Learning with Weak Supervision for Visual Scene Understanding

A thesis submitted in partial fulfillment
of the requirements for the degree of

Doctor of Philosophy
in
Computer Science and Engineering

by

Aditya Arun
201407532
aditya.arun@research.iiit.ac.in

Advisors: Prof. C.V. Jawahar
Prof. M. Pawan Kumar

International Institute of Information Technology Hyderabad
500 032, India

June 2025

To
my parents, Kalpana & Arun,
and my sister, Aditi
–
for inspiring the journey and
supporting me every step of the way.

International Institute of Information Technology Hyderabad
Hyderabad, India

CERTIFICATE

This is to certify that work presented in this thesis proposal titled ***Learning with Weak Supervision for Visual Scene Understanding*** by *Aditya Arun* has been carried out under my supervision and is not submitted elsewhere for a degree.

_____ _____

Date Advisor: Prof. C.V. Jawahar

_____ _____

Date Advisor: Prof. M. Pawan Kumar

Acknowledgments

I would like to express my deepest gratitude to my PhD advisors, Prof. C.V. Jawahar and Prof. M. Pawan Kumar, for their unwavering guidance and support throughout the journey of my PhD. Their profound insights and expertise in the field of machine learning have immensely benefited me, shaping my understanding and approach to research. Beyond technical inputs, I have learned invaluable lessons on research methodology, critical thinking, and the art of disseminating research effectively. I am especially thankful to them for their patience and understanding throughout this journey, which has been instrumental in my growth.

I am also thankful to my funding agency, the Visvesvaraya Fellowship, for supporting my research endeavors during this time. I acknowledge the travel grants from Google and IIIT Hyderabad, which enabled me to attend conferences and present my work on global platforms. My heartfelt thanks go to the CVIT staff— Satya, Siva, Rohitha, Ram, Silar, and others — for their consistent support and for ensuring the smooth functioning of administrative processes.

During my time at IIIT Hyderabad, I have been fortunate to meet incredible colleagues who have become lifelong friends. They have inspired and supported me, standing by me in both my highs and lows. Special thanks to Pritish for always being there as a friend, mentor, and source of encouragement. Friends like Pritish, Sourabh Daptardar, Yashaswi Verma, Aniket, Avijit, Thrupthi, Praveen, Suriya, Saurabh Saini, Arunava, Minesh, Jobin, Deepayan, Lovish, Zeeshan, Varun, Anand, Rajvi, Yasaswi Bharadwaj, Abhishek, Tejaswi, Isha, Govinda, Devendra, Ajeet, and many others have made this journey memorable and fulfilling.

Finally, I would like to express my heartfelt gratitude to my parents, Kalpana and Arun, and my sister, Aditi, for their unconditional love, support, and sacrifices, without which this journey would not have been possible. Their belief in me has been my greatest source of strength. I am deeply grateful for their constant encouragement and the foundation they have provided me. I am also thankful to Kaushik for his support and to Namrata for being a constant source of encouragement and standing by me.

Abstract

In recent years, computer vision has made remarkable progress in understanding visual scenes, including tasks such as object detection, human pose estimation, semantic segmentation, and instance segmentation. These advancements are largely driven by high-capacity models, such as deep neural networks, trained in fully supervised settings with large-scale labeled data sets. However, reliance on extensive annotations poses scalability challenges due to the significant human effort required to create these data sets. Fine-grained annotations, such as pixel-level segmentation masks, keypoint coordinates for pose estimation, or detailed object instance boundaries, provide the high precision needed for many tasks but are extremely time-consuming and costly to produce. Coarse annotations, on the other hand, such as image-level labels or approximate scribbles, are much easier and faster to create but lack the granularity required for detailed model supervision.

To address these challenges, researchers have increasingly explored alternatives to traditional supervised learning, with weakly supervised learning emerging as a promising approach. This approach mitigates annotation costs by utilizing coarse annotations (cheaper and less detailed) during training rather than the fine-grained annotations required at the output stage during testing. Despite its potential, weakly supervised learning faces challenges in transferring information from coarse annotations to fine-grained predictions, often encountering ambiguity and uncertainty during this process. Existing methods rely on various priors and heuristics to refine annotations, which are then used to train models for specific tasks. This involves managing uncertainty in latent variables during training and ensuring accurate predictions for both latent and output variables at test time.

This thesis introduces a unified approach to weakly supervised learning in computer vision, addressing tasks such as human pose estimation, object detection, and instance segmentation. Central to this work is a framework based on the dissimilarity coefficient loss, which models uncertainty in the location of objects and human poses using coarse annotations. The approach employs two key probability distributions:

- Conditional Distribution: Captures output probabilities using coarse annotations (e.g., action labels, image-level labels, object counts), modeled with deep generative models for efficient sampling.

- Prediction Distribution: Provides test-time predictions independent of coarse annotations.

The framework minimizes the difference between these distributions using the dissimilarity coefficient loss, facilitating the transfer of information from coarse annotations to accurate predictions. This methodology is consistently applied across diverse computer vision tasks, showcasing its versatility.

The efficacy of the proposed framework is demonstrated across three progressively complex visual scene recognition tasks:

- Human Pose Estimation: A probabilistic framework is introduced for learning human poses from still images using data sets with costly ground-truth pose annotations and inexpensive action labels. By aligning the conditional and prediction distributions through the dissimilarity coefficient loss, the method achieves significant improvements over baselines on the MPII and JHMDB data sets, effectively leveraging action information.

- Object Detection: The framework addresses weakly supervised object detection (WSOD) by modeling uncertainty in object locations using a dissimilarity coefficient-based objective. Leveraging discrete generative models, it efficiently samples from annotation-aware conditional distributions and integrates coarse annotations, such as image-level labels, object counts, points, and scribbles. Spatial cluster regularization and curriculum learning further enhance performance, achieving state-of-the-art results on benchmarks like PASCAL VOC and MS COCO.

- Instance Segmentation: The framework models uncertainty in pseudo-label generation using semantic class-aware, boundary-aware, and annotation-consistent higher-order terms. By aligning conditional and prediction distributions, it generates accurate pseudo-labels and trains Mask R-CNN-like architectures effectively. Experiments on the PASCAL VOC 2012 data set demonstrate state-of-the-art performance, with improved object boundary alignment and significant gains over baselines.

Contents

List of Figures

List of Tables

List of Related Publications

[P1] Aditya Arun, C.V. Jawahar, and M. Pawan Kumar, **"Learning Human Poses From Actions"**, in proceedings of *British Machine Vision Conference (BMVC)*, 2018.

[P2] Aditya Arun, C.V. Jawahar, and M. Pawan Kumar, **"Dissimilarity Coefficient based Weakly Supervised Object Detection"**, in proceedings of *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[P3] Aditya Arun, C.V. Jawahar, and M. Pawan Kumar, **"Weakly Supervised Instance Segmentation by Learning Annotation Consistent"**, in proceedings of *European Conference on Computer Vision (ECCV)*, 2020.

Related papers under review:

[P4] Aditya Arun, C.V. Jawahar, and M. Pawan Kumar, **"Spatial Consistency Enhanced Dissimilarity Coefficient based Weakly Supervised Object Detection"**, (in submission)

Chapter 1

Introduction

1.1 Motivation

In recent years, the computer vision community has seen significant progress in visual scene understanding, like object detection [1–3], human pose estimation [4, 5], semantic segmentation [6–8], instance segmentation [9, 10], etc. This has primarily been driven due to the application of high-capacity models, like deep neural network architectures, on these tasks in a fully-supervised setting, utilizing massively parallel compute resources (GPUs).

Although fully supervised deep learning based methods have created a profound impact, they suffer from scalability issues. It has been empirically shown that the performance of the deep learning based methods improves with more data [11]. Not only do the deep learning methods require huge amounts of labeled data, but they can also not generalize to multiple domains and tasks. This has lead to a community-driven effort to create *large-scale*, *well-curated*, and *task-specific labeled* data sets, such as ImageNet [12], PASCAL-VOC [13], MS-COCO [14], CityScapes [15], OpenImages [16], ADE-20k [17], MPII Human Pose [18] among others.

However, creating such a data set requires significant human labor. A study on PASCAL-VOC 2012 data set [13] shows that it takes 1 second per class to collect image-level labels (20 seconds for the whole image) [19], while it takes 79 seconds to get per object segmentation masks (239.7 seconds per image). Obtaining point-level annotations, scribble annotations, and bounding-box annotations take 2.4 seconds, 10.9 seconds, and 10.2 seconds per instance (or 22.1 seconds, 34.9 seconds, and 33.8 seconds per image) on PASCAL VOC data set [19] respectively.

Moreover, with the increase in the number of classes and image complexity, the annotation cost also increases significantly. On MS-COCO, it takes 27 seconds to annotate the 80 classes to obtain image-level labels [14]. An additional 14 seconds are required to spot the instances and provide point annotations, and it takes an additional 80 seconds per instance (19 minutes per image) to draw a polygon to provide instance masks [14,20]. It demands an additional 3 minutes to annotate the stuff regions [20], taking the total annotation time for an entire image to 22 minutes. Annotating every pixel in a single image in a more complex CityScapes data set [15] takes 1.5 hours.

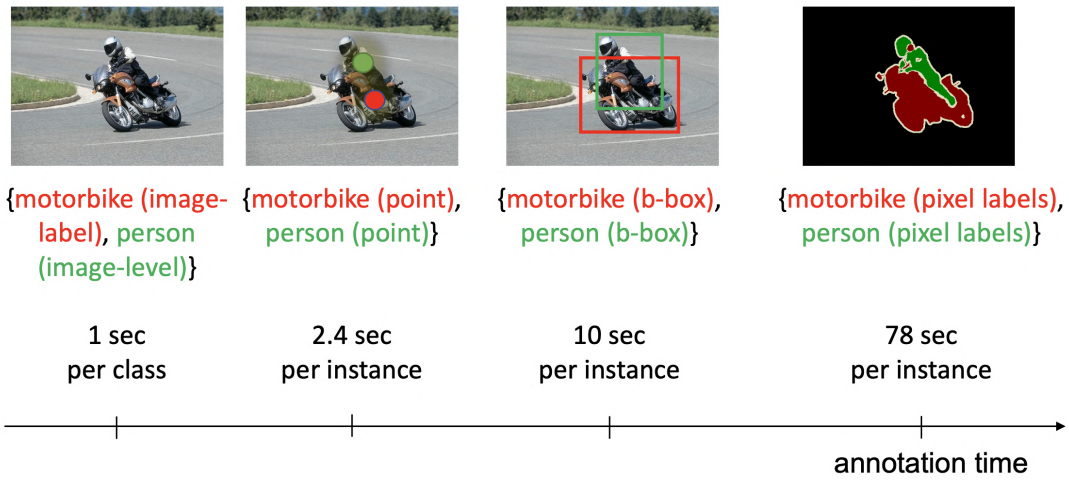
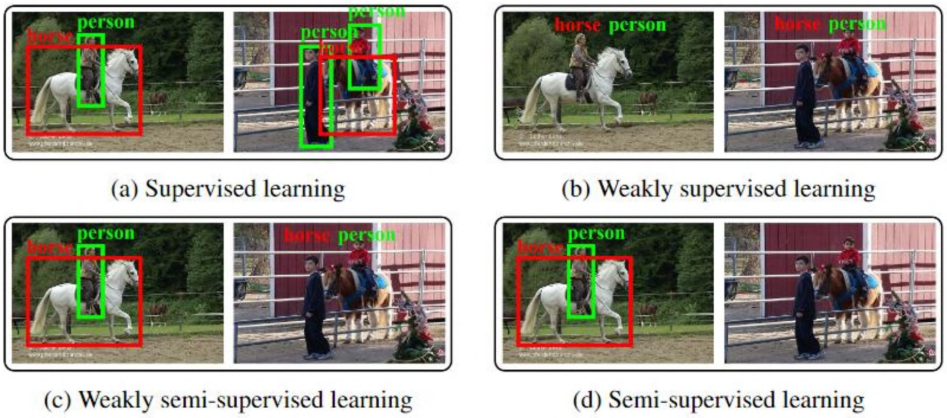


Figure 1.1 *Figure shows the cost of annotation for different annotation types. The cost of annotation increases significantly with more complex annotations. Notice that high-cost annotations can be converted to low-cost annotations easily, e.g. from bounding box to image level, whereas the low-cost annotation is hard to be transformed into the higher-cost annotation.*

To address these challenges, considerable research efforts have been dedicated within the deep learning community. Broadly, we can categorize these beyond supervised techniques into three categories (i) Data-centric techniques that solve the problem by generating a large amount of data similar to the original data set; (ii) algorithm-centric techniques that tweak the learning methods to harness limited data efficiently through various techniques like on-demand human intervention, exploiting the inherent structure of the data, capitalizing on freely available data on the web, or solving for an easier but unrelated surrogate task; and (iii) hybrid techniques that combine the ideas from both data and algorithm-centric approaches.

The data-centric techniques include data augmentation, which involves tweaking the data samples from pre-defined transformations to increase the overall size of the data set [21–23]. Algorithm-centric techniques try to relax the need for perfectly labeled data by altering the model requirements to acquire supervision through inexact [24] (weakly supervised learning), inaccurate [25] (learning with noisy supervision), and incomplete labels [26] (semi-supervised learning). These labels are cheaper and easier to obtain for most tasks than task-pertinent annotations. Techniques involving on-demand human supervision have also been used to label selective instances from the data set [27] (active learning). Another set of methods exploits the knowledge gained while learning from a related domain task and transferring it to the test environment [28, 29] (domain adaptation, k-shot learning). Yet another approach to avoiding task-pertinent annotations is to define an auxiliary task that provides the supervisory signal without using explicit labels [30] (self-supervised learning). Hybrid techniques improve the model's performance at both data and algorithm levels. Most of the beyond-supervised methods fall into this category.

This thesis will focus on Weakly Supervised Learning, its key concepts, and its application in visual scene understanding. By addressing the high cost and scalability challenges associated with fully super-



(a) Supervised learning (b) Weakly supervised learning

(c) Weakly semi-supervised learning (d) Semi-supervised learning

Figure 1.2 *The figure shows the comparison of annotations available at training time for various beyond-supervised approaches for the task of object detection. For supervised learning, a bounding box annotation is present in the training set for each foreground object's instance. For weakly supervised learning, lower-degree image-level annotation is present in the training set. For weak-semi supervised learning, a small number of images are annotated with higher-order bounding box annotations, and the remaining training data has lower-degree image-level annotations. In semi-supervised learning, a small fraction of images are annotated with higher-order bounding box annotations while the rest of the training set do not have any annotations.*

vised learning, weakly supervised learning provides a compelling alternative that leverages inexpensive and less detailed annotations while aiming to achieve fine-grained predictions. This work will delve into the proposed framework, which bridges the gap between coarse and fine-grained annotations, enabling its application across various visual scene recognition tasks with reduced annotation costs and improved efficiency.

1.2 Weakly Supervised Learning

1.2.1 Problem Setting

A weakly supervised learning (WSL) problem reduces the annotation cost by using coarse (cheaper-to-obtain) annotations at the training time instead of the fine-grained (expensive-to-obtain) annotations required at the test time. Coarse annotations, such as image-level labels, are less detailed and easier to acquire, often involving minimal manual effort. On the other hand, fine-grained annotations, such as pixel-level labels or bounding boxes, require extensive manual labeling and are thus more costly. In other words, the goal of WSL is to train a task-pertinent model using easier-to-obtain *weak* labels, as seen in Fig. 1.3.

Obtaining coarse annotations is not only cost-effective (Fig. 1.1) but, with the ever-increasing visual data available online, it is sometimes free. Indeed, a simple search for a keyword such as "car" on

an image search engine results in hundreds of freely available images. Many popular image-hosting websites such as Flickr[1] allow users to tag the images with labels that can be utilized as annotations.

Using such easy-to-obtain labels enables the creation of large-scale data sets. This helps not only scale the models to many classes but also achieve a more robust and accurate model. Moreover, in some domains such as 3D vision, medical imaging, etc., obtaining exact (supervised) annotations is either prohibitively expensive or impossible to acquire without expertise. For such tasks, WSL provides a practical approach to training such predictors.

Due to these advantages, weakly supervised learning has received much attention in the computer vision community. Numerous methods on this topic have been proposed in the past two decades to address challenging vision tasks, including object detection [2], semantic segmentation [7], and instance segmentation [10], among others.

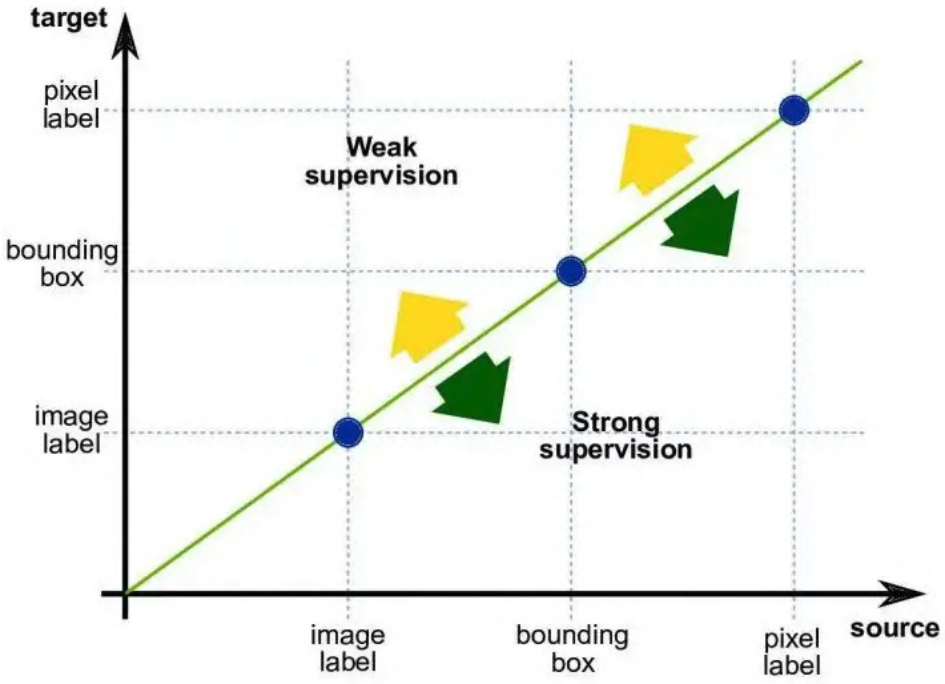


Figure 1.3 *Illustration of the weakly supervised learning task in the computer vision community. The x-axis represents the annotation available during training, and the y-axis denotes the target or output task. The blue points on the line represent the supervised learning scenario, where the annotations at the training time match the output task. The region below the line represents a strong supervision scenario where the information contained in the training label exceeds the output task. The region above the line represents a weakly supervised scenario where the annotation available at the training time is coarser than what is required at the output task.*

[1] https://flickr.com/

1.2.2 Comparison with Other Beyond-Supervised Approaches

In what follows, we compare how WSL approaches relate to other beyond-supervised approaches such as domain adaptation, semi-supervised learning, few/zero-shot learning, and unsupervised/self-supervised approaches. These approaches aim to reduce the annotation cost by using fewer fine-grained annotated training data, leveraging coarse annotations, or without using explicit annotations.

1.2.2.1 Weakly Supervised Learning vs Domain Adaptation

Domain adaptation [29] aims to adapt a model trained on one domain to perform well on a different domain. This is useful when the training data and the target domain are not well-matched, and the model needs to be adjusted to make accurate predictions on the target domain. **Fine-tuning** and **transfer learning** are among the popular approaches in domain adaptation.

In domain adaptation, a large-scale training data set is available in the training domain. Only a few training samples with fine-grained annotations are available for the target domain. This contrasts with WSL approaches that leverage large-scale coarse annotations for both the training and target domains.

1.2.2.2 Weakly Supervised Learning vs Semi-Supervised Learning

A semi-supervised learning approach [26] aims to leverage a partially labeled data set. This means that while some of the data in the training set has been given explicit fine-grained annotations, the rest of the data is not annotated. The semi-supervised approach is useful in situations where it is difficult or expensive to obtain fine-grained annotations.

The main difference between WSL and semi-supervised learning is the type of data used in training. In semi-supervised learning, the model is trained using a small amount of labeled data with fine-grained annotations and a large amount of unlabeled data, while in WSL, the model is trained using a large amount of data with coarse annotations.

To leverage the advantages of both these approaches, **semi-weakly supervised learning** [31, 32] is used, where only a small amount of fine-grained annotations and large-scale coarse annotations are available during training.

1.2.2.3 Weakly Supervised Learning vs Few-Shot Learning

Few-shot learning techniques [28] attempt to train a model using a very small number of examples with fine-grained annotations. The goal is to enable the model to generalize from the limited training data and perform well on novel examples from the same domain or tasks. **1-shot learning** and **0-shot learning** approaches are special cases of few-shot learning. In 1-shot learning, the aim is to train a model using exemplars, and in 0-shot learning, the aim is to learn novel object classes at inference when no training examples of it are present during training.

In few-shot learning, the training data is typically a small set of high-quality, carefully labeled examples, while in WSL, the training data is often larger but has less reliable annotations. In essence,

few-shot learning focuses on learning from a limited amount of data, while WSL focuses on learning from noisy or unreliable data.

1.2.2.4 Weakly Supervised Learning vs Unsupervised Learning

Unsupervised learning [33] is a type of machine learning algorithm where models are trained on data sets without any labels or pre-defined categories. The goal of unsupervised learning is to find hidden structures in the data and to use those patterns to make predictions or decisions. One of the main advantages of unsupervised learning is that it can be used on large data sets where it would be impractical or impossible to label all the data manually. **Self-supervised learning** [30] is a type of unsupervised learning algorithm where a model is trained on a data set that has been automatically labeled by the model itself, rather than being labeled by humans.

The key difference between unsupervised learning and WSL approaches is that unsupervised approaches use no annotations, while WSL approaches have coarse annotations labeled by an external source. On the other hand, in self-supervised learning, the labels are generated by the model itself. Another difference is the type of information learned from the labels. In unsupervised and self-supervised learning, the goal is to learn more complex or abstract representations of the data, using the data itself as the source of supervision. In contrast, WSL typically aims to make predictions or decisions using coarse annotations.

Considerable progress has been achieved in methods based on the self-supervised understanding of visual scenes [34–36]. Yet, there remains a gap in performance when these methods are compared to weakly supervised approaches. The current state-of-the-art self-supervised method for object detection and instance segmentation [35] achieves 31.0% $\mathrm{mAP}^{\mathrm{box}}_{50}$ on the PASCAL VOC 2012 data set for the object detection task using a ResNet50 base model. On the same data set but using an inferior base model (VGG16), our proposed approach [37] (Chapter 5) achieves 48.4% $\mathrm{mAP}^{\mathrm{box}}_{50}$. For the instance segmentation task, they achieve 31.0% $\mathrm{mAP}^{r}_{0.50}$ on the PASCAL VOC 2012 data set, while our proposed approach [38] (Chapter 6) with the same base network and setting achieves 50.9% $\mathrm{mAP}^{r}_{0.50}$. This demonstrates the relevance of weakly supervised learning for visual scene understanding over self-supervised approaches.

1.2.2.5 Hybrid Supervision

Hybrid supervision is a method of training machine learning models that combines multiple types of supervision or labeling [39–41]. This can include both human-provided labels, such as those used in traditional supervised learning, as well as automatically generated labels, such as those produced in WSL, semi-supervised, or self-supervised learning algorithms. The goal of hybrid supervision is to combine the strengths of different labeling methods in order to improve the performance of the models with the least annotation cost.

1.2.3 Challenges

As WSL methods employ coarse annotations, the learning framework needs to address not only typical issues, such as intra-class variations in appearance, transformations, scale, and aspect ratio, encountered in conventional fully supervised approaches, but also the challenges caused by the inconsistency between available coarse annotations and real supervisory signals. In WSL, the model's accuracy and learning process are often interdependent. The key is propagating the coarse supervisory signal to the task-specific supervisory signal for the learning process. Since each coarse annotation in the training image can correspond to numerous fine-grained annotations of varying accuracy, propagating such weak supervision inevitably involves a large amount of ambiguous and noisy information in each training instance. For example, in weakly supervised object detection, each image-level label can correspond to several bounding boxes with varying levels of accuracy. As a concrete example, consider an image annotated with the label *cat*. While this coarse annotation indicates the presence of a cat in the image, it does not specify its exact location or outline. Consequently, multiple candidate bounding boxes may represent the object, some capturing the full extent of the cat, others focusing on partial regions, and some even including irrelevant background objects.

More formally, the issue of ambiguous and noisy information transfer from coarse annotations to fine-grained annotations can be characterized under **learning under uncertainty**. The paradigm of learning under uncertainty introduces the following challenges in the WSL process:

- **Learning with inexact labels:** This issue arises primarily due to the ambiguity between coarse annotations and the output task. Without precise annotation or definition, a learner may struggle to determine whether an object category label corresponds to a discriminative object part or the entire object region. For example, consider an image annotated with the label *car*. The label does not specify whether the learner should focus on the entire car or its components, such as the wheels or headlights, leading to potential confusion during training. As a result, the output inferred by the learner may contain many inaccurate samples, including those with local object parts, thus negatively affecting its accuracy.

- **Learning with noisy samples:** As there is no additional information to separate objects from the background or other co-occurring objects, it is difficult to distinguish between them. For instance, in an image labeled with *train*, the learner might mistakenly associate the label with irrelevant elements like railway tracks, or in the case of an image labeled with *aeroplane*, it might associate the label with the surrounding sky. Therefore, the learner may incorrectly tag the background or co-occurring objects with the same label as the object of interest, affecting its accuracy.

Furthermore, issues encountered in supervised approaches, such as inter- and intra-class variance in appearance, diversity of training images, scale, and aspect ratio, are further exaggerated under the learning under uncertainty paradigm due to the lack of task-level supervisory signals.

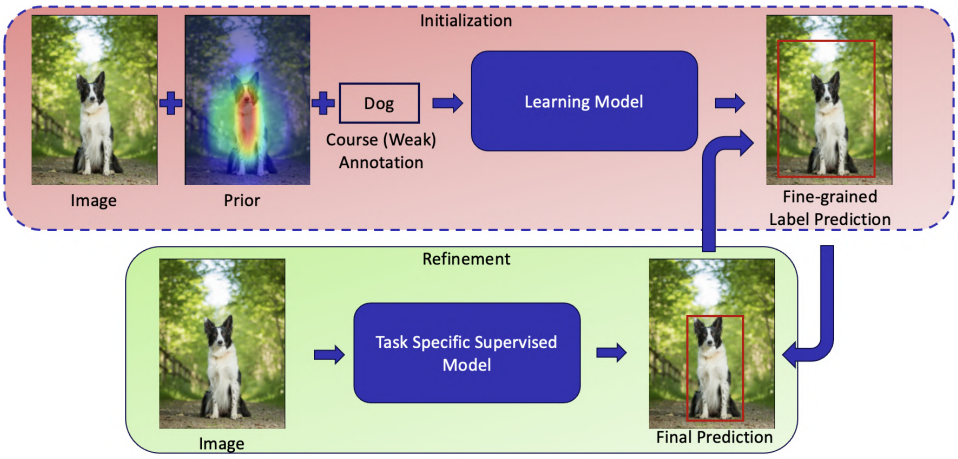


Figure 1.4 *Flowchart of the general framework of weakly supervised learning.*

1.2.4 General Framework

To address the issues of learning under uncertainty described above, a plethora of approaches have been developed in the past two decades. These approaches broadly contain one or a combination of the following steps:

- **Priors and Hints:** The key idea in WSL is to use coarse annotations to create a much more powerful predictor. The essential ingredients for WSL are the use of priors and hints.

  Priors refer to assumptions or beliefs about a task that are independent of specific images or annotations. These include information about the problem known before examining the data. Common examples include object priors such as shape, size, and contrast; relative motion to determine object boundaries; and similarity across images to identify standard object features. Priors can be explicitly defined or implicitly encoded through dataset biases, model architectures, or hyperparameters.

  Hints are forms of indirect supervision derived from available annotations for each image. Examples include image-level labels, bounding boxes, image captions, clicks, scribbles, and sparse temporal labels.

  Some common priors and hints are listed in Table 1.1.

- **Initialization:** The initialization stage leverages certain prior knowledge to propagate coarse annotations to the task level, thus generating fine-grained annotations (albeit with label noise, sample bias, and limited accuracy). These generated task-level annotations can either be treated as the final output or as training annotations for the next stage (refinement stage). During this stage, efforts focus on enhancing the quality of generated task-level annotations to create training instances with accurate labels, high diversity, and a high recall rate.

Table 1.1 *Most common priors and hints. Priors represent task-specific assumptions, while hints are indirect forms of supervision derived from available annotations.*

Prior	Hint
Size	Image labels
Shape	Image captions
Location (object-centered representation)	Bounding Boxes
Number of instances	Video labels
Contrast (boundaries, saliency)	Click inside object
Class distribution	Scribbles
Motion	Sparse temporal labels
Similarity across images	Eye gaze
Similarity with external images	Localized narrative

- **Refinement:** The refinement stage leverages new instance samples obtained from the initialization stage to train a task-specific supervised model and ultimately obtain the desired predictor. Since the annotations generated during the initialization stage may contain inaccuracies, efforts during the refinement stage focus on improving the learner's robustness to cope with noisy, biased labels and enhance its capacity to utilize unlabeled instance samples effectively.

The above learning steps should collaborate effectively to address the challenges of the learning under uncertainty paradigm. A typical approach to realizing such collaboration involves either optimizing initialization and refinement independently, iteratively alternating between these steps, or jointly optimizing them.

A typical WSL method incorporates one or more of these steps, often applying them in a cascaded sequence. Visually, these steps are related as shown in Fig. 1.4.

1.3 Scope and Contributions

1.3.1 Scope

This thesis focuses on advancing weakly supervised learning (WSL) to enable the training of deep neural networks for complex scene understanding tasks using static image data. It addresses the fundamental challenge of relying on coarse annotations or weak labels – which are less detailed but cheaper to obtain – during training, while aiming for fine-grained predictions – which are detailed and task specific outputs – at the test time. The primary focus is on developing a unified probabilistic framework that explicitly models uncertainty of transferring information from weak annotations to fine-grained predictions across a range of computer vision tasks. Specifically, the thesis focuses on:

- Developing a generalized approach to WSL using probabilistic principles that explicitly models uncertainty and can be applied across multiple scene recognition tasks.

- Addressing key challenges in WSL, such as learning with inexact and noisy labels, and ensuring effective propagation of coarse annotations to task-specific predictions.
- Demonstrating the versatility of the proposed framework through its application to tasks such as human pose estimation, object detection, and instance segmentation incorporating relevant priors and hints.
- Achieving state-of-the-art performance on benchmark data sets by refining coarse annotations to generate accurate fine-grained predictions.

1.3.2 Contributions

The thesis introduces a unified probabilistic framework for WSL that is adaptable to various visual scene understanding tasks. A key feature of this framework is the use of two distinct distributions to model the initialization and refinement tasks: generating accurate fine-grained labels during training for supervised task-specific models, and leveraging only the task-specific models at test time. These distributions are aligned using a novel dissimilarity coefficient-based objective, contrasting with existing methods that often burden a single model with conflicting tasks and lack explicit uncertainty modeling. Specifically:

- **Conditional Distribution:** This distribution generates task-specific prediction based on coarse annotations, such as action labels or image-level annotations, using deep generative models. It provides a mechanism to sample plausible outputs while accounting for the inherent ambiguity in weak annotations.
- **Prediction Distribution:** This distribution generates final test-time predictions independent of weak annotations. By aligning it with the conditional distribution, the framework ensures that information from weak annotations is effectively transferred to fine-grained predictions.

These distributions are aligned using a novel dissimilarity coefficient loss, which minimizes their divergence to improve prediction accuracy and robustness. The optimization proceeds by either jointly training both distributions or by employing coordinate descent strategy where the two distributions are optimized iteratively by keeping one constant. The framework's flexibility enables the integration of diverse priors and hints, such as activation maps and spatial constraints, making it adaptable to a wide range of vision tasks.

The efficacy of the proposed framework is demonstrated on three increasingly complex visual scene understanding tasks: human pose estimation, object detection, and instance segmentation.

- **Human Pose Estimation:** The framework predicts detailed human pose keypoints from coarse annotations like action labels. By modeling pose uncertainty and incorporating prior knowledge of human body structures through the conditional distribution, the framework aligns these with the prediction distribution using the dissimilarity coefficient loss. The framework employs a

novel deep generative model, DISCO Nets, for the conditional distribution and the state-of-the-art Hourglass Networks for the prediction distribution, achieving significant improvements on benchmarks like MPII and JHMDB.

- **Object Detection:** The proposed framework addresses uncertainty in object localization using a diverse range of coarse annotations, including image-level labels, count annotations, point annotations, and scribble annotations. The conditional distribution incorporates priors such as class activation maps, spatial regularization, and higher-order constraints to ensure that the generated fine-grained labels align with the provided coarse annotations. To efficiently sample from this complex conditional distribution, a greedy iterative algorithm is introduced, leveraging the discrete generative model Discrete DISCO Nets. The prediction distribution is modeled using the widely adopted Fast R-CNN model, and the two distributions are aligned through a dissimilarity coefficient loss. This approach achieves state-of-the-art performance on benchmark data sets, including PASCAL VOC 2007, PASCAL VOC 2012, and MSCOCO 2017.

