regions $\{z_i^r\}_{r=1}^{R}$. The matching score between region z_i^r and a region $z_j^{r'}$ is defined as the cosine similarity between their features f i.e. $\max(0, d_{\cos}(f(z_i^r), f(z_j^{r'})))$. Here the features $f(x)$ are extracted by ROI-pooling the layer 3 of the ResNet model f. The score of a track from region z_i^r to region $z_j^{r'}$ is defined using the following recursive expression:

$$S(z_i^r, z_j^{r'}) = \sum_k \frac{j-i-1}{j-i} S(z_i^r, z_{j-1}^k) * \max(0, d_{\cos}(f(z_{j-1}^k), f(z_j^{r'}))) \tag{2.4}$$

$$S(z_t^r, z_{t+1}^k) = \max(0, d_{\cos}(f(z_t^r), f(z_{t+1}^k)) \;\; \forall r, t, k \tag{2.5}$$

For any pair of frames, we only consider tracks that have a score above a chosen

Algorithm 1: MoCo-style Pseudocode for Frame Temporal Invairance.

```
# f_q, f_k: encoder networks for query and key
# queue: dictionary as a queue of K keys (CxK)
# m: momentum
# t: temperature
# use_tracks: True for Frame Temporal Invariance with tracks

def get_loss_and_keys(x1, x2):
    x_q = aug(x1)  # a randomly augmented version
    x_k = aug(x2)  # another randomly augmented version
    q = f_q.forward(x_q)  # queries: NxC
    k = f_k.forward(x_k)  # keys: NxC
    k = k.detach()  # no gradient to keys
    # positive logits: Nx1
    l_pos = bmm(q.view(N,1,C), k.view(N,C,1))
    # negative logits: NxK
    l_neg = mm(q.view(N,C), queue.view(C,K))
    # logits: Nx(1+K)
    logits = cat([l_pos, l_neg], dim=1)
    # contrastive loss, Eqn.(1)
    labels = zeros(N)  # positives are the 0-th
    loss = CrossEntropyLoss(logits/t, labels)
    return loss, k

f_k.params = f_q.params  # initialize
for x1, x2 in loader:  # load a minibatch of frame pairs x1, x2 with N samples
    loss, k = get_loss_and_keys(x1, x2)

    if use_tracks:
        x1_patch, x2_patch = sample_track(x1, x2) # Sample a patch pair tracked from frame x1 to frame x2
        loss_patch, k_patch = get_loss_and_keys(x1_patch, x2_patch)
        loss = 0.5*loss + 0.5*loss_patch

    # SGD update: query network
    loss.backward()
    update(f_q.params)

    # momentum update: key network
    f_k.params = m*f_k.params+(1-m)*f_q.params

    # update dictionary
    enqueue(queue, k)  # enqueue the current minibatch
    dequeue(queue)  # dequeue the earliest minibatch

    if use_tracks:
        enqueue(queue, k_patch)
        dequeue(queue)
```

threshold.

Sampling Frames Training the Frame Temporal Invariance model requires sampling pairs of frames that are temporally separated $\mathcal{V}_{\text{pairs}} = \{(z_i, z_{i+k}) \,|\, z \in \mathcal{V}, i \in \mathsf{N}(z), i \bmod k = 0\}$. We sample frames that are at least $k = 40$ frames apart.

Implementation Details We use ResNet-50 as the backbone following the architecture proposed in [30] for all models. We also use the same hyperparameters as MOCOv2 [30]. In order to extract features for patches (line 10,11 of Algorithm 1 when x_q, x_k are patches) in the Frame Temporal Invariance with tracks model, we use ROI-Pooling[71] at layer3 of the ResNet model. We plan to publicly release the code upon acceptance, for reproducing all the results presented in the main text.

Chapter 3

The Challenges of Continuous Self-Supervised Learning

3.1 Introduction

We are witnessing yet another paradigm shift in the field of computer vision: from supervised to self-supervised learning (SSL). This shift promises to unleash the true potential of data, as we are no longer bound by the cost of manual labeling. Unsurprisingly, recent work has begun to scale current methods to extremely large datasets of up to 1 billion images [22, 24, 77, 78, 80] with the hope of learning better representations. In this paper, we pose the question: *Are we ready to deploy SSL in-the-wild to harness the full potential of unlimited data?*

While SSL promises to exploit the infinite stream of data generated on the internet or by a robotic agent, current practices in SSL still rely on the traditional dataset setup. Images and videos are accumulated to create a training corpus, followed by optimization on hundreds of shuffled passes through the data. The primary reason for working with datasets is the need for reproducible benchmarks, but one question remains: is this traditional static learning setup right for benchmarking self-supervised learning? Does this setup accurately reflect the challenges of a self-supervised system deployed in the wild? We believe the answer is NO. For example, consider a self-supervised system attempting to learn representations of cars over the years from the web. Current setups only evaluate static learning and do not evaluate the ability to adapt representations to new car models (and not forget old ones). Another example is to consider a deployed robotic self-supervised learning agent that actively collects frames from its video feed. This data is heavily structured and correlated due to temporal coherence. However, existing SSL benchmarks do not reflect this challenge since they rely on datasets that can be randomly sampled to produce *IID* samples.

In this paper, we move past dataset-driven SSL and investigate the efficacy of existing methods on the **Continuous Self-Supervised Learning** problem. More specif-

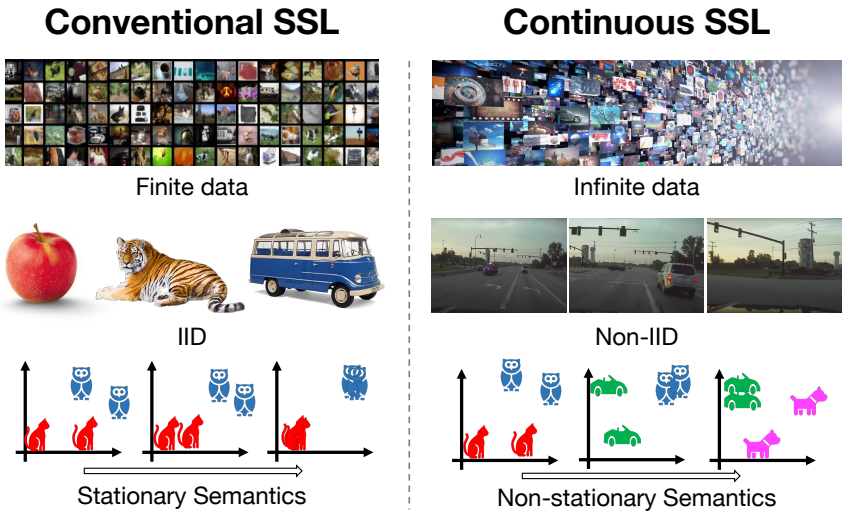


Figure 1: Conventional vs. Continuous Self-Supervised Learning. The conventional setup of fixed datasets for SSL violates key properties exhibited by data continuously gathered in-the-wild: infinite, non-IID and non-stationary semantics. Hence, for SSL methods that aim to be deployed in-the-wild, the conventional setup serves as a poor benchmark. In this work, we introduce the problem of continuous self-supervised learning to facilitate the evaluation of such methods and expose novel challenges.

ically, we explore the challenges faced in two possible methods of deployment: (a) an internet-based SSL model which relies on continuously acquired images/videos; (b) an agent-based SSL system that learns directly from an agent's sensors. Both settings rely on a streaming data source that continuously generates new data, presenting three unique challenges that should be reflected when benchmarking SSL approaches (see Figure 1).

First, storing infinite amounts of data is not feasible and obtaining data in the wild often incurs a cost of time due to bandwidth or sensor speed limitations. As a result, epoch-based training is impossible, and a naive deployment of conventional SSL approaches, using each sample only once, would lead to inefficient learners, often waiting for data to be made available, while under-utilizing the data at its disposal. One solution is to rely on replay buffers to decouple data acquisition from the training pipeline. The first question we pose is how effective replay mechanism are at allowing representations to continue to improve while data is being collected?

Second, streaming data sources cannot be "shuffled" to create mini-batches of IID samples. Instead, the ordering of samples is dictated by the source itself. We show that this creates challenges for conventional representation learning approaches, as training data is not necessarily IID. Hence, we also pose the question of how to adapt existing SSL methods to learn robust representations under various non-IID conditions?

Third, real-world data is non-stationary. For example, a higher number of football-

related images are seen during the world cup. Also, robots exploring indoor environments observe temporally clustered semantic distributions - a sequence of bedroom objects, followed by a sequence of kitchen objects, and so on. An intelligent lifelong learning system should be able to continuously learn new concepts without forgetting old ones from non-stationary data distributions. However, we show empirically that conventional contrastive learning approaches can overfit their representations to the current distribution, displaying signs of forgetting. We thus pose the question of how to design SSL methods that can learn under non-stationary conditions?

Overall, the main contributions of this work can be summarized as follows. We identify three critical challenges that arise in the continuous self-supervised learning setup, namely, training efficiency, robustness to non-IID data streams and learning under non-stationary semantic distributions. For each challenge, we construct a curated data stream that simulates this challenge and quantitatively demonstrate the shortcomings of existing SSL methods. We also propose initial solutions to these problems, with the goal of encouraging further research along these directions. We explore the idea of Buffered SSL, which involves augmenting existing approaches with a *replay buffer* to improve training efficiency. Second, we propose a novel method to handle non-IID data streams by decorrelating stored samples. Finally, we show that *decorrelated buffers* also prevent forgetting and improve continual learning under non-stationary data distributions.

3.2 Related Work

Self-supervised visual representation learning is now a mature area of research, capable of producing models that even outperform fully supervised methods when transferred to a variety of downstream tasks [24, 31, 85, 91]. Despite forgoing the use of labeled data, these methods are still trained on fixed-size curated datasets originally developed for the supervised setting. This paper explores the various challenges of deploying self-supervised learning systems truly in-the-wild.

Self-supervised learning has a long history in computer vision [21, 36, 126, 150, 162, 204] aiming to learn representations of visual data by solving tasks that can be defined without human annotations. A breadth of methodologies has been proposed from generative models such as denoising auto-encoders [236], sparse coding [131, 171, 172], inpainting [178] and colorization [42, 125, 271], to methods that learn representations predictive of spatial context [49, 69, 169], temporal context [59, 159, 177, 186, 239, 247], or concurrent modalities like audio [8, 36, 164, 175], text [41, 75, 187] or speech [153, 154].

One successful approach is to learn transformation invariant representations [29, 51, 88, 91, 158, 173, 191, 257]. Prior work has developed improved image augmentations [29, 158], backbone models [25, 77], stable (slow-moving) learning targets [24, 33, 91], and transformation invariant loss functions [25, 31, 85, 173, 268]. As a result, SSL has produced impressive models that improve state-of-the-art on a

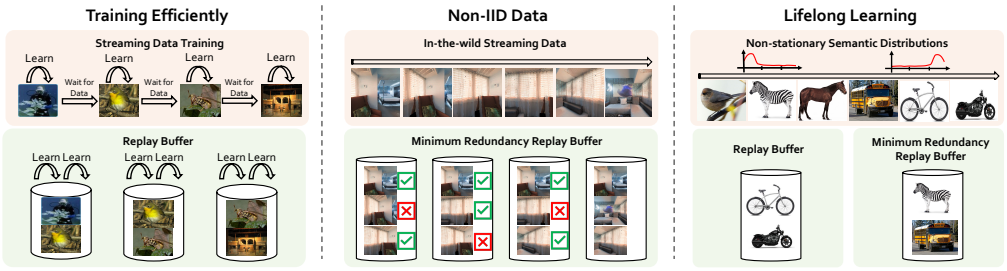


Figure 2: Overview: We investigate the problem of continuous self-supervised learning, exposing three challenges faced by SSL methods deployed in-the-wild. First, the infinite nature of data streams implies that samples cannot be repeated. We show that augmenting an existing SSL method [31] with replay buffers can significantly alleviate this issue. Second, data gathered continuously in-the-wild is often temporally correlated, violating the IID assumption of optimization algorithms. We show that enhancing replay buffers to maintain minimally redundant samples (MinRed), we can generate data that are less correlated. Finally, semantic distributions of data gathered in-the-wild are non-stationary. This poses the challenge of "forgetting" concepts seen in past distributions. We show that MinRed buffers can also alleviate the issue of "forgetting" by collecting unique samples from various semantic groups.

diverse set of downstream tasks like recognition [24, 25], detection [91] and video object segmentation [25].

Given its success, a few attempts have been made to scale SSL to large uncurated datasets, such as YFCC-100M [22, 80] and Instagram-1B [24, 77]. Goyal *et al.* [80] showed that tasks such as colorization [271], context prediction [169] and rotation [69] have diminishing returns on large datasets, due to the low complexity of the task, and argued for the development of more complex tasks. Transformation invariance objectives, coupled with heavy data augmentations, have increased the task's complexity substantially. As a result, recent attempts of scaling up augmentation invariance [24, 77, 78] have seen some performance gains. However, we argue that these methods are still not ready to be deployed truly in-the-wild. Beyond the difficulties of training on uncurated data, already studied in prior work [24, 77], training on fixed datasets ignores important challenges of streaming data, such as the non-iid nature of streaming sources, data acquisition costs, and model saturation due to its fixed capacity.

Continual and lifelong learning: The ability to continuously learn new concepts or tasks over time is often referred to as lifelong learning [225] or never-ending learning [32, 160]. Lifelong learning has traditionally been studied in supervised and reinforcement learning settings. In both cases, the model is expected to learn from a set of distinct tasks presented sequentially, without forgetting previous ones [118, 134, 190, 228, 269]. However, these works usually assume access to full supervision in the form of class labels or external rewards, not available in the streaming setup.

Techniques developed for supervised continual learning are nevertheless useful

in the Continuous SSL problem. Rehearsal techniques [5, 19, 193, 199, 213] store and replay a small set of training samples from previous tasks to avoid forgetting previously learned skills or concepts. While there is no notion of well-defined tasks in Continuous SSL, we show that replay buffers help improve training efficiency. We also propose replay buffers that minimize the redundancy of stored memories to decorrelate highly correlated streaming sources. Beyond rehearsal techniques, expandable models [203, 265] have also been used to reduce catastrophic forgetting in supervised continual learning. This is often accomplished either by progressively growing the model each time a new task is added [133, 203, 265], or maintaining a common backbone model which is adapted to each task separately using small task-specific adaptation blocks [148, 163, 193]. The lack of well-defined tasks in streaming SSL makes lifelong learning more challenging, as it needs to learn from data distributions that may shift over time.

Lifelong Generative Models: None of the existing literature has investigated how discriminative self-supervised representation learning methods perform in the full continuous learning setup (streaming, non-IID and non-stationary data). However, recent works [1, 189, 192, 263] have attempted to address a sub-problem of ours, *i.e.*, learning self-supervised representations using generative models in a continual learning setting where the domain of data exhibits significant shifts during training. These works present approaches to locate domain shifts in order to avoid the problem of catastrophic forgetting. These techniques are made possible by the fact that training data is constructed by collecting samples from images in significantly different datasets - for example, [263] uses Celeb-A[141] faces followed by 3D-Chair[9] images). In contrast, we consider a more realistic setting of ImageNet images with a smoothly changing distribution of classes. Furthermore, as highlighted above, these works do not address other critical challenges of deploying SSL in-the-wild, as they are limited to epoch-based optimization, do not consider non-curated and/or high correlated streaming sources, data efficiency, or the issue of early convergence.

3.3 Problem Setup and Challenges

The goal of this work is to investigate the efficacy of self-supervised representation learning on a continuous source of streaming data generated in the real world, which we refer to as the *continuous self-supervised learning problem*. First, we describe the distinction between conventional training and the continuous self-supervised learning setup. We then discuss the various unique challenges that appear in the continuous case.

3.3.1 Streaming vs Conventional Self-Supervised Learning

Existing self-supervised learning methods rely on fixed-size datasets. These datasets $\mathcal{D} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ are finite (*i.e.*, $N << \infty$), immutable (*i.e.*, $\mathcal{D}$ does not change) and readily available (*i.e.*, all its samples $\mathbf{x}_i$ can be easily accessed at all times). Due

to these properties, samples can be indexed, shuffled, and accessed at any point in training. Conventional SSL takes advantage of these possibilities by iterating over the datasets multiple times (epochs).

In contrast, Continuous SSL relies on a *streaming source* $\mathcal{S}$, defined as a time-series of unlabeled sensory data $\mathcal{S} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T)$, potentially of infinite length $T \to \infty$. At any given moment in time t, fetching data from a streaming source $\mathcal{S}$ yields the current sample $\mathbf{x}_t$. Future samples $\{\mathbf{x}_\tau \forall \tau > t\}$ are not accessible at time t, and past samples $\{\mathbf{x}_\tau \forall \tau < t\}$ are only accessible if stored when fetched.

In the Continuous SSL setup, one important parameter is the ratio between the data loading time t_{data} and the time taken to perform one optimization step t_{opt}. In most deployment setups $t_{\text{data}} > t_{\text{opt}}$, due to slower data transfer speed or low sensor frame rates. Therefore, even with parallelization, optimization algorithms can wait idle for $t_{\text{idle}} = t_{\text{data}} - t_{\text{opt}}$. Therefore, SSL methods developed for the continuous setup should be able to efficiently and continually build better representations, while training on samples obtained from a streaming source.

3.3.2 Why Continuous SSL? Does scaling the number of unique images help representation learning?

To understand the effect of increasing the scale of training data (potentially to infinite), we indexed all Creative Commons images uploaded to the photo-sharing website Flickr.com between 2008 and 2021. We then used this index to create datasets of varying sizes, and train visual representations through self-supervision over multiple epochs in the Conventional SSL setup.

We adopt SimSiam [31] as a prototypical example of contrastive learning methods, which have been shown to be effective for Conventional SSL. SimSiam learns representations by optimizing the augmentation invariance loss

$$\mathcal{L}(x_1, x_2) = -\texttt{sg}(\mathbf{z}_1)^T g(\mathbf{z}_2) - \texttt{sg}(\mathbf{z}_2)^T g(\mathbf{z}_1) \tag{3.1}$$

where x_1 and x_2 are two random transformations of an image x, $\mathbf{z}_i = f(x_i)$ is the model output representations, $\texttt{sg}(\cdot)$ the stop gradient and $g(\cdot)$ a prediction head. Refer to [31] for full details. Figure 3 shows the linear classification accuracy on ImageNet for models trained on different datasets as a function of the number of model updates. Unsurprisingly, training with more diverse data leads to better representations. This highlights the benefits of scaling *unique* images, which Continuous SSL will take to the extreme.

3.3.3 Challenges of Continuous SSL

Learning representations in the Continuous SSL setup poses novel challenges that Conventional SSL methods do not face.

- Epochs vs One Pass Streaming sources do not allow revisiting samples obtained in the past unless they were stored. Since storing the full stream is infeasible due

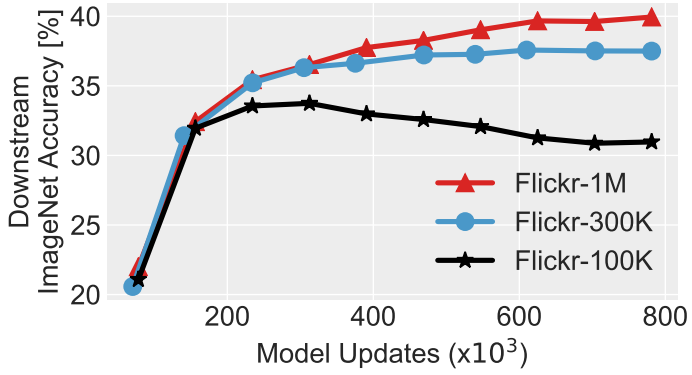


Figure 3: ImageNet downstream accuracy of a SimSiam model trained on datasets of different sizes with a ResNet-18 backbone.

to the potentially infinite length, Continuous SSL methods are required to learn representations in "one pass" over the samples.

- Sampling Efficiency Sampling data from streaming sources in the real world can be inefficient due to sensor frame rates or bandwidth limitations. This significantly increases the time taken to learn representations as optimization algorithms may have to wait idly while waiting for data.

- Correlated Samples Many streaming sources in the wild exhibit temporal coherence. For example, consecutive frames from online videos or from a robot exploring its environment display minimal changes. Such correlations break the *IID* assumption on which conventional optimization algorithms rely.

- Lifelong learning Access to infinite streams of data provides us the opportunity to continuously improve visual representations. However, the non-stationary nature of data streams in the wild cause conventional SSL methods to quickly forget features that are no longer relevant for the current distribution. This poses another challenge: as we continuously acquire new data, how can Continuous SSL methods integrate new concepts in their representations without forgetting previously learned ones?

While all these challenges co-exist in the wild, evaluating current SSL methods directly would prevent us from analyzing each one comprehensively and in isolation. Instead, we disentangled each challenge by designing a set of data streams that highlight each problem separately, and assess its effect on existing SSL methods. This helps us building a thorough characterization of each challenge and inform us on how to tackle them. We believe a disentangled analysis will help the community build intuitions about the impact of each challenge on continuous SSL as a whole. Section §3.4 introduces the challenge of one pass training and computational efficiency.

Section §3.5 introduces the non-iid data setup, and Section §3.6 analyses the lifelong learning setting.

3.4 Efficient Training

Computational and data efficiency are two challenges that currently prevent SSL from being deployed on continuous data streams in-the-wild. For most practical applications, $t_{\text{data}} : t_{\text{optim}}$ might be high, so SSL methods should use idle time to improve the models. Second, fetching new samples can still be costly. For example, exploration robots often run on batteries, and web crawlers are limited by network bandwidths. Trivially deploying current SSL methods to the streaming setup would discard each batch of data after being used once. However, current deep learning optimization practices show that iterating over the same samples over multiple epochs helps learn better representations. For example, supervised learning on ImageNet [93, 120] iterates over the dataset 100 times, and SSL approaches [29] have been shown to keep improving even after seeing each sample 800 times. Therefore, we would like to answer the question of how to improve data efficiency while still following the streaming setting.

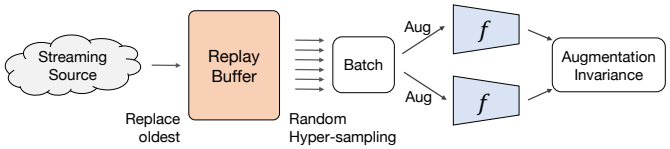
3.4.1 Buffered Self-Supervised Learning

We present a simple solution to the challenges above. The key idea is to maintain a fixed-size *replay buffer* that stores a small number of recent samples. This idea is inspired by experience replay [135] commonly used in reinforcement learning [7, 161, 206] and supervised continual learning [99, 199]. As shown in Figure 4a, the replay buffer decouples the streaming source from the training pipeline. The streaming data can be added to the replay buffer when available, replacing the oldest samples (*i.e.* first-in-first-out (FIFO) update rule). Simultaneously, mini-batches of training data can be generated at any time by randomly sampling from the buffer. As shown in Figure 4b, replay buffers allow us to continue training during the otherwise idle wait time t_{idle}. Replay buffers also allow us to reuse samples by sampling them multiple times, hence reducing the total data cost. We refer to this approach as *Buffered Self-Supervised Learning*.

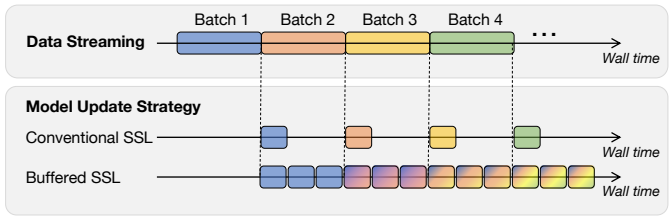
3.4.2 Single-pass training experiments

We study the effectiveness of replay buffers when training with a single pass of the data. We trained ResNet-18 SimSiam models with and without replay buffers, with various amounts of idle time $t_{\text{idle}} = t_{\text{data}} - t_{\text{optim}}$. All models were trained using the first 20 million images in our Flickr index as the streaming source.

Figure 5 shows the ImageNet linear classification performance for increasing t_{data}. By maintaining a small replay buffer (containing only the most recent 64k images),



(**a**) Overview of Buffered SSL



(**b**) Optimization under limited streaming bandwidth.

Figure 4: Buffered Self-Supervised Learning. Buffered SSL introduces a replay buffer, which allows the model to continuously train even under limited bandwidth settings.

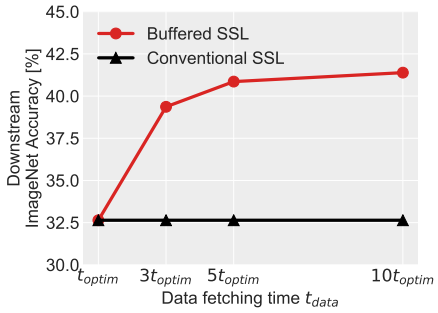


Figure 5: Streaming SSL with limited bandwidth. Comparison of buffered and non-buffered approaches for various limited bandwidth settings. $t_{\text{data}} : t_{\text{optim}}$ denotes the ratio of data acquisition time to the optimization time. Buffered SSL can take advantage of the idle time to effectively improve the learned representations instead of waiting idly for new data.

Buffered SSL was able to make good use of the idle time and improve representations significantly (41.4% accuracy on ImageNet) over the bottlenecked Conventional SSL approach (32.5% ImageNet accuracy). Replay buffers also improve data efficiency in the Continuous SSL setup, as each sample can be reused multiple times. Data usage is proportional to the hyper-sampling rate K, defined as the ratio between the number of mini-batches generated for training and acquired from the streaming source.

To understand the limits of hyper-sampling, we trained a ResNet-18 SimSiam model with a replay buffer for a fixed amount of updates (780 000 iterations). Table 1 shows a comparison of Buffered SSL at varying hyper-sampling rates K, to Conventional SSL trained on the same amount of data, and Epoch-based SSL methods trained for K epochs. Epoch-based SSL and Buffered SSL are optimized with the same number of updates, but the former violates the streaming setup. Despite being required to train on a single pass of the data, Buffered SSL with a hyper-sampling rate of $K = 10$ achieved similar performance to epoch-based training, even for buffers as small as 64K images (0.3% of the 20M unique images seen). Table 1 also shows that, as hyper-sampling rates increase, the size of the replay buffer becomes critical. For

Table 1: Data Efficiency: Augmenting SSL methods with replay buffers can improve efficiency allowing us to train on data streams with one pass. We show that Buffered SSL methods outperform the Conventional SSL methods and achieve performances close to training for multiple epochs.

	Epochs	Hyper Sampling	Memory Size	ImageNet Top1 Acc	iNaturalist Top1 Acc
Training DB: Flickr 20M					
Conventional SSL	1	-	-	32.3	9.8
Buffered SSL	1	10	16K	41.4	16.7
Buffered SSL	1	10	64K	41.8	17.3
Buffered SSL	1	10	256K	41.5	17.5
Epoch-based SSL*	10	-	-	41.9	17.5
Training DB: Flickr 5M					
Conventional SSL	1	-	-	14.5	2.8
Buffered SSL	1	40	16K	39.9	16.1
Buffered SSL	1	40	64K	41.0	17.1
Buffered SSL	1	40	256K	41.5	17.3
Epoch-based SSL*	40	-	-	41.8	17.0
Training DB: Flickr 1M					
Conventional SSL	1	-	-	8.0	1.5
Buffered SSL	1	200	16K	30.5	9.5
Buffered SSL	1	200	64K	36.4	14.3
Buffered SSL	1	200	256K	38.8	15.5
Epoch-based SSL*	200	-	-	41.7	17.3

**Epoch-based SSL violates the streaming setting (reference only).*

example, for $K = 200$, Buffered SSL still improves significantly over Conventional SSL on the same amount of data, regardless of buffer size. However, better representations are learned as the buffer size increases. Since, in high hyper-sampling regimes, the buffer is updated slowly with new images from the streaming source, increasing the buffer size prevents the model from quickly overfitting to the samples in the buffer.

