UC Riverside

UC Riverside Electronic Theses and Dissertations

Title

Learning with Limited Supervision for Static and Dynamic Tasks

Permalink

https://escholarship.org/uc/item/0qt2c0qn

Author

Paul, Sujoy

Publication Date

2020

Peer reviewed|Thesis/dissertation

eScholarship.org Powered by the California Digital Library
University of California

UNIVERSITY OF CALIFORNIA
RIVERSIDE

Learning with Limited Supervision for Static and Dynamic Tasks

A Dissertation submitted in partial satisfaction
of the requirements for the degree of

Doctor of Philosophy

in

Electrical Engineering

by

Sujoy Paul

September 2020

Dissertation Committee:

Dr. Amit K. Roy-Chowdhury, Chairperson
Dr. Ertem Tuncel
Dr. Samet Oymak

Copyright by
Sujoy Paul
2020

The Dissertation of Sujoy Paul is approved:

____

____

____

____

Committee Chairperson

University of California, Riverside

Acknowledgments

The satisfaction that accompanies my completion of PhD would be incomplete without the mention of the people who constantly supported and encouraged me. First and foremost, I would like to thank my advisor Dr. Amit K. Roy-Chowdhury for his constant motivation and help during the course of this dissertation. He gave me the freedom to work on problems that interest me the most. Through his constant guidance, I learned various aspects of doing good research. From problem formulation to its presentation, I learned a lot from him over the course of five years, which will be of even more value when I go out for my next ventures.

I would also like to express my gratitude to my dissertation committee members, Dr. Ertem Tuncel and Dr. Samet Oymak for giving me thoughtful feedback and constructive comments in improving the quality of this dissertation. While taking a couple of courses with Dr. Tuncel, I found him as an excellent teacher and learned a lot from him. Discussing ideas on research problems with Dr. Oymak has been an invaluable experience for me. I also want to express my gratitude to Prof Anastasios Mourikis, from whom I learned a lot while taking courses and discussing research problems. I look up to his way of teaching and presentation. I have also learned a lot from my internships and especially got to spend time with some of my mentors - Dr. Muthian Sivanthu and Dr. Ramachandran Ramjee at Microsoft Research India, Dr. Jeroen van Baar at Mitsubishi Electric Research Lab, Dr. Yi-Hsuan Tsai, Dr. Samuel Schulter and Dr. Manmohan Chandraker at NEC Lab. I learned a lot while working with them and I thank them for their constant support and encouragement. My special thanks to Jeroen van Baar, who introduced me to reinforcement learning, and since then my interests in the field have been ever-growing.

I spend some good time with my labmates at UC Riverside - Mahmudul Hasan, Jawad Bappy, Niluthpol Chowdhury Mithun, Rameswar Panda, Dripta Raychaudhuri, Miraj Ahmed, and Shasha Li. Without them, my life at UC Riverside would have been much more challenging. I am fortunate enough to get a friend like Sourya Roy during my PhD, who has been a constant support through the ups and downs. Over the years, I have learned a lot from him. He was my roommate for the better part of my PhD and never left an opportunity to cheer me up. The time we spent together exploring Riverside, the USA, and the UK will invaluable to me. I would also like to thank Bodhisattwa Majumder, who has been my friend since my undergrads. Spending a summer with him while doing an internship, the long hour discussions, exploring northern California, will be in memories forever.

Finally, I would like to express my heartfelt regards to my parents for their support and faith that always motivates me to work harder. I lost my father at the beginning of the second year of my PhD. My mother stood by my side and it is because of her support that I am able to finish my PhD. My father was always proud of me pursuing a PhD and I hope he is looking down onto me while I take the last mile.

I thank the National Science Foundation (IIS-1316934, CPS-1544969, IIS-1724341), Office of Naval Research (ONR N00014-15-C-5113), and Mitsubishi Electric Research Lab (MERL) for their grants to Dr. Amit K. Roy-Chowdhury, which partially supported my research. I also thank Victor Hill of UCR CS and Tom Gregory of UCR ECE for setting up the computing infrastructure used in most of the works presented in this thesis. Acknowledgment of previously published or accepted materials: The text of this dissertation, in part or in full, is a reprint of the material as appeared in three previously published papers that I first authored. The co-author Dr. Amit K. Roy-Chowdhury, listed in all publications, directed and supervised the research which forms the basis for this dissertation. The papers are as follows.

1. Sujoy Paul, Jawadul H Bappy, and Amit K Roy-Chowdhury. Non-uniform subset selection for active learning in structured data. CVPR, 2017

2. Sujoy Paul, Sourya Roy, and Amit K Roy-Chowdhury. W-talc: Weakly-supervised temporal activity localization and classification. ECCV, 2018

3. Sujoy Paul, Yi-Hsuan Tsai, Samuel Schulter, Amit K Roy-Chowdhury, and Manmohan Chandraker. Domain adaptive semantic segmentation using weak labels. ECCV, 2020

4. Sujoy Paul, Jeroen Vanbaar, and Amit Roy-Chowdhury. Learning from trajectories via subgoal discovery. NeurIPS, 2019

To my late father, my mother and my friends.

ABSTRACT OF THE DISSERTATION

Learning with Limited Supervision for Static and Dynamic Tasks

by

Sujoy Paul

Doctor of Philosophy, Graduate Program in Electrical Engineering
University of California, Riverside, September 2020
Dr. Amit K. Roy-Chowdhury, Chairperson

The recent successes in computer vision have been mostly around using a huge corpus of intricately labeled data for training recognition models. But, in real-world cases, acquiring such large datasets will require a lot of manual annotation, which may be strenuous, out of budget, or even prone to errors. Whereas, a lot of real data that are generated daily can be acquired at low to no annotation cost. Such data can be unlabeled or contain tag/meta-data information, termed as weak annotation. Our goal is to develop methods that can learn recognition models from such data involving limited manual supervision. In this thesis, we explore two dimensions of learning with limited supervision - first, reducing the *number* of manually labeled data required to learn recognition models, and second, reducing the *level* of supervision from strong to weak which can be mined from the web, easily queried from an oracle, or imposed as rule-based labels derived from domain knowledge.

In the first dimension of learning with limited supervision, we show that context information, often present in natural data, can be used to reduce the number of annotations required. We take an information-theoretic approach considering the relationship in data points, to select them for labeling, unlike works in literature which only use the uncertainty of individual samples. In the next dimension of learning with limited supervision, i.e., reducing the level of supervision, we use weak labels instead of dense strong labels, for learning dense prediction tasks. We develop frameworks to learn using weak labels for action detection in videos and domain adaptation of semantic segmentation models on images. In action detection, unlike using frame-wise annotations as in the literature, we use only video-level annotations, which is much easier to obtain from the annotator and can also be mined from the web. In domain adaptation of semantic segmentation models, we use weak image-level labels in two forms - pseudo weak labels, which are estimated using the source segmentation model, incurring no annotation cost, or oracle weak labels, which are obtained from the human annotator and incurring a very low cost. In spite of using such weak labels, our methods perform close to frameworks using strong supervision.

Continuing in the direction of learning from weak labels, we explore sequential decision-making problems. We learn robotics tasks with a small set of expert human demonstrations. Traditional imitation learning methods can only be as good as the expert, with a lot of human demos. We devise a strategy that divides a complex task into subgoals and solves them sequentially with reinforcement learning. We learn the subgoal partitions just from the human demos without any partition labels from the human annotator, by imposing only a temporal ordering based weak constraint among the subgoals, often arising in most real-world tasks. Our method is able to solve tasks with a low number of demos which other methods in the literature are not able to solve.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Frameworks in computer vision have been mostly around using a huge corpus of intricately labeled data to learn recognition models. For e.g., in dense predictions tasks such as semantic segmentation [29], object detection [61], or action detection [238], state-of-the-art methods utilize dense manual labeling for training, obtaining which is a tedious job, and prone to errors. Moreover, dense manual labeling may not scale well in real-world applications requiring an enormous amount of data and further in continual learning scenarios where drift in concept occurs and constant labeling may be necessary. Similar trends can also be seen in learning to solve dynamical decision making tasks via reinforcement learning [126], where either detailed specifications of the reward function are necessary or a huge number of human demonstrations [180] are required when using imitation learning. Thus, to utilize the ever-growing corpus of data for better performance of real-world computer vision and robotics applications, it is necessary to develop learning mechanisms that can learn from limited supervision.

In recent years, researchers have started focusing to develop algorithms that can learn from limited supervision. There can be potentially two dimensions of limited supervision. In the first case, we have access to only a small *number* of labeled data points, and probably a lot of unlabeled data points. Unsupervised feature learning [30, 141], semi-supervised learning [134], active learning [170] fall under this category and mostly considered for classification tasks that involve single prediction per data point. The second case includes tasks involving dense predictions. In this case, limited supervision would mean having access to partial information about the labels, often termed as *weak* labels. For e.g., weak labels can be the categories present in an entire image compared to having labels for every pixel, the latter termed as *strong* labels. Note that this form of limited supervision reduces the level of supervision, whereas the first case reduces the number of samples. Other forms of information such as domain or world knowledge [151, 159], physics-based constraints [44], side information [73], can be utilized to reduce the amount of supervision required to learn recognition models.

In this thesis, we primarily explore core problems in computer vision involving static prediction tasks and also one problem involving dynamic decision making. We study these problems in the light of learning from limited supervision and develop algorithms to solve these tasks in such scenarios. Fig. 1.1 presents a pictorial overview of the thesis. We start in Chapter 2 discussing our framework for learning with a limited number of data points for classification tasks in an active learning setup, which corresponds to the first dimension of learning from limited supervision as discussed in the previous paragraph. Active Learning [170] is a method of choosing the most informative samples to label from an

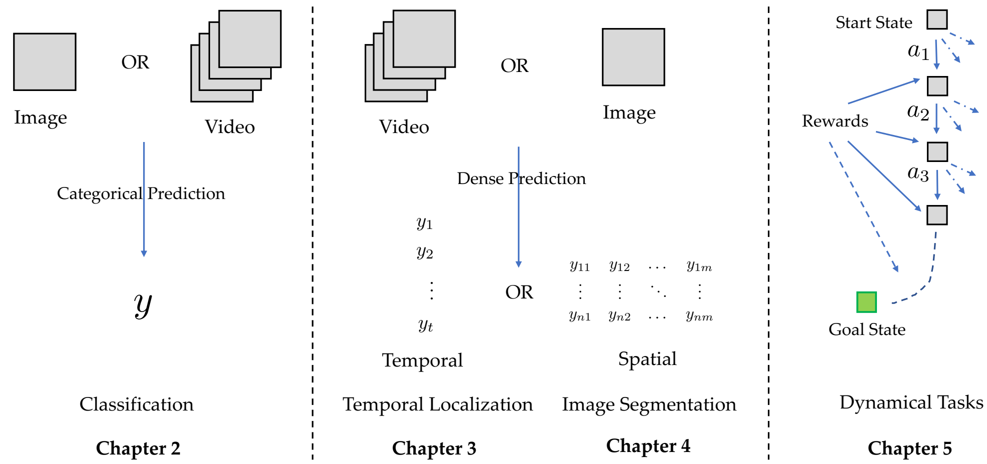


Figure 1.1: This figure presents the organization of the thesis. In the second chapter, we look into reducing the amount of supervision for classification tasks, where for a given image or video the task is to predict categorical labels, y. In the third and fourth chapters, we look into reducing the level of supervision from strong to weak for temporal action detection and image segmentation respectively. These are dense predictions tasks that need per frame predictions, y_t, for action detection in videos or per-pixel spatial predictions, y_{ij} in image segmentation. Finally, in the fifth chapter, we look into the sequential decision-making problems under the light of reduced level manual supervision. In these tasks, a starts from a certain state, is required to go to a goal state by taking sequential actions, a_t.

unlabeled set. Most existing active learning methods in literature formulate a utility score for each unlabeled sample, based on which only on a small subset of them are chosen for manual labeling. Information density [111], classifier uncertainty [112], expected error rate [41, 111], expected change in gradient [171], expected model output change [87] are some popular utility functions used in the literature. But, most of these methods do consider the inter-relationships that may occur in data points belonging to the same or different recognition tasks. For e.g., in a scene, containing multiple objects, as scene-objects and objects-objects co-occur in natural images, certainty about some objects in the image can help in better understanding of the rest and thus help in choosing to label only those instances, which results in a better holistic understanding of the scene, rather than dealing with them individually. We develop our framework using probabilistic graphical models [97], where the nodes represent the individual unlabeled data points and the edges represent the inter-relationships between them. We formulate an information-theoretic cost function to select a small number of nodes from the graph and query the human annotator to obtain their labels. We show that our method is general enough to be applied to a variety of problems - joint scene-object recognition where we utilize the co-occurrence between scenes and objects, action recognition where we exploit the spatio-temporal relationships in streaming videos, and document classification where we use the citations as relationship information. We show that utilizing the relationship information which is often available in most natural data, helps to reduce the number of labeled samples even further compared to state-of-the-art methods in the literature.

In Chapter 3 we move towards the second dimension of learning with limited supervision discussed above, i.e., the scenario where we have reduced level of supervision from strong dense labeling to weak labels. In this direction, we first look into the problem of action detection in videos, where given a long video, we need to temporally localize human action or event categories of interest. State-of-the-art methods on action detection [238, 227] use strong annotations to learn the detection model, i.e., start-end time of every action category that occurs in the video. Obtaining such precise frame-wise annotations requires enormous manual labor and often discrepancy arises between annotators with the start and end of events in the video. On the other hand, it is much easier for a person to provide a few categorical labels which encapsulate the content of a video. Moreover, in the absence of a human annotator, we can also mine such training data from the web as videos on the internet are often accompanied by tags that provide semantic discrimination. Such video-level labels are generally termed as *weak* labels, as they are weaker forms of annotation than obtaining the labels of every time frame in the video. We utilize such weak video-level labels to learn action detection models where during test time, the model temporally localizes the categories of interest. The motivation behind the prospective of using weak labels is driven by finding the similarities across all videos which have a certain category in common. We pose the problem as a Multiple Instance Learning (MIL) and impose certain constraints for better feature learning. Analogous to MIL, each video can be considered as a bag of frames and we need to predict the category of every frame in the bag on the test videos, given only bag-level labels during training. We develop two loss functions to learn the model parameters. First, the video classification loss based on the k-max-mean multiple instance learning and co-activity similarity loss via an attention mechanism. The co-activity similarity loss considers pairs of videos and enforces similar activity temporal regions to have similar features compared to dissimilar activity regions, which are identified using an attention mechanism. The model significantly performs better when using the co-activity loss than when using only the video classification loss. This work shows that we can obtain significant detection performance even when using only weak labels rather than strong dense labels used in the literature, thus opening the horizon for using the enormous corpus of videos accompanied by tags on the web.

Continuing in the direction of using weak labels, we look into the problem of semantic segmentation of images in Chapter 4. A segmentation model learned on one dataset (source) may not generalize well to images from a different distribution (target).

Thus, the model needs to be adapted to the target images. Pixel-wise labeling of real-world images takes a huge amount of time (90 min/image [39]). Due to such high annotation costs, the model needs to be adapted from source to target with none to minimal annotation cost. Current methods [202, 47] in literature have been mostly unsupervised, i.e., requiring no annotation on the target side. However, there exists a considerable performance gap between these methods and a fully supervised method with pixel-wise labels. To bridge the gap, we utilize weak labels of the target images in two different paradigms of learning. We use image-level weak labels, i.e., we only have information about the presence or absence of the categories appearing in the image. There are two ways by which we can obtain such weak labels. We can estimate them using the model trained on the source side, which would be pseudo-weak labels. However, as this does not involve a human, thus incurring no annotation cost, using such pseudo-weak labels falls under the category of unsupervised domain adaptation. On the other hand, we can ask a human annotator to obtain the weak labels, which would be true labels, and fall under the category of weakly-supervised domain adaptation and incurring a very low annotation cost (30-45 sec/image). Our experimental results with different combinations of source and target datasets show that we can considerably reduce the gap in performance with full target supervision, incurring none to very limited annotation cost.

Finally, in Chapter 5, we look into sequential decision-making tasks, in contrast to static tasks discussed until now. Reinforcement Learning (RL) aims to take sequential actions on an environment, to maximize a certain reward function, designed for a task. RL generally requires intricately designed dense rewards, as methods with sparse rewards require a lot of time and costly interaction with the environment to learn. Imitation Learning (IL) using human demos can be used to learn the policies faster [6], but due to the dynamic nature of the tasks, even a small error rate compounds quadratically with time and results in improper performance. To overcome this problem, we take the path of IL followed by RL to mitigate the errors. However, instead of using sparse rewards in RL, we propose the idea of using subgoals, often characterized in human behavior, for more dense rewards, but learned just from a few human demos. We learn to break down the long complex task into subgoals and make the agent solve them sequentially. However, to learn the subgoal partitions, we do not use any subgoal labels from annotator but impose a weak temporal order constraint on the discovered subgoals, i.e., the first subgoal should occur before the second, the second before the third, and so on. This natural ordering in subgoals, which often arises in most real-world tasks, can be used as a reward function in RL. Results show that the subgoals discovered play an important role in solving several sparse reward tasks sample-efficiently, which other methods in the literature are not able to solve.

Organization of the Thesis. Fig. 1.1 presents a pictorial overview of the thesis. The rest of the thesis is organized as follows. In Chapter 2, we present our framework on active learning in the presence of context to reduce the number of data points needed to learn classification tasks. In Chapter 3 and 4, we move towards dense prediction tasks - action detection and domain adaptation of semantic segmentation respectively. We develop algorithms that can learn from weak labels specifically for these problems. Then in Chapter 5, we move towards learning sequential decision-making tasks in a sample-efficient manner, using the concept of subgoals of an otherwise long complex task. Finally, we conclude the thesis in Chapter 6 with some interesting future directions of work in learning with limited supervision for computer vision and robotics tasks.

Chapter 2

Active Learning with Context

2.1 Introduction

In the recent years, due to advances in technology, huge amount of visual and text data is generated daily, which are mostly unlabeled for the purpose of learning machine learning models. Also, machine learning algorithms are becoming more commonplace in human life. A large proportion of these algorithms are based on supervised learning which requires a large quantity of data to be labeled. Moreover, these models need to be updated over time as new data becomes available in order to dynamically adapt to the different semantic concepts which may drift with time. Manually labeling this continuous flow of data is not only a tedious task for humans but also prone to incorrect labeling. Active Learning [171] can be a solution to this problem to reduce the amount of manual labeling, without compromising recognition performance.

The ability of active learning to reduce manual labeling effort is due to the fact that not all training samples are valuable for building a recognition model [104]. Most active learning approaches formulate a utility score for each unlabeled sample, based on which they are chosen for manual labeling. Uncertainty [112, 139], information density [111], expected change in gradient [171], expected error rate [41, 111], expected model output change [87] and their combinations are some popular techniques for designing the utility score. But, most of these techniques fail to consider the inter-relationships that may occur in data points belonging to the same or different recognition tasks.

Several works have shown that in many applications such as activity recognition [229, 220], object recognition [58, 37], text classification [168, 172], etc, the relationships between data points can be exploited to get better recognition performance. These relationships may also be exploited in active learning to significantly reduce the effort of manual labeling. Although there have been some works that consider relationships between data points in active learning [13, 119, 68, 77], they do not consider the flow of beliefs between samples to have a joint understanding of the samples, which may be helpful for choosing the most informative ones. Moreover, most of them are problem-specific algorithms and deal with active learning of a single recognition task. A general approach for active learning that considers the inter-relationships between data points, and which can be used across a variety of application domains, is lacking. Joint learning of tasks such as scene-object [230, 215] or activity-object [82, 98] classification may be required to be learned actively, to reduce the manual labeling effort. In such scenarios, it is challenging to choose the informative samples for manual labeling as they may belong to different recognition tasks.

In this chapter, we present our work [140] on the generalized active learning framework, which utilize contextual relationships between data points to reduce the manual la-

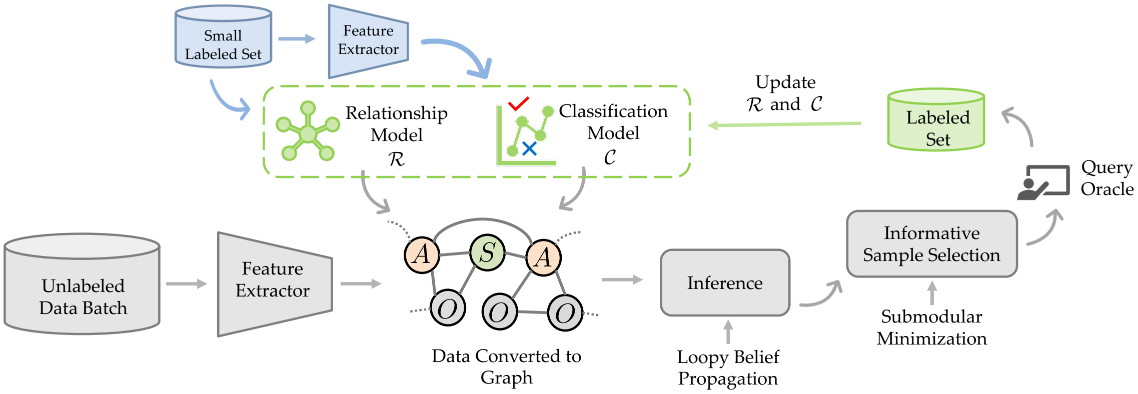


Figure 2.1: This figure presents the flow of the proposed framework. 1. A small set of labeled data is used to obtain the initial relationship ($\mathcal{R}$) and classification model ($\mathcal{C}$). 2. As a new unlabeled batch of data becomes available sequentially over time, we first extract features from the raw data. Then the current $\mathcal{C}$ and $\mathcal{R}$ models are used to construct a graph from the data to represent the relationships between them. Then inference on the graph is used to obtain the node and edge probabilities, which are used to choose the informative samples for manual labeling. The newly labeled instances are then used to update the models $\mathcal{C}$ and $\mathcal{R}$.

beling effort. Given an unlabeled set, our algorithm automatically determines the optimal number of informative samples to be labeled, by exploiting the structure of the data, i.e., the relationships between the samples. The relationship information can not only help to update the beliefs of the classifier for each data point but also plays an important role in selecting a small subset of informative samples, which when labeled can help the other unlabeled samples to have a better understanding of their labels. This framework can be applied for both single, as well as multiple, recognition tasks learned jointly.

Framework Overview. The flow of the proposed method is pictorially presented in Fig.2.1. The proposed method starts with a small set of labeled data and uses it to build the classification ($\mathcal{C}$) and relationship ($\mathcal{R}$) models. $\mathcal{R}$ represents the underlying relationship between the data points via categorical co-occurrence probabilities. Note that the classification models may contain multiple classifiers for multiple recognition tasks. After learning the initial models, given a new batch of unlabeled samples, the goal is to select a subset of informative samples for manual labeling which can be used to update the current classification and relationship models.

As a new batch of data becomes available, they are separated into different sets based on the recognition task to which they belong and their features are extracted. Using the current classifiers, a probability mass function over the possible categories is obtained for each unlabeled sample. It is used along with $\mathcal{R}$ to construct a graph whose nodes represent the samples. A message-passing algorithm is used to infer on the graph to obtain the beliefs of each node and the edges of the graphs. An information-theoretic objective function is derived, which utilizes the beliefs to select the informative nodes for manual labeling. The submodular nature of this optimization function allows us to achieve this in a computationally efficient manner. The newly labeled nodes are used to update the models $\mathcal{C}$ and $\mathcal{R}$. It may be noted that the number of samples selected per batch is *non-uniform*, dependent on the information content of each batch.

Main Contributions. The main contributions are the following.

- We propose a novel generalized active learning framework that exploits the relationships in data to reduce the manual labeling effort. It can be used for both single as well as multiple inter-related recognition tasks jointly.
- Our framework chooses a non-uniform number of samples for manual labeling from each batch of data, which is helpful as the amount of information contained in a batch of data varies and it may not be useful to select the same number of samples from each batch.
- Unlike other batch mode subset selection algorithms that exploit relationships in data

points, the optimization problem in our framework can be proved to be submodular minimization which makes it easy to obtain optimal solutions in polynomial time.

2.2 Related Works

Active Learning. An overview of the approaches which form the core of most active learning (AL) algorithms may be found at [170]. Most AL algorithms involve the uncertainty of the classifier for choosing the informative samples, best vs. second-best [110], entropy [111], classifier margin [212] being commonly used measures for classifier uncertainty. Along with classifier uncertainty, diversification in the chosen samples is also used via k-means [110] or sparse representative subset selection [53]. The scalability issue in terms of the number of categories was addressed in [86] by asking binary questions to the human. They selected samples from the unlabeled set based on expected misclassification risk and extracted a probabilistically similar image from the labeled set to ask whether they match. Another important concept used in AL is expected model change [21, 211, 87].

Active Learning with Relationship Information. Most of the above-mentioned works do not consider the relationships between the data points which may be exploited to reduce the amount of manual labeling. In [14], an AL algorithm was proposed which involves uncertainty, committee-based ensembles, and community-based clustering of networked data. A network-based utility score for each sample was proposed in [103] involving neighborhood information of the networked data. In [174], maximum uncertainty as well as the maximum impact on other unlabeled instances was used, where the link information enhances the feature-based similarity measure used to capture the impact of a sample. In [112], a hierarchical model for AL was proposed for scene classification where they also query the objects whenever there is a mismatch between the scene label provided by the classifier and human. An AL algorithm for scene and object classification is presented in [8]. The relationship in the feature space was exploited in [119] for AL. The concept of typicality in information theory is exploited in [9] to choose the optimal subset of samples.

In [23], an algorithm for batch mode AL was proposed which uses entropy and Kullback Leibler divergence to select informative and diverse samples. However, these algorithms do not incorporate the propagation of beliefs among samples. An AL algorithm is presented in [68] for activity recognition. They proposed an objective function based on intuition and provided a greedy solution to optimize it. Our algorithm on the other hand is not only mathematically validated, but also experimentally supported on different applications (beyond activity recognition), including multiple inter-related tasks. Moreover, our AL algorithm is computationally efficient due to the submodularity property and can be applied in scenarios involving joint learning of multiple recognition models. Also, unlike [68], we do not select a fixed number of samples from each batch; rather the number of samples is *non-uniform* based on the information content of each batch.

2.3 Methodology

2.3.1 Data Representation

The proposed method for informative sample selection is based on the assumption that the unlabeled data points have an underlying structure, i.e., relationships among them. We build a graph whose nodes represent the unlabeled samples to exploit the relationships between them. The two important measures which represent the graph are node and edge potentials.

Our active learning framework can select samples for single as well as multiple joint classification tasks simultaneously if the instances share relationship, e.g., scene-object, object-object, activity-object classification, etc. In order to generalize, let us consider that we have m tasks at hand which share relationships in data. Let us consider that we have a set of baseline classifiers $\mathcal{C} = \{\mathcal{C}_1, \ldots, \mathcal{C}_m\}$ for these m interrelated tasks. The node and edge potentials in the format we use are discussed below.

Node Potential. We represent each data point as a node. Consider that we a have total n categories $\{c_1, \ldots, c_n\}$ for these m classification problems. Consider an indicator function $\mathcal{I}(.)$ which takes as input a category name c and provides as output a unit standard basis vector, i.e., $\mathcal{I}(c = c_1) = [1, 0 \ldots, 0]^T$. If $\boldsymbol{f}_j$ is the feature of node j, then its node (unary) potential can be expressed as,

$$\boldsymbol{\phi}_j = \sum_{p=1}^{m} \sum_{i=1}^{n} \mathcal{C}_p(\boldsymbol{f}_j, c_i)\mathcal{I}(c = c_i) \tag{2.1}$$

where $\mathcal{C}_p(\boldsymbol{f}_j, c_i)$ is a scalar representing the probability of node j to belong to category c_i. $\mathcal{C}_p(\boldsymbol{f}_j, c_i) = 0$ if the training data of $\mathcal{C}_p$ does not contain data of category c_i.

Edge Potential. The edge (pair-wise) potential represents the relationships between the categories. The relationship model $\mathcal{R}$ contains the edge potential matrix $\boldsymbol{\psi}$ whose i,j location is the co-occurence frequency [58] of data point of category c_i with data point of category c_j. Co-occurrence, and thus edge potential, depends on the application and will be discussed in Section 2.4.

The node and edge potentials play an important role in our framework as we use it to construct a graph to represent the relationships between the data points. Note that our framework can be applied to any dataset containing relationships which can be modeled as edge potentials.

Graph Construction. Let us consider that we have a labeled set $\mathcal{L}$. We learn the baseline classification model $\mathcal{C}$ and a relationship model $\mathcal{R}$ with these labeled data $\mathcal{L}$. Now, consider that a new unlabeled dataset $\mathcal{U}$ becomes available with features $\{\boldsymbol{f}_j\}_{j=1}^N$, N being the size of the set $\mathcal{U}$. Instead of manually labeling this entire unlabeled set, our goal is to reduce the labeling effort by choosing an informative subset of $\mathcal{U}$ for manual labeling, such that it helps to improve the current models $\mathcal{C}$ and $\mathcal{R}$.

We start by constructing a graph $G = (V, E)$ with the instances in $\mathcal{U}$ using the current models $\mathcal{C}$ and $\mathcal{R}$. Each node in $V = \{v_1, \ldots, v_N\}$ represents each data point. The edges $E = \{(i, j)|v_i \text{ and } v_j \text{ are linked}\}$ represent the relationships between the data points. The link information between the nodes depends on the application and is discussed in Section 2.4. The nodes are assigned the corresponding node potentials $\boldsymbol{\phi}_i$ and the edges are assigned the edge potential $\boldsymbol{\psi}$. A message-passing algorithm can be used to infer the node and edge beliefs which are the marginal node probabilities and the pair-wise joint distribution of the edges respectively. We use Loopy Belief Propagation (LBP) [166] to accomplish this task.

2.3.2 Selection of Informative Samples

In this section, we discuss how we choose the informative samples based on the graphical model constructed from a batch of data. Using the node and edge probabilities, the goal is to choose a small subset $V^{l*} \subset V$ for manual labeling, which will improve the current models $\mathcal{C}$ and $\mathcal{R}$. We wish to select a subset of the nodes such that the joint entropy of all the nodes $H(V)$ is minimized. Below we derive an expression for the joint entropy of the graph G.