- **Instance Segmentation:** The proposed framework models uncertainty in pseudo-label generation through the conditional distribution, utilizing diverse coarse annotations such as image-level labels and bounding box annotations. The conditional distribution integrates a semantic class-aware unary term, a boundary-aware pairwise smoothness term, and an annotation-consistent higher-order term. Samples from this complex distribution are generated using a greedy iterative algorithm and a discrete generative model, such as Discrete DISCO Net. The prediction distribution is modeled with the widely adopted instance segmentation approach, Mask R-CNN. Alignment between the two distributions is achieved through a dissimilarity coefficient loss. This framework delivers state-of-the-art performance on the benchmark PASCAL VOC 2012 data set.

1.4 Thesis Outline

The thesis is organized as following chapters:

- **Chapter 1 Introduction** In the first chapter, we discuss the motivation behind our work, focusing on the use of weakly supervised learning for visual scene understanding. We present a comparison of this approach with other methods beyond supervised learning and analyze the general trends in research within this domain. Additionally, we outline the scope of our research.

- **Chapter 2 Prior Work** In this chapter, we discuss the history of weakly supervised approaches and their application to visual scene understanding. We examine the critical advances that have improved performance in this area.

- **Chapter 3 Dissimilarity Coefficient based Weakly Supervised Learning Framework** In this chapter, we present the proposed weakly supervised probabilistic framework based on the dissim-

ilarity coefficient loss. We also discuss the prior work that directly enabled the development of this framework.

- **Chapter 4 Weakly Supervised Human Pose Estimation** In this chapter, we discuss the problem of learning human pose estimation using cheaper-to-obtain action annotations. We present our probabilistic weakly supervised framework based on the dissimilarity coefficient objective and demonstrate its efficacy on the MPII and JHMDB data sets.

- In **Chapter 5 Weakly Supervised Object Detection** In this chapter, we present our work on the weakly supervised object detection task, which aims to learn object detection models using image-level (classification) annotations. We discuss the challenges of modeling the problem due to the complex conditional distribution and present an efficient solution by employing a discrete generative model. We demonstrate the efficacy of our proposed approach on the PASCAL VOC 2007, 2012 and MS COCO 2017 data sets. This chapter was part of the CVPR 2019 paper and also the under submission work

- **Chapter 6 Weakly Supervised Instance Segmentation** In this part we will show how to model for the task of weakly supervised instance segmentation, where we required to train an instance segmentation model using image-level (classification) annotations. In order to overcome the challenges of instance segmentation, we carefully design the conditional distribution that has a unary term, a pairwise term, and a higher-order term. We model the distribution using a discrete generative model and present an efficient sampling algorithm. We demonstrate the efficacy of our proposed approach on PASCAL VOC 2012 dataset.

- **Chapter 7 Conclusion and Future Work** This part of the thesis would include a summary of the contributions and would draw directions for future research in related areas.

Chapter 2

Prior Work

Over the past two decades, weakly supervised learning (WSL) has been an active area of research, driven by its ability to address the challenge of limited fully labeled data for complex visual tasks. This chapter explores the evolution of weakly supervised approaches in visual scene understanding, highlighting key milestones and seminal works that have significantly advanced performance in this domain.

WSL methods can be broadly classified into three categories: (i) classical approaches based on handcrafted features, (ii) methods leveraging feature representations from pre-trained deep models, and (iii) end-to-end deep weakly supervised learning algorithms. This chapter reviews foundational works and techniques from each category, examining their contributions, strengths, and limitations. By delving into these categories, we aim to provide a comprehensive perspective on the progression of WSL methodologies and their transformative impact on computer vision.

2.1 Classical Approaches to Weakly Supervised Learning

Classical approaches to Weakly Supervised Learning (WSL) predominantly rely on handcrafted features rather than deep features. These methods are typically grounded in **Latent Variable Models (LVMs)**, a statistical framework that links observed variables to a set of latent (or hidden) variables. Latent variables encapsulate an underlying structure that explains relationships within the observed data. In WSL, observed variables often represent tasks associated with coarse annotations (e.g., image classification), while latent variables model more intricate outputs (e.g., object detection or semantic segmentation).

This section explores three primary types of LVMs and their applications in WSL for visual tasks: **Multiple Instance Learning (MIL) Models**, **Max-Margin Models**, and **Probabilistic Graphical Models (PGMs)**.

2.1.1 Multiple Instance Learning (MIL) Models

Multiple Instance Learning (MIL), introduced by Dietterich *et al.* [42], organizes training data into sets known as "bags," each labeled with a single label. The defining assumption, termed the *standard*

MIL assumption, posits that a bag is positive if at least one of its instances is positive. MIL's goal is to infer instance labels or bag-level predictions by resolving ambiguities within training bags. For example, Dietterich *et al.* [42] applied MIL to predict drug activity, where molecular conformations (instances within a bag) could either induce a positive or negative effect.

The key challenge in MIL lies in identifying the specific instances within a positive bag that contribute to the bag's label. Early methods like axis-parallel rectangles [42] addressed this by constructing decision boundaries for instance-level classification. Maron and Lozano-Pérez [24] proposed a more generalized framework using **diversity density**, which evaluates the likelihood of an instance being a positive contributor across all positive bags while remaining absent in negative ones. This was implemented using an expectation-maximization (EM) algorithm for iterative optimization.

Foulds and Frank [43] expanded upon MIL by introducing the *collective MIL assumption*. Under this assumption, a positive bag may not be characterized by a single positive instance but rather by an interaction or distribution of instances. This collective perspective broadens the applicability of MIL, allowing it to model more complex relationships between instances within a bag.

MIL algorithms have further evolved to include techniques such as nearest-neighbor-based methods [44], which measure similarity between bags, and ensemble approaches like boosting [45], which combine weak classifiers to enhance prediction performance. Neural network-based methods [46] have also been proposed, enabling MIL to leverage nonlinear relationships between features.

In visual tasks, MIL has been extensively used for object detection and localization. Images are treated as bags containing potential object proposals (instances). The learning process alternates between training object classifiers and refining the selection of positive instances. Strategies such as initialization improvements [47], regularization with additional cues [48], and relaxed MIL constraints [49] have been developed to improve performance and address inherent challenges.

2.1.2 Max-Margin Models

Max-Margin Models introduce structured prediction methods leveraging latent variables. Yu and Joachims [50] pioneered a **Latent Structured Support Vector Machine (SVM)** for structured prediction tasks, utilizing the *concave-convex procedure (CCCP)* for optimization. This framework incorporates latent variables into the learning process to capture task-specific structures. By modeling latent variables, the method effectively handles hidden relationships within the data that are not directly observable.

Kumar *et al.* [47] advanced this approach by proposing *self-paced learning*, which iteratively selects easy samples to train latent SVMs. This curriculum-inspired approach mimics human learning, where simpler tasks are learned first, gradually increasing complexity. By focusing on "easier" samples early in training, self-paced learning mitigates the risk of local minima and improves model convergence.

In subsequent work, Kumar *et al.* [51] introduced a probabilistic framework for latent variable modeling. This approach uses two separate distributions to capture uncertainty over latent variables: one for training and another for inference. A *dissimilarity coefficient*-based loss encourages agreement between

these distributions, facilitating task-specific loss functions dependent on latent variables. For example, this framework allows seamless integration of structured losses like Intersection-over-Union (IoU) in segmentation tasks.

Other advancements include modeling output distributions conditioned on input and latent variables, as introduced by Miller *et al.* [52]. Their approach maximizes the margin between the Rényi entropies of correct and incorrect outputs, offering a principled way to handle ambiguity in weakly supervised tasks.

Max-margin models have found applications in object detection, semantic segmentation, and structured prediction, where modeling latent structures is critical. These methods' ability to incorporate task-specific constraints and loss functions makes them versatile for various WSL problems.

2.1.3 Probabilistic Graphical Models (PGMs)

Probabilistic Graphical Models (PGMs) provide a powerful framework for modeling dependencies among variables, making them suitable for WSL tasks. Boykov and Jolly [53] introduced a **graph-cut-based method** for interactive object segmentation. This method represents the image as a graph, where pixels are nodes and edges encode pairwise similarities. By minimizing a max-flow min-cut objective, the algorithm segments objects with minimal user interaction.

Rother *et al.* [54] extended this concept with **GrabCut**, a semi-automatic segmentation method. GrabCut initializes segmentation using user-provided bounding boxes or masks and iteratively refines the result using a **Gaussian Mixture Model (GMM)** for color distributions and a **Markov Random Field (MRF)** to enforce spatial smoothness. This iterative process ensures accurate and consistent segmentation boundaries.

Krähenbühl and Koltun [55] proposed a novel approach to fully connected conditional random fields (**dense CRFs**), introducing efficient inference algorithms for models with pairwise potentials defined over all pixel pairs. Dense CRFs capture fine-grained dependencies across the entire image, enabling high-resolution predictions for tasks like semantic segmentation.

PGMs have also been employed in tasks like object detection and scene understanding, where interactions among multiple components (e.g., objects, background, and context) are critical. These models excel in scenarios where domain-specific priors can be encoded as graphical structures, offering interpretability and flexibility.

2.1.4 Discussion

Classical WSL approaches have been instrumental in laying the foundation for weak supervision in machine learning. These methods primarily utilize two categories of cues:

- **Bottom-Up Cues**: These include region saliency, objectness, intra-class consistency, and inter-class discriminability. Such cues guide the model to learn discriminative features from the data, leveraging inherent properties of the input.

- **Top-Down Cues**: These provide high-level priors for appearance, structure, or semantic relationships, aiding the model in refining predictions and resolving ambiguities.

The advantages of classical WSL methods lie in their simplicity and efficiency. They require only small-scale training datasets, are computationally lightweight, and are relatively easy to implement. However, their reliance on handcrafted features limits their performance, as they lack the representational power and complexity of modern deep-learning-based methods. Additionally, they may struggle with generalization to complex, large-scale datasets.

Despite these limitations, ideas such as MIL frameworks, latent variable modeling, structured prediction, and uncertainty estimation remain integral to state-of-the-art WSL techniques. Classical WSL approaches have thus provided a strong conceptual and methodological foundation for subsequent advancements in the field.

2.2 Off-the-shelf Deep Model-based Approaches

This section examines WSL methods that leverage classical formulations alongside feature representations derived from deep neural networks. These approaches can be broadly categorized into three main types: (1) **off-the-shelf deep models** that utilize features extracted from pre-trained deep neural networks and train classical models on top of them, (2) **inherent cues from deep models** that exploit intermediate activations and semantic scores from neural networks, and (3) **fine-tuned deep models**, where pre-trained models are fine-tuned for specific target domains. Each of these categories represents a distinct way of integrating deep learning into weakly supervised frameworks.

2.2.1 Pre-trained Deep Features

Methods under this category rely on features extracted from pre-trained deep models to address weakly supervised learning tasks. For instance, Song *et al.* [56] leverage features from the DeCAF network [57] to address the challenge of object localization with minimal supervision. Their approach combines a **discriminative submodular cover problem** with a **smooth latent SVM** formulation, showcasing the utility of high-level features in identifying objects in weakly supervised scenarios. Similarly, Wang *et al.* [49] propose a relaxed formulation of multiple instance learning (MIL), which is differentiable and optimized using stochastic gradient descent (SGD). By using features extracted from the FC6 layer of AlexNet [21], they effectively demonstrate how pre-trained features can improve object discovery.

Ren *et al.* [58] contribute to this category with an **MIL-based bag-splitting algorithm** designed to reduce ambiguity in positive image bags. This iterative method generates new negative bags and utilizes AlexNet features to enhance object localization accuracy. Another significant contribution is by Cinbis *et al.* [59], who present a **multi-fold MIL objective** to avoid degenerate solutions. Their work incorporates contrastive background descriptors and Fisher Vectors, alongside AlexNet features, to improve the precision of object localization.

2.2.2 Inherent Cues from Deep Models

This category focuses on exploiting inherent properties of deep neural networks for weakly supervised learning. Oquab *et al.* [60] highlight the dual utility of fully convolutional neural networks trained for image classification tasks, demonstrating their ability to localize objects within images without additional supervision. This underscores the potential of classification-trained networks for spatial reasoning.

Building on this, Bency *et al.* [61] leverage spatial and semantic patterns captured in convolutional layers to propose an efficient beam search-based approach. Their method uses **deep feature maps** to localize multiple objects in images. Zhou *et al.* [62] introduce **Class Activation Maps (CAMs)** for CNNs with global average pooling (GAP). CAMs provide an interpretable mechanism to visualize discriminative object parts detected by CNNs, bridging the gap between model performance and human understanding.

Selvaraju *et al.* [63] expand on the concept of CAMs with **Grad-CAM**, a **gradient-weighted class activation map** technique. Grad-CAM offers visual explanations for CNN-based networks without requiring architectural changes, making it versatile for various tasks, including classification and visual question answering (VQA). This technique has become an essential tool for interpreting deep learning models.

2.2.3 Fine-tuned Deep Models

Fine-tuning pre-trained models for specific target domains represents another key approach in weakly supervised frameworks. Chen and Gupta [64] propose a two-stage methodology. Initially, a CNN is trained on easy images from Google search, followed by fine-tuning on complex images from Flickr. This process refines learned representations, enabling better generalization to challenging datasets.

Li *et al.* [65] introduce a two-stage domain adaptation process for object localization. The first stage involves classification adaptation, which refines object proposals, while the second stage employs a mask-out strategy to generate class-specific object proposals and mine confident candidates. Shi *et al.* [66] adopt a knowledge transfer approach, utilizing the concepts of "things" and "stuff" from a source dataset to improve object localization on a target dataset. Their iterative training of Fast RCNN models with semantic segmentation as pseudo-labels demonstrates the power of leveraging prior knowledge.

Khoreva *et al.* [67] iteratively train CNNs for semantic segmentation by generating pseudo-labels using a grabcut-like algorithm. This iterative refinement approach underscores the potential of combining weak supervision with algorithmic feedback loops to achieve accurate segmentation results.

2.2.4 Discussion

The integration of off-the-shelf deep models into weakly supervised learning frameworks has yielded several important insights. High-level features extracted from pre-trained models significantly enhance the weakly supervised learning process by providing rich data representations. CNN models trained

with image-level supervision are particularly adept at inferring discriminative spatial locations, making them invaluable for tasks like object localization and segmentation. Additionally, pre-training on large-scale auxiliary datasets has proven to be a simple yet effective strategy for encoding valuable cues that benefit weak supervision.

However, off-the-shelf approaches are not without limitations. They often lack the adaptability and specificity of end-to-end trainable deep weakly supervised models tailored for particular tasks. For example, the loss of error propagation and the absence of task-specific architectural optimization reduce their effectiveness in applications requiring precise localization or semantic segmentation. Future research focuses on bridging these gaps by integrating the strengths of pre-trained features with task-specific end-to-end learning approaches, thereby enhancing the overall capabilities of weakly supervised learning frameworks.

2.3 Deep Weakly Supervised Learning Frameworks

Deep weakly supervised learning frameworks represent a significant advancement over earlier methods by adopting an end-to-end training paradigm. This allows for the simultaneous learning of feature representations and task-specific networks, leveraging the power of deep neural networks. These approaches can broadly be categorized into two types: (i) **single-network training** approaches and (ii) **multi-network training** approaches.

2.3.1 Single-Network Training Approaches

In single-network training, the learning framework employs a **single deep neural network to jointly optimize feature representation and task objectives**. Pinheiro and Collobert [68] introduced a CNN-based model that emphasizes important pixels for classification. By incorporating image-level priors and smoothing constraints, the model effectively transitions from image-level to pixel-level labeling. Similarly, Pathak, Krahenbuhl, and Darrell [69] proposed a constrained optimization framework for CNNs, enabling the optimization of linear constraints using stochastic gradient descent. This method significantly improved weakly supervised segmentation performance.

Papandreou *et al.* [70] presented an expectation-maximization (EM)-based training algorithm capable of handling both weak and strong annotations. This provides a flexible framework for hybrid supervision in semantic segmentation. Expanding on this idea, Kolesnikov and Lampert [71] proposed a loss function guided by three principles: seeding with weak localization cues, expanding objects based on class occurrence, and constraining segmentation to align with object boundaries.

Bilen and Vedaldi [72] introduced a two-stream architecture that combined classification and detection streams, merging their outputs for optimized weakly supervised object detection. Diba *et al.* [73] extended this idea with a three-stage cascaded CNN to identify discriminative regions, compute object segmentation, and perform multiple instance learning (MIL) for object detection. Similarly, Vernaza and

Chandraker [74] introduced a differentiable random-walk-based label propagation algorithm, enhancing segmentation performance by learning pixel affinities.

Singh *et al.* [75] presented the "hide-and-seek" data augmentation technique, which randomly hides patches in training images to force the model to discover less discriminative but relevant regions. Building on the idea of pixel-level optimization, Ahn and Kwak [76] introduced AffinityNet, which generates semantic affinity labels using class activation maps (CAMs) and leverages these labels to produce segmentation masks.

2.3.2 Multi-Network Training Approaches

Multi-network training leverages **multiple specialized networks that collaborate to enhance the overall learning** process. Tang *et al.* [77] combined a weakly supervised detection network (WSDDN) with a multi-stage instance classifier for progressive refinement. The pseudo-labels generated by this process were then used to train Fast RCNN [1], further improving detection performance.

Ge, Yang, and Yu [78] introduced a curriculum learning approach for intermediate labeling and employed metric-learning and density-based clustering algorithms for object localization. This approach enabled more precise localization of objects in the training images. To address challenges in identifying less obvious regions, Zhang *et al.* [79] proposed the mean Energy Accumulated Scores (mEAS) criterion to measure image difficulty. By using feature masking, the network was guided to focus on less discriminative regions, thereby improving object detection performance.

Tang *et al.* [80] developed a weakly supervised region proposal network with a two-stage process. The first stage generated coarse proposals based on objectness scores, while the second stage refined these proposals to improve detection accuracy. Extending this line of work, Zhang *et al.* [81] introduced the W2F framework, which employs a pseudo-ground-truth excavation algorithm to refine bounding boxes. A fully supervised object detection model [1, 3] was then trained using these refined bounding boxes, achieving high detection precision.

2.3.3 Discussion

Deep weakly supervised learning frameworks combine the strengths of deep learning and weakly supervised learning. Single-network approaches simplify the process by introducing mechanisms like MIL and leveraging end-to-end training, which eliminates the need for complex initialization stages. These methods are particularly suited for tasks requiring efficient and scalable solutions without heavy computational overhead.

Multi-network approaches, on the other hand, enhance performance by integrating multiple task-specific networks. These methods benefit from the collaborative power of specialized components, which can complement each other to achieve superior results. The synergy between these components allows for more robust learning and better handling of complex tasks.

However, the effectiveness of these methods is often constrained by the quality of information extracted from the weakly supervised components. Incorporating prior knowledge into the training process

could significantly improve performance by reducing reliance on the weakly supervised modules alone. Future research could focus on hybrid strategies that combine the advantages of both single-network and multi-network approaches while leveraging external priors to guide the learning process. Such advancements would open new possibilities for tackling large-scale, weakly annotated datasets with greater efficiency and accuracy.

Chapter 3

Dissimilarity Coefficient based Weakly Supervised Learning Framework

In this chapter, we present a weakly supervised probabilistic framework based on the dissimilarity coefficient loss, a novel approach to addressing challenges in learning with limited or imprecise supervision. The chapter begins with a a discussion of the preliminaries that establish the foundation for the proposed framework. In the preliminaries, we formally define the dissimilarity coefficient objective and review key prior works that have directly influenced this research, emphasizing their relevance to weakly supervised learning.

Building on this foundation, we proceed to formally define the proposed probabilistic framework, which integrates the dissimilarity coefficient loss within a structured design to effectively model and utilize weakly labeled data. By addressing gaps identified in prior research and leveraging the strengths of the dissimilarity coefficient, our framework offers a scalable and robust solution for weakly supervised learning scenarios.

3.1 Preliminaries

This section introduces the key concepts and objectives underlying the proposed framework. We first define the dissimilarity coefficient objective, the central component of this work, and then review foundational studies that have motivated its development, setting the stage for the contributions detailed in subsequent sections.

3.1.1 Rao's Dissimilarity Coefficient

Dissimilarity and diversity are key concepts in understanding the relationships within and between distributions. **Dissimilarity** measures the separation or distinction between two distributions, quantifying how different they are from each other. In contrast, **diversity** captures the variability or heterogeneity within a single distribution, reflecting how diverse its samples are. These measures are particularly important in tasks where understanding both the internal structure of a distribution and the relationship between multiple distributions is essential, such as clustering, classification, and generative modeling.

Rao [82] introduces the *dissimilarity coefficient* as a measure of the difference between two distributions $\Pr_1(\cdot)$ and $\Pr_2(\cdot)$, combining their mutual differences with adjustments for their internal

variability. The *diversity coefficient* $DIV_\Delta(\mathrm{Pr}_1, \mathrm{Pr}_2)$ quantifies the expected difference between two distributions based on a task-specific loss function $\Delta(\cdot, \cdot)$, which measures the difference between samples drawn from these distributions. Formally, the diversity coefficient is defined as:

$$\begin{aligned} DIV_\Delta(\mathrm{Pr}_1, \mathrm{Pr}_2) &= \mathbb{E}_{\mathbf{y}_1 \sim \mathrm{Pr}_1(\cdot)}[\mathbb{E}_{\mathbf{y}_2 \sim \mathrm{Pr}_2(\cdot)}[\Delta(\mathbf{y}_1, \mathbf{y}_2)]] \\ &= \sum_{\mathbf{y}_1 \in \mathcal{Y}} \sum_{\mathbf{y}_2 \in \mathcal{Y}} \Delta(\mathbf{y}_1, \mathbf{y}_2) \, \mathrm{Pr}_1(\mathbf{y}_1) \, \mathrm{Pr}_2(\mathbf{y}_2), \end{aligned} \tag{3.1}$$

where $\mathcal{Y}$ is the sample space, and $\Delta(\mathbf{y}_1, \mathbf{y}_2)$ is a symmetric, non-negative function capturing the difference between two output samples $\mathbf{y}_1$ and $\mathbf{y}_2$. For a single distribution, the diversity $DIV_\Delta(\mathrm{Pr}_1, \mathrm{Pr}_1)$ represents the heterogeneity within $\mathrm{Pr}_1(\cdot)$ by averaging pairwise differences between its samples.

Using the diversity coefficient, the dissimilarity coefficient between two distributions $\mathrm{Pr}_1(\cdot)$ and $\mathrm{Pr}_2(\cdot)$ is defined as:

$$\begin{aligned} DISC_\Delta(\mathrm{Pr}_1, \mathrm{Pr}_2) = DIV_\Delta(\mathrm{Pr}_1, \mathrm{Pr}_2) &- \gamma DIV_\Delta(\mathrm{Pr}_2, \mathrm{Pr}_2) \\ &- (1-\gamma) DIV_\Delta(\mathrm{Pr}_1, \mathrm{Pr}_1), \end{aligned} \tag{3.2}$$

where $\gamma \in [0, 1]$ controls the relative contribution of the self-diversity terms. The self-diversities $DIV_\Delta(\mathrm{Pr}_1, \mathrm{Pr}_1)$ and $DIV_\Delta(\mathrm{Pr}_2, \mathrm{Pr}_2)$ measure the internal variability within $\mathrm{Pr}_1(\cdot)$ and $\mathrm{Pr}_2(\cdot)$, respectively. By subtracting a convex combination of self-diversities, the dissimilarity coefficient captures the "extra diversity" between the two distributions beyond their internal heterogeneity.

Rao, in their paper, uses $\gamma = 1/2$, which ensures symmetry, i.e., $DISC_\Delta(\mathrm{Pr}_1, \mathrm{Pr}_2) = DISC_\Delta(\mathrm{Pr}_2, \mathrm{Pr}_1)$. The formulation above, however, is more generic and allows flexibility in defining the relative importance of each self-diversity term.

In this context:

- **Self-Diversity** ($DIV_\Delta(\mathrm{Pr}_1, \mathrm{Pr}_1)$ or $DIV_\Delta(\mathrm{Pr}_2, \mathrm{Pr}_2)$): Measures the internal variability within a single distribution, reflecting how heterogeneous its samples are.
- **Cross-Diversity** ($DIV_\Delta(\mathrm{Pr}_1, \mathrm{Pr}_2)$): Captures the average difference between two distributions, quantifying how distinct their samples are in comparison to each other.
- **Dissimilarity** ($DISC_\Delta(\mathrm{Pr}_1, \mathrm{Pr}_2)$): Quantifies the distinction between two distributions, combining their cross-diversity with adjustments for their internal heterogeneity.

The dissimilarity coefficient has desirable properties:

- **Guarantee**: $DISC_\Delta(\mathrm{Pr}_1, \mathrm{Pr}_2) = 0 \iff \mathrm{Pr}_1(\cdot) = \mathrm{Pr}_2(\cdot)$ within the domain of the two distributions, given a symmetric, non-negative loss function $\Delta(\cdot, \cdot)$ appropriate for the task at hand.
- **Symmetry**: When $\gamma = 1/2$, the measure is symmetric, ensuring that $DISC_\Delta(\mathrm{Pr}_1, \mathrm{Pr}_2) = DISC_\Delta(\mathrm{Pr}_2, \mathrm{Pr}_1)$.

- **Non-Negativity**: The coefficient is non-negative ($DISC_{\Delta}(\mathrm{Pr}_1, \mathrm{Pr}_2) \geq 0$), as it is derived from a Jensen-type difference.

- **Flexibility**: By choosing a suitable loss function $\Delta(\cdot, \cdot)$ and varying γ, the framework can adapt to various application contexts.

This framework encourages the alignment of $\mathrm{Pr}_1(\cdot)$ and $\mathrm{Pr}_2(\cdot)$ by penalizing their dissimilarity, while the self-diversity terms promote meaningful variability within each distribution.

While the dissimilarity coefficient offers a principled way to compare and align distributions, its utility extends beyond analysis — it can also serve as a powerful training objective. In modern machine learning, model misspecification is the norm: no model class perfectly captures the true data-generating process. Under such conditions, there is no uniquely correct parameterization; the model's behavior is dictated entirely by the optimization criterion. Crucially, most models are trained using generic divergences such as Kullback–Leibler (KL) or Jensen–Shannon (JS) divergence, which prioritize fidelity to the full data distribution. These objectives penalize discrepancies uniformly, even in regions that do not influence downstream task performance—leading to inefficient or even counterproductive fits in practice.

The dissimilarity coefficient objective (DISCO) addresses this limitation by embedding the task-specific loss function Δ directly into the learning objective. This ensures that the model prioritizes aspects of the predictive distribution that actually matter for evaluation (e.g., mAP, IoU, or domain-specific metrics), while allowing flexibility in dimensions irrelevant to the task. Moreover, the tunable parameter γ governs a trade-off between mode-seeking and diversity, encouraging either sharper or more spread-out predictions as required. In contrast to KL and JS, which lack such adaptability, DISCO enables models to remain calibrated and robust under misspecification by aligning training with the true evaluation criterion. As a result, it forms a theoretically grounded and practically effective foundation for training task-aware probabilistic models.

3.1.2 Modeling Latent Variables for Loss-Based Learning

Kumar *et al.* [51] propose a framework based on a dissimilarity coefficient objective to address the challenges of weakly supervised training by explicitly modeling uncertainty over the latent variables. The approach separates the tasks of modeling uncertainty during training and making accurate predictions during testing through two distinct distributions:

- A **conditional distribution**, a log-linear model parameterized by θ, captures the uncertainty in latent variables given an input-output pair. This distribution models variability in latent variables and incorporates loss functions that depend on both the output and latent variables.

- A **delta distribution**, parameterized by $\mathbf{w}$, provides pointwise predictions for the output and latent variables given an input, ensuring accurate test-time predictions.

To align these distributions, Kumar *et al.* [51] minimize a loss-based dissimilarity coefficient inspired by Rao's framework [82]. This alignment ensures that the delta distribution's predictions not only match the observed outputs but also conform to high-probability configurations of the conditional distribution. While similar to the latent structured support vector machine (LSSVM) formulation [50], this framework generalizes it by:

1. Modeling uncertainty over latent variables instead of relying solely on pointwise estimates; and

2. Allowing loss functions that depend on both the output and latent variables, enhancing flexibility and applicability.

The training objective is optimized using block coordinate descent. Starting with an initial parameter set, the framework alternates between fixing one set of parameters and optimizing the other. The optimization of $\mathbf{w}$ is performed using the concave-convex procedure (CCCP), while θ is updated through stochastic subgradient descent (SSD).

This framework extends traditional latent variable models by accommodating loss functions dependent on latent variables, making it well-suited for complex tasks such as object detection and action detection. Experiments on public data sets demonstrate its effectiveness, underscoring its potential for a wide range of weakly supervised learning applications.

While this framework provides a strong foundation for modeling latent variables, adapting it to complex distributions and implementing it within deep learning systems remains a significant challenge. This limitation motivates our proposed probabilistic framework, which aims to address these gaps.

3.1.3 DISCO Nets: Dissimilarity Coefficient Networks

Dissimilarity Coefficient Networks (DISCO Nets) [83] are probabilistic generative models designed to address the challenge of modeling uncertainty in predictions for structured output problems. They leverage deep neural networks to efficiently sample from posterior distributions while aligning the training process with task-specific objectives. By employing a dissimilarity coefficient, DISCO Nets minimize divergence between the true and predicted distributions and balance accuracy with diversity in predictions through a tunable parameter, γ. This makes them particularly effective in scenarios where output uncertainty plays a critical role.

DISCO Nets generate samples by taking input data $\mathbf{x}$ and random noise $\mathbf{z}$ as a pair. The input $\mathbf{x}$ is processed through several convolutional layers, and the output is flattened and concatenated with $\mathbf{z}$. This combined vector is then passed through dense layers to produce the output $\mathbf{y}$. By varying the noise $\mathbf{z}$, the model generates diverse output candidates, demonstrating flexibility in where noise can be incorporated within the network architecture. The DISCO Nets architecture is shown in figure 3.1.

To make a single prediction from these diverse candidates, DISCO Nets employ the principle of Maximum Expected Utility (MEU). This approach selects the prediction $\mathbf{y}_{\Delta_{\text{task}}}$ that minimizes the expected task-specific loss Δ_{task}, calculated across all sampled candidates. Formally, the prediction is

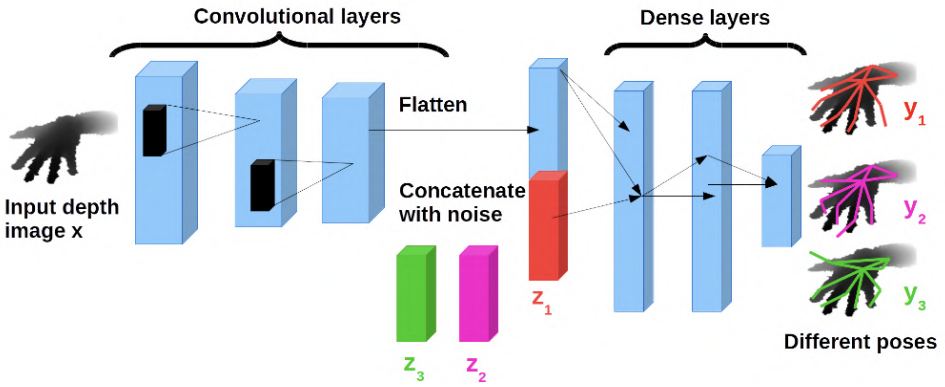


Figure 3.1 *For a single depth image* $\mathbf{x}$*, using 3 different noise samples* $(\mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3)$*, DISCO Nets output 3 different candidate poses* $(\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3)$ *(shown superimposed on depth image). Best viewed in color.*

expressed as:

$$\mathbf{y}_{\Delta_{\text{task}}} = \underset{k\in[1,K]}{\arg\min} \sum_{k'=1,k'\neq k}^{K} \Delta_{\text{task}}(\mathbf{y}^k, \mathbf{y}^{k'}), \tag{3.3}$$

where $\{\mathbf{y}^1, \ldots, \mathbf{y}^K\}$ are the candidate outputs generated for the input $\mathbf{x}$. This mechanism ensures that the selected prediction aligns closely with the task-specific objectives, balancing accuracy and uncertainty effectively.

The training objective of DISCO Nets is to minimize a dissimilarity coefficient that measures the divergence between the true data distribution P and the model's predicted distribution Q. This objective encourages the predicted distribution Q to align with P while maintaining diversity in the generated samples. The objective function is defined as:

$$\text{DISC}_\Delta(P, Q) = \text{DIV}_\Delta(P, Q) - \gamma \text{DIV}_\Delta(Q, Q), \tag{3.4}$$

where $\text{DIV}_\Delta(P, Q)$ represents the expected task-specific loss between samples from P and Q, and $\text{DIV}_\Delta(Q, Q)$ promotes diversity within Q. Note that the term $DIV_\Delta(P, P) = 0$ as P is a delta distribution since it denotes the true data distribution. The parameter $\gamma \in [0, 1]$ balances accuracy and diversity in predictions. This optimization is performed using gradient descent on the sampled candidates.

Unlike GANs or VAEs, DISCO Nets do not require adversarial training or specific network architectures, making them more robust and easier to implement. They outperform existing probabilistic and non-probabilistic models, including GANs, on tasks like hand pose estimation.