3.5 Correlated Data Sources

Visual data obtained in-the-wild is often correlated and non-*IID*. For example, video feed from a self-driving car collects very similar consecutive frames. This is in stark contrast to the data used in Conventional SSL methods. For example, the ImageNet dataset allows sampling images from a collection of 1000 uniformly distributed object classes. Even methods trained on larger datasets like Instagram-1B [77, 80] are less likely to encounter heavily correlated samples in the mini-batches. However, the constant flow of data in the Continuous SSL setup generally violates these assumptions even in the static image setup (images uploaded near events are likely

Table 2: Visually Correlated SSL: Linear classification performance of buffered and unbuffered SimSiam representations trained on data sources with high temporal coherence. MinRed buffers learns better representations by decorrelating the data.

	Epochs	Hyper Sampling	Memory Size	ImageNet Top1 Acc	iNaturalist Top1 Acc
Streaming source: Kinetics ($N_{seq}=16$)					
Conventional SSL	5	-	-	17.7	3.0
Buffered SSL	1	5	64K	25.9	8.4
Buffered SSL (MinRed)	1	5	64K	26.2	7.9
Decorrelated source*	5	-	-	25.9	7.9
Streaming source: Kinetics ($N_{seq}=64$)					
Conventional SSL	5	-	-	7.6	0.8
Buffered SSL	1	5	64K	11.7	1.4
Buffered SSL (MinRed)	1	5	64K	31.2	9.9
Decorrelated source*	5	-	-	30.7	9.9
Streaming source: Krishna CAM					
Conventional SSL	5	-	-	0.4	0.03
Buffered SSL	1	5	16K	0.5	0.05
Buffered SSL (MinRed)	1	5	16K	15.2	3.43
Buffered SSL	1	5	64K	1.7	0.07
Buffered SSL (MinRed)	1	5	64K	17.9	5.91
Decorrelated source*	5	-	-	19.2	6.94

**Decorrelated sources violate the streaming setting (reference only).*

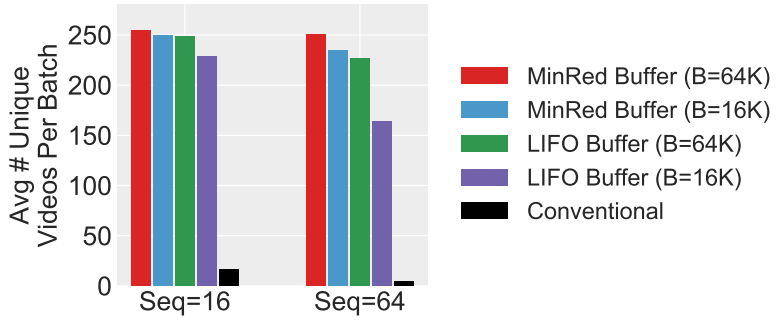
to be highly correlated).

Let $(x_i : i \in \mathcal{D})$ be a sequence of samples. When x_i is generated by randomly sampling from a large dataset, samples are close to IID. Hence, the probability p_c that two samples x_i and x_j are highly correlated is low, $p_c \approx 0$. Correlated samples may indicate images that are visually very similar or visually dissimilar but depict similar semantic content. However, in the Continuous SSL setup, the IID assumption is generally violated, leading to $p_c >> 0$. Under the assumption that consecutive samples in a continuous stream of data have the same correlation probability p_c, the likelihood of a random pair in a batch $(x_i, \ldots, x_{i+b})$ of size b being correlated (*correlation likelihood*) is large, and given by

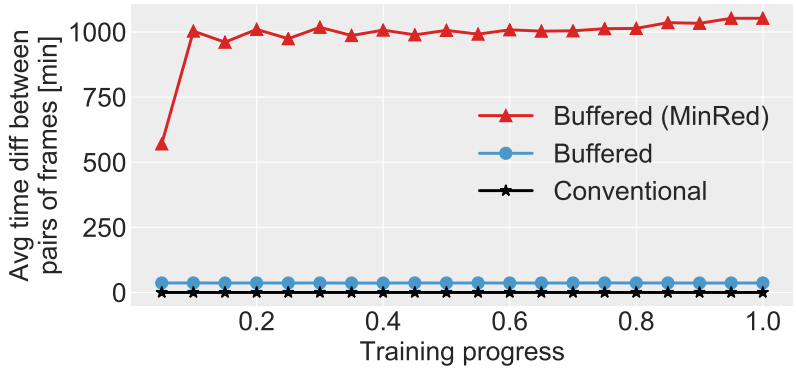
$$\mathcal{L}_{\text{Seq}} = P_c(b, p_c) = \frac{2}{b(b-1)} \sum_{i=1}^{b-1} \sum_{j=i+1}^{b} p_c^{j-1} = \frac{2p_c}{b(b-1)} \Big(\frac{p_c^b - 1}{(1-p_c)^2} + b \frac{p_c}{1-p_c} \Big). \tag{3.2}$$

Introducing a replay buffer of size $B >> b$, as proposed in §3.4.1, lowers the correlation likelihood to $\mathcal{L}_{\text{FIFO}} = P_c(B, p_c) \approx \frac{b}{B} \mathcal{L}_{\text{Seq}} < P_c(b, p_c)$[1], and enables more effective representation learning.

[1] Approximation holds for large values of B and b, and $p_c \neq 1$.



(**a**) Kinetics



(**b**) Krishna CAM

Figure 6: Estimate of within batch correlation while training w/ and w/o replay buffers.

3.5.1 Minimum Redundancy Replay Buffer

While replay buffers are able to reduce the correlation likelihood, prohibitively large replay buffers ($B >> b$) are required to significantly lower $\mathcal{L}_{\text{FIFO}}$ in heavily correlated setups ($p_c \approx 1$). In order to overcome this, we propose a modified replay buffer to only retain de-correlated samples, thereby actively reducing p_c. We call this the Minimum Redundancy Replay Buffer (MinRed).

To accomplish this, we rely on the learned embedding space to identify redundant samples. Consider a replay buffer $\mathcal{B}$ with a maximum capacity of B, already containing B samples with representation $\bar{\mathbf{z}}_i$. To add a new sample x to $\mathcal{B}$, we rely on the cosine distance between all pairs of samples to discard the most redundant:

$$\mathcal{B} \leftarrow \mathcal{B} \backslash i^* \cup \{x\} \quad \text{where} \quad i^* = \arg\min_{i \in \mathcal{B}} \min_{j \in \mathcal{B}} d_{\cos}(\bar{\mathbf{z}}_i, \bar{\mathbf{z}}_j). \tag{3.3}$$

In other words, we discard the sample with minimum distance to its nearest neighbor. To represent instances, we track the features $\bar{\mathbf{z}}_i$ of all samples in the buffer using a moving average $\bar{\mathbf{z}}_i = \alpha \bar{\mathbf{z}}_i + (1-\alpha)\mathbf{z}_i$, where $\mathbf{z}_i = f(\mathbf{x}_i)$ is the current feature of the i^{th} sample, and α the moving average coefficient. Since redundant samples are dropped

from the buffer, the probability p_c of two consecutive samples in the buffer being correlated decreases. If this probability decreases from p_c to ηp_c where $\eta << 1$, the correlation likelihood is lowered to $\mathcal{L}_{\text{MinRed}} = P_c(B, \eta p_c) < P_c(B, p_c)$, which facilitates representation learning.

3.5.2 Experiments with non-IID data streams

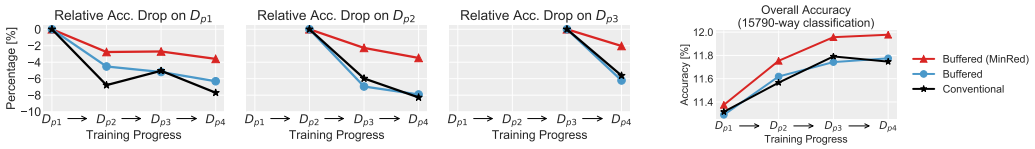
We assess the performance of SSL methods on two data streams with heavy temporal coherence. The first data stream is created by concatenating samples from videos in the Kinetics dataset [27]. From each video, we sample N_{seq} frames at random and add them sequentially to the data stream. The second training stream is taken as consecutive frames from the KrishnaCAM dataset[2] [217] which records ego-centric videos spanning nine months of the life of a computer vision graduate student. On each stream, we train the baseline SimSiam (Conventional SSL), SimSiam augmented with replay buffers (Buffered SSL) and SimSiam augmented with MinRed buffers (Buffered SSL (MinRed)). We evaluate these representations by training a linear classifier on the ImageNet [37] and iNaturalist [233] datasets. Results are shown in Table 2. We observe that the correlated nature of the data heavily disrupts training of the conventional models. While the regular replay buffers alleviate this issue to some extent, learned representations still suffer when trained on heavy correlated data streams (as in Kinetics $N_{\text{seq}} = 64$ and KrishaCAM). Finally, the proposed MinRed buffers demonstrate significant gains in these setups. Models trained with MinRed buffers are generally very close to the "oracle" setting of training from completely decorrelated streams of data (*i.e.* randomly sampling from the collection of all frames from all videos, and thus violating the streaming assumption).

Correlation of training samples: One of the benefits of Buffered SSL is the ability to generate training samples with low correlation likelihood and thus closer to *IID*. We analyzed the contents of the replay buffer over the duration of training to track the correlation likelihood (see Figure 6). We confirmed that the contents of MinRed replay buffers are significantly less correlated than FIFO buffers. In KrishnaCAM, MinRed buffers tend to maintain memories of past unique frames for longer periods of time. In Kinetics, MinRed buffers also yield training mini-batches with frames from a larger number of unique videos.

3.6 Lifelong Self-Supervised Learning

As we explore the world, we come across different distributions of object classes, some previously seen and some unseen. For example, we see furniture and appliances every day. But we also encounter novel concepts like zebras when we visit a zoo. This suggests that the distribution of semantic classes is often correlated in

[2]Concatenated videos are looped over 10 times to create a large stream.



(**a**) *Forgetting* (**b**) *Generalization*

Figure 7: Continual unsupervised representation learning on full ImageNet (14M images). The dataset is partitioned in 4 separate tasks which are seen in a sequence $D_{p1} \rightarrow D_{p2} \rightarrow D_{p3} \rightarrow D_{p4}$. Forgetting 7a is measured by computing the relative accuracy drop on each task after training on data of the task itself. Minimum redundancy buffers naturally retain instances from previous tasks, thus mitigating the catastrophic forgetting observed with conventional SSL and regular replay buffers. Generalization 7b is measured as the overall accuracy across all 15790 full ImageNet classes. By ensuring that images from past class distributions are not forgotten, minimum redundancy buffers can learn better representations overall. All results are averaged over 3 different sequences p_i.

time with occasional changes in distribution. However, Conventional SSL methods learn from a limited vocabulary of concepts that is repeatedly seen thousands of times (often uniformly). This provides a simplification of the learning setup that does not reflect the non-stationary nature of concepts in-the-wild.

3.6.1 A non-stationary data stream to benchmark SSL

To evaluate deployable SSL methods, we must use benchmarks that simulate the non-stationary semantic distributions we encounter in-the-wild. Inspired by supervised continual learning [118, 134], we introduce a setup with smooth shifting semantic distributions. Partitions will be made publically available.

First, we create four datasets $\mathcal{D}_1, \mathcal{D}_2, \mathcal{D}_3, \mathcal{D}_4$ by splitting the classes of the ImageNet-21K dataset [37]. We create the splits based on the Wordnet [156] hierarchy such that each $\mathcal{D}_i$ contains images from semantically similar classes. For each class, we hold out 25 images per class for evaluation. The training data stream is created by sampling images at random from the four datasets $\{\mathcal{D}_{p_1}, \mathcal{D}_{p_2}, \mathcal{D}_{p_3}, \mathcal{D}_{p_4}\}$ where $[p_1, p_2, p_3, p_4]$ is a permutation of the sequence $[1, 2, 3, 4]$. Images are sampled from the datasets sequentially such that images from $\mathcal{D}_{p_i}$ are seen only after most images of $\mathcal{D}_{p_{i-1}}$ are sampled (see Appendix for a detailed description of the sampling procedure), simulating a smooth change in semantic distribution. The goal is to learn a representation that can discriminate concepts from all datasets without overfitting or forgetting concepts seen earlier.

3.6.2 Experiments with non-stationary distributions

We train representations using conventional SimSiam, SimSiam with replay buffers (§3.4.1) and SimSiam with minimum redundancy buffers (§3.5.1) on a single pass of this stream of data. During evaluation, we train a linear classifier on the learned

representations to recognize all classes in the ImageNet-21k dataset, and measure the accuracy on the held-out set of each $\mathcal{D}_{p_i}$ separately. All results were averaged over 3 permutations of $p_1, \ldots, p_4$.

Figure 7a plots the drop in classification accuracy on each dataset D_{p_i} after the representation is trained on new data $D_{p_{i+1}}, D_{p_{i+2}}$, *etc.*, relative to the initial accuracy at the end of training on D_{p_i}. This serves as a measure of forgetting - a larger drop indicates that the representation is losing its ability to discriminate older classes. The results show that all methods suffer from forgetting. However, SimSiam with MinRed buffers displays less forgetting compared to conventional and buffered SimSiam. Intuitively, this can be attributed to the MinRed criteria that leads to retention of images from the older semantic distributions. Figure 7b also shows the accuracy on all classes as training progresses. We observe that SimSiam with MinRed buffers consistently yields better generalization. In supplementary material, we also evaluated the learned representations on unseen classes, by testing only on future data streams $D_{p_{i+t}}$. Since MinRed buffers maintain training buffers with wider coverage of semantics, the learned representations were also shown to be more generalizable even to unseen concepts.

3.7 Discussion and Future Work

In this work, we exposed three challenges that require investigation to build robust deployable self-supervised learners. We improve the efficiency of Continuous SSL by leveraging replay buffers to revisit old samples. In future work, developing approaches for quickly rejecting samples by preemptively evaluating their value might yield improved data efficiency. We also propose a novel minimum redundancy buffer to discard correlated samples allowing us to mimic the generation of IID training data, even in highly correlated settings. An alternative future direction could focus on learning representations that take advantage of the correlated nature of the data stream to learn from fine-grained discrepancies.

In data streams with non-stationary semantic distributions, we show that MinRed buffers alleviate the issue of catastrophic forgetting, as they are capable of maintaining unique samples from past distributions. However, we observed signs of saturating generalization as new concepts are introduced. Some possible reasons could be: 1) the cosine decay learning rate schedule and 2) the fixed capacity of our models that prohibits learning a large sequence of novel concepts. In preliminary experiments (see supplementary material), we saw that training with a constant learning rate (on 100M images from Flickr) does not lead to significant improvements in performance. We also observed that trivially expanding the architecture at regular intervals does not lead to noticeable improvements. However, we believe that further exploration in this direction is required to continually learning novel concepts in a self-supervised manner.

3.8 Conclusion

One of the grand goals of self-supervised learning is to build systems capable of continually learning from unlimited sources of unlabelled data. However, due to the need for benchmarking, existing SSL methods have primarily focused on curated datasets of limited size. Unfortunately, while the existing approaches work well in the dataset setup, we are still not close to deployable continual self-supervised methods. In this work, we advocate for a more realistic SSL setup that will facilitate deployment, while retaining the benefits of benchmarking. To this end, we identified three broad challenges of deployable SSL - training efficiency, correlated data, and lifelong learning, - and proposed potential solutions to address them. We believe however that further research is needed to develop deployable systems that deliver on the promise of self-supervised learning, and hope future efforts in SSL research focus on these challenges.

Appendices

Appendix 3.A Additional results

3.A.1 Generalization towards unseen categories

To assess the open set generalization ability of models trained with Minimum Redundancy (MinRed) buffers, we extended the continual learning experiment described in Section 6.2 and Figure 7 of the main paper, and further evaluate on future data partitions, *i.e.*, data partitions containing categories yet unseen in the training sequence. The results are shown in Fig. 3.A.1. Training models with MinRed buffers also lead to better generalizable towards unseen categories. This is likely explained by the fact that MinRed buffers maintain higher semantic diversity in the training data, which encourages the model to learn more general representations, likely to generalize better to unseen categories.

3.A.2 Buffer contents during lifelong learning

To understand why MinRed buffers allow SimSiam to learn from non-stationary distributions with less forgetting (Section 6 of the paper), we analysed the contents of the buffer used to generate training samples. Figure 3.A.2 shows the number of images in the buffer from each of the $D_{p1}, \ldots, D_{p4}$ partitions, as training progresses from D_{p1} to D_{p4}. As can be seen, only MinRed buffers are capable of retaining

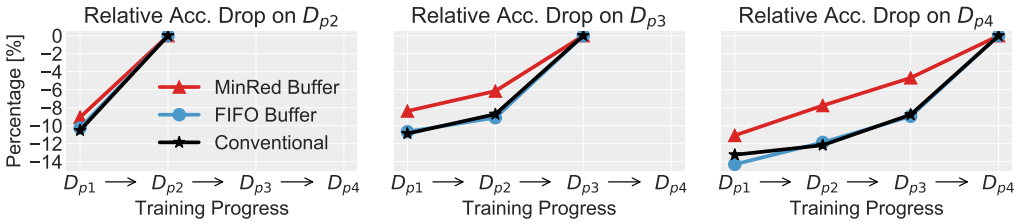


Figure 3.A.1: Open set generalization. While training on the data stream used for assessing continual learning, we also evaluated the models on future data partitions, which contain images from images from categories not yet seen during training. By training models with MinRed buffers, we can learn representations the can better generalize to unseen categories.

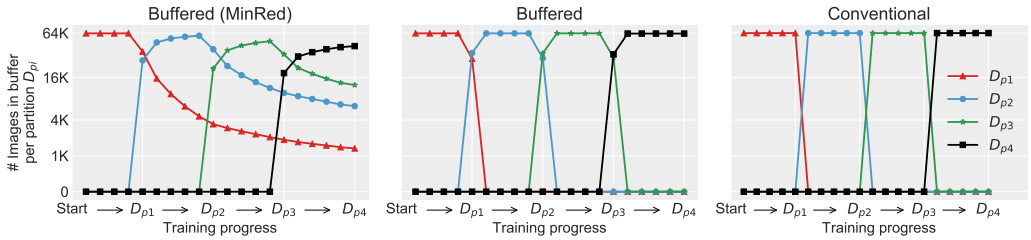


Figure 3.A.2: Contents of 64K buffers as the data distribution shifts over the course of training. For Conventional SSL (which has no buffer), we count images in a sequence of training batches totaling the same 64K images. Buffered SSL with a Minimum Redundancy (MinRed) buffer retains a significant number of images from previous data distributions. This is in contrast to Buffered SSL with LIFO buffers or Conventional SSL which have no ability to retain images for long periods of time.

images from prior distributions. Since these images can then be sampled for training, MinRed buffers enable continual training with less forgetting.

3.A.3 Learning rate schedules for continual learning

The cosine learning rate schedule is not applicable to continuous SSL, as it requires a pre-determined end. We tested several learning rate schedules. Results are shown in Figure 3.A.3. With a simple constant learning rate, models can still learn from a continuous (non-stationary) data stream, while still being able to achieve similar performances in the static case, when combined with a short learning rate decay before evaluation.

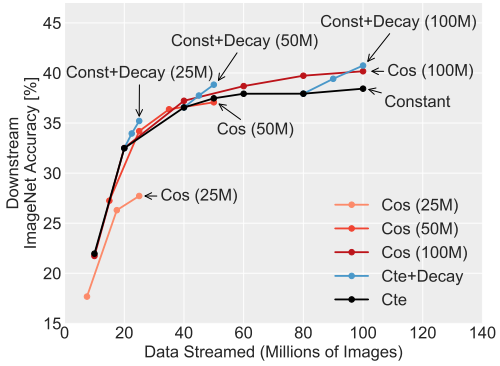


Figure 3.A.3: Downstream performance on ImageNet throughout self-supervised training with various learning rate schedules. "Cos (xM)" stands for cosine decay ending at iteration x/batch size. "Const+Decay (xM)" represents a learning rate schedule with a constant start (for about 80% of the total training time), followed by a short cosine decay for the remainder of training.

Appendix 3.B Pseudo-code for Buffered SSL with MinRed buffer

Algorithm 2:

```
def train(f, SimSiam, stream, num_updates):
  B = []  # Init empty buffer
  for ims in stream: # Load batch from stream
    Add2Buffer(B, ims)

    # Hyper-sampling: Update num_updates times
    for _ in range(num_updates):
      # Sample batch from buffer
      x = RandomSample(B)
      x1, x2 = aug(x), aug(x)
      z1, z2 = f(x1), f(x2)

      # Track features
      TrackRepresentations(B, x, (z1+z2)/2)

      # Compute loss and update models
      L = SimSiam(z1, z2)
      L.backward()        # Back-propagation
      update(f, SimSiam)  # SGD update

def Add2Buffer(B, ims):
  n_excess = len(B) + len(ims) - maxlen(B)
  if n_excess > 0:  # If full, remove n_excess.
    for _ in range(n_excess):
      # Pairwise dist
      d = pdist(B.feat, B.feat)

      # Distance to nearest neig
      d_nneig = d.min(dim=1)

      # Remove sample with smallest d_nneig
      i_redundant = d_nneig.argmin(dim=0)
      B.remove(i_redundant)

  # Add new images to buffer
  for x in ims:
    B.add(x)

def TrackRepresentations(B, x, z, alpha=0.5):
  # EMA update
  B.feat[x] = alpha*B.feat[x] + (1 - alpha)*z
```

Part II

Semantic Recognition beyond Supervision

Chapter 4

Task-Driven Modular Networks for Zero-Shot Compositional Learning

4.1 Introduction

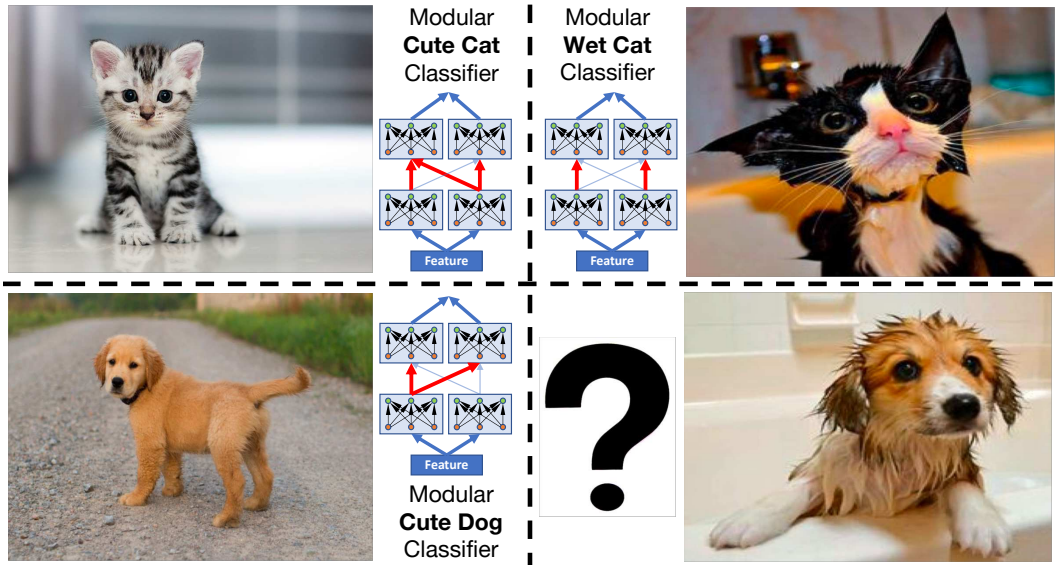


Figure 1: We investigate how to build a classifier on-the-fly, for a new concept ("wet dog") given knowledge of related concepts ("cute dog", "cute cat", and "wet cat"). Our approach consists of a modular network operating in a semantic feature space. By rewiring its primitive modules, the network can recognize new structured concepts.

How can machines reliably recognize the vast number of possible visual concepts? Even simple concepts like "envelope" could, for instance, be divided into a seemingly

infinite number of sub-categories, e.g., by size (large, small), color (white, yellow), type (plain, windowed), or condition (new, wrinkled, stamped). Moreover, it has frequently been observed that visual concepts follow a long-tailed distribution [205, 234, 253]. Hence, most classes are rare, and yet humans are able to recognize them without having observed even a single instance. Although a surprising event, most humans wouldn't have trouble to recognize a "tiny striped purple elephant sitting on a tree branch". For machines, however, this would constitute a daunting challenge. It would be impractical, if not impossible, to gather sufficient training examples for the long tail of all possible categories, even more so as current learning algorithms are data-hungry and rely on large amounts of labeled examples. How can we build algorithms to cope with this challenge?

One possibility is to exploit the *compositional* nature of the prediction task. While a machine may not have observed any images of "wrinkled envelope", it may have observed many more images of "white envelope", as well as "white paper" and "wrinkled paper". If the machine is capable of compositional reasoning, it may be able to *transfer* the concept of being "wrinkled" from "paper" to "envelope", and generalize without requiring additional examples of actual "wrinkled envelope".

One key challenge in compositional reasoning is *contextuality*. The meaning of an attribute, and even the meaning of an object, may be dependent on each other. For instance, how "wrinkled" modifies the appearance of "envelope" is very different from how it changes the appearance of "dog". In fact, contextuality goes beyond semantic categories. The way "wrinkled" modifies two images of "dog" strongly depends on the actual input dog image. In other words, the model should capture intricate interactions between *the image, the object and the attribute* in order to perform correct inference. While most recent approaches [157, 168] capture the contextual relationship between object and attribute, they still rely on the original feature space being rich enough, as inference entails matching image features to an embedding vector of an object-attribute pair.

In this paper, we focus on the task of compositional learning, where the model has to predict the object present in the input image (e.g., "envelope"), as well as its corresponding attribute (e.g., "wrinkled"). We believe there are two key ingredients required: (a) learning high-level sub-tasks which may be useful to transfer concepts, and (b) capturing rich interactions between the image, the object and the attribute. In order to capture both these properties, we propose **Task-driven Modular Networks** (TMN).

First, we tackle the problem of transfer and re-usability by employing modular networks in the high-level semantic space of CNNs [94, 127]. The intuition is that by modularizing in the concept space, modules can now represent common high-level sub-tasks over which "reasoning" can take place: in order to recognize a new object-attribute pair, the network simply re-organizes its computation on-the-fly by appropriately reweighing modules for the new task. Apart from re-usability and transfer, modularity has additional benefits: (a) sample efficiency: transfer reduces

to figuring out how to gate modules, as opposed to how to learn their parameters; (b) computational efficiency: since modules operate in smaller dimensional subspaces, predictions can be performed using less compute; and (c) interpretability: as modules specialize and similar computational paths are used for visually similar pairs, users can inspect how the network operates to understand which object-attribute pairs are deemed similar, which attributes drastically change appearance, etc. (§4.4.2).

Second, the model extracts features useful to assess the *joint-compatibility* between the input image and the object-attribute pair. While prior work [157, 168] mapped images in the embedding space of objects and attributes by extracting features only based on images, our model instead extracts features that depend on all the members of the input triplet. The input object-attribute pair is used to rewire the modular network to ultimately produce features *invariant* to the input pair. While in prior work the object and attribute can be extracted from the output features, in our model features are exclusively optimized to discriminate the validity of the input triplet.