Joint Entropy of Nodes. The entropy of each node and the mutual information between a pair of nodes can be expressed as $H(v_i) = \mathbb{E}[-\log_2 p_i]$ and $I(v_i, v_j) = \mathbb{E}[\log_2 p_{ij}/p_i p_j]$ and p_i, p_j and p_{ij} are the node and edge probabilities respectively. The joint entropy of the nodes of the graph G can be expressed as follows,

$$\begin{aligned}
H(V) &\overset{(a)}{=} H(v_1) + \sum_{i=2}^{N} H(v_i|v_1, \ldots, v_{i-1}) \\
&\overset{(b)}{=} H(v_1) + \sum_{i=2}^{N} \Big[H(v_i) - I(v_1, \ldots, v_{i-1}; v_i)\Big] \\
&\overset{(c)}{=} H(v_1) + \sum_{i=2}^{N} \Big[H(v_i) - \sum_{j=1}^{i} I(v_j; v_i|v_1, \ldots, v_{j-1})\Big] \\
&= \sum_{i=1}^{N} H(v_i) - \sum_{i=2}^{N}\sum_{j=1}^{i} I(v_j; v_i|v_1, \ldots, v_{j-1}) \\
&\overset{(d)}{\approx} \sum_{v_i \in V} H(v_i) - \sum_{(i,j) \in E} I(v_j; v_i) \qquad (2.2)
\end{aligned}$$

(a)Joint entropy chain rule [40]

(b)Using $I(v_1, \ldots, v_{j-1}; v_j) = H(v_j) - H(v_j|v_1, \ldots, v_{j-1})$, where, $I(.;.)$ represents the mutual information between the set of random variables separated by ';'.

(c)Mutual information chain rule [40]

(d)Computing the conditional mutual information $I(v_j; v_i|v_1, \ldots, v_{j-1})$ becomes computationally intractable as the number of nodes on which it is conditioned increases. Moreover, we construct our graph using just unary (node) and pair-wise (edge) potentials and ignoring higher-order potentials. Thus, we approximate the conditional mutual information as $I(v_j; v_i|v_1, \ldots, v_{j-1}) \approx I(v_j; v_i)$. Furthermore, we consider two nodes to be independent if there exists no link between them. It is also known that the mutual information between two random variables is zero if they are independent.

The expression in Eqn 2.2 is similar to the expression of joint entropy using Bethe Approximation [224]. Moreover, this expression for joint entropy is exact for an acyclic graph but an approximation in case of graphs containing cycles. We use this expression to derive an objective function to be optimized in order to obtain the most informative nodes for manual labeling.

Objective Function Derivation. Our goal is to choose a subset of nodes from V, the size of which may vary across each batch of data, such that the joint entropy $H(V)$ in Eqn. 2.2 is minimized after inferring on the graph G conditioned on the obtained labels of the chosen nodes. To set up the optimization problem, let us partition the node-set V into two sets, V^l which will be selected for manual labeling and V^{nl} which will not be manually labeled. We need to find the optimal partition of V into these two sets by optimizing an objective function. The motivation is that the classifier is either confident or will become confident about the set V^{nl} if we gain information about the subset V^l. Here l means Labeled and nl means Not Labeled.

Let us define the two subgraphs of G as follows: $G^l = (V^l, E^l)$ be the subgraph whose nodes will be labeled and $G^{nl} = (V^{nl}, E^{nl})$ be the subgraph whose nodes will remain unlabeled. For the sake of completeness, the following are defined:

$$\begin{aligned} E^l &= \{(i,j)|(i,j) \in E, v_i, v_j \in V^l\} \\ E^{nl} &= \{(i,j)|(i,j) \in E, v_i, v_j \in V^{nl}\} \end{aligned} \tag{2.3}$$

Following the above partition, the joint entropy $H(V)$ can be partitioned as follows,

$$\begin{aligned} H(V) =& \Big[\sum_{v_i \in V^l} H(v_i) - \sum_{(i,j)\in E^l} I(v_j; v_i)\Big] + \\ & \Big[\sum_{v_i \in V^{nl}} H(v_i) - \sum_{(i,j)\in E^{nl}} I(v_j; v_i)\Big] - \sum_{\substack{(i,j)\in E \\ v_i \in V^l, v_j \in V^{nl}}} I(v_j; v_i) \\ =& H(V^l) + H(V^{nl}) - \sum_{\substack{(i,j)\in E \\ v_i \in V^l, v_j \in V^{nl}}} I(v_j; v_i) \end{aligned} \tag{2.4}$$

Once the nodes in V^l are manually labeled and we run inference on the graph conditioned on the acquired labels, the first and last term of the above expression becomes zero. This is because after acquiring labels for V^l, its every node become deterministic. Thus $H(V^l)$ becomes zero. Also, mutual information $I(v_i; v_j) = H(v_i) + H(v_j) - H(v_i, v_j) = 0$ when v_i is deterministic, as then $H(v_i) = 0, H(v_i, v_j) = H(v_j)$.

Most active learning algorithms assume that for each batch of unlabeled data, there is a fixed budget, i.e., the number of samples for manual labeling. If the budget for manual labeling is $K(\leq N)$, then the optimal subset V^{l*} which minimizes the joint entropy of the node can be expressed as,

$$V^{l*} = \underset{\substack{V^l \\ s.t. |V^l| = K}}{\arg\max} \Big[H(V^l) - \sum_{\substack{(i,j) \in E \\ v_i \in V^l, v_j \in V^{nl}}} I(v_j; v_i) \Big] \tag{2.5}$$

However, each batch of data may contain a non-uniform amount of information, and choosing the same number of budget-constrained samples (i.e., K) from each batch may not be a good idea. Instead, the number of samples could be determined based on the information content of each batch. This motivates us to modify the above objective function, such that we choose a non-uniform number of informative samples from a different batch of data. We rewrite Eqn. 2.5 as an unconstrained minimization problem as follows:

$$V^{l*} = \underset{V^l}{\arg\min} \Big[\sum_{\substack{(i,j) \in E \\ v_i \in V^l, v_j \in V^{nl}}} I(v_j; v_i) - H(V^l) + \lambda |V^l| \Big] \tag{2.6}$$

where λ is a positive trade-off parameter between maximizing the objective function in Eqn. 2.5 and minimizing the number of nodes chosen for manual labeling. The choice of λ is discussed at the end of this section.

The optimization problem can be represented in vector and matrix notations. For that, we define the following: consider a vector $\boldsymbol{x}$ of length N with elements being 1 or 0, where 1 represents the node is selected to be in the set V^l and 0 represents the opposite. We need to find the optimal $\boldsymbol{x}$ which solves the optimization problem in Eqn. 2.6. Let us define a N dimensional vector $\boldsymbol{h}$ of node entropies and a $N \times N$ matrix $\boldsymbol{M}$ of pairwise mutual informations as follows,

$$\boldsymbol{h} \triangleq [H(v_1), H(v_2) \dots H(v_N)]^T$$

$$\boldsymbol{M}(i,j) \triangleq \begin{cases} I(v_i; v_j), & \text{if } (i,j) \in E \\ 0, & \text{otherwise} \end{cases}$$

where $i, j \in \{1, \dots, N\}$. The objective function in Eqn. 2.6 can be represented as

$$\boldsymbol{x}^* = \arg\min_{\boldsymbol{x}} \frac{1}{2} \boldsymbol{x}^T \boldsymbol{Q} \boldsymbol{x} + \boldsymbol{x}^T \boldsymbol{f} + \lambda \boldsymbol{x}^T \mathbf{1} \tag{2.7}$$

where $\boldsymbol{Q} \triangleq -\boldsymbol{M}$ and $f \triangleq \boldsymbol{M}\mathbf{1} - \boldsymbol{h}$ and where $\mathbf{1} = [1 \;\; 1 \;\; \dots 1]^T$ of size $N \times 1$. The objective function in Eqn. 2.7 can be proved to be submodular which makes the optimization problem simpler compared to Eqn. 2.5. Details of the optimization is discussed next.

Proof of Submodularity. Considering $\mathcal{P}(S)$ as the power set of a finite set S, a submodular function is a set function $f : \mathcal{P}(S) \to \mathbb{R}$ if it satisfies the following,

$$f(X \cup \{v\}) - f(X) \geq f(Y \cup \{v\}) - f(Y) \tag{2.8}$$

where $X \subseteq Y \subseteq S$ and $v \in S - Y$. The sets are presented in Fig. 2.2 for a better understanding. Let us consider two vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ representing the two sets X and Y, i.e., if a node exists in a set, the corresponding element of the vector will be 1 else 0. Consider a vector $\boldsymbol{v}$ which represents the node v of Eqn. 2.8, i.e., $\boldsymbol{v}$ is a vector of all zeros and one at the v^{th} element location. Consider the objective function in Eqn. 2.7 be f.

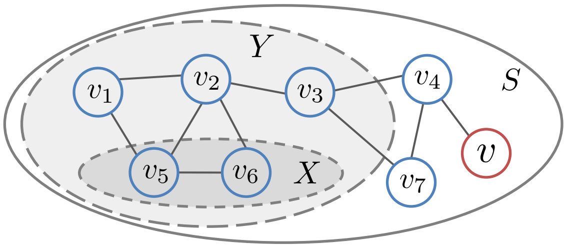


Figure 2.2: This figure is an example illustration of the sets S, X, Y, and the element v involved in proving that the proposed objective function is submodular.

Then,

$$\begin{aligned} f(X \cup \{v\}) - f(X) &= \left[\frac{1}{2}(\boldsymbol{x}+\boldsymbol{v})^T\boldsymbol{Q}(\boldsymbol{x}+\boldsymbol{v}) + (\boldsymbol{x}+\boldsymbol{v})^T\boldsymbol{f} + \lambda(\boldsymbol{x}+\boldsymbol{v})^T\boldsymbol{1}\right] \\ &\quad - \left[\frac{1}{2}\boldsymbol{x}^T\boldsymbol{Q}\boldsymbol{x} + \boldsymbol{x}^T\boldsymbol{f} + \lambda\boldsymbol{x}^T\boldsymbol{1}\right] \\ &= \frac{1}{2}\boldsymbol{v}^T\boldsymbol{Q}\boldsymbol{v} + \boldsymbol{x}^T\boldsymbol{Q}\boldsymbol{v} + \boldsymbol{v}^T\boldsymbol{f} + \lambda \end{aligned} \tag{2.9}$$

Also, $f(Y \cup \{v\}) - f(Y) = \frac{1}{2}\boldsymbol{v}^T\boldsymbol{Q}\boldsymbol{v} + \boldsymbol{y}^T\boldsymbol{Q}\boldsymbol{v} + \boldsymbol{v}^T\boldsymbol{f} + \lambda$

$$\{f(X \cup \{v\}) - f(X)\} - \{f(Y \cup \{v\}) - f(Y)\} = (\boldsymbol{x}-\boldsymbol{y})^T\boldsymbol{Q}\boldsymbol{v} \tag{2.10}$$

Now, as $X \subseteq Y$, $\boldsymbol{y}$ contains 1 at least in the positions where $\boldsymbol{x}$ contains 1. Thus, the entries of the vector $\boldsymbol{x}-\boldsymbol{y}$ are either 0 or -1. Also, the entries of $\boldsymbol{Q}$ are non-positive as $\boldsymbol{Q} = -\boldsymbol{M}$ and mutual information is always non-negative. Also, $\boldsymbol{v}$ is a vector of 1 at a single element and 0 otherwise. Thus, $(\boldsymbol{x}-\boldsymbol{y})^T\boldsymbol{Q}\boldsymbol{v} \geq 0$ and Eqn. 2.8 is satisfied, which makes the objective function in Eqn. 2.7 submodular and the optimization problem is submodular minimization.

Algorithm 1 Proposed Framework

```
Input: Sequential Batch of Unlabeled Data $\{\mathcal{U}_1, \mathcal{U}_2, \dots\}$.
Output: Classification $\mathcal{C}$ & Relationship $\mathcal{R}$ models after processing every batch of data.
Variable $\mathcal{L}$: Labeled Set, $k$: batch number
1. $\mathcal{L} \leftarrow \mathcal{U}_1$: Ask human to label the first batch $\mathcal{U}_1$.
2. Construct the models $\mathcal{C}$ and $\mathcal{R}$ using $\mathcal{L}$.
$k \leftarrow 2$
while new batch $\big(\mathcal{U}_k\big)$ available do
    3. Construct graph $G = (V, E)$ using $\mathcal{U}_k$
    4. Use the $\mathcal{C}$ and $\mathcal{R}$ to assign the node and edge potentials to $G$
    5. Run inference on $G$ to obtain the node $(p_i)$ and edge $(p_{ij})$ probabilities
    6. Compute the entropy & mutual information to construct $\boldsymbol{h}$ & $\boldsymbol{M}$ respectively.
    7. Find $\lambda$ using Eqn. 2.11
    8. Obtain $\boldsymbol{x}^*$ in Eqn. 2.7 using Fujishige-Wolfe Min Norm Point algorithm
    9. Use $\boldsymbol{x}^*$ to select the samples for query to human, denoted by $V^{l*}$. Then, $\mathcal{L} \leftarrow \mathcal{L} \cup V^{l*}$
    10. Infer conditioned on the acquired labels and $\mathcal{L} \leftarrow \mathcal{L} \cup \{$Highly confident instances$\}$
    11. Use $\mathcal{L}$ to update the models $\mathcal{C}$ and $\mathcal{R}$
    $k \leftarrow k + 1$
end while
```

Optimization Procedure. Submodular Function Minimization (SFM) often arises in fields of machine learning, game theory, information theory, etc. Detailed description may be found here [121]. There exist some algorithms which can be used to solve SFM in polynomial time. We use the Fujishige-Wolfe Min Norm Point algorithm [57] in the Submodular Function Optimization (SFO) [99] toolbox to solve the submodular minimization problem in Eqn. 2.7. It is one of the most well-known algorithms to solve SFM.

Trade-off Parameter. The parameter λ in Eqn. 2.7 is a trade-off between the two objectives as discussed previously. If $f(\boldsymbol{x})$ is the objective function in Eqn. 2.7, then λ can be expressed as,

$$\lambda = \alpha \frac{\min_{\boldsymbol{x}} f(\boldsymbol{x})|_{\lambda=0} - 0}{0 - \max_{\boldsymbol{x}} \boldsymbol{x}^T \mathbf{1}} \tag{2.11}$$

where α is a scalar parameter. In Eqn. 2.11, a fraction is obtained using the range of values of the two objective functions, such that the scaling between the two objective functions using λ is appropriate. λ now depends on α, which can be kept close to 1 for all applications due to the scaling done in Eqn. 2.11 between the two objective functions.

Model Update After the chosen samples are labeled by a human annotator, we perform inference on the graph, conditioned on the acquired labels to update the beliefs of the nodes and then we apply the concept of weak teacher [235], which does not involve the human. We choose those nodes having the confidence in classification $> \epsilon$, with the corresponding label, to be in the labeled set $\mathcal{L}$. ϵ should be high enough to avoid incorrect labeling. The classification model $\mathcal{C}$ is updated by retraining the classifier using $\mathcal{L}$. Model $\mathcal{R}$ is comprised of only the co-occurrence matrix $\boldsymbol{\psi}$ and it is incremented using the new labeled instances. An overview of the entire framework is presented in Algorithm 1.

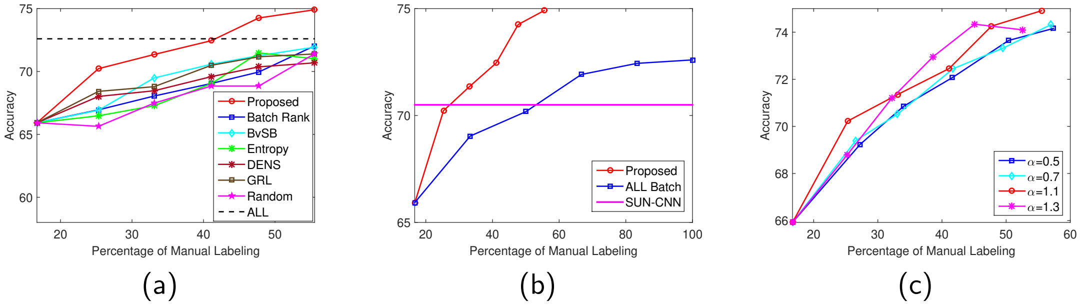
Special Case of Archived Data. Note we discussed the proposed method to be used in a continual learning set-up where data comes in batches with time. However, the proposed framework can also be used in cases where the entire dataset is available at the outset. In that case, a small set of samples is randomly selected from the unlabeled dataset and their labels are obtained. These labeled samples are used to construct the initial models $\mathcal{C}$ and $\mathcal{R}$. These models are used to choose the informative samples from the rest of the unlabeled pool of samples and then the models are updated after acquiring the labels. This process continues until the joint entropy of the remaining subset becomes less than a certain threshold.

2.4 Experiments

In this section, we present an experimental analysis of our proposed active learning framework for three different applications - joint scene-object classification, activity recognition, and document classification. These applications are chosen as they have data that share relationships among them. For each application, we perform the following experiments.

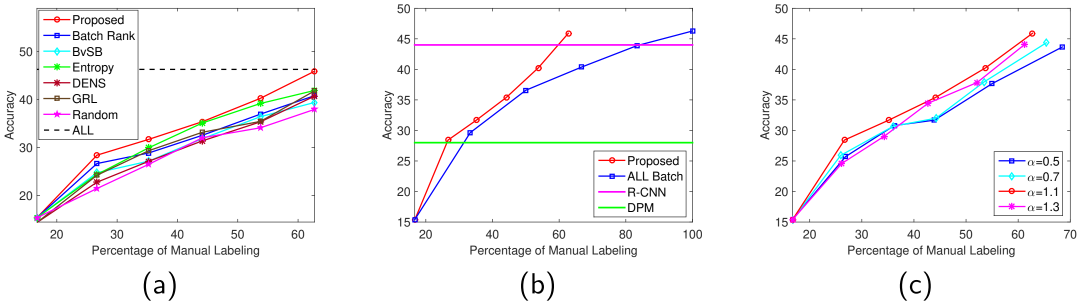
- We compare the proposed method with commonly used and state-of-the-art active learning methods namely - Batch Rank [23], BvSB [110], Entropy [171, 76], Density Based Sampling (DENS) [171], Expected Gradient Length (GRL) [172] and Random Sampling. We also compare with CAAL [68] for activity recognition.
- We compare the results of our algorithm with other state-of-the-art methods that use the entire dataset for training, details of which are mentioned subsequently.
- We perform a sensitivity analysis of the proposed method on the parameter α in Eqn. 2.11.

We use Support Vector Machine (SVM) [24] as a baseline classifier in our proposed method as well as for all the active learning methods with which we compare, to have a fair comparison. We use the Undirected Graphical Model (UGM) toolbox [166] to perform inference on the graph. We will use the following short-notations. “ALL” represents the accuracy obtained by using the entire dataset for training.“ALL Batch” denotes that the classifier is updated using ALL the instances of the current batch.



(a) (b) (c)

Figure 2.3: This figure presents the results on the SUN dataset for **scene** recognition. (a) presents the comparison of the proposed method with other active learning methods. (b) presents the comparison with other methods which use the entire dataset for training. (c) presents the sensitivity of the proposed method to the parameter α.



(a) (b) (c)

Figure 2.4: This figure presents the results on the SUN dataset for **object** detection. (a) presents the comparison of the proposed method with other active learning methods. (b) presents the comparison with other methods which use the entire dataset for training. (c) presents the sensitivity of the proposed method to the parameter α.

2.4.1 Scene-Object Classification

Scene and objects tend to co-occur in images. Although scene and objects classifiers are separate, their joint understanding can be beneficial [230], which can be exploited in our active learning framework to reduce manual labeling.

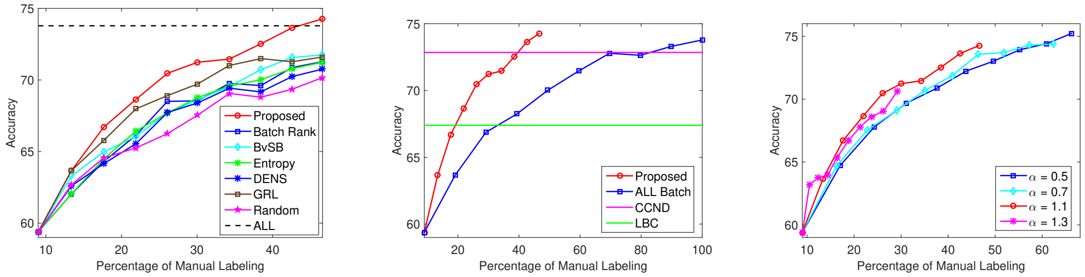
Dataset. We use the SUN dataset [37, 226] for our experiments on scene-object classification. We use that portion of the dataset which has both scene and object annotations as we aim to exploit their relationship. In order to represent the scene nodes, we extract

CNN features ($\in \mathbb{R}^{4096}$) from the fc-7 layer of VGG-net [241] pre-trained on the places-205 dataset. We use the pipeline of R-CNN [62] to detect the objects and then extract CNN features from fc-7 layer of Alex-net [101], pre-trained on ImageNet [45].

Experimental Set-up. We perform 5 Fold Cross-Validation (FCV) for this dataset. The training data of 4 folds are divided into 6 batches and fed sequentially to our active learning framework. We consider that the first batch is manually labeled and use it to construct the initial models $\mathcal{C}$ and $\mathcal{R}$. We assume that the other batch of data is unlabeled and we choose only the informative samples for manual labeling, which is then used to update the models. It may be noted that this application is an example that depicts that our algorithm can be applied for active learning of different recognition tasks jointly. Each image is represented by a single scene node and multiple object nodes as detected by the detector. The graph for this application is considered to be fully connected and the i, j position of the edge potential matrix is a count of the number of times an object of category i appears in a scene of category j.

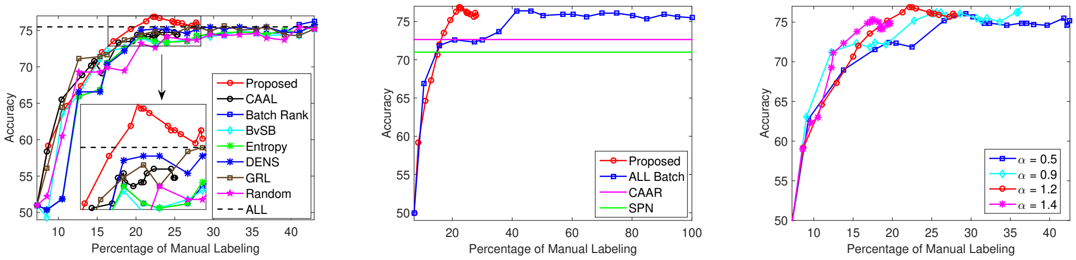
Results. Fig. 2.3(a) and 2.4(a) presents the comparison of the proposed method with other state-of-the-art active learning methods. The proposed method performs better than the other methods and reaches the "ALL" mark with only 41% and 62% manual labeling for scene and objects respectively.

Fig. 2.3(b) and 2.4(b) presents the results of the proposed method along with methods that consider that the entire dataset is manually labeled and available for training. We compare our method with SUN-CNN [241] for scene classification and with R-CNN [62] and DPM [54] for object recognition. As may be observed, the proposed method requires



(a) (b) (c)

Figure 2.5: This figure presents the results on the CORA dataset for **document** classification. (a) presents the comparison of the proposed method with other active learning methods. (b) presents the comparison with other methods which use the entire dataset for training. (c) presents the sensitivity of the proposed method to the parameter α.



(a) (b) (c)

Figure 2.6: This figure presents the results on the VIRAT dataset for **activity** classification. (a) presents the comparison of the proposed method with other active learning methods. (b) presents the comparison with other methods which use the entire dataset for training. (c) presents the sensitivity of the proposed method to the parameter α.

a much lesser number of samples to be manually labeled to obtain the same accuracy as "ALL Batch".

Fig. 2.3(c) and 2.4(c) present the results of the proposed method for different values of the parameter α in Eqn. 2.11. It may be noted that $\alpha = 1.1$ have been used for all the results corresponding to the SUN dataset.

2.4.2 Document Classification

Documents are generally inter-linked by citations and hyperlinks, which may be exploited using our active learning approach to reduce manual labeling effort.

Dataset. We use the CORA dataset [169] for our experiments on document classification. It is a dataset containing 2708 scientific publications divided into seven categories. There are a total of 5429 links (citations) between the publications. The publications are represented using a dictionary of 1433 unique words and the feature vectors $\boldsymbol{f}_i \in \{0, 1\}^{1433}$ indicate the absence or presence of these words.

Experimental Set-up. We perform 10 FCV for this dataset following [169] and follow a similar set-up as discussed previously for scene-object. We construct the graph such that each node is connected to its five nearest neighbors in the feature space. The i, j position of the edge potential matrix is a count of the number of times a publication belonging to category i is related to category j via a citation link.

Results. The results of the proposed AL method along with other state-of-the-art AL methods is presented in Fig. 2.5(a). It may be observed that the proposed method performs much better than the other algorithms and requires only 42% manual labeling to reach "ALL".

We also compare our proposed method with other methods which consider that the entire dataset is manually labeled and use it for training. Fig. 2.5(b) presents the comparison with two such methods namely CCND [169] and LBC [168] [1]. The proposed method performs much better than "ALL Batch", which signifies that the proposed method ex-

[1] Please note that the horizontal lines should be points at 100% manual labeling, but for better visualization, we have presented them as lines.

tracts maximum possible information from the unlabeled set, but using much lesser manual labeling.

We also present analysis of the parameter α in Eqn. 2.11 and the plots are presented in Fig. 2.5(c). The results in Fig. 2.5(a) and 2.5(b) is with $\alpha = 1.1$. Lower the value of α, lesser will be the penalty for the number of samples chosen per batch (Eqn. 2.7), thus more samples will be chosen. This is also evident from Fig. 2.5(c). Although, the performance with $\alpha = 0.5$ is similar to $\alpha = 1.1$ at the end, the later chooses much lesser number of samples for manual labeling.

2.4.3 Activity Classification

Activities are generally spatio-temporally related which can be exploited to reduce the number of instances chosen for manual labeling. **Dataset.** We use the VIRAT dataset [133] on human activity for our experiments on activity classification. The dataset consists of 11 videos segmented into 329 activity sequences. We extracted features using the pre-trained model of 3D convolutional networks [199]. We extract the features for 16 frames at a time with a temporal stride of 8 and then apply max pooling along the temporal dimension to obtain a single vector $\in \mathbb{R}^{4096}$ for each activity.

Experimental Set-up. We have used the first 176 sequence (761 activity) for training and 153 sequence (661 activities) for testing. We have divided the training set into 20 batches and fed them sequentially to our active learning algorithm. We consider that there exists a link between two activities if they have occurred within a certain spatio-temporal distance. We consider the edge potential to be the spatio-temporal co-occurrence between the two activities.

Results. The results of the proposed active learning algorithm with other state-of-the-art active learning methods is presented in Fig. 2.6(a). It may be observed that the proposed method not only reaches the accuracy of “ALL” with only 18% manual labeling, but also performs better than “ALL”. The fact that an algorithm can perform better than “ALL”, i.e. using the entire dataset for training is discussed in [104]. Although Batch Rank reaches “ALL”, it requires much more manual labeling than required by the proposed method. ”CAAL” remains close to the proposed algorithm initially, but the latter peaks up thereafter.

We compare the proposed method in in Fig. 2.6(b) with other learning algorithms which consider the entire dataset to be manually labeled and use it for training namely - Context Aware Activity Recognition (CAAR) [244] and Sum Product Network (SPN) [3]. It may be observed that the proposed method peaks much faster than “ALL Batch” which indicates that the former requires lesser manual labeling in each batch to obtain the same accuracy as when the entire batch is manually labeled and used for training. The plots for sensitivity analysis of the parameter α for the VIRAT dataset is presented in Fig. 2.6(c).

2.5 Conclusions

In this chapter, we presented and evaluated a novel generalized active learning framework for inter-related data. Our framework can be applied for active learning of both single as well as multiple recognition tasks simultaneously by exploiting the inter-relationships in data. Our proposed method selects *non-uniform* number of samples from each batch depending on the information content. The proposed informative subset selection methodology is not only fast due to its submodular property, but also performs well on a wide range of applications. Further, in [67] we show that our method can also be used when we have a given fixed budget for manual annotation per batch of unlabeled data. We also show that other contextual information, such as objects in case of activity recognition, can be exploited as side information for better performance of our model. An interesting future direction of work could be to investigate the scenario the scenario where the labels provided by human is not always correct.

Chapter 3

Weakly Supervised Event Localization

3.1 Introduction

Temporal activity localization and classification in continuous videos is a challenging and interesting problem in computer vision [1]. Its recent success [227, 238] has evolved around a *fully* supervised setting, which considers the availability of frame-wise activity labels. However, acquiring such precise frame-wise information requires enormous manual labor. This may not scale efficiently with a growing set of cameras and activity categories. On the other hand, it is much easier for a person to provide a few categorical labels which encapsulate the content of a video. Moreover, videos available on the web are often accompanied by tags that provide semantic discrimination. Such video-level labels are generally termed as *weak* labels, which may be utilized to learn models with the ability to classify

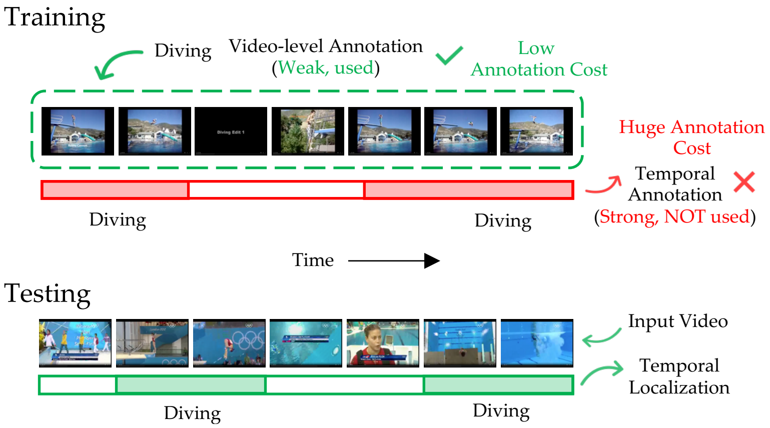


Figure 3.1: This figure presents the train-test protocol of weakly supervised action localization. The training set consists of videos with their video-level activity tags and NOT the temporal annotation. Whereas, while testing, the network not only estimates the labels of the activities in the video but also temporally locates their occurrence.

and localize activities in videos. In this chapter, we present a novel framework [142] for Temporal Activity Localization and Classification (TALC) from such weak labels. Fig. 3.1 presents the train-test protocol of W-TALC.