3.1.3.1 Discrete DISCO Nets: Making DISCO Nets Discrete

Discrete DISCO Nets [84] extend the original DISCO Nets in their ability to handle discrete and structured output spaces. For discrete outputs, the sampling process involves the use of a scoring function $\mathcal{S}^k_{\boldsymbol{\theta}}(\mathbf{y}_c)$, which evaluates each potential output $\mathbf{y}_c$ based on the input $\mathbf{x}$, random noise $\mathbf{z}^k$, and model

parameters $\boldsymbol{\theta}$. This scoring rule assigns a value to each candidate output, reflecting how well it aligns with the model's learned distribution. The sampling process is mathematically defined as:

$$\hat{\mathbf{y}}_c^k = \arg\max_{\mathbf{y}\in\mathcal{Y}} \mathcal{S}_{\boldsymbol{\theta}}^k(\mathbf{y}_c) \tag{3.5}$$

Here, the $\arg\max$ operator selects the output $\hat{\mathbf{y}}$ with the highest score, ensuring that the sampled output reflects the structured dependencies in the data. The inclusion of random noise $\mathbf{z}$ introduces variability into the process, promoting diversity in the predictions. In Discrete DISCO Nets, the loss is non-differentiable due to the $\arg\max$ operation in the sampling step, unlike continuous DISCO Nets where the loss is inherently differentiable. To address this, the differentiability of the scoring function is leveraged to estimate gradients. Direct loss minimization is used to approximate the gradient of the non-differentiable loss function, ensuring convergence to the true gradient and enabling effective optimization. This approach allows Discrete DISCO Nets to efficiently handle discrete outputs.

DISCO Nets and their discrete variant are versatile models capable of adapting to diverse applications by incorporating task-specific loss functions, making them particularly suitable for structured prediction tasks under uncertainty. Although these models show improved performance in such scenarios, their current formulation relies on supervised training, assuming the true data distribution P to be a delta distribution. To address this limitation, our proposed framework extends DISCO Nets to handle weakly supervised data and more complex data distributions, broadening their applicability and enhancing their capabilities for real-world problems.

3.2 Dissimilarity Coefficient based Weakly Supervised Learning Framework

This section introduces the proposed probabilistic framework for weakly supervised learning, which is built upon the dissimilarity coefficient as a central measure. We begin by defining the notations used throughout the framework, ensuring clarity and consistency in the mathematical formulations. Next, we formalize the learning objective, which focuses on minimizing the dissimilarity coefficient to align weakly supervised data distributions. Finally, we outline the optimization process for both continuous and discrete cases, highlighting key distinctions and ensuring adaptability to various learning scenarios.

3.2.1 Notation

Let an input image be denoted as $\mathbf{x} \in \mathbb{R}^{H\times W\times 3}$, where H and W represent the height and width of the image, respectively. Coarse or weak annotations are denoted by $\mathbf{a}$, while the fine-grained, task-specific prediction label is represented by $\mathbf{y}$. A weakly supervised data set $\mathcal{W} = \{(\mathbf{x}_i, \mathbf{a}_i)|i = 1, \ldots, N\}$ consists of N image-annotation pairs, where $\mathbf{x}_i$ is an image and $\mathbf{a}_i$ its corresponding weak annotation. A probability distribution is denoted as $\Pr(\cdot|\cdot;\boldsymbol{\theta})$, where $\boldsymbol{\theta}$ are the parameters of the distribution. We

define the task-specific loss function $\Delta(\mathbf{y}_1, \mathbf{y}_2)$ as a non-negative, symmetric function that quantifies the similarity between two predictions, $\mathbf{y}_1$ and $\mathbf{y}_2$, for a given task.

3.2.2 Probabilistic Modeling

Given the weakly supervised data set $\mathcal{W}$, our goal is to learn a task-specific supervised model capable of predicting the fine-grained label $\mathbf{y}$ for a previously unseen image $\mathbf{x}$. To address the challenges of learning from coarse annotations or weak labels, we propose a probabilistic framework. Building on the work of Kumar *et al.* [51] (Section 3.1.2), we define two key probability distributions: the **prediction distribution** and the **conditional distribution**.

The **prediction distribution**, $\Pr_p(\mathbf{y}|\mathbf{x};\boldsymbol{\theta}_p)$, models test-time predictions independently of the weak annotations $\mathbf{a}$. It is implemented using a *prediction net*, which typically adopts a state-of-the-art, task-specific, fully supervised model architecture. To enable sampling, the prediction net is either converted into a probabilistic network using DISCO Nets [83] (Section 3.1.3) or its outputs are interpreted probabilistically. The parameters of this distribution, $\boldsymbol{\theta}_p$, correspond to the weights of the neural network.

The **conditional distribution**, $\Pr_c(\mathbf{y}|\mathbf{a},\mathbf{x};\boldsymbol{\theta}_c)$, captures the uncertainty in generating fine-grained labels based on the weak annotations $\mathbf{a}$ and the input image $\mathbf{x}$. It is modeled using a probabilistic *conditional net*, which takes the training image and weak annotation as input and outputs the fine-grained predictions. To ensure diverse sampling from this distribution, we employ DISCO Nets [83] (Section 3.1.3). The parameters of this distribution, $\boldsymbol{\theta}_c$, are also represented by the weights of the neural network.

The conditional distribution provides additional information by ensuring that the fine-grained labels are consistent with the weak annotations. This alignment embeds task-specific constraints and enhances the representation of uncertainty in the fine-grained labels. The key to the success of the framework lies in accurately modeling the conditional distribution so that it makes precise task-specific predictions. During training, this enriched information from the conditional distribution is leveraged to guide the learning of the prediction network, enabling it to achieve higher accuracy in generating fine-grained predictions at test time. Note that, unlike Kumar *et al.* [51] (Section 3.1.2), we allow the two distributions to be arbitrarily complex.

3.2.3 Learning Objective

Given the weakly supervised dataset $\mathcal{W}$ and the prediction and conditional distributions $\Pr_p$ and $\Pr_c$, our objective is to learn the parameters $\boldsymbol{\theta}_p$ and $\boldsymbol{\theta}_c$ such that information from the conditional distribution is effectively transferred to the prediction distribution. Due to the inherent task similarity between the two distributions, we aim to bring them closer, ensuring that the rich information embedded in the conditional distribution is utilized to improve the prediction distribution. Inspired by Kumar *et al.* [51] (Section 3.1.2), we formulate a joint learning objective that minimizes the dissimilarity coefficient [82] (Section 3.1.1) between the prediction and conditional distributions. Formally, we specify our learning

objective as,

$$\boldsymbol{\theta}_p^*, \boldsymbol{\theta}_c^* = \arg\min_{\boldsymbol{\theta}_p, \boldsymbol{\theta}_c} \sum_{i=1}^{N} DISC_{\Delta}(\Pr_p(\mathbf{y}|\mathbf{x}; \boldsymbol{\theta}_p), Pr_c(\mathbf{y}|\mathbf{a}, \mathbf{x}; \boldsymbol{\theta}_c)). \tag{3.6}$$

Here $DISC(\Pr_p(\cdot), \Pr_c(\cdot))$ is the dissimilarity coefficient between the two distribution $\Pr_p(\cdot)$ and $\Pr_c(\cdot)$ as defined in (3.2).

In other words, the training objective encourages the prediction distribution and the conditional distribution to align closely (i.e., have a small dissimilarity coefficient) for all training images. By minimizing the dissimilarity, the learning objective ensures that the prediction distribution assigns a high probability to task-specific labels $\mathbf{y}$ that are consistent with the given weak annotations. During testing, only the prediction distribution is used to infer the pose of a given image.

Computing the objective (3.6) requires evaluating the diversity coefficient terms (see (3.1)), which involve calculating the expected loss over all pairs of $\mathbf{y}$, where $\mathbf{y} \in \mathcal{Y}$. This computation is infeasible due to its combinatorial complexity. However, an unbiased estimate of these terms, as well as their gradients, can be obtained by sampling from the distributions $\Pr_p(\cdot)$ and $\Pr_c(\cdot)$. Using DISCO Nets, we draw K samples $\mathbf{y}_c^k$ from the conditional net.

For the prediction net, there are two scenarios. First, we can employ DISCO Nets to draw K samples $\mathbf{y}_p^k$ directly from the network. Alternatively, if a vanilla supervised model is used, we interpret its output probabilistically by selecting $\mathbf{y}_p$ with a prediction probability of $\Pr_p(\mathbf{y}_p; \boldsymbol{\theta}_p)$. This flexibility allows the framework to adapt to different modeling approaches.

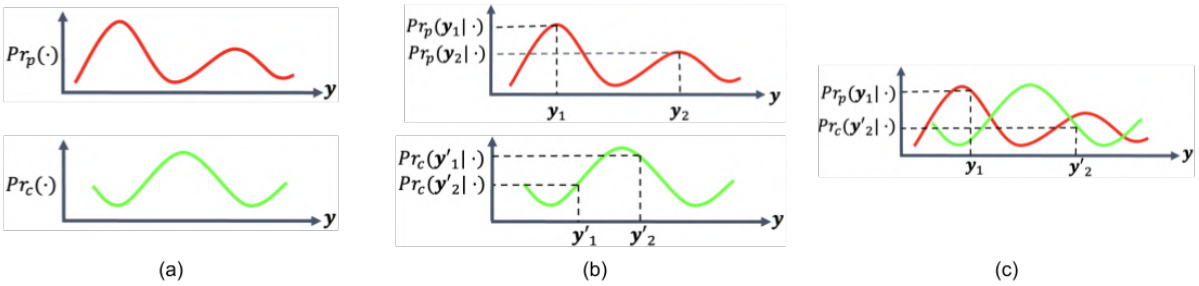


Figure 3.2 *The figure shows two sample distribution* $\Pr_p$ *and* $\Pr_c$*. Minimizing the dissimilarity coefficient minimizes the cross-diversity term (c) and maxmimizes the self-diversity terms. This encourages the samples obtained from the individual distributions in (b) to be spread out, while encouraging the samples between the distributions (c) to be close to each other.*

The dissimilarity coefficient objective (3.6) seeks to balance two competing forces as shown in figure 3.2. First, it maximizes the "self-diversity" of each distribution. This encourages samples drawn from either $\Pr_p$ or $\Pr_c$ to spread out and explore their support thoroughly. Second, it minimizes the "cross-diversity" between the distributions, ensuring paired samples from $\Pr_p$ and $\Pr_c$ are drawn closer together. In practice, this means the model is encouraged to generate a rich variety of outcomes within

each distribution (avoiding mode collapse or overly narrow predictions) but also to align corresponding samples across distributions, ensuring that they remain tightly coupled. By jointly optimizing these terms, the objective both preserves internal variability and enforces cross-distribution agreement.

To prevent degenerate solutions, a task-specific, non-negative loss function is carefully chosen. This loss is well-defined for all valid inputs and yields zero only when the model makes a perfect prediction for the task, assigning a positive value otherwise. As a result, the model can compute a meaningful diversity coefficient. Additionally, the conditional distribution is thoughtfully constructed by incorporating implicit or explicit priors about the task, guiding the model toward plausible outputs that reflect domain-specific structure. This informed design prevents the distribution from collapsing into a delta function and avoids degenerate scenarios where both the prediction and the conditional distribution become overly deterministic.

In the following subsections, we derive stochastic unbiased estimators for both the self-diversity and cross-diversity terms.

3.2.3.1 Cross-Diversity between the Prediction Net and the Conditional Net

Following equation (3.1), the diversity between the prediction and the conditional distribution can be written as,

$$DIV_{\Delta}(\Pr_p, \Pr_c) = \mathbb{E}_{\mathbf{y}_p \sim \Pr_p(\mathbf{y}|\mathbf{x};\boldsymbol{\theta}_p)}\Big[\mathbb{E}_{\mathbf{y}_c \sim \Pr_c(\mathbf{y}|\mathbf{x},\mathbf{h};\boldsymbol{\theta}_c)}[\Delta(\mathbf{y}_p, \mathbf{y}_c)]\Big]. \tag{3.7}$$

Rewriting the expectation with respect to the conditional distribution (the inner distribution) as expectation over the random variables $\mathbf{z}$ with distribution $\Pr(\mathbf{z})$ using the Law of the Unconscious Statistician (LOTUS).

$$DIV_{\Delta}(\Pr_p, \Pr_c) = \mathbb{E}_{\mathbf{y}_p \sim \Pr_p(\mathbf{y}|\mathbf{x};\boldsymbol{\theta}_p)}\Big[\mathbb{E}_{\mathbf{z} \sim \Pr(\mathbf{z})}[\Delta(\mathbf{y}_p, \mathbf{y}_c^k)]\Big]. \tag{3.8}$$

The expectation over the random variable $\mathbf{z}$ with distribution $\Pr(\mathbf{z})$ is approximated by taking K samples from $\Pr(\mathbf{z})$,

$$DIV_{\Delta}(\Pr_p, \Pr_c) = \mathbb{E}_{\mathbf{y}_p \sim \Pr_p(\mathbf{y}|\mathbf{x};\boldsymbol{\theta}_p)}\Big[\frac{1}{K}\sum_{k=1}^{K}\Delta(\mathbf{y}_p, \mathbf{y}_c^k)\Big]. \tag{3.9}$$

Finally the expectation with respect to the prediction distribution is computed as,

$$DIV_{\Delta}(\Pr_p, \Pr_c) = \frac{1}{K}\sum_{k=1}^{K}\sum_{\mathbf{y}_p^{(i)}}\Pr_p(\mathbf{y}_p^{(i)};\boldsymbol{\theta}_p)\Delta(\mathbf{y}_p, \mathbf{y}_c^k). \tag{3.10}$$

When DISCO Nets [83] are employed to draw K samples from the prediction net, the expectation is written as,

$$DIV_{\Delta}(\Pr_p, \Pr_c) = \frac{1}{K^2}\sum_{k=1}^{K}\sum_{k'=1}^{K}\Delta(\mathbf{y}_p^{k'}, \mathbf{y}_c^k). \tag{3.11}$$

3.2.3.2 Self-Diversity of Conditional Net

Following the approach outlined above and referencing equation (3.1), the diversity coefficient of the conditional distribution is expressed as,

$$DIV_{\Delta}(\Pr_c, \Pr_c) = \mathbb{E}_{\mathbf{y}_c \sim \Pr_c(\mathbf{y}|\mathbf{x},\mathbf{h};\boldsymbol{\theta}_c)}[\mathbb{E}_{\mathbf{y}'_c \sim \Pr_c(\mathbf{y}|\mathbf{x},\mathbf{h};\boldsymbol{\theta}_c)}[\Delta(\mathbf{y}_c, \mathbf{y}'_c)]]. \tag{3.12}$$

Now, rewriting the two expectations with respect to the conditional distribution as the expectation over the random variables $\mathbf{z}$ and $\mathbf{z}'$ respectively, we obtain the above equation as,

$$DIV_{\Delta}(\Pr_c, \Pr_c) = \mathbb{E}_{\mathbf{z} \sim \Pr(\mathbf{z})}\Big[\mathbb{E}_{\mathbf{z}' \sim \Pr(\mathbf{z})}[\Delta(\mathbf{y}_c^k, \mathbf{y}_c^{k'})]\Big]. \tag{3.13}$$

In order to approximate the expectation over the random variables $\mathbf{z}$ and $\mathbf{z}'$, we use K samples from the distribution $\Pr(\mathbf{z})$ as,

$$DIV_{\Delta}(\Pr_c, \Pr_c) = \frac{1}{K}\sum_{k=1}^{K}\frac{1}{K-1}\sum_{\substack{k'=1,\\ k'\neq k}}^{K}\Delta(\mathbf{y}_c^k, \mathbf{y}_c^{k'}). \tag{3.14}$$

On re-arranging the above equation, we get,

$$DIV_{\Delta}(\Pr_c, \Pr_c) = \frac{1}{K(K-1)}\sum_{\substack{k,k'=1\\ k'\neq k}}^{K}\Delta(\mathbf{y}_c^k, \mathbf{y}_c^{k'}). \tag{3.15}$$

3.2.3.3 Self-Diversity of Prediction Net

Analogous to previous two cases, and using equation (3.1), the diversity coefficient of the prediction net can be expressed as,

$$DIV_{\Delta}(\Pr_p, \Pr_p) = \mathbb{E}_{\mathbf{y}_p \sim \Pr_p(\mathbf{y}|\mathbf{x};\boldsymbol{\theta}_p)}\Big[\mathbb{E}_{\mathbf{y'}_p \sim \Pr_p(\mathbf{y}|\mathbf{x};\boldsymbol{\theta}_p)}[\Delta(\mathbf{y}_p, \mathbf{y'}_p)]\Big]. \tag{3.16}$$

When the prediction distribution is modeled using a vanilla supervised model with a probabilistic interpretation of its output, the two expectations are computed as,

$$\begin{aligned} DIV_{\Delta}(\Pr_p, \Pr_p) &= \mathbb{E}_{\mathbf{y}_p \sim \Pr_p(\mathbf{y}'|\mathbf{x};\boldsymbol{\theta}_p)}\Big[\sum_{\mathbf{y'}_p}\Pr_p(\mathbf{y'}_p;\boldsymbol{\theta}_p)\Delta(\mathbf{y}_p, \mathbf{y'}_p)\Big], \\ &= \sum_{\mathbf{y}_p}\sum_{\mathbf{y'}_p}\Pr_p(\mathbf{y}_p;\boldsymbol{\theta}_p)\Pr_p(\mathbf{y'}_p;\boldsymbol{\theta}_p)\Delta(\mathbf{y}_p, \mathbf{y'}_p). \end{aligned} \tag{3.17}$$

When K samples are drawn from the prediction net using DISCO Nets [83] (like the conditional net above), the expression is formulated as,

$$DIV_{\Delta}(\Pr_p, \Pr_p) = \frac{1}{K(K-1)}\sum_{\substack{k,k'=1\\ k'\neq k}}^{K}\Delta(\mathbf{y}_p^k, \mathbf{y}_p^{k'}). \tag{3.18}$$

3.2.4 Optimization

The prediction and conditional distributions are modeled using deep neural networks, which makes our objective (3.6) well-suited for minimization via stochastic gradient descent. It is possible to jointly optimize the parameters of both the networks as shown in algorithm 1. However, computing the gradients of both networks simultaneously can be computationally and memory intensive. To address this, a more efficient *coordinate descent* strategy is employed to train the parameters of the two networks iteratively.

Algorithm 1 Joint Optimization over prediction and conditional net $\boldsymbol{\theta}_p, \boldsymbol{\theta}_c$

Input: Data set $\mathcal{W}$ and initial estimate $\boldsymbol{\theta}_p^0, \boldsymbol{\theta}_c^0$
  for $t = 1 \dots T$ *epochs* **do**
    Sample mini-batch of b training example pairs
    for $n = 1 \dots b$ **do**
      Sample K random noise vectors $\mathbf{z}_k$
      Generate K candidate output from $\Pr_c(\mathbf{a}, \mathbf{x}, \mathbf{z}_k)$ and $\Pr_p(\mathbf{x}, \mathbf{z}_k)$.
    end for
    Compute objective as given in equation (3.6) here.
    Update parameters $\mathbf{w}$ via SGD with momentum
  end for

The *coordinate descent* proceeds by iteratively fixing the parameters of the prediction net and learning the conditional net, followed by learning the prediction net for the fixed conditional net. The two steps of the iterative algorithm are described below.

3.2.4.1 Optimization over Prediction Net

As the parameters $\boldsymbol{\theta}_c$ of the conditional nets are fixed, the learning objective (3.6) of the prediction net results in a fully supervised training shown below,

$$\begin{aligned}\boldsymbol{\theta}_p^* &= \arg\min_{\boldsymbol{\theta}_p} \sum_{i=1}^{N} DISC_{\Delta}(\Pr_p(\mathbf{y}|\mathbf{x};\boldsymbol{\theta}_p), Pr_c(\mathbf{y}|\mathbf{a},\mathbf{x};\boldsymbol{\theta}_c)), \\ &= \arg\min_{\boldsymbol{\theta}_p} \sum_{i=1}^{N} DIV_{\Delta}(\Pr_p(\mathbf{y}|\mathbf{x};\boldsymbol{\theta}_p), Pr_c(\mathbf{y}|\mathbf{a},\mathbf{x};\boldsymbol{\theta}_c)) - \\ &\qquad \gamma DIV_{\Delta}(\Pr_p(\mathbf{y}|\mathbf{x};\boldsymbol{\theta}_p), \Pr_p(\mathbf{y}|\mathbf{x};\boldsymbol{\theta}_p)).\end{aligned} \tag{3.19}$$

Here $DIV_{\Delta}(\Pr_p(\mathbf{y}|\mathbf{x};\boldsymbol{\theta}_p), Pr_c(\mathbf{y}|\mathbf{a},\mathbf{x};\boldsymbol{\theta}_c))$ is the cross-diversity term and $DIV_{\Delta}(\Pr_p(\mathbf{y}|\mathbf{x};\boldsymbol{\theta}_p), \Pr_p(\mathbf{y}|\mathbf{x};\boldsymbol{\theta}_p))$ is the self-diversity term (Section 3.2.3).

The prediction net takes an image as input along with the K predictions sampled from the conditional net. These predictions are treated as pseudo ground-truth labels to compute the gradient of the prediction net objective (3.19). Since the objective (3.19) is differentiable with respect to the parameters $\boldsymbol{\theta}_p$, the prediction net is updated using stochastic gradient descent.

When the prediction distribution is modeled using a vanilla supervised model with a probabilistic interpretation of its output, the training process directly corresponds to supervised learning, guided by the objective specified in Equation (3.19).

Alternatively, if the prediction distribution is modeled using DISCO Nets [83], the optimization is performed as outlined in Algorithm 2.

Algorithm 2 Optimization over prediction net $\boldsymbol{\theta}_p$

Input: Data set $\mathcal{W}$ and initial estimate $\boldsymbol{\theta}_p^0$
  for $t = 1 \dots T$ *epochs* **do**
    Sample mini-batch of b training example pairs
    for $n = 1 \dots b$ **do**
      Sample K random noise vectors $\mathbf{z}^k$
      Generate K candidate output from $\Pr_c(\mathbf{a}, \mathbf{x}, \mathbf{z}^k; \boldsymbol{\theta}_c)$ and $\Pr_p(\mathbf{x}, \mathbf{z}^k; \boldsymbol{\theta}_p)$.
    end for
    Compute the objective as given in equation (3.19) here.
    Update parameters $\boldsymbol{\theta}_p$ via SGD with momentum
  end for

3.2.4.2 Optimization over Conditional Net

As in the previous section, when the parameters $\boldsymbol{\theta}_p$ of the prediction net are fixed, the learning objective (3.6) for the conditional net reduces to a fully supervised training setup, as shown below,

$$\begin{aligned}\boldsymbol{\theta}_p^* &= \arg\min_{\boldsymbol{\theta}_p} \sum_{i=1}^{N} DISC_{\Delta}(\Pr_p(\mathbf{y}|\mathbf{x};\boldsymbol{\theta}_p), Pr_c(\mathbf{y}|\mathbf{a},\mathbf{x};\boldsymbol{\theta}_c)), \\ &= \arg\min_{\boldsymbol{\theta}_p} \sum_{i=1}^{N} DIV_{\Delta}(\Pr_p(\mathbf{y}|\mathbf{x};\boldsymbol{\theta}_p), Pr_c(\mathbf{y}|\mathbf{a},\mathbf{x};\boldsymbol{\theta}_c)) - \\ &\qquad (1-\gamma) DIV_{\Delta}(Pr_c(\mathbf{y}|\mathbf{a},\mathbf{x};\boldsymbol{\theta}_c), Pr_c(\mathbf{y}|\mathbf{a},\mathbf{x};\boldsymbol{\theta}_c)).\end{aligned} \tag{3.20}$$

Here $DIV_{\Delta}(\Pr_p(\mathbf{y}|\mathbf{x};\boldsymbol{\theta}_p), Pr_c(\mathbf{y}|\mathbf{a},\mathbf{x};\boldsymbol{\theta}_c))$ is the cross-diversity term and $DIV_{\Delta}(Pr_c(\mathbf{y}|\mathbf{a},\mathbf{x};\boldsymbol{\theta}_c), Pr_c(\mathbf{y}|\mathbf{a},\mathbf{x};\boldsymbol{\theta}_c))$ is the self-diversity term (Section 3.2.3).

The conditional net receives the image, weak annotations, and predictions from the prediction net as input. These predictions serve as pseudo ground-truth labels for calculating the gradient of the conditional net objective (3.20). With the objective function being differentiable with respect to the parameters $\boldsymbol{\theta}_c$, the conditional net is optimized using stochastic gradient descent as shown in algorithm 3.

3.2.4.2.1 Optimization over Discrete Conditional Net When the conditional net operates in a discrete output space, the optimization process requires careful handling. Unlike the continuous case, the discrete nature of the conditional net necessitates approximations to compute gradients efficiently

Algorithm 3 Optimization over conditional net $\boldsymbol{\theta}_c$

Input: Data set $\mathcal{W}$ and initial estimate $\boldsymbol{\theta}_c^0$
  for $t = 1 \dots T$ *epochs* **do**
    Sample mini-batch of b training example pairs
    for $n = 1 \dots b$ **do**
      Sample K random noise vectors $\mathbf{z}^k$
      Generate K candidate output from $\Pr_p(\mathbf{x}, \mathbf{z}^k)$ and $\Pr_c(\mathbf{a}, \mathbf{x}, \mathbf{z}^k)$
    end for
    Compute the objective as given here in equation (3.20) here.
    Update parameters $\boldsymbol{\theta}_c$ via SGD with momentum
  end for

while preserving the essential properties of the learning objective. In the following, we describe the optimization process for the conditional net when modeled using discrete DISCO Nets.

A non-differentiable training procedure The conditional net is modeled using a Discrete DISCO Nets [84] (Section 3.1.3.1 which employs a sampling step from the scoring function $\mathcal{S}^k(\mathbf{y}_c)$ (see Equation (3.5), where the parameters of the discrete DISCO Nets, now representing the conditional distribution, are denoted by $\boldsymbol{\theta}_c$ instead of $\boldsymbol{\theta}$). This sampling step makes the objective function non-differentiable with respect to the parameters $\boldsymbol{\theta}_c$, even though the scoring function $\mathcal{S}^k(\mathbf{y}_c)$ itself is differentiable. However, as the prediction network is fixed, the above objective function reduces to the one used in Bouchacourt *et al.* [84] for fully supervised training. Therefore, adapting the optimization approach outlined by Bouchacourt *et al.* [84], we address this problem by estimating the gradients of our objective function using the temperature parameter ϵ as,

$$\nabla_{\boldsymbol{\theta}_c} DISC^{\epsilon}_{\Delta}(\Pr_p(\boldsymbol{\theta}_p), \Pr_c(\boldsymbol{\theta}_c)) = \pm \lim_{\epsilon \to 0} \frac{1}{\epsilon}(DIV^{\epsilon}_{\Delta}(\Pr_p, \Pr_c) - \gamma DIV^{\epsilon}_{\Delta}(\Pr_c, \Pr_c)) \tag{3.21}$$

where,

$$DIV^{\epsilon}_{\Delta}(\Pr_p, \Pr_c) = \mathbb{E}_{\mathbf{y}_p \sim \Pr_p(\boldsymbol{\theta}_p)}[\mathbb{E}_{\mathbf{z}_k \sim \Pr(\mathbf{z})}[\nabla_{\boldsymbol{\theta}_c} \mathcal{S}^k(\hat{\mathbf{y}}_a) - \nabla_{\boldsymbol{\theta}_c} \mathcal{S}^k(\hat{\mathbf{y}}_c)]] \tag{3.22}$$

$$DIV^{\epsilon}_{\Delta}(\Pr_c, \Pr_c) = \mathbb{E}_{\mathbf{z}_k \sim \Pr(\mathbf{z})}[\mathbb{E}_{\mathbf{z'}_k \sim \Pr(\mathbf{z})}[\nabla_{\boldsymbol{\theta}_c} \mathcal{S}^k(\hat{\mathbf{y}}_b) - \nabla_{\boldsymbol{\theta}_c} \mathcal{S}^{k'}(\hat{\mathbf{y}}'_c)]] \tag{3.23}$$

and,

$$\begin{aligned}
\hat{\mathbf{y}}_c &= \arg\max_{y \in \mathcal{Y}} \mathcal{S}^k(\mathbf{y}_c) \\
\hat{\mathbf{y}}'_c &= \arg\max_{y \in \mathcal{Y}} \mathcal{S}^{k'}(\mathbf{y}_c) \\
\hat{\mathbf{y}}_a &= \arg\max_{y \in \mathcal{Y}} \mathcal{S}^k(\mathbf{y}_c) \pm \epsilon \Delta(\mathbf{y}_p, \hat{\mathbf{y}}_c) \\
\hat{\mathbf{y}}_b &= \arg\max_{y \in \mathcal{Y}} \mathcal{S}^k(\mathbf{y}_c) \pm \epsilon \Delta(\hat{\mathbf{y}}_c, \hat{\mathbf{y}}'_c)
\end{aligned} \tag{3.24}$$

In this thesis, we fix the temperature parameter ϵ as $\epsilon = +1$.

Intuition for the gradient computation: We now present an intuitive explanation of the computation of gradient, as given in equation (3.21). For an input $\mathbf{x}$ and two noise samples $\mathbf{z}_k, \mathbf{z}_{k'}$, the conditional net outputs two scores $\mathcal{S}^k(\mathbf{y}_c)$ and $\mathcal{S}^{k'}(\mathbf{y}_c)$, with the corresponding maximum scoring outputs $\hat{\mathbf{y}}_c$ and $\hat{\mathbf{y}}'_c$. The model parameters $\boldsymbol{\theta}_c$ are updated via gradient descent in the negative direction of $\nabla_{\boldsymbol{\theta}_c} DISC^{\epsilon}_{\Delta}(\mathrm{Pr}_p(\boldsymbol{\theta}_p), \mathrm{Pr}_c(\boldsymbol{\theta}_c))$.

- The term $DIV^{\epsilon}_{\Delta}(\mathrm{Pr}_p, \mathrm{Pr}_c)$ updates the model parameters towards the maximum scoring prediction $\hat{\mathbf{y}}_c$ of the score $\mathcal{S}^k(\mathbf{y}_c)$ while moving away from $\hat{\mathbf{y}}_a$, where $\hat{\mathbf{y}}_a$ is the sample corresponding to the maximum loss augmented score $\mathcal{S}^k(\mathbf{y}_c) \pm \epsilon\Delta(\mathbf{y}_p, \hat{\mathbf{y}}_c)$ with respect to the fixed prediction distribution samples $\mathbf{y}_p$. This encourages the model to move away from the prediction, which provides high loss with respect to the pseudo ground truth labels.

- The term $\gamma DIV^{\epsilon}_{\Delta}(\mathrm{Pr}_c, \mathrm{Pr}_c)$ updates the model towards $\mathbf{y}_b$ and away from the $\hat{\mathbf{y}}_c$. Note the two negative signs giving the update in the positive direction. Here $\mathbf{y}_b$ is the sample corresponding to the maximum loss augmented score $\mathcal{S}^k(\mathbf{y}_c) \pm \epsilon\Delta(\hat{\mathbf{y}}_c, \hat{\mathbf{y}}'_c)$ with respect to the other prediction $\hat{\mathbf{y}}'_c$, encouraging greater diversity between $\hat{\mathbf{y}}_c$ and $\hat{\mathbf{y}}'_c$.

Training algorithm for conditional net: Pseudo-code for training the conditional network for a single sample from weakly supervised data is presented in algorithm 4 below. In algorithm 4, statements 1 to 3 describe the sampling process and computing the loss augmented prediction. We first sample K different predictions $\hat{\mathbf{y}}^k_c$ corresponding to each noise vector $\mathbf{z}_k$ in statement 2. For the sampled prediction $\hat{\mathbf{y}}^k_c$ we compute the maximum loss augmented score $\mathcal{S}^k(\mathbf{y}_c) \pm \epsilon\Delta(\mathbf{y}_p, \hat{\mathbf{y}}_c)$. This is then used to find the loss augmented prediction $\hat{\mathbf{y}}_a$ given in statement 3.

In order to compute the gradients of the self diversity of conditional distribution, we need to find the maximum loss augmented prediction $\mathbf{y}_b$. Here, the loss is computed between a pair of K different predictions of the conditional net that we have already obtained. This is shown by statements 4 to 9 in algorithm 4.

For the purpose of optimizing the conditional net using gradient descent, we need to find the gradients for the objective function of the conditional net defined in equation (3.20). The computation of the unbiased approximate gradients for the individual terms in the objective function is shown in statement 10. We finally optimize the conditional net by the employing gradient descent step and updating the model parameters by descending to the approximated gradients as shown in statement 11 of algorithm 4.