Our experiments in §4.4.1 demonstrate that our approach outperforms all previous approaches under the "generalized" evaluation protocol on two widely used evaluation benchmarks. The use of the generalized evaluation protocol, which tests performance on both unseen *and seen* pairs, gives a more precise understanding of the generalization ability of a model [28]. In fact, we found that under this evaluation protocol baseline approaches often outperform the current state of the art. Furthermore, our qualitative analysis shows that our fully differentiable modular network learns to cluster together similar concepts and has intuitive interpretation.

4.2 Related Work

Compositional zero-shot learning (CZSL) is a special case of zero-shot learning (ZSL) [121, 124, 176, 258]. In ZSL the learner observes input images and corresponding class descriptors. Classes seen at test time never overlap with classes seen at training time, and the learner has to perform a prediction of an unseen class by leveraging its class descriptor without any training image (zero-shot). In their seminal work, Chao et al. [28] showed that ZSL's evaluation methodology is severely limited because it only accounts for performance on unseen classes, and they propose i) to test on both seen and unseen classes (so called "generaralized" setting) and ii) to calibrate models to strike the best trade-off between achieving a good performance on the seen set and on the unseen set. In our work, we adopt the same methodology and calibration technique, although alternative calibration techniques have also been explored in literature [20, 140]. The difference between our generalized CZSL setting and generalized ZSL is that we predict not only an object id, but also its attribute. The prediction of such pair makes the task compositional as given N objects and M attributes, there are potentially $N * M$ possible pairs the learner could predict.

Most prior approaches to CZSL are based on the idea of embedding the object-

attribute pair in image feature space [157, 168]. In our work instead, we propose to learn the joint compatibility [129] between the input image and the pair by learning a representation that depends on the input triplet, as opposed to just the image. This is potentially more expressive as it can capture intricate dependencies between image and object-attribute pair.

A major novelty compared to past work is also the use of modular networks. Modular networks can be interpreted as a generalization of hierarchical mixture of experts [55, 105, 109], where each module holds a distribution over all the modules at the layer below and where the gatings do not depend on the input image but on a task descriptor. These networks have been used in the past to speed up computation at test time [3] and to improve generalization for multi-task learning [152, 200], reinforcement learning [60], continual learning [232], visual question answering [6, 180], etc. but never for CZSL.

The closest approach to ours is the concurrent work by Wang et al. [251], where the authors factorize convolutional layers and perform a component-wise gating which depends on the input object-attribute pair, therefore also using a task driven architecture. This is akin to having as many modules as feature dimensions, which is a form of degenerate modularity since individual feature dimensions are unlikely to model high-level sub-tasks.

Finally, our gating network which modulates the computational blocks in the recognition network, can also be interpreted as a particular instance of meta-learning [207, 237], whereby the gating network predicts on-the-fly a subset of task-specific parameters (the gates) in the recognition network.

4.3 Approach

Consider the visual classification setting where each image $\mathcal{I}$ is associated with a visual concept c. The manifestation of the concepts c is highly structured in the visual world. In this work, we consider the setting where images are the composition of an object (e.g., "envelope") denoted by c_o, and an attribute (e.g., "wrinkled") denoted by c_a; therefore, $c = (c_o, c_a)$. In a fully-supervised setting, classifiers are trained for each concept c using a set of human-labelled images and then tested on novel images belonging to the same set of concepts. Instead, in this work we are interested in leveraging the compositional nature of the labels to extrapolate classifiers to novel concepts at test time, even without access to any training examples on these new classes (zero-shot learning).

More formally, we assume access to a training set $\mathcal{D}_{\text{train}} = \{(\mathcal{I}^{(k)}, c^{(k)}) \mid k = 1, 2, ..., N_{\text{train}}\}$ consisting of image $\mathcal{I}$ labelled with a concept $c \in \mathcal{C}_{\text{train}}$, with $\mathcal{C}_{\text{train}} \subset \mathcal{C}_o \times \mathcal{C}_a = \{(c_o, c_a) \mid c_o \in \mathcal{C}_o, c_a \in \mathcal{C}_a\}$ where $\mathcal{C}_o$ is the set of objects and $\mathcal{C}_a$ is the set of attributes.

In order to evaluate the ability of our models to perform zero-shot learning, we use a similar validation ($\mathcal{D}_{\text{val}}$) and test ($\mathcal{D}_{\text{test}}$) sets consisting of images labelled

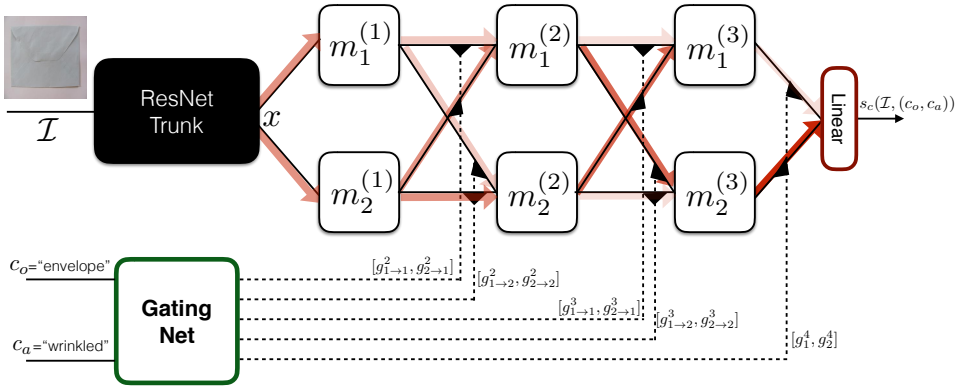


Figure 2: Toy illustration of the task-driven modular network (TMN). A pre-trained ResNet trunk extracts high level semantic representations of an input image. These features are then fed to a modular network (in this case, three layers with two modules each) whose blocks are gated (black triangle amplifiers) by a gating network. The gating network takes as input an object and an attribute id. Task driven features are then projected into a single scalar value representing the joint compatibility of the triplet (image, object and attrtibute). The overlaid red arrows show the strength of the gatings on each edge.

with concepts from $\mathcal{C}_{\text{val}}$ and $\mathcal{C}_{\text{test}}$, respectively. In contrast to a fully-supervised setting, validation and test concepts do not fully overlap with training concepts, i.e. $\mathcal{C}_{\text{val}} \setminus \mathcal{C}_{\text{train}} \neq \emptyset$, $\mathcal{C}_{\text{test}} \setminus \mathcal{C}_{\text{train}} \neq \emptyset$ and $\mathcal{C}_{\text{cal}} \cap \mathcal{C}_{\text{train}} \neq \emptyset$, $\mathcal{C}_{\text{test}} \cap \mathcal{C}_{\text{train}} \neq \emptyset$. Therefore, models trained to classify training concepts must also generalize to "unseen" concepts to successfully classify images in the validation and test sets. We call this learning setting, *Generalized* Zero-Shot Compositional learning, as both seen and unseen concepts appear in the validation and test sets. Note that this setting is unlike standard practice in prior literature where a common validation set is absent and only unseen pairs are considered in the test set [157, 168, 251].

In order to address this compositional zero-shot learning task, we propose a Task-Driven Modular Network (TMN) which we describe next.

4.3.1 Task-Driven Modular Networks (TMN)

The basis of our architecture design is a *scoring model* [129] of the joint compatibility between image, object and attribute. This is motivated by the fact that each member of the triplet exhibits intricate dependencies with the others, i.e. how an attribute modifies appearance depends on the object category as well as the specific input image. Therefore, we consider a function that takes as input the whole triplet and extracts representations of it in order to assign a compatibility score. The goal of training is to make the model assign high score to correct triplets (using the provided labeled data), and low score to incorrect triplets. The second driving principle is *modularity*. Since the task is compositional, we add a corresponding inductive bias by using a modular network. During training the network learns to decompose each recognition task into sub-tasks that can then be combined in novel ways at test time,

consequently yielding generalizeable classifiers.

The overall model is outlined in Fig. 2. It consists of two components: a gating model $\mathcal{G}$ and a feature extraction model $\mathcal{F}$. The latter $\mathcal{F}$ consists of a set of neural network modules, which are small, fully-connected layers but could be any other parametric differentiable function as well. These modules are used on top of a standard ResNet pre-trained trunk. Intuitively, the ResNet trunk is used to map the input image $\mathcal{I}$ to a semantic concept space where higher level "reasoning" can be performed. We denote the mapped $\mathcal{I}$ in such semantic space with x. The input to each module is a weighted-sum of the outputs of all the modules at the layer below, with weights determined by the gating model $\mathcal{G}$, which effectively controls how modules are composed.

Let L be the number of layers in the modular part of $\mathcal{F}$, $M^{(i)}$ be the number of modules in the i-th layer, $m_j^{(i)}$ be j-th module in layer i and $x_j^{(i)}$ be the input to each module[1], then we have:

$$x_j^{(i)} = \sum_{k=1}^{M^{(i-1)}} g_{k\to j}^{(i)} * o_k^{(i-1)}, \tag{4.1}$$

where $*$ is the scalar-vector product, the output of the k-th module in layer $(i-1)$ is $o_k^{(i-1)} = m_k^{(i-1)}\left[x_k^{(i-1)}\right]$ and the weight on the edge between $m_k^{(i-1)}$ and $m_j^{(i)}$ is denoted by $g_{k\to j}^{(i)} \in \mathbb{R}$. The set of gatings $g = \{g_{k\to j}^{(i)} \mid i \in [1, L], j \in [1, M^{(i)}], k \in [1, M^{(i-1)}]\}$ jointly represent how modules are composed for scoring a given concept.

The gating network $\mathcal{G}$ is responsible for producing the set of gatings g given a concept $c = (c_o, c_a)$ as input. c_o and c_a are represented as integer ids, and are then embedded using a learned lookup table[2]. These embeddings are then concatenated and processed by a multilayer neural network which computes the gatings as:

$$\mathcal{G}(c) = [q_{1\to 1}^{(1)}, q_{2\to 1}^{(1)}, ...q_{M^{(L-1)}\to M^{(L)}}^{(L)}], \tag{4.2}$$

$$g_{k\to j}^{(i)} = \frac{\exp[q_{k\to j}^{(i)}]}{\sum_{k'=1}^{M^{(i-1)}} \exp[q_{k'\to j}^{(i)}]}. \tag{4.3}$$

Therefore, all incoming gating values to a module are positive and sum to one.

The output of the feature extraction network $\mathcal{F}$ is a feature vector, $o_1^{(L)}$, which is linearly projected into a real value scalar to yield the final score, $s_c(\mathcal{I}, (c_o, c_a))$. This represents the compatibility of the input triplet, see Fig. 2.

[1] We set $o_1^{(0)} = x$, $M^{(0)} = 1$, and $M^{(L)} = 1$.

[2] Our framwork can be trivially extended to the case where c_o and c_a are structured, e.g., word2vec vectors [155], enabling generalization to novel objects and attributes.

4.3.2 Training & Testing

Our proposed training procedure involves jointly learning the parameters of both gating and feature extraction networks (without fine-tuning the ResNet trunk for consistency with prior work [157, 168]). Using the training set described above, for each sample image $\mathcal{I}$ we compute scores for all concepts $c = (c_o, c_a) \in \mathcal{C}_{\text{train}}$ and turn scores into normalized probabilities with a softmax: $p_c = \frac{\exp[s_c]}{\sum_{c' \in C_{\text{train}}} \exp[s_{c'}]}$. The standard (per-sample) cross-entropy loss is then used to update the parameters of both $\mathcal{F}$ and $\mathcal{G}$: $\mathcal{L}(\mathcal{I}, \hat{c}) = -\log p_{\hat{c}}$, if $\hat{c}$ is the correct concept.

In practice, computing the scores of all concepts may be computationally too expensive if $\mathcal{C}_{\text{train}}$ is large. Therefore, we approximate the probability normalization factor by sampling a random subset of negative candidates [15].

Finally, in order to encourage the model to generalize to unseen pairs, we regularize using a method we dubbed *ConceptDrop*. At each epoch, we choose a small random subset of pairs, exclude those samples and also do not consider them for negative pairs candidates. We cross-validate the size of the ConceptDrop subset for all the models.

At test time, given an image we score all pairs present in $\mathcal{C}_{\text{test}} \cup \mathcal{C}_{\text{train}}$, and select the pair yielding the largest score. However, often the model is not calibrated for unseen concepts, since the unseen concepts were not involved in the optimization of the model. Therefore, we could add a scalar bias term to the score of any unseen concept [28]. Varying the bias from very large negative values to very large positive values has the overall effect of limiting classification to only seen pairs or only unseen pairs respectively. Intermediate values strike a trade-off between the two.

4.4 Experiments

We first discuss datasets, metrics and baselines used in this paper. We then report our experiments on two widely used benchmark datasets for CZSL, and we conclude with a qualitative analysis demonstrating how TMN operates.

Datasets We considered two datasets. The **MIT-States** dataset [103] has 245 object classes, 115 attribute classes and about 53K images. On average, each object is associated with 9 attributes. There are diverse object categories, such as "highway" and "elephant", and there is also large variation in the attributes, e.g. "mossy" and "diced" (see Fig. 4 and 7 for examples). The training set has about 30K images belonging to 1262 object-attribute pairs (the *seen* set), the validation set has about 10K images from 300 seen and 300 unseen pairs, and the test set has about 13K images from 400 seen and 400 unseen pairs.

The second dataset is **UT-Zappos50k** [266, 267] which has 12 object classes and 16 attribute classes, with a total of about 33K images. This dataset consists of different types of shoes, e.g. "rubber sneaker", "leather sandal", etc. and requires fine grained

classification ability. This dataset has been split into a training set containing about 23K images from 83 pairs (the seen pairs), a validation set with about 3K images from 15 seen and 15 unseen pairs, and a test set with about 3K images from 18 seen and 18 unseen pairs.

The splits of both datasets are different from those used in prior work [157, 168], now allowing fair cross-validation of hyperparameters and evaluation in the *generalized* zero-shot learning setting. We will make the splits publicly available to facilitate easy comparison for future research.

Architecture and Training Details The common trunk of the feature extraction network is a ResNet-18 [94] pretrained on ImageNet [202] which is not finetuned, similar to prior work [157, 168]. Unless otherwise stated, our modular network has 24 modules in each layer. Each module operates in a 16 dimensional space, i.e. the dimensionality of $x_j^{(i)}$ and $o_j^{(i)}$ in eq. 4.1 is 16. Finally, the gating network is a 2 layer neural network with 64 hidden units. The input lookup table is initialized with Glove word embeddings [179] as in prior work [168]. The network is optimized by stochastic gradient descent with ADAM [115] with minibatch size equal to 256. All hyper-parameters are found by cross-validation on the validation set (see §4.4.1.1 for robustness to number of layers and number of modules).

Baselines We compare our task-driven modular network against several baseline approaches. First, we consider the *RedWine* method [157] which represents objects and attributes via SVM classifier weights in CNN feature space, and embeds these parameters in the feature space to produce a composite classifier for the (object, attribute) pair. Next, we consider *LabelEmbed+* [168] which is a common compositional learning baseline. This model involves embedding the concatenated (object, attribute) Glove word vectors and the ResNet feature of an image, into a joint feature space using two separate multilayer neural networks. Finally, we consider the recent *AttributesAsOperators* approach [168], which represents the attribute with a matrix and the object with a vector. The product of the two is then multiplied by a projection of the ResNet feature space to produce a scalar score of the input triplet. All methods use the same ResNet features as ours. Note that architectures from [157, 168] have more parameters compared to our model. Specifically, RedWine, LabelEmbed+ and AttributesAsOperators have approximately 11, 3.5 and 38 times more parameters (excluding the common ResNet trunk) than the proposed TMN. We also adapt a more recent ZSL approach [258] (referred as "*FeatureGen*") and train it for the CZSL task. This work proposes to use adversarial training to generate feature samples for the unseen classes.

Metrics We follow the same evaluation protocol introduced by Chao et al. [28] in *generalized* zero-shot learning, as all prior work on CZSL only tested performance on unseen pairs without controlling accuracy on seen pairs. Most recently, Nagarajan

Table 1: AUC (multiplied by 100) for MIT-States and UT-Zappos. Columns correspond to AUC computed using precision at k=1,2,3.

	MIT-States						UT-Zappos					
	Val AUC			Test AUC			Val AUC			Test AUC		
Model Top k $\rightarrow$	1	2	3	1	2	3	1	2	3	1	2	3
AttrAsOp [168]	2.5	6.2	10.1	1.6	4.7	7.6	21.5	44.2	61.6	25.9	51.3	67.6
RedWine [157]	2.9	7.3	11.8	2.4	5.7	9.3	30.4	52.2	63.5	27.1	54.6	68.8
LabelEmbed+ [168]	3.0	7.6	12.2	2.0	5.6	9.4	26.4	49.0	66.1	25.7	52.1	67.8
FeatureGen [258]	3.1	6.9	10.5	2.3	5.7	8.8	20.1	45.1	61.1	25.0	48.2	63.21
TMN (ours)	**3.5**	**8.1**	**12.4**	**2.9**	**7.1**	**11.5**	**36.8**	**57.1**	**69.2**	**29.3**	**55.3**	**69.8**

Table 2: Best seen and unseen accuracies, and best harmonic mean of the two. See companion Fig. 3 for the operating points used.

	MIT-States			UT-Zappos		
Model	Seen ($\bigcirc$)	Unseen ($\times$)	HM ($\blacklozenge$)	Seen	Unseen	HM
AttrAsOp	14.3	17.4	9.9	59.8	54.2	40.8
RedWine	20.7	17.9	11.6	57.3	62.3	41.0
LabelEmbed+	15.0	20.1	10.7	53.0	61.9	40.6
FeatureGen	24.8	13.4	11.2	61.9	52.8	40.0
TMN (ours)	20.2	20.1	**13.0**	58.7	60.0	**45.0**

et al. [168] introduced an "open world" setting whereby both seen and unseen pairs are considered during scoring but only unseen pairs are actually evaluated. As pointed out by Chao et al. [28], this methodology is flawed because, depending on how the system is trained, seen pairs can evaluate much better than unseen pairs (typically when training with cross-entropy loss that induces negative biases for unseen pairs) or much worse (like in [168] where unseen pairs are never used as negatives when ranking at training time, resulting in an implicit positive bias towards them). Therefore, for a given value of the calibration bias (a single scalar added to the score of all unseen pairs, see §4.3.2), we compute the accuracy on both seen and unseen pairs, (recall that our validation and test sets have equal number of both). As we vary the value of the calibration bias we draw a curve and then report its area (AUC) to describe the overall performance of the system.

For the sake of comparison to prior work, we also report the "closed-world" accuracy [157, 168], i.e. the accuracy of unseen pairs when considering only unseen pairs as candidates.

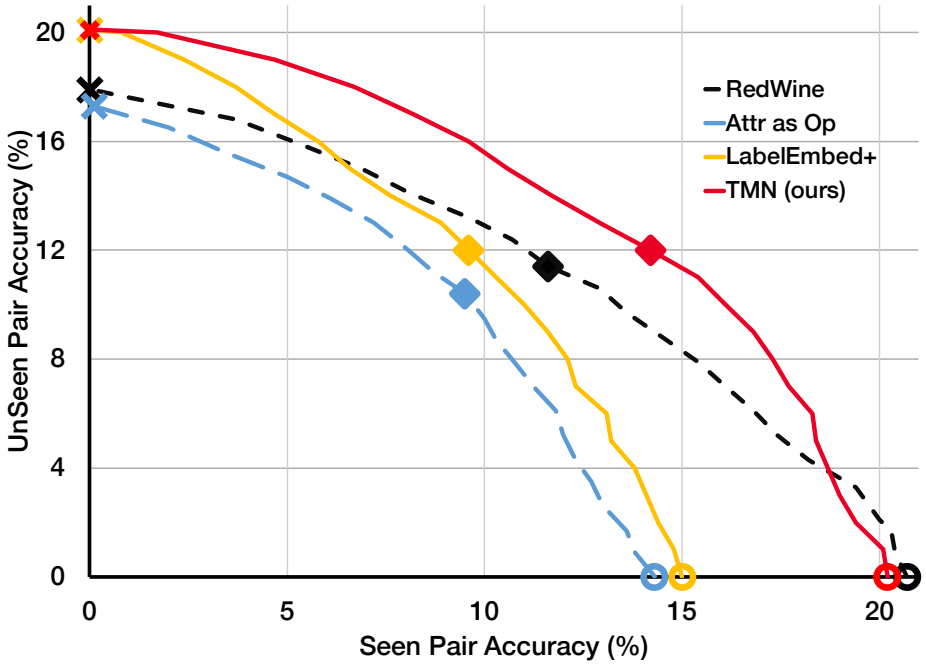


Figure 3: Unseen-Seen accuracy curves on MIT-States dataset. Prior work [168] reported unseen accuracy at different (unknown) values of seen accuracy, making comparisons inconclusive. Instead, we report AUC values [28], see Tab. 1.

4.4.1 Quantitative Analysis

The main results of our experiments are reported in Tab. 1. On both datasets we observe that TMN performs consistently better than the other tested baselines. We also observe that the overall absolute values of AUC are fairly low, particularly on the MIT-States dataset which has about 2000 attribute-object pairs and lots of potentially valid pairs for a given image due to the inherent ambiguity of the task.

The importance of using the generalized evaluation protocol becomes apparent when looking directly at the seen-unseen accuracy curve, see Fig. 3. This shows that as we increase the calibration bias we improve classification accuracy on unseen pairs but decrease the accuracy on seen pairs. Therefore, comparing methods at different operating points is inconclusive. For instance, *FeatureGen* yields the best seen pair accuracy of 24.8% when the unseen pair accuracy is 0%, compared to *TMN* which achieves 20.2%, but this is hardly a useful operating point.

For comparison, we also report the best seen accuracy, the best unseen accuracy and the best harmonic mean of the two for all these methods in Tab. 2. Although our task-driven modular network may not always yield the best seen/unseen accuracy, it significantly improves the harmonic mean, indicating an overall better trade-off between the two accuracies.

Our model not only performs better in terms of AUC but also trains efficiently.

Table 3: Ablation study: Top-1 valid. AUC; see §4.4.1.1 for details.

Model	**MIT-States**	**UT-Zappos**
TMN	3.5	36.8
a) without task driven gatings	3.2	32.7
b) like a) & no joint extraction	0.8	20.1
c) without ConceptDrop	3.3	35.7

Table 4: AUC(*100) on validaton set of MIT-States varying the number of modules per layer and the number of layers.

	Modules			
Layers	12	18	24	30
1	1.86	2.14	2.50	2.51
3	3.23	3.44	**3.51**	3.44
5	3.48	3.31	3.24	3.19

We observed that it learns from fewer updates during training. For instance, on the MIT-States datatset, our method reaches the reported AUC of 3.5 within 4 epochs. In contrast, embedding distance based approaches such as AttributesAsOperators [168] and LabelEmbed+ require between 400 to 800 epochs to achieve the best AUC values using the same minibatch size. This is partly attributed to the processing of a larger number of negatives candidate pairs in each update of TMN(see §4.3.2). The modular structure of our network also implies that for a similar number of hidden units, the modular feature extractor has substantially fewer parameters compared to a fully-connected network. A fully-connected version of each layer would have D^2 parameters, if D is the number of input and output hidden units. Instead, our modular network has M blocks, each with $(\frac{D}{M})^2$ parameters. Overall, one layer of the modular network has $D^2/(M * (\frac{D}{M})^2) = M$ times less parameters (which is also the amount of compute saved). See the next section for further analogies with fully connected layers.

4.4.1.1 Ablation Study

Our first control experiment assesses the importance of using a modular network by considering the same architecture with two modifications. First, we learn a common set of gatings for all the concepts; thereby removing the task-driven modularity. And second, we feed the modular network with the concatenation of the ResNet features and the object-attribute pair embedding; thereby retaining the joint modeling of the triplet. To better understand this choice, consider the transformation of layer i of the

Figure 4: t-SNE embedding of Attribute-Object gatings on MIT-States dataset. Colors indicate high-level WordNet categories of objects. Text boxes with white background indicate examples where changing the attribute results in similar gatings (e.g., large/small table); conversely, pairs in black background indicate examples where the change of attribute/object leads to very dissimilar gatings (e.g., molten/brushed/coil steel, rusty water/rusty wire).

modular network in Fig. 2 which can be equivalently rewritten as:

$$\begin{bmatrix} o_1^{(i)} \\ o_2^{(i)} \end{bmatrix} = \text{ReLU}\left(\begin{bmatrix} g_{1\to1}^{(i)} m_1^{(i)} & g_{2\to1}^{(i)} m_1^{(i)} \\ g_{1\to2}^{(i)} m_2^{(i)} & g_{2\to2}^{(i)} m_2^{(i)} \end{bmatrix} * \begin{bmatrix} o_1^{(i-1)} \\ o_2^{(i-1)} \end{bmatrix} \right)$$

assuming each square block $m_j^{(i)}$ is a ReLU layer. In a task driven modular network, gatings depend on the input object-attribute pair, while in this ablation study we use gatings *agnostic* to the task, as these are still learned but shared across all tasks. Each layer is a special case of a fully connected layer with a more constrained

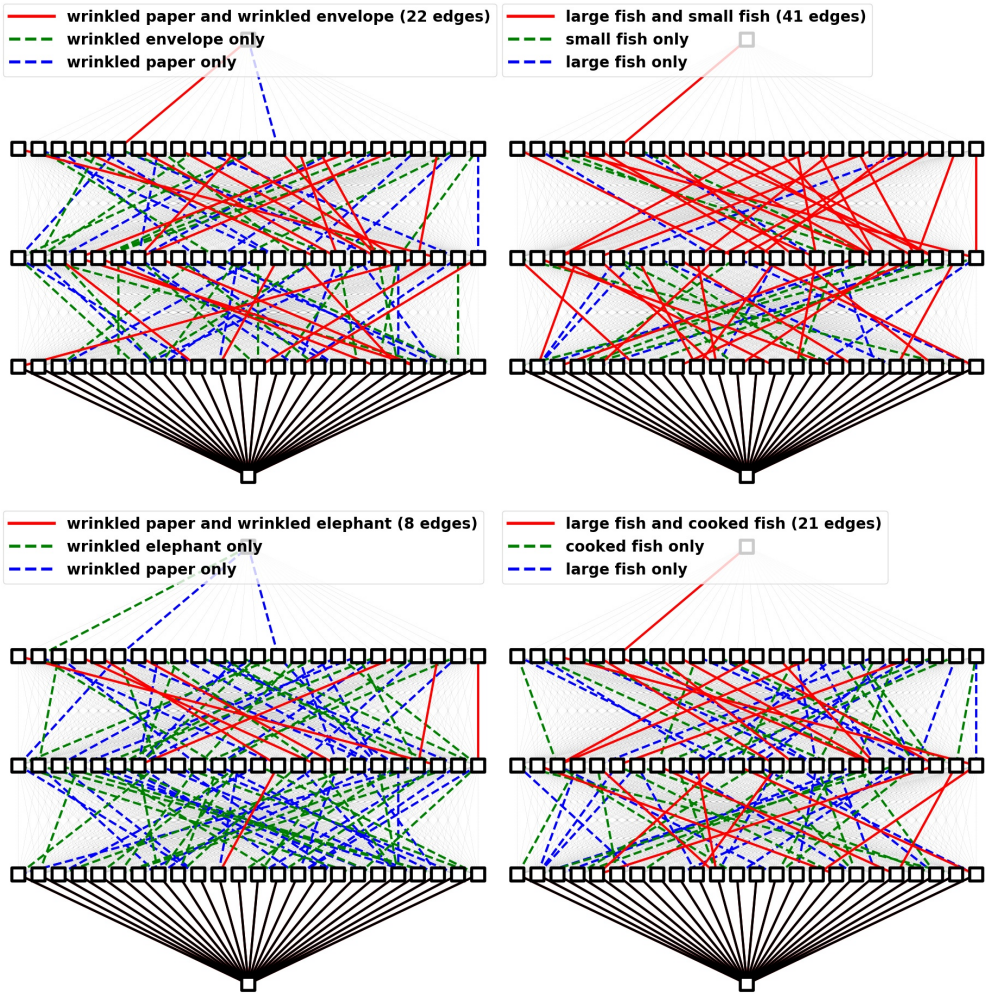


Figure 5: Examples of task driven topologies learned in TMN. Edges whose associated weight is within 3% of the highest weight for that edge are displayed. Source features x at the bottom are projected to a scalar score at the top. Each subplot compares the gatings of two object-attribute pairs. The red edges are the edges that are common between the two pairs. The green and the blue segments are edges active only in one of the two pairs. Left: two sets of pairs sharing the same attribute, "wrinkled". Right: Two sets of pairs sharing the same object, "fish". Top: examples of visually similar pairs. Bottom: example of visually dissimilar pairs (resulting in less overlapping graphs).

parameterization. This is the baseline shown in row a) of Tab. 3. On both datasets performance is deteriorated showing the importance of using task driven gates. The second baseline shown in row b) of Tab. 3, is identical to the previous one but we also make the features agnostic to the task by feeding the object-attribute embedding at the *output* (as opposed to the input) of the modular network. This is similar to LabelEmbed+ baseline of the previous section, but replacing the fully

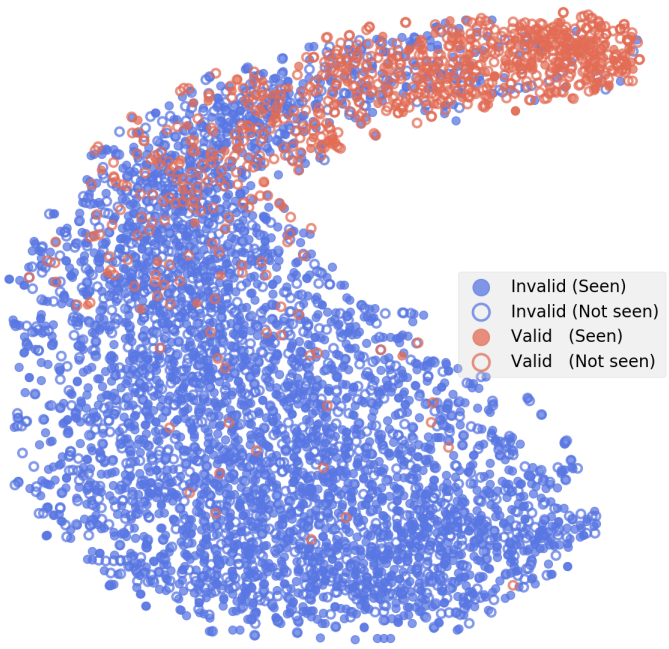


Figure 6: t-SNE embedding of the output features (penultimate layer) on MIT-States dataset. Red markers show valid (image, object, attribute) triplets (from either seen or unseen pairs), while blue markers show invalid triplets.

connected layers with the same (much more constrained) architecture we use in our TMN (without task-driven gates). In this case, we can see that performance drastically drops, suggesting the importance of extracting joint representations of input image and object-attribute pair. The last row c) assesses the contribution to the performance of the ConceptDrop regularization, see §4.3.2. Without it, AUC has a small but statistically significant drop.