In computer vision, researchers have utilized weak labels to learn models for several tasks including semantic segmentation [66, 92, 228], visual tracking [239], reconstruction [204, 88], video summarization [136], learning robotic manipulations [184], video captioning [173], object boundaries [93], place recognition [5], and so on. The weak TALC problem is analogous to weak object detection in images, where object category labels are provided at the image-level. There have been several works in this domain mostly utilizing the techniques of Multiple Instance Learning (MIL) [242] due to their close relation in terms of the structure of information available for training. The positive and negative bags required for MIL are generated by state-of-the-art region proposal techniques [109, 85]. On the other hand, end-to-end learning with categorical loss functions are presented in [51, 52, 46, 185] and the authors in [243] incorporated the proposal generation network in an end-to-end manner.

Temporal localization using weak labels is a much more challenging task compared to weakly-supervised object detection. The key reason is the additional variation in content as well as the length along the temporal axis in videos. Activity localization from weakly labeled data remains relatively unexplored. Some works [186, 228, 192] focus on weakly-supervised spatial segmentation of the actor region in short videos. Another set of works [16, 102, 153, 78] considers video-level labels of the activities and their temporal ordering during training. However, such information about the activity order may not be available or may not make sense for a majority of web-videos. [222] utilizes state-of-the-art object detectors for spatial annotations but considers full temporal supervision. In [218], a soft selection module is introduced for untrimmed video classification along with activity localization and a sparsity constraint is included in [132].

In W-TALC, as we have labels only for the entire video, we need to process them at once. Processing long videos at fine temporal granularity may have considerable memory and computation requirements. On the other hand, coarse temporal processing may result in reduced detection granularity. Thus, there is a trade-off between performance and computation. Over the past few years, networks trained on ImageNet [45] and recently on Kinetics [90], has been used widely in several applications. Based on these advances in literature and the aforementioned trade-off, we may want to ask the question that: *is it possible to utilize these networks just as feature extractors and develop a framework for weakly-supervised activity localization which learns only the task-specific parameters, thus*

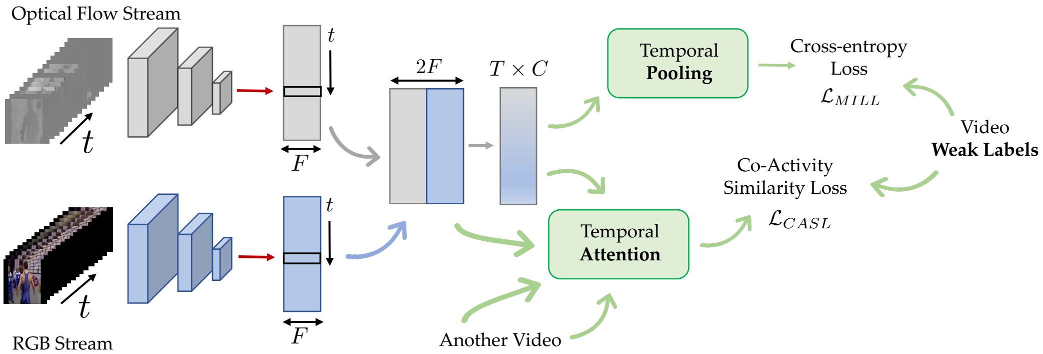


Figure 3.2: This figure presents the proposed framework for weakly-supervised activity localization and classification. Given a video, we extract features from two streams - RGB and Optical Flow. After concatenating the feature vectors from the two streams, we learn a few layers specific to the task of weak localization and finally project to the category space to obtain a $T \times C$ matrix where T and C are the number of time steps and categories respectively. We utilize two loss functions to learn the network parameters - Cross-entropy loss on the temporally pooled predictions, and Co-Activity Loss obtained using a pair of videos containing at least one category in common.

scaling up to long videos and processing them at fine temporal granularity? To address this question, *we present a framework (W-TALC) for weakly-supervised temporal activity localization and video classification, which utilizes pair-wise video similarity constraints via an attention-based mechanism along with multiple instances learning to learn only the task-specific parameters.*

Framework Overview. A pictorial representation of W-TALC is presented in Fig. 3.2. The proposed method utilizes off-the-shelf Two-Stream networks [218, 22] as a feature extractor. The number of frame inputs depend on the network used and will be discussed in Section 3.3.1. After passing the frames through the networks, we obtain a matrix of feature vectors with one dimension representing the temporal axis. Thereafter, we apply a FullyConnected-ReLU-Dropout layer followed by the label space projection layer, both of which is learned for the weakly-supervised task.

The activations over the label space are then used to compute two complimentary loss functions using video-level labels. The first one is Multiple Instance Learning Loss, where the category-wise k-max-mean strategy is employed to pool the category-wise activations and obtain a probability mass function over the categories. Its cross-entropy with the ground-truth label is the Multiple Instance Learning Loss (MILL). The second one is the Co-Activity Similarity Loss (CASL), which is based on the motivation that a pair of videos having at least one activity category (say biking) in common should have similar features in the temporal regions which correspond to that activity. Also, the features from one video corresponding to biking should be different from the features of the other video (of the pair) not corresponding to biking. However, as the temporal labels are not known in weakly-supervised data, we use the attention obtained from the label space activations as weak temporal labels, to compute CASL. Thereafter, we jointly minimize the two loss functions to learn the network parameters.

Main contributions. The main contributions of the proposed method are as follows. 1. We propose a novel approach for weakly-supervised temporal activity localization and video classification, without fine-tuning the feature extractor, but learning only the task-specific parameters. Our method does not consider any ordering of the labels in the video during training and can detect multiple activities in the same temporal duration.

2. We introduce the Co-Activity Similarity Loss and jointly optimize it with the Multiple Instance Learning Loss to learn the network weights specific to the weakly-supervised task. We empirically show that the two loss functions are complementary in nature.

3. We perform extensive experiments on two challenging datasets and show that the pro-

posed method performs better than the current state-of-the-art methods.

3.2 Related Works.

The problem of learning from weakly-supervised data has been addressed in several computer vision tasks including object detection [12, 52, 109, 175, 38, 185], segmentation [210, 138, 11, 92, 221], video captioning [173] and summarization [136]. Here, we discuss in detail the other works which are more closely related to our work.

Weakly-supervised Spatial Action Localization. Some researchers have looked into the problem of spatial localization of actors in mostly short and trimmed videos using weak supervision. In [28] a framework is developed for localization of players in sports videos, using detections from a state-of-the-art fully supervised player detector, as inputs to their network. Person detectors are also used in [186, 223] to generate person tubes, which is used to learn different Multiple Instance Learning-based classifiers. Conditional Random Field (CRF) is used in [228] to perform actor-action segmentation from video-level labels but on short videos.

Scripts as Weak Supervision. Some works in the literature use scripts or subtitles generally available with videos as weak labels for activity localization. In [105, 50] words related to human actions are extracted from subtitles to provide coarse temporal localization of actions for training. In [15], actor-action pairs extracted from movie scripts serve as weak labels for spatial actor-action localization by using discriminative clustering. Our algorithm on the other hand only considers that the label of the video is available as a whole, agnostic to the source from where the labels are acquired, i.e., movie scripts, subtitles, humans.

Temporal Localization with Ordering. Few works in the literature have considered the availability of temporal order of activities, apart from the video-level labels during training. The activity orderings in the training videos are used as constraints in discriminative clustering to learn activity detection models [16]. A similar approach was taken in [17]. In [78], the authors propose a dynamic programming-based approach to evaluate and search for possible alignments between video frames and the corresponding labels. The authors in [153] use a Recurrent Neural Network (RNN) to iteratively train and realign the activity regions until convergence. A similar iterative process is presented also in [102], but without employing an RNN. Unlike these works in literature, our work does not consider any information about the orderings of the activity.

The works in [218, 132] are closely related to the problem setting presented in this chapter. However, as the framework in [218] is based on the temporal segments network [219], a fixed number of segments, irrespective of the length of the video, are considered during training, which may lead to a reduction in localization granularity. Moreover, they only employ the MILL, which may not be enough to localize activities at a fine temporal granularity. A sparsity-based loss function is optimized in [132], along with a loss function similar to that obtained using the soft selection method in [218]. We introduce a novel loss function named Co-Activity Similarity Loss (CASL) which imposes pair-wise constraints for better localization performance. We also propose a mechanism for dealing with long videos and yet detecting activities at a high temporal granularity. In spite of not finetuning the feature extractor, we can still achieve better performance than state-of-the-art methods on weak TALC. Moreover, results show that CASL is complementary in nature with MILL.

3.3 Methodology

In this section, we present our framework (W-TALC) for weakly-supervised activity localization and classification. First, we present our mechanism to extract features from the two standard networks, followed by the layers of the network we learn. Thereafter, we present two loss functions MILL and CASL, which we jointly optimize to learn the network parameters. It may be noted that we compute both the loss functions using only the video-level labels of the training videos. Before going into the details of our framework, let us define the notations and problem statement formally.

Problem Statement. Consider that we have a training set of n videos $\mathcal{X} = \{\boldsymbol{x}_i\}_{i=1}^{n}$ with variable temporal duration denoted by $L = \{l_i\}_{i=1}^{n}$ (after feature extraction) and activity label set $\mathcal{A} = \{a_i\}_{i=1}^{n}$, where $a_i = \{a_i^j\}_{j=1}^{m_i}$ are the $m_i(\geq 1)$ labels for the i^{th} video. We also define the set of activity categories as $\mathcal{S} = \bigcup_{i=1}^{n} a_i = \{\alpha_i\}_{i=1}^{n_c}$. During test time, given a video $\boldsymbol{x}$, we need to predict a set $x_{det} = \{(s_j, e_j, c_j, p_j)\}_{j=1}^{n(\boldsymbol{x})}$, where $n(\boldsymbol{x})$ is the number of detections for $\boldsymbol{x}$. s_j, e_j are the start time and end time of the j^{th} detection, c_j represents its predicted activity category with confidence p_j. With these notations, our proposed framework is presented next.

3.3.1 Feature Extraction

We focus particularly on two architectures - UntrimmedNets [218] and I3D [22] for feature extraction, mainly due to their two-stream nature, which incorporates rich temporal temporal information in one of the streams, necessary for activity recognition. Please note that the rest of our framework is agnostic to the features used.

UntrimmedNet Features. In this case, we pass one frame through the RGB stream and 5 frames through the Optical Flow stream as in [218]. We extract the features from just before the classification layer at 2.5 fps. We use the network which is pre-trained on ImageNet [45], and finetuned using weak labels and MILL on the task-specific dataset as in [218]. Thus, this feature extractor has no knowledge about activities using strong labels.

I3D Features. As in [132], we also experiment with features extracted from the Kinetics pre-trained I3D network [22]. The input to the two streams is non-overlapping 16 frame chunks. The output is passed through a 3D average pooling layer of kernel size $2 \times 7 \times 7$ to obtain features of dimension 1024 each from the two streams.

At the end of the feature extraction procedure, each video $\boldsymbol{x}_i$ is represented by two matrices $\boldsymbol{X}_i^r$ and $\boldsymbol{X}_i^o$, denoting the RGB and optical flow features respectively, both of which are of dimension $1024 \times l_i$. Note that l_i is not only dependent on the video index i, but also on the feature extraction procedure used, but it is proportional to the length of the video. These matrices become the input to our weakly-supervised learning module.

Memory Constraints. As mentioned previously, natural videos may have large variations in length, from a few seconds to more than an hour. In the weakly-supervised setting, we have information about the labels for the video as a whole, thus requiring it to process the entire video at once. This may be problematic for very long videos due to GPU memory constraints. A possible solution to this problem may be to divide the videos into chunks along the temporal axis [219] and apply a temporal pooling technique to reduce the length of each chunk to a single representation vector. The number of chunks depends on the available GPU memory. However, this will introduce unwanted background activity feature in the representation vectors as the start and end period of the activities in the video will not overlap with the pre-defined chunks for most of the videos. To cope with this problem, we introduce a simple video sampling technique.

Long Video Sampling. As the granularity of localizations is important for activity localization, we take an approach alternative to the one mentioned above. We process the entire video if its length is less than the pre-defined length T necessary to meet the GPU bandwidth. However, if the length of the video is greater than T, we randomly extract from it a clip of length T with contiguous frames and assign all the labels of the entire video to the extracted video clip. It may be noted that although this may introduce some errors in the labels, this way of sampling does have advantages, as will be discussed in more detail in Section 3.4.

Computational Budget and Finetuning. The error introduced by the video sampling strategy will increase with a decrease in the pre-defined length of T, which meets the GPU bandwidth. If we want to jointly finetune the feature extractor along with training our weakly-supervised module, T may be very small in order to maintain a reasonable batch size for Stochastic Gradient Descent (SGD) [19]. Although the value of T may be increased by using multiple GPUs simultaneously, it may not be a scalable approach. Moreover, the time to train both the modules may be high. Considering these problems, we do not finetune the feature extractors, but only learn the task-specific parameters, described next, from scratch. The advantages for doing this are twofold - the weakly-supervised module is light-weight in terms of the number of parameters, thus requiring less time to train, and it increases T considerably, thus reducing labeling error while sampling long videos.

3.3.2 Weakly Supervised Layer

In this section, we present the proposed weakly-supervised learning scheme, which uses only weak labels to learn models for simultaneous activity localization and classification.

Fully Connected Layer. We introduce a fully connected layer followed by ReLU [130] and Dropout [189] on the extracted features. The operation can be formalized for a video with index i as follows.

$$\boldsymbol{X}_i = \mathcal{D}\Big(\max\Big(0, \boldsymbol{W}_{fc} \begin{bmatrix} \boldsymbol{X}_i^r \\ \\ \boldsymbol{X}_i^o \end{bmatrix} \oplus \boldsymbol{b}_{fc}\Big), k_p\Big) \tag{3.1}$$

where $\mathcal{D}$ represents `Dropout` with k_p representing its keep probability, $\oplus$ is the addition with broadcasting operator, $\boldsymbol{W}_{fc} \in \mathbb{R}^{2048\times 2048}$ and $\boldsymbol{b}_{fc} \in \mathbb{R}^{2048}$ are the parameters to be learned from the training data and $\boldsymbol{X}_i \in \mathbb{R}^{2048\times l_i}$ is the output feature matrix for the entire video.

Label Space Projection We use the feature representation $\boldsymbol{X}_i$ to classify and localize the activities in the videos. We project the representations $\boldsymbol{X}_i$ to the label space ($\in \mathbb{R}^{n_c}$, n_c is the number of categories), using a fully connected layer, with weight sharing along the temporal axis. The category-wise activations we obtain after this projection can be represented as follows.

$$\boldsymbol{\mathcal{A}}_i = \boldsymbol{W}_a \boldsymbol{X}_i \oplus \boldsymbol{b}_a \tag{3.2}$$

where $\boldsymbol{W}_a \in \mathbb{R}^{n_c\times 2048}$, $\boldsymbol{b}_a \in \mathbb{R}^{n_c}$ are to be learned and $\boldsymbol{\mathcal{A}}_i \in \mathbb{R}^{n_c\times l_i}$. These category-wise activations represent the possibility of activities at each of the temporal instants.

3.3.3 k-max Multiple Instance Learning

As discussed in Section 3.1, the weakly-supervised activity localization and classification problem as addressed in this chapter can be directly mapped to the problem of Multiple Instance Learning (MIL) [242]. In MIL, individual samples are grouped into two bags, namely positive and negative bags. A positive bag contains at least one positive instance and a negative bag contains no positive instance. Using these bags as training data, we need to learn a model, which will be able to distinguish each instance to be positive or negative, besides classifying a bag. In our case, we consider the entire video as a bag of instances, where each instance is represented by a feature vector at a certain time instant. In order to compute the loss for each bag, i.e., video in our case, we need to represent each video using a single confidence score per category. For a given video, we compute the activation score corresponding to a particular category as the average of k-max activation over the temporal dimension for that category. As in our case, the number of elements in a bag varies widely, we set k proportional to the number of elements in a bag. Specifically,

$$k_i = \max\left(1, \left\lfloor \frac{l_i}{s} \right\rfloor\right) \tag{3.3}$$

where s is a design parameter. Thus, our category-wise confidence scores for the j^{th} category of the i^{th} video can be represented as,

$$s_i^j = \frac{1}{k_i} \max_{\substack{\mathcal{M} \subset \boldsymbol{\mathcal{A}}_i[j,:] \\ |\mathcal{M}| = k_i}} \sum_{l=1}^{k_i} \mathcal{M}_l \tag{3.4}$$

where $\mathcal{M}_l$ indicates the l^{th} element in the set $\mathcal{M}$. Thereafter, a softmax non-linearity is applied to obtain the probability mass function over the all the categories as follows, $p_i^j = \frac{\exp(s_i^j)}{\sum_{j=1}^{n_c}\exp(s_i^j)}$. We need to compare this pmf with the ground truth distribution of labels for each video in order to compute the MILL. As each video can have multiple activities occurring in it, we represent the label vector for a video with ones at the positions if that activity occurs in the video, else zero. We then normalize this ground truth vector in order to convert it to a legitimate pmf. The MILL is then the cross-entropy between the predicted pmf $\boldsymbol{p}_i$ and ground-truth, which can then be represented as follows,

$$\mathcal{L}_{MILL} = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{n_c} -y_i^j \log(p_i^j) \tag{3.5}$$

where $\boldsymbol{y}_i = [y_i^1, \ldots, y_i^{n_c}]^T$ is the normalized ground truth vector. This loss function is semantically similar to that used in [218]. We next present the novel Co-Activity Similarity Loss, which enforces constraints to learn better network parameters for activity localization.

3.3.4 Co-Activity Similarity

As discussed previously, the W-TALC problem motivates us to identify the correlations between videos of similar categories. Before discussing in more detail, let us define category-specific sets for the j^{th} category as, $\mathcal{S}_j = \{\boldsymbol{x}_i \mid \exists\ a_i^k \in \boldsymbol{a}_i, \text{s.t. } a_i^k = \alpha_j\}$, i.e., the set $\mathcal{S}_j$ contains all the videos of the training set, which has activity α_j as one of its labels. Ideally, we may want the following properties in the learned feature representations $\boldsymbol{X}_i$ in Eqn. 3.1.

- A video pair belonging to the set $\mathcal{S}_j$ (for any $j \in \{1, \ldots, n_c\}$) should have similar feature representations in the portions of the video where the activity α_j occurs.

- For the same video pair, feature representation of the portion where α_j occurs in one video should be different from that of the other video where α_j does not occur.

These properties are not directly enforced in MILL. Thus, we introduce Co-Activity Similarity Loss to embed the desired properties in the learned feature representations. As we do not have frame-wise labels, we use the category-wise activations obtained in Eqn. 3.2 to identify the required activity portions. The loss function is designed in a way that helps to learn simultaneously the feature representation as well as the label space projection. We first normalize the per-video category-wise activations scores along the temporal axis using softmax non-linearity as follows:

$$\hat{\boldsymbol{\mathcal{A}}}_i[j,t] = \frac{\exp(\boldsymbol{\mathcal{A}}_i[j,t])}{\sum_{t'=1}^{l_i} \exp(\boldsymbol{\mathcal{A}}_i[j,t'])} \tag{3.6}$$

where t indicates the time instants and $j \in \{1, \ldots, n_c\}$. We refer to these as *attention*, as they attend to the portions of the video where the activity of a certain category occurs. A high value of attention for a particular category indicates its high occurrence-probability of that category. In order to formulate the loss function, let us first define the category-wise feature vectors of regions with high and low attention as follows:

$$\begin{aligned} {}^{H}\boldsymbol{f}_i^j &= \boldsymbol{X}_i \hat{\boldsymbol{\mathcal{A}}}_i[j,:]^T \\ {}^{L}\boldsymbol{f}_i^j &= \frac{1}{l_i - 1} \boldsymbol{X}_i \Big(\mathbf{1} - \hat{\boldsymbol{\mathcal{A}}}_i[j,:]^T\Big) \end{aligned} \tag{3.7}$$

where ${}^{H}\boldsymbol{f}_i^j, {}^{L}\boldsymbol{f}_i^j \in \mathbb{R}^{2048}$ represents the high and low attention region aggregated feature representations respectively of video i for category j. It may be noted that in Eqn. 3.7 the low attention feature is not defined if a video contains a certain activity and the number of feature vectors, i.e., $l_i = 1$. This is also conceptually valid and in such cases, we cannot compute the CASL. We use cosine similarity in order to obtain a measure of the degree of similarity between two feature vectors and it may be expressed as follows:

$$d[\boldsymbol{f}_i, \boldsymbol{f}_j] = 1 - \frac{\langle \boldsymbol{f}_i, \boldsymbol{f}_j \rangle}{\langle \boldsymbol{f}_i, \boldsymbol{f}_i \rangle^{\frac{1}{2}} \langle \boldsymbol{f}_j, \boldsymbol{f}_j \rangle^{\frac{1}{2}}} \tag{3.8}$$

In order to enforce the two properties discussed above, we use the ranking hinge loss. Given a pair of videos $\boldsymbol{x}_m, \boldsymbol{x}_n \in \mathcal{S}_j$, the loss function may be represented as follows:

$$\begin{aligned} \mathcal{L}_j^{mn} = \frac{1}{2}\Big\{ & \max\Big(0, d[{}^{H}\boldsymbol{f}_m^j, {}^{H}\boldsymbol{f}_n^j] - d[{}^{H}\boldsymbol{f}_m^j, {}^{L}\boldsymbol{f}_n^j] + \delta\Big) \\ & + \max\Big(0, d\big[{}^{H}\boldsymbol{f}_m^j, {}^{H}\boldsymbol{f}_n^j\big] - d\big[{}^{L}\boldsymbol{f}_m^j, {}^{H}\boldsymbol{f}_n^j\big] + \delta\Big)\Big\} \end{aligned} \tag{3.9}$$

where δ is the margin parameter and we set it to 0.5 in our experiments. The two terms in the loss function are equivalent in meaning, and they represent that the high attention region features in both the videos should be more similar than the high attention region feature in one video and the low attention region feature in the other video. The total loss for the entire training set may be represented as follows:

$$\mathcal{L}_{CASL} = \frac{1}{n_c} \sum_{j=1}^{n_c} \frac{1}{\binom{|\mathcal{S}_j|}{2}} \sum_{\boldsymbol{x}_m, \boldsymbol{x}_n \in \mathcal{S}_j} \mathcal{L}_j^{mn} \tag{3.10}$$

Optimization. The total loss function we need to optimize in order to learn the weights of the weakly supervised layer can be represented as follows:

$$\mathcal{L} = \lambda\mathcal{L}_{MILL} + (1 - \lambda)\mathcal{L}_{CASL} + \alpha||\boldsymbol{W}||_F^2 \tag{3.11}$$

where the weights to be learned in our network are lumped to $\boldsymbol{W}$. We use $\lambda = 0.5$ and $\alpha = 5 \times 10^{-4}$ in our experiments. We optimize the above loss function using Adam [94] with a batch size of 10. We create each batch in a way such that it has a minimum of three pairs of videos such that each pair has at least one category in common. We use a constant learning rate of 10^{-4} in all our experiments.

Classification and Localization. After learning the weights of the network, we use them to classify an untrimmed video as well as localize the activities in it during test time. Given a video, we obtain the category-wise confidence scores as in Eqn. 3.4 followed by softmax to obtain a pmf over the possible categories. Then, we can threshold the pmf to classify the video to contain one or more activity categories. However, as defined by the dataset [80] and used in literature [218], we use mAP for classification performance comparison, which does not require the thresholding operation, but directly uses the pmf.

For localization, we employ a two-stage thresholding scheme. First, we discard the categories which have a confidence score (Eqn. 3.4) below a certain threshold (0.0 used in our experiments). Thereafter, for each of the remaining categories, we apply a threshold on the corresponding activation in $\boldsymbol{\mathcal{A}}$ (Eqn. 3.2) along the temporal axis to obtain the localizations. It may be noted that as l_i is generally less than the frame rate of the videos, we upsample the activations to meet the frame rate.

3.4 Experiments

In this section, we experimentally evaluate the proposed framework for activity localization and classification from weakly labeled videos. We first discuss the datasets we use, followed by the implementation details, quantitative, and some qualitative results.

3.4.1 Datasets

We perform experimental analysis on two datasets namely ActivityNet v1.2 [70] and Thumos14 [80]. These two datasets contain untrimmed videos with frame-wise labels of activities occurring in the video. However, as our algorithm is weakly-supervised, we use only the activity tags associated with the videos.

ActivityNet1.2. This dataset has 4819 videos for training, 2383 videos for validation and 2480 videos for testing whose labels are withheld. The number of categories involved is 100, with an average of 1.5 temporal activity segments per video. As in literature [218, 132], we use the training videos to train our network, and the validation set to report the test performance.

Thumos14. The Thumos14 dataset has 1010 validation videos and 1574 test videos divided into 101 categories. Among these videos, 200 validation videos and 213 test videos have temporal annotations belonging to 20 categories. Although this is a smaller dataset than ActivityNet1.2, the temporal labels are very precise and with an average of 15.5 temporal activity segments per video. This dataset has several videos where multiple activities occur, thus making it even more challenging. The length of the videos also varies widely from a few seconds to more than an hour. The lower number of videos make it challenging to efficiently learn the weakly-supervised network. Following literature [218, 132], we use the validation videos for training and the test videos for testing.

Implementation Details. We use the corresponding repositories to extract the features for UntrimmedNets[1] and I3D[2]. We do not finetune the feature extractors. The weights of the weakly supervised layers are initialized by Xavier method [63]. We use TVL1 optical flow [3]. We train our network on a single Tesla K80 GPU using Tensorflow. We set $s = 8$ in Eqn. 3.3 for both the datasets.

Table 3.1: Detection performance comparisons on Thumos14. UNTF & I3DF are abbreviations for UntrimmedNet features and I3D features respectively. The symbol ⇣ represents that following [132], those models are trained using only the 20 categories with temporal annotations, but without these annotations.

Supervision	IoU →	0.1	0.2	0.3	0.4	0.5	0.7
Strong	Saliency-Pool [89]	04.6	03.4	02.1	01.4	00.9	00.1
	FV-DTF [135]	36.6	33.6	27.0	20.8	14.4	-
	SLM-mgram [152]	39.7	35.7	30.0	23.2	15.2	-
	S-CNN [177]	47.7	43.5	36.3	28.7	19.0	05.3
	Glimpse [231]	48.9	44.0	27.0	20.8	14.4	-
	PSDF [232]	51.4	42.6	33.6	26.1	18.8	-
	SMS [233]	51.0	45.2	36.5	27.8	17.8	-
	CDC [176]	-	-	40.1	29.4	23.3	**07.9**
	R-C3D [227]	54.5	51.5	44.8	35.6	28.9	-
	SSN [238]	**60.3**	**56.2**	**50.6**	**40.8**	**29.1**	-
Weak	HAS [185]	36.4	27.8	19.5	12.7	06.8	-
	UntrimmedNets [218]	44.4	37.7	28.2	21.1	13.7	-
	STPN (UNTF) [132] ⇣	45.3	38.8	31.1	23.5	16.2	05.1
	STPN (I3DF) [132] ⇣	52.0	44.7	35.5	25.8	16.9	04.3
Weak (Ours)	MILL+CASL+UNTF⇣	**49.0**	**42.8**	**32.0**	**26.0**	**18.8**	**06.2**
	MILL+I3DF	46.5	39.9	31.2	24.0	16.9	04.4
	MILL+CASL+I3DF	53.7	48.5	39.2	29.9	22.0	07.3
	MILL+CASL+I3DF⇣	**55.2**	**49.6**	**40.1**	**31.1**	**22.8**	**07.6**

[1] www.github.com/wanglimin/UntrimmedNet

[2] www.github.com/deepmind/kinetics-i3d

[3] www.github.com/yjxiong/temporal-segment-networks

Table 3.2: Detection performance comparisons over the ActivityNet1.2 dataset. The last column (Avg.) indicates the average mAP for IoU thresholds 0.5:0.05:0.95.

Supervision	IoU →	0.1	0.2	0.3	0.4	0.5	0.7	Avg.
Strong	SSN-SW [238]	-	-	-	-	-	-	24.8
	SSN-TAG [238]	-	-	-	-	-	-	**25.9**
Weak	W-TALC (Ours)	**53.9**	**49.8**	**45.5**	**41.6**	**37.0**	**14.6**	**18.0**

3.4.2 Activity Localization

We first perform a quantitative analysis of our framework for the task of activity localization. We use mAP with different Intersection over Union (IoU) thresholds as a performance metric, as followed in the literature [80]. We compare our results with several state-of-the-art methods on both strong and weak supervision in Table 3.1 and 3.2 for Thumos14 and ActivityNet1.2 respectively. We show results for different combinations of features and loss functions used. It may be noted that our framework performs much better than the other weakly supervised methods with similar feature usage. It is important to note that although the Kinetics pre-trained I3D features (I3DF) have some knowledge about activities, using only MILL as in [218] along with I3DF performs much worse than combining it with CASL. Moreover, our framework performs much better than other state-of-the-art methods even when using UNTF, which is not trained using any strong labels of activities. A detailed analysis of the two loss functions MILL and CASL will be presented subsequently.

3.4.3 Activity Classification

We now present the performance of our framework for activity classification. We use mean average precision (mAP) to compute the classification performance from the

Table 3.3: Classification performance comparisons on the Thumos14 dataset. † indicates that the algorithm uses both videos from Thumos14 and trimmed videos from UCF101 for training. Without † indicates that the algorithm uses only videos from Thumos14 for training.

Methods	mAP	Supervision
EMV + RGB [234]	61.5	Strong †
iDT+FV [216]	63.1	Strong †
iDT+CNN [217]	62.0	Strong †
Objects + Motion [84]	71.6	Strong †
Feat. Agg. [83]	71.0	Strong †
Extreme LM [208]	63.2	Strong †
TSN [219]	78.5	Strong †
Two Stream [181]	66.1	Strong †
TSN [219]	67.7	Strong
UntrimmedNets [218]	74.2	Weak
UntrimmedNets [218]	82.2	Weak †
W-TALC (Ours w. I3D)	**85.6**	Weak

Table 3.4: Classification performance comparisons on the ActivityNet1.2 dataset. † indicates that the algorithm uses the training and validation set of ActivityNet1.2 for training and tested on the server. Without † means that the algorithm is trained on the training set and tested on the validation set.