Algorithm 4 *Optimization over Discrete Conditional Net* $\boldsymbol{\theta}_c$

Input: Training input $(\mathbf{x}, \mathbf{a}) \in \mathcal{W}$, prediction net output $\mathbf{y}_p$

Output: $\hat{\mathbf{y}}_c^1, \ldots, \hat{\mathbf{y}}_c^K$, sample K predictions from the model

1: **for** $k = 1$ to K **do** ▷ Generate K candidate outputs

2: Sample noise vector $\mathbf{z}_k$, Generate output $\hat{\mathbf{y}}_c^k$:

$$\hat{\mathbf{y}}_c^k = \arg\max_{y \in \mathcal{Y}} \mathcal{S}^k(\mathbf{y}_c)$$

3: Find loss-augmented prediction $\hat{\mathbf{y}}_a^k$ with respect to $\mathbf{y}_p$:

$$\hat{\mathbf{y}}_a^k = \arg\max_{y \in \mathcal{Y}} \mathcal{S}^k(\mathbf{y}_c) \pm \epsilon\Delta(\mathbf{y}_p, \hat{\mathbf{y}}_c^k)$$

4: **end for**

5: **for** $k = 1$ to K **do** ▷ Compute loss-augmented predictions

6: **for** $k' = 1$ to K, $k' \neq k$ **do**

7: Find loss-augmented prediction $\hat{\mathbf{y}}_b^{k,k'}$:

$$\hat{\mathbf{y}}_b^{k,k'} = \arg\max_{y \in \mathcal{Y}} \mathcal{S}^k(\mathbf{y}_c) \pm \epsilon\Delta(\hat{\mathbf{y}}_c^k, \hat{\mathbf{y}}_c^{k'})$$

8: **end for**

9: **end for**

10: Compute unbiased approximate gradients:

$$DIV_\Delta^\epsilon(\mathrm{Pr}_p, \mathrm{Pr}_c) = \frac{1}{K}\sum_{k=1}^{K}\left[\nabla_{\boldsymbol{\theta}_c}\mathcal{S}^k(\hat{\mathbf{y}}_a) - \nabla_{\boldsymbol{\theta}_c}\mathcal{S}^k(\hat{\mathbf{y}}_c)\right] \tag{3.25}$$

$$DIV_\Delta^\epsilon(\mathrm{Pr}_c, \mathrm{Pr}_c) = \frac{2}{K(K-1)}\sum_{\substack{k,k'=1 \\ k' \neq k}}^{K}\left[\nabla_{\boldsymbol{\theta}_c}\mathcal{S}^k(\hat{\mathbf{y}}_b) - \nabla_{\boldsymbol{\theta}_c}\mathcal{S}^{k'}(\hat{\mathbf{y}}_c)\right] \tag{3.26}$$

11: Update model parameters using gradient descent:

$$\boldsymbol{\theta}_c^{t+1} = \boldsymbol{\theta}_c^t - \eta\nabla_{\boldsymbol{\theta}_c} DISC_\Delta(\mathrm{Pr}_p(\boldsymbol{\theta}_p), \mathrm{Pr}_c(\boldsymbol{\theta}_c))$$

Chapter 4

Weakly Supervised Human Pose Estimation

4.1 Introduction

Current approaches to learning human pose estimation from still images rely on collecting fully annotated datasets, where each training sample includes an image of a person along with their ground-truth joint locations. However, obtaining such detailed annotations is both challenging and costly, making this approach impractical at scale. To overcome the limitations of fully supervised learning, we propose leveraging a diverse dataset. In this setup, a subset of the images is labeled with expensive pose annotations, while the remaining images are annotated with inexpensive action labels.

This type of dataset offers two key advantages. First, it can be collected at a significantly lower cost. For instance, performing a simple keyword search, such as 'running,' on an image search engine yields hundreds of thousands of freely available images that can be easily curated with the help of human annotators. Second, action labels provide valuable contextual information about poses. For example, the action running' excludes poses where a person is lying down or upside down, effectively narrowing the range of plausible poses (Figure 4.1).

We assume that the distribution of images labeled with different types of annotations is the same (a necessary assumption for learning) and that the annotations themselves are noise-free. Under these assumptions, we argue that action information can facilitate learning pose estimation. Note that earlier works have exploited the relationship between action and pose for action recognition. However, our problem is significantly more challenging due to the high uncertainty in the pose associated with a given action (Figure 4.1). To model this uncertainty, we propose using a probabilistic learning formulation. A typical probabilistic formulation would learn a joint distribution of the pose and the action given an image. To make a prediction on a test sample, where action information is not known, it would sum over all possible actions to marginalize their effects. In other words, it would use one set of parameters for two distinct tasks: (i) modeling the uncertainty in the pose for each action; and (ii) predicting the pose given an image.

As our goal is to make an accurate pose prediction, we argue that such an approach would waste the modeling capability of a distribution in representing pose uncertainty in the presence of action information. In other words, the parameters of the distribution will be tuned to perform well in the presence

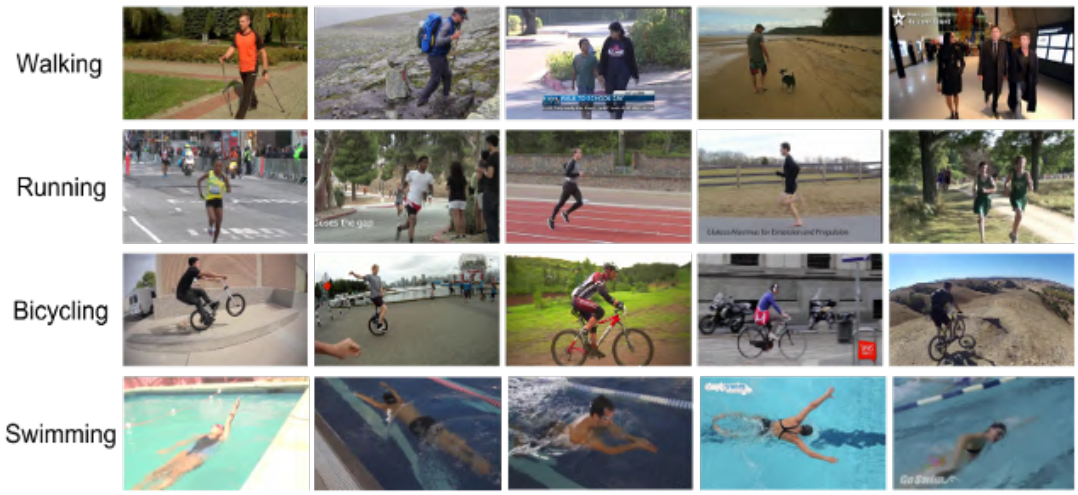


Figure 4.1 *The figure displays images of action classes from the MPII Human Pose dataset. It illustrates that human poses within a single action class can vary significantly. Additionally, the figure highlights the high intra-class variance, as seen in the swimming action label, where swimmers are depicted in a variety of swimming poses.*

of action information, which is not available during testing. Instead, we use two distinct distributions for the two tasks: (i) a *conditional distribution* of the pose given the image and the action; and (ii) a *prediction distribution* of the pose given the image.

We jointly estimate the parameters of the two distributions by minimizing their dissimilarity coefficient [82], which uses a task-specific loss function to measure the distance between the samples from the two distributions. By transferring the information from the conditional distribution to the prediction distribution, we learn to estimate the pose of a human using a diverse dataset. Figure 4.2 illustrates the importance of using a probabilistic model. Specifically, the figure depicts the average entropy of each joint predicted by our model on test images. We observe that the most articulate joints, such as wrists and ankles, have the highest entropy, which a non-probabilistic network does not explicitly model.

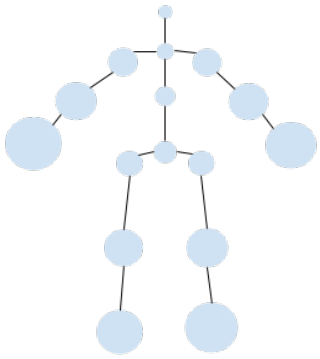


Figure 4.2 *The average entropy of joints in test images is visualized over a stick figure. The radius of a circle around a joint is proportional to the joint's entropy.*

Although our approach can be applied to any parametric family of distributions, in this work, we focus on state-of-the-art deep probabilistic networks. Specifically, we model both the conditional and prediction distributions using a DISCO Net [83], which allows us to efficiently sample from the two distributions. As we will show, efficient sampling is sufficient to make both training and testing computationally feasible.

We demonstrate the efficacy of our approach using the publicly available MPII Human Pose [18] and JHMDB [85] data sets. We discard the pose information of a portion of the training samples but retain the action information for all the samples to generate a diverse dataset. We provide a thorough comparison of our probabilistic approach with two commonly used baselines. The first is a fully supervised approach, which discards the weakly supervised samples labeled using only the action information. The second is a pointwise model that uses self-paced learning [86], first learning from easy samples and then gradually increasing the difficulty of the training samples. We show that, by explicitly modeling the uncertainty for the pose of diverse supervised samples, our approach significantly outperforms both baselines under various experimental settings.

4.2 Related Work

With the introduction of "DeepPose" by Toshev *et al.* [87], research on human pose estimation shifted from classic approaches based on pictorial structures [88–96] to deep networks. Subsequent methods include [97], which simultaneously captures features at a variety of scales using heatmaps, and [98], which employs a hierarchical model to capture the relationships between joints. A popular approach by Newell *et al.* [4] uses a conv-deconv architecture and residual models to efficiently generate heatmaps without requiring hierarchical processing. This approach has been further extended by incorporating visual attention [99] and feature pyramids [100]. However, these methods rely on the network's capacity to capture highly articulated human poses and handle occlusions, without explicitly modeling the uncertainty of the pose.

Modeling the uncertainty for the human pose becomes crucial in a diverse data setting, where some training samples only provide action information. While pose has often been used to predict action [101–104], the use of action for pose estimation has primarily been studied for 3D human pose [105] or videos with available temporal information [106–109]. To the best of our knowledge, our work is the first to exploit action information for 2D pose estimation in still images.

While pose estimation using action information has received limited attention, the general problem of diverse data learning has a rich history in machine learning and computer vision. Most traditional approaches relied on simple parametric structured models such as conditional random fields or structured support vector machines [50–52, 110–112]. However, as traditional structured prediction models have been replaced by deep learning, these formulations must be adapted for neural network parameter estimation. Indeed, our work can be viewed as a natural generalization of [51] for deep probabilistic models that admit efficient sampling mechanisms.

The deep learning community also realizes the importance of using diverse datasets to scale up data-hungry neural network-based approaches. This has led to recent research in deep multiple instance learning [68, 113, 114], as well as expectation-maximization-based methods [69, 70]. However, most deep diverse data learning approaches have been designed for specific tasks, such as semantic segmentation [71, 115]. It is not clear how these methods can be adapted to learn human poses from action labels. In contrast, our general formulation (presented in the next section) can be adapted to any task by defining a task-specific loss function. While we are primarily interested in pose estimation, our formulation may be of interest to the broader audience working on diverse data deep learning.

4.3 Problem Formulation

Our approach uses the recently proposed deep probabilistic network, DISCO Nets [83]. The DISCO Nets framework allows us to adapt a pointwise network (that is, a network that provides a single pointwise prediction) to a probabilistic one by introducing a noise filter in the pointwise network (Section 3.1.3).

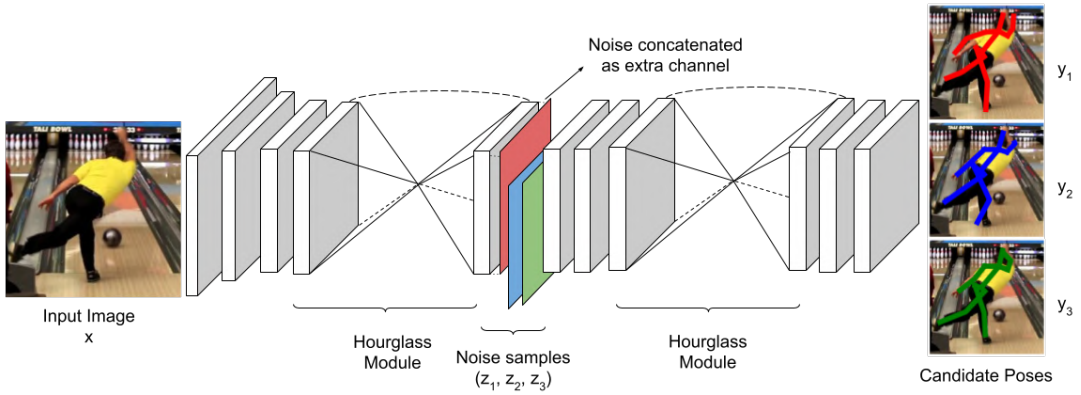


Figure 4.3 *For a single input image* $\mathbf{x}$ *and three different noise samples* $\{\mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3\}$ *(represented as red, green, blue matrix respectively),* DISCO *Nets produces three different candidate poses* $\{\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3\}$*. Here each block is a residual layer and two hourglass shaped blocks represent the hourglass module proposed by Newell* et al. *[4]. Best viewed in color.*

As a concrete example, consider the modified stacked hourglass network in Figure 4.3, which can be used for human pose estimation. The colored filters in the middle of the network represent the noise that is sampled from a uniform distribution. Each value of the noise filter results in a different pose estimate for the same image, thereby enabling us to generate samples from the underlying distribution encoded by the network parameters. Note that obtaining a single sample is as efficient as a forward pass through the network. By placing the filters sufficiently far away from the output layer of the network, we can learn a highly non-linear mapping from the uniform distribution (used to generate the noise filter) to the output distribution (used to generate the pose estimates).

In [83], the parameters of the DISCO Nets were learned by minimizing the dissimilarity of the network distribution and the true distribution (as specified by fully supervised training samples). However, we show how the DISCO Nets framework can be extended to enable diverse data learning.

4.3.1 Model

Due to the uncertainty inherent in the task of pose estimation (occlusion of joints, articulation of human body) as well as the uncertainty introduced by the use of a diverse data set during training, we advocate the use of a probabilistic formulation. To this end, we define two distributions. The first is the *prediction distribution* that models the probability of a pose $\mathbf{y}$ given an image $\mathbf{x}$. As the name suggests, this distribution is used to make a prediction during test time. In this work, we model the prediction distribution $\Pr_p(\mathbf{y}|\mathbf{x};\boldsymbol{\theta}_p)$ as a DISCO Nets, where $\boldsymbol{\theta}_p$ are the parameters of the network.

In addition to the prediction distribution, we also model a *conditional distribution* of the pose given the image and the action label. As the conditional distribution contains additional information, it can be expected to provide better pose estimates. We will use this property during training to learn an accurate prediction distribution using the conditional distribution. As will be seen shortly, the conditional distribution will not be used during testing. Similar to the prediction distribution, the conditional distribution $\Pr_c(\mathbf{y}|\mathbf{x},\mathbf{a};\boldsymbol{\theta}_c)$ is modeled using a DISCO Nets, with parameters $\boldsymbol{\theta}_c$. Note that, while we do not have access to the partition function of the two aforementioned distributions, the use of a DISCO Net ensures that we can efficiently sample from them. This property will be exploited to make both the testing and the training computationally feasible.

4.3.2 Prediction

We assume a task-specific loss function $\Delta(\cdot,\cdot)$ that measures the difference between two putative poses of an image. Given an image $\mathbf{x}$ containing a human, we would like to estimate the pose $\mathbf{y}$ of the human such that it minimizes the risk of prediction (as measured by the loss function Δ). Since the ground-truth pose is unknown, we use the principle of maximum expected utility (MEU) [116]. The MEU criterion minimizes the expected loss using a set of samples $\mathcal{Y} = \{\mathbf{y}^k, k = 1,\ldots,K\}$ obtained from the distribution $\Pr_p(\mathbf{y}|\mathbf{x};\boldsymbol{\theta}_\mathbf{p})$.

Formally, given an image $\mathbf{x}$, we provide a pointwise prediction of the pose in two steps. First, we estimate K pose samples using K different noise filters, each of which is sampled from a uniform distribution. Second, we use the MEU criterion to obtain the prediction as,

$$\mathbf{y}^*_\Delta(\mathbf{x};\boldsymbol{\theta}_p) = \underset{k\in[1,K]}{\arg\min} \sum_{k'=1}^{K} \Delta(\mathbf{y}^k,\mathbf{y}^{k'}). \tag{4.1}$$

As can be seen, the above criterion can be easily applied for any loss function. For human pose estimation, we adopt the commonly used loss function that measures the mean squared error between the belief maps of two poses over all the joints [4, 97, 98]. The belief map $b_\mathbf{y}(j)$ of a joint j is created by defining

a 2D Gaussian whose mean is at the estimated location of the joint, and whose standard deviation is a fixed constant.

4.3.3 Diverse Data Set

In order to learn the parameters $\boldsymbol{\theta}_p$ of the prediction distribution, we require a training data set. Current methods rely on a fully supervised setting, where each training sample is labeled with its ground-truth pose. In order to avoid the cost of such detailed annotations, we advocate the collection of a diverse data set, with a small number of fully supervised samples and a large number of weakly supervised samples. The presence of fully supervised samples helps disambiguate the problem of pose estimation from the problem of action classification.

Formally, we denote our training data set as $\mathcal{D} = \{\mathcal{W}, \mathcal{S}\}$, where $\mathcal{W} = \{(\mathbf{x}_i, \mathbf{a}_i), i = 1 \dots N\}$ is the weakly annotated data set, and $\mathcal{S} = \{(\mathbf{x}_j, \mathbf{a}_j, \mathbf{h}_j), j = 1 \dots M\}$ is the strongly annotated data set and $M < N$. Here $\mathbf{x}_i$ refers to the i-th training image and $\mathbf{a}_i$ denotes its action. We denote the underlying pose of the image $\mathbf{x}_i$ as the output variable $\mathbf{y}_i$. Note that we do not assume a single underlying pose. Instead, we model the distribution over all putative poses given the image and action.

4.3.4 Learning Objective

Given the diverse data set $\mathcal{D}$, our goal is to learn the parameters $\boldsymbol{\theta}_p$ such that it provides an accurate pose estimate $\mathbf{y}^*_\Delta(\mathbf{x}; \boldsymbol{\theta}_p)$ (specified in equation (4.1)) for a test image $\mathbf{x}$. A typical learning objective for this purpose would estimate the joint distribution $\Pr_p(\mathbf{y}, \mathbf{a}|\mathbf{x}; \boldsymbol{\theta}_p)$ using expectation-maximization or its variants [117]. Given an image $\mathbf{x}$, the pose would then be obtained by marginalizing over all actions $\mathbf{a}$. However, as discussed in Section 3.2, this approach needlessly places the burden of accurately representing the uncertainty of the pose and the action of an image on a single distribution. Since the action information would not be provided during testing, such an approach may fail to fully utilize the modeling capacity of the distribution parameters to obtain the best pose.

Inspired by the work of Kumar *et al.* [51], we design a joint learning objective that minimizes the dissimilarity coefficient between the prediction distribution and the conditional distribution (defined in Section 3.2.3). Formally, our learning objective is defined in Equation (3.6), which is shown below:

$$\boldsymbol{\theta}^*_p, \boldsymbol{\theta}^*_c = \underset{\boldsymbol{\theta}_p, \boldsymbol{\theta}_c}{\arg\min} \sum_{i=1}^{N} DISC_\Delta(\Pr_p(\mathbf{y}|\mathbf{x}; \boldsymbol{\theta}_p), Pr_c(\mathbf{y}|\mathbf{a}, \mathbf{x}; \boldsymbol{\theta}_c)). \tag{4.2}$$

To ensure computational efficiency, we employ the stochastic unbiased estimation method for self-diversities and cross-diversity as described in Section 3.2.3. Following equations (3.11, 3.18, 3.17), the

objective (equation (3.6)) can be computed as,

$$\frac{1}{K^2}\sum_{k=1}^{K}\sum_{k'=1}^{K}\Delta(\mathbf{y}_p^{k'},\mathbf{y}_c^{k}) - \frac{1}{K(K-1)}\Big(\gamma\sum_{\substack{k,k'=1\\k'\neq k}}^{K}\Delta(\mathbf{y}_c^{k},\mathbf{y}_c^{k'}) + (1-\gamma)\sum_{\substack{k,k'=1\\k'\neq k}}^{K}\Delta(\mathbf{y}_p^{k},\mathbf{y}_p^{k'})\Big). \tag{4.3}$$

Here, Δ is tuned for human pose estimation and computes the mean squared error between the belief maps of two poses over all joints.

4.3.5 Optimization

As detailed in Section 3.2.4, both the prediction and conditional distributions are modeled using DISCO Nets, making them well-suited for optimization via stochastic gradient descent. In order to make the most use of the diverse nature of the data set, as well as the learning objective, we estimate the parameters of the two networks in three stages. First, we use supervised training for the two networks using the small amount of the ground truth pose data. Second, we perform iterative training of the two networks, that is, we update one network while keeping the other fixed. Third, we jointly optimization of both the networks together. At each stage, we use stochastic gradient descent in a similar manner to [83]. Joint training of the two network is expensive in terms of memory and time. However, by first training the two networks using strong supervision and then using iterative optimization strategy, we significantly reduce the number of iterations required in the third stage of the optimization. The details of each step is discussed in Section 3.2.4.

4.3.5.1 Visualization of the Learning Process

We visualize the predictions of the two networks during the iterative training process to understand how information is transferred from the conditional network to the prediction network. Figure 4.4 provides an overview of this process using a representative example, while Figures 4.5, 4.6, and 4.7 delve into specific cases categorized as easy, moderate, and difficult, respectively. Each visualization shows 100 superimposed pose estimates from both the prediction and conditional networks. The opacity and spread of the lines represent the agreement among the samples: thin and opaque lines indicate low uncertainty, while spread-out and less opaque lines indicate high uncertainty.

To represent uncertainty levels, bounding boxes are drawn around the images:

- **Green bounding box:** Indicates low uncertainty, where the expected loss is less than 3.
- **Blue bounding box:** Indicates high uncertainty, where the expected loss is greater than 3.

4.3.5.1.1 Representative Example Figure 4.4 illustrates the iterative learning process using a common action, such as riding a bike. Initially, the prediction network (Pr_p) and the conditional network (Pr_c) have different levels of uncertainty: the conditional network is more confident, while the prediction network is less so. Over the iterations, the predictions from both networks align more closely,

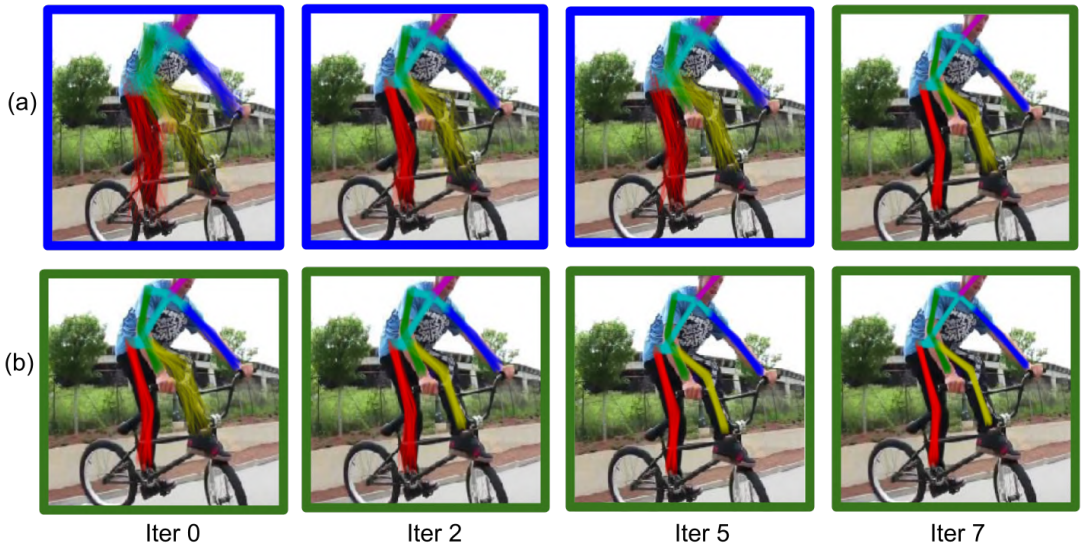


Figure 4.4 *Example of superimposed pose predictions by* DISCO *Nets illustrating the uncertainty in the pose across training iterations. The blue box around the images represent a high diversity coefficient value, and the green box around them represents low diversity coefficient value. Row (a) represents outputs from the prediction network and row (b) represents outputs from the conditional network. The first column shows the initial prediction of the networks; columns* 2 *through* 4 *shows prediction of the networks at second, fifth and final iteration respectively. The images show a common action of riding a bike where the conditional network performs well from the beginning of the optimization procedure and transfers its knowledge to the prediction network. Best viewed in color.*

reflecting the successful transfer of information. This example serves as a general depiction of how the two networks evolve during training, setting the stage for the specific cases discussed below.

4.3.5.1.2 Easy Cases Figure 4.5 represents easy cases, where both the prediction network ($\Pr_{\mathbf{w}}$) and the conditional network ($\Pr_{\boldsymbol{\theta}}$) initially have low uncertainty for the predicted pose. These examples often involve clear, unoccluded images of individuals in standard poses for common actions, such as walking or standing. The fully annotated training dataset is typically sufficient for the prediction network to achieve high confidence in these cases, requiring little benefit from weakly supervised training. However, even in these cases, minor improvements in pose estimation can be observed over the iterations of the optimization algorithm.

4.3.5.1.3 Moderate Cases Figure 4.6 shows moderate cases, such as individuals performing actions like exercising, riding a bike, or running. These examples may feature occluded joints or variations of standard poses. Initially, the prediction network ($\Pr_p$) has high uncertainty, whereas the conditional network ($\Pr_c$) exhibits low uncertainty and high confidence in the predicted pose. Over the iterations, the prediction network benefits significantly as information from the conditional network is successfully

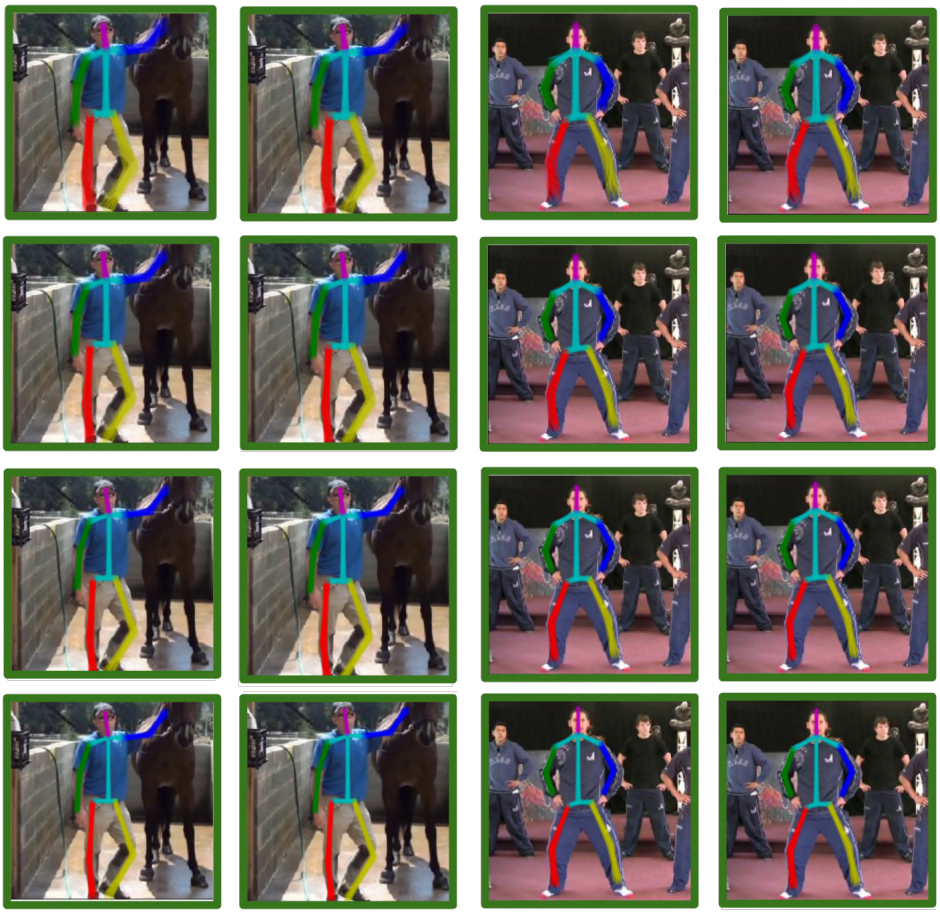


Figure 4.5 *Example of superimposed pose predictions by* DISCO *Nets illustrating the uncertainty in the pose across training iterations for an easy case. The blue box around the images represent a high diversity coefficient value, and the green box around them represents low diversity coefficient value. Columns* 1 *and* 3 *are outputs of the prediction network and columns* 2 *and* 4 *are outputs of conditional network. Row* 1 *represents initial prediction of networks; rows 2 and 3 represents prediction of networks in second and fifth iteration respectively and last row represents prediction of networks when they have converged. The images in the first and second column show an easy example of a person standing straight with his one hand held out and the third and fourth columns show a person standing in relaxed upright pose. where both the conditional network and the prediction network performs well from the beginning of the optimization procedure. For each example, the first column shows estimated pose from prediction network and the second column shows estimated pose from conditional network. Best viewed in color.*

transferred to it. This results in a notable improvement in the accuracy and confidence of the predictions for these cases. The majority of the dataset falls into this category.

4.3.5.1.4 Difficult Cases The final example, shown in Figure 4.7, illustrates challenging cases, such as rare actions like underwater swimming or a person kicking a ball in mid-air. Due to the rarity of

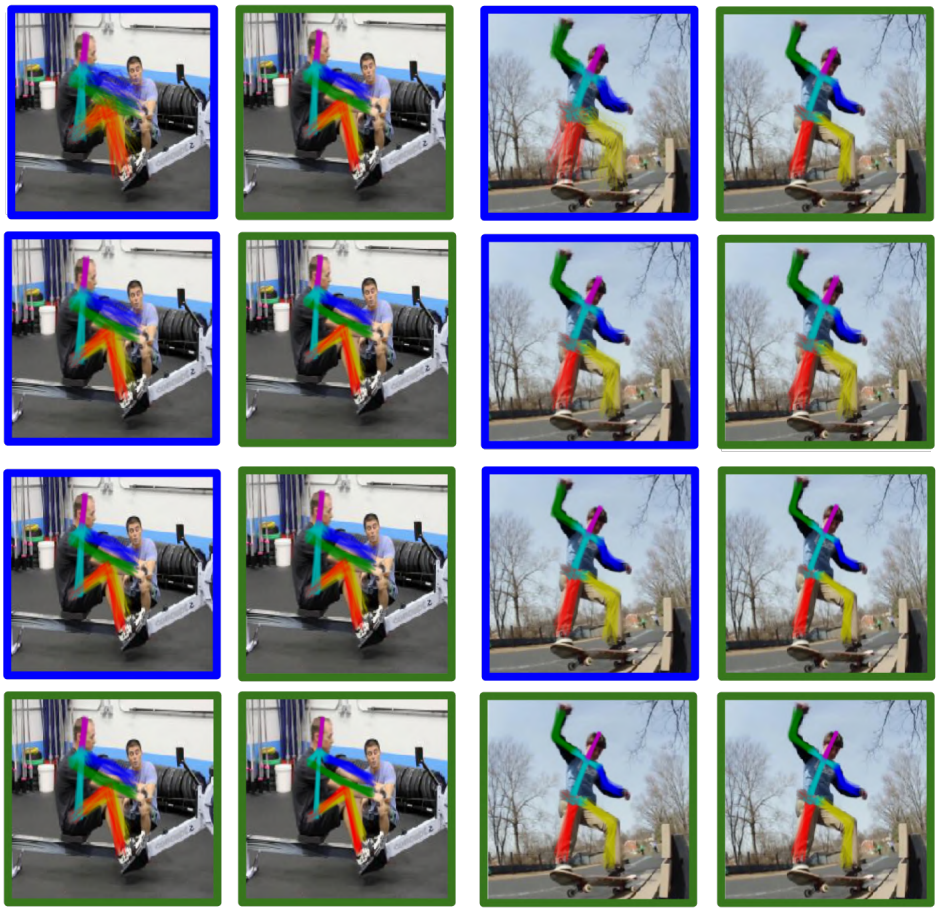


Figure 4.6 *Example of superimposed pose predictions by* DISCO *Nets illustrating the uncertainty in the pose across training iterations for examples with moderate difficulty. The blue box around the images represent a high diversity coefficient value, and the green box around them represents low diversity coefficient value. Columns* 1 *and* 3 *are outputs of the prediction network and columns* 2 *and* 4 *are outputs of conditional network. Row* 1 *represents initial prediction of networks; rows 2 and 3 represents prediction of networks in second and fifth iteration respectively and last row represents prediction of networks when they have converged. The images in the first and second column show a common action of a person exercising and the third and fourth column shows a person riding a skate board. In these cases, the conditional network performs well from the beginning of the optimization procedure. At convergence, both the prediction network provides accurate pose estimates for such moderately difficult images by transferring information from conditional network. For each example, the first column shows estimated pose from prediction network and the second column shows estimated pose from conditional network. Best viewed in color.*

such poses in the fully annotated training set, both the prediction and conditional networks ($\Pr_p$ and $\Pr_c$) exhibit high uncertainty in their initial predictions. However, through iterative optimization and by leveraging the information gained from simpler examples in the weakly supervised dataset, the accuracy of the predicted poses for these challenging cases improves significantly over time.