Finally, we examine the robustness to the number of layers and modules per layer in Tab. 4. Except when the modular network is very shallow, AUC is fairly robust to the choice of these hyper-parameters.

Table 5: Edge analysis. Example of the top 3 object-attribute pairs (rows) from MIT-States dataset that respond most strongly on 6 edges (columns) connecting blocks in the modular network.

dry river	tiny animal	cooked pasta	unripe pear	old city
dry forest	small animal	raw pasta	unripe fig	ancient city
dry stream	small snake	steaming pasta	unripe apple	old town

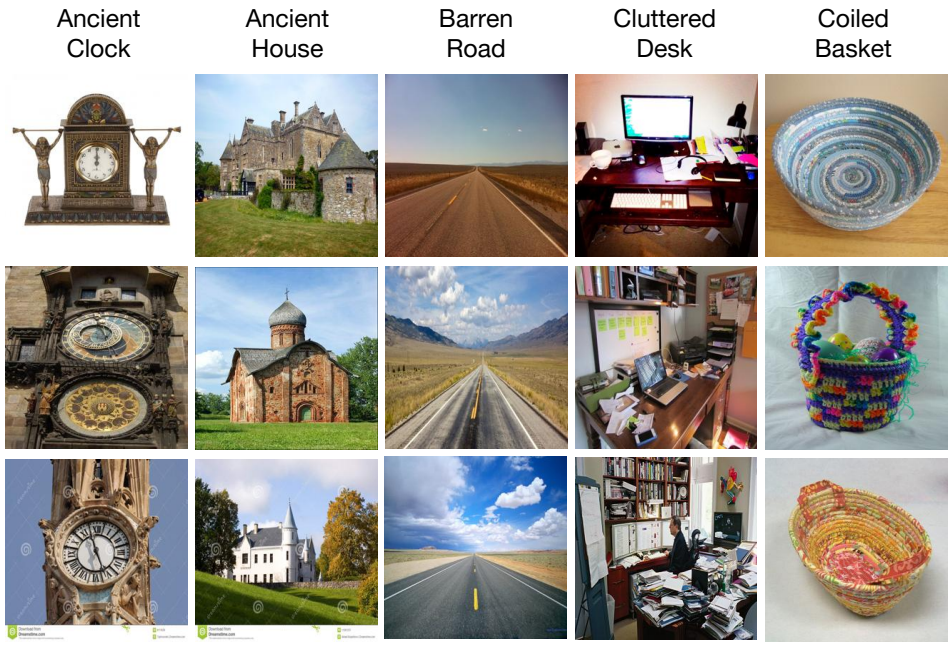


Figure 7: Example of image retrievals from the test set when querying an unseen pair (title of each column).

4.4.2 Qualitative Analysis

Task-driven modular networks offer both an increase in performance and improved interpretability. In this section, we explore simple ways to visualize them and inspect their inner workings. We start by visualizing the learned gatings in three ways. First, we look at which object-attribute pair has the largest gating value on a given edge of the modular network. Tab. 5 shows some examples indicating that visually similar pairs exhibit large gating values on the same edge of the computational graph. Similarly, we can inspect the blocks of the modular architecture. We can easily do so by associating a module to those pairs that have largest total outgoing gatings. This indicates how much a module effects the next layer for the considered pair. As shown in Tab. 6, we again find that modules take ownership for explaining specific kinds of visually similar object-attribute pairs. A t-SNE [146] embedding of the gating values associated with all the object-attribute pairs provides a more comprehensive visualization, as shown in Fig. 4. This visualization shows that the gatings are mainly

Table 6: Module analysis. Example of the top 3 object-attribute pairs (rows) for 6 randomly chosen modules (columns) according to the sum of outgoing edge weights in each pair's gating.

dark fire	large tree	wrinkled dress	small elephant	pureed soup
dark ocean	small tree	ruffled dress	young elephant	large pot
dark cloud	mossy tree	ruffled silk	tiny elephant	thick soup

organized by visual similarity. Within this map, there are clusters that correspond to the same object with various attributes. Instances where the attribute greatly changes the visual appearance of the object are interesting exceptions ("coiled steel" VS "molten steel", see other examples highlighted with dark tags). Likewise, pairs sharing the same attribute may be located in distant places if the object is visually dissimilar ("rusty water" VS "rusty wire"). The last gating visualization is through the topologies induced by the gatings, as shown in Fig. 4.B.1, where only the edges with sufficiently large gating values are shown. Overall, the degree of edge overlap between object-attribute pairs strongly depends on their visual similarity.

Besides gatings and modules, we also visualized the task-driven visual features $o_1^{(L)}$, just before the last linear projection layer, see Fig. 2. The map in Fig. 6 shows that valid (image, object, attribute) triplets are well clustered together, while invalid triplets are nicely spread on one side of the plane. This is quite different than the feature organization found by methods that match concept embeddings in the image feature space [157, 168], which tend to be organized by concept. While TMN extracts largely *task-invariant* representations using a *task-driven* architecture, they produce representations that contain information about the task using a *task-agnostic* architecture[3]. TMN places all valid triplets on a tight cluster because the shared top linear projection layer is trained to discriminate between valid and invalid triplets (as opposed to different types of concepts).

Finally, Fig. 7 present image retrieval results. Given a query of an unseen object-attribute pair, the highest scoring images in the test set are returned. The model is able to retrieve relevant images despite not having been exposed to these concepts during training.

4.5 Conclusion

The distribution of highly structured visual concepts is very heavy tailed in nature. Improvement in sample efficiency of our current models is crucial, since labeled data will never be sufficient for concepts in the tail of the distribution. A promising approach is to leverage the intrinsic compositionality of the label space. In this work, we investigate this avenue of research using the Zero-Shot Compositional Learning task as a use case. Our first contribution is a novel architecture: TMN, which outperforms all the baseline approaches we considered. There are two important ideas behind its design. First, the joint processing of input image, object and attribute to account for contextuality. And second, the use of a modular network with gatings dependent on the input object-attribute pair. Our second contribution is to advocate for the use of the generalized evaluation protocol which not only tests accuracy on unseen concepts but also seen concepts. Our experiments show that TMN provides

[3] A linear classifier trained to predict the input object-attribute pair achieves only 5% accuracy on TMN's features, 40% on LabelEmbed+ features and 41% on ResNet features.

better performance, while being efficient and interpretable. In future work, we will explore other gating mechanisms and applications in other domains.

Acknowledgements This work was partly supported by ONR MURI N000141612007 and Young Investigator Award. We would like to thank Ishan Misra, Ramakrishna Vedantam and Xiaolong Wang for the helpful discussions.

Appendices

Appendix 4.A Hyperparameter tuning

The results we reported in the main paper were obtaining using the best hyper-parameters found on the validation set. We used the same cross-validation procedure for all methods, including ours. Here, we present the ranges of hyper-parameters used in the grid-search and the selected values.

4.A.1 Task Driven Modular Networks

Hyper-parameter values:

- Feature extractor learning rates: 0.1, 0.01, 0.001, 0.0001 (chosen: 0.001)
- Gating network learning rates: 0.1, 0.01, 0.001, 0.0001 (chosen: 0.01)
- Number of sampled sampled negatives for Eq 3: for MIT States 200, 400, 600 (chosen: 600), for UT-Zappos we choose all negatives
- Batch size: 64, 128, 256, 512 (chosen: 256)
- Fraction of train concepts dropped in ConceptDrop: 0%, 5%, 10%, 20% (chosen: 5%)
- Number of modules per layer: 12, 18, 24, 30 (chosen: 24)
- Output dimensions of each module: 8, 16 (chosen: 16)
- Number of layers: 1, 2, 3, 5 (chosen: 3 for MIT States, 2 for UT-Zappos)

4.A.2 LabelEmbed+

Hyper-parameter values:

- Learning rates: 0.1, 0.01, 0.001, 0.0001 (chosen: 0.0001 for MIT States, 0.001 for UT-Zappos)
- Batch size: 64, 128, 256, 512 (chosen: 512)
- Fraction of train concepts dropped in ConceptDrop: 0%, 5%, 10%, 20% (chosen: 5%)

4.A.3 RedWine

Hyper-parameter values:

- Learning rates: 0.1, 0.01, 0.001, 0.0001 (chosen: 0.01)
- Batch size: 64, 128, 256, 512 (chosen: 256 for MIT States, 512 for UT-Zappos)
- Fraction of train concepts dropped in ConceptDrop: 0%, 5%, 10%, 20% (chosen: 0%)

4.A.4 Attributes as Operators

Hyper-parameter values:

- Fraction of train concepts dropped in ConceptDrop: 0%, 5%, 10%, 20% (chosen: 5%)

Learning rate, batch size, regularization weights chosen from the original paper and executed using the implementation at: `https://github.com/Tushar-N/attributes-as-operators`.

Appendix 4.B Additional Topology Visualizations

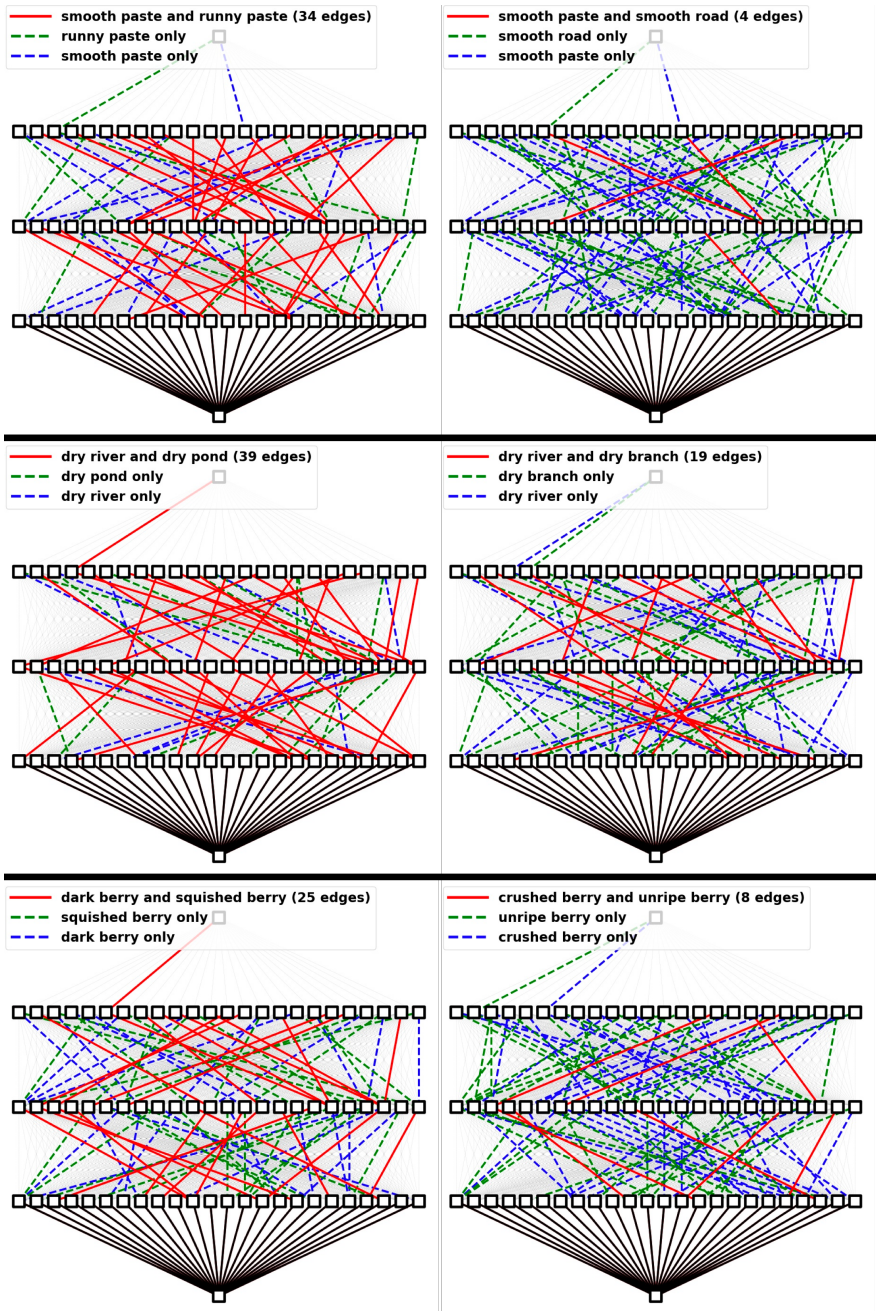


Figure 4.B.1: Additional Examples of task driven topologies learned in TMN (similar to Figure 5 of the main text).

Chapter 5

Aligning Videos in Space and Time

5.1 Introduction

> Ask not "what is this?", ask "what is this like".
>
> *Moshe Bar*

What does it mean to understand a video? The most popular answer right now is labeling videos with categories such as "opening bottle". However, action categories hardly tell us anything about the process – it doesn't tell us where is the bottle or when it was opened, let alone the different other states it can exist in, and what parts are involved in what transitions. Dense semantic labeling is a non-starter because exhaustive and accurate labels for objects, their states and actions are not easy to gather.

In this paper, we investigate the alternative of *understanding via association*, *i.e*. video understanding by extracting visual correspondences between training and test videos. Focusing on 'what is a given video like', rather than 'what class it belongs to', side-steps the problem of hand-defining a huge taxonomy and dense labeling. Inspired by this, in this paper, we focus on the task of creating associations or visual correspondences across training and test videos. More specifically, we try to align videos in both space and time. This poses two core and inter-related questions: (a) what is the granularity of visual correspondence? (b) what is the right distance metric or features to extract this correspondence?

Let us focus on the first issue: the granularity, *i.e*. the level at which we should establish correspondence: pixel-level, patch-level or frame-level. The trade-off here is between discriminability and the amount of data required for good correspondences. While full frames are more discriminative (and easy to match), they are also quite specific. For example, finding a frame that depicts the same relation between the

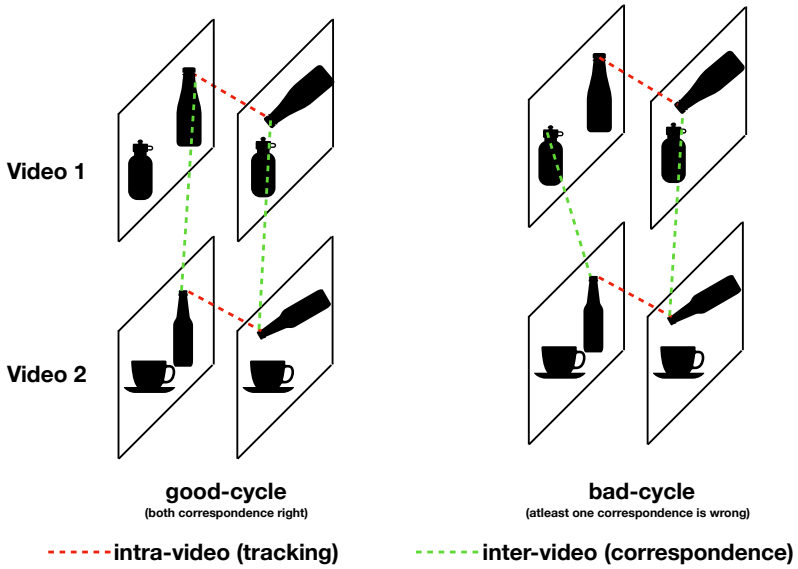


Figure 1: Learning Correspondence via Cycle Supervision. Features that allow sequences of matches (cycles) that begin and end at the same patch are desired.

bottle and the cup as shown in Figure 1 would require large amounts of training data before a good full-frame correspondence can be found. Consequently, past work with hand-crafted descriptors focused on establishing visual correspondence by matching interest points [144, 238] and image patches [218]. However, given lack of dense supervision, recent work that tries to revisit these ideas through learning [54] seeks to correspond whole frames, through temporal consistency of frames. While this works well for full frame correspondence, it doesn't produce patch-level correspondences which is both richer, and more widely applicable. This motivates our pursuit for a method to obtain dense patch-level correspondences across videos.

The second issue at hand is of how to learn a distance metric (or equivalently an appropriate feature space) for extracting visual correspondences. Classical work focused on using manually-defined features [144, 238] with a variety of distance metrics. However, given the widespread effectiveness of supervised end-to-end learning for computer vision tasks [119] (including visual correspondence [198]), it is natural to ask how to leverage learning for this task, *i.e.* what is the right objective function and supervision for learning features for obtaining correspondences? The conventional approach would be to reuse generic features from a standard task such as image classification or action recognition. As our experiments will demonstrate, neither features learned for ImageNet classification, nor ones trained for action recognition generate good correspondences due to their inability to encode object states. At the same time, direct manual annotation for visual correspondence across videos is challenging and infeasible to scale. This necessitates design of a self-supervised approach.

Interestingly, some recent efforts pursue this direction, and exploit consistency

in correspondences as supervision to learn frame-level correspondence [54], or intra-video correspondence (tracking) [250]. Our proposed method extends these methods to learn patch-level correspondences across videos via *cross video cycle-consistency*. During training, given a pair of videos, we compute matches for a patch forward in time in the first video, then match to a patch in the second video, match this patch backward in time in the second video and finally match back to a patch in the first video. This sequence of patches is referred to as a 'cycle'. Cycles that start and end at overlapping patches are encouraged to score higher than cycles that connect non-overlapping patches (see Figure 1). This allows our approach to generate finer level correspondence across videos (as SIFT Flow[138] does for images), while also harnessing the capabilities of the modern end-to-end learning approaches. Our experiments show that features learned using our approach are more effective at corresponding objects in the same state across videos, than features trained for ImageNet classification, or for action classification.

5.2 Related Work

Our work learns space-time visual correspondence by use of cycle consistency. In this section, we present a survey of related literature on video understanding (datasets, tasks and techniques), correspondence techniques in videos, and use of self-supervision and cycle consistency for learning features and correspondences.

Video Datasets and Tasks. A number of past efforts have been devoted to collecting new video understanding datasets, and extending static image tasks to videos. Leading efforts in recent times include datasets like Kinetics [111], AvA [86], Charades [214], EPIC Kitchen [35], VLOG [64], MultiTHUMOS [264]. While some of these datasets focus on action classification, a number of them investigate new tasks, such as temporal action localization [264], detection of subjects, verbs and objects [86], classification in first-person videos [35], and analysis of crowd-sourced videos [81, 214]. These works extend video understanding by scaling it up.

Architectures for Action Classification. Researchers have also pursued design of expressive neural network architectures for the task of action classification [27, 215, 229, 235, 241, 260]. Some works investigate architectures to encourage the modelling of time flow [159, 210], or long-range temporal dependencies [58, 246, 255], or object tracking [70]. While these models often capture useful intuitions, their focus is still on optimizing models for the task of action classification. Hence, even though the model has the right inductive biases, learning is bottle-necked by the low-entropy output space that of action class labels.

Beyond Action Recognition. Many efforts have also pursued the task of detailed video understanding in recent times. For example, video prediction tasks [40, 130] have the promise to go beyond action classification, as they force the model to predict much more than what can be effectively annotated. Wang *et al*. [244] model actions as operators that transform states of objects, and Nagarajan *et al*. [167] learn about

how humans interact with different objects. In contrast, we take a non-parametric approach, and understand videos by understanding what they are like, and corresponding them with other videos in space and time.

Cycle Consistency and Correspondence. Forward-backward consistency and cycle consistency have been used in computer vision for establishing correspondence in an unsupervised manner [110, 211]. Zhou *et al*. [278] use cycle-consistency to establish dense correspondence between 3D shapes, Godard *et al*. [73], use cycle consistency for learning to predict depth, Zhu *et al*. [279] use cycle consistency to learn how to generate images, and Wang *et al*. [250] use cycle consistency to learn features for correspondence over time in videos. Work from Wang *et al*. [250] is a primary motivation for our work, and we investigate use of cycle consistency to learn *cross-video* correspondences. To our knowledge, ours is the first work to investigate spatio-temporal alignment across videos with cycle consistency.

Spatial Correspondence. Finding correspondences across video frames is a fundamental problem and has been actively studied for decades. Optical flow [17] seeks to establish correspondences at the pixel-level. While numerous effective approaches have been proposed [145, 151, 221, 222], optical flow estimation is still challenging over long time periods, and fails across videos. This issue is partially alleviated by performing correspondence at a patch level. SIFT Flow[138], a seminal work in this domain, uses SIFT descriptors [144] to match patches across scene. SIFT Flow can be used to transfer labels from training data to test samples in many applications [65, 137, 201, 270]. However, patch correspondence approaches [89, 113, 277], rely on the local appearance of the patches for matching. We use a similar method to obtain spatio-temporal correspondences across videos, but account for the object states and not just the local appearance.

Cross-video Spatio-Temporal Alignment. Past works have studied spatio-temporal alignment in videos. Sermanet *et al*. [210] learn time sensitive features in a supervised manner by collecting time aligned data for an action. Alayrac *et al*. [4] learn features sensitive to object states by classifying object bounding box into before or after action. Dwibedi *et al*. [54] focus on learning temporal correspondence by enforcing consistency in nearest neighbors at frame-level. This focus on frame-level modeling ignores spatial alignment. In contrast, we focus on corresponding image patches across videos in time and *space*. This leads to learning of state-sensitive *object* representations (as opposed to scene representations). We are not aware of any past work that tackles the problem of establishing spatio-temporal correspondences across videos.

Self-supervision. A number of past works employ self-supervised learning to alleviate the need for semantic supervision from humans to acquire generic image representations. Past works have employed images [47, 272], videos [159, 177, 210, 248, 250], and also motor actions [2, 106]. Our alignment of videos in space and time, can also be seen as a way to learn representations in a self-supervised manner. However, we learn features that are sensitive to object state, as opposed to generic

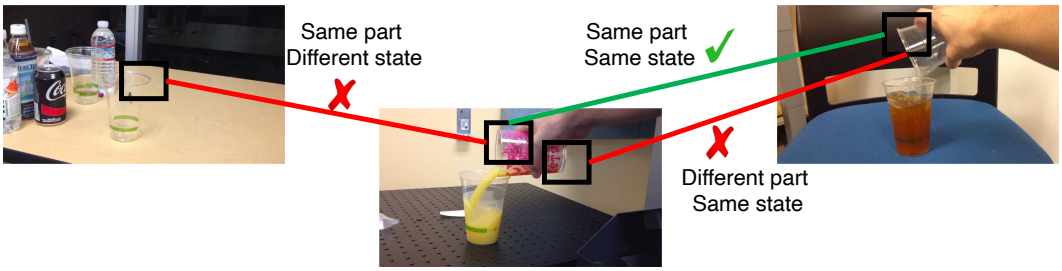


Figure 2: What is a good correspondence? A good correspondence is a match where patches correspond to the same semantic part, and are in the same state with respect to the depicted action.

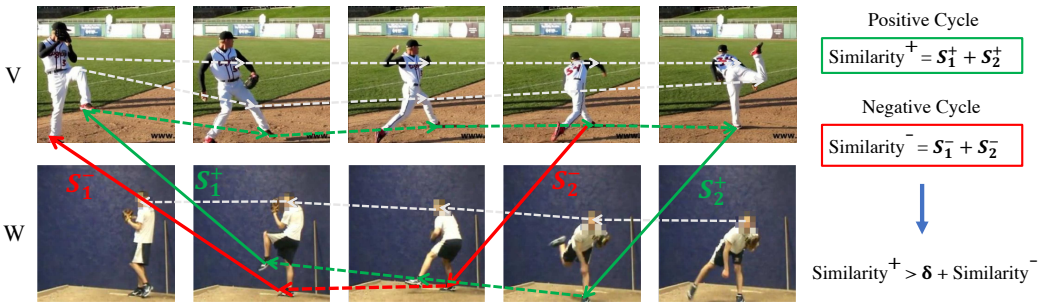


Figure 3: Overview: Given tracks in two video of the same class (shown by white dotted lines), we learn an embedding to correspond patches across videos. This is done by computing cycles (pair of cross-video edges) that correctly track a patch back to itself. We compute the best cycle that corresponds a patch to itself (shown in green) and encourage it to have a higher similarity than the best cycle that corresponds a patch to a different patch (shown in red) via a margin loss.

image features learned by these past methods.

5.3 Alignment via Cross-Video Cycle Consistency

Our goal is to learn how to spatio-temporally align two videos. We tackle this problem by extracting patch level visual correspondence across two videos. But what defines a good correspondence? A good spatio-temporal correspondence is one where two patches from different videos are linked when they depict the same objects (or their parts) and are in similar states. For example, two patches depicting rim of the cups are in correspondence as shown in Figure 2 because the patches correspond to same part and the cups are in same state (tilted for pouring). On the other hand, the other two correspondences are bad because either the patches correspond to different object parts or the states of object do not match.