Algorithms	mAP	Supervision
C3D [199]	74.1	Strong †
iDT+FV [216]	66.5	Strong †
Depth2Action [84]	78.1	Strong †
TSN [219]	88.8	Strong †
Two Stream [181]	71.9	Strong †
TSN [219]	86.3	Strong
UntrimmedNets [218]	87.7	Weak
UntrimmedNets [218]	91.3	Weak †
W-TALC (Ours w. I3D)	**93.2**	Weak

predicted videos-level scores in Eqn. 3.4 after applying softmax. We compare with both fully supervised and weakly-supervised methods and the results are presented in Table 3.3 and 3.4 for Thumos14 and ActivityNet1.2 respectively. The proposed method performs significantly better than other state-of-the-art approaches. Please note that the methods indicated with † utilize a larger training set compared to ours as mentioned in the tables.

3.4.4 Ablation Study

Relative Weights on Loss Functions. In our framework, we jointly optimize two loss functions - MILL and CASL defined in Eqn. 3.11 to learn the weights of the weakly-supervised module. It is interesting to investigate the relative contributions of the loss

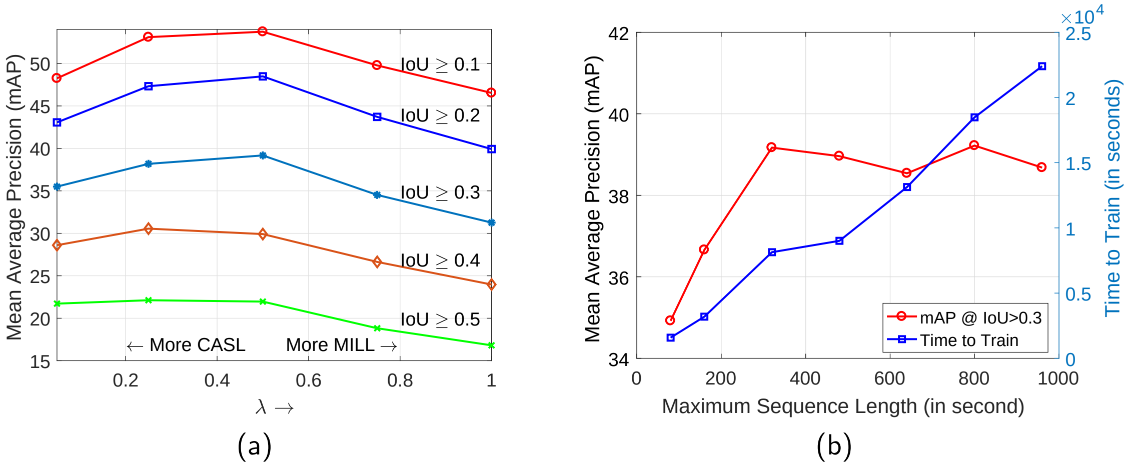


Figure 3.3: (a) presents the variations in detection performance on Thumos14 by changing weights on MILL and CASL. Higher λ represents more weight on the MILL and vice versa. (b) presents the variations in detection performance (@IoU $\geq$ 0.3) and training time on Thumos14 dataset by changing the maximum possible length of video sequence during training (T) as discussed in the text.

functions to the detection performance. In order to do that, we performed experiments, using the I3D features, with different values of λ (higher value indicate larger weight on MILL) and present the detection performance on the Thumos14 dataset in Fig. 3.3(a).

As may be observed from the plot, the proposed method performs best at $\lambda = 0.5$, i.e., when both the loss functions have equal weights. Moreover, using only MILL, i.e., $\lambda = 1.0$, results in a decrease of $7 - 8\%$ mAP compared to when both CASL and MILL are given equal weights in the loss function. This shows that the CASL introduced in this work has a major effect on the better performance of our framework compared to using I3D features along with the loss function in [218], i.e., MILL.

Sensitivity to Length of Sequence. Natural videos may often be very long. As mentioned previously, in the weakly-supervised setting, we have only video-level labels, so we need to process the entire video at once in order to compute the loss functions. In Section

3.3.1, we discuss a simple sampling strategy, which we use to maintain the length of the videos in a batch to be less than a pre-defined length T to meet GPU memory constraints. This method has the following advantages and disadvantages.

- *Advantages*: First, we can learn from long-length videos using this scheme. Secondly, this strategy will act as a data augmentation technique as we randomly crop, along the temporal axis to make it a fixed-length sequence, if the length of the video $\geq T$. Also, a lower value of T reduces computation time.

- *Disadvantage*: In this sampling scheme, errors will be introduced in the labels of the training batch, which may increase with the number of training videos with length $> T$.

The above factors induce a trade-off between performance and computation time. This can be seen in Figure 3.3(b), wherein the initial portion of the plot, with an increase of T, the detection performance improves, but the computational time increases. However, the detection performance eventually reaches a plateau suggesting $T = 320s$ to be a reasonable choice for this dataset.

Qualitative Results. We present a few interesting example localizations with ground truths in Fig. 3.4. The figure has four examples from Thumos14 and ActivityNet1.2 datasets. To test how the proposed framework performs on videos outside the aforementioned datasets, we tested the learned networks on randomly collected videos from YouTube. We present two such example detections in Fig. 3.4, using the model trained on Thumos14.

The first example in Fig. 3.4 is quite challenging as the localization should precisely be the portions of the video, where Golf Swing occurs, which has very similar features in the RGB domain to portions of the video where the player prepares for the swing. In spite

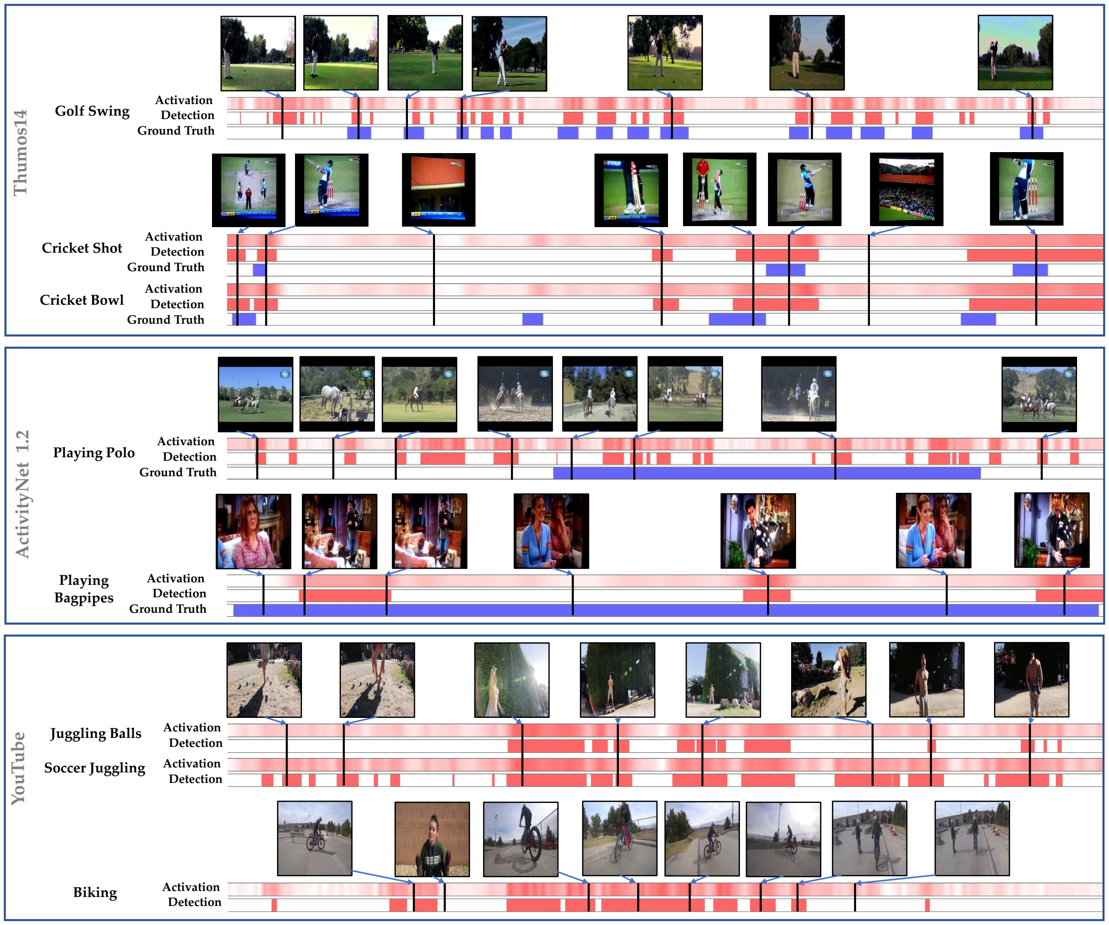


Figure 3.4: This figure presents some detection results for qualitative analysis on Thumos14, ActivityNet1.2 and a couple of random videos from YouTube

of this, our model is able to localize the relevant portions of Golf Swing, potentially based on the flow features. In the second example from Thumos14, the detections of Cricket Shot and Cricket Bowl appear to be correlated in time. This is because Cricket Shot and Bowl are two activities that generally co-occur in videos. To have fine-grained localization for such activities, videos that have only one of these activities are required. However, in the Thumos14 dataset, very few training examples contain only one of these two activities, which explains the behavior noted in the figure.

In the third example, which is from ActivityNet1.2, although ‘Playing Polo’ occurs in the first portion of the video, it is absent in the ground truth. However, our model is able to localize those activity segments as well. The same discussion is also applicable to the fourth example, where ‘Bagpiping’ occurs in the frames in a sparse manner, and our model’s response is aligned with its occurrence, but the ground truth annotations are for almost the entire video. These two examples are motivations behind weakly-supervised localization because obtaining precise unanimous ground truths from multiple labelers is difficult, costly, and sometimes even infeasible.

The fifth example is on a randomly selected video from YouTube. It has a person, who is juggling balls in an outdoor environment. But, most of the examples in Thumos14 of the same category are indoors, with the person taking up a significant portion of the frames spatially. Despite such differences in data, our model is able to localize some portions of the activity. However, the model also predicts some portions of the video to be ‘Soccer Juggling’, which may be because its training samples in Thumos14 contains a combination of feet, hand, and head, and a subset of such movements are present in ‘Juggling Balls’. Moreover, it is interesting to note that the first two frames show some maneuver of a ball with feet and it is detected as ‘Soccer Juggling’ as well.

3.5 Conclusions

In this chapter, we present an approach to learn temporal activity localization and video classification models using only weak supervision with video-level labels. We present the novel Co-Activity Similarity loss, which is empirically shown to be complementary with the Multiple Instance Learning Loss. We also show a simple mechanism to deal with long length videos, yet processing them at high granularity. Experiments on two challenging datasets demonstrate that the proposed method achieves state-of-the-art results in the weak TALC problem. In an extension of this work [124], we present a mechanism to learn a model for a text to video moment retrieval task using weak labels. An interesting future direction can be to explore the usefulness of co-video similarities for this type of task.

Chapter 4

Domain Adaptation of Semantic Segmentation using Weak Labels

4.1 Introduction

Unsupervised domain adaptation (UDA) methods for semantic segmentation have been developed to tackle the issue of domain gap. Existing methods aim to adapt a model learned on the source domain with pixel-wise ground truth annotations, e.g., from a simulator which requires the least annotation efforts, to the target domain that does not have any form of annotations. These UDA methods in the literature for semantic segmentation are developed mainly using two mechanisms: pseudo label self-training and distribution alignment between the source and target domains. For the first mechanism, pixel-wise pseudo labels are generated via strategies such as confidence scores [113, 79] or self-paced learning [245], but such pseudo-labels are specific to the target domain, and do not consider align-

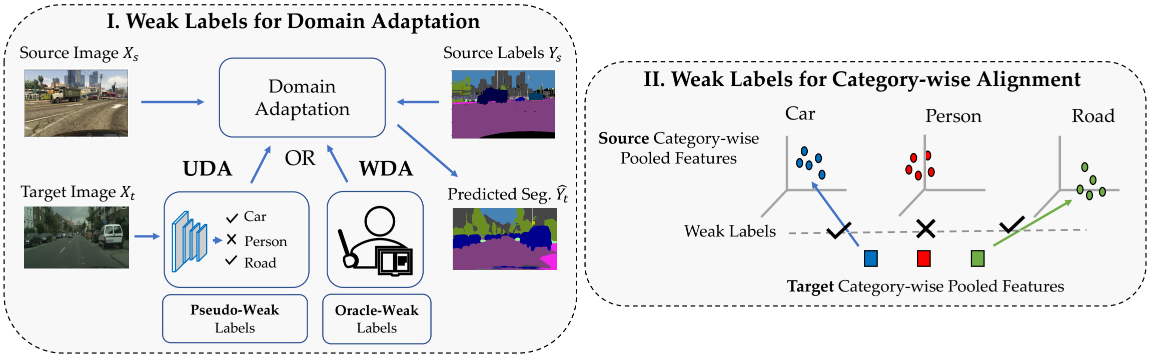

I. Weak Labels for Domain Adaptation
Source Image X_s
Domain Adaptation
Source Labels Y_s
UDA
OR
WDA
Target Image X_t
Car
Person
Road
Predicted Seg. $\hat{Y}_t$
Pseudo-Weak Labels
Oracle-Weak Labels
II. Weak Labels for Category-wise Alignment
Car
Person
Road
Source Category-wise Pooled Features
Weak Labels
Target Category-wise Pooled Features

Figure 4.1: Our work introduces two key ideas to adapt semantic segmentation models across domains. I: Using image-level weak annotations for domain adaptation, either estimated, i.e., pseudo-weak labels (Unsupervised Domain Adaptation, UDA) or acquired from a human oracle (Weakly-supervised Domain Adaptation (WDA). II: We utilize weak labels to improve the category-wise feature alignment between the source and target domains. ✓/✗ depicts weak labels, i.e., the categories present/absent in an image.

ment between domains. For the second mechanism, numerous spaces could be considered to operate the alignment procedure, such as pixel [74, 128], feature [75, 236], output [202, 33], and patch [203] spaces. However, alignment performed by these methods are agnostic to the category, which may be problematic as the domain gap may vary across categories.

To alleviate the issue of lacking annotations in the target domain, we propose [143] a concept of utilizing *weak labels* on the domain adaptation task for semantic segmentation, in the form of image- or point-level annotations in the target domain. Such weak labels can be used for category-wise alignment between the source and target domain, and also to enforce constraints on the categories present in an image. It is important to note that our weak labels could be estimated from the model prediction in the UDA setting, or provided by the human oracle in the weakly-supervised domain adaptation (WDA) paradigm (see left of weakda. 4.1). We are the first to introduce the WDA setting for semantic segmentation with image-level weak-labels, which is practically useful as collecting such annotations is much easier than pixel-wise annotations on the target domain. Benefiting from the concept of weak labels introduced in this chapter, we aim to utilize such weak labels to act as an enabler for the interplay between the alignment and pseudo labeling procedures, as they are much less noisy compared to pixel-wise pseudo labels. Specifically, we use weak labels to perform both 1) image-level classification to identify the presence/absence of categories in an image as a regularization, and 2) category-wise domain alignment using such categorical labels. For the image-level classification task, weak labels help our model obtain a better pixel-wise attention map per category. Then, we utilize the category-wise attention maps as the guidance to further pool category-wise features for proposed domain alignment procedure (right of Fig. 4.1).

Note that, although weak labels have been used in domain adaptation for object detection [81], our motivation is different from theirs. More specifically, [81] uses the weak labels to choose pseudo labels for self-training, while we formulate a general framework to learn from weak labels with different forms, i.e., UDA and WDA (image-level or point supervision), as well as to improve feature alignment across domains using weak labels.

We conduct experiments on the road scene segmentation problem from GTA5 [154] / SYNTHIA [156] to Cityscapes [39]. We perform extensive experiments to verify the usefulness of each component in the proposed framework, and show that our approach performs favorably against state-of-the-art algorithms for UDA. In addition, we show that our proposed method can be used for WDA and present its experimental results as a new benchmark. For the WDA setting, we also show that our method can incorporate various types of weak labels, such as image-level or point supervision. The **main contributions** of our work are: 1) we propose a concept of using weak labels to help domain adaptation for semantic segmentation; 2) we utilize weak labels to improve category-wise alignment for better feature space adaptation; and 3) we demonstrate that our method is applicable to both UDA and WDA settings.

4.2 Related Work

In this section, we discuss the literature of unsupervised domain adaptation (UDA) for image classification and semantic segmentation. In addition, we also discuss weakly-supervised methods for semantic segmentation.

UDA for Image Classification. The UDA task for image classification has been developed via aligning distributions across source and target domains. To this end, hand-crafted features [55, 64] and deep features [59, 205] have been considered to minimize the domain discrepancy and learn domain-invariant features. To further enhance the alignment procedure, maximum mean discrepancy [116] and adversarial learning [60, 206] based approaches have been proposed. Recently, several algorithms focus on improving deep models [117, 161, 106, 43], combining distance metric learning [187, 188], utilizing pixel-level adaptation [20, 200], or incorporating active learning [191].

UDA for Semantic Segmentation. Existing UDA methods in literature for semantic segmentation can be categorized primarily into to two groups: domain alignment and pseudo-label self-training. For domain alignment, numerous algorithms focus on aligning distributions in the pixel [26, 36, 74, 128, 225, 237], feature [31, 75, 236], and output [202, 33] spaces. For pseudo-label re-training, current methods [164, 245, 114] aim to generate pixel-wise pseudo labels on the target images, which is utilized to finetune the segmentation model trained on the source domain.

To achieve better performance, recent works [47, 113, 203, 213] attempt to combine the above two mechanisms. AdvEnt [213] adopts adversarial alignment and self-training in the entropy space, while BDL [113] combines output space and pixel-level adaptation with pseudo-label self-training in an iterative updating scheme. Moreover, Tsai et al. [203] propose a patch-level alignment method and show that their approach is complementary to existing modules such as output space adaptation and pseudo-label self-training. Similarly, Du et al. [47] integrate category-wise adversarial alignment with pixel-wise pseudo-labels, which may be noisy, leading to incorrect alignment. In addition, [47] needs to progressively change a ratio for selecting pseudo-labels, and the final performance is sensitive to this chosen parameter.

Compared to the above-mentioned approaches, we propose to exploit weak labels by learning an image classification task, while improving domain alignment through category-wise attention maps. Furthermore, we show that our approach can be utilized even in the case where oracle-weak labels are available on the target domain, in which case the performance will be further improved.

Weakly-supervised Semantic Segmentation. Since we are specifically interested in how weak labels can help domain adaptation, we also discuss the literature for weakly-supervised semantic segmentation, which has been tackled through different types of weak labels, such as image-level [2, 27, 96, 138, 146], video-level [32, 201, 240], bounding box [137, 42, 91], scribble [115, 209], and point [10] supervisions. Under this setting, these methods train the model using ground truth weak labels and perform testing in the same domain, which does not require domain adaptation. In contrast, we use a source domain with pixel-wise ground truth labels, but in the target domain, we consider pseudo-weak labels (UDA) or oracle-weak labels (WDA). As a result, we note that performance of weakly-supervised semantic segmentation methods which do not utilize any source domain, is usually much lower than the domain adaptation setting adopted in this chapter, e.g., the mean IoU on Cityscapes is only 24.9% as shown in [163].

4.3 Domain Adaptation with Weak Labels

In this section, we first introduce the problem and then describe details of the proposed framework - the image-level classification module and category-wise alignment method using weak labels. Finally, we present our method of obtaining the weak labels for the UDA and WDA settings.

Problem Definition. In the source domain, we have images and pixel-wise labels denoted as $\mathcal{I}_s = \{X_s^i, Y_s^i\}_{i=1}^{N_s}$. Whereas, our target dataset contains images and only image-level labels as $\mathcal{I}_t = \{X_t^i, y_t^i\}_{i=1}^{N_t}$. Note that $X_s, X_t \in \mathbb{R}^{H \times W \times 3}$, $Y_s \in \mathbb{B}^{H \times W \times C}$ with pixel-wise one-hot vectors, $y_t \in \mathbb{B}^C$ is a multi-hot vector representing the categories present in the image and C is the number of categories, same for both the source and target datasets. Such image-level labels y_t are often termed as weak labels. We can either estimate them, in which case we call them pseudo-weak labels (Unsupervised Domain Adptation, UDA) or acquire them from a human oracle that is called oracle-weak labels (Weakly-supervised Domain Adaptation, WDA). We will further discuss details of acquiring weak labels in

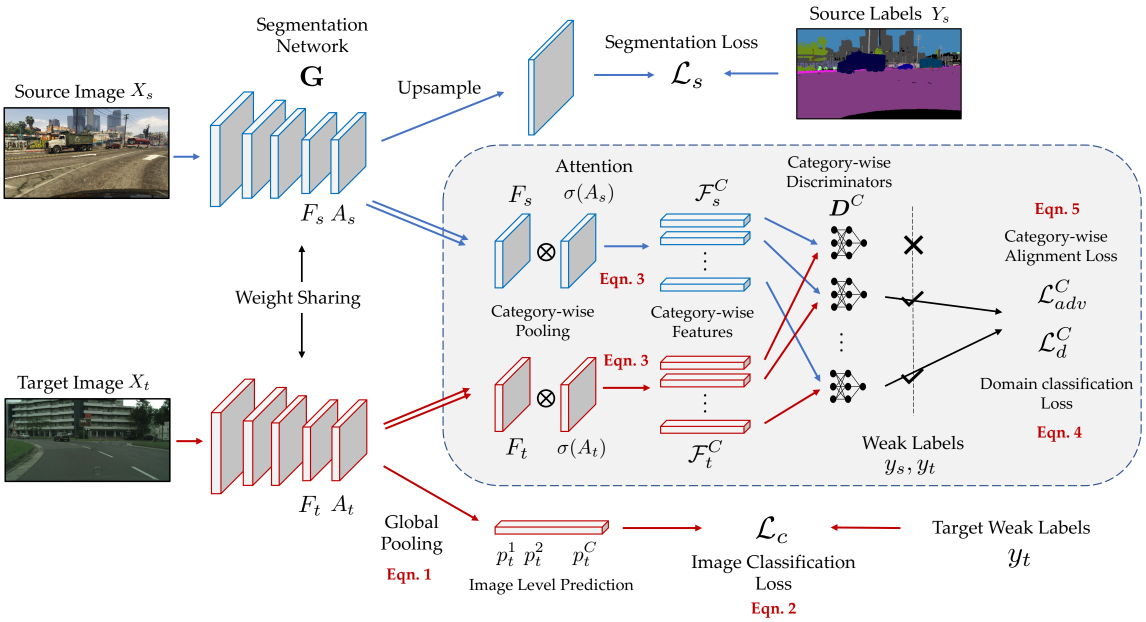


Figure 4.2: The proposed architecture consists of the segmentation network $\boldsymbol{G}$ and the weak label module. We compute the pixel-wise segmentation loss $\mathcal{L}_s$ for the source images and image classification loss $\mathcal{L}_c$ using the weak labels y_t for the target images. Note that the weak labels can be estimated as pseudo-weak labels or provided by a human oracle. We then use the output prediction A, convert it to an attention map $\sigma(A)$ and pool category-wise features $\mathcal{F}^C$. Next, these features are aligned between source and target domains using the category-wise alignment loss $\mathcal{L}_{adv}^C$ guided by the category-wise discriminators $\boldsymbol{D}^C$ learned via the domain classification loss $\mathcal{L}_d^C$.

Section 4.3.4. Given such data, the problem is to adapt a segmentation model $\boldsymbol{G}$ learned on the source dataset $\mathcal{I}_s$ to the target dataset $\mathcal{I}_t$.

Algorithm Overview. Fig. 4.2 presents an overview of our proposed method. We first pass both the source and target images through the segmentation network G and obtain their features $F_s, F_t \in \mathbb{R}^{H' \times W' \times 2048}$, segmentation predictions $A_s, A_t \in \mathbb{R}^{H' \times W' \times C}$, and the up-sampled pixel-wise predictions $O_s, O_t \in \mathbb{R}^{H \times W \times C}$. Note that $H'(< H), W'(< W)$ are the downsampled spatial dimensions of the image after passing through the segmentation network. As a baseline, we use the source pixel-wise annotations to learn $\boldsymbol{G}$, while aligning the output space distribution O_s and O_t, following [202].

In addition to having pixel-wise labels on the source data, we also have image-level weak labels on the target data. As discussed before, such weak labels can be either estimated (UDA) or acquired from an oracle (WDA). We then utilize these weak labels to update the segmentation network $\boldsymbol{G}$ in two different ways. First, we introduce a module which learns to predict the categories that are present in a target image. Second, we formulate a mechanism to align the features of each individual category between source and target domains. To this end, we use category-specific domain discriminators $\boldsymbol{D^c}$ guided by the weak labels to determine which categories should be aligned. In the following sections, we present these two modules in more detail.

4.3.1 Weak Labels for Category Classification

In order to predict whether a category is absent/present in a particular image, we define an image classification task using the weak labels, such that the segmentation network $\boldsymbol{G}$ can discover those categories. Specifically, we use the weak labels y_t and learn to predict the categories present/absent in the target images. We first feed the target images X_t through $\boldsymbol{G}$ to obtain the predictions A_t and then apply a global pooling layer to obtain a single vector of predictions for each category:

$$p_t^c = \sigma_s \left[\frac{1}{k} \log \frac{1}{H'W'} \sum_{h',w'} \exp kA_t^{(h',w',c)} \right], \tag{4.1}$$

where σ_s is the sigmoid function such that p_t represents the probability that a particular category appears in an image. Note that (4.1) is a smooth approximation of the `max` function. The higher the value of k, the better it approximates to `max`. We set $k = 1$ as we do not want the network to focus only on the maximum value of the prediction, which may be noisy, but also on other predictions that may have high values. Using p_t and the weak labels y_t, we can compute the category-wise binary cross-entropy loss:

$$\mathcal{L}_c(X_t; \boldsymbol{G}) = \sum_{c=1}^{C} -y_t^c \log(p_t^c) - (1 - y_t^c) \log(1 - p_t^c). \tag{4.2}$$

This is shown at the bottom stream of Fig. 4.2. This loss function $\mathcal{L}_c$ helps to identify the categories which are absent/present in a particular image and enforces the segmentation network $\boldsymbol{G}$ to pay attention to those objects/stuff that are partially identified when the source model is used directly on the target images.

4.3.2 Weak Labels for Feature Alignment

The classification loss using weak labels introduced in (4.2) regularizes the network focusing on certain categories. However, distribution alignment across the source and target domains is not considered yet. As discussed in the previous section, methods in literature either align feature space [75] or output space [202] across domains. However, such alignment is agnostic to the category, so it may align features of categories that are not present in certain images. Moreover, features belonging to different categories may have different domain gaps. Thereby, performing category-wise alignment could be beneficial but has not been widely studied in UDA for semantic segmentation. Although an existing work [47] attempts to align category-wise features, it utilizes pixel-wise pseudo labels, which may be noisy, and performs alignment in a high-dimensional feature space, which is not only difficult to optimize but also requires more computations.

To alleviate all the above issues, we use image-level weak labels to perform category-wise alignment in the feature space. Specifically, we obtain the category-wise features for each image via an attention map, i.e., segmentation prediction, guided by our classification module using weak labels, and then align these features between the source and target domains. We next discuss the category-wise feature pooling mechanism followed by the adversarial alignment technique.

Category-wise Feature Pooling. Given the last layer features F and the segmentation prediction A, we obtain the category-wise features by using the prediction as an attention over the features. Specifically, we obtain the category-wise feature $\mathcal{F}^c$ as a 2048-dimensional vector for the c^{th} category as follows:

$$\mathcal{F}^c = \sum_{h',w'} \sigma(A)^{(h',w',c)} F^{(h',w')}, \tag{4.3}$$

where $\sigma(A)$ is a tensor of dimension $H' \times W' \times C$, with each channel along the category dimension representing the category-wise attention obtained by the softmax operation σ over the spatial dimensions. As a result, $\sigma(A)^{(h',w',c)}$ is a scalar and $F^{(h',w')}$ is a 2048-dimensional vector, while $\mathcal{F}^c$ is the summed feature of $F^{(h',w')}$ weighted by $\sigma(A)^{(h',w',c)}$ over the spatial map $H' \times W'$. Note that we drop the subscripts s, t for source and target, as we employ the same operation to obtain the category-wise features for both domains. We next present the mechanism to align these features across domains. Note that we will use $\mathcal{F}^c$ to denote the pooled feature for the c^{th} category and $\mathcal{F}^C$ to denote the set of pooled features for all the categories. Category-wise feature pooling is shown in the middle of Fig. 4.2.

Category-wise Feature Alignment. To learn the segmentation network $\boldsymbol{G}$ such that the source and target category-wise features are aligned, we use an adversarial loss while using category-specific discriminators $\boldsymbol{D}^C = \{\boldsymbol{D}^c\}_{c=1}^C$. The reason of using category-specific discriminators is to ensure that the feature distribution for each category could be aligned independently, which avoids the noisy distribution modeling from a mixture of categories. In practice, we train C distinct category-specific discriminators to distinguish between category-wise features drawn from the source and target images. The loss function to train the discriminators $\boldsymbol{D}^C$ is as follows:

$$\mathcal{L}_d^C(\mathcal{F}_s^C, \mathcal{F}_t^C; \boldsymbol{D}^C) = \sum_{c=1}^{C} -y_s^c \log \boldsymbol{D}^c\big(\mathcal{F}_s^c\big) - y_t^c \log\big(1 - \boldsymbol{D}^c\big(\mathcal{F}_t^c\big)\big). \tag{4.4}$$

Note that, while training the discriminators, we only compute the loss for those categories which are present in the particular image via the weak labels $y_s, y_t \in \mathbb{B}^C$ that indicate whether a category occurs in an image or not. Then, the adversarial loss for the target images to train the segmentation network $\boldsymbol{G}$ can be expressed as follows:

$$\mathcal{L}_{adv}^C(\mathcal{F}_t^C; \boldsymbol{G}, \boldsymbol{D}^C) = \sum_{c=1}^{C} -y_t^c \log \boldsymbol{D}^c\big(\mathcal{F}_t^c\big). \tag{4.5}$$

Similarly, we use the target weak labels y_t to align only those categories present in the target image. By minimizing $\mathcal{L}_{adv}^C$, the segmentation network tries to fool the discriminator by maximizing the probability of the target category-wise feature being considered as drawn from the source distribution. These loss functions in (4.4) and (4.5) are obtained in the right of the middle box in Fig. 4.2.