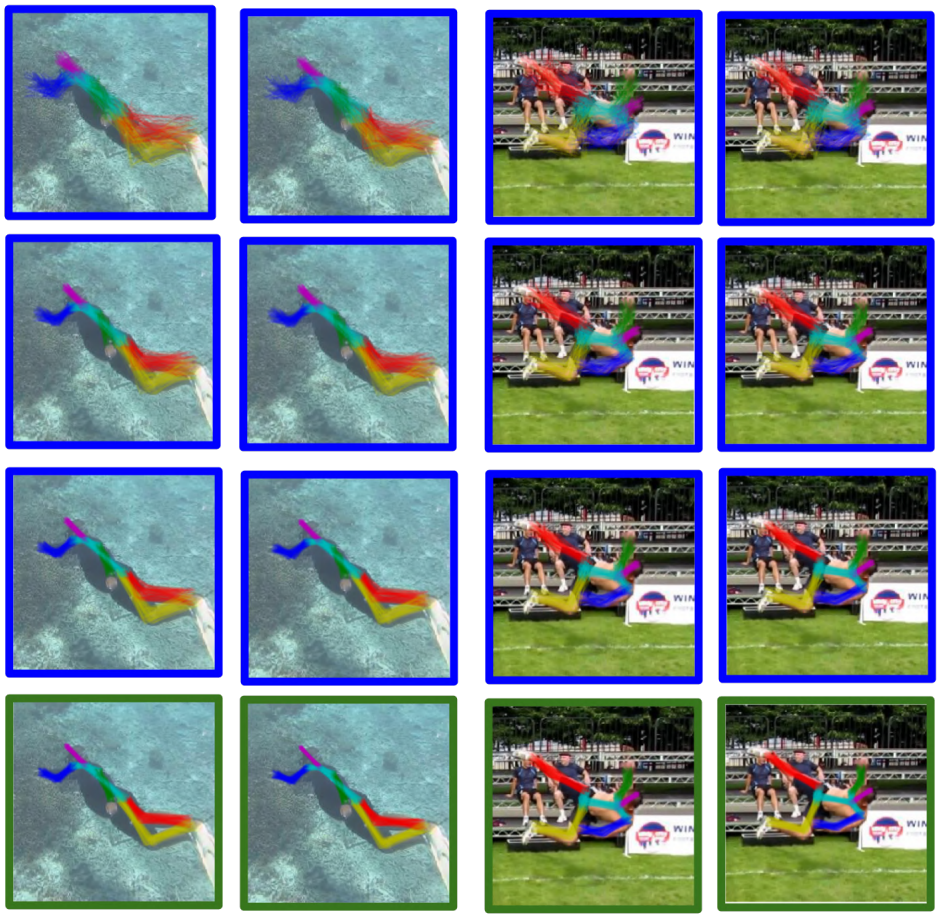


Figure 4.7 *Example of superimposed pose predictions by* DISCO *Nets illustrating the uncertainty in the pose across training iterations for difficult examples. The blue box around the images represent a high diversity coefficient value, and the green box around them represents low diversity coefficient value. Columns* 1 *and* 3 *are outputs of the prediction network and columns* 2 *and* 4 *are outputs of conditional network. Row* 1 *represents initial prediction of networks; rows 2 and 3 represents prediction of networks in second and fifth iteration respectively and last row represents prediction of networks when they have converged. The images in the first and second column show a rare action of person swimming underwater, and the third and fourth columns show a person in an unusual pose, where he is kicking the ball in air. Such rarity in pose leads to high uncertainty in both the networks initially. At convergence, both the networks provided accurate pose estimates for the difficult image by learning from the easier images. For each example, the first column shows estimated pose from prediction network and the second column shows estimated pose from conditional network. Best viewed in color.*

4.3.5.1.5 Summary of Observations Throughout the iterative learning process, we observe the following trends:

1. **Representative cases:** The representative example shows the general behavior of the two networks, where the prediction network starts with high uncertainty but aligns more closely with the conditional network as training progresses.

2. **Easy cases:** The prediction network starts with high confidence and only marginally benefits from weakly supervised training.

3. **Moderate cases:** The prediction network starts with high uncertainty but improves significantly as information from the conditional network is transferred to it.

4. **Difficult cases:** Both networks initially exhibit high uncertainty, but the weakly supervised dataset enables gradual improvement through information sharing from simpler examples.

These visualizations comprehensively illustrate the learning process for different levels of difficulty in the dataset.

4.4 Experiments

4.4.1 Data set

We use the MPII Human pose data set [18], which consists of 17.4k images with publicly available action and ground-truth pose annotations. We split the images into $\{70, 15, 15\}\%$ training, validation and test sets, which corresponds to $12,156$ images in the training set and 2605 images each in the testing and the validation set. In order to obtain a diverse data set, we discard the pose information for a random subset of training examples, while retaining action labels for all samples. This results in (i) a fully annotated training set, which contains both the ground truth pose annotations and the action labels; and (ii) a weakly annotated training set, which only contains action labels.

To obtain the tasks of varying levels of difficulty, we choose three different data splits, $\{25-75, 50-50, 75-25\}\%$, where we randomly discard 75%, 50%, and 25% of the pose annotations from the training images respectively. We note here that for each split, we augment our training set by rotating the images with an angle $(+/-30^{\circ})$ and by horizontal flipping the original image.

We additionally use the JHMDB data set [85]. The JHMDB data set, which consists of 33183 frames from 21 action class, have 13 annotated joint locations. We split the frames from each action class into $\{70, 15, 15\}\%$ training, validation and test sets, which corresponds to 22883 frames in the training set, and 4150 frames in the validation and the test set. We present our results on $50 - 50$ split. To create a diverse data set with $50 - 50$ split, we randomly drop pose annotations from 50% from the frames of the training set.

4.4.2 Implementation and Experimental Setup

To implement our probabilistic DISCO network, shown in Figure 4.3, we adopt the stacked hourglass network [4], a widely used architecture for human pose estimation. The stacked hourglass network

consists of 8 hourglass modules. For the prediction network Pr_p, a noise filter of size 64×64 is added to the output of the penultimate hourglass module, which itself consists of 256 64×64 filters. The 257 channels (including the noise channel) are convolved with a 1×1 filter to bring the number of channels back to 256, followed by a final hourglass module, as shown in Figure 4.3. This architecture ensures that all parameters remain differentiable and can be trained via backpropagation.

The conditional network Pr_c is modeled similarly to the prediction network, except that it includes $\mathbf{a}$ different output branches, one for each possible action class. These branches are stacked on top of the penultimate hourglass module. Each branch has its own noise filter followed by a final hourglass module. During forward and backward propagation, only the branch corresponding to the current action class is active, while outputs from other branches are masked. This setup ensures efficient processing without unnecessary computations.

When drawing K samples from the modified stacked hourglass architecture for a single input image, we reuse the output of the penultimate layer of the 8-stacked hourglass network. Only the final hourglass module is recomputed K times to generate K samples, significantly reducing runtime complexity. For $K = 100$, a single forward pass of the probabilistic network takes 114 ms compared to 68 ms for the vanilla stacked hourglass network on an NVIDIA GTX 1080Ti GPU.

4.4.2.1 Network Initialization and Training

The prediction network Pr_p is initialized by training on a small, fully annotated training dataset. The conditional network Pr_c is initialized by fine-tuning the weights of the prediction network using action-specific samples from the fully annotated training set. After initialization, we optimize the two networks first using an iterative optimization procedure, followed by joint optimization as described in Section 3.2.3.

We employ data augmentation to expand the training dataset by $4\times$, including weakly annotated and fully annotated data. For the fully supervised (FS) network, additional random crops are performed to maintain an equal number of training samples across all networks. The probabilistic networks Pr_p and Pr_c are trained with $K = 100$ samples. However, previous studies [83] indicate that results remain robust even with $K = 2$ for different tasks.

4.4.2.2 Optimization and Early Stopping

All networks are trained for 100 epochs, with early stopping based on validation accuracy to prevent overfitting. The training is performed using stochastic gradient descent (SGD) with a learning rate $\eta = 0.025$ and momentum $m = 0.9$. For weight decay regularization, the parameter C is cross-validated in the range $[0.1, 0.01, 0.001, 0.0001]$, with optimal values of 0.001 for FS, 0.0001 for PW, and 0.01 for the probabilistic networks. We save the network parameters corresponding to the best validation accuracy and report results on the held-out test set.

4.4.3 Methods

We compare our proposed probabilistic method, learned with diverse data, with two baselines: (i) a fully supervised human pose estimation network, the stacked hourglass network [4], which we refer to as FS Net; and (ii) a non-probabilistic pointwise network trained with diverse data, which uses the same architecture as shown in figure 4.3 but provides a single prediction. We refer this pointwise network as PW Net. The first baseline helps us to compare the performance of a fully supervised network with a network trained on the diverse collection of data, and the second baseline demonstrates the benefit of our probabilistic network when compared to a non probabilistic pointwise network.

We train FS net on the fully annotated data set using stochastic gradient descent, as discussed in [4]. The PW net is trained using diverse data, making use of the action annotations.

4.4.3.1 Baseline Comparisons and Regularization

To compare with baseline methods, we train non-probabilistic pointwise networks (PW) that discard the last two self-diversity terms from the probabilistic objective function and compute predictions using the principle of maximum expected utility (MEU). The prediction and conditional pointwise networks (PW_p and PW_c) are initialized similarly to their probabilistic counterparts and fine-tuned using action-specific samples. During training, self-paced learning is employed for PW networks, where backpropagation is applied only if the loss is below a threshold t. This ensures the network learns from confident predictions while avoiding erroneous or uncertain samples.

In contrast, the probabilistic network Pr_p does not require such a threshold. The diversity coefficient in the objective function inherently ensures that the network learns only from confident predictions, reducing the need for additional parameters compared to the baseline.

4.4.4 Results

4.4.4.1 Results on MPII Human Pose Data Set

We evaluate the three trained networks, FS, PW and Pr_p, by computing their accuracy on the held out test set. We use the normalized "Probability of Correct Keypoint" (PCKh) metric [95] to report our results. Table 4.1 shows the performance of the three networks when trained on varying splits of the training set.

Here, we observe that, for all the data splits, our proposed probabilistic network $\text{Prob}_{\mathbf{w}}$ outperforms the other baseline networks FS and PW. This superior performance is seen consistently across predictions of all joints as well as on the overall pose prediction.

Performance of the three networks, FS, PW and Pr_p, increases with the increase in level of supervision. In the more challenging $25 - 75$ split, there are far fewer fully supervised examples present for each action category which results in PW and Pr_p to learn a poor initial estimate of action specific pose from diverse data. This leads to overall poor performance when compared to $50 - 50$ or $75 - 25$ split case, where we have more supervised data.

Table 4.1 *Results on MPII Human Pose (*PCKh@0.5*), where FS is trained on varying percentages of fully annotated data and PW and* Pr_p *are trained on varying splits of fully annotated and weakly annotated training data. Here FS and PW are the fully supervised and the pointwise networks respectively, and* Pr_p *(iterative) and* Pr_p *(joint) is our proposed probabilistic network trained with iterative optimization and joint optimization respectively. The supervised subset is the fully supervised stacked hourglass net [4] trained with all the available labels and defines the upper bound on the total accuracy that can be achieved through this architecture.*

Method	Split	Head	Sho.	Elb.	Wri.	Hip	Knee	Ank.	Total
Supervised Subset	100%	98.16	96.22	91.23	87.08	90.11	87.39	83.55	90.92
FS	25%	59.17	46.98	30.00	21.33	36.32	20.05	23.93	37.54
	50%	90.18	80.60	64.29	52.43	67.44	55.41	51.30	67.88
	75%	94.61	90.56	81.28	74.15	81.86	73.20	67.19	80.88
PW	25-75	73.77	55.69	37.21	25.32	43.24	28.01	30.82	45.16
	50-50	92.97	83.56	71.08	59.18	72.56	60.49	57.27	73.11
	75-25	95.46	93.50	86.47	81.05	85.58	80.98	76.81	85.89
Pr_p (iterative)	25-75	78.21	60.98	42.01	28.75	42.37	29.07	33.54	48.12
	50-50	93.42	86.91	75.03	66.56	77.22	67.38	60.96	76.43
	75-25	96.28	94.53	88.36	83.31	87.54	82.45	79.48	88.16
Pr_p (joint)	25-75	79.54	62.87	43.38	29.38	43.38	30.91	34.86	**49.41**
	50-50	94.07	88.32	75.93	67.53	78.20	67.80	61.49	**78.01**
	75-25	97.45	95.87	90.21	86.09	89.42	86.26	82.92	**90.21**

Moreover, both the methods trained using diverse data, PW and Pr_p, show a significant gain in accuracies when compared to the fully supervised network, FS. This empirically shows us that the action information present in the weakly annotated set is helpful for predicting pose.

As our proposed probabilistic network Pr_p performs better than the pointwise network PW, we see the significance of modeling uncertainty over pose. Though the proposed probabilistic network only marginally improves the prediction for joints with low uncertainty, like the head, shoulder and hips, the difference in the accuracies of the two networks is due to better performance of the probabilistic network Pr_p on difficult joints like wrists, elbows, knees and ankles. We see that the Pr_p network provides a significant improvement of up to 5% improvement in accuracies over the PW Net on joints with high uncertainty (wrists, elbows, ankles and knees).

Joint training of the two set of networks improves our prediction by around 1.5%. We also note that, while the supervised subset, which is the fully supervised stacked hourglass network [4] trained using all available labels in the training set, achieves 90.9% [4], our probabilistic network provides comparable results when trained only on 75% pose annotations and 25% action annotations. Note that the supervised subset defines the upper bound on accuracy that can be achieved through this architecture.

We argue that the relative position of joints like head, shoulder and hip remains largely in similar spatial location with respect to each other across various actions and therefore have low entropy, whereas, joints like wrists, elbows, knees and ankles not only show huge variations in their relative spatial location across various action categories but also within same action category, resulting in large entropy. Therefore, even though pointwise network PW does a good job of estimating pose locations for joints

with low uncertainty, it fails to capture the high inter-class and intra-class variability of joints with high uncertainty. On the other hand, $\mathrm{Pr}_{\mathbf{w}}$ explicitly models uncertainty over joint locations as can be seen in figure 4.2.

The detailed PcKh graphs on MPII data set by training an 8-stack hourglass network on various setting described in the paper are presented in figure 4.8.

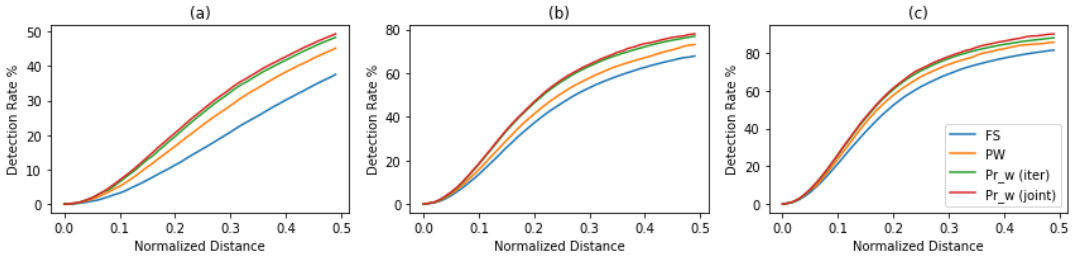


Figure 4.8 *Total PcKh comparison on MPII when trained on (a)* $25-75$ *split, (b)* $50-50$ *split; and (c)* $75-25$ *split.*

In the figure, we can see that we consistently outperform the baseline FS and PW networks across all normalized distances. The networks trained on diverse data set (the PW and the $\mathrm{Pr}_{\mathbf{w}}$ network) performs significantly better on lower normalized scores than the FS net which does not utilize the action annotations when there are only a few strong pose annotations available. This shows the utility of using action annotations when pose annotations are missing. The importance of using the probabilistic framework can be seen for lower normalized distance for all three splits, where the $\mathrm{Pr}_{\mathbf{w}}$ network effectively captures the uncertainty present in the data set. We observe that as the number of supervised samples in our diverse data set increase, the accuracy of all the networks improves for smaller normalized distance. The joint training of the $\mathrm{Pr}_{\mathbf{w}}$ network also improves the results over the iterative optimization of $\mathrm{Pr}_{\mathbf{w}}$ network.

4.4.4.2 Results on JHMDB data set

The result for training the FS, PW and Pr_p networks for the $50-50$ split on the JHMDB data set are summarized in table 4.2.

Table 4.2 *Results on JHMDB data set (*PCKh@0.5*), where FS is trained using* 50% *percentage of fully annotated data and PW and* Pr_p *are trained on* $50-50$ *split of fully annotated and weakly annotated training data. Here FS and PW are the fully supervised and the pointwise networks respectively, and* Pr_p *(iterative) and* Pr_p *(joint) is our proposed probabilistic network trained with block coordinate optimization and joint optimization respectively.*

Method	FS	PW	Pr_p (iter)	Pr_p (joint)
Total Accuracy	80.01	85.77	89.90	91.25

We observe that the accuracies of the three networks FS, PW and $\Pr_p$) holds similar trends as we had seen for the MPII data set.

4.4.5 Additional Results

To prove the generality of our method, we provide additional results using a different architecture, as proposed by Belagiannis *et al.* [118]. The authors pose the problem of estimating human poses as regression and propose to minimize a novel Tukey's biweight function as loss function for their ConvNet. They empirically show that their method outperforms the simple $L2$ loss. The point-wise architecture, consisting of five convolutional layers and two fully connected layers is modified to a Disco Nets as shown in the figure 4.9 below. A 1024 dimensional noise vector, sampled from a uniform distribution, is appended to the flattened CNN features, before applying fully connected layers.

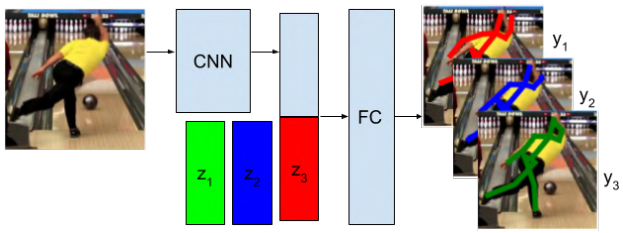


Figure 4.9 Modified architecture, as proposed by Belagiannis *et al.* [118]. The figure shows the sampling process of DISCO Net. The block CNN consists of 5 convolution layers. The middle block is the flattened feature vector obtained after convolution. The block FC consists of two fully connected layers.

We evaluate the performance of the FS, PW and our proposed probabilistic network $\Pr_p$ on $50-50$ split of two data sets, namely (i) MPII Human Pose data set [18], and (ii) JHMDB data set [85]. The MPII and the JHMDB data set is split exactly as it was done for the stacked hourglass network. The results are summarized in Table 4.3.

Table 4.3 *Results on MPII Human Pose data set and JHMDB data set (*PCKh@0.5*), where FS is trained using* 50% *percentage of fully annotated data and PW and* $\Pr_p$ *are trained on* $50-50$ *split of fully annotated and weakly annotated training data. Here FS and PW are the fully supervised and the pointwise networks respectively, and* $\Pr_p$ *(iterative) and* $\Pr_p$ *(joint) is our proposed probabilistic network trained with coordinate optimization and joint optimization respectively.*

Method	MPII	JHMDB
FS	41.89	54.31
PW	54.37	66.19
$\Pr_p$ (iterative)	56.09	71.02
$\Pr_p$ (joint)	**57.28**	**72.61**

We observe that the results shown in Table 4.3 on both the data sets are consistent with our observations on the stacked hourglass network. Networks PW and $\Pr_p$ trained on the diverse data, outperforms

the FS Net, which is trained only using the fully supervised annotations. This demonstrates the advantage of using diverse learning over a fully supervised method. Moreover, our proposed probabilistic net Pr_p outperforms the pointwise network PW, this signifies the importance of modeling uncertainty over pose. We also note that performing joint optimization, after iterative optimization step, further increases our accuracy by 1.2% on MPII Human Pose data set and by 1.4% on JHMDB data set.

4.5 Discussion

We presented a novel framework to learn human pose using diverse data set. Our framework uses two separate distributions: (i) a conditional distribution for modeling uncertainty over pose given the image and the action during training time; and (ii) a prediction distribution to provide pose estimates for a given image. We model the two aforementioned distributions using a deep probabilistic network. We learn these separate yet complimentary distributions by minimizing a dissimilarity coefficient based learning objective. Empirically, we show that: (i) action serves as an important cue for predicting human pose; and (ii) modeling uncertainty over pose is essential for its accurate prediction.

Chapter 5

Weakly Supervised Object Detection

5.1 Introduction

Object detection requires us to localize all the instances of an object category of interest in a given image. In recent years, significant advances in speed and accuracy have been achieved by detection frameworks based on Convolutional Neural Networks (CNNs) [3, 10, 119–123]. Most of the existing methods require a strongly supervised data set, where each image is labeled with the ground-truth bounding boxes of all the object instances. Given the high cost of obtaining such detailed annotations, researchers have explored the weakly supervised object detection (WSOD) problem [32, 124–135]. The goal of Weakly Supervised Object Detection (WSOD) is to learn an accurate detector using training samples that are annotated with more cost-effective labels, such as image-level, count, point, and scribble annotations. Image-level annotations can be as simple as object category labels that indicate the presence of an object, or they can include richer information like per-class object counts, which offer slightly more detailed supervision. Additionally, point and scribble annotations provide a more refined level of guidance by indicating specific object locations (points) or rough object boundaries (scribbles). Although these annotations come at a marginally higher cost than image-level labels, as we shall see, they significantly improve the model's ability to localize objects more accurately during training.

Given the wide availability of such cheaper-to-obtain labels, WSOD offers a cost-effective and highly scalable learning paradigm. However, this comes at the cost of introducing uncertainty in the location of the object instances during training. For example, consider the task of detecting a car. Given a training image annotated with only the presence of a car, we still face the challenge of identifying the precise bounding box for the car. This challenge is somewhat mitigated when additional annotations, such as object counts, points, or scribbles, are available. Object count annotations provide information on the number of instances present, reducing ambiguity about the number of objects to detect. Point annotations, by marking specific locations within the object, help in narrowing down the potential area where the object is located. Scribble annotations, which roughly outline the object, offer even more spatial guidance, making it easier to determine the approximate shape and boundary of the object. Despite these enhancements, WSOD must still contend with the inherent uncertainty introduced by the lack of full supervision as the extent of an object is not known.

In order to effectively model uncertainty in weakly supervised learning, Kumar *et al.* [51], introduced in Section 3.1.2, proposed a probabilistic framework that models two distributions: (i) a *conditional distribution*, which represents the probability of an output conditioned on the given annotation during training; and (ii) a *prediction distribution* which represents the probability of an output at test time. The parameters of the two distributions are estimated jointly by minimizing the *dissimilarity coefficient* [82], which measures the distance between any two distributions using a task specific loss function. This proposed framework was introduced in Section 3.2.

The aforementioned dissimilarity coefficient based framework has provided promising results in domains where the conditional distribution is simple to model (that is, consists of terms that depend on a few variables at a time) [51, 136]. However, WSOD poses greater difficulty due to the complexity of the underlying conditional distribution. Specifically, given the hundreds or even thousands of bounding box proposals for an image, the annotation constraint imposes a term over all of these bounding box proposals such that at least one of them corresponds to the given weak labels, such as image-level, count, point, or scribble annotations. This leads to a challenging scenario where the distribution is not factorizable over the bounding box proposals. While previous works have approximated this uncertainty as a fully factorized distribution for computational efficiency, we argue that such a choice leads to poor accuracy.

To overcome the difficulty of a complex conditional distribution, we make the key observation that deep learning relies on stochastic optimization. Therefore, we do not need to explicitly model this complex distribution but simply estimate the distribution using samples. This observation opens the door to the use of appropriate deep generative models such as the Discrete DISCO Nets [83, 84].

We test the efficacy of our approach on the challenging PASCAL VOC 2007, 2012, and MS COCO 2014, 2017 data sets. To generate the weakly supervised data sets, we discard the bounding box annotations, keeping only the image-level labels and, optionally, keeping the per-class object count, points, or scribbles. Using simple image-level labels we achieve 58.1%, 55.4%, 28.6%, and 28.9% detection mAP@0.5 on PASCAL VOC 2007, 2012, MS COCO 2014 and 2017 data sets respectively, significantly improving the state-of-the-art on all the data sets. Using count supervision provides an average increase of 2.3% detection mAP@0.5 across all data sets. Additionally, using point and scribble annotations we obtain a further increase of 3.3%, and 0.8% detection mAP@0.5 on MS COCO 2014 data set respectively giving state-of-the-art results for WSOD using various types of inexpensive annotations.

To summarize, we make the following contributions.

- A unified weakly supervised framework to train object detectors with varying levels of weak labels, such as image-level, count, point, and scribble annotations.
- Efficiently model the complex non-factorizable, annotation aware, spatially consistent conditional distribution using the deep generative model, the Discrete DISCO Net.
- Empirically show the importance of modeling the uncertainty in the annotations in a single unified probabilistic learning objective, the dissimilarity coefficient.

- State-of-the-art performance for the task of WSOD on challenging PASCAL VOC 2007, PASCAL VOC 2012, MS COCO 2014, and MS COCO 2017 data sets.

5.2 Related Work

Conventional methods often treat WSOD as a Multiple Instance Learning (MIL) problem [42] by representing each image as a bag of instances (that is, putative bounding boxes) [48, 49, 137–139]. The learning procedure alternates between training an object classifier and selecting the most confident positive instances. However, these methods are susceptible to poor initialization. To address this, different strategies have been developed, which aim to improve the initialization [47,138,140,141], regularize the model with extra cues [48, 137], or relax the MIL constraint [49] to make the objective differentiable. These hard-MIL based methods have demonstrated their effectiveness, especially when CNN features are used to represent object proposals [137]. However, these models are not end to end trainable and do not explicitly model the uncertainty.

A more interesting line of work is to integrate MIL strategy as deep networks such that they are end to end trainable [124,125,131–134,142,143]. In their work, Bilen *et al.* [124] proposed a smoothed version of MIL that softly labels object proposals instead of choosing the highest scoring ones. Building on their work, Tang *et al.* [131] refine the prediction iteratively through a multi-stage instance classifier. Gao *et al.* [127] presents a greedy approach to training a WSOD using per-class object count. Ren *et al.* [144] presents a unified framework that can utilize all weakly supervised labels, such as image-level supervision, point supervision, and scribble supervision, but they don't consider count supervision. Chen *et al.* [145] presented their work that leverages point annotations to train object detectors. In contrast, we propose a unified framework that can learn from any weakly supervised labels. Zhang *et al.* [133] add curriculum learning using the MIL framework. In our formulation, we explicitly incorporate curriculum learning based on object instance count. Tang *et al.* [146] proposes to cluster similar object proposals to better distinguish between the object and background noise. In our framework, we cluster object proposals such that the number of clusters are consistent with object count. Other end-to-end trainable frameworks for WSOD employ domain adaptation [129, 138], expectation-maximization algorithm [32, 126] and saliency based methods [128]. Although these methods are end to end trainable, they not only model a single distribution for two related tasks but also model the complex distribution with a fully factorized one. This design choice makes these approaches sub-optimal as what we truly want is to model a distribution that enforces at least one bounding box proposal corresponding to the given weak label.

To enhance weakly supervised detectors, some approaches combine them with strongly supervised ones, typically using predictions from the weakly supervised detector as pseudo-strong labels to train a strongly supervised network [131, 144, 147–151]. However, this usually involves a unidirectional connection between the two. Wang *et al.* [143] propose a collaborative training approach for weakly and strongly supervised models, similar in spirit to our use of two distributions, though they fully factorize

their weakly supervised detector. Yin *et al.* [151] employ a teacher-student network, using an ensemble of students for diverse pseudo ground truth, but without explicitly modeling uncertainty and using a fully factorized distribution. In contrast, we model uncertainty in the conditional distribution to ensure annotation consistency. The improvements reported in these works highlight the importance of modeling separate distributions. In this work, we explicitly define and jointly train two distributions, minimizing the dissimilarity coefficient [82] based objective function.

5.3 Model

5.3.1 Notation

Following the notations prescribed in Section 3.2.1, we denote an input image as $\mathbf{x} \in \mathbb{R}^{(H \times W \times 3)}$, where H and W are the height and the width of the image respectively. For the sake of simplifying the subsequent description of our approach, we assume that we have extracted B bounding box proposals from each image. In our experiments, we use Selective Search [152] to obtain the aforementioned bounding boxes. Each bounding box proposal, $b^{(i)}$, can belong to one of $C+1$ categories from the set $\{0, 1, \ldots, C\}$, where category 0 is background, and categories $\{1, \ldots, C\}$ are object classes.

We denote the weak annotation by $\mathbf{a} \in 0 \cup \mathbb{Z}^+$. Here, $\mathbf{a}^{(j)} = r$ if image $\mathbf{x}$ contains r instances of the j-th object. We assume $r = 1$ where only object category labels are provided and count information is absent. Furthermore, we denote the unknown bounding box labels by $\mathbf{y} = \{\mathbf{y}^{(i)} \mid \mathbf{y}^{(i)} \in \{0, \ldots, C\}^B \wedge i = 1, \ldots B\}$, where $\mathbf{y}^{(i)} = j$ if the i-th bounding box $b^{(i)}$ is of the j-th category. A weakly supervised data set $\mathcal{W} = \{(\mathbf{x}_i, \mathbf{a}_i) \mid i = 1, \ldots, N\}$ contains N pairs of images $\mathbf{x}_i$ and their corresponding image-level labels $\mathbf{a}_i$. For point and scribble annotations, we retain only those bounding box proposals that fully encompass the annotation. This approach ensures their compatibility with count supervision.

5.3.2 Probabilistic Modeling

Given a weakly supervised data set $\mathcal{W}$, we wish to learn an object detector that can predict the bounding box labels $\mathbf{y}$ of a previously unseen image. Due to the uncertainty inherent in this task, we advocate the use of a probabilistic formulation. Following Section 3.2, we define two distributions. The first one is the *prediction distribution* $\Pr_p(\mathbf{y}|\mathbf{x}; \boldsymbol{\theta}_p)$, which models the probability of the bounding box labels $\mathbf{y}$ given an input image $\mathbf{x}$. Here $\boldsymbol{\theta}_p$ are the parameters of the distribution. As the name suggest, this distribution is used to make the prediction at test time.

In addition to the prediction distribution, we also construct a *conditional distribution* $\Pr_c(\mathbf{y}|\mathbf{x}, \mathbf{a}; \boldsymbol{\theta}_c)$, which models the probability of the bounding box labels $\mathbf{y}$ given the input image $\mathbf{x}$ and its weak annotations $\mathbf{a}$. Here $\boldsymbol{\theta}_c$ are the parameters of the distribution. The conditional distribution contains additional information, namely the presence of foreground objects in each image, or optionally object instance count or localization information through point or scribble annotations. Thus, we can expect it to provide better predictions for the bounding box labels $\mathbf{y}$. We will use this property during training in

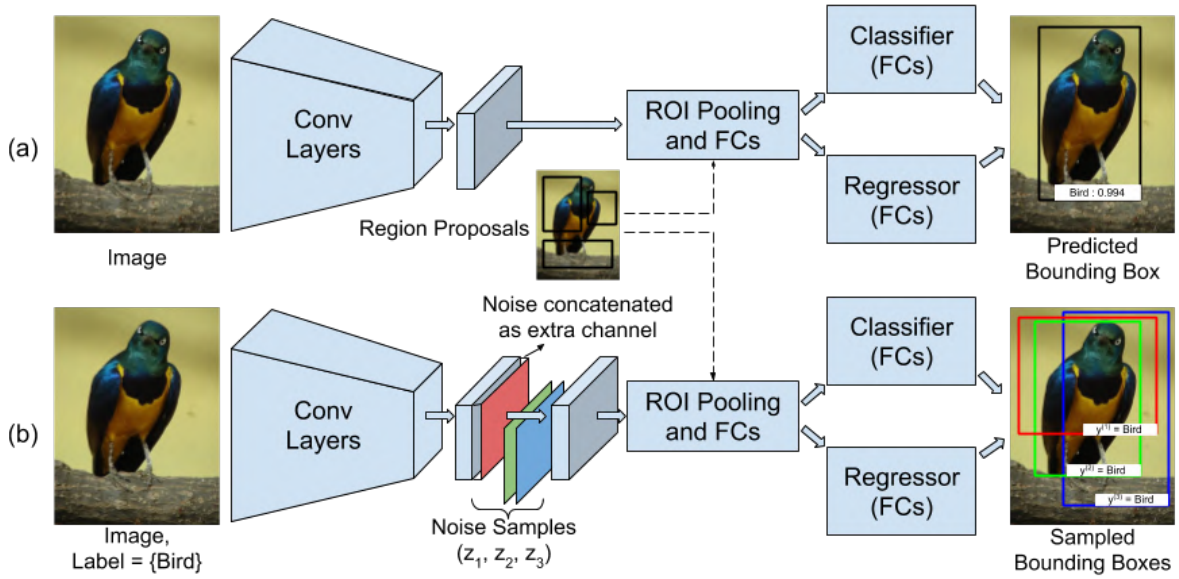


Figure 5.1 *The overall architecture. (a) Prediction Network: a standard Fast-RCNN architecture is used to model the prediction net. For an input image, bounding box proposals are generated from selective search [152]. Features from each of these proposals are computed by the region of interest (ROI) pooling layers, which are then passed through the classifier and regressor to predict the final bounding box. (b) Conditional Network: a modified Fast-RCNN architecture is used to model the conditional net. For a single input image* $\mathbf{x}$ *and three different noise samples* $\{\mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3\}$ *(represented as red, green and blue matrix), three different bounding boxes* $\{\mathbf{y}^{(1)}, \mathbf{y}^{(2)}, \mathbf{y}^{(3)}\}$ *are sampled for the given image-level label (bird in this example). Here the noise filter is concatenated as an extra channel to the final convolutional layer. For both the networks, the initial conv-layers are fixed during training. Best viewed in color.*

order to learn an accurate prediction distribution using the conditional distribution. The details on the modeling of the two distributions are discussed below.