While it is easy to learn features that can correspond the same objects in various states over time by learning to track [248, 250], it is far more challenging to learn features that correspond different objects in the same state. We specifically tackle

this problem in our proposed approach. One of the biggest challenge here is the supervision. It is difficult to obtain supervision for such a dense correspondence task, thus we pursue a weakly-supervised approach. Our central idea is to employ *cross-video cycle-consistency*. Specifically, we create cycles in videos of the same action class, that track patches within a video, match it to a patch in another video, track this patch back in time, and then match back to the original video. Figure 3 illustrates the idea. Cycles that can track back to the same patch are encouraged (green cycle), while cycles that get back to a different patch in the first video are discouraged (red cycles). Enforcing this objective on a large collection of foreground patches would lead to choosing semantically aligned tracks. However, note that this could lead to some trivial cycles involving very short (or single frame) tracks in the second video. It is important to disregard such solutions in order to focus on cycles where object states vary (we disregard cycles that involve tracks of length 3 or less). We now formally describe the training objective.

5.3.1 Formulation

Let's assume we have a tracker $\mathcal{T}$, that given a video V, produces a set of tracks on the video. We will use $V^i_{m:n}$ to denote the sequence of patches in track i starting from frame m and ending at frame n. The image patch for track i in frame m is denoted as V^i_m (see Figure 4). In this work, for obtaining tracks, we use the tracker proposed in [250] which is trained in an unsupervised manner. f_θ, realized via convolutional neural networks, denotes the desired feature embedding that establishes visual correspondence across *different* videos.

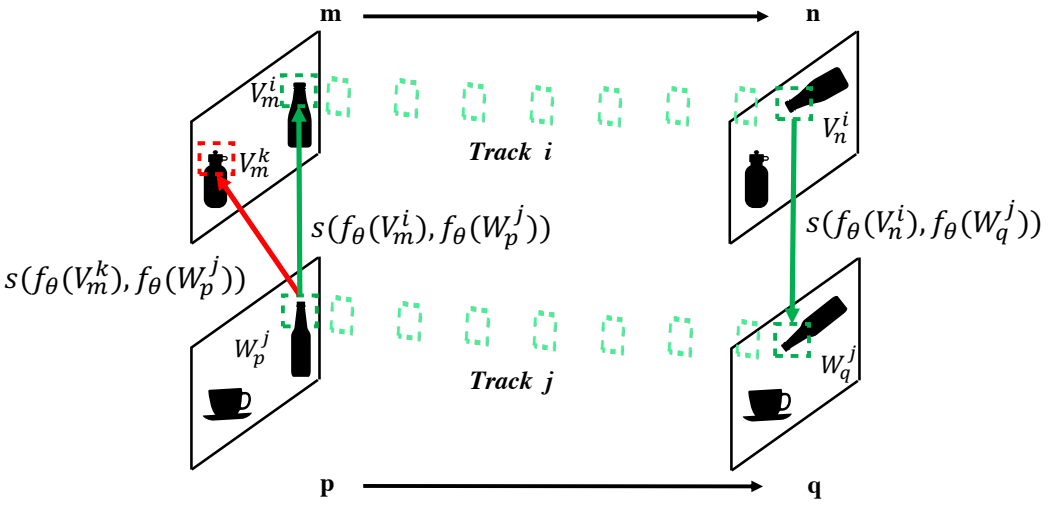


Figure 4: Formulation: The score of a cycle is sum of the scores of two jumps as per f_θ.

Consider the cycle shown in Figure 4: $V_m^i \rightarrow V_n^i \rightarrow W_q^j \rightarrow W_p^j \rightarrow V_m^k$. This cycle has following jumps: forward-tracking in V, matching V to W, backward-tracking in W and matching back from W to V. We represent this cycle as $\{V_{m:n}^i, W_{p:q}^j, V_m^k\}$. The score of this cycle can be expressed as the sum of patch similarities of the matches involved. However, note that the first and third matches in a cycle are extracted using off-the-shelf tracker, therefore do not depend on f_θ and can be assumed to have a constant score. Therefore, the final score of a cycle can be computed using cosine similarity s as:

$$S(\{V_{m:n}^i, W_{p:q}^j, V_m^k\}) = \underbrace{s(f_\theta(V_n^i), f_\theta(W_q^j))}_{\substack{\text{Jump from video } V \text{ (frame } n, \\ \text{patch } i) \text{ to video } W \text{ (frame } q, \\ \text{patch } j)}} + \underbrace{s(f_\theta(W_p^j), f_\theta(V_m^k))}_{\substack{\text{Jump from video } W \text{ (frame } p, \\ \text{patch } j) \text{ to video } V \text{ (frame } m, \\ \text{patch } k)}} \tag{5.1}$$

Given a starting patch V_m^i and an ending patch V_m^k, there can be numerous cycles depending on the length n considered in video V, the segment (p, q) of video W considered and the track j chosen in video W. When the patches V_m^i and V_m^k are highly overlapping, we expect the best cycle to have a high score. On the other hand, when these patches do not overlap, we want all the cycles to score low. We formulate this objective to optimize f_θ as a margin loss. First, for the pair of patches V_m^i, V_m^k, we compute the score of the best cycle as:

$$\kappa(V_m^i, V_m^k) = \max_{n,p,q,j} S(\{V_{m:n}^i, W_{p:q}^j, V_m^k\}) \tag{5.2}$$

The margin loss can then be formulated as:

$$\begin{aligned} &\max\left[0, -\kappa(V_m^i, V_m^{i+}) + \kappa(V_m^i, V_m^{i-}) + \delta\right] \\ &\forall i_+, i_- : \text{IoU}(V_m^i, V_m^{i+}) \geq 0.5 \text{ and } \text{IoU}(V_m^i, V_m^{i-}) < 0.5 \end{aligned} \tag{5.3}$$

where, δ is the fixed margin. This can be optimized using stochastic gradient descent, to learn function f_θ.

We found that using a *soft version of the max function* (Γ as defined below) instead of the *max function* in Eq. 5.2 was important for training. Soft version of max function, Γ is defined as follows:

$$\Gamma(\mathbf{x}) = \sum_c \mathbf{x}_c \frac{e^{\mathbf{x}_c}}{\sum_{c'} e^{\mathbf{x}_{c'}}} \tag{5.4}$$

Here c represents a cycle and $\mathbf{x}_c$ represents the score of that cycle. This prevents the model from getting stuck in the local minima of greedily boosting the single best cycle. The soft version of max also allows computation of gradients *w.r.t* all patches that participate in score computation, thereby updating the representations of a larger number of samples.

5.3.2 Using Features for Spatio-Temporal Alignment

The representation f_θ trained using our approach can be used to extract cross-video correspondences at the level of patches, tracks, frames and videos:
Patch Correspondence. f_θ can be used to correspond image patches. As f_θ learns features sensitive to state of the object, it allows us to correspond and retrieve objects that are in the same state. See Section 5.4 for results.
Track Correspondence. Cycles in our formulation correspond tracks with one another. Given a set of tracks in videos V and W, we correspond each track i in video V, to the track in W that maximizes the score in Eq. 5.1:

$$\arg\max_j \left(\max_{n,p,q} S\left(\{V^i_{m:n}, W^j_{p:q}, V^i_m\}\right) \right). \tag{5.5}$$

Temporal Alignment. We compute the similarity between a given pair of frames (V_m and W_p) in the two videos V and W by computing the total similarity between corresponding patches in the two frames:

$$T(V_m, W_p) = \sum_i \max_j s\left(f_\theta(V^i_m), f_\theta(W^j_p)\right). \tag{5.6}$$

These frame-level similarities can be used to obtain sub-video alignments. For example, if one wants to align K frames in video 1 to K frames in video 2 we can pick temporally-consistent top-K correspondences.
Video Retrieval. f_θ provides a natural metric for retrieving videos. Given a query video V and a set of videos $\mathcal{W}$, we retrieve the most similar video to V, by maximizing the total frame-level temporal alignment score:

$$W = \arg\max_{W \in \mathcal{W}} \sum_m \max_p T(V_m, W_p). \tag{5.7}$$

5.4 Experiments

Our goal is to demonstrate that we can align videos in space and time by leveraging f_θ learned using cross-video cycle-consistency supervision. Quantitatively measuring performance of dense spatio-temporal alignment is challenging due to the lack of ground-truth data. Therefore, in order to demonstrate the effectiveness of our approach, our experiments involve factored quantitative evaluations, and qualitative visualizations. More specifically, we study performance of our model at track correspondence, and temporal alignment.
Datasets: We perform alignment experiments on the Penn Action Dataset [273] and the Pouring Dataset [210].
Baselines: We compare our learned features to three alternate popular feature learning paradigms that focus on:

- semantics (image classification, object detection),
- local patch appearance (object trackers),
- motion and therefore object transformations (action classification models).

For models that capture semantics, we compare to ImageNet-trained ResNet-18 model `layer4` features (earlier layers do not improve results significantly), and a Mask-RCNN [92] object detection model trained on the MS-COCO [136] dataset. These models capture rich object-level semantics. For models that capture local patch appearance, we compare to features obtained via learning to track from Wang *et al*. [250]. Lastly, for models that focus on motion, we compare to features obtained via training for action classification on Kinetics [111] (ResNet-3D-18), and for frame-level action classification on Penn Action Dataset. Note, these together represent existing feature learning paradigms. Comparisons to these help us understand the extent to which our learned representations capture object state. Lastly, we also compare to recent paper from Dwibedi *et al*. [54] which only performs temporal alignment. To demonstrate the need for also modeling spatial alignment, we a consider a spatial downstream task of detecting the contact point between the thumb and a cup in the Pouring Dataset (since models from [54] are only available for the Pouring Dataset).

Tracks: We use an off-the-shelf tracker[250] to obtain tracks on videos for training and testing. Since we wish to focus on the foreground of videos for alignment, the pre-processing requires extracting tracks of foreground patches. To show robustness to patch extraction mechanism, we experiment with the following patch generation schemes (use of more sophisticated schemes is future work). For the Penn Action dataset, we track patches sampled on human detections from a Mask-RCNN detector [92]. For the Pouring dataset, we perform foreground estimation by clustering optical flow. As an ablation, we also experiment with ground-truth tracks of human keypoints in Penn Action dataset.

Training Details. We use a ResNet-18 [95] pre-trained on the ImageNet dataset [38] as our backbone model, and extract features from the last convolutional layer using RoI pooling. These features are further processed using 2 fully connected layers (and ReLU non-linearities) to obtain a 256-dimensional embedding for the input patch. We optimize the model using the Adam optimizer [116], with a learning rate of 0.0001, and a weight decay of 0.00001. We train the model for 30000 iterations on the Penn Action dataset and 500 iterations on the Pouring Dataset with each batch consisting of 8 pairs of videos. For computational efficiency, we divide each video into 8 temporal chunks. During training, we randomly sample one frame from each chunk to construct a sequence of 8 frames.

5.4.1 Qualitative Results

First we show some qualitative results of correspondences that can be extracted by our approach. Figure 5 shows some examples. We show the query frame on the

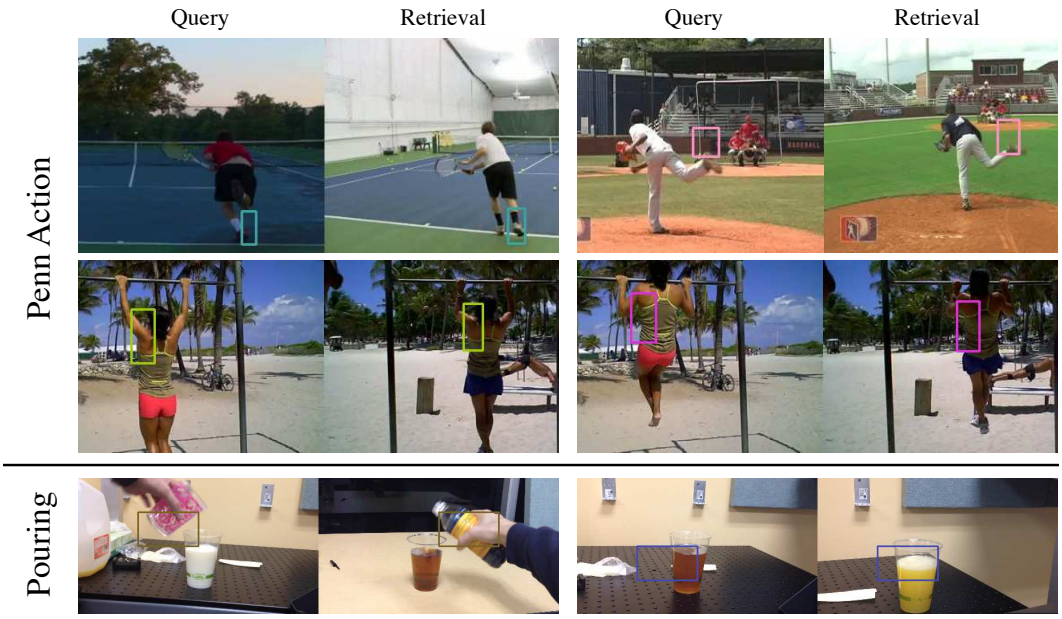


Figure 5: Nearest neighbor patch correspondence. For random patches in query videos (left), we show the nearest neighbor patch across all frames (right) in a video retrieved using our method. We observe that our learned feature space is sensitive to the state of the object. Example in row 2 further highlights this point where our features match similar appearing patches differently based on the state of the person in the query. Row 3 shows an example from the Pouring dataset.

left, and the corresponding nearest neighbor patch across all frames on the right. We observe that our model matches based on both the appearance and the state of the object. Next, we show that our approach can temporally align videos. Figure 6 visualizes temporal alignment on the pouring task.

Finally, we qualitatively compare the correspondence using our features compared to ImageNet and action classification features. Figure 7 shows the spatio-temporal alignment on Penn-Action dataset. Given a query video, we retrieve the most similar video based on spatio-temporal alignment. We use human keypoints to form tracks. The spatial alignment is shown by shape and color of keypoints, and the temporal alignment is shown in vertical (frames on top and bottom are temporally aligned). As compared to baseline methods, our approach is able to retrieve a more similar video, better align the frames in time, and more accurately correspond tracks with one other.

5.4.2 Quantitative Evaluation

Evaluating Temporal Alignment. Given a query video, we first obtain the closest video and then do temporal alignment as described in Section 5.3.2. For a given

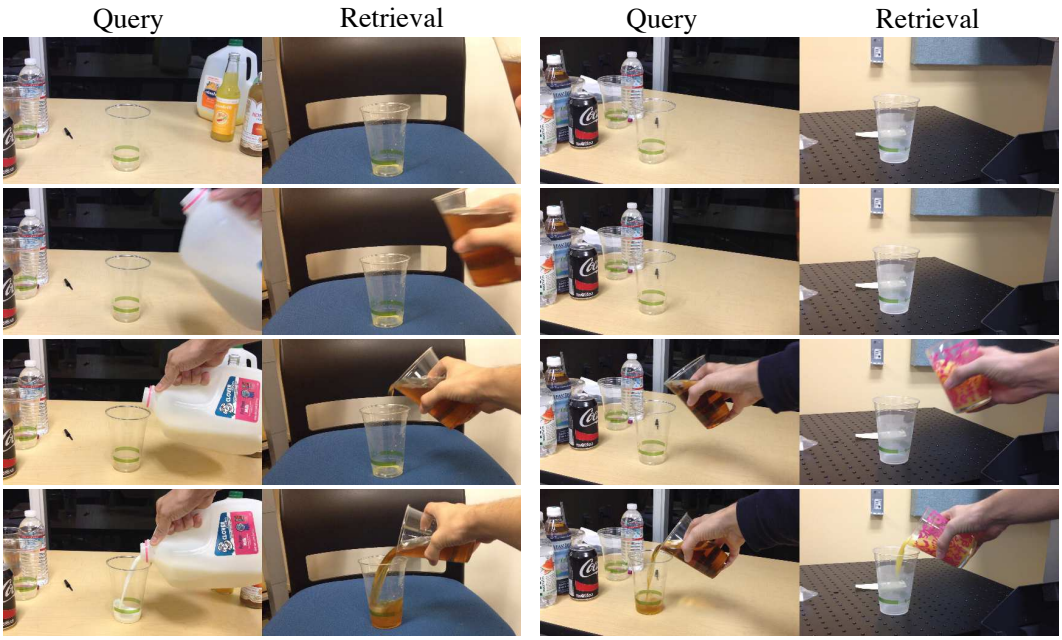


Figure 6: Qualitative Results on Pouring Dataset: We show qualitative examples of retrieval and temporal alignment (query on left, retrieval on right) from the Pouring Dataset, based on the similarity metric learned by our model.

Table 1: Temporal Alignment on Penn Action Dataset [273]: We measure temporal alignment by measuring alignment in keypoint configuration at point of temporal alignment.

Method	**Temporal Alignment Error** $\downarrow$
ImageNet features	0.509
Features from Mask-RCNN [92]	0.504
Features from cycle-consistency based tracker [250]	0.501
Features from Kinetics [111] action classification model	0.492
Features from action classification	0.521
Our features (using tracks from [250] to train)	**0.448**

pair of frames V_m and W_p, we densely sample foreground patches and compute an average similarity using f_θ as the feature extractor. We can then temporally align the frames of videos V and W using the similarity measure in Eq. 5.6. Starting with 8 frames each, we align 4 frames from the query video to 4 frames in the retrieved video.

We evaluate the quality of the temporal alignment, by comparing the pose configuration of the human in the aligned frames (*i.e.* is the human in the same state in query and retrieved video). More specifically, we use the ground truth keypoint annotations to estimate and compare the angle between the surrounding limbs at left and right knee, left and right elbow, left and right hip and the neck. We report the average absolute angle difference over all joints (lower is better) in Table 1. We observe

Figure 7: We show qualitative examples of retrieval and spatio-temporal alignment on the Penn Action Dataset to compare different feature spaces. The top row shows snapshots from the query video, the second row shows video retrieved from our model (trained on tracks from [250]), the third row shows retrievals using ImageNet features, and the fourth row shows retrievals using features obtained by finetuning on the dataset using the class labels. Each columns shows temporally aligned frames, while coloured markers show spatial alignment. For all methods, we use keypoint tracks at inference time in order to showcase spatial alignment.

that features learned using our proposed cross-video cycle consistency leads to better temporal alignment than features from ImageNet classification, Mask-RCNN [92], frame and video classification, and intra-video correspondence [250].

Evaluating Spatial Alignment with Patches. Our proposed model can also perform spatial alignment. Given temporally aligned video frames, we use the similarity function s with the learned features f_θ to correspond image patches in temporally aligned video frames. We measure the quality of alignment by counting how many of the corresponding keypoints lie in aligned patches. We report the average accuracy using various feature extractors in Table 2.

Evaluating Keypoint Tracks Correspondence. Given a track in query video V, a spatially aligned track in reference video W can be identified, by using the same similarity function s with the learned features f_θ. We evaluate this by aligning keypoint tracks provided in the Penn Action dataset. Given a track of a keypoint in video V, we measure the accuracy which the aligned track corresponds to the same keypoint in video W. We report this accuracy in Table 3. Note that this alignment uses keypoint tracks only for performing inference and quantitative evaluations.

Table 2: Spatial Alignment on Penn Action Dataset [273]: We measure spatial alignment by measuring how accurately we can match keypoint by corresponding random patches between query and reference videos.

Method	Spatial Alignment Accuracy ↑
ImageNet features	0.153
Features from Mask-RCNN [92]	0.202
Features from cycle-consistency based tracker [250]	0.060
Features from Kinetics [111] action classification model	0.150
Features from action classification	0.157
Our features (using tracks from [250] to train)	**0.284**

Model was trained using tracks from Wang *et al*. [250] on foreground patches as before.

5.4.3 Ablations

Additionally, we also compare to 3 variants of our model, to understand the effectiveness of the different parts of our model. We discuss spatial alignment results (as measured by accuracy at keypoint track correspondence).

Impact of quality of tracks used during training. We experiment with using tracks derived from ground truth key-point labels during training. We find that this leads to better features, and achieves a keypoint track correspondence accuracy of 0.650 *vs*. 0.551 when using tracks from Wang *et al*. [250]. The next ablations also uses ground-truth tracks for training.

Not searching for temporal alignment during training. Our formulation searches over temporal alignment at training time. This is done by searching for frames to jump between the two videos (max over n, p and q in Eq. 5.2). In this ablation, we learn features without searching for this temporal alignment, *i.e*. simply assume that the frames are aligned. The resulting features are worse at spatial alignment (keypoint track correspondence accuracy of 0.584 *vs*. 0.650).

Importance of reference video retrieval. As a first step for spatio-temporal alignment, we retrieve the best video to align. In order to ablate the performance of this retrieval task, we measure the average keypoint track correspondence accuracy by

Table 3: Track Correspondence on Penn Action Dataset [273]: We measure spatial alignment by measuring how accurately we can match keypoint tracks across videos. We compare our learned cross-video features with those obtained by pre-training on ImageNet and for action classification on the Penn Action dataset.

Method	Track Correspondence Accuracy ↑
ImageNet features	0.252
Features from action classification	0.110
Our features (using tracks from [250] to train)	**0.551**

aligning all the queries to all reference videos. We observe that the accuracy drops by 15% indicating that the retrieval step is effective at choosing relevant videos.

5.4.4 Comparison on Pouring Dataset

Method	Accuracy ↑
ImageNet features	27.1%
TCC [54]	32.7%
Ours	**38.6%**

We now show the necessity of learning spatial alignment by considering a spatial downstream task of predicting contact locations. We annotate the Pouring Dataset [210] with locations of the contact point between the human thumb and the cup. We train a linear 1×1 convolution layer on the spatial features in various models to predict the probability of the contact point. We compare features from our model that are sensitive to locations of objects, *vs*. features from Dwibedi *et al*. [54] that only focus on learning good temporal alignment. We split the data into 210 training and 116 test images. We train a linear classifier on top of different features. Table shows the Percentage of Correct Keypoint (PCK) [262] metric for the localization of this contact point within a $16\text{px} \times 16\text{px}$ neighborhood of the ground truth. We see that our features perform better than both ImageNet features, and features from [54]. Thus, features that are sensitive to object locations are essential for obtaining a rich understanding of videos.

5.5 Discussion

In this work, we address the problem of video understanding in the paradigm of "understanding via associations". More specifically, we address the problem of finding dense spatial and temporal correspondences between two videos. We propose a weakly supervised cycle-consistency loss based approach to learn meaningful representations that can be used to obtain patch, track and frame level correspondences. In our experimental evaluation, we show that the features learned are more effective at encoding the states of the patches or objects involved in the videos compared to existing work. We demonstrate the efficacy of the spatio-temporal alignment through exhaustive qualitative and quantitative experiments conducted on multiple datasets.

Part III

Representation beyond Semantics

Chapter 6

The Functional Correspondence Problem

6.1 Introduction

> To perceive an affordance is not to classify an object. The fact that a stone is a missile does not imply that it cannot be other things as well. It can be a paperweight, a bookend, a hammer, or a pendulum bob.
>
> *James J. Gibson*

Computer vision and visual representation learning has been bound by shackles of semantic categories. Our training data is built with semantic categories - ImageNet has 1K categories of breeds of dogs, cats and mushrooms. Our supervision is semantic categories. And our evaluation tasks are semantic – image classification, object detection, image segmentation and list goes on. So it is not surprising that our approaches are bound by the limits of semantic categories. Our representations are not effective in capturing affordances for robotics tasks. And our representations fail to generalize effectively to new object categories due to focus on learning intra-class invariances. On the other hand, humans have marvelous ability to think beyond categories. We can use a screwdriver for opening screws but also to clean printer, hammer nails and what-not. Clearly, our current semantically-driven computer vision needs rethinking.

In classical computer vision, semantics did not play such an important role. Instead, correspondence was cited as one of the most important tasks in the field of computer vision. It is also the fundamental goal of visual representation learning – an embedding space where similar objects/parts/pixels have similar embedding. In an anecdotal conversation about the three most important problems in computer vision,

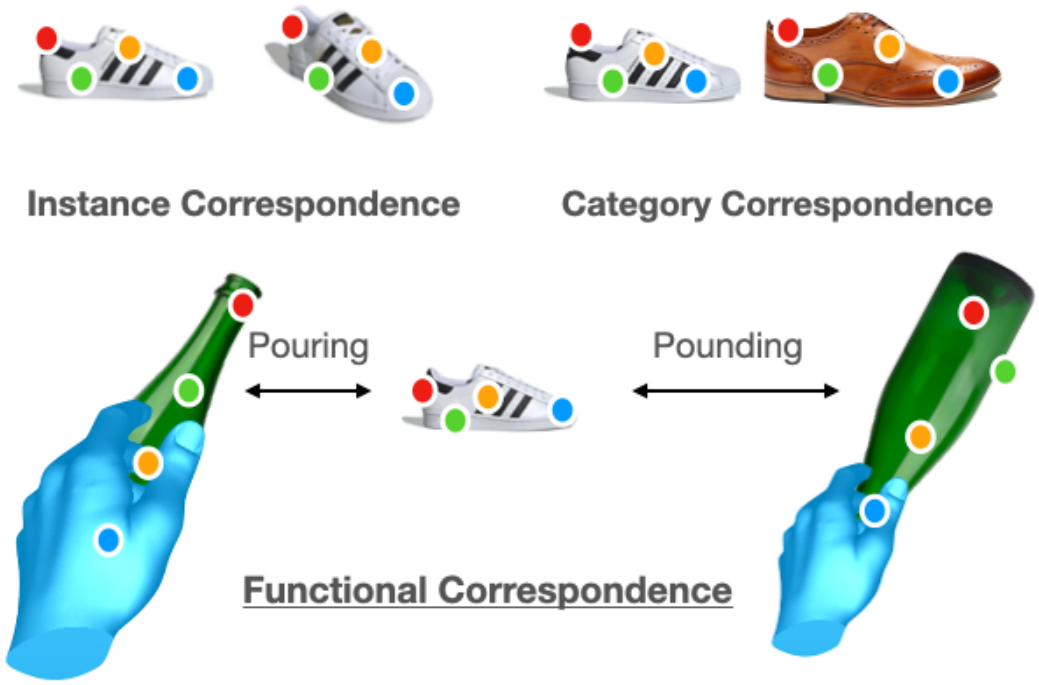


Figure 1: Given a pair of images, functional correspondence establish correspondence between points that are *functionally* the same. In this example, we hold the body of the bottle when pouring but the neck when pounding, therefore we can establish correspondence (bottle body, shoe front) for pouring and correspondence (bottle neck, shoe front) for pounding.

Takeo Kanade stated that they are "*Correspondence, Correspondence, Correspondence*". Yet, this fundamental task of visual correspondence is ambiguous and ill-defined. What is visual correspondence? Does there exist correspondence between any pair of images? The visual correspondence problem is most well-defined and often studied in context of tracking and multi-view reconstruction where the goal is to create correspondences between two images of same object [243]. It has also been studied in the context of semantic categories where the goal is to create correspondences between images of object instances from same categories [114, 170]. But it often stops at cat what are the right correspondences between two seemingly different object categories (for example, a bottle and a shoe)?