4.3.3 Network Optimization

Discriminator Training. We learn a set of C distinct discriminators for each category c. We use the source and target images to train the discriminators, which learn to distinguish between the category-wise features drawn from either the source or the target domain. The optimization problem to train the discriminator can be expressed as: $\min_{\boldsymbol{D}^C} \mathcal{L}_d^C(\mathcal{F}_s^C, \mathcal{F}_t^C)$. Note that each discriminator is trained only with features pooled specific to that particular category. Therefore, given an image, we only update those discriminators corresponding to those categories which are present in the image and ignore the rest.

Segmentation Network Training. We train the segmentation network with the pixel-wise cross-entropy loss $\mathcal{L}_s$ on the source images, image classification loss $\mathcal{L}_c$ and adversarial loss $\mathcal{L}_{adv}^C$ on the target images. We combine these loss functions to learn $\mathbf{G}$ as follows :

$$\min_{\boldsymbol{G}} \mathcal{L}_s(X_s) + \lambda_c \mathcal{L}_c(X_t) + \lambda_d \mathcal{L}_{adv}^C(\mathcal{F}_t^C). \tag{4.6}$$

We follow the standard GAN training procedure [65] to alternatively update $\boldsymbol{G}$ and $\boldsymbol{D}^C$. Note that, computing $\mathcal{L}_{adv}^C$ involves the category-wise discriminators $\boldsymbol{D}^C$. Therefore, we fix $\boldsymbol{D}^C$ and backpropagate gradients only for the segmentation network $\boldsymbol{G}$.

4.3.4 Acquiring Weak Labels

In the above sections, we have proposed a mechanism to utilize image-level weak labels of the target images and adapt the segmentation model between source and target domains. In this section, we explain two methods to obtain such image-level weak labels.

Pseudo-Weak Labels (UDA). One way of obtaining weak labels is to directly estimate them using the data we have, i.e., source images/labels and target images, which is the unsupervised domain adaptation (UDA) setting. In this work, we utilize the baseline model [202] to adapt a model learned from the source to the target domain, and then obtain the weak labels of the target images as follows:

$$y_t^c = \begin{cases} 1, & \text{if } p_t^c > T, \\ 0, & \text{otherwise} \end{cases} \tag{4.7}$$

where p_t^c is the probability for category c as computed in (4.1) and T is a threshold, which we set to 0.2 in all the experiments unless specified otherwise. In practice, we compute the weak labels online during training and avoid any additional inference step. Specifically, we forward a target image, obtain the weak labels using (4.7), and then compute the loss functions in (4.6). As the weak labels obtained in this manner do not require human supervision, adaptation using such labels is unsupervised.

Oracle-Weak Labels (WDA). In this form, we obtain the weak labels by querying a human oracle to provide a list of the categories that occur in the target image. As we use supervision from an oracle on the target images, we refer to this as weakly-supervised domain adaptation (WDA). It is worth mentioning that the WDA setting could be practically useful, as collecting such human annotated weak labels is much easier than pixel-wise annotations. Also, there has not been any prior research involving this setting for domain adaptation.

To show that our method can use different forms of oracle-weak labels, we further introduce the point supervision as in [10], which only increases effort by a small amount compared to the image-level supervision. In this scenario, we randomly obtain one pixel coordinate of each category that belongs in the image, i.e., the set of tuples $\{(h^c, w^c, c) | \forall y_t^c = 1\}$. For an image, we compute the loss as follows: $\mathcal{L}_{point} = -\sum_{\forall y_t^c=1} y_t^c \log(O_t^{(h^c,w^c,c)})$, where $O_t \in \mathbb{R}^{H \times W \times C}$ is the output prediction of target after pixel-wise softmax.

4.4 Experimental Results

In this section, we perform an evaluation of our domain adaptation framework for semantic segmentation. We present the results for using both pseudo-weak labels, i.e., unsupervised domain adaptation (UDA) and human oracle-weak labels, i.e., weakly-supervised domain adaptation (WDA) and compare it with existing state-of-the-art methods. We also perform ablation studies to analyse the benefit of using pseudo/oracle-weak labels via our proposed weak-label classification module and category-wise alignment.

Datasets and Metric. We evaluate our domain adaptation method under the Sim-to-Real case with two different source-target scenarios. First, we adapt from GTA5 [154] to the Cityscapes dataset [39]. Second, we use SYNTHIA [156] as the source and Cityscapes as the target, which has a larger domain gap than the former case. For all experiments, we use the Intersection-over-Union (IoU) ratio as the metric. For SYNTHIA$\rightarrow$Cityscapes, following the literature [213], we report the performance averaged over 16 categories (listed in Table 4.2) and 13 categories (removing wall, fence and pole), which we denote as mIoU*.

Network Architectures. For the segmentation network $\boldsymbol{G}$, to have a fair comparison with works in literature, we use the DeepLab-v2 framework [29] with the ResNet-101 [69] architecture. We extract features F_s, F_t before the Atrous Spatial Pyramid Pooling (ASPP) layer. For the category-wise discriminators $D^C = \{D^c\}_{c=1}^C$, we use C separate networks, where each consists of three fully-connected layers, having number of nodes $\{2048, 2048, 1\}$ with ReLU activation.

Training Details. We implement our framework using PyTorch on a single Titan X GPU with 12G memory for all our experiments. We use the SGD method to optimize the segmentation network and the Adam optimizer [95] to train the discriminators. We set the initial learning rates to be 2.5×10^{-4} and 1×10^{-4} for the segmentation network and discriminators, with polynomial decay of power 0.9 [29]. As a common practice in weakly-supervised semantic segmentation [2], we use Dropout of 0.1 and 0.3 for oracle-weak labels and pseudo-weak labels respectively, on the spatial predictions before computing the loss $\mathcal{L}_c$. We choose λ_c to be 0.2 for oracle-weak labels and use a smaller $\lambda_c = 0.01$ for pseudo-weak labels to account for its inaccurate prediction. For the weight on the category-wise adversarial loss $\mathcal{L}_{adv}^C$, we set $\lambda_{adv} = 0.001$. For experiments using pseudo weak labels, to avoid noisy pseudo weak label prediction in the early training stage, we first train the segmentation baseline network using [202] for 60K iterations. Then, we include the proposed weak-label classification and alignment procedure, and train the entire framework.

4.4.1 Comparison with State-of-the-art Methods

Unsupervised Domain Adaptation (UDA). We compare our method with existing state-of-the-art UDA methods in Table 4.1 for GTA5→Cityscapes and in Table 4.2 for SYNTHIA→Cityscapes. Recent methods [26, 74, 113, 203] show that adapting images from source to target on the pixel level and then adding those translated source images in training enhances the performance. We follow this practice in the final model via adding these adapted images to the source dataset, as their pixel-wise annotations do not change after adaptation. Thus adaptation using weak labels aligns the features not only between the original source and target images, but also between the translated source images and the target images. We show that our method is also complementary to pixel-level adaptation. We also test our method with GTA5 as source and Foggy Cityscapes [162] as target. There is a parameter to choose the level of fog in the images, and we set that to 0.02 in our experiments. The results are presented in Table 4.3. We can observe consistent improvements as in other datasets.

All of the results presented till now are with ResNet-100 as the backbone architecture. We also test our framework on the VGG16 architecture and present the results in Table 4.4. Our method performs better than other UDA methods.

Discussions. In terms of applied techniques, e.g, pseudo-label re-training and domain alignment, the closest comparisons to our method are DISE [26], BDL [113], and Patch Space alignment [203]. We show that our method performs favorably against these approaches on both benchmarks. This can be attributed to our introduced concept of using weak labels, in which our UDA model explores pseudo-weak image-level labels, instead of using pixel-level pseudo-labels [203, 113] that may be noisy and degrade the performance. In addition, these methods do not perform domain alignment guided by such pseudo labels, whereas we use weak labels to enable our category-wise alignment procedure.

The only prior work that adopts category-wise feature alignment is SSF-DAN [47]. However, our method is different from theirs in three aspects: 1) We introduce the weak-

Table 4.1: Results of adapting GTA5 to Cityscapes. The top group is for UDA, while the bottom group presents our method's performance using the oracle-weak labels for WDA that use either image-level or point supervision.

	GTA5 → Cityscapes																		
Method	road	sidewalk	building	wall	fence	pole	light	sign	veg	terrain	sky	person	rider	car	truck	bus	train	mbike	bike	mIoU
No Adapt.	75.8	16.8	77.2	12.5	21.0	25.5	30.1	20.1	81.3	24.6	70.3	53.8	26.4	49.9	17.2	25.9	6.5	25.3	36.0	36.6
Road [33]	76.3	36.1	69.6	28.6	22.4	28.6	29.3	14.8	82.3	35.3	72.9	54.4	17.8	78.9	27.7	30.3	4.0	24.9	12.6	39.4
AdaptOutput [202]	86.5	25.9	79.8	22.1	20.0	23.6	33.1	21.8	81.8	25.9	75.9	57.3	26.2	76.3	29.8	32.1	7.2	29.5	32.5	41.4
AdvEnt [213]	89.4	33.1	81.0	26.6	26.8	27.2	33.5	24.7	83.9	36.7	78.8	58.7	30.5	84.8	**38.5**	44.5	1.7	31.6	32.4	45.5
CLAN [118]	87.0	27.1	79.6	27.3	23.3	28.3	35.5	24.2	83.6	27.4	74.2	58.6	28.0	76.2	33.1	36.7	6.7	**31.9**	31.4	43.2
SWD [106]	**92.0**	46.4	82.4	24.8	24.0	**35.1**	33.4	34.2	83.6	30.4	80.9	56.9	21.9	82.0	24.4	28.7	6.1	25.0	33.6	44.5
SSF-DAN [47]	90.3	38.9	81.7	24.8	22.9	30.5	37.0	21.2	**84.8**	38.8	76.9	58.8	30.7	**85.7**	30.6	38.1	5.9	28.3	36.9	45.4
DISE [26]	91.5	47.5	82.5	31.3	25.6	33.0	33.7	25.8	82.7	28.8	82.7	**62.4**	**30.8**	85.2	27.7	34.5	6.4	25.2	24.4	45.4
BDL [113]	91.4	47.9	**84.2**	**32.4**	26.0	31.8	37.3	33.0	83.3	**39.2**	79.2	57.7	25.6	81.3	36.3	39.7	2.6	31.3	33.5	47.2
AdaptPatch [203]	92.3	**51.9**	82.1	29.2	25.1	24.5	33.8	33.0	82.4	32.8	82.2	58.6	27.2	84.3	33.4	**46.3**	2.2	29.5	32.3	46.5
Ours (UDA)	91.6	47.4	84.0	30.4	**28.3**	31.4	**37.4**	**35.4**	83.9	38.3	**83.9**	61.2	28.2	83.7	28.8	41.3	**8.8**	24.7	**46.4**	**48.2**
Ours (WDA: Image)	89.5	54.1	83.2	31.7	34.2	37.1	43.2	39.1	85.1	39.6	85.9	61.3	34.1	82.3	42.3	51.9	34.4	33.1	45.4	53.0
Ours (WDA: Point)	94.0	62.7	86.3	36.5	32.8	38.4	44.9	51.0	86.1	43.4	87.7	66.4	36.5	87.9	44.1	58.8	23.2	35.6	55.9	56.4

label classification module to take advantage of image-level weak labels that enables an efficient feature alignment process and the novel WDA setting; 2) Our unified framework can be applied for both UDA and WDA settings with various types of supervisions; 3) Due to the introduced weak-label module, our category-wise feature alignment is operated in the pooled feature space in (4.3) guided by an attention map, rather than in a much higher-dimensional spatial space as in [47] that uses pixel-wise pseudo-labels. This essentially improves the training efficiency compared to [47], which requires a GPU with 16 GB memory as their discriminator needs much more computation time ($> 20\times$) and GPU memory ($> 8\times$) compared to our combined output space and category-wise discriminators. Also, the discriminators in [47] require 130 GFLOPS, whereas our discriminators require a total of only 0.5 GFLOPS.

4.4.2 Weakly-supervised Domain Adaptation (WDA)

Image-level Supervision. We present the results of our method when using oracle-weak labels (obtained from the ground truth of the training set) in the last rows of Table 4.1, 4.2, 4.3, 4.4. To the best of our knowledge, we are the first to work on WDA, i.e., using human oracle-weak labels on domain adaptation for semantic segmentation, and there are no other methods to compare against in the literature. From the results, it is interesting to note that the major boost in performance using WDA compared to UDA occurs for categories such as truck, bus, train, and motorbike for both cases using GTA5 and SYNTHIA as the source domain. One reason is that those categories are most underrepresented in both the source and the target datasets. Thus, they are not predicted in most of the target images, but using the oracle-weak labels helps to identify them better.

Point Supervision. We introduce another interesting setting of point supervision as in [10], which adds only a slight increase of annotation time compared to the image-level supervision. We follow [10] and randomly sample one pixel per category in each target image as the supervision. Note that, all the details and the modules are the same during training in this setting. In Table 4.1 and 4.2, the results show that using point supervision improves performance ($3.4 - 6.6\%$) on both benchmarks compared to the image-level supervision. This shows that our method is a general framework that can be applied to the conventional UDA setting as well as the WDA setting using either image-level or point supervision, while all the settings achieve consistent performance gains.

Fig. 4.3(b) shows a comparison of annotation time v.s. performance for various levels of supervision. With low annotation cost in WDA cases, our model bridges the gap

Table 4.2: Results of adapting SYNTHIA to Cityscapes. The top group is for UDA, while the bottom group presents the WDA setting using oracle-weak labels. mIoU and mIoU* are averaged over 16 and 13 categories.

	SYNTHIA → Cityscapes																	
Method	road	sidewalk	building	wall	fence	pole	light	sign	veg	sky	person	rider	car	bus	mbike	bike	mIoU	mIoU*
No Adapt.	55.6	23.8	74.6	9.2	0.2	24.4	6.1	12.1	74.8	79.0	55.3	19.1	39.6	23.3	13.7	25.0	33.5	38.6
AdaptOutput [202]	79.2	37.2	78.8	10.5	0.3	25.1	9.9	10.5	78.2	80.5	53.5	19.6	67.0	29.5	21.6	31.3	39.5	45.9
AdvEnt [213]	85.6	42.2	79.7	8.7	0.4	25.9	5.4	8.1	80.4	84.1	57.9	23.8	73.3	36.4	14.2	33.0	41.2	48.0
CLAN [118]	81.3	37.0	80.1	-	-	-	16.1	13.7	78.2	81.5	53.4	21.2	73.0	32.9	22.6	30.7	-	47.8
SWD [106]	82.4	33.2	**82.5**	-	-	-	**22.6**	**19.7**	**83.7**	78.8	44.0	17.9	75.4	30.2	14.4	39.9	-	48.1
DADA [214]	89.2	44.8	81.4	6.8	0.3	26.2	8.6	11.1	81.8	84.0	54.7	19.3	79.7	**40.7**	14.0	38.8	42.6	49.8
SSF-DAN [47]	84.6	41.7	80.8	-	-	-	11.5	14.7	80.8	**85.3**	57.5	21.6	82.0	36.0	19.3	34.5	-	50.0
DISE [26]	91.7	53.5	77.1	2.5	0.2	**27.1**	6.2	7.6	78.4	81.2	55.8	19.2	**82.3**	30.3	17.1	34.3	41.5	48.8
AdaptPatch [203]	82.4	38.0	78.6	8.7	**0.6**	26.0	3.9	11.1	75.5	84.6	53.5	21.6	71.4	32.6	19.3	31.7	40.0	46.5
Ours (UDA)	**92.0**	**53.5**	80.9	**11.4**	0.4	21.8	3.8	6.0	81.6	84.4	**60.8**	**24.4**	80.5	39.0	**26.0**	**41.7**	**44.3**	**51.9**
Ours (WDA: Image)	92.3	51.9	81.9	21.1	1.1	26.6	22.0	24.8	81.7	87.0	63.1	33.3	83.6	50.7	33.5	54.7	50.6	58.5
Ours (WDA: Point)	94.9	63.2	85.0	27.3	24.2	34.9	37.3	50.8	84.4	88.2	60.6	36.3	86.4	43.2	36.5	61.3	57.2	63.7

in performance between UDA and full supervision ones (more results are shown in the supplementary material). Note that, other forms of weak labels such as object count and density can also be effective.

4.4.3 Ablation Study

Effect of Weak Labels. We show results for using both pseudo-weak labels as well as human oracle-weak labels. Table 4.5 and 4.6 present the results for different combinations of the modules used in our framework with and without pixel-level adaptation (PA) [74]. It is interesting to note that on GTA5→Cityscapes, even when using pseudo-weak labels, our method obtains a 4.2% boost in performance (41.4 → 45.6), as well as a $3-4\%$

Table 4.3: Results of adapting GTA5 to Foggy Cityscapes with ResNet101. The top group is for Unsupervised Domain Adaptation (UDA), while the bottom group presents our method's performance using the oracle-weak labels for Weakly-supervised Domain Adaptation (WDA) that use either image-level or point supervision.

	GTA5 → Cityscapes																			
Method	road	sidewalk	building	wall	fence	pole	light	sign	veg	terrain	sky	person	rider	car	truck	bus	train	mbike	bike	mIoU
No Adapt.	78.8	11.8	67.8	15.1	15.6	19.5	20.6	12.1	63.6	19.3	60.3	49.3	22.6	55.6	17.2	14.9	0.0	19.2	27.0	31.0
AdaptOutput [202]	87.3	24.9	70.2	15.4	18.7	19.6	24.9	18.6	**69.3**	28.2	64.4	49.5	24.1	74.0	**17.6**	21.2	2.1	**27.5**	35.9	36.5
Ours (UDA)	**88.8**	**27.8**	**71.0**	**21.7**	**21.8**	**26.4**	**33.1**	**26.2**	68.7	**29.4**	**66.3**	**55.4**	**27.2**	**77.1**	11.8	**24.0**	**5.7**	14.7	**39.3**	**38.8**
Ours (Image)	89.0	32.8	76.5	22.0	26.5	29.8	35.3	34.8	77.4	32.8	71.7	60.1	35.0	84.7	33.6	42.0	19.0	30.8	44.1	46.2
Ours (Point)	92.7	55.0	80.0	28.3	29.3	34.2	37.4	45.8	79.9	32.8	73.4	62.4	34.0	85.8	37.2	50.6	19.3	28.1	53.7	50.5

Table 4.4: Results of adapting GTA5 to Cityscapes with VGG16. The top group is for Unsupervised Domain Adaptation (UDA), while the bottom group presents our method's performance using the oracle-weak labels for Weakly-supervised Domain Adaptation (WDA) that use either image-level or point supervision.

	GTA5 → Cityscapes																			
Method	road	sidewalk	building	wall	fence	pole	light	sign	veg	terrain	sky	person	rider	car	truck	bus	train	mbike	bike	mIoU
AdaptOutput [202]	87.3	29.8	78.6	21.1	18.2	22.5	21.5	11.0	79.7	29.6	71.3	46.8	6.5	80.1	23.0	26.9	0.0	10.6	0.3	35.0
AdvEnt [213]	86.9	28.7	78.7	28.5	**25.2**	17.1	20.3	10.9	80.0	26.4	70.2	47.1	8.4	81.5	26.0	17.2	**18.9**	11.7	1.6	36.1
CLAN [118]	88.0	30.6	79.2	23.4	20.5	**26.1**	23.0	**14.8**	81.6	**34.5**	72.0	45.8	7.9	80.5	**26.6**	**29.9**	0.0	10.7	0.0	36.6
SSF-DAN [47]	**88.7**	32.1	**79.5**	29.9	22.0	23.8	21.7	10.7	80.8	29.8	**72.5**	49.5	16.1	**82.1**	23.2	18.1	3.5	**24.4**	8.1	37.7
AdaptPatch [203]	87.3	**35.7**	**79.5**	**32.0**	14.5	21.5	24.8	13.7	80.4	32.0	70.5	50.5	16.9	81.0	20.8	28.1	4.1	15.5	4.1	37.5
Ours (UDA)	87.1	**35.7**	78.6	24.9	22.7	21.8	**26.5**	11.7	**82.1**	32.1	70.4	**50.6**	**18.3**	77.4	21.7	24.6	7.6	16.3	**19.3**	**38.4**
Ours (Image)	88.0	46.8	81.6	22.3	35.2	27.4	29.2	27.0	82.4	35.4	80.7	57.1	29.0	83.2	38.0	56.4	23.3	29.8	5.5	46.2
Ours (Point)	93.6	62.7	81.4	29.6	33.7	30.7	29.7	38.2	81.5	43.0	81.7	54.3	28.8	83.8	42.9	52.5	38.4	27.1	49.8	51.8

boost for SYNTHIA→Cityscapes. In addition, as expected, using oracle-weak labels performs better than pseudo-weak labels by 6.5% on GTA5→Cityscapes and $6.5 - 7.3\%$ on SYNTHIA→Cityscapes. It it also interesting to note that using the category-wise alignment consistently improves the performance for all the cases, i.e., different types of weak labels and for different datasets.

Effect of Pseudo-Weak Label Threshold. We use a threshold T in (4.7) to convert

Table 4.5: Ablation of the proposed loss functions for GTA5→Cityscapes.

	Supervision	$\mathcal{L}_c$	$\mathcal{L}_{adv}^C$	PA	mIoU
UDA	No Adapt.				36.6
	Baseline [202]				41.4
	Pseudo-Weak	✓			44.2
		✓	✓		45.6
		✓		✓	46.7
		✓	✓	✓	**48.2**
WDA	Oracle-Weak	✓			50.8
		✓	✓		52.1
		✓		✓	52.0
		✓	✓	✓	**53.0**

Table 4.6: Ablation of the proposed loss functions for SYNTHIA→Cityscapes.

	Supervision	$\mathcal{L}_c$	$\mathcal{L}_{adv}^C$	PA	mIoU	mIoU*
UDA	No Adapt.				33.5	38.6
	Baseline [202]				39.5	45.9
	Pseudo-Weak	✓			41.7	49.0
		✓	✓		42.7	49.9
		✓		✓	43.0	50.6
		✓	✓	✓	**44.3**	**51.9**
WDA	Oracle-Weak	✓			47.8	56.0
		✓	✓		49.2	57.2
		✓		✓	49.8	57.8
		✓	✓	✓	**50.6**	**58.5**

the image-level prediction probability to a multi-hot vector denoting the pseudo-weak labels that indicates absence/presence of the categories. Note that the threshold is on a probability between 0 and 1. We then study the effect of T by varying it and plot the performance in Fig. 4.3(a) on GTA5→Cityscapes. The figure shows that our model generally works well with T in a range of 0.05 to 0.25. However, when we make T larger than 0.3, the performance starts to drop significantly, as in this case, the recall of the pseudo-weak labels would be very low compared with the oracle-weak labels (i.e., ground truths), which makes the segmentation network fail to predict most categories.

Output Space Visualization. We present some visualizations of the segmentation prediction probability for each category in Fig. 4.4. Before using any weak labels (third row), the probabilities may be low, even though there is a category present in that image. However, based on these initial predictions, our model can estimate the categories and then

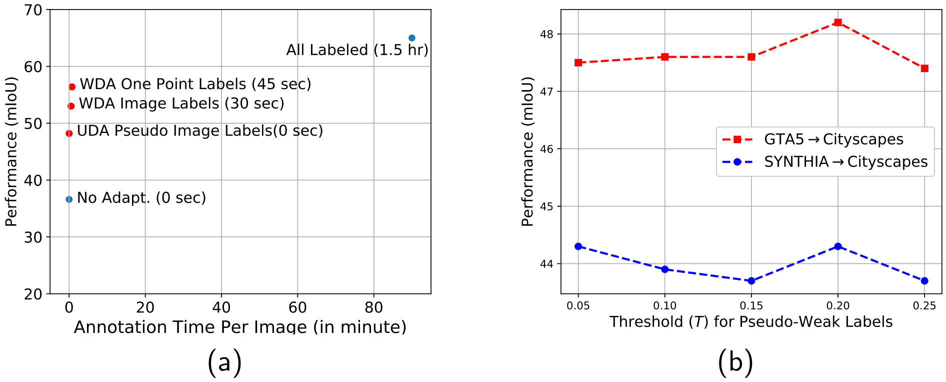


Figure 4.3: (a) Performance comparison on GTA5→Cityscapes with different levels of supervision on target images: no target labels ("No Adapt." and "UDA"), weak image labels (30 seconds), one point labels (45 seconds), and fully-supervised setting with all pixels labeled ("All Labeled") that takes 1.5 hours per image according to [39]. (b) Performance of our method on GTA5→Cityscapes with variations in the threshold, i.e., T in (4.7), for obtaining the pseudo-weak labels.

enforce their presence/absence explicitly in the proposed classification loss and alignment loss. The fourth row in Fig. 4.4 shows that such pseudo-weak labels help the network discover object/stuff regions towards better segmentation. For example, the fourth and fifth column shows that, although the original prediction probabilities are quite low, results using pseudo-weak labels are estimated correctly. Moreover, the last row shows that the predictions can be further improved when we have oracle-weak labels. Please refer to Appendix B for semantic segmentation visualizations.

4.5 Conclusions

In this chapter, we use weak labels to improve domain adaptation for semantic segmentation in both the UDA and WDA settings, with the latter being a novel setting. Specifically, we design an image-level classification module using weak labels, enforcing

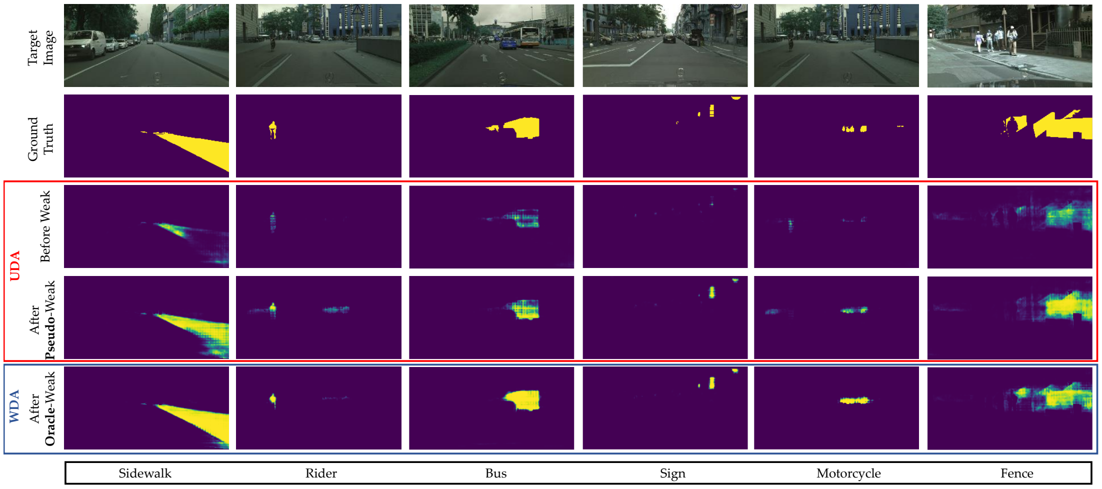


Figure 4.4: Visualizations of category-wise segmentation prediction probability before and after using the pseudo-weak labels on GTA5→Cityscapes. Before adaptation, the network only highlights the areas partially with low probability, while using the pseudo-weak labels helps the adapted model obtain much better segments, and is closer to the model using oracle-weak labels.

the network to pay attention to categories that are present in the image. With such a guidance from weak labels, we further utilize a category-wise alignment method to improve adversarial alignment in the feature space. Based on these two mechanisms, our formulation generalizes both to pseudo-weak and oracle-weak labels. We conduct extensive ablation studies to validate our approach against state-of-the-art UDA approaches.

Chapter 5

Learning from Trajectories via Subgoal Discovery

5.1 Introduction

Reinforcement Learning (RL) aims to take sequential actions while interacting with an environment, to maximize a certain pre-specified reward function, designed for the purpose of solving a task. RL using Deep Neural Networks (DNNs) has shown tremendous success in several tasks such as playing games [126, 180], solving complex robotics tasks [107, 48], etc. However, with sparse rewards, these algorithms often require a huge number of interactions with the environment, which is costly in real-world applications such as self-driving cars [18], and manipulations using real robots [107]. Manually designed dense reward functions could mitigate such issues, however, in general, it is difficult to design detailed reward functions for complex real-world tasks.

Imitation Learning (IL) using trajectories generated by an expert can potentially be used to learn the policies faster [6]. But, the performance of IL algorithms [158] are not only dependent on the performance of the expert providing the trajectories, but also on the state-space distribution represented by the trajectories, especially in case of high dimensional states. In order to avoid such dependencies on the expert, some methods in the literature [194, 34] take the path of combining RL and IL. However, these methods assume access to the expert value function, which may become impractical in real-world scenarios.

In this chapter, we present a strategy that starts with IL and then switches to RL. In the IL step, our framework performs supervised pre-training which aims at learning a policy that best describes the expert trajectories. However, due to the limited availability of expert trajectories, the policy trained with IL will have errors, which can then be alleviated using RL. Similar approaches are taken in [34] and [129], where the authors show that supervised pre-training does help to speed-up learning. However, note that the reward function in RL is still sparse, making it difficult to learn. With this in mind, we pose the following question: *can we make more efficient use of the expert trajectories, instead of just supervised pre-training?*

Given a set of trajectories, humans can quickly identify waypoints, which need to be completed in order to achieve the goal. We tend to break down the entire complex task into sub-goals and try to achieve them in the best order possible. Prior knowledge of humans helps to achieve tasks much faster [4, 49] than using only the trajectories for learning. The human psychology of divide-and-conquer has been crucial in several applications and it serves as a motivation behind our algorithm which learns to partition the state-space into sub-goals using expert trajectories. The learned sub-goals provide a discrete reward signal, unlike value-based continuous reward [131, 193], which can be erroneous, especially with a limited number of trajectories in long time horizon tasks. As the expert trajectories set may not contain all the states where the agent may visit during exploration in the RL step, we augment the sub-goal predictor via one-class classification to deal with such under-represented states. We perform experiments on three goal-oriented tasks on MuJoCo [197] with sparse terminal-only reward, which state-of-the-art RL, IL, or their combinations are not able to solve.