5.3.2.1 Prediction Distribution

The task of the prediction distribution is to accurately model the probability of the bounding box labels given the input image. Taking inspiration from the supervised models [3, 120, 121], we assume independence between the probability of the output for each bounding box proposal. Therefore, the overall distribution for an image equals the product of the probabilities of each proposal,

$$\Pr_p(\mathbf{y}|\mathbf{x};\boldsymbol{\theta}_p) = \prod_{i=1}^{B} \Pr_p(\mathbf{y}^{(i)}|\mathbf{x};\boldsymbol{\theta}_p). \tag{5.1}$$

We model this distribution using the Fast-RCNN architecture [120] (see Figure 5.1(a)). As the prediction distribution is specified by a neural network, we henceforth refer to it as the *prediction net*. In this setting, the parameters of the distribution $\boldsymbol{\theta}_p$ are the weights of the prediction net.

5.3.2.2 Conditional Distribution

Given B bounding box proposals for an image $\mathbf{x}$ and the weak annotation $\mathbf{a}$, the conditional distribution $\Pr_c(\mathbf{y}|\mathbf{x}, \mathbf{a}; \boldsymbol{\theta}_c)$ models the probability of bounding box labels $\mathbf{y}$ under the constraint that they are compatible with the annotation $\mathbf{a}$. Specifically, we divide the B proposals into multiple clusters. Each cluster of bounding boxes corresponds to a foreground object. The total number of clusters of each foreground class should be equal to their image-level annotation $\mathbf{a}^j = r$.

Note that due to the requirement that the bounding box labels $\mathbf{y}$ are compatible with the annotation $\mathbf{a}$, the conditional distribution cannot be trivially decomposed over bounding box proposals. This is in stark contrast to the simple prediction net, which uses a fully factorized distribution. If one were to explicitly model the conditional distribution, then one would be required to compute its partition function during training, which would be prohibitively expensive. To alleviate this computational challenge, we make a key observation that in practice we only need access to a representative set of samples from the conditional distribution. This opens the door to the use of Discrete DISCO Nets [84]. In what follows, we briefly describe how Discrete DISCO Nets are adapted to our framework.

5.3.2.2.1 Discrete DISCO Nets Discrete DISCO Nets [84] is a deep probabilistic framework that implicitly represents a probability distribution over a discrete structured output space. The strength of the framework lies in the fact that it allows us to adapt a pointwise deep network (a network that provides a single pointwise prediction) to a probabilistic one by the introduction of noise. Further discussion on Discrete DISCO Nets is presented in Section 3.1.3.1.

In the context of our setting, consider the modified Fast-RCNN network in Figure 5.1(b) for the conditional distribution. Once again, as we are using a neural network, we will henceforth refer to it as the *conditional net*. The parameters of the conditional distribution $\boldsymbol{\theta}_c$ are the weights of the conditional net. The colored filters in the middle of the network represent the noise that is sampled from a uniform distribution. Each value of the noise filter $\mathbf{z}^k$ results in a different score function[1] $\mathcal{F}^k_{u,\mathbf{y}_u}(\boldsymbol{\theta}_c) \in \mathbb{R}^{B\times C}$ for each bounding box proposal u, and the corresponding putative label $\mathbf{y}_u$. We generate K different score functions using K different noise samples. These score functions are then used to sample the corresponding bounding box labels $\hat{\mathbf{y}}^k_c$ such that all ground truth labels are included in it. This enables us to generate samples from the underlying distribution encoded by the network parameters. Note that obtaining a single sample is as efficient as a simple forward pass through the network. By placing the filters sufficiently far away from the output layer of the network, we can learn a highly non-linear

[1]The use of score function in this paper should not be confused with the scoring rule theory, which is used to design the learning objective of DISCO Nets.

mapping from the uniform distribution (used to generate the noise filter) to the output distribution (used to generate bounding box labels).

In what follows, we will discuss how to redefine the score function $\mathcal{F}^k_{u,\mathbf{y}_u}(\boldsymbol{\theta}_c)$ to obtain a final score function such that it is used to sample the bounding box proposal $\hat{\mathbf{y}}^k_c$.

Initialization by Class Activation Maps In order to incorporate prior knowledge about potential object location, we weigh the score function $\mathcal{F}^k_{u,\mathbf{y}_u}(\boldsymbol{\theta}_c)$ with class activation maps (CAMs) $\mathcal{C}(\mathbf{y_c})$ [62, 63].

$$\mathcal{G}^k_{u,\mathbf{y}_u}(\mathbf{y}_c) = \mathcal{C}(\mathbf{y}_c) \times \mathcal{F}^k_{u,\mathbf{y}_u}(\boldsymbol{\theta}_c). \tag{5.2}$$

While, we can employ any CAM algorithm, in our experiments, we employ Layer-CAM [153]. When no CAM based algorithm is used, we set $\mathcal{C}(\mathbf{y_c}) = 1$.

Cluster Construction In order to effectively use the count information whenever they are available, we propose to cluster the bounding box proposals such that the number of clusters is equal to the count annotation. To form clusters, the proposals are sorted by their object confidences $\mathcal{G}^k_{u,\mathbf{y}_u}(\mathbf{y}_c)$ and the following steps are iteratively performed:

1. Construct a cluster using the proposal with the highest object confidence for the r non-overlapping instances. This ensures that the number of clusters is consistent with image-level label $\mathbf{a}^{(j)} = r$.

2. Find proposals that overlap with a proposal in the cluster by more than 0.7 and merge them into the cluster.

All object instances not forming part of the foreground objects are considered background boxes. The pseudocode for cluster construction is presented in algorithm 5.

Spatial cluster regularization For each bounding box in a cluster corresponding to the foreground object instance, we can redefine our score function such that highly overlapping proposal bounding boxes should have similar scores and labels

$$\mathcal{G}^{k,n}_{u,\mathbf{y}_u}(\mathbf{y}_c) = \mathcal{G}^{k,n-1}_{u,\mathbf{y}_u}(\mathbf{y}_c) + \sum_{i=1}^{B^r \setminus u} \mathbf{w}_i \mathcal{G}^{k,n-1}_{i,\mathbf{y}_i}(\mathbf{y}_c), \tag{5.3}$$

where n is the iterator, B^r are the bounding boxes belonging to a particular cluster, and $\mathbf{w}_i = IOU(b_u, b_i)$ is the IOU between the two proposal boxes. Equation (5.3) is iteratively updated until the scores, weighted by their IOUs, converge. While the algorithm guarantees convergence to a local minimum, in practice, we limit the process to 5 iterations or until the scores stabilize. Empirical evidence shows that 5 iterations are typically sufficient for convergence, providing a good balance between accuracy and speed.

Annotation consistent constraint Finally, we would like to add a constraint such that there must exist at least one bounding box in each clique that satisfies the annotation $\mathbf{a}$.

$$\mathcal{S}^k(\mathbf{y}_c) = \sum_{i=1}^{B^r} \mathcal{G}^{k,n}_{u,\mathbf{y}_u}(\mathbf{y}_c) - \mathcal{H}_k(\mathbf{y}_c), \tag{5.4}$$

Algorithm 5 Cluster Construction

Input: Bounding boxes B, scores $\mathcal{G}^k_{u,\mathbf{y}_u}(\mathbf{y}_c)$, annotations $\mathbf{a}^{(j)} = r$, IoU threshold τ

Output: Dictionary of class-specific clusters with keys $\mathbf{a}^{(j)}$ and values as a list of exactly r clusters

Initialize a dictionary ‘dict‘ with keys $\mathbf{a}^{(j)}$ and values as empty lists
for all annotations $\mathbf{a}^{(j)} > 0$ **do**
    Initialize variables:
    Initialize an empty list ‘clusters‘
    Initialize a boolean array ‘used_boxes‘ of length $|B|$ to track used boxes
    Sort boxes and scores based on the maximum scores corresponding to $\mathbf{a}^{(j)}$ in descending order
    for all boxes b in sorted order **do**
        if number of clusters $\geq r$ **then**
            break
        end if
        if b is not used **then**
            Create a new cluster with b
            Mark b as used
            for all remaining boxes b' **do**
                if b' is not used and IoU(b, b') $\geq \tau$ **then**
                    Add b' to the cluster
                    Mark b' as used
                end if
            end for
            Add the cluster to ‘clusters‘
        end if
    end for
    Ensure exactly r clusters by merging or splitting:
    if number of clusters $< r$ **then**
        while number of clusters $< r$ **do**
            Split the largest cluster into two smaller clusters
        end while
    else if number of clusters $> r$ **then**
        while number of clusters $> r$ **do**
            Merge the two most similar or overlapping clusters
        end while
    end if
    ‘dict‘[$\mathbf{a}^{(j)}$] $\leftarrow$ ‘clusters‘
end for
return ‘dict‘

where,

$$\mathcal{H}_k(\mathbf{y}_c) = \begin{cases} 0 & \text{if } \forall j \in \{1, \ldots, C\} \text{ s.t. } \mathbf{a}^{(j)} = r, \\ & \exists i \in \{1, \ldots, B\} \text{ s.t. } \mathbf{y}^{(i)} = j, \\ \infty & \text{otherwise.} \end{cases} \tag{5.5}$$

Algorithm 6 Conditional Net Inference Algorithm

Input: A dictionary of class-specific clusters 'dict', original scores $\mathcal{S}^k(\mathbf{y}_c)$, annotations $\mathbf{a}^{(j)} = r$
Output: A dictionary Y containing a list of r maximum scoring boxes for each $\mathbf{a}^{(j)}$

for all annotation $\mathbf{a}^{(j)}$ in 'dict' **do**
  Initialize an empty list 'max_boxes'
  for all clusters B^r in 'dict[$\mathbf{a}^{(j)}$]' **do** ▷ Iterative algorithm for spatial cluster regularization
    repeat$\mathcal{G}^{k,n}_{u,\mathbf{y}_u}(\mathbf{y}_c)$ has converged
      for all $b_u, b_i \in B^r$ **do**
        Update the scoring function:

$$\mathcal{G}^{k,n}_{u,\mathbf{y}_u}(\mathbf{y}_c) = \mathcal{G}^{k,n-1}_{u,\mathbf{y}_u}(\mathbf{y}_c) + \sum_{i=1}^{B^r \setminus u} \mathbf{w}_i \mathcal{G}^{k,n-1}_{i,\mathbf{y}_i}(\mathbf{y}_c)$$

      end for
    until $\mathcal{G}^{k,n}_{u,\mathbf{y}_u}(\mathbf{y}_c)$ has converged
    ▷ Greedily select the maximum scoring bounding box
    $Y[\mathbf{a}^{(j)}] \leftarrow \arg\max_{y \in B^r} \mathcal{G}^{k,n}_{u,\mathbf{y}_u}(\mathbf{y}_c)$
  end for
end for
return Y

Given the scoring function in equation (5.4), we compute the k-th sample as

$$\hat{\mathbf{y}}^k_c = \arg\max_{\mathbf{y} \in \mathcal{Y}} \mathcal{S}^k(\mathbf{y}_c). \tag{5.6}$$

Note that in equation (5.6) the $\arg\max$ needs to be computed over the entire output space $\mathcal{Y}$. A naïve brute force algorithm for this would be computationally infeasible. However, by using the structure of the higher order term $\mathcal{H}_k$, we can design an efficient yet exact algorithm for equation (5.6). Specifically, we assign each bounding box proposal u to its maximum scoring object class. If all the ground truth annotations $\mathbf{a}$ are not present in the generated bounding box labels, then we sample the bounding box that has the highest score corresponding to the foreground label. The pseudocode for conditional net inference is presented in algorithm 6.

For point and scribble supervision, we retain only the bounding box proposals that fully contain the annotations. This approach not only narrows the problem's search space but also ensures compatibility with object instance count supervision.

5.4 Learning Objective

In order to estimate the parameters of the prediction and conditional distribution, $\boldsymbol{\theta}_p$ and $\boldsymbol{\theta}_c$, we define a unified probabilistic learning objective based on the dissimilarity coefficient [82]. To this end, we require a task specific loss function, which we define next.

5.4.1 Task Specific Loss Function

We define a loss function for object detection that decomposes over the bounding box proposals as follows:

$$\Delta(\mathbf{y}_1, \mathbf{y}_2) = \frac{1}{B}\sum_{i=1}^{B} \Delta(\mathbf{y}_1^{(i)}, \mathbf{y}_2^{(i)}). \tag{5.7}$$

Following the standard practice in most modern object detectors [154], $\Delta(\mathbf{y}_1^{(i)}, \mathbf{y}_2^{(i)})$ is further decomposed as a weighted combination of the classification loss and the localization loss. We use λ to denote the loss ratio (ratio of the weight of localization loss to the weight of classification loss). We use a simple $0-1$ loss as our classification loss Δ_{cls}, and $smoothL1$ [120] for our localization loss Δ_{loc}. Formally, the task specific loss is given by,

$$\Delta(\mathbf{y}_1^{(i)}, \mathbf{y}_2^{(i)}) = \Delta_{cls}(\mathbf{y}_1^{(i)}, \mathbf{y}_2^{(i)}) + \lambda\Delta_{loc}(b_1^{(i)}, b_2^{(i)}). \tag{5.8}$$

Here, $b_1^{(i)}$ and $b_2^{(i)}$ are the corresponding bounding box proposals for $\mathbf{y}_1^{(i)}$ and $\mathbf{y}_2^{(i)}$.

5.4.2 Objective Function

The task of both the prediction distribution and the conditional distribution is to predict the bounding box labels. Moreover, as the conditional distribution utilizes the extra information in the form of the image-level label, it is expected to provide more accurate predictions for the bounding box labels $\mathbf{y}$. Leveraging the task similarity between the two distributions, we aim to bring them closer so that the extra knowledge of the conditional distribution can be effectively transferred to the prediction distribution.

To achieve this, we use the joint learning objective introduced in Section 3.2.3, which minimizes the dissimilarity coefficient [82] between the prediction and conditional distributions. Our overall learning objective for the task-specific loss Δ, tuned for object detection, follows the formulation presented in Equation (3.2.3) and is expressed as:

$$\boldsymbol{\theta}_p^*, \boldsymbol{\theta}_c^* = \underset{\boldsymbol{\theta}_p, \boldsymbol{\theta}_c}{\arg\min} \sum_{i=1}^{N} DISC_{\Delta}(\Pr_p(\mathbf{y}|\mathbf{x};\boldsymbol{\theta}_p), Pr_c(\mathbf{y}|\mathbf{a},\mathbf{x};\boldsymbol{\theta}_c)). \tag{5.9}$$

where $DISC_{\Delta}$ measures the dissimilarity between the prediction and conditional distributions using Δ, a task-specific loss function designed for object detection. This formulation ensures that the prediction distribution learns to replicate the conditional distribution's enhanced accuracy for bounding box labels.

The dissimilarity coefficient consists of self-diversity terms and a cross-diversity term, as outlined in Section 3.1.1. As discussed in Section 5.3.2, directly modeling the conditional distribution is challenging. Consequently, the corresponding diversity terms are estimated stochastically using K samples $\hat{\mathbf{y}}_c^k$ generated by the conditional network.

Thus, using equations (3.15, 3.17) to compute self-diversity terms, and equation (3.10) to compute cross-diversity term for the given task specifc loss (equation (5.7)), we obtain,

$$DIV_{\Delta}(\mathrm{Pr}_p, \mathrm{Pr}_c) = \frac{1}{BK} \sum_{i=1}^{B} \sum_{k=1}^{K} \sum_{\mathbf{y}_p^{(i)}} \mathrm{Pr}_p(\mathbf{y}_p^{(i)}; \boldsymbol{\theta}_p) \Delta(\mathbf{y}_p^{(i)}, \hat{\mathbf{y}}_c^{k,(i)}), \tag{5.10}$$

$$DIV_{\Delta}(\mathrm{Pr}_c, \mathrm{Pr}_c) = \frac{1}{K(K-1)B} \sum_{\substack{k,k'=1 \\ k' \neq k}}^{K} \sum_{i=1}^{B} \Delta(\hat{\mathbf{y}}_c^{k,(i)}, \hat{\mathbf{y}'}_c^{k',(i)}), \tag{5.11}$$

$$DIV_{\Delta}(\mathrm{Pr}_p, \mathrm{Pr}_p) = \frac{1}{B} \sum_{i=1}^{B} \sum_{\mathbf{y}_p^{(i)}} \sum_{\mathbf{y'}_p^{(i)}} \mathrm{Pr}_p(\mathbf{y}_p^{(i)}; \boldsymbol{\theta}_p) \, \mathrm{Pr}_p(\mathbf{y'}_p^{(i)}; \boldsymbol{\theta}_p) \Delta(\mathbf{y}_p^{(i)}, \mathbf{y'}_p^{(i)}). \tag{5.12}$$

Here, $DIV_{\Delta}(\mathrm{Pr}_p, \mathrm{Pr}_c)$ measures the diversity between the prediction net and the conditional net, which is the expected difference between the samples from the two distributions as measured by the task specific loss function Δ. Here Pr_p is explicitly modeled, hence the expectation of its sample can be computed easily. However, as Pr_c is not explicitly modeled, we compute the required expectation by drawing K samples from the distribution. Likewise, $DIV_{\Delta}(\mathrm{Pr}_c, \mathrm{Pr}_c)$ measures the self diversity of the conditional net. We draw K samples from the distribution to compute the required expectation. Also, the self diversity of the prediction net $DIV_{\Delta}(\mathrm{Pr}_p, \mathrm{Pr}_p)$ can be exactly computed as Pr_p is explicitly modeled.

5.5 Optimization

As explained in Section 3.2.4, since we employ deep neural networks to model the two distributions, our objective function (3.6) is ideally suited to be minimized by stochastic gradient descent. While it may be possible to compute the gradients of both networks simultaneously, in this work we use a simple coordinate descent optimization strategy. In more detail, the optimization proceeds by iteratively fixing the prediction network and learning the conditional network, followed by learning the prediction network for the fixed conditional network. The details of the learning process is specified in Sections 3.2.4.1 and 3.2.4.2.

For the case where object count labels are present, we employ a simple *curriculum-learning* based strategy. We first iteratively train the two networks for images with images that have a single object count. Next, we progressively increase the number of objects present in the training image.

5.5.1 Visualization of the learning process

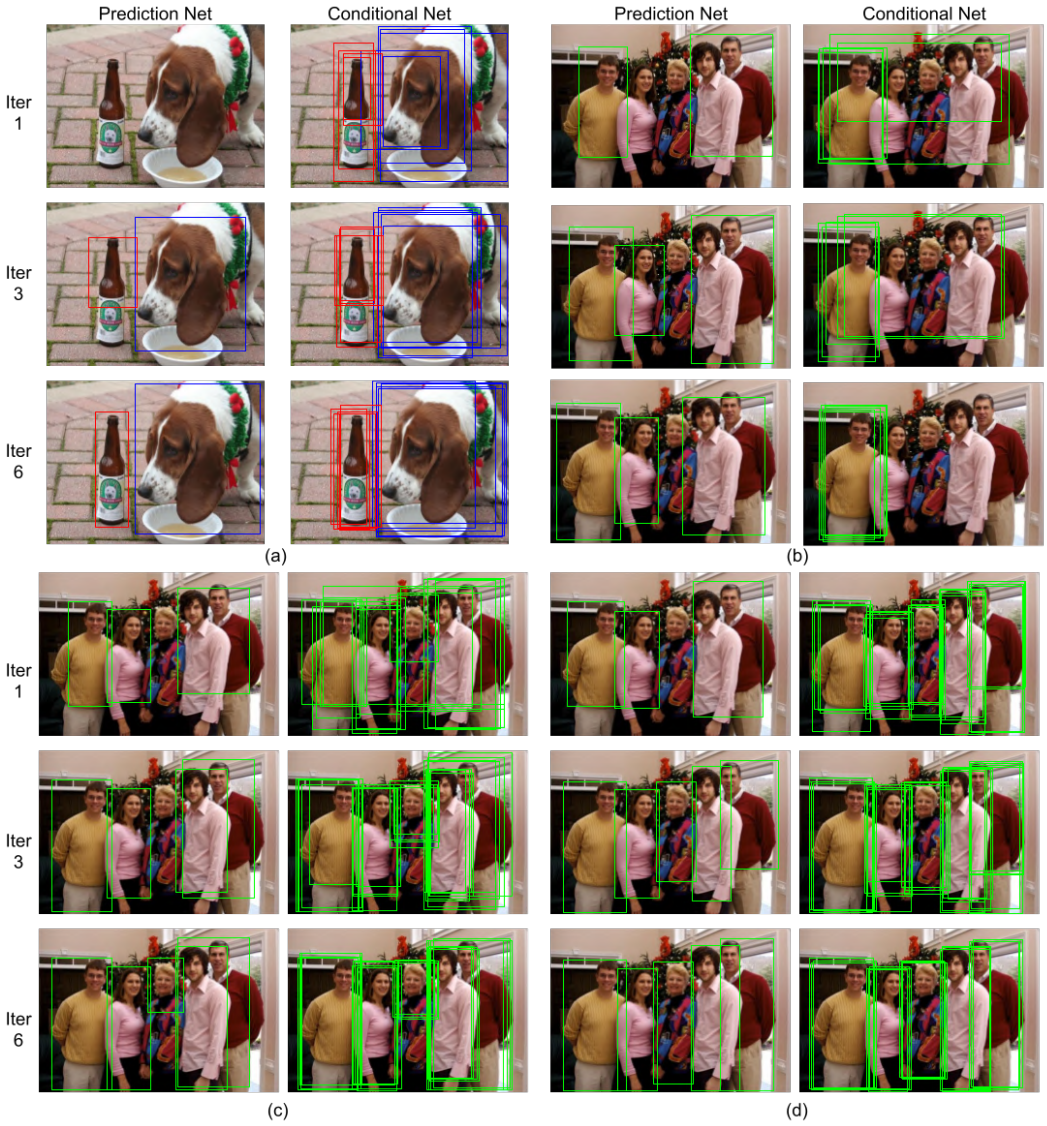


Figure 5.2 *Visualization of prediction and conditional network outputs. For the prediction network, results are shown after applying non-maximal suppression with a score threshold of* 0.7. *Columns 1 and 3 show predictions from the prediction network, while columns 2 and 4 show those from the conditional network, with rows 1, 2, and 3 corresponding to predictions after the first, third, and sixth iterations, respectively. Image set (a) represents a simple case with single object instances and image-level annotations during training, while image set (b) illustrates a complex scenario with multiple object instances and image-level annotations. Image sets (c) and (d) depict the same complex scenario with count and point supervision, respectively. Object classes are color-coded: green for* person*, red for* bottle*, and blue for* dog.

Figure 5.2 provides the visualization of the performance of the two networks over the different iterations of the iterative learning procedure. Figure 5.2(a) demonstrates a simple example where single instance of each object is present and only image-level annotations are present during training. Figure 5.2(b) demonstrates a more complex example where several instances of the same object are present and only image-level annotations are present during training. Figure 5.2(c) and 5.2(d) demonstrates the complex example in presence of count annotations and point annotations during training respectively. The estimated bounding box labels from the prediction net and those sampled from the conditional net are depicted. For conditional net, we superimpose five different samples of bounding box labels. If all the samples agree with each other on bounding box labels, the bounding boxes will have a high overlap, otherwise they will be scattered across the image. For visualization purposes only, a standard non maximal suppression (NMS) is applied with a score threshold of 0.7 on the output of the prediction net. However, note that the non maximal suppression is not used during the training of the prediction net. The two steps of the iterative algorithm are described below in brief. For completeness, the details are provided in Appendix B.

In order to visualize the learning process, let us first consider the simple example (Figure 5.2(a)), where only image-level annotations are present during training. We observe that initially (in iteration 1), the conditional net's samples for both *dog* and *bottle* objects have high uncertainty, meaning the samples are spread out and lack consensus. However, they are broadly localized over the object, an information that can be exploited by our algorithm. The same is also reflected in the output of the prediction net, which is unable to detect either object. Over the iterations, the knowledge from the conditional net is transferred to the prediction net, and we see a gradual improvement in the uncertainty of both the prediction net and the conditional net, finally resulting in accurate localization of both the objects.

Figure 5.2(b) presents a challenging example where multiple instances of the object *person* are present, and only image-level annotations are present during training. We observe that initially the conditional net samples are extremely diverse (and has high uncertainty). Some samples correctly localizes one of the instances of the class *person*, but others span multiple instances of that class. The output of the prediction net also reflects this, with partial localization of one of the object instances and incorrect localization that contains multiple instances or no localization of an instance of the *person* class. Over the iterations, the uncertainty of the prediction and the conditional net reduces, and we see a better localization. Finally, the conditional network has low uncertainty in its samples, even though it misses several instances of the object. The prediction net succesfully localizes several instances of the class *person*, as it also learns from other images containing the *person* class in the data set during iterative training. However, we see that the final output of the prediction net remains imperfect with some instances not localized and some localization containing multiple instances of the same object.

In figure 5.2(c), we see the sample challenging example where count annotations are present during training. We observe that due to our cluster construction (section 5.3.2.2), we can now take multiple samples from the conditional net. Although, initially the uncertainty of the conditional net is high, the

samples obtained are better localized than the case where only image-level annotations were present. Over the iterations, we see the uncertainty in both the prediction and conditional nets reducing. We note that in this case, many of the instances are correctly localized but some instances are either partially localized and some localization contains multiple instances.

Finally, in figure 5.2(d), we consider the challenging example where point annotations are available. We observe that initially the uncertainty of the prediction net is high, but the uncertainty in the conditional net is low. Over the iterations, the information present in the conditional net is successfully transferred to the prediction net, where the final output accurately localizes all instances of the same class.

5.6 Experiments

5.6.1 Data set and Evaluation Metrics

Data set: We evaluate our method on the challenging VOC 2007, and VOC 2012 in PASCAL VOC [13], and COCO 2014 and COCO 2017 in MS COCO [14] data sets. We use the trainval set in VOC 2007 and VOC 2012 data sets that has $5,011$ and $11,540$ images respectively for 20 object categories, and the test set contains $4,951$ and $10,991$ images for evaluation. COCO 2014 data includes around $82,783$ images for training and $40,504$ images for validation for 80 object categories. COCO 2017 has $118,287$ images in the train set and $5,000$ images in the validation set.

As we focus on weakly supervised detection, only image-level labels (I) are utilized during training. We retain instance count information (C) for count supervision. For point annotations (P), we use quasi-center point annotations, where the center of the ground-truth bounding boxes serves as the point annotation. However, if there is an overlap between bounding boxes, we select the nearest non-overlapping point from the center box. In cases where the point annotation falls outside the object or is contained inside other bounding box, we do not make corrections. For scribble (S) supervision, we adopt the setup proposed by Ren *et al.* [144]. Note that Ren *et al.* [149] provide scribble annotations only for COCO 2014 data set. Therefore, for scribble supervision, we only consider COCO 2014 data set.

Evaluation Metric We use two metrics to evaluate our detection performance on the PASCAL VOC data set. First, we evaluate detection using mean Average Precision (mAP) on the PASCAL VOC 2007 and 2012 test sets, following the standard PASCAL VOC protocol [13]. Second, we compute CorLoc [155] on the PASCAL VOC 2007 and 2012 trainval splits. CorLoc is the fraction of positive training images in which we localize an object of the target category correctly. Following [13], a detected bounding box is considered correct if it has at least 0.5 IoU with a ground truth bounding box.

MS-COCO presents a greater challenge compared to PASCAL VOC, as it contains significantly more instances per image (approximately 7 versus 2) and a larger number of classes (80 versus 20). We report mAP results at IoU thresholds of 0.5 and 0.75, along with the more comprehensive AP metric. AP is calculated as the average mAP across 10 IoU thresholds, ranging from 0.5 to 0.95 in 0.05 increments.

5.6.2 Implementation Details

We use standard Fast-RCNN [120] to model prediction distribution and a modified Fast-RCNN to model the conditional distribution, as shown in Figure 5.1(a). We use the ImageNet pre-trained VGG16 Network [156] and ImageNet pre-trained ResNet network [157] as the base CNN architectures for both our prediction and conditional nets.

The Fast-RCNN architecture is modified by adding a noise filter in its 5^{th} conv-layer as an extra channel as shown in Figure 5.1(b). A 1×1 filter is used to bring the number of channels back to the original dimensions (512 channels). No architectural changes are made to the prediction net. The bounding box proposals required for the Fast-RCNN are obtained from the Selective Search algorithm [152]. Results based on the Region Proposal Networks are given in the supplementary material.

For all our experiments we choose $K = 5$ for the conditional net. That is, we sample 5 bounding boxes corresponding to 5 noise filters, which are themselves sampled from a uniform distribution. For all other hyper-parameters, we use the same configurations as described in [120].

In order to initiate the training of our proposed framework, we first train the conditional network using the thresholded CAM output as a pseudo bounding box label. Specifically, we threshold the CAM output at 0.7 and create a bounding box that tightly encloses the resulting mask. When count information (C) is available, we ensure that the number of pseudo bounding boxes matches the count annotation. If point (P) or scribble (S) annotations are available, we retain only those bounding box proposals that contain the corresponding point or scribble annotation.

5.6.3 Results

In this subsection, we first compare our method with the current state-of-the-art approaches for detection and correct localization tasks on the PASCAL VOC data sets, as well as for detection task on the MS COCO data sets. Next, through ablation experiments, we examine how the different components used to redefine the score function and various terms in our dissimilarity coefficient-based objective function contribute to the improvement in accuracy.

5.6.3.1 Comparison with other methods

We compare our proposed method with other state-of-the-art weakly supervised methods with varying levels of weak supervision. The performance on detection average precision and correct localization metrics for the PASCAL VOC data sets and the detection average precision metrics for the MS COCO data sets are presented in table 5.1. We employ two different backbones for our networks, VGG-16 [156], and ResNet-50 [157]. Compared with the other methods, our proposed framework achieves state-of-the-art performance using a single model and using the selective search for bounding box proposals across varying levels of weak supervision. This demonstrates the efficacy and generalizability of our proposed approach. We also observe a consistent gain of accuracy ($> 1\%$) when using a bigger model that uses ResNet-50, over the baseline model that has VGG-16 as its backbone. Although not

Table 5.1 *Comparison with the state-of-the-art WSOD methods on PASCAL VOC and MS COCO data sets.*

Method	Sup.	Backbone	VOC 2007		VOC 2012		COCO 2014			COCO 2017		
			mAP	CorLoc	mAP	CorLoc	Avg. Precision, IoU:			Avg. Precision, IoU:		
							0.5:0.95	0.5	0.75	0.5:0.95	0.5	0.75
WSDDN [124]	*I*	VGG16	34.8	53.5	–	–	9.5	19.2	8.2	–	–	–
OICR [131]	*I*	VGG16	47.0	64.3	42.5	65.6	7.7	17.4	–	–	–	–
WSOD^2 [147]	*I*	VGG16	53.6	69.5	47.2	71.9	10.8	22.7	–	–	–	–
C-MIDN [148]	*I*	VGG16	52.6	68.7	50.2	71.2	9.6	21.4	–	–	–	–
MIST [149]	*I*	VGG16	54.9	68.8	52.1	70.9	11.4	24.3	9.4	12.4	25.8	10.5
OD-WSCL [150]	*I*	VGG16	56.1	69.8	54.6	71.2	14.4	**29.0**	12.4	13.6	27.4	12.2
CBL [151]	*I*	VGG16	57.4	71.8	53.5	72.6	13.6	27.6	–	–	–	–
PredNet (Ours)	*I*	VGG16	**58.1**	**72.4**	**55.4**	**72.9**	**14.8**	28.6	**14.2**	**15.1**	**28.9**	**14.6**
OICR [131]	*I*	R-50	50.1	–	–	–	–	–	–	–	–	–
OD-WSCL [150]	*I*	R-50	56.6	–	–	–	13.9	29.1	11.8	13.8	27.8	12.1
PredNet (Ours)	*I*	R-50	**59.4**	**73.9**	**56.6**	**74.8**	**15.4**	**28.9**	**14.9**	**15.9**	**29.8**	**15.1**
C-WSL [127]	*C*	VGG16	48.2	66.1	45.4	66.9	–	–	–	–	–	–
PredNet (Ours)	*C*	VGG16	**59.6**	**74.1**	**56.8**	**75.1**	**17.2**	**31.6**	**15.5**	**17.8**	**32.1**	**16.4**
PredNet (Ours)	*C*	R-50	**60.7**	**74.9**	**57.0**	**76.3**	**17.6**	**31.9**	**15.7**	**18.3**	**32.3**	**16.7**
UFO^2 [144]	*P*	VGG16	–	–	–	–	12.4	27.0	–	13.5	27.9	–
PredNet (Ours)	*P*	VGG16	**60.1**	**74.4**	**57.2**	**75.4**	**19.0**	**34.9**	**17.9**	**19.6**	**35.2**	**19.3**
P2BNet [145]	*P*	R-50	–	–	–	–	19.4	**43.5**	–	**22.1**	**47.3**	–
PredNet (Ours)	*P*	R-50	**61.0**	**75.4**	**57.4**	**75.7**	**19.9**	36.0	**18.7**	20.7	36.5	**20.1**
UFO^2 [144]	*S*	VGG16	–	–	–	–	13.7	29.8	–	–	–	–
PredNet (Ours)	*S*	VGG16	–	–	–	–	**19.8**	**35.7**	**19.0**	–	–	–
PredNet (Ours)	*S*	R-50	–	–	–	–	**21.1**	**37.8**	**20.2**	–	–	–

surprising, this trends demonstrate that our method is scalable and the accuracies can further improve when using a bigger model that has better representational capacity (such as ResNet-101 or ViT).