In contrast, we humans can identify correspondences between semantically different objects. We unconsciously use this ability to transfer our object manipulation skills to novel objects in order to efficiently accomplish everyday tasks. Specifically, humans possess three interesting capabilities: (a) the ability to visually infer affordances for objects, (b) the ability to generalize beyond semantic categories and (c) the ability to adapt affordances for different tasks. In order to facilitate exploration of these capabilities, we introduce the problem of functional correspondence. Given images of two objects, we ask a simple question: for a given task, what would be

the set of correspondence between two objects? For example, the correspondences between shoe and bottle for the task of pouring are shown in the figure 1. The grasp locations are shown by green, storage by orange and pouring spout by red keypoints. On the the other hand, the correspondences between shoe and bottle for the task of pounding (hitting with a force) are quite different and shown in figure 1. Note that the correspondence between two objects is driven by both 3D shape and physical/material properties.

We also introduce a new dataset called FunKPoint (Section 6.3). FunKPoint has ground-truth keypoints labeled for 10 tasks across 20 object categories. We also propose a modular task-driven architecture. More specifically, our modular architecture computes the image representation given an input task. We show our architecture is highly effective in modeling functional correspondences although there is still a significant gap with respect to human performance. But most importantly, in proof-of-concept experiments, we demonstrate the underlying promise of learning functional correspondence. Because our task has functional supervision and there is cross-category supervision, our representation can outperform semantically-learned representations for few-shot learning.

6.1.1 Why Functional Correspondence?

In this paper, we introduce the problem of functional correspondence. We believe this task forms the core of visual learning because of the following reasons:

(a) **Object Affordances and Functional Representations:** Ability to predict object affordances is a cornerstone of human intelligence and a key requirement for robotics tasks. The task of functional correspondence allows us to learn functional representations useful for robotics tasks. But more importantly, beyond predicting primary affordances (screwdriver is used for screwing), humans are really good at predicting secondary affordances (how we can use novel objects to fulfil the task – e.g. using screwdriver to clean paper jam in printer). Modeling functional correspondences across different object categories should help in predicting novel use of objects.

(b) **Generalization Beyond Semantic Categories:** Unlike other vision tasks such as object classification/detection or even learning 3D from image collections, this task cuts across object semantics. It attempts to model commonalities across different categories of object and hence open up the possibility of generalization beyond semantic categories.

(c) **Task-Driven Representation:** Finally, the ground-truth is conditioned on the task itself, the correspondences between pair of objects depends on how you envision using these objects. This allows us to formulate a task-driven representation (unlike current existing task-agnostic ConvNet representations).

6.2 Related Work

Correspondences: The correspondence problem has always been a focus of the computer vision community, and many sub-problems have been proposed with solutions offered. The classical correspondence problem establish correspondence between different views of the same object. Such correspondence is crucial for multi view geometry based algorithms and are typically solved by matching local descriptors of interesting points [13, 16, 90, 143, 144]. More recently, researchers looked into category-level correspondence [114, 170, 196, 197, 230], which does not restrict correspondence to a single instance. Such methods often model correspondence in deep feature space, and relies on simulated transformations for training. Because object of the same category usually perform similar actions, our work could also establish correspondence at category level. However, we consider any object, regardless of its object class, could correspond if they share parts that have similar functional semantics. Thus, our *functional correspondence* could be considered more general in that we also establish cross-category correspondences.

Dense correspondence between pixels across video frames (optical flow) is also studied as a separate problem. Traditionally, the optical flow estimation problem is addressed as an energy minimization problem based on color constancy [18, 98, 194, 219]. Recent optical flow estimation algorithms make use of neural networks [10, 101, 107, 112] as models and explores self-supervision as the training method [74, 139]. Another line of work focuses on the mid-level optical flow problem [104, 123, 243] where consistency between the regions around the pixels is also considered. Such approaches often leverage the spatial temporal coherence nature of videos to provides a natural supervision signal. However, because the main training loss is usually a photometric loss, the learned correspondence is inevitably local. In this work, we try to establish a higher level *functional correspondence*. Such correspondence involves a knowledge of object affordances, which is still hard to learn from unlabeled raw videos.

Functional Representations and Affordances: The core idea of affordances was introduced by James J. Gibson [67]. Gibson described object affordances as "opportunities for interactions". Inspired by Gibson's idea of affordances, a long-term goal for robotic perception has been to perform function recognition [195, 220]. Approaches such as [220, 254] used manually-defined rules to predict affordances. However, these approaches were too brittle and failed to generalize.

In recent years, with the advances in 3D scene understanding and with the large-scale availability of interaction data the idea of affordances has been revisited as well [34, 63, 82, 87, 274]. Approaches such as [87, 274] have attempted to use 3D understanding followed by affordance estimation. More recently, approaches have tried to collect large-scale data for affordance estimation [245] and used ConvNets to predict affordances in the scene [62, 240]. AffordanceNet [44] simultaneously localizes multiple objects and predicts pixel-wise affordances by training on a large-

scale dataset with affordance labels. In contrast, our approach focuses on affordances as a vehicle to target generalization beyond semantic categories and learn task-driven representations. More specifically, we target using primary and secondary object affordances to learn visual correspondences across different object categories. Our work is also closely related to some recent work in robotics which focuses on extraction of keypoints for robotics tasks [61, 149]. However, in most of these scenarios, the goal is to learn to predict dense keypoints/correspondences across two objects of same categories. In this work, we focus on the more general problem of how to do task-driven functional correspondences across multiple object categories.

Task-Driven Representations and Modular Networks: Classification models in deep learning have largely been trained as discriminative models[96, 119, 216]. Recently, energy based models[128] have gained popularity and demonstrated success on image classification[83], continual learning[53], compositional zero-shot learning[185, 252] and generative modeling of text[11]. In [185, 252], the key idea is to construct a task-dependent (or label-dependent) neural network for classifying whether an image belongs to the considered label. In [185], this compatibility of an image x to a label y is computed using a sequence of neural network modules which are reweighted using a function of the considered label y. The modular architecture proposed in [185] allows sharing of learned filters across different labels which is crucial for domains where the labels are heavily related. These modular neural networks have also demonstrated great success in multi-task reinforcement learning[43, 261] where modules are shared among related tasks to learn policies efficiently. For estimating functional correspondences, we require representations that vary according to the considered task. Therefore, we adopt a similar modular task-driven architecture for learning a task-dependent representation which also allows us to share neural network modules between related tasks.

6.3 The FunKPoint Dataset

To explore the study of functional correspondences, we present a novel dataset: FunKPoint (short for Functional KeyPoints). FunKPoint consists of 2K objects covering 20 object categories. In order to learn and evaluate functional correspondences between pairs of images, we require dense human annotations of such correspondences. However, such an approach is unscalable due to the quadratic number of image pairs and pixels. Instead, we first identify 5 semantically meaningful points that are essential for each task. For each task, we then collect annotations for the 5 keypoints for each relevant object image. Figure 2 shows examples from the dataset. Note that a single image could be labeled differently for each task. In total, around 24K such labeled keypoints are obtained. Any two objects that can be used to perform an action are then used to establish a correspondence relationship (w.r.t. that action). This correspondence between two images, conditioned on a specific action, is referred to as a *Functional Correspondence*. For example, in the top left figure of Fig. 2, both

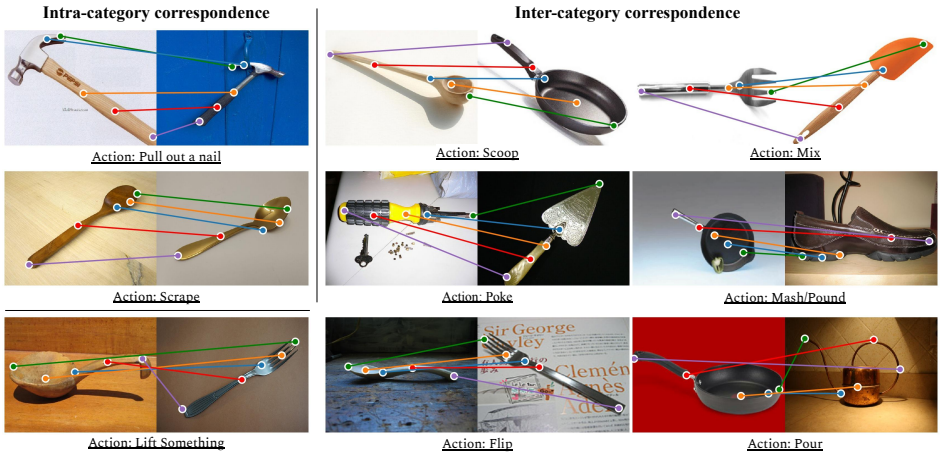


Figure 2: The FunKPoint Dataset: Here we present examples from the proposed dataset. For each image and associated task, we collect human annotations for 5 keypoints. Associating keypoints between images provides us with numerous intra-category and inter-category functional correspondences.

hammers can be used to pull out a nail, so a *functional correspondence* relationship (consists of 5 pairs of corresponding points) could be established between the two objects. Similarly, both the spoon and the frying pan (Fig. 2 top-middle) can be used to scoop things, so we can also generate a *functional correspondence* relationship between them.

Data Collection First, we curate an action vocabulary consisting 10 common tasks (or actions). Our action vocabulary is inspired from the TaskGrasp [166] dataset, which focuses on task-dependent robot grasps. Therefore, the 10 actions in our vocabulary are not only common, but also useful as a benchmark in robotics. For each action, we identify 5 object categories that can be used to perform that task. Note that many object categories can be relevant for multiple tasks. This allows us to generate different correspondence for the same objects under the condition of performing different tasks. For example, the object category *frying pan* has 2 possible actions (among others): *Scoop* and *Mash/Pound*. The rim of the pan is a functional keypoint that is important for scooping, but for pounding, the bottom of the pan becomes the relevant functional keypoint. See Table 1 for the list of 20 objects and their associated tasks.

For each of the 20 object categories, we collect 100 images from the ImageNet dataset [39], but supplementing with creative commons images from Google image search to reach 100. Note we manually filter out images that contain multiple object instances, missing parts or occluded parts.

We use Amazon Mechanical Turk to collect human annotations for the keypoints. Each (image, task) pair is labeled with the 5 functional key points as well as a choice of labelling difficulty (between easy, medium or hard). In the interface, we provide a

Table 1: Object categories corresponding to 10 action classes used in FunKPoint:

Action	Objects
Pour	bottle, frying pan, watering can, cup, dustpan
Scoop	spoon, basket, cup, frying pan, shoe
Mix	spoon, tablefork, spatula, tongs, whisk
Mash/Pound	bottle, frying pan, hammer, ladle, shoe
Lift Something	ladle, tablefork, basket, tongs, dustpan
Scrape	scraper, tablefork, spatula, trowel, spoon
Poke	scraper, watering can, screwdriver, trowel, scissors
Brush/Dust	whisk, scrub brush, toothbrush, scraper, spoon
Pull out a nail	hammer, ladle, scissors, frying pan, tablefork
Flip	spoon, tablefork, spatula, ladle, tongs

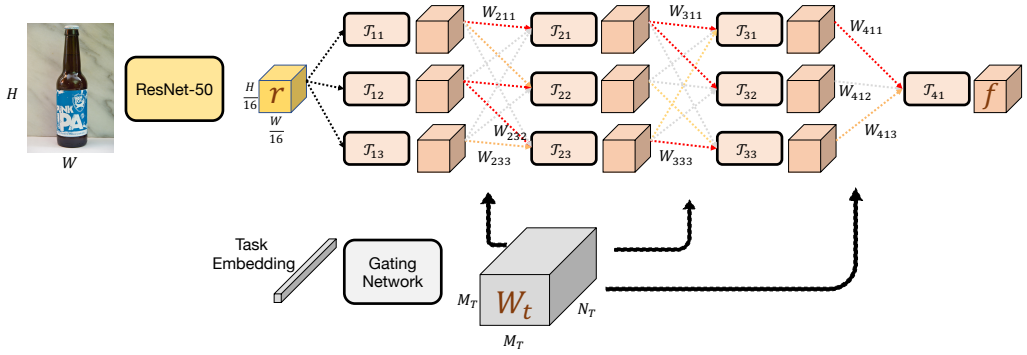


Figure 3: Approach: We use a task-driven modular architecture for learning functional representations. We show that the learned representation can be effectively used to identify functional correspondences between objects. Note that we show 3 modules per layer here only for illustration, see supplementary material for the chosen value for this hyperparameter.

simple definition for each point, current action, and also examples of labeled images. See supplementary material for a visualization of the interface. As explained, each object could be associated with multiple actions (see Supp. for statistics). From the collected data, we create a train split containing 4044 (image, task) pairs and a test split contains 741 (image, task) pairs.

6.4 Approach

Estimating *semantic* correspondences has been well studied in the past. Most approaches[114, 170, 196, 197, 230] involve learning a pixel or patch level representation which can be used to match corresponding points on similar objects. As we will demonstrate via experiments, for the problem of functional correspondence, such representations are not suitable. We wish to estimate correspondences even across semantically varied objects and second, the correspondences vary according to the task being performed. Therefore, we propose an approach that produces task-driven representations that

can be used to find functional correspondences across varied objects.

First, we formalize the problem setup of functional correspondence. Consider two images $\mathcal{I}$ depicting an object o and $\mathcal{I}'$ depicting an object o' such that both objects can be used to perform task t. Given any point p on object o, the goal of the functional correspondence problem is to estimate the functionally corresponding location p' on object o'. However, as described in Sec 6.3, we only have access to correspondences for specific keypoints due to the prohibitive cost of annotation. Therefore, for each task $t \in \mathcal{T}$ that can be performed with object o, we have a set of functional keypoints $\{p_{t1}, p_{t2}, ..., p_{tK}\}$. The goal of the functional correspondence problem can then be restated as estimation of the functionally corresponding locations of the keypoints $\{p'_{t1}, p'_{t2}, ..., p'_{tK}\}$ on object o'.

Recently, task-driven classifiers have gained popularity for the problem of zero-shot learning[185, 252]. Taking inspiration from these approaches, we adopt a similar approach to learn a task-driven representation. More formally, we propose a model $\mathcal{F}_\theta$ with parameters θ which takes as input an image $\mathcal{I}$, a task t and outputs a representation $f = \mathcal{F}_\theta(I, t)$. In order for the representation f to be useful for functional correspondence, we propose to learn the parameters θ using the dataset presented in Sec 6.3. The goal is to ensure that the representation f at location p of an image $\mathcal{I}$ and location p' of an image $\mathcal{I}'$ are identical only when p, p' are functional correspondences. To achieve this, we propose a contrastive learning objective [174] as follows:

$$L(\mathcal{I}, \mathcal{I}', t, \theta) = \sum_{k=1}^{K} -\log \frac{\exp\left(f[p_{tk}]^\intercal \; f'[p'_{tk}]\right)}{\sum_{p'} \exp\left(f[p_{tk}]^\intercal \; f'[p']\right)} \tag{6.1}$$

$$\text{where} \;\; f = \mathcal{F}_\theta(\mathcal{I}, t), \qquad f' = \mathcal{F}_\theta(\mathcal{I}', t)$$

$$f[p] \text{ is the indexed feature } f \text{ at spatial location } \; p$$

here p_{tk}, p'_{tk} are the k-th functional keypoints for task t in image images I, I' respectively. Intuitively, minimizing this objective effectively minimizes the distance between features of functionally corresponding points in the two images (numerator) and maximizes the distance between the feature of a keypoint and the features at all non-corresponding locations p' (denominator). Note that the locations p' includes all keypoint and non-keypoint locations.

This general contrastive learning formulation can be applied to any convolutional neural network architecture that jointly encodes the image $\mathcal{I}$ and task t. In order to model the dependencies between functional keypoints of different tasks, we propose to use a modular architecture allowing us to share filters across tasks. We adopt the architecture proposed in [185]. For the sake of completeness, we describe the architecture here in detail.

6.4.1 Implementation Details

Figure 3 shows an overview of our proposed model $\mathcal{F}$. For an image $\mathcal{I}$, we first extract task-agnostic features using a ResNet trunk upto the `conv4_x` layer (defined in [96]) as $r = R(\mathcal{I})$. For an image with dimensions $H \times W$, the representation r has spatial dimensions $H/16 \times W/16$ with a C dimensional feature at each location. The representation r is then processed by a modular task-driven feature extractor $\mathcal{T}$ to produce the final features $f = \mathcal{T}(\mathcal{I}, t)$.

The modular task-driven feature extractor $\mathcal{T}$ consists of N_T layers with each layer comprising of M_T *modules* except for the last layer which comprises of a single module. A module can be any differentiable operation. In our proposed architecture, we use convolution layers with batch normalization[102] and ReLU activation functions (see supplementary for details of kernel size, number of filters, etc). We denote the j-th module of the i-th layer as $\mathcal{T}_{ij}$. Given a task-dependent weight tensor $W_t \in \mathbb{R}^{N_T - 1 \times M_T \times M_T}$ for task t, the output of a module $\mathcal{T}_{ij}$ is computed as:

$$o_{ij} = \mathcal{T}_{ij}\Big(\sum_{k=1}^{M_T} W_t[i, j, k] * o_{(i-1)k} \Big) \tag{6.2}$$

Intuitively, the input to a module is a weighted sum of the outputs of the modules in the previous layer. For modules in the first layer, the inputs are taken as the task agnostic representation produced previously *i.e.* $o_{0k} = r$. Finally, the output task-dependent representation is taken as the input of last module $o_{(N_T)1}$.

Note that we assumed that we are given a task-dependent weight tensor $W_t \in \mathbb{R}^{N_T - 1 \times M_T \times M_T}$. This weight tensor is estimated using a separate fully-connected neural network $\mathcal{G}$ known as the gating network (see supplementary for parameter details). The gating network takes as input a task-embedding t and outputs the weight tensor as $W_t = \mathcal{G}(t)$. As explained in [185], the input weights to each module ($W[i, j, :]$) needs to be projected to the probability simplex using a softmax operation to encourage separate paths for different tasks. In summary, the output feature representation is computed as $f = \mathcal{T}\big[R(\mathcal{I}), \mathcal{G}(t)\big]$.

6.4.2 Training

The objective presented in Equation 6.1 is used to learn the parameters θ which comprises of the gating network $\mathcal{G}$, modules $\mathcal{T}_{ij}$ and task embeddings t (which are initialized randomly for each task). We pretrain the ResNet model R on ImageNet[39] and fix its parameters. We optimize the parameters using SGD with a learning rate of 0.01, weight decay of 0.00001 and momentum of 0.9. Each batch consists of 256 pairs of images randomly sampled from the training split of the FunKPoint dataset.

6.5 Experiments

Modeling functional correspondences provides numerous practical benefits. In this section, we demonstrate this by evaluating our presented model on a suite of tasks. First, we show that our model can effectively identify functional correspondences and outperform numerous baseline methods. We then demonstrate the efficacy of our learned representation for few-shot learning, grasp prediction, and ADROIT manipulation tasks [188]. Note that due to the domain of our training data, we focus our experiments on manipulation related datasets for all tasks.

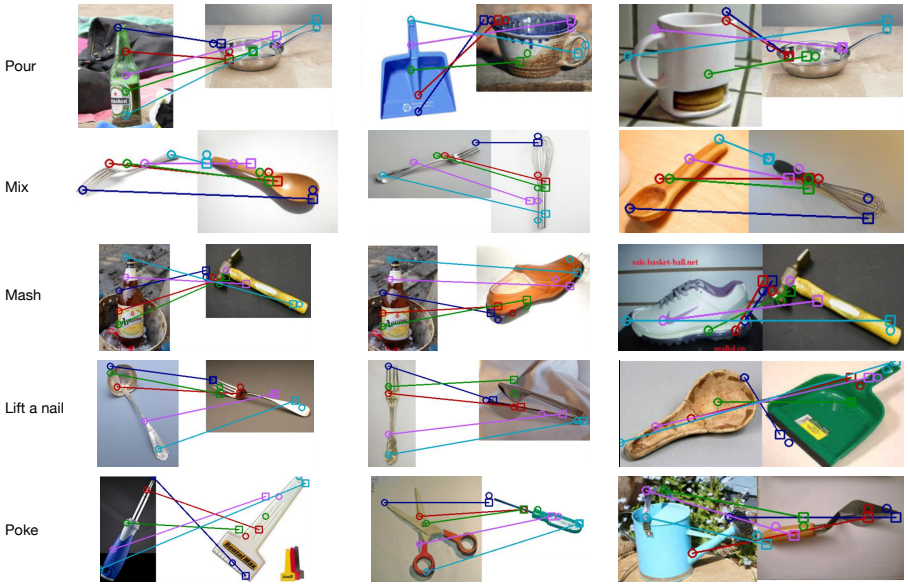


Figure 4: Qualitative Correspondences. We demonstrate some qualitative correspondences generated by our algorithm. Squares indicate predicted functionally corresponding keypoints in the second image and circles indicate ground truth keypoints in both images. Notice how the correspondences differ for input task. For example, for the task of mashing/-pounding the bottle base corresponds with hammerhead. Similarly for poking, the spout of watering can corresponds to the tip of gardening tool.

6.5.1 Functional Correspondences

We first evaluate the performance of our model on the task of estimating functional correspondences. We create an evaluation benchmark of (training image,test image, task) triplets using the FunKPointdataset. As explained earlier, each image is associated with 5 keypoint annotations. The goal is to identify the location of each keypoint in the test image using the associated train image and task.

Given representations of the train and test images $f_{\text{train}}, f_{\text{test}}$, the corresponding test image location for a training image keypoint p_{train} can be identified as:

$$p_{\text{test}} = \arg\max_{p}(f_{\text{train}}[p_{\text{train}}]^{\intercal}\ f_{\text{test}}[p]) \tag{6.3}$$

We use the PCK metric to evaluate the quality of estimated keypoints. An estimated keypoint in the test image is considered correct if it lies within 23 pixels from the ground truth annotation.

The ImageNet baseline involves using features from a pretrained ResNet-50 `conv4_x` layer (same as the trunk of our model) to compute the correspondences. We also compare to self-supervised semantic correspondence estimation model presented in [122] and features from a task-agnostic affordance estimation method [45]. Finally, we evaluate three variants of our model. Ours with task-embedding refers to our full model. Ours without task-embedding (uniform) refers to a model with $W_t[i, j, k] = 1/M_T$ *i.e.* the gating weights are constant, uniform and not dependent on the task embedding t. Ours without task-embedding (learned) refers to a similar task-independent model, where a single learned gating weight W_t is shared for all tasks. We observe that our proposed model substantially outperforms the ImageNet model and self-supervised learning methods. This further illustrates the difference between learning representations for semantic and functional correspondence. In the ablation of our model, we observe that our proposed model outperforms its task-independent variants by a substantial margin. This emphasizes the need for task-dependent features since functional correspondences are closely tied to the task considered.

We also train a variant of our task-dependent model by initializing from a self-supervised learning method DINO [26]. We observe that while this underperforms the model initialized from ImageNet, it still significantly outperforms all the baseline methods. This indicates that learning using the FunKPoint dataset is crucial.

Finally, we measure the consistency of annotations across humans in the functional correspondences by collecting a second set of human annotations. We observe that the new annotations of correspondence achieve a PCK of 82.5%. Additionally, on a subset of randomly chosen 200 pairs of images, we collected annotations from 4 humans. We observe that the median distance between estimated functional keypoints was 13.07 pixels. These results demonstrate the ambiguity in the functional correspondence task is minimal.

In Figure 4, we present a visualization of the estimated correspondences for five tasks. We observe that our model is able to learn inter-category correspondences. For example, it is able to learn correspondence between bottlehead and pan spouts for pouring. Some interesting correspondences include correspondence between hammerhead and sole of the shoe and correspondence between spout and tip of gardening-tool. While our model was trained to estimate correspondences for keypoints, our model learns to estimate correspondences for all points on objects. In Figure 5, we visualize densely sampled points on objects and their estimated correspondences on test images. While our model is trained on 5 key points in each image, we observe that the model can approximately associate each densely sampled locations on the reference object to the functionally appropriate location on the target object. For example, the rim of the mug in the first image is appropriately associated

Table 2: Correspondences Quantitative Evaluation:

Method	PCK
ImageNet (ResNet50)	22.0
MAST [122] (ResNet18)	8.3
AffordanceNet [45]	15.3
Ours without task-embedding (uniform)	52.8
Ours without task-embedding (learned)	52.5
Ours with task-embedding	58.4
Ours with task-embedding (+DINO Init.)	43.5
Human Annotator	82.5

Table 3: Fewshot Learning Accuracy: We observe that the representation learned for functional correspondence (row 3) demonstrates superior generalization in a few-shot learning setup compared to the baseline ImageNet-based representation (row 1) and an ImageNet representation finetuned to classify the objects in the FunKPoint dataset (row 2).

Method	Accuracy		
	1-shot	2-shot	5-shot
ImageNet	44.68	52.52	54.63
ImageNet FT FunKPoint	45.03	53.91	55.55
Ours	47.46	55.68	56.32

with the spout of the watering can.

6.5.2 Few-shot Generalization

Classification of objects requires understanding its appearance and 3D structure. However, exhaustively modeling appearance and 3D properties from a few samples is challenging and ambiguous in many cases. For example, observing an image of a white conical coffee mug could lead to the belief that all mugs are conical. What we need is a way to use the data from other categories to help learn what makes mug a mug? Since, in the task of functional correspondence, we already label correspondences across multiple categories, our learned model might have better ability to create cross-category generalization. This is the hypothesis we want to test in this experiment.

First, we curate a small dataset of 5 manipulable objects (shovel, water jug, coffee mug, wok and letter opener) with 20 images each. We create train-test splits by including 1, 2 or 5 images for each object in the train set and the rest in the test set. In each of the settings, we generate 3 different random samples for the splits leading to a total of 9 unique splits.

We train a linear classifier to classify the features extracted from our proposed model. Since our model extracts task-dependent features, for each image, we concatenate the features extracted for all 10-tasks and perform spatial average pooling to

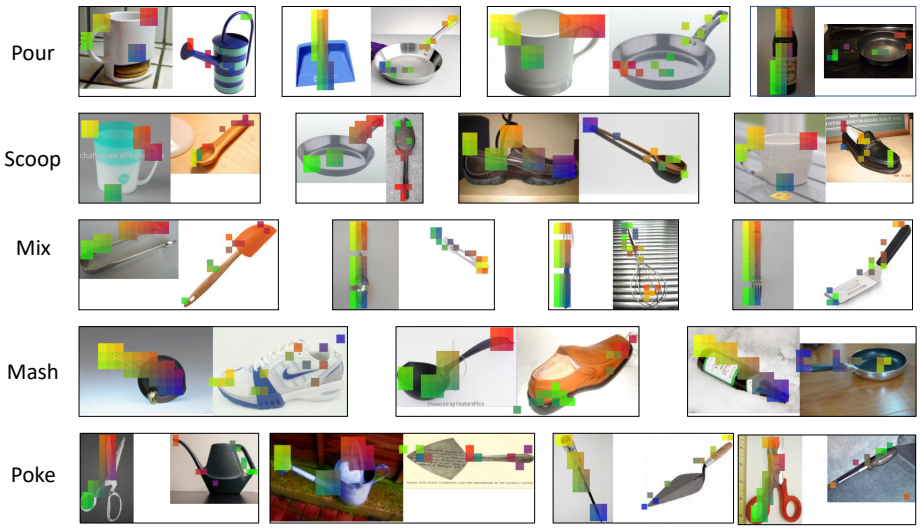


Figure 5: Beyond Keypoint Correspondences: The representation learned by our proposed model can be used effectively to identify dense functional correspondences from a reference image (left in each pair) to target images (right in each pair). The colors indicate matched points. Observe that the identified correspondences (same color points) are consistent in terms of functionality.

reduce the dimensionality. As a baseline, we similarly train a linear classifier on the ResNet-50 `conv4_x` features pretrained on ImageNet. For a fair comparison, we also finetune an ImageNet representation to classify objects in our presented FunKPoint dataset. We present the results in Table 3. The representation learned by our model outperforms the ImageNet based representation on all three settings by substantial margins. We also note that our model outperforms the representation optimized for classifying the objects in the FunKPoint dataset. This indicates that the task of functional correspondence leads to representations that generalize better to novel manipulable objects.