5.2 Related Works

Our work [145], is closely related to learning from demonstrations or expert trajectories as well as discovering sub-goals in complex tasks. We first discuss works on imitation learning using expert trajectories or reward-to-go. We also discuss the methods which aim to discover sub-goals, in an online manner during the RL stage from its past experience.

Imitation Learning. Imitation Learning [165, 178, 35, 148, 72] uses a set of expert trajectories or demonstrations to guide the policy learning process. A naive approach to use such trajectories is to train a policy in a supervised learning manner. However, such a policy would probably produce errors that grow quadratically with increasing steps. This can be alleviated using Behavioral Cloning (BC) algorithms [158, 157, 198], which queries expert action at states visited by the agent, after the initial supervised learning phase. However, such query actions may be costly or difficult to obtain in many applications. Trajectories are also used by [108], to guide the policy search, with the main goal of optimizing the return of the policy rather than mimicking the expert. Recently, some works [194, 25, 193] aim to combine IL with RL by assuming access to experts reward-to-go at every state visited by the RL agent. [34] take a moderately different approach where they switch from IL to RL and show that randomizing the switch point can help to learn faster. The authors in [149] use demonstration trajectories to perform skill segmentation in an Inverse Reinforcement Learning (IRL) framework. The authors in [127] also perform expert trajectory segmentation, but do not show results on learning the task, which is our main goal. SWIRL [100] makes certain assumptions on the expert trajectories to learn the reward function and their method is dependent on the discriminability of the state features, which we on the other hand learn end-to-end.

Learning with Options. Discovering and learning options have been studied in the literature [195, 147, 190] which can be used to speed-up the policy learning process. [179] developed a framework for planning based on options in a hierarchical manner, such that low-level options can be used to build higher-level options. [56] propose to learn a set of options, or skills, by augmenting the state space with a latent categorical skill vector. A separate network is then trained to learn a policy over options. The Option-Critic architecture [7] developed a gradient-based framework to learn the options along with learning the policy. This framework is extended in [155] to handle a hierarchy of options. [71] proposed a framework where the goals are generated using Generative Adversarial Networks (GAN) in a curriculum learning manner with increasingly difficult goals. Researchers have shown that an important way of identifying sub-goals in several tasks is identifying bottle-neck regions in tasks. Diverse Density [122], Relative Novelty [182], Graph Partitioning [183], clustering [120] can be used to identify such sub-goals. However, unlike our method, these algorithms do not use a set of expert trajectories, and thus would still be difficult to identify useful sub-goals for complex tasks with high sample-efficiency.

5.3 Methodology

We first provide a formal definition of the problem we are addressing in this chapter, followed by a brief overall methodology, and then present a detailed description of our framework.

Problem Definition. Consider a standard RL setting where an agent interacts with an environment which can be modeled by a Markov Decision Process (MDP) $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{P}, r, \gamma, \mathcal{P}_0)$, where $\mathcal{S}$ is the set of states, $\mathcal{A}$ is the set of actions, r is a scalar reward function, $\gamma \in [0, 1]$ is the discount factor and $\mathcal{P}_0$ is the initial state distribution. Our goal is to learn a policy $\pi_\theta(\boldsymbol{a}|\boldsymbol{s})$, with $\boldsymbol{a} \in \mathcal{A}$, which optimizes the expected discounted reward $\mathbb{E}_\tau[\sum_{t=0}^{\infty} \gamma^t r(\boldsymbol{s}_t, \boldsymbol{a}_t)]$, where $\tau = (\dots, \boldsymbol{s}_t, \boldsymbol{a}_t, r_t, \dots)$ and $\boldsymbol{s}_0 \sim \mathcal{P}_0$, $\boldsymbol{a}_t \sim \pi_\theta(\boldsymbol{a}|\boldsymbol{s}_t)$ and $\boldsymbol{s}_{t+1} \sim \mathcal{P}(\boldsymbol{s}_{t+1}|\boldsymbol{s}_t, \boldsymbol{a}_t)$.

With sparse rewards, optimizing the expected discounted reward using RL may be difficult. In such cases, it may be beneficial to use a set of state-action trajectories $\mathcal{D} = \{\{(\boldsymbol{s}_{ti}, \boldsymbol{a}^*_{ti})\}_{t=1}^{n_i}\}_{i=1}^{n_d}$ generated by an expert to guide the learning process. n_d is the number of trajectories in the dataset and n_i is the length of the i^{th} trajectory. We propose a methodology to efficiently use $\mathcal{D}$ by discovering sub-goals from these trajectories and use them to develop an extrinsic reward function.

Overall Methodology. Several complex, goal-oriented, real-world tasks can often be

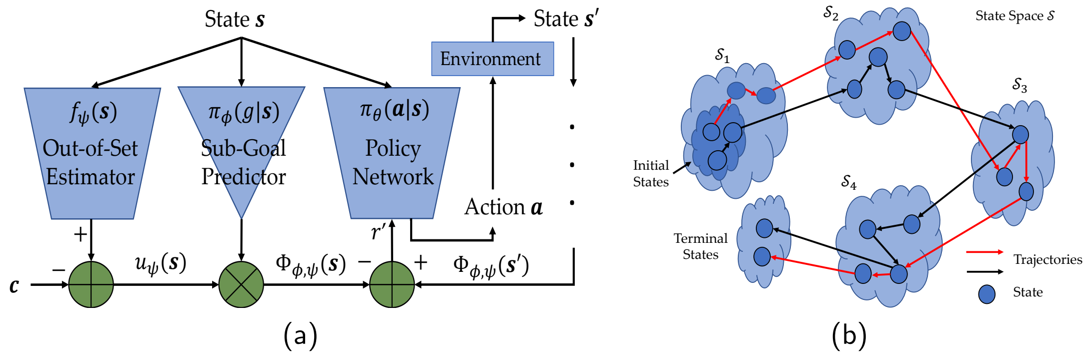


Figure 5.1: (a) This shows an overview of our proposed framework to train the policy network along with sub-goal based reward function with out-of-set augmentation. (b) An example state-partition with two independent trajectories in black and red. Note that the terminal state is shown as a separate state partition because we assume it to be indicated by the environment and not learned.

broken down into sub-goals with some natural ordering. Providing positive rewards after completing these sub-goals can help to learn much faster compared to sparse, terminal-only rewards. We advocate that such sub-goals can be learned directly from a set of expert demonstration trajectories, rather than manually designing them.

A pictorial description of our method is presented in Fig. 5.1(a). We use the set $\mathcal{D}$ to first train a policy using supervised learning. This serves a good initial point for policy search using RL. However, with sparse rewards, the search can still be difficult and the network may forget the learned parameters in the first step if it does not receive sufficiently useful rewards. To avoid this, we use $\mathcal{D}$ to learn a function $\pi_\phi(g|\boldsymbol{s})$, which given a state, predicts sub-goals. We use this function to obtain a new reward function, which intuitively informs the RL agent whenever it moves from one sub-goal to another. We also learn a utility function $u_\psi(\boldsymbol{s})$ to modulate the sub-goal predictions over the states which are not well-represented in the set $\mathcal{D}$. We approximate the functions π_θ, π_ϕ, and u_ψ using neural networks. Next, we define sub-goals and present an algorithm to learn them.

5.3.1 Sub-goal Definition

Definition 1. Consider that the state-space $\mathcal{S}$ is partitioned into sets of states as - $\{\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_{n_g}\}$, s.t., $\mathcal{S} = \cup_{i=1}^{n_g} \mathcal{S}_i$ and $\cap_{i=1}^{n_g} \mathcal{S}_i = \emptyset$ and n_g is the number of sub-goals specified by the user. For each $(\boldsymbol{s}, \boldsymbol{a}, \boldsymbol{s}')$, we say that the particular action takes the agent from one sub-goal to another iff $\boldsymbol{s} \in \mathcal{S}_i$, $\boldsymbol{s}' \in \mathcal{S}_j$ for some $i, j \in G = \{1, 2, \ldots, n_g\}$ and $i \neq j$.

We assume that there is an ordering in which groups of states appear in the trajectories as shown in Fig. 5.1(b). However, the states within these groups of states may appear in any random order in the trajectories. These groups of states are not defined a priori and our algorithm aims at estimating these partitions. Note that such orderings are natural in several real-world applications where a certain sub-goal can only be reached after completing one or more previous sub-goals. We show (empirically in Appendix A) that our assumption is soft rather than being strict, i.e., the degree by which the trajectories deviate from the assumption determines the granularity of the discovered sub-goals. We may consider that states in the trajectories of $\mathcal{D}$ appear in increasing order of sub-goal indices, i.e., achieving sub-goal j is harder than achieving sub-goal i $(i < j)$. This gives us a natural way of defining an extrinsic reward function, which would help towards faster policy search. Also, all the trajectories in $\mathcal{D}$ should start from the initial state distribution and end at the terminal states.

5.3.2 Learning Sub-Goal Prediction

We use $\mathcal{D}$ to partition the state-space into n_g sub-goals, with n_g being a hyperparameter. We learn a neural network to approximate $\pi_\phi(g|\boldsymbol{s})$, which given a state $\boldsymbol{s} \in \mathcal{S}$

predicts a probability mass function (p.m.f.) over the possible sub-goal partitions $g \in G$. The order in which the sub-goals occur in the trajectories, i.e., $\mathcal{S}_1 < \mathcal{S}_2 < \cdots < \mathcal{S}_{n_g}$, which can be derived from our assumption mentioned above, acts as a supervisory signal.

We propose an iterative framework to learn $\pi_\phi(g|\boldsymbol{s})$ using these ordered constraints. In the first step, we learn a mapping from states to sub-goals using equipartition labels among the sub-goals. Then we infer the labels of the states in the trajectories and correct them by imposing temporal ordering constraints. We use the new labels to again train the network and follow the same procedure until convergence. These two steps are as follows.

Learning Step. In this step, we consider that we have a set of tuples $(\boldsymbol{s}, g)$, which we use to learn the function π_ϕ. This can be posed as a multi-class classification problem with n_g categories. We optimize the following cross-entropy loss function,

$$\pi_\phi^* = \arg\min_{\pi_\phi} \frac{1}{N} \sum_{i=1}^{n_d} \sum_{t=1}^{n_i} \sum_{k=1}^{n_g} -\mathbf{1}\{g_{ti} = k\} \log \pi_\phi(g = k|\boldsymbol{s}_{ti}) \tag{5.1}$$

where $\mathbf{1}$ is the indicator function and N is the number of states in the dataset $\mathcal{D}$. To begin with, we do not have any labels g, and thus we consider equipartition of all the sub-goals in G along each trajectory. That is, given a trajectory of states $\{\boldsymbol{s}_{1i}, \boldsymbol{s}_{2i}, \ldots, \boldsymbol{s}_{n_i i}\}$ for some $i \in \{1, 2, \ldots, n_d\}$, the initial equi-partition sub-goals are,

$$g_{ti} = j, \quad \forall \lfloor \frac{(j-1)}{n_g} n_i \rfloor < t <= \lfloor \frac{j}{n_g} n_i \rfloor, \ j \in G \tag{5.2}$$

Using this initial labeling scheme, similar states across trajectories may have different labels, but the network is expected to converge at the Maximum Likelihood Estimate (MLE) of the entire dataset. We also optimize CASL [142] presented in Section 3.3.4. for stable learning as the initial labels can be erroneous. In the next iteration of the learning step, we use the inferred sub-goal labels, which we obtain as follows.

Inference Step. Although the equipartition labels in Eqn. 5.2 may have similar states across different trajectories mapped to dissimilar sub-goals, the learned network modeling π_ϕ maps similar states to the same sub-goal. But, Eqn. 5.1, and thus the predictions of π_ϕ do not account for the natural temporal ordering of the sub-goals. Even when using architectures such as Recurrent Neural Networks (RNN), it may be better to impose such temporal order constraints explicitly rather than relying on the network to learn them. We inject such order constraints using Dynamic Time Warping (DTW).

Formally, for the i^{th} trajectory in $\mathcal{D}$, we obtain the following set: $\{(\boldsymbol{s}_{ti}, \boldsymbol{\pi_\phi}(g|\boldsymbol{s}_{ti})\}_{t=1}^{n_i}$, where $\boldsymbol{\pi_\phi}$ is a vector representing the p.m.f. over the sub-goals G. However, as the predictions do not consider temporal ordering, the constraint that sub-goal j occurs after sub-goal i, for $i < j$, is not preserved. To impose such constraints, we use DTW between the two sequences $\{\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_{n_g}\}$, which are the standard basis vectors in the n_g dimensional Euclidean space and $\{\boldsymbol{\pi_\phi}(g|\boldsymbol{s}_{1i}), \boldsymbol{\pi_\phi}(g|\boldsymbol{s}_{2i}), \ldots, \boldsymbol{\pi_\phi}(g|\boldsymbol{s}_{n_i i})\}$. We use the $l1$-norm of the difference between two vectors as the similarity measure in DTW. In this process, we obtain a sub-goal assignment for each state in the trajectories, which become the new labels for training in the learning step.

We then invoke the learning step using the new labels (instead of Eqn. 5.2), followed by the inference step to obtain the next sub-goal labels. We continue this process until the number of sub-goal labels changed between iterations is less than a certain threshold.

This method is presented in Algorithm 2, where the superscript k represents the iteration number in learning-inference alternates.

Reward Using Sub-Goals. The ordering of the sub-goals, as discussed before, provides a natural way of designing a reward function as follows:

$$r'(\boldsymbol{s}, a, \boldsymbol{s}') = \gamma * \arg\max_{j \in G} \pi_\phi(g = j|\boldsymbol{s}') - \arg\max_{k \in G} \pi_\phi(g = k|\boldsymbol{s}) \tag{5.3}$$

where the agent in state $\boldsymbol{s}$ takes action a and reaches state $\boldsymbol{s}'$. The augmented reward function would become $r + r'$. Considering that we have a function of the form $\Phi_\phi(\boldsymbol{s}) = \arg\max_{j \in G} \pi_\phi(g = j|\boldsymbol{s})$, and without loss of generality that $G = \{0, 1, \ldots, n_g - 1\}$, so that for the initial state $\Phi_\phi(\boldsymbol{s}_0) = 0$, it follows from [131] that every optimal policy in $\mathcal{M}' = (\mathcal{S}, \mathcal{A}, \mathcal{P}, r + r', \gamma, \mathcal{P}_0)$, will also be optimal in $\mathcal{M}$, the original MDP. However, the new reward function may help to learn the task faster.

5.3.3 Out-of-Set Augmentation

In several applications, it might be the case that the trajectories only cover a small subset of the state space, while the agent, during the RL step, may visit states outside of the states in $\mathcal{D}$. The sub-goals estimated at these out-of-set states may be erroneous.

To alleviate this problem, we use a logical assertion on the potential function $\Phi_\phi(\boldsymbol{s})$ that the sub-goal predictor is confident only for states which are well-represented in $\mathcal{D}$, and not elsewhere. We learn a neural network to model a utility function $u_\psi : \mathcal{S} \to \mathbb{R}$, which given a state, predicts the degree by which it is seen in the dataset $\mathcal{D}$. To do this, we build upon Deep One-Class Classification [160], which performs well on the task of anomaly detection. Their idea is derived from Support Vector Data Description (SVDD) [196], which aims to find the smallest hypersphere enclosing the given data points with minimum error. Data points outside the sphere are then deemed as anomalous. We learn the parameters of u_ψ by optimizing the following function:

$$\psi^* = \arg\min_{\psi} \frac{1}{N} \sum_{i=1}^{n_d} \sum_{t=1}^{n_i} ||f_\psi(\boldsymbol{s}_{ti}) - \boldsymbol{c}||^2 + \lambda ||\psi||_2^2,$$

where $\boldsymbol{c} \in \mathbb{R}^m$ is a vector determined a priori [160], f is modeled by a neural network with parameters ψ, s.t. $f_\psi(\boldsymbol{s}) \in \mathbb{R}^m$. The second part is the $l2$ regularization loss with all the parameters of the network lumped to ψ. The utility function u_ψ can be expressed as follows:

$$u_\psi(\boldsymbol{s}) = ||f_\psi(\boldsymbol{s}) - \boldsymbol{c}||_2^2 \tag{5.4}$$

A lower value of $u_\psi(\boldsymbol{s})$ indicates that the state has been seen in $\mathcal{D}$. We modify the potential function $\Phi_\phi(\boldsymbol{s})$ and thus the extrinsic reward function, to incorporate the utility score as follows:

$$\begin{aligned} \Phi_{\phi,\psi}(\boldsymbol{s}) = \mathbf{1}\{u_\psi(\boldsymbol{s}) \leq \delta\} * \arg\max_{j \in G} \pi_\phi(g = j|\boldsymbol{s}), \\ r'(\boldsymbol{s}, a, \boldsymbol{s}') = &\gamma \Phi_{\phi,\psi}(\boldsymbol{s}') - \Phi_{\phi,\psi}(\boldsymbol{s}), \end{aligned} \tag{5.5}$$

where $\Phi_{\phi,\psi}$ denotes the modified potential function. It may be noted that as the extrinsic reward function is still a potential-based function [131], the optimality conditions between the MDP $\mathcal{M}$ and $\mathcal{M}'$ still hold as discussed previously.

Algorithm 2 Learning Sub-Goal Prediction

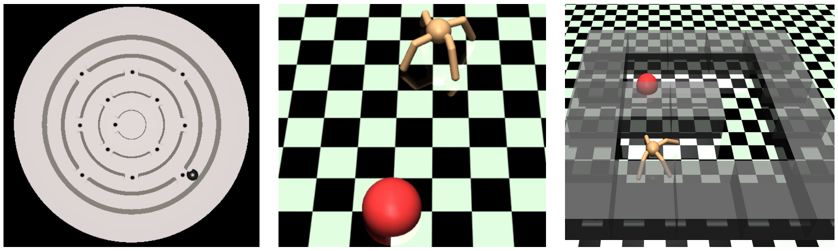
```
Input: Expert trajectory set $\mathcal{D}$
Output: Sub-goal predictor $\pi_\phi(g|\boldsymbol{s})$
$k \leftarrow 0$
Obtain $g^k$ for each $\boldsymbol{s} \in \mathcal{D}$ using Eqn. 5.2
repeat
    Optimize Eqn. 5.1 to obtain $\pi_\phi^k$
    Predict p.m.f of $G$ for each $\boldsymbol{s} \in \mathcal{D}$ using $\pi_\phi^k$
    Obtain new sub-goals $g^{k+1}$ using the p.m.f in DTW
    done = True, if $|g^k - g^{k+1}| < \epsilon$, else False
    $k \leftarrow k + 1$
until done is True
```

5.4 Supervised Pre-Training

As discussed previously, an initial way to utilize the trajectories is by pre-training the policy network π_θ using the trajectory set $\mathcal{D}$ in a supervised learning framework. We pre-train the network by optimizing the following:

$$\theta^* = \arg\min_\theta \sum_{i=1}^{n_d} \sum_{t=1}^{n_i} l(\pi_\theta(\boldsymbol{a}|\boldsymbol{s}_{ti}), \boldsymbol{a}_{ti}^*) + \lambda||\theta||_F^2 \tag{5.6}$$

where l is the loss function which can be cross-entropy or regression loss depending on discrete or continuous actions. Note that the continuous actions comprise of (μ, σ) which are parameters of a Gaussian distribution. The second part of Eqn. 5.6 is the $l2$ regularization loss. The policy obtained after optimizing Eqn. 5.6 possesses the ability to take actions with low error rates at the states sampled from the distribution induced by the trajectory set $\mathcal{D}$. However, a small error at the beginning would compound quadratically [158] with time as the agent starts visiting states which are not sampled from the distribution of $\mathcal{D}$. Algorithms like DAgger can be used to fine-tune the policy by querying expert actions at



(a) BiMGame (b) AntTarget (c) AntMaze

Figure 5.2: This figure presents the three environments we use - (a) Ball-in-Maze Game (BiMGame) (b) Ant locomotion in an open environment with an end goal (AntTarget) (c) Ant locomotion in a maze with an end goal (AntMaze).

states visited after executing the learned policy. This query to the expert is often very costly and even may not be feasible in some applications. More importantly, as DAgger aims to mimic the expert, it can only reach its performance and not better than that. For this reason, we fine-tune the policy using RL with the extrinsic reward function obtained after identifying the sub-goals.

5.5 Experiments

In this section, we perform an experimental evaluation of the proposed method of learning from trajectories and compare it with other state-of-the-art methods. We also perform ablation of different modules of our framework.

Tasks. We perform experiments on three challenging environments as shown in Fig. 5.2. The first environment is Ball-in-Maze Game (BiMGame) introduced in [207], where the task is to move a ball from the outermost to the innermost ring using a set of five discrete actions - clock-wise and anti-clockwise rotation by 1° along the two principal dimensions of the board and “no-op” where the current orientation of the board is maintained. The states are images of size 84 × 84. The second environment is AntTarget which involves the Ant [167]. The task is to reach the center of a circle of radius 5m with the Ant being initialized on a 45° arc of the circle. The state and action are continuous with 41 and 8 dimensions respectively. The third environment, AntMaze, uses the same Ant, but in a U-shaped maze used in [71]. The Ant is initialized on one end of the maze with the goal being the other end indicated as red in Fig. 5.2(c).

Network Architectures We follow the architecture of A3C [125] and share parameters between the policy and the state value estimation network. To model π_θ in BiMGame, we use a CNN with architecture Conv-Conv-FC-RNN followed by two heads: one for policy network and another for state value estimation. We append the previous step action as an additional input to the RNN step [123]. To model π_θ for AntTarget and AntMaze, we use the architecture FC-FC-FC-RNN, again followed by two heads for policy and state value estimation. For the policy, we predict the mean and standard deviation and sample actions from a Gaussian distribution. We use similar architectures (without RNN) for the respective tasks for π_ϕ and $f_\psi(s)$ with modifications in the final layer to suit their purpose.

Reward. For all tasks, we use sparse terminal-only reward, i.e., +1 only after reaching the goal state and 0 otherwise. Standard RL methods such as A3C [125] are not able to solve these tasks with such sparse rewards.

5.5.1 Trajectory Generation

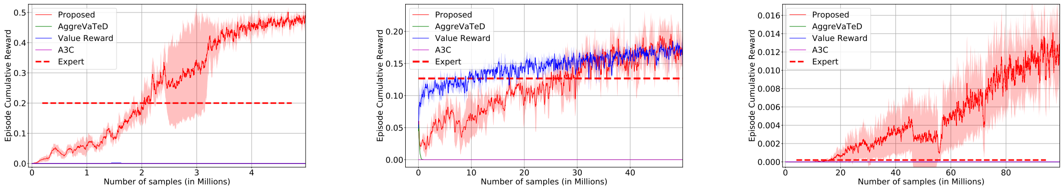
We generate trajectories from A3C [125] policies trained with dense reward, which we do not use in any other experiments. We also generate sub-optimal trajectories for BiMGame and AntMaze. To do so for AntMaze, we generate sub-optimal trajectories from an A3C policy stopped much before convergence. But for BiMGame, we use the simulator via Model Predictive Control (MPC) as in [144]. We leverage the internal physics engine of the simulator to forward propagate the state in time and generate trajectories by optimizing the cumulative reward function in a Model Predictive Control (MPC) manner. Formally, at time step t, we obtain the optimal action set $a^*_{t:t+H-1}$ from t to $t+H-1$ by solving the following:

$$\arg\max_{a_{t:t+H-1}} \sum_{t'=t}^{t+H-1} r(s_{t'}, a_{t'}, s_{t'+1}), \text{ s.t.}, s_{t'+1} = M(s_{t'}, a_{t'}), \tag{5.7}$$

where M is the simulator, $r(s_t, a_t, s_{t+1}) = d(s_{t+1}) - d(s_t)$ is the reward, $d(s_t)$ is the radial distance of the ball at time t from the center of the board, H is the horizon of optimization and $a_{t:t+H-1}$ is a set of actions. We only take the first action a^*_t, move to state s_{t+1} and repeat Eqn. 5.7. As we use a non-differentiable simulator, we employ a random shooting strategy [150] where we sample K sets of $a_{t:t+H-1}$ and choose the one which maximizes the rewards. We use $K, H = 10$ empirically. Note that the reward and the random shooting may not lead to the shortest path, thus making the trajectories sub-optimal. We generate around 400 trajectories for BiMGame and AntMaze, and 250 for AntTarget. As we generate two separate sets of trajectories for BiMGame and AntTarget, we use the sub-optimal set for all experiments, unless otherwise mentioned.

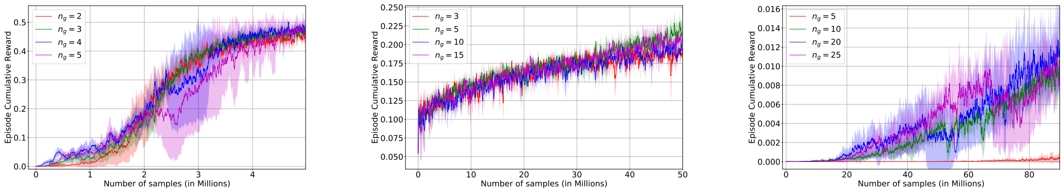
5.5.2 Comparison with Baselines

Baselines. We primarily compare our method with other RL methods which utilize trajectory or expert information - AggreVaTeD [194] and value-based reward shaping



(a) BiMGame (b) AntTarget (c) AntMaze

Figure 5.3: This figure shows the comparison of our proposed method with the baselines. Some lines may not be visible as they overlap. For tasks (a) and (c) our method clearly outperforms others. For task (b), although value reward initially performs better, our method eventually achieves the same performance. For a fair comparison, we do not use the out-of-set augmentation to generate these plots.



(a) BiMGame (b) AntTarget (c) AntMaze

Figure 5.4: (a) This figure presents the learning curves associated with a different number of learned sub-goals for the three tasks. For BiMGame and AntTarget, the number of sub-goals hardly matters. However, due to the inherently longer length of the task for AntMaze, a lower number of sub-goals such as $n_g = 5$ perform much worse than with higher n_g.

[131], equivalent to the $K = \infty$ in THOR [193]. For these methods, we use $\mathcal{D}$ to fit a value function to the sparse terminal-only reward of the original MDP $\mathcal{M}$ and use it as the expert value function. We also compare with standard A3C, but pre-trained using $\mathcal{D}$. It may be noted that we pre-train all the methods using the trajectory set to have a fair comparison. We report results with mean cumulative reward and $\pm\sigma$ over 3 independent runs.

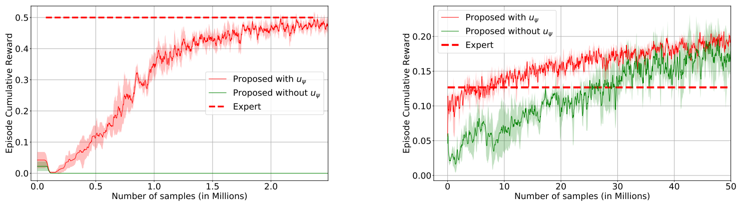
Comparison. First, we compare our method with other baselines in Fig 5.3. Note that as out-of-set augmentation using u_ψ can be applied for other methods that learn from trajectories, such as value-based reward shaping, we present the results for comparison with baselines without using u_ψ, i.e., Eqn. 5.3. Later, we perform an ablation study with and without using u_ψ. As may be observed, none of the baselines show any sign of learning for the tasks, except for ValueReward, which performs comparably with the proposed method for AntTarget only. Our method, on the other hand, is able to learn and solve the tasks consistently over multiple runs. The expert cumulative rewards are also drawn as straight lines in the plots and imitation learning methods like DAgger [158] can only reach that mark. Our method is able to surpass the expert for all the tasks. In fact, for AntMaze, even with a rather sub-optimal expert (an average cumulative reward of only 0.0002), our algorithm achieves about 0.012 cumulative reward at 100 million steps.

The poor performance of the ValueReward and AggreVaTeD can be attributed to the imperfect value function learned with a limited number of trajectories. Specifically, with an increase in the trajectory length, the variations in cumulative reward in the initial set of states are quite high. This introduces a considerable amount of error in the estimated value function in the initial states, which in turn traps the agent in some local optima when such value functions are used to guide the learning process.

5.5.3 Ablation Study

In this section we perform ablation study of different modules of our framework. We present ablation on the number of subgoals, effect of out-of-set augmentation, effect of sub-optimal expert and visualizations of the learned subgoals.

Variations in Sub-Goals. The number of sub-goals n_g is specified by the user, based on domain knowledge. For example, in the BiMGame, the task has four bottle-necks, which are states to be visited to complete the task and they can be considered as sub-goals. We



(a) BiMGame

(b) AntTarget

Figure 5.6: This plot presents the comparison of our proposed method for with and without using the one-class classification method for out-of-set augmentation.

perform experiments with different number of sub-goals and present the plots in Fig. 5.4. It may be observed that for BiMGame and AntTarget, our method performs well over a large variety of sub-goals. On the other hand for AntMaze, as the length of the task is much longer than AntTarget (12m vs 5m), $n_g \geq 10$ learn much faster than $n_g = 5$, as higher number of sub-goals provides more frequent rewards. Note that the variations in speed of learning with number of sub-goals is also dependent on the number of expert trajectories. If the pre-training is good, then less frequent sub-goals might work fine, whereas if we have a small number of expert trajectories, the RL agent may need more frequent reward (see Fig. 5.5).

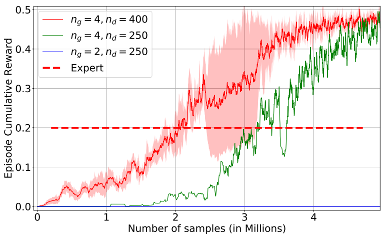
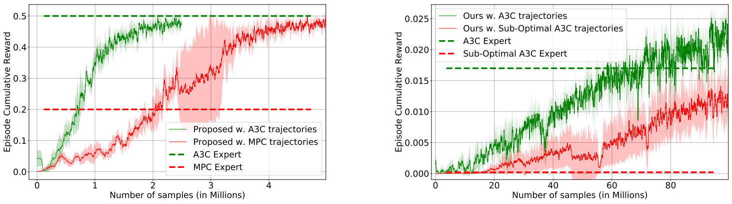


Figure 5.5: Effect of number of sub-goals and trajectories on BiMGame.