Using image level annotations (I), our method significantly outperforms other state-of-the-art methods. Inspired by Bilen *et al.* [124], prior arts [124, 131, 147–151] employ a fully factorized distribution in MIL objective. We empirically demonstrate the usefulness of modeling a complex distribution. Compared to previous arts [131, 147–151] that uses two different networks, one for pseudo bounding box generation, and another Fast-RCNN [120] for inference, our iterative training of both the networks using a joint objective enables us to achieve superior performance. Compared to CBL [151], that generates multiple pseudo bounding box labels using an ensemble of student networks to train a teacher network, we get better results by explicitly modeling the uncertainty over the pseudo label generation process and generating unbiased samples using the conditional network.

When we have access to instance count annotations (C), our results improve significantly over the image-level annotation baseline. This is especially noticeable ($+2.4\%$ and $+2.7\%$ AP for COCO 2014 and COCO 2017 respectively) for the MS COCO data sets that have higher instance counts per image compared to the PASCAL VOC data sets. This improvement is attributed to the cluster construction and the use of curriculum learning based on the instance count during training.

Using point annotation (P), our method further improves the baseline based on count supervision and achieves competitive results overall. Again, this is again noticeable in the more complex MS COCO data sets, where several instances of the same object can be cluttered together, thus making the ground truth point annotation more relevant. Using, scribble supervision (S), we further improve the results obtained using count supervision owing to the use of more accurate annotations. Note that due to our use spatial consistency, the improvement achieved after using scribble supervision over point supervision is not as high, thus highlighting the fact that our spatial consistency term effectively captures the extent of an object.

5.6.4 Ablation Experiments

In this section, we examine the impact of applying CAMs, spatial regularization, and annotation consistency constraints to redefine the score function on the COCO 2017 data set, where instance count information is available. Additionally, we will explore the effects of the diversity coefficient terms and the influence of curriculum learning within the same context.

Table 5.2 *Ablation Experiment: Detection Average Precision on COCO 2017 data set with count annotation (C) under different settings. CAM is Class Activation Maps, SR is Spatial Regularization, and AC is Annotation Consistent Constraint.*

CAM	SR	AC	COCO 2017 (AP (0.5:0.95))
			12.8
✓			14.9
	✓		14.3
		✓	13.6
✓	✓		16.9
✓		✓	15.6
	✓	✓	15.2
✓	✓	✓	17.8

5.6.4.1 Effect of redefining the score function

To obtain accurate bounding box samples from the conditional network, we redefined the score function by incorporating CAM scores, spatial regularization, and an annotation consistency constraint. Table 5.2 illustrates the performance impact of each component and their combinations.

Row 1 represents the baseline scenario, where the highest-scoring bounding box is sampled from the conditional network. While the performance is comparable to other image-level weakly supervised approaches [150].

Incorporating CAM scores yields a significant improvement of 2.1%, underlining the importance of integrating strong priors in the proposed method. Similarly, adding spatial regularization alone leads to a notable performance boost of 1.5%. This improvement can be attributed to spatial regularization's

ability to address the common issue where bounding boxes that cover only the most discriminative part of an object are assigned the highest scores, thereby leading to the selection of more accurate bounding boxes. When the model is constrained to select bounding boxes that are consistent with annotations, a further performance gain of 0.8% is observed. This suggests that enforcing annotation consistency encourages more accurate bounding box sampling.

Moreover, the table demonstrates that these three components are complementary to one another. When combined, their performance improves beyond the individual gains, showing an even more substantial boost in accuracy. The best result, with an AP improvement of 5%, is achieved when all three components—CAM scores, spatial regularization, and annotation consistency—are used together to redefine the score function. This indicates that the synergy between these components is crucial for maximizing detection performance.

5.6.4.2 Effect of the diversity coefficient terms

In order to understand the effect of various diversity coefficient terms in our objective (3.6), we remove the self-diversity term in one or both of our probabilistic networks (Pr_c and Pr_p). To obtain a single sample from our conditional network, we feed a zero noise vector (denoted by PW_c). The prediction network still outputs the probability of each bounding box belonging to each class. However, by removing the self-diversity term, we encourage it to output a peakier distribution (denoted by PW_p). Table 5.3 shows that both the self-diversity terms are important to obtain the maximum accuracy. Relatively speaking, it is more important to include the self-diversity in the conditional network in order to deal with the difficult examples. Moreover, this enforces a diverse set of outputs from the conditional network, which helps the prediction network to avoid overfitting the samples during training.

5.6.4.3 Effect of instance count based curriculum learning

We examine the effect of curriculum learning, which leverages count information (when available) to train the model with increasingly complex images progressively. Implementing curriculum learning results in a performance improvement from 59.4% to 59.6% on the VOC 2007 data set. A more substantial gain is observed on the more complex COCO 2017 data set, where the performance increases from 16.7% to 17.8%. Given that COCO 2017 contains an average of 7 instances per image (compared to VOC 2007 that has an average of 2 instances per image), we argue that employing a simple curriculum aids the model in learning better and more discriminative features during the early stages of training. This enables the model to better grasp the concept of an object, ultimately enhancing its performance. These results also show that our proposed approach is amenable to more complex data sets.

5.6.5 Additional Comments

Weakly supervised approaches have been shown to improve performance when trained with extra data [144], CLIP alignment [158], or when using better region proposals such as MCG [150] or using

Segment Anything Model (SAM) [159]. We consider these approaches to be complementary to our method and can be easily incorporated. However, the scope of our study was to obtain the best performance using diverse weakly supervised data without the need for external data. Additionally, in their paper, Zhou *et al.* [158] uses ground truth bounding boxes during training, violating the weakly supervised setting. A similar issue is present in Seo *et al.* [159] that uses SAM based proposals to obtain superior results. However, SAM [160] itself is partially trained with ground truth segmentation masks, thus violating the weakly supervised setting.

Table 5.3 *Detection Average Precision (%) for various ablative settings on COCO 2017 with instance count annotation (C).*

Method	$\mathrm{Pr}_p, \mathrm{Pr}_c$ (proposed)	Pr_p, PW_c	PW_p, Pr_c	PW_p, PW_c
AP (0.5:0.95)	17.8	15.2	17.4	14.8

5.7 Discussion

We presented a novel framework to train an object detector using a weakly supervised data set. Our framework employs a probabilistic objective based on dissimilarity coefficient to model the uncertainty in the location of objects. We show that explicitly modeling the complex non-factorizable conditional distribution is a necessary modeling choice and present an efficient mechanism based on a discrete generative model, the Discrete DISCO Nets, to do so. Extensive experiments on the benchmark data sets have shown that our framework successfully transfers the information present in the image-level annotations for the task of object detection.

Chapter 6

Weakly Supervised Instance Segmentation

6.1 Introduction

The instance segmentation task is to jointly estimate the class labels and segmentation masks of the individual objects in an image. Significant progress on instance segmentation has been made based on the convolutional neural networks (CNN) [10, 161–164]. However, the traditional approach of learning CNN-based models requires a large number of training images with instance-level pixel-wise annotations. Due to the high cost of collecting these supervised labels, researchers have looked at training these instance segmentation models using weak annotations, ranging from bounding boxes [67, 165] to image-level labels [166–171].

Many of the recent approaches for weakly supervised instance segmentation can be thought of as consisting of two components. First, a pseudo label generation model, which provides instance segmentations that are consistent with the weak annotations. Second, an instance segmentation model which is trained by treating the pseudo labels as ground-truth, and provides the desired output at test time.

Seen from the above viewpoint, the design of a weakly supervised instance segmentation approach boils down to three questions. First, how do we represent the instance segmentation model? Second, how do we represent the pseudo label generation model? And third, how do we learn the parameters of the two models using weakly supervised data? The answer to the first question is relatively clear: we should use a model that performs well when trained in a supervised manner, for example, Mask R-CNN [10]. However, we argue that the existing approaches fail to provide a satisfactory answer to the latter two questions.

Specifically, the current approaches do not take into account the inherent uncertainty in the pseudo label generation process [166, 169]. Consider, for instance, a training image that has been annotated to indicate the presence of a person. There can be several instance segmentations that are consistent with this annotation, and thus, one should not rely on a single pseudo label to train the instance segmentation model. Furthermore, none of the existing approaches provide a coherent learning objective for the two models. Often they suggest a simple two-step learning approach, that is, generate one set of pseudo labels followed by a one time training of the instance segmentation model [166]. While some works

consider an iterative training procedure [169], the lack of a learning objective makes it difficult to analyse and adapt them in varying settings.

In this work, we address the deficiencies of prior work by (i) proposing suitable representations for the two aforementioned components; and (ii) estimating their parameters using a principled learning objective. In more detail, we explicitly model the uncertainty in pseudo labels via a *conditional distribution*. The conditional distribution consists of three terms: (i) a semantic class aware unary term to predict the score of each segmentation proposal; (ii) a boundary aware pairwise term that encourages the segmentation proposal to completely cover the object; and (iii) an annotation consistent higher order term that enforces a global constraint on all segmentation proposals (for example, in the case of image-level labels, there exists at least one corresponding segmentation proposal for each class, or in the case of bounding boxes, there exists a segmentation proposal with sufficient overlap to each bounding box). All three terms combined enable the samples drawn from the conditional distribution to provide accurate annotation consistent instance segmentations. Furthermore, we represent the instance segmentation model as an annotation agnostic prediction distribution. This choice of representation allows us to define a joint probabilistic learning objective that minimizes the dissimilarity between the two distributions. The dissimilarity is measured using a task-specific loss function, thereby encouraging the models to produce high quality instance segmentations.

We test the efficacy of our approach on the Pascal VOC 2012 data set. We achieve 50.9% $\text{mAP}^r_{0.5}$, 28.5% $\text{mAP}^r_{0.75}$ for image-level annotations and 32.1% $\text{mAP}^r_{0.75}$ for bounding box annotations, resulting in an improvement of over 4% and 10% respectively over the state-of-the-art.

To summarize, we make the following contributions:

- We provide an efficient model for the complex non-factorizable, annotation consistent and boundary aware conditional distribution.
- We propose a joint probabilistic learning objective for training the conditional and the prediction distributions.
- Our overall framework is easily extendable to different weakly supervised labels such as image-level and bounding box annotations.
- Our approach provides state-of-the-art performance for the task of weakly supervised instance segmentation on the Pascal VOC 2012 data set.

6.2 Related Work

Due to the taxing task of acquiring the expensive per-pixel annotations, many weakly supervised methods have emerged that can leverage cheaper labels. For the task of semantic segmentation various types of weak annotations, such as image-level [68, 76, 172, 173], point [19], scribbles [74, 174], and bounding boxes [175, 176], have been utilized. However, for the instance segmentation, only image-level [166–171] and bounding box [67, 165] supervision have been explored. Our setup considers both

the image-level and the bounding box annotations as weak supervision. For the bounding box annotations, Hsu *et al.* [165] employs a bounding box tightness constraint and train their method by employing a multiple instance learning (MIL) based objective but they do not model the annotation consistency constraint for computational efficiency.

Most of the initial works [170, 171] on weakly supervised instance segmentation using image-level supervision were based on the class activation maps (CAM) [60, 62, 63, 177]. In their work, Zhou *et al.* [170] identify the heatmap as well as its peaks to represent the location of different objects. Although these methods are good at finding the spatial location of each object instance, they focus only on the most discriminative regions of the object and therefore, do not cover the entire object. Ge *et al* [168] uses the CAM output as the initial segmentation seed and refines it in a multi-task setting, which they train progressively. We use the output of [170] as the initial segmentation seed of our conditional distribution but the boundary aware pairwise term in our conditional distribution encourages pseudo labels to cover the entire object.

Most recent works on weakly supervised learning adopt a two-step process - generate pseudo labels and train a supervised model treating these pseudo labels as ground truth. Such an approach provides state-of-the-art results for various weakly supervised tasks like object detection [37, 77, 80], semantic segmentation [67, 175], and instance segmentation [166, 169]. Ahn *et al.* [166] synthesizes pseudo labels by learning the displacement fields and pairwise pixel affinities. These pseudo labels are then used to train a fully supervised Mask R-CNN [10], which is used at the test time. Laradji *et al.* [169] iteratively samples the pseudo segmentation label from MCG segmentation proposal set [178] and train a supervised Mask R-CNN [10]. This is similar in spirit to our approach of using the two distributions. However, they neither have a unified learning objective for the two distribution nor do they model the uncertainty in their pseudo label generation model. Regardless, the improvement in the results reported by these two methods advocates the importance of modeling two separate distributions. In our method, we explicitly model the two distributions and define a unified learning objective that minimizes the dissimilarity between them.

Our framework has been inspired by the work of Kumar *et al.* [51] who were the first to show the necessity of modeling uncertainty by employing two separate distributions in a latent variable model. This framework has been adopted for weakly supervised training of CNNs for learning human poses and object detection tasks [37, 136]. While their framework provides an elegant formulation for weakly supervised learning, its various components need to be carefully constructed for each task. Our work can be viewed as designing conditional and prediction distributions, as well as the corresponding inference algorithms, which are suited to instance segmentation.

6.3 Method

6.3.1 Notation

In line with the notation specified in Section 3.2.1, we denote an input image as $\mathbf{x} \in \mathbb{R}^{(H \times W \times 3)}$, where H and W are the height and the width of the image respectively. For each image, a set of segmentation proposals $\mathcal{R} = \{r_1, \ldots, r_P\}$ are extracted from a class-agnostic object proposal algorithm. In this work, we use Multiscale Combinatorial Grouping (MCG) [178] to obtain the object proposals. For the sake of simplicity, we only consider image-level annotations in our description. However, our framework can be easily extended to other annotations such as bounding boxes. Indeed, we will use bounding box annotations in our experiments. Given an image and the segmentation proposals, our goal is to classify each of the segmentation proposals to one of the $C + 1$ categories from the set $\{0, 1, \ldots, C\}$. Here category 0 is the background and categories $\{1, \ldots, C\}$ are object classes.

We denote the image-level annotations by $\mathbf{a} = \{0, 1\}^C$, where $\mathbf{a}^{(j)} = 1$ if image x contains the $j-$th object. Furthermore, we denote the unknown instance-level (segmentation proposal) label as $\mathbf{y} = \{0, \ldots, C\}^P$, where $\mathbf{y}^{(i)} = j$ if the $i-$th segmentation proposal is of the $j-$th category. A weakly supervised data set $\mathcal{W} = \{(\mathbf{x}_n, \mathbf{a}_n) \mid n = 1, \ldots, N\}$ contains N pairs of images $\mathbf{x}_n$ and their corresponding image-level annotations $\mathbf{a}_n$.

6.3.2 Conditional Distribution

Given the weakly supervised data set $\mathcal{W}$, we wish to generate pseudo instance-level labels $\mathbf{y}$ such that they are annotation consistent. Specifically, given the segmentation proposals $\mathcal{R}$ for an image $\mathbf{x}$, there must exists at least one segmentation proposal for each image-level annotation $\mathbf{a}^{(j)} = 1$. Since the annotations are image-level, there is inherent uncertainty in the figure-ground separation of the objects. We model this uncertainty by defining a distribution $\Pr_c(\mathbf{y} \mid \mathbf{x}, \mathbf{a}; \boldsymbol{\theta}_c)$ over the pseudo labels conditioned on the image-level weak annotations. Here, $\boldsymbol{\theta}_c$ are the parameters of the distribution. We call this a *conditional distribution*.

As highlighted in Section 3.2 , the conditional distribution itself is not explicitly represented. Instead, we use a neural network with parameters $\boldsymbol{\theta}_c$ which generates samples that can be used as pseudo labels. For the generated samples to be accurate, we wish that they have the following three properties: (i) they should have high fidelity with the scores assigned by the neural network for each region proposal belonging to each class; (ii) they should cover as large a portion of an object instance as possible; and (iii) they should be consistent with the annotation.

6.3.2.1 Modeling

In order for the conditional distribution to be annotation consistent, the instance-level labels $\mathbf{y}$ need to be compatible with the image-level annotation $\mathbf{a}$. This constraint cannot be trivially decomposed over each segmentation proposal. As a result, it would be prohibitively expensive to model the conditional

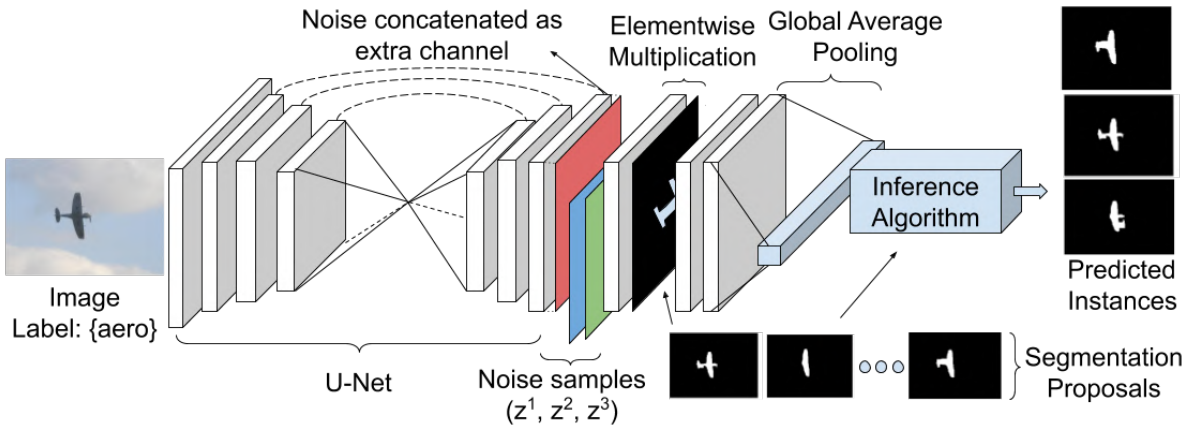


Figure 6.1 *The conditional net: a modified U-Net architecture is used to model the conditional net. For a single input image and three different noise samples* $\{z^1, z^2, z^3\}$ *(represented as red, green, and blue matrix) and a pool of segmentation proposals, three different instances are predicted for the given weak annotation (aeroplane in this example). Here the noise sample is concatenated as an extra channel to the final layer of the U-Net. The segmentation proposals are multiplied element-wise with the global feature to obtain the proposal specific feature. A global average pooling is applied to get class specific score. Finally, an inference algorithm generates the predicted samples.*

distribution directly as one would be required to compute its partition function. Taking inspiration from Arun *et al.* [37], we instead draw representative samples from the conditional distribution using the Discrete DISCO Nets [84]. We will now describe how we model the conditional distribution through a Discrete DISCO Nets, which we will now call a *conditional net*. Further discussion on Discrete DISCO Nets is present in Section 3.1.3.

Consider the modified fully convolutional U-Net [8] architecture shown in figure 6.1 for the conditional distribution. The parameters of the conditional distribution $\boldsymbol{\theta}_c$ are modeled by the weights of the conditional net. Similar to [179], noise sampled from a uniform distribution is added after the U-Net block (depicted by the colored filter). Each forward pass through the network takes the image $\mathbf{x}$ and noise sample $\mathbf{z}^k$ as input and produces a score function $F^k_{u,\mathbf{y}_u}(\boldsymbol{\theta}_c)$ for each segmentation proposal u and the corresponding putative label $\mathbf{y}_u$. We generate K different score functions using K different noise samples. These score functions are then used to sample the segmentation region proposals $\mathbf{y}^k_c$ such that they are annotation consistent. This enables us to efficiently generate the samples from the underlying distribution.

6.3.2.2 Inference

Given the input pair $(\mathbf{x}, \mathbf{z}^k)$ the conditional net outputs K score functions for each of the segmentation proposal $F^k_{u,\mathbf{y}_u}(\boldsymbol{\theta}_c)$. We redefine these score functions to obtain a final score function such that it is then used to sample the segmentation region proposals $\mathbf{y}^k_c$. The final score function has the following three properties.

1. The score of the sampled segmentation region proposal should be consistent with the score function. This *semantic class aware unary term* ensures that the final score captures the class specific features of each segmentation proposal. Formally, $G^k_{u,\mathbf{y}_u}(\mathbf{y}_c) = F^k_{u,\mathbf{y}_u}(\boldsymbol{\theta}_c)$.

2. The unary term alone is biased towards segmentation proposals that are highly discriminative. This results in selecting a segmentation proposal which does not cover the object in its entirety. We argue that all the neighboring segmentation proposals must have the same score discounted by the edge weights between them. We call this condition *boundary aware pairwise term*.

   In order to make the score function $G^k_{u,\mathbf{y}_u}(\mathbf{y}_c)$ pairwise term aware, we employ a simple but efficient iterative algorithm. The algorithm proceeds by iteratively updating the scores $G^k_{u,\mathbf{y}_u}(\mathbf{y}_c)$ by adding the contribution of their neighbors discounted by the edge weights between them until convergence. In practice, we fix the number of iteration to 3. Note that, it is possible to back-propagate through the iterative algorithm by simply unrolling its iterations, similar to a recurrent neural networks (RNN). Formally,

$$G^{k,n}_{u,\mathbf{y}_u}(\mathbf{y}_c) = G^{k,n-1}_{u,\mathbf{y}_u}(\mathbf{y}_c) + \frac{1}{H^{k,n-1}_{u,v}(\mathbf{y}_c) + \delta} \exp\left(-I_{u,v}\right). \tag{6.1}$$

   Here, n denotes the iteration step for the iterative algorithm and δ is a small positive constant added for numerical stability. In our experiments, we set $\delta = 0.1$. The term $H^{k,n-1}_{u,v}(\mathbf{y}_c)$ is the difference between the scores of the neighboring segmentation proposal. It helps encourage same label for the neighboring segmentation proposals that are not separated by the edge pixels. It is given as,

$$H^{k,n-1}_{u,v}(\mathbf{y}_c) = \sum_{u,v \in \mathcal{N}_u} \left(G^{k,n-1}_{u,\mathbf{y}_u}(\mathbf{y}_c) - G^{k,n-1}_{v,\mathbf{y}_u}(\mathbf{y}_c) \right)^2. \tag{6.2}$$

   The term $I_{u,v}$ is the sum of the edge pixel values between the two neighboring segmentation regions. Note that the pairwise term is a decay function weighted by the edge pixel values. This ensures a high contribution to the pairwise term is only from the pair of segmentation proposals that does not share an edge.

3. In order to ensure that at there must exist at least one segmentation proposal for every image-level annotation,a higher order penalty is added to the score. We call this *annotation consistent higher order term*. Formally,

$$S^k(\mathbf{y}_c) = \sum_{u=1}^{P} G^{k,n}_{u,\mathbf{y}_u}(\mathbf{y}_c) + Q^k(\mathbf{y}_c). \tag{6.3}$$

   Here,

$$Q^k(\mathbf{y}_c) = \begin{cases} 0 & \text{if } \forall j \in \{1, \ldots, C\} \text{ s.t. } \mathbf{a}^{(j)} = 1, \\ & \exists i \in \mathcal{R} \text{ s.t. } \mathbf{y}^{(i)} = j, \\ -\infty & \text{otherwise.} \end{cases} \tag{6.4}$$

Given the scoring function in equation (6.3), we compute the $k-$th sample of the conditional net as,

$$\mathbf{y}_c^k = \arg\max_{\mathbf{y} \in \mathcal{Y}} S^k(\mathbf{y}_c). \tag{6.5}$$

Observe that in equation (6.5), the $\arg\max$ is computed over the entire output space $\mathcal{Y}$. A naïve brute force algorithm is therefore not feasible. We design an efficient greedy algorithm that selects the highest scoring non-overlapping proposal. The inference algorithm is described in Algorithm 7.

Algorithm 7 Inference Algorithm for the Conditional Net

Input: Region masks R, Image-level labels a
Output: Predicted instance-level instances $\mathbf{y}_c^k$

Initialize: $G^k_{u,\mathbf{y}_u}(\mathbf{y}_c) \leftarrow F^k_{u,\mathbf{y}_u}(\boldsymbol{\theta}_c)$ ▷ Iterative Algorithm
repeat$G^{k,n}_{u,\mathbf{y}_u}(\mathbf{y}_c)$ has converged
  for all $v \in \mathcal{N}_u$ **do**
    $H^{k,n-1}_{u,v}(\mathbf{y}_c) \leftarrow \sum_{u,v \in \mathcal{N}_u} (G^{k,n-1}_{u,\mathbf{y}_u}(\mathbf{y}_c) - G^{k,n-1}_{v,\mathbf{y}_v}(\mathbf{y}_c))^2$
    $G^{k,n}_{u,\mathbf{y}_u}(\mathbf{y}_c) \leftarrow G^{k,n-1}_{u,\mathbf{y}_u}(\mathbf{y}_c) + \frac{1}{H^{k,n-1}_{u,v}(\mathbf{y}_c)+\delta} \exp(-I_{u,v})$
  end for
until $G^{k,n}_{u,\mathbf{y}_u}(\mathbf{y}_c)$ has converged
Greedily select highest scoring non-overlapping proposal:
$Y \leftarrow \emptyset$
for all $j \leftarrow \{1, \ldots, C\}$ **and** $\mathbf{a}^{(j)} = 1$ **do**
  $Y_j \leftarrow \emptyset$
  $R_j \leftarrow \text{sort}(G^{k,n}_{u,\mathbf{y}_u}(\mathbf{y}_c))$
  for $i \leftarrow 1$ to P **do**
    $Y_j \leftarrow Y_j \cup \{r_i\}$
    $R_j \leftarrow R_j \setminus \{r_i\}$
    for all $l \in R_j$ **and** $\frac{r_i \cap r_l}{r_l} > t$ **do**
      $R_j \leftarrow R_j \setminus \{r_l\}$
    end for
  end for
  $Y \leftarrow Y \cup Y_j$
end for
return $\mathbf{y}_c^k \leftarrow Y$

6.3.3 Prediction Distribution

The task of the supervised instance segmentation model is to predict the instancemask given an image. We employ Mask R-CNN [18] for this task. As predictions for each of the regions in the Mask R-CNN is computed independently, we can view the output of the Mask R-CNN as the following fully

factorized distribution,

$$\Pr_p(\mathbf{y} \mid \mathbf{x}; \boldsymbol{\theta}_p) = \prod_{i=1}^{R} \Pr(\mathbf{y}_i \mid \mathbf{r}_i, \mathbf{x}_i; \boldsymbol{\theta}_p). \tag{6.6}$$

Here, R are the set of bounding box regions proposed by the region proposal network and $\mathbf{r}_i$ are its corresponding region features. The term $\mathbf{y}_i$ is the corresponding prediction for each of the bounding box proposals. We call the above distribution a *prediction distribution* and the Mask R-CNN a *prediction network*.

6.4 Learning Objective

In this section, we present the learning objective for instance segmentation in a weakly supervised setting. The goal is to learn the parameters of the prediction and conditional distributions, $\boldsymbol{\theta}_p$ and $\boldsymbol{\theta}_c$, respectively. Both distributions aim to predict instance segmentation masks, but the conditional distribution benefits from additional image-level annotations, enabling it to produce more accurate predictions. By leveraging the task similarity between the two distributions, we align them to facilitate the transfer of knowledge from the conditional distribution to the prediction distribution.

The joint learning objective, introduced in the previous chapter (Section 3.2.3), minimizes the dissimilarity coefficient [82] between the prediction and conditional distributions. We build upon this objective to address the task-specific requirements of instance segmentation.

6.4.1 Task-Specific Loss Function

The dissimilarity coefficient requires a task-specific loss function, Δ, to compute the alignment between the two distributions. For instance segmentation, we adopt the multi-task loss defined by Mask R-CNN [10]:

$$\Delta(\mathbf{y}_1, \mathbf{y}_2) = \Delta_{\text{cls}}(\mathbf{y}_1, \mathbf{y}_2) + \Delta_{\text{box}}(\mathbf{y}_1, \mathbf{y}_2) + \Delta_{\text{mask}}(\mathbf{y}_1, \mathbf{y}_2). \tag{6.7}$$

Here, Δ_{cls} is the classification loss (log loss), Δ_{box} is the bounding box regression loss (smooth-L1 loss), and Δ_{mask} is the segmentation loss (pixel-wise cross-entropy).

For the conditional network, which outputs only segmentation regions $\mathbf{y}$, Δ_{box} is inactive, leaving only Δ_{cls} and Δ_{mask} active. In contrast, all three components are active for the prediction network. To handle bounding box information, a pseudo bounding box is constructed around the segmentation label, which serves as a bounding box label for Mask R-CNN.

6.4.2 Learning Objective for Instance Segmentation

Using the task-specific loss Δ, the learning objective for instance segmentation extends the formulation introduced in the previous chapter. It minimizes the dissimilarity coefficient between the prediction and conditional distributions:

$$\boldsymbol{\theta}_p^*, \boldsymbol{\theta}_c^* = \arg\min_{\boldsymbol{\theta}_p, \boldsymbol{\theta}_c} DISC_{\Delta}\left(\Pr_p(\boldsymbol{\theta}_p), \Pr_c(\boldsymbol{\theta}_c)\right). \tag{6.8}$$

As discussed in Section 6.3.2, modeling the conditional distribution directly is challenging. Instead, we approximate the diversity terms by drawing K samples, $\mathbf{y}_c^k$, from the conditional network. In accordance with the equations (3.10, 3.15, 3.17) The diversity terms are computed as:

$$DIV_\Delta(\Pr_p, \Pr_c) = \frac{1}{K} \sum_{k=1}^{K} \sum_{\mathbf{y}_p^{(i)}} \Pr_p(\mathbf{y}_p^{(i)}; \boldsymbol{\theta}_p) \Delta(\mathbf{y}_p^{(i)}, \mathbf{y}_c^k), \tag{6.9a}$$

$$DIV_\Delta(\Pr_c, \Pr_c) = \frac{1}{K(K-1)} \sum_{\substack{k,k'=1 \\ k' \neq k}}^{K} \Delta(\mathbf{y}_c^k, \mathbf{y}_c^{k'}), \tag{6.9b}$$

$$DIV_\Delta(\Pr_p, \Pr_p) = \sum_{\mathbf{y}_p^{(i)}} \sum_{\mathbf{y'}_p^{(i)}} \Pr_p(\mathbf{y}_p^{(i)}; \boldsymbol{\theta}_p) \Pr_p(\mathbf{y'}_p^{(i)}; \boldsymbol{\theta}_p) \Delta(\mathbf{y}_p^{(i)}, \mathbf{y'}_p^{(i)}). \tag{6.9c}$$

Here, $DIV_\Delta(\Pr_p, \Pr_c)$ measures the cross-diversity between the prediction and conditional distributions, representing the expected loss between samples from the two distributions. While $\Pr_p$ is fully factorized, enabling direct computation of expectations, $\Pr_c$ is approximated using K samples. This approach ensures efficient computation while preserving the alignment between distributions.

6.5 Optimization

As highligted in Section 3.2.4, the parameters of the two distribution, θ_p and θ_c are modeled by a neural network, it is ideally suited to be minimized by stochastic gradient descent. We employ a coordinate descent strategy to optimize the two sets of parameters. The algorithm proceeds by iteratively fixing the prediction network and training the conditional network, followed by learning the prediction network for a fixed conditional network.

The iterative learning strategy results in a fully supervised training of each network by using the output of the other network as the pseudo label. This allows us to readily use the algorithms developed in Mask R-CNN [10] and Discrete DISCO Nets [84]. Note that, as the conditional network obtains samples over the $\arg\max$ operator in equation (6.5), the objective (6.8) for the conditional network is non-differentiable. However, the scoring function $S^k(\mathbf{y}_c)$ in equation (6.3) itself is differentiable. This allows us to use the direct loss minimization strategy [180, 181] developed for computing estimated gradients over the $\arg\max$ operator [84, 182].The details of the algorithm is discussed in Section 3.1.3 and algorithm 4.