6.5.3 Grasp Prediction

Functional representations are ideally suited for facilitating downstream robotic manipulation tasks. A common challenge addressed in robotic manipulation is the task of grasp prediction. In [182], this is formalized as prediction of grasp success given an image and a hypothesized grasp angle. For this task, we evaluate the efficacy of the features learned by performing functional correspondence. We extract features using our proposed model, concatenate with the hypothesized grasp angle (as a discretized 18-way one-hot vector), the extracted feature is then fed into a 2-layer neural network appended at the end of the modular network to predict the grasp success label. The feature extractor and the classifier are jointly finetuned on the training set of the benchmark using a smaller learning rate of $2e^{-4}$, until the

model converges. As a baseline, we use an ImageNet-based ResNet-50 (similarly truncated at `layer3` as ours). Table 4 contains the numerical results for our model on the Grasp benchmark. Our method outperform the baseline method by 1.7% accuracy, demonstrating the advantage of functional representations.

Table 4: Classification accuracy on Grasp Dataset [182]:

Method	Accuracy
ImageNet	88.17
ImageNet FT FunKPoint	88.64
Ours	**89.85**

6.5.4 ADROIT Manipulation Task

A lot of recent research has focused on learning robotic manipulation of objects through reinforcement learning. In this section, we investigate whether standard reinforcement learning (RL) based methods can take advantage of functional representations. We adopt the method proposed in RRL [212], a simple RL algorithm that uses pre-trained ResNet [96] features which can be easily replaced by our representation. We evaluate this algorithm on the ADROIT manipulation suite [188], which consists of several complex dexterous manipulation tasks. In the Tool Use task environment, we evaluate for the task of hammering a nail.

In Figure 6, we present the success rate of our representation compared to the baseline ImageNet-based based features. We observe that our representation leads to improved sample efficiency and final performance at convergence. We believe these results demonstrate the promise of functional representations for robotics problems. We hope that this will inspire more exhaustive investigations of functional representations and their role in robotics.

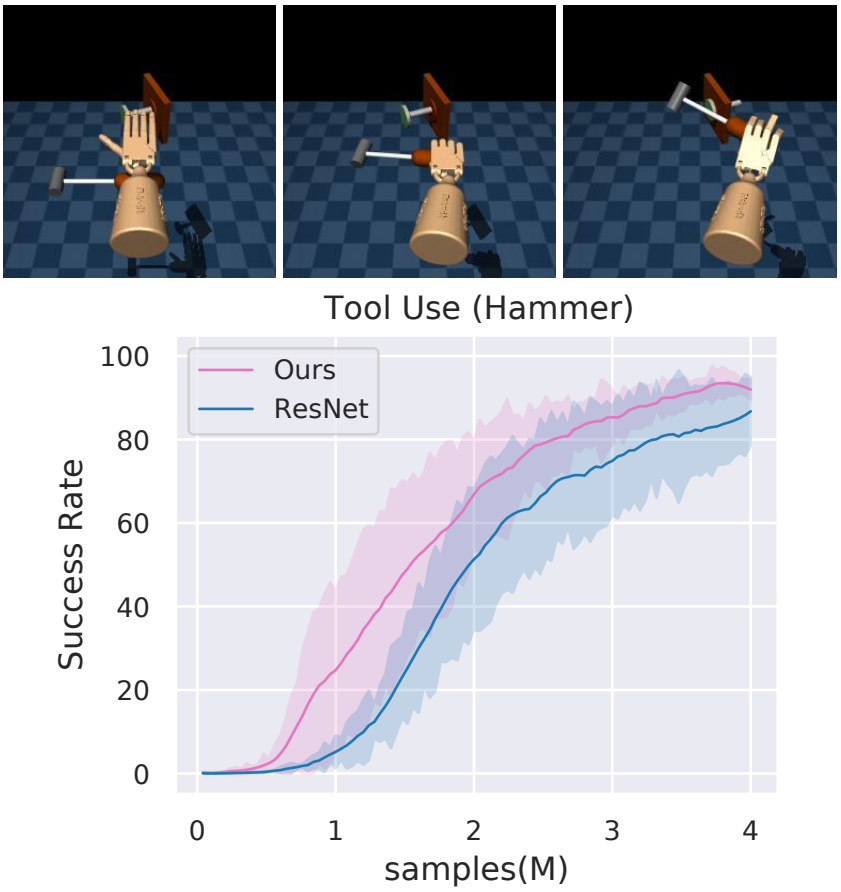


Figure 6: RL-based Manipulation: We evaluate on the hammering a nail Tool Use task (top) introduced in [188]. We observe that our functional representation demonstrates improved sample efficiency and success rate compared to a baseline ImageNet-based representations that does not encode functional aspects of objects.

Appendices

Appendix 6.A Annotation Interface

Figure 6.B.2 shows the annotation interface we used in the Amazon Mechanical Turk system. The image for labelling is shown to the left, together with the specific action we are considering. In the middle, we show 5 examples of labelled images. To the right, we show specific instructions and definitions of each point that is being labelled. Both image examples and point definition are conditioned on the given action. Three extra constraints are put on the labelled points:

1. The worker must add all keypoint annotations and use each label only once
2. The worker must annotates all points inside the given "Image Area".
3. The worker must add annotation for the difficulty.

The labelling process took around 5 days. We then check for errors in the annotations and relabel as described in the main text.

Appendix 6.B Annotation Difficulties

Figure 6.B.1 shows the level of difficulties provided by the annotators. Note annotators could be different for different categories, and the difficulty values may not be consistent across all annotators (different annotators may feel different difficulty for labelling the same image). Here, 0 means *Easy*, 0.5 means *Medium* and 1 means *Hard*. The values shown are computed from an average over all objects in the class. As one could expect, screwdriver is the easiest object category because there are very little ambiguities in defintions of each point and the shape variations are small. Two most difficult object classes are baskets and dustpan. The potential reason could be their large shape variance. For baskets, there are woven basket, shopping basket, basket with lids, without lids, and many others. Similarly, dustpan could have different handle length and orientation. Some so called lobby dustpans could have another structure that functions as a lid. As comparison, the easier object classes such as bottle, cup and tablefork have relatively little shape variance.

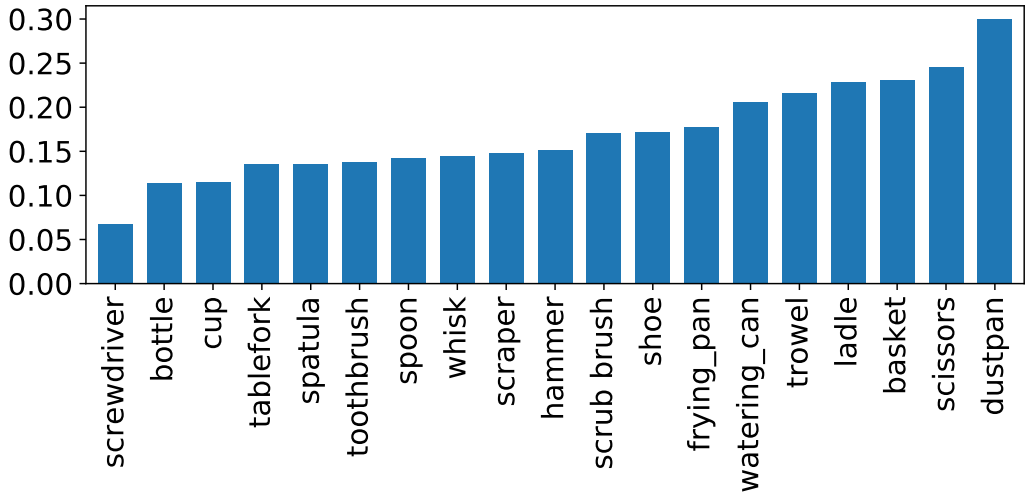


Figure 6.B.1: Level of difficulties for each object category.

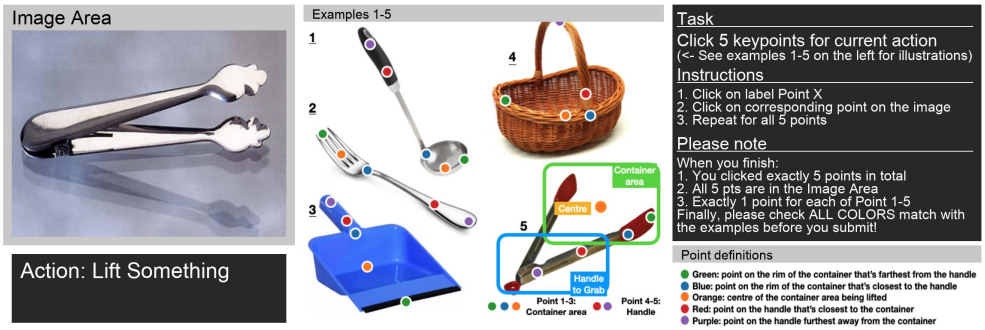


Figure 6.B.2: Annotation interface: The workers are asked the label the image to the left with instructions and examples given.

Appendix 6.C Implementation Details

In this section, we provide additional hyperparameter details to facilitate easy reproduction of our results.

As explained in Sec 4.1 of the main text, our proposed model is inspired from the task-driven modular networks proposed in [185]. We design the modular network as 4 layers with 6,6,6,1 modules in each layer respectively. Here we present the hyperparameters of the convolution layers:
1st layer: 128 filters, kernel size=7, stride=1, padding=3
2nd layer: 128 filters, kernel size=3, stride=1, padding=1
3rd layer: 128 filters, kernel size=3, stride=1, padding=1
4th layer: 128 filters, kernel size=1, stride=1, padding=0

We train the modules using SGD with learning rate 0.01, momentum 0.9 and weight decay 0.00001 with a batch size of 256. The gating network takes as input a 100-dimensional embedding based on the task under consideration. The 10 embeddings

for the 10 tasks in the FunKPoint dataset are randomly initialize and learned during the optimization process. The gating network consists of a 2-layer fully-connected neural network with a hidden embedding size of 100.

Appendix 6.D Dataset Statistics

As explained in the main text, each object could be associated with multiple actions. This leads to varying number of keypoint annotations based on object category. In Figure 6.D.1, we present these statistics:

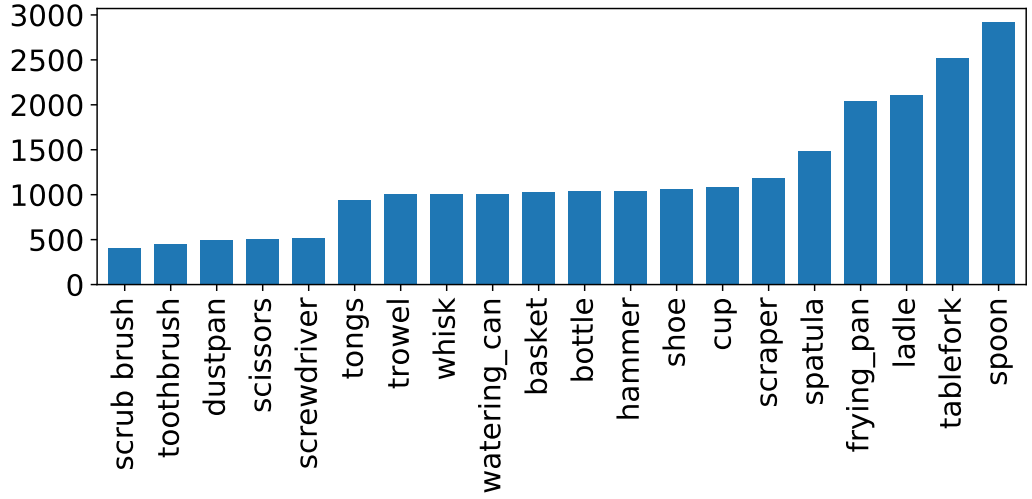


Figure 6.D.1: Number of keypoint annotations for each object category.

Chapter 7

Conclusion and Future Research

In this thesis, we investigate approaches for minimizing the role of human-supervision in *representation* and *recognition* problems. The exploration of approaches for learning visual perception models without supervision possesses the potential for more scalable, robust and generalizable systems that can enable a wider range of applications. We believe that learning without human supervision is also crucial for bridging the gap between current computer vision systems and the extraordinary capabilities of human visual perception.

In Part I of this thesis, we investigated and exposed the limitations of current state-of-the-art self-supervised representation learning methods. We demonstrate that experimenting with curated datasets has led to methods that heavily leverage dataset biases (Chapter 2, Chapter 3). In order to allow scaling up of these methods, we propose approaches to alleviate the dependence on clean and curated data (Chapter 2, Chapter 3). In future research, in order to disallow similar unintentional biases, we believe it is crucial to construct realistic training and evaluation benchmarks that reflect the data observed in-the-wild.

In Part II, we presented a novel modular neural network architecture (Chapter 4) that facilitates construction of classifiers for compositions of seen concepts without additional supervision. The idea of leveraging compositionality can be extended beyond object-attribute annotations to free-form descriptions of images which are more readily available. We also present the novel "understanding via associations" paradigm for describing visual signals by finding other associated samples (Chapter 5). We apply this to dense video understanding by proposing a weakly-supervised cycle-consistency loss. We show that using this method, we can retrieve similar videos,identify dense spatio-temporal correspondences and densely describe a video via associations.

Finally, in Part III, we propose and investigate the novel "Functional Correspondence Problem". In this work, we explore beyond the extensively addressed computer vision problems of identifying semantic similarity. We propose an approach to identify correspondences between objects of different semantic categories

based on the intended functionality. We believe that further research on identifying functional correspondences between objects can enable robotic applications that are not limited to interacting with known object categories.

Bibliography

[1] A. Achille, T. Eccles, L. Matthey, C. Burgess, N. Watters, A. Lerchner, and I. Higgins. Life-long disentangled representation learning with cross-domain latent homologies. *Advances in Neural Information Processing Systems*, 31, 2018. 29

[2] P. Agrawal, J. Carreira, and J. Malik. Learning to see by moving. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015. 68

[3] K. Ahmed and L. Torresani. Maskconnect: Connectivity learning by gradient descent. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2018. 48

[4] J.-B. Alayrac, J. Sivic, I. Laptev, and S. Lacoste-Julien. Joint discovery of object states and manipulation actions. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017. 68

[5] R. Aljundi, M. Lin, B. Goujaud, and Y. Bengio. Gradient based sample selection for online continual learning. *Advances in Neural Information Processing Systems*, 32:11816–11825, 2019. 29

[6] J. Andreas, M. Rohrbach, T. Darrell, and D. Klein. Deep compositional question answering with neural module networks. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 48

[7] M. Andrychowicz, F. Wolski, A. Ray, J. Schneider, R. Fong, P. Welinder, B. McGrew, J. Tobin, P. Abbeel, and W. Zaremba. Hindsight experience replay. *arXiv preprint arXiv:1707.01495*, 2017. 32

[8] R. Arandjelovic and A. Zisserman. Look, listen and learn. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 609–617, 2017. 27

[9] M. Aubry, D. Maturana, A. A. Efros, B. C. Russell, and J. Sivic. Seeing 3d chairs: exemplar part-based 2d-3d alignment using a large dataset of cad models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3762–3769, 2014. 29

[10] C. Bailer, K. Varanasi, and D. Stricker. Cnn-based patch matching for optical flow with thresholded hinge embedding loss. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017. 83

[11] A. Bakhtin, Y. Deng, S. Gross, M. Ott, M. Ranzato, and A. Szlam. Residual energy-based models for text. *Journal of Machine Learning Research*, 22(40):1–41, 2021. 84

[12] D. Bau, J.-Y. Zhu, H. Strobelt, B. Zhou, J. B. Tenenbaum, W. T. Freeman, and A. Torralba. Visualizing and understanding generative adversarial networks. *arXiv preprint arXiv:1901.09887*, 2019. 10

[13] H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. In *European conference on computer vision*, pages 404–417. Springer, 2006. 83

[14] A. Bearman, O. Russakovsky, V. Ferrari, and L. Fei-Fei. What's the point: Semantic segmentation with point supervision. In *European conference on computer vision*, pages 549–565. Springer, 2016. 2

[15] Y. Bengio and J.-S. Senecal. Quick training of probabilistic neural nets by importance sampling. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2003. 51

[16] A. C. Berg, T. L. Berg, and J. Malik. Shape matching and object recognition using low distortion correspondences. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, volume 1, pages 26–33. IEEE, 2005. 83

[17] P. H. BK and B. G. Schunck. Determining optical flow. *Artificial intelligence*, 17(1–3), 1981. 68

[18] T. Brox, C. Bregler, and J. Malik. Large displacement optical flow. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2009. 83

[19] P. Buzzega, M. Boschini, A. Porrello, D. Abati, and S. Calderara. Dark experience for general continual learning: a strong, simple baseline. In *Advances in Neural Information Processing Systems*, 2020. 29

[20] Y. L. Cacheux, H. L. Borgne, and M. Crucianu. From classical to generalized zero-shot learning: a simple adaptation process. In *Proceedings of the 25th International Conference on MultiMedia Modeling*, 2019. 47

[21] M. Caron, P. Bojanowski, A. Joulin, and M. Douze. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 132–149, 2018. 27

[22] M. Caron, P. Bojanowski, J. Mairal, and A. Joulin. Unsupervised pre-training of image features on non-curated data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2959–2968, 2019. 25, 28

[23] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *arXiv preprint arXiv:2006.09882*, 2020. 3

[24] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In *Thirty-fourth Conference on Neural Information Processing Systems (NeurIPS)*, 2020. 25, 27, 28

[25] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021. 27, 28

[26] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin. Emerging properties in self-supervised vision transformers. *arXiv preprint arXiv:2104.14294*, 2021. 90

[27] J. Carreira and A. Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2, 37, 67

[28] W.-L. Chao, S. Changpinyo, B. Gong, and F. Sha. An empirical study and analysis of generalized zero-shot learning for object recognition in the wild. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2016. 47, 51, 52, 53, 54

[29] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020. 3, 7, 10, 11, 27, 32

[30] X. Chen, H. Fan, R. Girshick, and K. He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020. 8, 10, 11, 21, 22, 24

[31] X. Chen and K. He. Exploring simple siamese representation learning. *arXiv preprint arXiv:2011.10566*, 2020. 3, 27, 28, 30

[32] X. Chen, A. Shrivastava, and A. Gupta. Neil: Extracting visual knowledge from web data. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1409–1416, 2013. 28

[33] X. Chen*, S. Xie*, and K. He. An empirical study of training self-supervised vision transformers. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021. 27

[34] S. T. D. Xie and S. Zhu. Inferring 'dark matter' and 'dark energy' from videos. In *ICCV*, 2013. 83

[35] D. Damen, H. Doughty, G. Maria Farinella, S. Fidler, A. Furnari, E. Kazakos, D. Moltisanti, J. Munro, T. Perrett, W. Price, et al. Scaling egocentric vision: The epic-kitchens dataset. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018. 67

[36] V. R. de Sa. Learning classification with unlabeled data. In *Advances in Neural Information Processing Systems*, pages 112–119. Citeseer, 1994. 27

[37] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR*, 2009. 1, 2, 37, 38

[38] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009. 73

[39] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009. 85, 88

[40] E. Denton and R. Fergus. Stochastic video generation with a learned prior. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2018. 67

[41] K. Desai and J. Johnson. Virtex: Learning visual representations from textual annotations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11162–11173, 2021. 27

[42] A. Deshpande, J. Rock, and D. Forsyth. Learning large-scale automatic image colorization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 567–575, 2015. 27

[43] C. Devin, A. Gupta, T. Darrell, P. Abbeel, and S. Levine. Learning modular neural network policies for multi-task and multi-robot transfer. In *2017 IEEE international conference on robotics and automation (ICRA)*, pages 2169–2176. IEEE, 2017. 84

[44] T.-T. Do, A. Nguyen, and I. Reid. Affordancenet: An end-to-end deep learning approach for object affordance detection. In *International Conference on Robotics and Automation (ICRA)*, 2018. 83

[45] T.-T. Do, A. Nguyen, and I. Reid. Affordancenet: An end-to-end deep learning approach for object affordance detection. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 5882–5889. IEEE, 2018. 90, 91

[46] C. Doersch. Tutorial on variational autoencoders. *arXiv preprint arXiv:1606.05908*, 2016. 9

[47] C. Doersch, A. Gupta, and A. A. Efros. Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015. 3, 68

[48] C. Doersch, A. Gupta, and A. A. Efros. Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1422–1430, 2015. 7, 9, 11

[49] C. Doersch, A. Gupta, and A. A. Efros. Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1422–1430, 2015. 27

[50] C. Doersch and A. Zisserman. Multi-task self-supervised visual learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2051–2060, 2017. 9

[51] A. Dosovitskiy, P. Fischer, J. T. Springenberg, M. Riedmiller, and T. Brox. Discriminative unsupervised feature learning with exemplar convolutional neural networks. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 38(9):1734–1747, 2015. 27

[52] A. Dosovitskiy, J. T. Springenberg, M. Riedmiller, and T. Brox. Discriminative unsupervised feature learning with convolutional neural networks. In *Advances in neural information processing systems*, pages 766–774, 2014. 7, 9

[53] Y. Du and I. Mordatch. Implicit generation and modeling with energy based models. 2019. 84

[54] D. Dwibedi, Y. Aytar, J. Tompson, P. Sermanet, and A. Zisserman. Temporal cycle-consistency learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 66, 67, 68, 73, 78

[55] D. Eigen, I. Sutskever, and M. Ranzato. Learning factored representations in a deep mixture of experts. In *Workshop at the International Conference on Learning Representations*, 2014. 48

[56] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html. 14

[57] B. G. Fabian Caba Heilbron, Victor Escorcia and J. C. Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 961–970, 2015. 2

[58] C. Feichtenhofer, H. Fan, J. Malik, and K. He. Slowfast networks for video recognition. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019. 67

[59] B. Fernando, H. Bilen, E. Gavves, and S. Gould. Self-supervised video representation learning with odd-one-out networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3636–3645, 2017. 27

[60] C. Fernando, D. Banarse, C. Blundell, Y. Zwols, D. Ha, A. A. Rusu, A. Pritzel, and D. Wierstra. Pathnet: Evolution channels gradient descent in super neural networks. *arXiv:1701.08734*, 2017. 48

[61] P. Florence, L. Manuelli, and R. Tedrake. Dense object nets: Learning dense visual object descriptors by and for robotic manipulation. In *CoRL*, 2018. 84

[62] D. Fouhey, X. Wang, and A. Gupta. In defense of the direct perception of affordances. In *arXiv:1505.01085*, 2015. 83

[63] D. F. Fouhey, V. Delaitre, A. Gupta, A. A. Efros, I. Laptev, and J. Sivic. People watching: Human actions as a cue for single-view geometry. In *ECCV*, 2012. 83

[64] D. F. Fouhey, W. Kuo, A. A. Efros, and J. Malik. From lifestyle vlogs to everyday interactions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 67

[65] V. Garro, A. Fusiello, and S. Savarese. Label transfer exploiting three-dimensional structure for semantic segmentation. In *Proceedings of the 6th International Conference on Computer Vision/Computer Graphics Collaboration Techniques and Applications*, 2013. 68

[66] J.-M. Geusebroek, G. J. Burghouts, and A. W. Smeulders. The amsterdam library of object images. *International Journal of Computer Vision*, 61(1):103–112, 2005. 13, 21

[67] J. Gibson. *The ecological approach to visual perception*. Boston: Houghton Mifflin, 1979. 83

[68] S. Gidaris, P. Singh, and N. Komodakis. Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*, 2018. 9

[69] S. Gidaris, P. Singh, and N. Komodakis. Unsupervised representation learning by predicting image rotations. In *International Conference on Learning Representations*, 2018. 27, 28

[70] R. Girdhar, J. Carreira, C. Doersch, and A. Zisserman. Video action transformer network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 67

[71] R. Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015. 2, 17, 24

[72] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2014. 2

[73] C. Godard, O. Mac Aodha, and G. J. Brostow. Unsupervised monocular depth estimation with left-right consistency. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 68

[74] C. Godard, O. Mac Aodha, M. Firman, and G. J. Brostow. Digging into self-supervised monocular depth estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3828–3838, 2019. 83

[75] L. Gomez, Y. Patel, M. Rusiñol, D. Karatzas, and C. Jawahar. Self-supervised learning of visual features through embedding images into text topic spaces. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4230–4239, 2017. 27

[76] I. Goodfellow, H. Lee, Q. V. Le, A. Saxe, and A. Y. Ng. Measuring invariances in deep networks. In *Advances in neural information processing systems*, pages 646–654, 2009. iv, 10, 12, 13, 20, 21

[77] P. Goyal, M. Caron, B. Lefaudeux, M. Xu, P. Wang, V. Pai, M. Singh, V. Liptchinsky, I. Misra, A. Joulin, et al. Self-supervised pretraining of visual features in the wild. *arXiv preprint arXiv:2103.01988*, 2021. 25, 27, 28, 34

[78] P. Goyal, Q. Duval, I. Seessel, M. Caron, M. Singh, I. Misra, L. Sagun, A. Joulin, and P. Bojanowski. Vision models are more robust and fair when pretrained on uncurated images without supervision. *arXiv preprint arXiv:2202.08360*, 2022. 25, 28

[79] P. Goyal, D. Mahajan, A. Gupta, and I. Misra. Scaling and benchmarking self-supervised visual representation learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6391–6400, 2019. 10, 11

[80] P. Goyal, D. Mahajan, A. Gupta, and I. Misra. Scaling and benchmarking self-supervised visual representation learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6391–6400, 2019. 25, 28, 34

[81] R. Goyal, S. E. Kahou, V. Michalski, J. Materzynska, S. Westphal, H. Kim, V. Haenel, I. Fruend, P. Yianilos, M. Mueller-Freitag, et al. The "something something" video database for learning and evaluating visual common sense. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, volume 1, 2017. 67

[82] H. Grabner, J. Gall, and L. van Gool. What makes a chair a chair? In *CVPR*, 2011. 83

[83] W. Grathwohl, K.-C. Wang, J.-H. Jacobsen, D. Duvenaud, M. Norouzi, and K. Swersky. Your classifier is secretly an energy based model and you should treat it like one. *arXiv preprint arXiv:1912.03263*, 2019. 84

[84] R. L. Gregory. Knowledge in perception and illusion. *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences*, 352(1358):1121–1127, 1997. 1

[85] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. Richemond, E. Buchatskaya, C. Doersch, B. Pires, Z. Guo, M. Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. In *Advances in Neural Information Processing Systems*, 2020. 27

[86] C. Gu, C. Sun, D. A. Ross, C. Vondrick, C. Pantofaru, Y. Li, S. Vijayanarasimhan, G. Toderici, S. Ricco, R. Sukthankar, et al. Ava: A video dataset of spatio-temporally localized atomic visual actions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 67

[87] A. Gupta, S. Satkin, A. Efros, and M. Hebert. From 3D scene geometry to human workspace. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2011. 83

[88] R. Hadsell, S. Chopra, and Y. LeCun. Dimensionality reduction by learning an invariant mapping. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, volume 2, pages 1735–1742. IEEE, 2006. 27

[89] B. Ham, M. Cho, C. Schmid, and J. Ponce. Proposal flow. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 68

[90] C. G. Harris, M. Stephens, et al. A combined corner and edge detector. In *Alvey vision conference*, volume 15, pages 10–5244. Citeseer, 1988. 83

[91] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick. Momentum contrast for unsupervised visual representation learning. *arXiv preprint arXiv:1911.05722*, 2019. 3, 7, 8, 10, 11, 27, 28