Effect of Out-of-Set Augmentation. The set $\mathcal{D}$ may not cover the entire state-space. To deal with this situation we developed the extrinsic reward function in Eqn. 5.5 using u_ψ. To evaluate its effectiveness we execute our algorithm using Eqn. 5.3 and Eqn. 5.5, and show the results in Fig. 5.6, with legends showing without and with u_ψ respectively. For



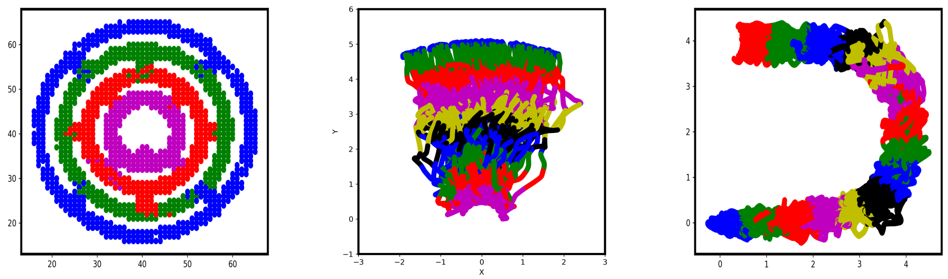
(a) BiMGame (b) AntMaze

Figure 5.7: This plot presents a comparison of our proposed method for two different types of expert trajectories. The corresponding expert rewards are also plotted as horizontal lines.

BiMGame, we used the optimal A3C trajectories, for this evaluation. This is because, using MPC trajectories with Eqn. 5.3 can still solve the task with similar reward plots, since MPC trajectories visit a lot more states due to its short-tem planning. The (optimal) A3C trajectories on the other hand, rarely visit some states, due to its long-term planning. In this case, using Eqn. 5.3 actually traps the agents to a local optimum (in the outermost ring), whereas using u_ψ as in Eqn. 5.5, learns to solve the task consistently (Fig. 5.6(a)).

For AntTarget in Fig. 5.6(b), using u_ψ performs better than without using u_ψ (and also surpasses value-based Reward Shaping). This is because the trajectories only span a small sector of the circle (Fig. 5.8(b)) while the Ant is allowed to visit states outside of it in the RL step. Thus, u_ψ avoids incorrect sub-goal assignments to states not well-represented in $\mathcal{D}$ and helps in the overall learning.

Effect of Sub-Optimal Expert. In general, the optimality of the expert may have an effect on performance. The comparison of our algorithm with optimal vs. sub-optimal expert trajectories are shown in Fig. 5.7. As may be observed, the learning curve for both the tasks is better for the optimal expert trajectories. However, in spite of using such sub-optimal experts, our method is able to surpass and perform much better than the experts.



(a) BiMGame $n_g = 4$ (b) AntTarget $n_g = 10$ (c) AntMaze $n_g = 15$

Figure 5.8: (a) This figure presents the learned sub-goals for the three tasks which are color-coded. Note that for (b) and (c), multiple sub-goals are assigned the same color, but they can be distinguished by their spatial locations.

We also see that our method performs better than even the optimal expert (as it is only *optimal* w.r.t. some cost function) used in AntMaze.

Visualization. We visualize the sub-goals discovered by our algorithm and plot it on the x-y plane in Fig. 5.8. As can be seen in BiMGame, with 4 sub-goals, our method is able to discover the bottle-neck regions of the board as different sub-goals. For AntTarget and AntMaze, the path to the goal is more or less equally divided into sub-goals. This shows that our method of sub-goal discovery can work for both environments with and without bottle-neck regions. (See Appendix A for more visualizations).

5.6 Discussions

The experimental analysis we presented in the previous section contain the following key observations:

- Our method to discover sub-goals works both for tasks with inherent bottlenecks (e.g. BiMGame) and without any bottlenecks (e.g. AntTarget and AntMaze), but with temporal order between groups of states in the expert demos, which occurs in many applications.

- Experiments show, that our assumption on the temporal ordering of groups of states in expert trajectories is soft, and determines the granularity of the discovered sub-goals (see Appendix A).
- Discrete rewards using sub-goals performs much better than value function based continuous rewards. Moreover, value functions learned from the along and limited number of trajectories may be erroneous, whereas segmenting the trajectories based on temporal ordering may still work well.
- As the expert trajectories may not cover the entire state-space regions the agent visits during exploration in the RL step, augmenting the sub-goal based reward function using out-of-set augmentation performs better compared to not using it.

5.7 Conclusion

In this chapter, we presented a framework to utilize the demonstration trajectories in an efficient manner by discovering sub-goals, which are waypoints that need to be completed in order to achieve a certain complex goal-oriented task. We use these sub-goals to augment the reward function of the task, without affecting the optimality of the learned policy. Experiments on three complex task show that unlike state-of-the-art RL, IL, or methods which combines them, our method is able to solve these tasks consistently. We also show that our method is able to perform much better than sub-optimal experts used to obtain the expert trajectories and at least as good as the optimal experts. Future work will concentrate on extending our method for repetitive non-goal oriented tasks.

Chapter 6

Conclusions

6.1 Thesis Summary

In this thesis, we focused on learning with limited supervision for a variety of computer vision tasks and a sequential decision making task. We explored two different dimensions of limited supervision - a limited number of labeled data points for classification tasks and a limited level of supervision for dense prediction tasks. In Chapter 2, we presented our algorithm for learning with a limited number of labeled data via active learning. However, in contrast to active learning in the literature which considers informativeness scores individually for the data points, we devised a framework that utilizes the contextual information often present in natural data. Our method is general enough to be applied to a variety of tasks where contextual information can be exploited. Experimental results on three different applications - object recognition, action recognition, and document classification showed that our method can significantly reduce the number of labeled samples.

In Chapter 3 and 4 we looked into the second dimension of learning with limited supervision, i.e., reducing the level of supervision from strong labels to weak labels for dense prediction tasks. In Chapter 3, we looked into the problem of weakly supervised action detection. We developed a framework, which can learn to localize action categories during test time while using only video-level categorical labels during training, compared to dense frame-wise labeling used in the literature. We framed the problem as Multiple Instance Learning with the pair-wise video feature similarity constraint, which empirically proved to be very effective for the overall performance of the framework. In Chapter 4, we looked into the problem of domain adaptation of semantic segmentation models with weak image-level labels. We showed that we can either estimate the weak labels, which would be Unsupervised Domain Adaptation (UDA), or we can obtain them from human annotators, which would be Weakly-supervised Domain Adaptation (WDA). We showed via an array of weak labels that our method brided the gap between UDA methods in literature and the fully-supervised model while incurring none to very low annotation cost.

We finally looked into sequential decision-making tasks in Chapter 5 where we developed a framework to learn from human demos in case of sparse terminal-only rewards. As imitation learning using human demonstrations can produce errors at states out of the distribution of the demonstrations, it may require an enormous amount of demonstrations to cover the state-space. We developed a method that learns from the demonstrations to divide the long complex task into subgoals which are much easier to solve. We used these learned subgoals as an extrinsic reward function in reinforcement learning to mitigate the errors learned in the first step via imitation learning. Results showed that our framework is able to solve the tasks that other methods in the literature are not able to solve. Continuing with the lines of works discussed in this thesis, next we discuss some interesting research directions for future works.

6.2 Future Research Directions

6.2.1 Integrating different levels of supervision with Human-in-the-loop

Data points vary widely in terms of complexity and the amount of information in them. If we choose the informative samples to label using active learning, we may not need to label some of the neighboring samples, thus leaving them unlabeled, but using them in the learning process. Moving a step further, it is also an interesting problem to choose which of the samples to label with strong vs weak supervision. This is because, as weak labels are for the entire bag and not individual elements within the bag, the learning algorithm requires some de-correlation in the label space [142]. In categories where natural de-correlation does not occur, e.g., biker always occurs with a bike, cricket-shot always appears with cricket bowling, we may need to label some samples via strong supervision. Then given a huge corpus of unlabeled data for some task, the problem is to choose which samples we should query the human oracle to label strongly, weakly, or leave unlabeled, given a certain budget for manual annotation. Our works on active learning in Chapter 2 and weakly supervised learning in Chapter 3 and 4 is a strong starting point for this problem.

6.2.2 Continuous Domain Adaptation

In our work on domain adaptation of semantic segmentation models in Chapter 4, we show how weak target labels can be effective in adaptation. However, in several real-world applications, there is often drift in data distribution with time. A model learned initially may not perform well after a certain period of time when the model is deployed in changing environments, e.g. changes in geographic location, weather, light conditions, demographics, and so on. Thus, we need our model to continuously adapt to the new changes in the environment. More importantly, the adaptation needs to be such that the knowledge about the past encountered environments is not forgotten, as the model might encounter those data points at a later stage. Our works on active learning in Chapter 2 and domain adaptation in Chapter 4 can be a strong starting point as for the new domains, we may need to select informative samples for manual annotations to mitigate large domain differences.

6.2.3 Learning from Interaction

We as humans learn about physical entities in our environment through a process of constant interaction in our daily lives. The actions we take while interacting play an important role in assigning and thus learning the categorical properties of the entities. Thus, our data acquisition (via interaction) and processing (learning properties about them), are correlated and it helps in better and much faster learning. However, this is not true for learning algorithms in the current literature, where the acquisition and processing pipelines are independent. It is an interesting research direction to correlate the acquisition and learning problems in a way where both can help each other. The recent advances in high fidelity simulators can play an important role to explore such research directions.

6.2.4 Commonsense in Learning

Human cognition combines visual perception along with commonsense and knowledge. While some commonsense, such as context information can be learned from the data, other more complex commonsense is often not a part of the visual data. Commonsense such as gravity pulls objects downwards, a pin is required to hang something onto the wall, a natural indoor/outdoor scene is almost always navigable, are a few examples of commonsense which is often not visible, but may help to learn, reason, infer and generalize about scenes in new domains with much lesser samples. Introducing such commonsense in learning algorithms is an interesting research direction beyond the context-based commonsense.

6.2.5 Adaptation of Policies

Similar to static tasks as recognition in computer vision, policies learned to solve dynamic tasks may not work well when there are changes in the environment. Such changes may be due to the shits in the distribution of the state space, changes in physical parameters that affect the transition probabilities between states, and so on. However, we as humans are able to quickly adapt to the changing environment around us with very little interaction with the new environment. Thus, an interesting research problem may be to find a principled approach to adapt policies to changes in environments with a much lower number of interactions with the new environment. Our work on domain adaptation of static tasks in computer vision in Chapter 4 can be a strong starting point, especially for adaptation to changes in state distribution with similar semantic meaning.

Bibliography

[1] Jake K Aggarwal and Michael S Ryoo. Human activity analysis: A review. *ACM Computing Surveys (CSUR)*, 43(3):16, 2011.

[2] Jiwoon Ahn and Suha Kwak. Learning pixel-level semantic affinity with image-level supervision for weakly supervised semantic segmentation. In *CVPR*, 2018.

[3] Mohamed R Amer and Sinisa Todorovic. Sum-product networks for modeling activities with stochastic structure. In *CVPR*, pages 1314–1321. IEEE, 2012.

[4] Jacob Andreas, Dan Klein, and Sergey Levine. Modular multitask reinforcement learning with policy sketches. In *ICML*, pages 166–175, 2017.

[5] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Netvlad: Cnn architecture for weakly supervised place recognition. In *CVPR*, pages 5297–5307, 2016.

[6] Brenna D Argall, Sonia Chernova, Manuela Veloso, and Brett Browning. A survey of robot learning from demonstration. *RAS*, 57(5):469–483, 2009.

[7] Pierre-Luc Bacon, Jean Harb, and Doina Precup. The option-critic architecture. In *AAAI*, pages 1726–1734, 2017.

[8] Jawadul H Bappy, Sujoy Paul, and Amit K Roy-Chowdhury. Online adaptation for joint scene and object classification. In *ECCV*, pages 227–243. Springer, 2016.

[9] Jawadul H Bappy, Sujoy Paul, Ertem Tuncel, and Amit K Roy-Chowdhury. The impact of typicality for informative representative selection. In *CVPR*. IEEE, 2017.

[10] Amy Bearman, Olga Russakovsky, Vittorio Ferrari, and Li Fei-Fei. What's the point: Semantic segmentation with point supervision. In *ECCV*, 2016.

[11] Amy Bearman, Olga Russakovsky, Vittorio Ferrari, and Li Fei-Fei. What's the point: Semantic segmentation with point supervision. In *ECCV*, pages 549–565. Springer, 2016.

[12] Hakan Bilen and Andrea Vedaldi. Weakly supervised deep detection networks. In *CVPR*, pages 2846–2854, 2016.

[13] Mustafa Bilgic and Lise Getoor. Link-based active learning. In *NIPS-Workshop*, 2009.

[14] Mustafa Bilgic, Lilyana Mihalkova, and Lise Getoor. Active learning for networked data. In *ICML*, pages 79–86, 2010.

[15] Piotr Bojanowski, Francis Bach, Ivan Laptev, Jean Ponce, Cordelia Schmid, and Josef Sivic. Finding actors and actions in movies. In *ICCV*, pages 2280–2287. IEEE, 2013.

[16] Piotr Bojanowski, Rémi Lajugie, Francis Bach, Ivan Laptev, Jean Ponce, Cordelia Schmid, and Josef Sivic. Weakly supervised action labeling in videos under ordering constraints. In *ECCV*, pages 628–643. Springer, 2014.

[17] Piotr Bojanowski, Rémi Lajugie, Edouard Grave, Francis Bach, Ivan Laptev, Jean Ponce, and Cordelia Schmid. Weakly-supervised alignment of video with text. In *ICCV*, pages 4462–4470. IEEE, 2015.

[18] Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, et al. End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*, 2016.

[19] Léon Bottou. Large-scale machine learning with stochastic gradient descent. In *COMPSTAT*, pages 177–186. Springer, 2010.

[20] Konstantinos Bousmalis, Nathan Silberman, David Dohan, Dumitru Erhan, and Dilip Krishnan. Unsupervised pixel-level domain adaptation with generative adversarial networks. In *CVPR*, 2017.

[21] Wenbin Cai, Ya Zhang, and Jun Zhou. Maximizing expected model change for active learning in regression. In *ICDM*, pages 51–60. IEEE, 2013.

[22] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, pages 4724–4733. IEEE, 2017.

[23] Shayok Chakraborty, Vineeth Balasubramanian, Qian Sun, Sethuraman Panchanathan, and Jieping Ye. Active batch selection via convex relaxations with guaranteed solution bounds. *TPAMI*, 37(10):1945–1958, 2015.

[24] Chih-Chung Chang and Chih-Jen Lin. Libsvm: a library for support vector machines. *TIST*, 2(3):27, 2011.

[25] Kai-Wei Chang, Akshay Krishnamurthy, Alekh Agarwal, Hal Daume III, and John Langford. Learning to search better than your teacher. *ICML*, 2015.

[26] Wei-Lun Chang, Hui-Po Wang, Wen-Hsiao Peng, and Wei-Chen Chiu. All about structure: Adapting structural information across domains for boosting semantic segmentation. In *CVPR*, 2019.

[27] Yu-Ting Chang, Qiaosong Wang, Wei-Chih Hung, Robinson Piramuthu, Yi-Hsuan Tsai, and Ming-Hsuan Yang. Weakly-supervised semantic segmentation via subcategory exploration. In *CVPR*, 2020.

[28] Lei Chen, Mengyao Zhai, and Greg Mori. Attending to distinctive moments: Weakly-supervised attention models for action localization in video. In *CVPR*, pages 328–336, 2017.

[29] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *CoRR*, abs/1606.00915, 2016.

[30] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *ICML*, 2020.

[31] Yi-Hsin Chen, Wei-Yu Chen, Yu-Ting Chen, Bo-Cheng Tsai, Yu-Chiang Frank Wang, and Min Sun. No more discrimination: Cross city adaptation of road scene segmenters. In *ICCV*, 2017.

[32] Yi-Wen Chen, Yi-Hsuan Tsai, Yen-Yu Lin, and Ming-Hsuan Yang. Vostr: Video object segmentation via transferable representations. *IJCV*, 02 2020.

[33] Yuhua Chen, Wen Li, and Luc Van Gool. Road: Reality oriented adaptation for semantic segmentation of urban scenes. In *CVPR*, 2018.

[34] Ching-An Cheng, Xinyan Yan, Nolan Wagener, and Byron Boots. Fast policy learning through imitation and reinforcement. *UAI*, 2018.

[35] Sonia Chernova and Manuela Veloso. Interactive policy learning through confidence-based autonomy. *JAIR*, 34:1–25, 2009.

[36] Jaehoon Choi, Taekyung Kim, and Changick Kim. Self-ensembling with gan-based data augmentation for domain adaptation in semantic segmentation. In *ICCV*, 2019.

[37] Myung Jin Choi, Joseph J Lim, Antonio Torralba, and Alan S Willsky. Exploiting hierarchical context on a large database of object categories. In *CVPR*, pages 129–136. IEEE, 2010.

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

[39] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016.

[40] Thomas M Cover and Joy A Thomas. *Elements of information theory*. John Wiley & Sons, 2012.

[41] Nguyen Viet Cuong, Wee Sun Lee, Nan Ye, Kian Ming A Chai, and Hai Leong Chieu. Active learning for probabilistic hypotheses using the maximum gibbs error criterion. In *NIPS*, pages 1457–1465, 2013.

[42] Jifeng Dai, Kaiming He, and Jian Sun. Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation. In *ICCV*, 2015.

[43] Shuyang Dai, Kihyuk Sohn, Yi-Hsuan Tsai, Lawrence Carin, and Manmohan Chandraker. Adaptation across extreme variations using unlabeled domain bridges. *arXiv preprint arXiv:1906.02238*, 2019.

[44] Filipe de Avila Belbute-Peres, Kevin Smith, Kelsey Allen, Josh Tenenbaum, and J Zico Kolter. End-to-end differentiable physics for learning and control. In *NeurIPS*, pages 7178–7189, 2018.

[45] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255. IEEE, 2009.

[46] Ali Diba, Vivek Sharma, Ali Pazandeh, Hamed Pirsiavash, and Luc Van Gool. Weakly supervised cascaded convolutional networks. *CVPR*, 2016.

[47] Liang Du, Jingang Tan, Hongye Yang, Jianfeng Feng, Xiangyang Xue, Qibao Zheng, Xiaoqing Ye, and Xiaolin Zhang. Ssf-dan: Separated semantic feature based domain adaptation network for semantic segmentation. In *ICCV*, 2019.

[48] Yan Duan, Xi Chen, Rein Houthooft, John Schulman, and Pieter Abbeel. Benchmarking deep reinforcement learning for continuous control. In *ICML*, pages 1329–1338, 2016.

[49] Rachit Dubey, Pulkit Agrawal, Deepak Pathak, Thomas L Griffiths, and Alexei A Efros. Investigating human priors for playing video games. *ICML*, 2018.

[50] Olivier Duchenne, Ivan Laptev, Josef Sivic, Francis Bach, and Jean Ponce. Automatic annotation of human actions in video. In *ICCV*, pages 1491–1498. IEEE, 2009.

[51] Thibaut Durand, Taylor Mordan, Nicolas Thome, and Matthieu Cord. Wildcat: Weakly supervised learning of deep convnets for image classification, pointwise localization and segmentation. In *CVPR*, 2017.

[52] Thibaut Durand, Nicolas Thome, and Matthieu Cord. Weldon: Weakly supervised learning of deep convolutional neural networks. In *CVPR*, pages 4743–4752, 2016.

[53] Ehsan Elhamifar, Guillermo Sapiro, Allen Yang, and S Shankar Sasrty. A convex optimization framework for active learning. In *ICCV*, pages 209–216. IEEE, 2013.

[54] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *PAMI*, 32(9):1627–1645, 2010.

[55] Basura Fernando, Amaury Habrard, Marc Sebban, and Tinne Tuytelaars. Unsupervised visual domain adaptation using subspace alignment. In *ICCV*, 2013.

[56] Carlos Florensa, Yan Duan, and Pieter Abbeel. Stochastic neural networks for hierarchical reinforcement learning. *ICLR*, 2017.

[57] Satoru Fujishige, Takumi Hayashi, and Shigueo Isotani. *The minimum-norm-point algorithm applied to submodular function minimization and linear programming*. Citeseer, 2006.

[58] Carolina Galleguillos, Andrew Rabinovich, and Serge Belongie. Object categorization using co-occurrence, location and appearance. In *CVPR*, pages 1–8. IEEE, 2008.

[59] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *ICML*, 2015.

[60] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. In *JMLR*, 2016.

[61] Ross Girshick. Fast r-cnn. In *ICCV*, 2015.

[62] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, pages 580–587. IEEE, 2014.

[63] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *AISTATS*, pages 249–256, 2010.

[64] Boqing Gong, Yuan Shi, Fei Sha, and Kristen Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *CVPR*, 2012.

[65] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NIPS*, 2014.

[66] Glenn Hartmann, Matthias Grundmann, Judy Hoffman, David Tsai, Vivek Kwatra, Omid Madani, Sudheendra Vijayanarasimhan, Irfan Essa, James Rehg, and Rahul Sukthankar. Weakly supervised learning of object segmentations from web-scale video. In *ECCVW*, pages 198–208. Springer, 2012.

[67] Mahmudul Hasan, Sujoy Paul, Anastasios I Mourikis, and Amit K Roy-Chowdhury. Context-aware query selection for active learning in event recognition. *T-PAMI*, 2018.

[68] Mahmudul Hasan and Amit K Roy-Chowdhury. Context aware active learning of activity recognition models. In *ICCV*, pages 4543–4551. IEEE, 2015.

[69] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[70] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *CVPR*, pages 961–970. IEEE, 2015.

[71] David Held, Xinyang Geng, Carlos Florensa, and Pieter Abbeel. Automatic goal generation for reinforcement learning agents. *ICML*, 2017.

[72] Todd Hester, Matej Vecerik, Olivier Pietquin, Marc Lanctot, Tom Schaul, Bilal Piot, Dan Horgan, John Quan, Andrew Sendonaris, Ian Osband, et al. Deep q-learning from demonstrations. In *AAAI*, 2018.

[73] Judy Hoffman, Saurabh Gupta, and Trevor Darrell. Learning with side information through modality hallucination. In *CVPR*, pages 826–834, 2016.

[74] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei A. Efros, and Trevor Darrell. Cycada: Cycle-consistent adversarial domain adaptation. In *ICML*, 2018.

[75] Judy Hoffman, Dequan Wang, Fisher Yu, and Trevor Darrell. Fcns in the wild: Pixel-level adversarial and constraint-based adaptation. *CoRR*, abs/1612.02649, 2016.

[76] Alex Holub, Pietro Perona, and Michael C Burl. Entropy-based active learning for object recognition. In *CVPR-Workshops*, pages 1–8. IEEE, 2008.

[77] Xia Hu, Jiliang Tang, Huiji Gao, and Huan Liu. Actnet: Active learning for networked texts in microblogging. In *SDM*, pages 306–314. SIAM, 2013.

[78] De-An Huang, Li Fei-Fei, and Juan Carlos Niebles. Connectionist temporal modeling for weakly supervised action labeling. In *ECCV*, pages 137–153. Springer, 2016.

[79] Wei-Chih Hung, Yi-Hsuan Tsai, Yan-Ting Liou, Yen-Yu Lin, and Ming-Hsuan Yang. Adversarial learning for semi-supervised semantic segmentation. In *BMVC*, 2018.

[80] Haroon Idrees, Amir R Zamir, Yu-Gang Jiang, Alex Gorban, Ivan Laptev, Rahul Sukthankar, and Mubarak Shah. The thumos challenge on action recognition for videos "in the wild". *CVIU*, 155:1–23, 2017.

[81] Naoto Inoue, Ryosuke Furuta, Toshihiko Yamasaki, and Kiyoharu Aizawa. Cross-domain weakly-supervised object detection through progressive domain adaptation. In *CVPR*, pages 5001–5009, 2018.

[82] Ashesh Jain, Amir R Zamir, Silvio Savarese, and Ashutosh Saxena. Structural-rnn: Deep learning on spatio-temporal graphs. *CVPR*, 2015.

[83] Mihir Jain, Jan van Gemert, Cees GM Snoek, et al. University of amsterdam at thumos challenge 2014. *ECCVW*, 2014, 2014.

[84] Mihir Jain, Jan C van Gemert, and Cees GM Snoek. What do 15,000 object categories tell us about classifying and localizing actions? In *CVPR*, pages 46–55, 2015.

[85] Zequn Jie, Yunchao Wei, Xiaojie Jin, Jiashi Feng, and Wei Liu. Deep self-taught learning for weakly supervised object localization. *CVPR*, 2017.

[86] Ajay J Joshi, Fatih Porikli, and Nikolaos P Papanikolopoulos. Scalable active learning for multiclass image classification. *TPAMI*, 34(11):2259–2273, 2012.

[87] Christoph Käding, Alexander Freytag, Erik Rodner, Andrea Perino, and Joachim Denzler. Large-scale active learning with approximations of expected model output changes. In *GCPR*, pages 179–191. Springer, 2016.

[88] Angjoo Kanazawa, David W Jacobs, and Manmohan Chandraker. Warpnet: Weakly supervised matching for single-view reconstruction. In *CVPR*, pages 3253–3261, 2016.

[89] Svebor Karaman, Lorenzo Seidenari, and Alberto Del Bimbo. Fast saliency based pooling of fisher encoded dense trajectories. In *ECCVW*, page 5, 2014.

[90] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.

[91] A. Khoreva, R. Benenson, J. Hosang, M. Hein, and B. Schiele. Simple does it: Weakly supervised instance and semantic segmentation. In *CVPR*, 2017.

[92] Anna Khoreva, Rodrigo Benenson, Jan Hosang, Matthias Hein, and Bernt Schiele. Simple does it: Weakly supervised instance and semantic segmentation. In *CVPR*, 2017.

[93] Anna Khoreva, Rodrigo Benenson, Mohamed Omran, Matthias Hein, and Bernt Schiele. Weakly supervised object boundaries. In *CVPR*, pages 183–192, 2016.

[94] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2014.

[95] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[96] Alexander Kolesnikov and Christoph H. Lampert. Seed, expand and constrain: Three principles for weakly-supervised image segmentation. In *ECCV*, 2016.

[97] Daphne Koller and Nir Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

[98] Hema Swetha Koppula, Rudhir Gupta, and Ashutosh Saxena. Learning human activities and object affordances from rgb-d videos. *IJRR*, 32(8):951–970, 2013.

[99] Andreas Krause. Sfo: A toolbox for submodular function optimization. *JMLR*, 11(Mar):1141–1144, 2010.

[100] Sanjay Krishnan, Animesh Garg, Richard Liaw, Brijen Thananjeyan, Lauren Miller, Florian T Pokorny, and Ken Goldberg. Swirl: A sequential windowed inverse reinforcement learning algorithm for robot tasks with delayed rewards. *IJRR*, 38(2-3):126–145, 2019.

[101] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1097–1105, 2012.

[102] Hilde Kuehne, Alexander Richard, and Juergen Gall. Weakly supervised learning of actions from transcripts. *CVIU*, 163:78–89, 2017.

[103] Ankit Kuwadekar and Jennifer Neville. Relational active learning for joint collective classification models. In *ICML*, pages 385–392, 2011.

[104] Agata Lapedriza, Hamed Pirsiavash, Zoya Bylinskii, and Antonio Torralba. Are all training examples equally valuable? *arXiv preprint arXiv:1311.6510*, 2013.

[105] Ivan Laptev, Marcin Marszalek, Cordelia Schmid, and Benjamin Rozenfeld. Learning realistic human actions from movies. In *CVPR*, pages 1–8. IEEE, 2008.

[106] Chen-Yu Lee, Tanmay Batra, Mohammad Haris Baig, and Daniel Ulbricht. Sliced wasserstein discrepancy for unsupervised domain adaptation. In *CVPR*, 2019.

[107] Sergey Levine, Chelsea Finn, Trevor Darrell, and Pieter Abbeel. End-to-end training of deep visuomotor policies. *JMLR*, 17(1):1334–1373, 2016.

[108] Sergey Levine and Vladlen Koltun. Guided policy search. In *ICML*, pages 1–9, 2013.

[109] Dong Li, Jia-Bin Huang, Yali Li, Shengjin Wang, and Ming-Hsuan Yang. Weakly supervised object localization with progressive domain adaptation. In *CVPR*, pages 3512–3520, 2016.

[110] Xianglin Li, Runqiu Guo, and Jun Cheng. Incorporating incremental and active learning for scene classification. In *ICMLA*, volume 1, pages 256–261. IEEE, 2012.

[111] Xin Li and Yuhong Guo. Adaptive active learning for image classification. In *CVPR*, pages 859–866, 2013.

[112] Xin Li and Yuhong Guo. Multi-level adaptive active learning for scene classification. In *ECCV*, pages 234–249. Springer, 2014.

[113] Yunsheng Li, Lu Yuan, and Nuno Vasconcelos. Bidirectional learning for domain adaptation of semantic segmentation. In *CVPR*, 2019.

[114] Qing Lian, Fengmao Lv, Lixin Duan, and Boqing Gong. Constructing self-motivated pyramid curriculums for cross-domain semantic segmentation: A non-adversarial approach. In *ICCV*, 2019.

[115] Di Lin, Jifeng Dai, Jiaya Jia, Kaiming He, and Jian Sun. Scribblesup: Scribble-supervised convolutional networks for semantic segmentation. In *CVPR*, 2016.

[116] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. Learning transferable features with deep adaptation networks. In *ICML*, 2015.

[117] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. Unsupervised domain adaptation with residual transfer networks. In *NIPS*, 2016.

[118] Yawei Luo, Liang Zheng, Tao Guan, Junqing Yu, and Yi Yang. Taking a closer look at domain shift: Category-level adversaries for semantics consistent domain adaptation. In *CVPR*, 2019.

[119] Oisin Mac Aodha, Neill Campbell, Jan Kautz, and Gabriel Brostow. Hierarchical subquery evaluation for active learning on a graph. In *CVPR*, pages 564–571. IEEE, 2014.

[120] Shie Mannor, Ishai Menache, Amit Hoze, and Uri Klein. Dynamic abstraction in reinforcement learning via clustering. In *ICML*, page 71. ACM, 2004.

[121] S Thomas McCormick. Submodular function minimization. *Handbooks in Operations Research and Management Science*, 12:321–391, 2005.

[122] Amy McGovern and Andrew G Barto. Automatic discovery of subgoals in reinforcement learning using diverse density. *ICML*, 2001.

[123] Piotr W. Mirowski, Razvan Pascanu, Fabio Viola, Hubert Soyer, Andrew J. Ballard, Andrea Banino, Misha Denil, Ross Goroshin, Laurent Sifre, Koray Kavukcuoglu, Dharshan Kumaran, and Raia Hadsell. Learning to navigate in complex environments. *ICLR*, 2017.

[124] Niluthpol Chowdhury Mithun, Sujoy Paul, and Amit K Roy-Chowdhury. Weakly supervised video moment retrieval from text queries. In *CVPR*, pages 11592–11601, 2019.

[125] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *ICML*, pages 1928–1937, 2016.

[126] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529, 2015.

[127] Adithyavairavan Murali, Animesh Garg, Sanjay Krishnan, Florian T Pokorny, Pieter Abbeel, Trevor Darrell, and Ken Goldberg. Tsc-dl: Unsupervised trajectory segmentation of multi-modal surgical demonstrations with deep learning. In *ICRA*, pages 4150–4157, 2016.

[128] Zak Murez, Soheil Kolouri, David Kriegman, Ravi Ramamoorthi, and Kyungnam Kim. Image to image translation for domain adaptation. In *CVPR*, 2018.

[129] Anusha Nagabandi, Gregory Kahn, Ronald S Fearing, and Sergey Levine. Neural network dynamics for model-based deep reinforcement learning with model-free fine-tuning. In *ICRA*, pages 7559–7566. IEEE, 2018.

[130] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *ICML*, pages 807–814, 2010.

[131] Andrew Y Ng, Daishi Harada, and Stuart Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *ICML*, volume 99, pages 278–287, 1999.

[132] Phuc Nguyen, Ting Liu, Gautam Prasad, and Bohyung Han. Weakly supervised action localization by sparse temporal pooling network. *CVPR*, 2018.

[133] Sangmin Oh, Anthony Hoogs, Amitha Perera, Naresh Cuntoor, Chia-Chih Chen, Jong Taek Lee, Saurajit Mukherjee, JK Aggarwal, Hyungtae Lee, Larry Davis, et al. A large-scale benchmark dataset for event recognition in surveillance video. In *CVPR*, pages 3153–3160. IEEE, 2011.

[134] Avital Oliver, Augustus Odena, Colin A Raffel, Ekin Dogus Cubuk, and Ian Goodfellow. Realistic evaluation of deep semi-supervised learning algorithms. In *NIPS*, pages 3235–3246, 2018.

[135] Dan Oneata, Jakob Verbeek, and Cordelia Schmid. The lear submission at thumos 2014. 2014.

[136] Rameswar Panda, Abir Das, Ziyan Wu, Jan Ernst, and Amit K Roy-Chowdhury. Weakly supervised summarization of web videos. In *ICCV*, pages 3657–3666, 2017.

[137] George Papandreou, Liang-Chieh Chen, Kevin Murphy, and Alan L Yuille. Weakly- and semi-supervised learning of a dcnn for semantic image segmentation. In *ICCV*, 2015.

[138] Deepak Pathak, Philipp Krahenbuhl, and Trevor Darrell. Constrained convolutional neural networks for weakly supervised segmentation. In *ICCV*, pages 1796–1804, 2015.

[139] Sujoy Paul, Jawadul H Bappy, and Amit K Roy-Chowdhury. Efficient selection of informative and diverse training samples with applications in scene classification. In *ICIP*, pages 494–498. IEEE, 2016.

[140] Sujoy Paul, Jawadul H Bappy, and Amit K Roy-Chowdhury. Non-uniform subset selection for active learning in structured data. In *CVPR*, pages 6846–6855, 2017.

[141] Sujoy Paul, Sourya Roy, and Amit K Roy-Chowdhury. Incorporating scalability in unsupervised spatio-temporal feature learning. In *ICASSP*, pages 1503–1507. IEEE, 2018.

[142] Sujoy Paul, Sourya Roy, and Amit K Roy-Chowdhury. W-talc: Weakly-supervised temporal activity localization and classification. In *ECCV*, pages 563–579, 2018.

[143] Sujoy Paul, Yi-Hsuan Tsai, Samuel Schulter, Amit K Roy-Chowdhury, and Manmohan Chandraker. Domain adaptive semantic segmentation using weak labels. *ECCV*, 2020.

[144] Sujoy Paul and Jeroen van Baar. Trajectory-based learning for ball-in-maze games. *NeurIPS-Workshop*, 2018.

[145] Sujoy Paul, Jeroen Vanbaar, and Amit Roy-Chowdhury. Learning from trajectories via subgoal discovery. In *NeurIPS*, pages 8411–8421, 2019.

[146] Pedro O. Pinheiro and Ronan Collobert. From image-level to pixel-level labeling with convolutional networks. In *CVPR*, 2015.

[147] Doina Precup. *Temporal abstraction in reinforcement learning.* University of Massachusetts Amherst, 2000.

[148] Aravind Rajeswaran, Vikash Kumar, Abhishek Gupta, Giulia Vezzani, John Schulman, Emanuel Todorov, and Sergey Levine. Learning complex dexterous manipulation with deep reinforcement learning and demonstrations. *RSS*, 2017.

[149] Pravesh Ranchod, Benjamin Rosman, and George Konidaris. Nonparametric bayesian reward segmentation for skill discovery using inverse reinforcement learning. In *IROS*, pages 471–477. IEEE, 2015.

[150] Anil V Rao. A survey of numerical methods for optimal control. *Advances in the Astronautical Sciences*, 135(1):497–528, 2009.

[151] Alexander Ratner, Stephen H Bach, Henry Ehrenberg, Jason Fries, Sen Wu, and Christopher Ré. Snorkel: Rapid training data creation with weak supervision. In *VLDB*, volume 11, page 269. NIH Public Access, 2017.

[152] Alexander Richard and Juergen Gall. Temporal action detection using a statistical language model. In *CVPR*, pages 3131–3140, 2016.

[153] Alexander Richard, Hilde Kuehne, and Juergen Gall. Weakly supervised action learning with rnn based fine-to-coarse modeling. *CVPR*, 2017.

[154] Stephan R. Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. Playing for data: Ground truth from computer games. In *ECCV*, 2016.

[155] Matthew Riemer, Miao Liu, and Gerald Tesauro. Learning abstract options. *NIPS*, 2018.

[156] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio M. Lopez. The SYNTHIA Dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *CVPR*, 2016.

[157] Stephane Ross and J Andrew Bagnell. Reinforcement and imitation learning via interactive no-regret learning. *arXiv preprint arXiv:1406.5979*, 2014.

[158] Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *AISTATS*, pages 627–635, 2011.

[159] Sourya Roy, Sujoy Paul, Neal E Young, and Amit K Roy-Chowdhury. Exploiting transitivity for learning person re-identification models on a budget. In *CVPR*, pages 7064–7072, 2018.

[160] Lukas Ruff, Nico Görnitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Robert Vandermeulen, Alexander Binder, Emmanuel Müller, and Marius Kloft. Deep one-class classification. In *ICML*, pages 4390–4399, 2018.

[161] Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada. Maximum classifier discrepancy for unsupervised domain adaptation. In *CVPR*, 2018.

[162] Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Semantic foggy scene understanding with synthetic data. *IJCV*, 126(9):973–992, 2018.

[163] Fatemeh Sadat Saleh, Mohammad Sadegh Aliakbarian, Mathieu Salzmann, Lars Petersson, and Jose M. Alvarez. Bringing background into the foreground: Making all classes equal in weakly-supervised video semantic segmentation. In *ICCV*, 2017.

[164] Fatemeh Sadat Saleh, Mohammad Sadegh Aliakbarian, Mathieu Salzmann, Lars Petersson, and Jose M. Alvarez. Effective use of synthetic data for urban scene semantic segmentation. In *ECCV*, 2018.

[165] Stefan Schaal. Is imitation learning the route to humanoid robots? *Trends in cognitive sciences*, 3(6):233–242, 1999.

[166] Mark Schmidt. Ugm: A matlab toolbox for probabilistic undirected graphical models, 2007.

[167] John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. *ICLR*, 2015.

[168] Prithviraj Sen and Lise Getoor. Link-based classification. *ICML*, 2003.

[169] Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad. Collective classification in network data. *AI magazine*, 29(3):93, 2008.

[170] Burr Settles. Active learning literature survey. *University of Wisconsin, Madison*, 52(55-66):11, 2010.

[171] Burr Settles. Active learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 6(1):1–114, 2012.

[172] Burr Settles and Mark Craven. An analysis of active learning strategies for sequence labeling tasks. In *EMNLP*, pages 1070–1079. Association for Computational Linguistics, 2008.

[173] Zhiqiang Shen, Jianguo Li, Zhou Su, Minjun Li, Yurong Chen, Yu-Gang Jiang, and Xiangyang Xue. Weakly supervised dense video captioning. In *CVPR*, volume 2, page 10, 2017.

[174] Lixin Shi, Yuhang Zhao, and Jie Tang. Batch mode active learning for networked data. *TIST*, 3(2):33, 2012.

[175] Zhiyuan Shi, Parthipan Siva, and Tao Xiang. Transfer learning by ranking for weakly supervised object annotation. *BMVC*, 2012.

[176] Zheng Shou, Jonathan Chan, Alireza Zareian, Kazuyuki Miyazawa, and Shih-Fu Chang. Cdc: convolutional-de-convolutional networks for precise temporal action localization in untrimmed videos. In *CVPR*, pages 1417–1426. IEEE, 2017.

[177] Zheng Shou, Dongang Wang, and Shih-Fu Chang. Temporal action localization in untrimmed videos via multi-stage cnns. In *CVPR*, pages 1049–1058, 2016.

[178] David Silver, James Bagnell, and Anthony Stentz. High performance outdoor navigation from overhead data using imitation learning. *RSS*, 2008.

[179] David Silver and Kamil Ciosek. Compositional planning using optimal option models. *arXiv preprint arXiv:1206.6473*, 2012.

[180] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587):484, 2016.

[181] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *NIPS*, pages 568–576, 2014.

[182] Özgür Şimşek and Andrew G Barto. Using relative novelty to identify useful temporal abstractions in reinforcement learning. In *ICML*, page 95. ACM, 2004.

[183] Özgür Şimşek, Alicia P Wolfe, and Andrew G Barto. Identifying useful subgoals in reinforcement learning by local graph partitioning. In *ICML*, pages 816–823. ACM, 2005.

[184] Avi Singh, Larry Yang, and Sergey Levine. Gplac: Generalizing vision-based robotic skills using weakly labeled images. *ICCV*, 2017.

[185] Krishna Kumar Singh and Yong Jae Lee. Hide-and-seek: Forcing a network to be meticulous for weakly-supervised object and action localization. In *ICCV*, 2017.

[186] Parthipan Siva and Tao Xiang. Weakly supervised action detection. In *BMVC*, volume 2, page 6, 2011.

[187] Kihyuk Sohn, Sifei Liu, Guangyu Zhong, Xiang Yu, Ming-Hsuan Yang, and Manmohan Chandraker. Unsupervised domain adaptation for face recognition in unlabeled videos. In *ICCV*, 2017.

[188] Kihyuk Sohn, Wenling Shang, Xiang Yu, and Manmohan Chandraker. Unsupervised domain adaptation for distance metric learning. In *ICLR*, 2019.

[189] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *JMLR*, 15(1):1929–1958, 2014.

[190] Martin Stolle and Doina Precup. Learning options in reinforcement learning. In *SARA*, pages 212–223. Springer, 2002.

[191] Jong-Chyi Su, Yi-Hsuan Tsai, Kihyuk Sohn, Buyu Liu, Subhransu Maji, and Manmohan Chandraker. Active adversarial domain adaptation. In *WACV*, 2020.

[192] Waqas Sultani and Mubarak Shah. What if we do not have multiple videos of the same action?–video action localization using web images. In *CVPR*, pages 1077–1085, 2016.

[193] Wen Sun, J Andrew Bagnell, and Byron Boots. Truncated horizon policy search: Combining reinforcement learning & imitation learning. *ICLR*, 2018.

[194] Wen Sun, Arun Venkatraman, Geoffrey J Gordon, Byron Boots, and J Andrew Bagnell. Deeply aggrevated: Differentiable imitation learning for sequential prediction. In *ICML*, pages 3309–3318, 2017.

[195] Richard S Sutton, Doina Precup, and Satinder Singh. Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial intelligence*, 112(1-2):181–211, 1999.

[196] David MJ Tax and Robert PW Duin. Support vector data description. *Machine learning*, 54(1):45–66, 2004.

[197] Emanuel Todorov. Convex and analytically-invertible dynamics with contacts and constraints: Theory and implementation in mujoco. In *ICRA*, pages 6054–6061, 2014.

[198] Faraz Torabi, Garrett Warnell, and Peter Stone. Behavioral cloning from observation. *IJCAI*, 2018.

[199] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, pages 4489–4497. IEEE, 2015.

[200] Luan Tran, Kihyuk Sohn, Xiang Yu, Xiaoming Liu, and Manmohan Chandraker. Gotta adapt 'em all: Joint pixel and feature-level domain adaptation for recognition in the wild. In *CVPR*, 2019.

[201] Y.-H. Tsai, G. Zhong, and M.-H. Yang. Semantic co-segmentation in videos. In *ECCV*, 2016.

[202] Yi-Hsuan Tsai, Wei-Chih Hung, Samuel Schulter, Kihyuk Sohn, Ming-Hsuan Yang, and Manmohan Chandraker. Learning to adapt structured output space for semantic segmentation. In *CVPR*, 2018.

[203] Yi-Hsuan Tsai, Kihyuk Sohn, Samuel Schulter, and Manmohan Chandraker. Domain adaptation for structured output via discriminative patch representations. In *ICCV*, 2019.

[204] Stepan Tulyakov, Anton Ivanov, and Francois Fleuret. Weakly supervised learning of deep metrics for stereo reconstruction. In *CVPR*, pages 1339–1348, 2017.

[205] Eric Tzeng, Judy Hoffman, Trevor Darrell, and Kate Saenko. Simultaneous deep transfer across domains and tasks. In *ICCV*, 2015.

[206] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *CVPR*, 2017.

[207] Jeroen van Baar, Alan Sullivan, Radu Cordorel, Devesh Jha, Diego Romeres, and Daniel Nikovski. Sim-to-real transfer learning using robustified controllers in robotic tasks involving complex dynamics. *ICRA*, 2018.

[208] Gül Varol and Albert Ali Salah. Efficient large-scale action recognition in videos using extreme learning machines. *Expert Systems with Applications*, 42(21):8274–8282, 2015.

[209] Paul Vernaza and Manmohan Chandraker. Learning random-walk label propagation for weakly-supervised semantic segmentation. In *CVPR*, 2017.

[210] Alexander Vezhnevets and Joachim M Buhmann. Towards weakly supervised semantic segmentation by means of multiple instance and multitask learning. In *CVPR*, pages 3249–3256. IEEE, 2010.

[211] Alexander Vezhnevets, Joachim M Buhmann, and Vittorio Ferrari. Active learning for semantic segmentation with expected change. In *CVPR*, pages 3162–3169. IEEE, 2012.

[212] Sudheendra Vijayanarasimhan and Kristen Grauman. Large-scale live active learning: Training object detectors with crawled data and crowds. *IJCV*, 108(1-2):97–114, 2014.

[213] Tuan-Hung Vu, Himalaya Jain, Maxime Bucher, Matthieu Cord, and Patrick Pérez. Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation. In *CVPR*, 2019.

[214] Tuan-Hung Vu, Himalaya Jain, Maxime Bucher, Matthieu Cord, and Patrick Pérez. Dada: Depth-aware domain adaptation in semantic segmentation. In *ICCV*, 2019.

[215] Botao Wang, Dahua Lin, Hongkai Xiong, and YF Zheng. Joint inference of objects and scenes with efficient learning of text-object-scene relations. *TMM*, 18(3):507–520, 2016.

[216] Heng Wang and Cordelia Schmid. Action recognition with improved trajectories. In *ICCV*, pages 3551–3558, 2013.

[217] Limin Wang, Yu Qiao, and Xiaoou Tang. Action recognition and detection by combining motion and appearance features. *THUMOS14 Action Recognition Challenge*, 1(2):2, 2014.

[218] Limin Wang, Yuanjun Xiong, Dahua Lin, and Luc Van Gool. Untrimmednets for weakly supervised action recognition and detection. In *CVPR*, 2017.

[219] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *ECCV*, pages 20–36. Springer, 2016.

[220] Zhenhua Wang, Qinfeng Shi, Chunhua Shen, and Anton Van Den Hengel. Bilinear programming for human activity recognition with unknown mrf graphs. In *CVPR*, pages 1690–1697, 2013.

[221] Yunchao Wei, Xiaodan Liang, Yunpeng Chen, Xiaohui Shen, Ming-Ming Cheng, Jiashi Feng, Yao Zhao, and Shuicheng Yan. Stc: A simple to complex framework for weakly-supervised semantic segmentation. *PAMI*, 39(11):2314–2320, 2017.

[222] Philippe Weinzaepfel, Xavier Martin, and Cordelia Schmid. Human action localization with sparse spatial supervision. *arXiv preprint arXiv:1605.05197*, 2016.

[223] Philippe Weinzaepfel, Xavier Martin, and Cordelia Schmid. Towards weaklysupervised action localization. *arXiv preprint arXiv:1605.05197*, 3(7), 2016.

[224] Adrian Weller, Kui Tang, David Sontag, and Tony Jebara. Understanding the bethe approximation: when and how does it go wrong? *UAI*, 2014.

[225] Zuxuan Wu, Xintong Han, Yen-Liang Lin, Mustafa Gkhan Uzunbas, Tom Goldstein, Ser Nam Lim, and Larry S. Davis. Dcan: Dual channel-wise alignment networks for unsupervised scene adaptation. In *ECCV*, 2018.

[226] Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *CVPR*, pages 3485–3492. IEEE, 2010.

[227] Huijuan Xu, Abir Das, and Kate Saenko. R-c3d: Region convolutional 3d network for temporal activity detection. In *ICCV*, volume 6, page 8, 2017.

[228] Yan Yan, Chenliang Xu, Dawen Cai, and Jason Corso. Weakly supervised actor-action segmentation via robust multi-task ranking. *CVPR*, 48:61, 2017.

[229] Bangpeng Yao and Li Fei-Fei. Modeling mutual context of object and human pose in human-object interaction activities. In *CVPR*, pages 17–24. IEEE, 2010.

[230] Jian Yao, Sanja Fidler, and Raquel Urtasun. Describing the scene as a whole: Joint object detection, scene classification and semantic segmentation. In *CVPR*, pages 702–709. IEEE, 2012.

[231] Serena Yeung, Olga Russakovsky, Greg Mori, and Li Fei-Fei. End-to-end learning of action detection from frame glimpses in videos. In *CVPR*, pages 2678–2687, 2016.

[232] Jun Yuan, Bingbing Ni, Xiaokang Yang, and Ashraf A Kassim. Temporal action localization with pyramid of score distribution features. In *CVPR*, pages 3093–3102, 2016.

[233] Zehuan Yuan, Jonathan C Stroud, Tong Lu, and Jia Deng. Temporal action localization by structured maximal sums. *CVPR*, 2017.

[234] Bowen Zhang, Limin Wang, Zhe Wang, Yu Qiao, and Hanli Wang. Real-time action recognition with enhanced motion vector cnns. In *CVPR*, pages 2718–2726, 2016.

[235] Chicheng Zhang and Kamalika Chaudhuri. Active learning from weak and strong labelers. In *NIPS*, pages 703–711, 2015.

[236] Yang Zhang, Philip David, and Boqing Gong. Curriculum domain adaptation for semantic segmentation of urban scenes. In *ICCV*, 2017.

[237] Yiheng Zhang, Zhaofan Qiu, Ting Yao, Dong Liu, and Tao Mei. Fully convolutional adaptation networks for semantic segmentation. In *CVPR*, 2018.

[238] Yue Zhao, Yuanjun Xiong, Limin Wang, Zhirong Wu, Xiaoou Tang, and Dahua Lin. Temporal action detection with structured segment networks. *ICCV*, 2017.

[239] Bineng Zhong, Hongxun Yao, Sheng Chen, Rongrong Ji, Tat-Jun Chin, and Hanzi Wang. Visual tracking via weakly supervised learning from multiple imperfect oracles. *Pattern Recognition*, 47(3):1395–1410, 2014.

[240] G. Zhong, Y.-H. Tsai, and M.-H. Yang. Weakly-supervised video scene co-parsing. In *ACCV*, 2016.

[241] Bolei Zhou, Agata Lapedriza, Jianxiong Xiao, Antonio Torralba, and Aude Oliva. Learning deep features for scene recognition using places database. In *NIPS*, pages 487–495, 2014.

[242] Zhi-Hua Zhou. Multi-instance learning: A survey. *Department of Computer Science & Technology, Nanjing University, Tech. Rep*, 2004.

[243] Yi Zhu, Yanzhao Zhou, Qixiang Ye, Qiang Qiu, and Jianbin Jiao. Soft proposal networks for weakly supervised object localization. *ICCV*, 2017.

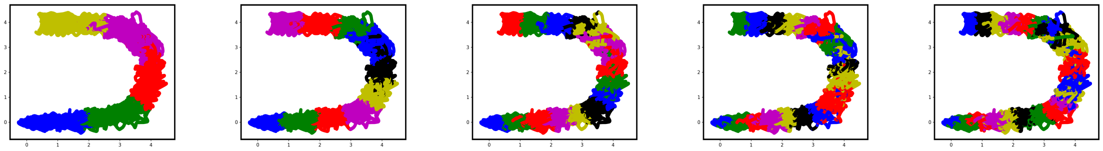
[244] Yingying Zhu, Nandita M Nayak, and Amit K Roy-Chowdhury. Context-aware modeling and recognition of activities in video. In *CVPR*, pages 2491–2498. IEEE, 2013.

[245] Yang Zou, Zhiding Yu, B. V. K. Vijaya Kumar, and Jinsong Wang. Domain adaptation for semantic segmentation via class-balanced self-training. In *ECCV*, 2018.

Appendix A

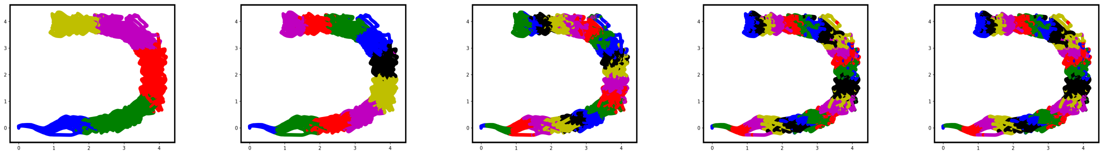
Variations in Learned Subgoals with Trajectories

We show visualizations of the learned sub-goals for different number of sub-goals. Fig. A.1 and Fig. A.2 shows the visualizations for the AntMaze task using sub-optimal and optimal trajectories respectively. Fig. A.3 and Fig. A.4 shows the visualizations for BiMGame and AntTarget respectively. It may be observed in Fig. A.1, that with high n_g, although our algorithm starts from the specified number of sub-goals, at the end of the sub-goal learning process, it ends up discovering fewer sub-goals (shown in brackets), $25 \rightarrow 21$ and $20 \rightarrow 18$. However, with optimal trajectories (Fig. A.2), our algorithm is able to discover the pre-specified number of sub-goals (at least till $n_g = 30$). This is due to the fact that the variations in the path taken by the optimal trajectories are much less than the sub-optimal trajectories. Thus, our algorithm is able to cluster the states more appropriately for optimal than sub-optimal trajectories. This actually shows the claim we



(a) $(5, 5)$ (b) $(10, 10)$ (c) $(15, 15)$ (d) $(20, 18)$ (e) $(25, 21)$

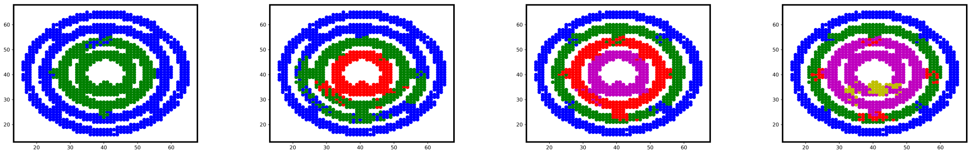
Figure A.1: (a) This figure presents the visualizations of the discovered sub-goals for AntMaze using the **sub-optimal** set of expert trajectories with different number of pre-specified sub-goals (n_g). The values as caption denote (no. of pre-specified sub-goals, no. of sub-goals learned).



(a) $(5, 5)$ (b) $(10, 10)$ (c) $(20, 20)$ (d) $(25, 25)$ (e) $(30, 30)$

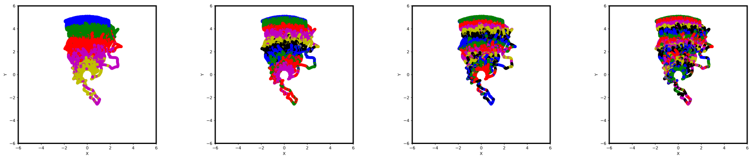
Figure A.2: This figure presents the visualizations of the discovered sub-goals for AntMaze using the **optimal** set of expert trajectories with different number of sub-goals (n_g) as input. The values as caption denote (no. of pre-specified sub-goals, no. of sub-goals learned).

make in the chapter, that our assumption of certain groups of states should follow some temporal ordering in the trajectories, are only soft and the degree by which they deviate determine the number and thus the granularity of the discovered sub-goals. Moreover, as we see in Fig. 5.4(c), even with sub-optimal trajectories, a low number of pre-specified sub-goals (such as $n_g = 10$) performs almost as good as with pre-specified $n_g = 25$, which actually discovers 21 sub-goals.



(a) $(2,2)$ (b) $(3,3)$ (c) $(4,4)$ (d) $(5,5)$

Figure A.3: This figure presents the visualizations of the discovered sub-goals for BiMGame with different number of sub-goals (n_g) as input. The values as caption denote (no. of pre-specified sub-goals, no. of sub-goals learned).



(a) $(2,2)$ (b) $(3,3)$ (c) $(4,4)$ (d) $(5,5)$

Figure A.4: This figure presents the visualizations of the discovered sub-goals for AntTarget using the expert trajectories with different number of sub-goals (n_g) as input. The values as caption denote (no. of pre-specified sub-goals, no. of sub-goals learned).

Appendix B

Semantic Segmentation Visualization

Fig. B.1 presents the semantic segmentation results before and after using weak labels for adaptation. The UDA method without using any weak labels produces more erroneous results in some portions and may miss some of the categories within a small area, such as sign, pole, etc. However, using the pseudo-weak labels enhances the segmentation and helps our model better identify the categories which originally have a lower confidence. Moreover, using oracle-weak labels is able to further improve the segmentation performance.

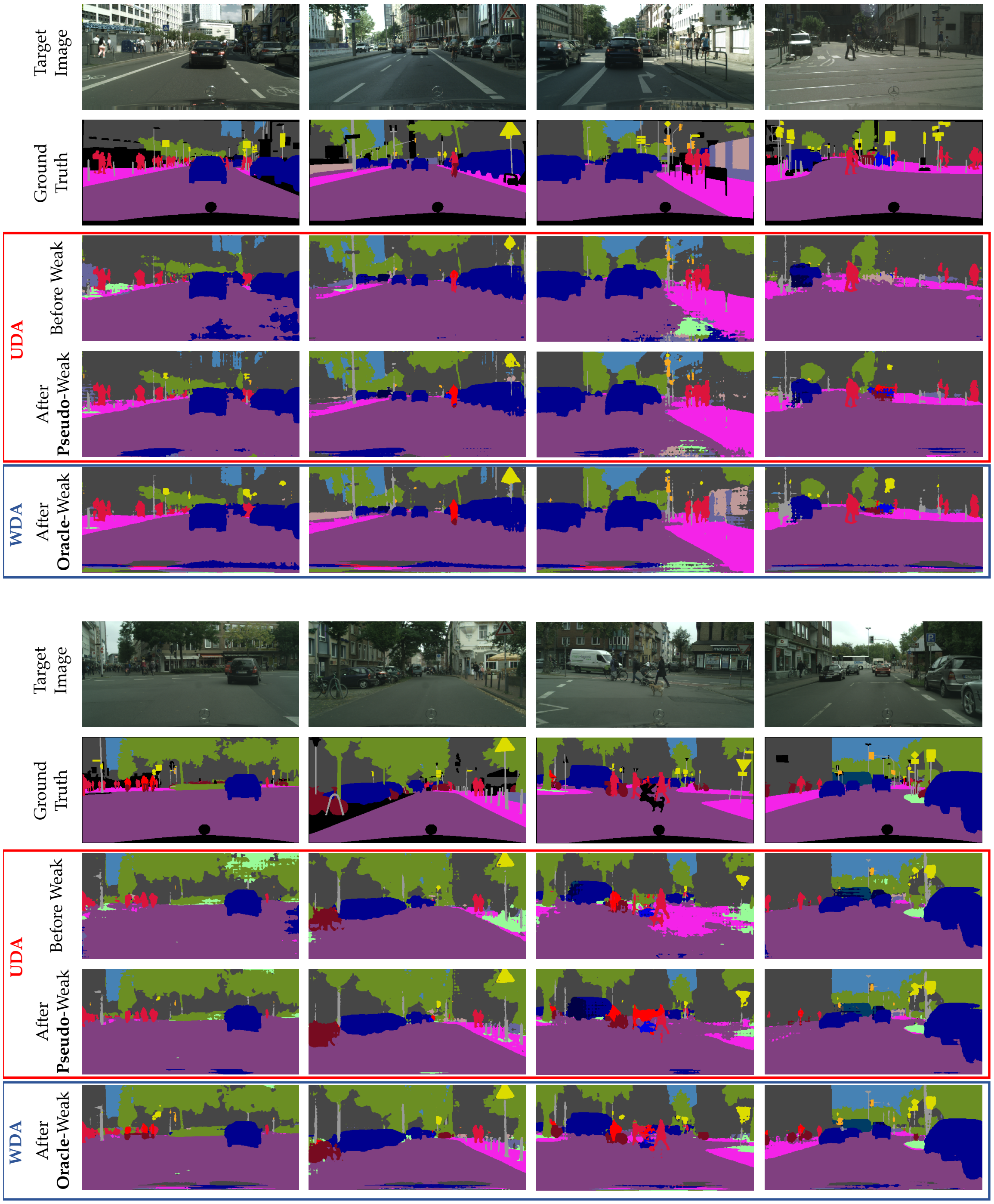


Figure B.1: Example results of adapted segmentation for GTA5 → Cityscapes with and without using weak labels for adaptation. The visualizations show that using pseudo-weak labels, the segmentation become more structured and some of the categories are better segmented. Using oracle-weak labels further improves the segmentation quality.