[92] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask R-CNN. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017. 2, 73, 75, 76, 77

[93] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 32

[94] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 46, 52

[95] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 73

[96] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 84, 88, 93

[97] O. J. Hénaff, A. Srinivas, J. De Fauw, A. Razavi, C. Doersch, S. Eslami, and A. v. d. Oord. Data-efficient image recognition with contrastive predictive coding. *arXiv preprint arXiv:1905.09272*, 2019. 3, 7, 10

[98] B. K. Horn and B. G. Schunck. Determining optical flow. *Artificial intelligence*, 17(1-3):185–203, 1981. 83

[99] Y.-C. Hsu, Y.-C. Liu, A. Ramasamy, and Z. Kira. Re-evaluating continual learning scenarios: A categorization and case for strong baselines. *arXiv preprint arXiv:1810.12488*, 2018. 32

[100] L. Huang, X. Zhao, and K. Huang. Got-10k: A large high-diversity benchmark for generic object tracking in the wild. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019. 13

[101] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017. 83

[102] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015. 88

[103] P. Isola, J. J. Lim, and E. H. Adelson. Discovering states and transformations in image collections. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1383–1391, 2015. 51

[104] A. Jabri, A. Owens, and A. A. Efros. Space-time correspondence as a contrastive random walk. *arXiv preprint arXiv:2006.14613*, 2020. 83

[105] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, G. E. Hinton, et al. Adaptive mixtures of local experts. *Neural computation*, 3(1):79–87, 1991. 48

[106] D. Jayaraman and K. Grauman. Learning image representations tied to ego-motion. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015. 68

[107] X. Jia, R. Ranftl, and V. Koltun. Accurate optical flow via direct cost volume processing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017. 83

[108] S. P. Johnson. Development of visual perception. *Wiley Interdisciplinary Reviews: Cognitive Science*, 2(5):515–528, 2011. 3

[109] M. I. Jordan and R. A. Jacobs. Hierarchical mixtures of experts and the em algorithm. In *International Joint Conference on Neural Networks*, 1993. 48

[110] Z. Kalal, K. Mikolajczyk, and J. Matas. Forward-backward error: Automatic detection of tracking failures. In *International Conference on Pattern Recognition (ICPR)*, 2010. 68

[111] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017. 2, 67, 73, 75, 77

[112] A. Kendall, H. Martirosyan, S. Dasgupta, P. Henry, R. Kennedy, A. Bachrach, and A. Bry. End-to-end learning of geometry and context for deep stereo regression. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 66–75, 2017. 83

[113] J. Kim, C. Liu, F. Sha, and K. Grauman. Deformable spatial pyramid matching for fast dense correspondences. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013. 68

[114] S. Kim, D. Min, B. Ham, S. Jeon, S. Lin, and K. Sohn. Fcss: Fully convolutional self-similarity for dense semantic correspondence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017. 81, 83, 86

[115] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 52

[116] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 73

[117] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 9

[118] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017. 28, 38

[119] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2012. 1, 66, 84

[120] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012. 32

[121] V. Kumar Verma, G. Arora, A. Mishra, and P. Rai. Generalized zero-shot learning via synthesized examples. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4281–4289, 2018. 47

[122] Z. Lai, E. Lu, and W. Xie. MAST: A memory-augmented self-supervised tracker. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 90, 91

[123] Z. Lai and W. Xie. Self-supervised learning for video correspondence flow. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2019. 83

[124] C. H. Lampert, H. Nickisch, and S. Harmeling. Attribute-based classification for zero-shot visual object categorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 3(36):453–465, 2014. 47

[125] G. Larsson, M. Maire, and G. Shakhnarovich. Learning representations for automatic colorization. In *Proceedings of the European Conference on Computer Vision*, pages 577–593. Springer, 2016. 27

[126] Q. V. Le. Building high-level features using large scale unsupervised learning. In *Proceeding of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 8595–8598. IEEE, 2013. 27

[127] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 46

[128] Y. LeCun, S. Chopra, R. Hadsell, M. Ranzato, and F. Huang. A tutorial on energy-based learning. *Predicting structured data*, 1(0), 2006. 84

[129] Y. LeCun, S. Chopra, R. Hadsell, M. Ranzato, and F. Huang. A tutorial on energy-based learning. In G. Bakir, T. Hofman, B. Schölkopf, A. Smola, and B. Taskar, editors, *Predicting Structured Data*. MIT Press, 2006. 48, 49

[130] A. X. Lee, R. Zhang, F. Ebert, P. Abbeel, C. Finn, and S. Levine. Stochastic adversarial video prediction. 2018. 67

[131] H. Lee, A. Battle, R. Raina, and A. Y. Ng. Efficient sparse coding algorithms. In *Advances in Neural Information Processing Systems*, pages 801–808, 2007. 27

[132] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In *Proceedings of the 26th annual international conference on machine learning*, pages 609–616, 2009. 9

[133] X. Li, Y. Zhou, T. Wu, R. Socher, and C. Xiong. Learn to grow: A continual structure learning framework for overcoming catastrophic forgetting. In *Proceedings of the International Conference on Machine Learning*, pages 3925–3934. PMLR, 2019. 29

[134] Z. Li and D. Hoiem. Learning without forgetting. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 40(12):2935–2947, 2017. 28, 38

[135] L.-J. Lin. *Reinforcement learning for robots using neural networks*. Carnegie Mellon University, 1992. 32

[136] 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 *Proceedings of the European Conference on Computer Vision (ECCV)*, 2014. 2, 3, 15, 73

[137] C. Liu, J. Yuen, and A. Torralba. Nonparametric scene parsing: Label transfer via dense scene alignment. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009. 68

[138] C. Liu, J. Yuen, and A. Torralba. Sift flow: Dense correspondence across scenes and its applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 33(5), 2010. 67, 68

[139] P. Liu, M. Lyu, I. King, and J. Xu. Selflow: Self-supervised learning of optical flow. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4571–4580, 2019. 83

[140] S. Liu, M. Long, J. Wang, and M. I. Jordan. Generalized zero-shot learning with deep calibration network. In *Advances in Neural Information Processing Systems (NIPS)*, 2018. 47

[141] Z. Liu, P. Luo, X. Wang, and X. Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3730–3738, 2015. 29

[142] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015. 18

[143] D. G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the seventh IEEE international conference on computer vision*, volume 2, pages 1150–1157. Ieee, 1999. 83

[144] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision (IJCV)*, 60(2), 2004. 66, 68, 83

[145] B. D. Lucas, T. Kanade, et al. An iterative image registration technique with an application to stereo vision. 1981. 68

[146] L. v. d. Maaten and G. Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008. 59

[147] A. Makhzani, J. Shlens, N. Jaitly, I. Goodfellow, and B. Frey. Adversarial autoencoders. *arXiv preprint arXiv:1511.05644*, 2015. 9

[148] A. Mallya, D. Davis, and S. Lazebnik. Piggyback: Adapting a single network to multiple tasks by learning to mask weights. In *Proceedings of the European Conference on Computer Vision*, pages 67–82, 2018. 29

[149] L. Manuelli, W. Gao, P. Florence, and R. Tedrake. kpam: Keypoint affordances for category level manipulation. In *ISRR*, 2019. 84

[150] J. Masci, U. Meier, D. Cireşan, and J. Schmidhuber. Stacked convolutional auto-encoders for hierarchical feature extraction. In *Proceedings of the International Conference on Artificial Neural Networks*, pages 52–59. Springer, 2011. 27

[151] E. Mémin and P. Pérez. Dense estimation and object-based segmentation of the optical flow with robust techniques. *IEEE Transactions on Image Processing*, 7(5), 1998. 68

[152] E. Meyerson and R. Miikkulainen. Beyond shared hierarchies: Deep multitask learning through soft layer ordering. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2018. 48

[153] A. Miech, J.-B. Alayrac, L. Smaira, I. Laptev, J. Sivic, and A. Zisserman. End-to-end learning of visual representations from uncurated instructional videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9879–9889, 2020. 27

[154] A. Miech, D. Zhukov, J.-B. Alayrac, M. Tapaswi, I. Laptev, and J. Sivic. Howto100m: Learning a text-video embedding by watching hundred million narrated video clips. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2630–2640, 2019. 27

[155] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781, 2013. 50

[156] G. A. Miller. *WordNet: An electronic lexical database*. MIT press, 1998. 38

[157] I. Misra, A. Gupta, and M. Hebert. From red wine to red tomato: Composition with context. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 46, 47, 48, 49, 51, 52, 53, 60

[158] I. Misra and L. van der Maaten. Self-supervised learning of pretext-invariant representations. *arXiv preprint arXiv:1912.01991*, 2019. 7, 8, 10, 14, 27

[159] I. Misra, C. L. Zitnick, and M. Hebert. Shuffle and learn: unsupervised learning using temporal order verification. In *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer, 2016. 27, 67, 68

[160] T. Mitchell, W. Cohen, E. Hruschka, P. Talukdar, B. Yang, J. Betteridge, A. Carlson, B. Dalvi, M. Gardner, B. Kisiel, et al. Never-ending learning. *Communications of the ACM*, 61(5):103–115, 2018. 28

[161] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015. 32

[162] H. Mobahi, R. Collobert, and J. Weston. Deep learning from temporal coherence in video. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 737–744. ACM, 2009. 27

[163] P. Morgado and N. Vasconcelos. Nettailor: Tuning the architecture, not just the weights. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3044–3054, 2019. 29

[164] P. Morgado, N. Vasconcelos, and I. Misra. Audio-visual instance discrimination with cross-modal agreement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12475–12486, 2021. 27

[165] M. Muller, A. Bibi, S. Giancola, S. Alsubaihi, and B. Ghanem. Trackingnet: A large-scale dataset and benchmark for object tracking in the wild. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 300–317, 2018. 16, 23

[166] A. Murali, W. Liu, K. Marino, S. Chernova, and A. Gupta. Same object, different grasps: Data and semantic knowledge for task-oriented grasping. In *Conference on Robot Learning*, 2020. 85

[167] T. Nagarajan, C. Feichtenhofer, and K. Grauman. Grounded human-object interaction hotspots from video. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019. 67

[168] T. Nagarajan and K. Grauman. Attributes as operators: Factorizing unseen attribute-object compositions. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2018. 46, 47, 48, 49, 51, 52, 53, 54, 55, 60

[169] M. Noroozi and P. Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *Proceedings of the European Conference on Computer Vision*, pages 69–84. Springer, 2016. 27, 28

[170] D. Novotny, D. Larlus, and A. Vedaldi. Anchornet: A weakly supervised network to learn geometry-sensitive features for semantic matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017. 81, 83, 86

[171] B. A. Olshausen. Sparse coding of time-varying natural images. In *Proc. of the Int. Conf. on Independent Component Analysis and Blind Source Separation*, volume 2. Citeseer, 2000. 27

[172] B. A. Olshausen and D. J. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583):607–609, 1996. 27

[173] A. v. d. Oord, Y. Li, and O. Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 3, 7, 10, 27

[174] A. v. d. Oord, Y. Li, and O. Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 87

[175] A. Owens and A. A. Efros. Audio-visual scene analysis with self-supervised multisensory features. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 631–648, 2018. 27

[176] M. Palatucci, D. Pomerleau, G. E. Hinton, and T. M. Mitchell. Zero-shot learning with semantic output codes. In *Advances in neural information processing systems*, pages 1410–1418, 2009. 47

[177] D. Pathak, R. Girshick, P. Dollár, T. Darrell, and B. Hariharan. Learning features by watching objects move. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 3, 10, 27, 68

[178] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2536–2544, 2016. 9, 27

[179] J. Pennington, R. Socher, and C. Manning. Glove: Global vectors for word representation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014. 52

[180] E. Perez, F. Strub, H. de Vries, V. Dumoulin, and A. Courville. Film: Visual reasoning with a general conditioning layer. In *AAAI*, 2018. 48

[181] L. Pinto and A. Gupta. Supersizing self-supervision: Learning to grasp from 50K tries and 700 robot hours. In *Proceedings of the International Conference On Robotics and Automation (ICRA)*, 2016. 2

[182] L. Pinto and A. Gupta. Learning to push by grasping: Using multiple tasks for effective learning. In *2017 IEEE international conference on robotics and automation (ICRA)*, pages 2161–2168. IEEE, 2017. 92, 93

[183] S. Purushwalkam and A. Gupta. Pose from action: Unsupervised learning of pose features based on motion. *arXiv preprint arXiv:1609.05420*, 2016. 3

[184] S. Purushwalkam and A. Gupta. Demystifying contrastive self-supervised learning: Invariances, augmentations and dataset biases. *arXiv preprint arXiv:2007.13916*, 2020. 3

[185] S. Purushwalkam, M. Nickel, A. Gupta, and M. Ranzato. Task-driven modular networks for zero-shot compositional learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3593–3602, 2019. 84, 87, 88, 96

[186] R. Qian, T. Meng, B. Gong, M.-H. Yang, H. Wang, S. Belongie, and Y. Cui. Spatiotemporal contrastive video representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6964–6974, 2021. 27

[187] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. *arXiv preprint arXiv:2103.00020*, 2021. 27

[188] A. Rajeswaran, V. Kumar, A. Gupta, G. Vezzani, J. Schulman, E. Todorov, and S. Levine. Learning Complex Dexterous Manipulation with Deep Reinforcement Learning and Demonstrations. In *Proceedings of Robotics: Science and Systems (RSS)*, 2018. 89, 93, 94

[189] J. Ramapuram, M. Gregorova, and A. Kalousis. Lifelong generative modeling. *Neurocomputing*, 404:381–400, 2020. 29

[190] A. Rannen, R. Aljundi, M. B. Blaschko, and T. Tuytelaars. Encoder based lifelong learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1320–1328, 2017. 28

[191] M. Ranzato, F. J. Huang, Y.-L. Boureau, and Y. LeCun. Unsupervised learning of invariant feature hierarchies with applications to object recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007. 27

[192] D. Rao, F. Visin, A. Rusu, R. Pascanu, Y. W. Teh, and R. Hadsell. Continual unsupervised representation learning. *Advances in Neural Information Processing Systems*, 32:7647–7657, 2019. 29

[193] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2001–2010, 2017. 29

[194] J. Revaud, P. Weinzaepfel, Z. Harchaoui, and C. Schmid. Epicflow: Edge-preserving interpolation of correspondences for optical flow. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2015. 83

[195] E. Rivlin, S. Dickinson, and A. Rosenfeld. Recognition by functional parts. In *CVIU*, 1995. 83

[196] I. Rocco, R. Arandjelovic, and J. Sivic. Convolutional neural network architecture for geometric matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017. 83, 86

[197] I. Rocco, R. Arandjelovic, and J. Sivic. End-to-end weakly-supervised semantic alignment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018. 83, 86

[198] I. Rocco, R. Arandjelović, and J. Sivic. End-to-end weakly-supervised semantic alignment. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 66

[199] D. Rolnick, A. Ahuja, J. Schwarz, T. P. Lillicrap, and G. Wayne. Experience replay for continual learning. In *Advances in Neural Information Processing Systems*, pages 350–360, 2019. 29, 32

[200] C. Rosenbaum, T. Klinger, and M. Riemer. Routing networks: Adaptive selection of non-linear functions for multi-task learning. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2018. 48

[201] M. Rubinstein, A. Joulin, J. Kopf, and C. Liu. Unsupervised joint object discovery and segmentation in internet images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013. 68

[202] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *arXiv preprint arXiv:1409.0575*, 2014. 52

[203] A. A. Rusu, N. C. Rabinowitz, G. Desjardins, H. Soyer, J. Kirkpatrick, K. Kavukcuoglu, R. Pascanu, and R. Hadsell. Progressive neural networks. *arXiv preprint arXiv:1606.04671*, 2016. 29

[204] R. Salakhutdinov and G. Hinton. Deep boltzmann machines. In *Artificial Intelligence and Statistics*, pages 448–455. PMLR, 2009. 27

[205] R. Salakhutdinov, A. Torralba, and J. Tenenbaum. Learning to share visual appearance for multiclass object detection. In *CVPR 2011*, pages 1481–1488. IEEE, 2011. 46

[206] T. Schaul, J. Quan, I. Antonoglou, and D. Silver. Prioritized experience replay. In *Proceedings of the International Conference on Learning Representations*, 2016. 32

[207] J. Schmidhuber. Evolutionary principles in self-referential learning. *On learning how to learn: The meta-meta-... hook.*) *Diploma thesis, Institut f. Informatik, Tech. Univ. Munich*, 1:2, 1987. 48

[208] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017. 10

[209] R. R. Selvaraju, A. Das, R. Vedantam, M. Cogswell, D. Parikh, and D. Batra. Grad-cam: Why did you say that? *arXiv preprint arXiv:1611.07450*, 2016. 10

[210] P. Sermanet, C. Lynch, Y. Chebotar, J. Hsu, E. Jang, S. Schaal, S. Levine, and G. Brain. Time-contrastive networks: Self-supervised learning from video. In *Proceedings of the IEEE International Conference on Robotics and Automation* (*ICRA*). Pouring dataset licensed under (CC BY 4.0)., 2018. 67, 68, 72, 78

[211] I. K. Sethi and R. Jain. Finding trajectories of feature points in a monocular image sequence. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (*TPAMI*), (1), 1987. 68

[212] R. Shah and V. Kumar. Rrl: Resnet as representation for reinforcement learning. In *ICML*, 2021. 93

[213] H. Shin, J. K. Lee, J. Kim, and J. Kim. Continual learning with deep generative replay. In *Advances in Neural Information Processing Systems*, pages 2994–3003, 2017. 29

[214] G. A. Sigurdsson, G. Varol, X. Wang, A. Farhadi, I. Laptev, and A. Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding. In *Proceedings of the European Conference on Computer Vision* (*ECCV*), 2016. 2, 67

[215] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in Neural Information Processing Systems* (*NIPS*), 2014. 67

[216] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 84

[217] K. K. Singh, K. Fatahalian, and A. A. Efros. Krishnacam: Using a longitudinal, single-person, egocentric dataset for scene understanding tasks. In *2016 IEEE Winter Conference on Applications of Computer Vision* (*WACV*), pages 1–9. IEEE, 2016. 37

[218] S. Singh, A. Gupta, and A. A. Efros. Unsupervised discovery of mid-level discriminative patches. In *Proceedings of the European Conference on Computer Vision* (*ECCV*). Springer, 2012. 66

[219] S. Smith and J. Brady. Asset-2: Real-time motion segmentation and shape tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(8):814–820, 1995. 83

[220] L. Stark and K. Bowyer. Achieving generalized object recognition through reasoning about association of function to structure. In *PAMI*, 1991. 83

[221] D. Sun, S. Roth, and M. J. Black. Secrets of optical flow estimation and their principles. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010. 68

[222] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 68

[223] Y. Tang, R. Salakhutdinov, and G. Hinton. Robust boltzmann machines for recognition and denoising. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2264–2271. IEEE, 2012. 9

[224] B. Thomee, D. A. Shamma, G. Friedland, B. Elizalde, K. Ni, D. Poland, D. Borth, and L.-J. Li. Yfcc100m: The new data in multimedia research. *Communications of the ACM*, 59(2):64–73, 2016. 14

[225] S. Thrun. A lifelong learning perspective for mobile robot control. In *Intelligent robots and systems*, pages 201–214. Elsevier, 1995. 28

[226] Y. Tian, D. Krishnan, and P. Isola. Contrastive multiview coding. *arXiv preprint arXiv:1906.05849*, 2019. 10

[227] Y. Tian, C. Sun, B. Poole, D. Krishnan, C. Schmid, and P. Isola. What makes for good views for contrastive learning. *arXiv preprint arXiv:2005.10243*, 2020. 10

[228] M. K. Titsias, J. Schwarz, A. G. d. G. Matthews, R. Pascanu, and Y. W. Teh. Functional regularisation for continual learning with gaussian processes. *arXiv preprint arXiv:1901.11356*, 2019. 28

[229] D. Tran, L. D. Bourdev, R. Fergus, L. Torresani, and M. Paluri. C3D: generic features for video analysis. *CoRR, abs/1412.0767*, 2(7), 2014. 67

[230] N. Ufer and B. Ommer. Deep semantic feature matching. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6914–6923, 2017. 83, 86

[231] J. R. Uijlings, K. E. Van De Sande, T. Gevers, and A. W. Smeulders. Selective search for object recognition. *International journal of computer vision*, 104(2):154–171, 2013. 17, 23

[232] L. Valkov, D. Chaudhari, A. Srivastava, C. Sutton, and S. Chaudhuri. Houdini: Lifelong learning as program synthesis. In *Advances in Neural Information Processing Systems (NIPS)*, 2018. 48

[233] G. Van Horn, O. Mac Aodha, Y. Song, Y. Cui, C. Sun, A. Shepard, H. Adam, P. Perona, and S. Belongie. The inaturalist species classification and detection dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8769–8778, 2018. 37

[234] G. Van Horn and P. Perona. The devil is in the tails: Fine-grained classification in the wild. *arXiv preprint arXiv:1709.01450*, 2017. 2, 46

[235] G. Varol, I. Laptev, and C. Schmid. Long-term temporal convolutions for action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 40(6), 2017. 67

[236] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103, 2008. 9, 27

[237] O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra, et al. Matching networks for one shot learning. In *Advances in neural information processing systems*, pages 3630–3638, 2016. 48

[238] H. Wang, M. M. Ullah, A. Klaser, I. Laptev, and C. Schmid. Evaluation of local spatio-temporal features for action recognition. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2009. 66

[239] J. Wang, J. Jiao, and Y.-H. Liu. Self-supervised video representation learning by pace prediction. In *Proceedings of the European Conference on Computer Vision*, pages 504–521. Springer, 2020. 27

[240] J. Wang, H. Xu, J. Xu, S. Lu, and X. Wang. Synthesizing long-term 3d human motion and interaction in 3d scenes. In *CVPR*, 2021. 83

[241] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer, 2016. 67

[242] T. Wang and P. Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. *arXiv preprint arXiv:2005.10242*, 2020. 10

[243] X. Wang, A. Jabri, and A. Efros. Learning correspondence from the cycle-consistency of time. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019. 81, 83

[244] X. Wang, A. Farhadi, and A. Gupta. Actions ˜ transformations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 67

[245] X. Wang, R. Girdhar, and A. Gupta. Binge watching: Scaling affordance learning from sitcoms. In *arXiv:1804.03080*, 2018. 83

[246] X. Wang, R. Girshick, A. Gupta, and K. He. Non-local neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 67

[247] X. Wang and A. Gupta. Unsupervised learning of visual representations using videos. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2794–2802, 2015. 3, 7, 9, 10, 11, 27

[248] X. Wang and A. Gupta. Unsupervised learning of visual representations using videos. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015. 68, 69

[249] X. Wang, K. He, and A. Gupta. Transitive invariance for self-supervised visual representation learning. In *Proceedings of the IEEE international conference on computer vision*, pages 1329–1338, 2017. 10

[250] X. Wang, A. Jabri, and A. A. Efros. Learning correspondence from the cycle-consistency of time. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 10, 67, 68, 69, 70, 73, 75, 76, 77

[251] X. Wang, F. Yu, R. Wang, T. Darrell, and J. E. Gonzalez. Tafe-net: Task-aware feature embeddings for efficient learning and inference. *arXiv:1806.01531*, 2018. 48, 49

[252] X. Wang, F. Yu, R. Wang, T. Darrell, and J. E. Gonzalez. Tafe-net: Task-aware feature embeddings for low shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1831–1840, 2019. 84, 87

[253] Y.-X. Wang, D. Ramanan, and M. Hebert. Learning to model the tail. In *Advances in Neural Information Processing Systems*, pages 7029–7039, 2017. 46

[254] P. Winston, T. Binford, B. Katz, and M. Lowry. Learning physical description from functional definitions, examples and precedents. In *MIT Press*, 1984. 83

[255] C.-Y. Wu, C. Feichtenhofer, H. Fan, K. He, P. Krahenbuhl, and R. Girshick. Long-term feature banks for detailed video understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 67

[256] Z. Wu, Y. Xiong, S. X. Yu, and D. Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3733–3742, 2018. 3, 9, 10, 11

[257] Z. Wu, Y. Xiong, S. X. Yu, and D. Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3733–3742, 2018. 27

[258] Y. Xian, T. Lorenz, B. Schiele, and Z. Akata. Feature generating networks for zero-shot learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5542–5551, 2018. 47, 52, 53

[259] Y. Xiang, R. Mottaghi, and S. Savarese. Beyond pascal: A benchmark for 3d object detection in the wild. In *IEEE winter conference on applications of computer vision*, pages 75–82. IEEE, 2014. 13

[260] S. Xie, C. Sun, J. Huang, Z. Tu, and K. Murphy. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018. 67

[261] R. Yang, H. Xu, Y. Wu, and X. Wang. Multi-task reinforcement learning with soft modularization. *arXiv preprint arXiv:2003.13661*, 2020. 84

[262] Y. Yang and D. Ramanan. Articulated human detection with flexible mixtures of parts. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2012. 78

[263] F. Ye and A. G. Bors. Learning latent representations across multiple data domains using lifelong vaegan. In *Proceedings of the European Conference on Computer Vision*, pages 777–795. Springer, 2020. 29

[264] S. Yeung, O. Russakovsky, N. Jin, M. Andriluka, G. Mori, and L. Fei-Fei. Every moment counts: Dense detailed labeling of actions in complex videos. *International Journal of Computer Vision (IJCV)*, 126(2-4), 2018. 67

[265] J. Yoon, E. Yang, J. Lee, and S. J. Hwang. Lifelong learning with dynamically expandable networks. *arXiv preprint arXiv:1708.01547*, 2017. 29

[266] A. Yu and K. Grauman. Fine-grained visual comparisons with local learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 192–199, 2014. 51

[267] A. Yu and K. Grauman. Semantic jitter: Dense supervision for visual comparisons via synthetic images. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5570–5579, 2017. 51

[268] J. Zbontar, L. Jing, I. Misra, Y. LeCun, and S. Deny. Barlow twins: Self-supervised learning via redundancy reduction. In *Proceedings of the International Conference on Machine Learning*, 2021. 27

[269] F. Zenke, B. Poole, and S. Ganguli. Continual learning through synaptic intelligence. In *Proceedings of the International Conference on Machine Learning*, pages 3987–3995. PMLR, 2017. 28

[270] H. Zhang, J. Xiao, and L. Quan. Supervised label transfer for semantic segmentation of street scenes. In *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer, 2010. 68

[271] R. Zhang, P. Isola, and A. A. Efros. Colorful image colorization. In *European conference on computer vision*, pages 649–666. Springer, 2016. 3, 7, 9, 27, 28

[272] R. Zhang, P. Isola, and A. A. Efros. Split-brain autoencoders: Unsupervised learning by cross-channel prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 68

[273] W. Zhang, M. Zhu, and K. G. Derpanis. From actemes to action: A strongly-supervised representation for detailed action understanding. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2013. 72, 75, 77

[274] Y. Zhao and S. Zhu. Scene parsing by integrating function, geometry and appearance models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2013. 83

[275] B. Zhou, D. Bau, A. Oliva, and A. Torralba. Comparing the interpretability of deep networks via network dissection. In *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, pages 243–252. Springer, 2019. 10

[276] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 633–641, 2017. 18

[277] T. Zhou, Y. Jae Lee, S. X. Yu, and A. A. Efros. Flowweb: Joint image set alignment by weaving consistent, pixel-wise correspondences. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 68

[278] T. Zhou, P. Krahenbuhl, M. Aubry, Q. Huang, and A. A. Efros. Learning dense correspondence via 3d-guided cycle consistency. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 68

[279] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017. 68