6.5.1 Visualization of the learning process

Figure 6.2 provides the visualization of the output of the two networks for the first and the final iterations of the training process. The first three columns are the three output samples of the conditional distribution. Note that in our experiments, we output 10 samples corresponding to 10 different noise

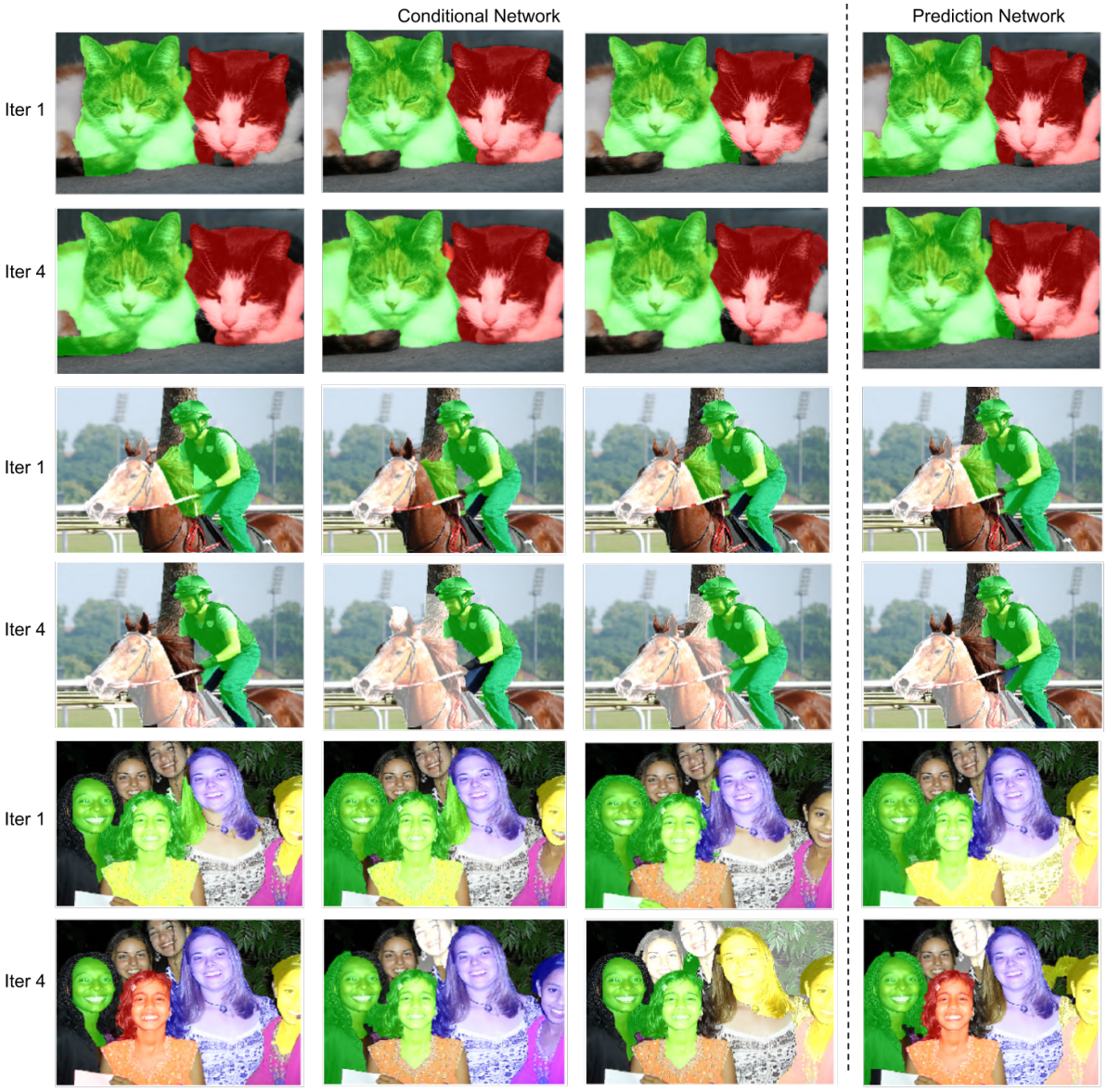


Figure 6.2 *Examples of the predictions from the conditional and prediction networks for three different cases of varying difficulty. Columns 1 through 3 are different samples from the conditional network. For each case, its first row shows the output of the two networks after the first iteration and its second row represents the output of the two networks after the fourth (final) iteration. Each instance of an object is represented by different mask color. Best viewed in color.*

samples. The fourth column shows the output of the prediction distribution. The output for the prediction network is selected by employing a non-maximal suppression (NMS) with its score threshold kept at 0.7, as is the default setting in [10]. The first row represents the output of the two networks after the first iteration and the second row shows their output after the fourth (final) iteration.

The first case demonstrates an easy example where two cats are present in the image. Initially, the conditional distribution samples the segmentation proposals which do not cover the entire body of the cat but still manages to capture the boundaries reasonably well. However, due to the variations in these samples, the prediction distribution learns to better predict the extent of the cat pixels. This, in turn, encourages the conditional network to generate a better set of samples. Indeed, by the fourth iteration, we see an improvement in the quality of samples by the conditional network and they now cover the entire body of the cat, thereby improving the performance. As a result, we can see that finally the prediction network successfully learns to segment the two cats in the image.

The second case presents a challenging scenario where a person is riding a horse. In this case, the person is occluding the front and the rear parts of the horse. Initially, we see that the conditional network only provides samples for the most discriminative region of the horse - its face. The samples generated for the person class, though not accurate, covers the entire person. We observe that over the subsequent iterations, we get an accurate output for the person class. The output for the horse class also expands to cover its front part completely. However, since its front and the rear parts are completely separated, the final segmentation could not cover the rear part of the horse.

The third case presents another challenging scenario where there are multiple people present. Four people standing in front and two are standing at the back. Here, we observe that initially, the conditional network fails to distinguish between the two people standing in the front-left of the image and fails to detect persons standing at the back. The samples for the third and the fourth persons standing in front-center and front-right respectively are also not accurate. Over the iterations, the conditional network improves its predictions for the four people standing in front and also sometimes detect the people standing at the back. As a result, prediction network finally detects five of the six people in the image.

6.6 Experiments

6.6.1 Data set and Evaluation Metric

6.6.1.1 Data Set

We evaluate our proposed method on Pascal VOC 2012 segmentation benchmark [13]. The data set consists of 20 foreground classes. Following previous works [67, 165, 166, 168], we use the augmented Pascal VOC 2012 data set [183], which contains $10,582$ training images.

From the augmented Pascal VOC 2012 data set, we construct two different weakly supervised data sets. The first data set is where we retain only the image-level annotations. For the second data set, we retain the bounding box information along with the image-level label. In both the data sets, the pixel-level labels are discarded.

6.6.1.2 Evaluation Metric

We adopt the standard evaluation metric for instance segmentation, mean average precision (mAP) [184]. Following the same evaluation protocol from other competing approaches, we report mAP with four intersection over union (IoU) thresholds, denoted by mAP_k^r where k denotes the different values of IoU and $k = \{0.25, 0.50, 0.70, 0.75\}$.

6.6.2 Initialization

We now discuss various strategies to initialize our conditional network for different levels of weakly supervised annotations.

6.6.2.1 Image Level Annotations

Following the previous works on weakly supervised instance segmentation from image-level annotations [166, 169, 171], we use the Class Activation Maps (CAMs) to generate the segmentation seeds for each image in the training set. Specifically, like [166, 169, 171], we rely on the Peak Response Maps (PRM) [170] to generate segmentation seeds that identify the salient parts of the objects. We utilize these seeds as pseudo segmentation labels to initially train our conditional network. We also filter the MCG [178] segmentation proposal such that each selected proposal has at least a single pixel overlap with the PRM segmentation seeds. This helps us reduce the number of segmentation proposals needed thereby reducing the memory requirement. Once the initial training for the conditional network is over, we proceed with the iterative optimization strategy, described in section 6.5.

6.6.2.2 Bounding Box Annotations

For the weakly supervised data set where bounding box annotations are present, we filter the MCG [178] segmentation proposals such that only those who have a high overlap with the ground-truth bounding boxes are retained. The PRM [170] segmentation seeds are also pruned such that they are contained within each of the bounding box annotations.

6.6.3 Implementation Details

We use the standard Mask R-CNN as the prediction network and adapt the U-Net architecture for the conditional network, as shown in figure 6.1. For a fair comparison, the prediction network, we use ImageNet pre-trained ResNet-50 architecture for experiments with image-level annotation and a pretrained ResNet-101 architecture for the bounding box annotations.

Similar to [179], the U-Net architecture is modified by adding a noise sample as an extra channel after the deconvolutional layers as shown in figure 6.1. A 1×1 convolution is applied to bring the number of channels back to the original dimensions (32 channels). The segmentation region proposal masks taken from MCG [178] is then multiplied element-wise with the features from all the channels. This allows us

to extract features only from the segmentation proposal. A 1×1 convolution is applied again to make the number of channels equal to the number of classes. This is followed by a global average pooling layer which gives us, for each of the segmentation proposals, a vector of dimensions equal to the number of classes. This vector for each of the segmentation proposal is passed to the inference algorithm which in turn provides the output segmentation masks corresponding to the image-level annotations. For all our experiments we choose K=10 for the conditional network and use the Adam optimizer. For all the other hyper-parameters we use the same configuration as described in [179]. For the prediction network, we use default hyper-parameters described in [10].

We also study the effect of an alternative architecture for the conditional network. In what follows, we provide the details of this ResNet based conditional network.

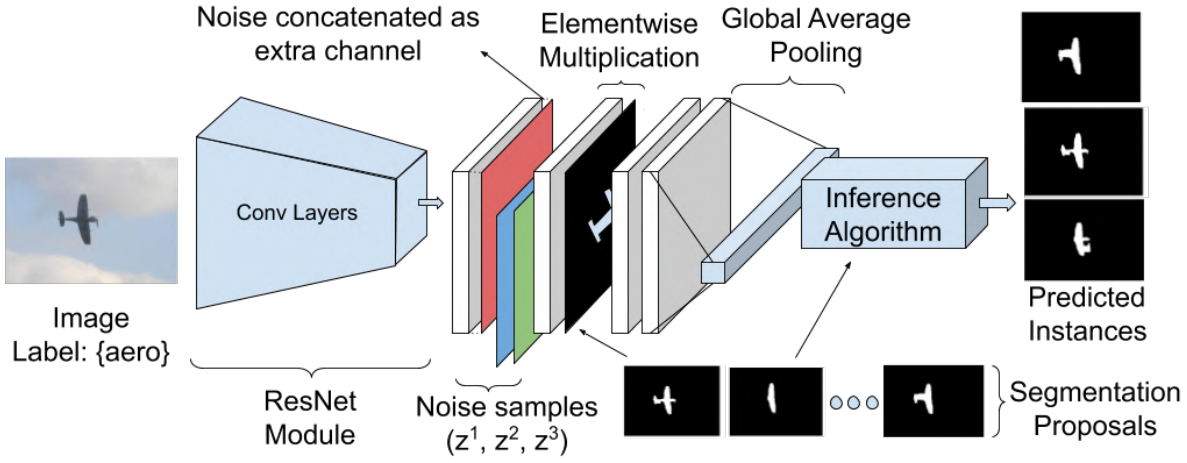


Figure 6.3 *ResNet based conditional network*

The architecture for the ResNet based conditional network is shown in figure 6.3. The image is first passed through the ResNet module to obtain low-resolution high-level features. For the experiments where we use only the image-level annotations, a ResNet-50 module is employed and where we use the bounding-box level annotations, a ResNet-101 module is used. A noise filter is appended as an extra channel followed by a 1×1 convolutional filter, which brings the number of channels back to the original dimensions. The segmentation proposal masks are then multiplied element-wise to obtain segmentation proposal specific features. Next, a 1×1 convolutional is applied to make the number of channels equal to the number of classes. Finally, a global average pooling is applied to obtain a vector whose dimensions is equal to the number of classes in the data set. This vector is then passed through the inference algorithm to obtain the final predicted samples. The results obtained using this ResNet based conditional network architecture are called as Ours (ResNet-50) and Ours (ResNet-101).

Note that, the U-Net based conditional network provides a higher resolution image features as compared to its ResNet based counterparts. These are then used to obtain the individual features of the segmentation mask proposals. The higher resolution features thus provide richer per-mask features. These are especially useful for smaller objects and cluttered environment where context resolution is

important. The superior results of our method when using a U-Net based conditional network empirically verify this claim.

6.6.4 Results

6.6.4.1 Comparison with other methods

Table 6.1 *Evaluation of instance segmentation results from different methods with varying level of supervision on Pascal VOC 2012* val *set. The terms $\mathcal{F}$, $\mathcal{B}$, and $\mathcal{I}$ denotes a fully supervised approach, methods that uses the bounding box labels, and methods that uses the image-level labels respectively. Our prediction network results when using a ResNet based conditional network is presented as 'Ours (ResNet-*)' and the results of the prediction network using a U-Net based conditional network is presented as 'Ours'.*

Method	Supervision	Backbone	$\text{mAP}^r_{0.25}$	$\text{mAP}^r_{0.50}$	$\text{mAP}^r_{0.70}$	$\text{mAP}^r_{0.75}$
Mask R-CNN [10]	$\mathcal{F}$	R-101	76.7	67.9	52.5	44.9
PRN [170]	$\mathcal{I}$	R-50	44.3	26.8	-	9.0
IAM [171]	$\mathcal{I}$	R-50	45.9	28.8	-	11.9
OCIS [167]	$\mathcal{I}$	R-50	48.5	30.2	-	14.4
Label-PEnet [168]	$\mathcal{I}$	R-50	49.1	30.2	-	12.9
WISE [169]	$\mathcal{I}$	R-50	49.2	41.7	-	23.7
IRN [166]	$\mathcal{I}$	R-50	-	46.7	-	23.5
Ours (ResNet-50)	$\mathcal{I}$	R-50	59.1	49.7	29.2	27.1
Ours	$\mathcal{I}$	R-50	**59.7**	**50.9**	**30.2**	**28.5**
SDI [67]	$\mathcal{B}$	R-101	-	44.8	-	17.8
BBTP [165]	$\mathcal{B}$	R-101	**75.0**	**58.9**	30.4	21.6
Ours (ResNet-101)	$\mathcal{B}$	R-101	73.1	57.7	33.5	31.2
Ours	$\mathcal{B}$	R-101	73.8	58.2	**34.3**	**32.1**

We compare our proposed method with other state-of-the-art weakly supervised instance segmentation methods. The mean average precision (mAP) over different IoU thresholds are shown in table 6.1. Compared with the other methods, our proposed framework achieves state-of-the-art performance for both image-level and the bounding box labels. We also study the effect of using a different conditional network architecture based on ResNet-50 and ResNet-101. This is shown in the table as 'Ours (ResNet-50)' and 'Ours (ResNet 101)' respectively. Our main result employs a U-Net based architecture for the conditional network and is presented by 'Ours' in the table. The implementation details and the details of the alternative architecture are presented in the supplementary. The encoder-decoder architecture of the U-Net allows us to learn better features. As a result, we observe that our method which adopts U-Net architecture for the conditional network consistently outperforms the one which adopts a ResNet based architecture. In table 6.1, observe that our approach performs particularly well for the higher IoU thresholds ($\text{mAP}^r_{0.70}$ and $\text{mAP}^r_{0.75}$) for both the image-level and the bounding-box labels. This demonstrates that our model can predict the instance segments most accurately by respecting the

object boundaries. The per-class quantitative and qualitative results for our method is presented in the supplementary material.

6.6.4.2 Class-specific results

We present the per-class result for our method on the Pascal VOC 2012 data set in table 6.2. The first two rows correspond to the result where our method was trained only using the image-level annotations. The last two rows correspond to the results where our methods were trained using the bounding box annotations.The qualitative results for each class is presented in figure 6.4.

Figure 6.4 *Qualitative results of our proposed approach on VOC 2012 validation set.*

Table 6.2 *Per class result for $mAP^r_{0.5}$ metric on Pascal VOC 2012 data set for methods that are trained on using image-level supervision $\mathcal{I}$ and bounding box annotations $\mathcal{B}$*

Method	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	pson	plant	sheep	sofa	train	tv	mAP
Ours (ResNet-50) $\mathcal{I}$	74.2	52.6	68.6	44.1	25.0	63.4	35.9	72.6	18.2	47.1	24.6	63.5	53.7	67.3	40.9	29.4	42.8	39.6	69.5	61.2	49.7
Ours $\mathcal{I}$	75.5	53.6	69.9	45.3	26.7	64.3	37.4	73.7	19.3	48.7	25.3	64.6	55.0	68.3	42.1	30.8	44.2	40.5	70.6	62.2	50.9
Ours (ResNet-101) $\mathcal{B}$	77.9	62.6	73.8	49.0	35.9	72.6	45.8	78.4	29.7	55.7	31.9	70.6	61.3	73.6	49.2	39.9	50.8	47.9	76.5	69.6	57.7
Ours $\mathcal{B}$	79.1	63.9	75.1	49.3	36.5	73.1	46.4	78.8	30.1	56.4	32.1	71.3	61.6	74.8	49.5	40.2	51.1	48.3	77.2	69.9	58.2

6.6.5 Ablation Experiments

Table 6.3 *Evaluation of the instance segmentation results for the various ablative settings of the conditional distribution on Pascal VOC 2012 data set.*

$mAP^r_{0.25}$			$mAP^r_{0.50}$			$mAP^r_{0.75}$		
U	U+P	U+P+H	U	U+P	U+P+H	U	U+P	U+P+H
57.9	59.1	59.7	47.6	49.9	50.9	23.1	26.9	28.5

6.6.5.1 Effect of the unary, the pairwise and the higher order terms

We study the effect of the conditional distributions unary, pairwise and the higher order terms have on the final output in table 6.3. We use the terms U, U+P, and U+P+H to denote the settings where only the unary term is present, both the unary and the pairwise terms are present and all three terms are present in the conditional distribution. We see that unary term alone performs poorly across the different IoU thresholds. We argue that this is because of the bias of the unary term for segmenting only the most discriminative regions. The pairwise term helps allay this problem and we observe a significant improvement in the results. This is specially noticeable for higher IoU thresholds that require more accurate segmentation. The higher order term helps in improving the accuracy by ensuring that correct samples are generated by the conditional distribution.

6.6.5.2 Effect of the probabilistic learning objective

Table 6.4 *Evaluation of the instance segmentation results for the various ablative settings of the loss function's diversity coefficient terms on Pascal VOC 2012 data set.*

Method mAP^r_k	$\Pr_p, \Pr_c$ (proposed)	$PW_p, \Pr_c$	$\Pr_p, PW_c$	PW_p, PW_c
$mAP^r_{0.25}$	59.7	59.5	57.3	57.2
$mAP^r_{0.50}$	50.9	50.3	46.9	46.6
$mAP^r_{0.75}$	28.5	27.7	23.4	23.0

To understand the effect of explicitly modeling the two distributions ($\Pr_p$ and $\Pr_c$), we compare our approach with their corresponding pointwise network. In order to sample a single output from our

conditional network, we remove the self-diversity coefficient term and feed a zero noise vector (denoted by PW_c). For a pointwise prediction network, we remove its self-diversity coefficient. The prediction network still outputs the probability of each proposal belonging to a class. However, by removing the self-diversity coefficient term, we encourage it to output a peakier distribution (denoted by PW_p). Table 6.4 shows that both the diversity coefficient term is important for maximum accuracy. We also note that modeling uncertainty over the pseudo label generation model by including the self-diversity in the conditional network is relatively more important. The self-diversity coefficient in the conditional network enforces it to sample a diverse set of outputs which helps in dealing with the difficult cases and in avoiding overfitting during training.

6.7 Conclusion

We present a novel framework for weakly supervised instance segmentation. Our framework efficiently models the complex non-factorizable, annotation consistent and boundary aware conditional distribution that allows us to generate accurate pseudo segmentation labels. Furthermore, our framework provides a joint probabilistic learning objective for training the prediction and the conditional distributions and can be easily extendable to different weakly supervised labels such as image-level and bounding box annotations. Extensive experiments on the benchmark Pascal VOC 2012 data set has shown that our probabilistic framework successfully transfers the information present in the image-level annotations for the task of instance segmentation achieving state-of-the-art result for both image-level and bounding box annotations.

Chapter 7

Conclusion and Future Work

We conclude the thesis by highlighting the key contributions and discussing the future directions.

7.1 Summary

This thesis tackles the challenge of training deep neural networks using weak annotations for visual scene recognition tasks. Weak annotations, being less detailed and more ambiguous than fine-grained, task-specific labels, pose significant obstacles to achieving accurate predictions. To overcome these challenges, this research proposes a novel probabilistic framework that seamlessly transforms coarse annotations into the fine-grained outputs essential for accurate and high-quality scene understanding.

7.1.1 Key Contributions

In the following sections, we will outline the key contributions of this thesis.

7.1.1.1 Probabilistic Framework

The primary contribution of this thesis is the development of a *probabilistic framework* based on the *dissimilarity coefficient* objective. This framework aligns two distinct distributions: the conditional distribution, which leverages weak annotations to generate task-specific predictions, and the prediction distribution, which produces test-time outputs independent of the weak annotations. By explicitly modeling uncertainty inherent in learning from imprecise labels, the framework effectively addresses challenges posed by weak annotations. The dissimilarity coefficient objective ensures efficient knowledge transfer, enabling the prediction network to deliver fine-grained outputs with high accuracy and robustness.

This framework offers a principled and generalized approach to weakly supervised learning (WSL), adaptable to a wide range of visual scene recognition tasks. By explicitly modeling uncertainty and aligning distributions using the dissimilarity coefficient loss, the framework overcomes limitations of existing methods, which often depend on a single distribution for conflicting tasks or lack mechanisms to handle uncertainty effectively.

The framework employs deep generative models, such as DISCO Nets and Discrete DISCO Nets, to efficiently sample from complex conditional distributions. Additionally, it integrates state-of-the-art architectures like Hourglass Networks, Fast R-CNN, and Mask R-CNN to manage prediction distributions. This design enables flexibility in handling various types of weak annotations, including image-level, point-level, and bounding box labels, while preserving the strengths of leading supervised models during inference.

To train the framework, the thesis introduces both iterative and joint optimization strategies, striking a balance between computational efficiency and convergence quality. These strategies ensure scalability to large datasets and complex tasks.

The framework is validated across three challenging tasks—human pose estimation, object detection, and instance segmentation—highlighting its versatility and effectiveness. For each task, the framework incorporates task-specific priors and cues, such as activation maps, spatial constraints, and higher-order terms, to achieve state-of-the-art performance.

7.1.1.2 Visual Scene Recognition Tasks

The effectiveness of the proposed framework is demonstrated through extensive experiments on benchmark datasets across multiple tasks.

For human pose estimation, the framework aligns pose predictions with coarse action labels, achieving substantial improvements on the MPII and JHMDB datasets. By utilizing DISCO Nets to model pose uncertainty and incorporate structural priors, the framework establishes a new benchmark for weakly supervised pose estimation.

In object detection, the framework adeptly handles a variety of coarse annotations, including image-level labels, count annotations, point annotations, and scribble annotations. By integrating class activation maps, spatial regularization, and higher-order constraints, it outperforms existing methods on the PASCAL VOC 2007, 2012, and MS COCO 2014, 2017 datasets. The efficient sampling algorithm introduced for discrete generative models ensures precise modeling of complex conditional distributions, directly enhancing accuracy. Additionally, the application of curriculum learning further boosts performance.

For instance segmentation, the framework combines complex conditional distributions with unary, pairwise, and higher-order terms. By leveraging weak annotations such as image-level and bounding box labels, it achieves state-of-the-art performance on the PASCAL VOC 2012 dataset. These results highlight the critical role of explicitly modeling uncertainty in pseudo-label generation for fine-grained tasks.

7.1.2 Significance of the Work

This thesis establishes a foundation for weakly supervised learning in complex visual scene understanding tasks. The proposed probabilistic framework addresses a significant gap in the literature by providing a principled and unified approach to handle weak annotations while maintaining scalability,

efficiency, and robustness. By explicitly modeling uncertainty, the framework ensures reliable transfer of knowledge from coarse annotations to fine-grained predictions, a critical requirement for real-world applications where fully supervised data is often unavailable.

The flexibility of the framework makes it adaptable to a wide range of vision tasks beyond those explored in this thesis. The integration of diverse priors, hints and constraints demonstrates its ability to handle task-specific challenges, paving the way for further research in areas such as weakly supervised video analysis, 3D scene understanding, and medical image analysis.

7.2 Future Directions

While this thesis presents significant advancements in weakly supervised learning (WSL), it also opens up numerous opportunities for further exploration and innovation. The proposed framework and methodologies provide a strong foundation for addressing the challenges of weak annotations, but there remain several avenues for future research to enhance and extend the capabilities of WSL across diverse tasks and domains.

One promising direction is the extension of the framework to video data. Unlike static images, videos present additional challenges such as temporal coherence, motion dynamics, and the need to model long-term dependencies across frames. Incorporating temporal information into the framework could enable weakly supervised approaches to tackle tasks like action recognition, object tracking, and video segmentation more effectively. Addressing these challenges would require integrating spatiotemporal priors and exploring novel architectures capable of handling sequential data.

Another area of focus is improving scalability to handle extremely large datasets. While the current framework demonstrates scalability to standard benchmarks, real-world datasets in domains such as autonomous driving and social media often contain millions of samples. An exciting direction here is the potential application of the framework to train foundational models using weak supervision. Foundational models often rely on costly supervised datasets for alignment after their pretraining stage, creating a significant bottleneck. By leveraging weak supervision, the proposed framework can facilitate the alignment process, reducing reliance on expensive, fine-grained annotations while maintaining performance. Incorporating efficient uncertainty quantification into this process can further enhance the reliability of foundational models in large-scale weakly annotated settings.

The applicability of the framework to other domains represents another exciting direction. For instance, in medical imaging, weak annotations such as image-level labels or approximate bounding boxes are commonly available due to the cost and expertise required for fine-grained annotations. Adapting the framework to tasks like disease detection, organ segmentation, and anomaly localization could significantly impact healthcare by reducing dependency on detailed labels. Similarly, in autonomous driving, weak annotations from large-scale driving datasets could be utilized to train models for object detection, scene segmentation, and trajectory prediction, improving safety and efficiency. Efficient un-

certainty quantification would play a critical role here, enabling models to provide reliable predictions in safety-critical applications by highlighting uncertain outputs for further review or refinement.

Emerging paradigms such as open-set weakly supervised learning present another intriguing avenue for research. Unlike traditional WSL, which assumes that all classes in the test set are known during training, open-set WSL aims to handle unseen or novel classes during inference. This requires designing frameworks that can generalize to unseen categories while effectively leveraging weak annotations. Incorporating techniques such as zero-shot learning, domain adaptation, and uncertainty-aware models could be pivotal in addressing these challenges.

Long-tailed weakly supervised learning is yet another important research area. Real-world datasets often exhibit a long-tailed distribution where a few classes dominate, and many others are underrepresented. Adapting the framework to handle such imbalances could involve designing loss functions and sampling strategies that account for class distribution, ensuring robust performance across both frequent and rare classes. Additionally, efficient uncertainty quantification could help focus learning on underrepresented classes by identifying regions of high uncertainty and guiding targeted data augmentation or annotation efforts.

Lastly, integrating uncertainty estimation into more complex tasks and leveraging it for active learning could further enhance the framework's utility. By quantifying uncertainty efficiently, the framework could prioritize uncertain samples for expert labeling, thereby improving overall performance with minimal annotation effort. Exploring active learning strategies in combination with the proposed probabilistic framework and uncertainty quantification techniques could open up new possibilities for efficient annotation pipelines and enhanced model performance.

In conclusion, while this thesis marks a significant step forward in weakly supervised learning, it lays the groundwork for numerous future directions. By extending the framework to video data, scaling to larger datasets, exploring new domains, and addressing emerging challenges such as open-set and long-tailed WSL, along with advancing efficient uncertainty quantification techniques, the research community can continue to push the boundaries of what is achievable with limited annotations.

Bibliography

[1] R. Girshick, "Fast r-cnn," in *ICCV*, 2015.

[2] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *CVPR*, 2014.

[3] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *NIPS*, 2015.

[4] A. Newell, K. Yang, and J. Deng, "Stacked hourglass networks for human pose estimation," in *ECCV*, 2016.

[5] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh, "Convolutional pose machines," in *CVPR*, 2016.

[6] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *T-PAMI*, 2017.

[7] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *CVPR*, 2015.

[8] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *MICCAI*, 2015.

[9] D. Bolya, C. Zhou, F. Xiao, and Y. J. Lee, "Yolact: Real-time instance segmentation," in *ICCV*, 2019.

[10] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *ICCV*, 2017.

[11] C. Sun, A. Shrivastava, S. Singh, and A. Gupta, "Revisiting unreasonable effectiveness of data in deep learning era," in *ICCV*, 2017.

[12] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *CVPR*, 2009.

[13] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *IJCV*, 2010.

[14] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *ECCV*, 2014.

[15] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *CVPR*, 2016.

[16] A. Kuznetsova, H. Rom, N. Alldrin, J. Uijlings, I. Krasin, J. Pont-Tuset, S. Kamali, S. Popov, M. Malloci, A. Kolesnikov *et al.*, "The open images dataset v4," *IJCV*, 2020.

[17] B. Zhou, H. Zhao, X. Puig, T. Xiao, S. Fidler, A. Barriuso, and A. Torralba, "Semantic understanding of scenes through the ade20k dataset," *IJCV*, 2019.

[18] M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele, "2d human pose estimation: New benchmark and state of the art analysis," in *CVPR*, 2014.

[19] A. Bearman, O. Russakovsky, V. Ferrari, and L. Fei-Fei, "What's the point: Semantic segmentation with point supervision," in *ECCV*, 2016.

[20] H. Caesar, J. Uijlings, and V. Ferrari, "Coco-stuff: Thing and stuff classes in context," in *CVPR*, 2018.

[21] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *NIPS*, 2012.

[22] A. Le Guennec, S. Malinowski, and R. Tavenard, "Data augmentation for time series classification using convolutional neural networks," in *ECML/PKDD workshop on advanced analytics and learning on temporal data*, 2016.

[23] A. J. Ratner, H. Ehrenberg, Z. Hussain, J. Dunnmon, and C. Ré, "Learning to compose domain-specific transformations for data augmentation," *NIPS*, 2017.

[24] O. Maron and T. Lozano-Pérez, "A framework for multiple-instance learning," *NIPS*, 1997.

[25] N. Natarajan, I. S. Dhillon, P. K. Ravikumar, and A. Tewari, "Learning with noisy labels," *NIPS*, 2013.

[26] X. Zhu and A. B. Goldberg, "Introduction to semi-supervised learning," *Synthesis lectures on artificial intelligence and machine learning*, 2009.

[27] S. Tong and E. Chang, "Support vector machine active learning for image retrieval," in *ACM Multimedia*, 2001.

[28] L. Fei-Fei, R. Fergus, and P. Perona, "One-shot learning of object categories," *T-PAMI*, 2006.

[29] K. Saenko, B. Kulis, M. Fritz, and T. Darrell, "Adapting visual category models to new domains," in *ECCV*, 2010.

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

[31] Y. Shen and E. Elhamifar, "Semi-weakly-supervised learning of complex actions from instructional task videos," in *CVPR*, 2022.

[32] Z. Yan, J. Liang, W. Pan, J. Li, and C. Zhang, "Weakly-and semi-supervised object detection with expectation-maximization algorithm," *arXiv preprint arXiv:1702.08740*, 2017.

[33] C. M. Bishop and N. M. Nasrabadi, *Pattern recognition and machine learning*. Springer, 2006, vol. 4.

[34] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin, "Emerging properties in self-supervised vision transformers," in *CVPR*, 2021.

[35] D. Niu, X. Wang, X. Han, L. Lian, R. Herzig, and T. Darrell, "Unsupervised universal image segmentation," in *CVPR*, 2024.

[36] X. Wang, R. Girdhar, S. X. Yu, and I. Misra, "Cut and learn for unsupervised object detection and instance segmentation," in *CVPR*, 2023.

[37] A. Arun, C. Jawahar, and M. P. Kumar, "Dissimilarity coefficient based weakly supervised object detection," in *CVPR*, 2019.

[38] ——, "Weakly supervised instance segmentation by learning annotation consistent instances," in *ECCV*, 2020.

[39] L. Lin, X. Wang, H. Liu, and Y. Qian, "Guided learning for weakly-labeled semi-supervised sound event detection," in *ICASSP*, 2020.

[40] J. Pan, P. Zhu, K. Zhang, B. Cao, Y. Wang, D. Zhang, J. Han, and Q. Hu, "Learning self-supervised low-rank network for single-stage weakly and semi-supervised semantic segmentation," *IJCV*, 2022.

[41] X. Zhai, A. Oliver, A. Kolesnikov, and L. Beyer, "S4l: Self-supervised semi-supervised learning," in *ICCV*, 2019.

[42] T. G. Dietterich, R. H. Lathrop, and T. Lozano-Pérez, "Solving the multiple instance problem with axis-parallel rectangles," *Artificial intelligence*, 1997.

[43] J. Foulds and E. Frank, "A review of multi-instance learning assumptions," *The knowledge engineering review*, 2010.

[44] J. Wang and J.-D. Zucker, "Solving the multiple-instance problem: A lazy learning approach," in *International Conference on Machine Learning*, 2000.

[45] S. Andrews and T. Hofmann, “Multiple instance learning via disjunctive programming boosting,” *NIPS*, 2003.

[46] Z.-H. Zhou and M.-L. Zhang, “Neural networks for multi-instance learning,” in *ICIIT*, 2002.

[47] M. Kumar, B. Packer, and D. Koller, “Self-paced learning for latent variable models,” *NIPS*, 2010.

[48] H. Bilen, M. Pedersoli, and T. Tuytelaars, “Weakly supervised object detection with convex clustering,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015.

[49] X. Wang, Z. Zhu, C. Yao, and X. Bai, “Relaxed multiple-instance svm with application to object discovery,” in *ICCV*, 2015.

[50] C.-N. J. Yu and T. Joachims, “Learning structural svms with latent variables,” in *ICML*, 2009.

[51] M. P. Kumar, B. Packer, and D. Koller, “Modeling latent variable uncertainty for loss-based learning,” *ICML*, 2012.

[52] K. Miller, M. P. Kumar, B. Packer, D. Goodman, and D. Koller, “Max-margin min-entropy models,” in *AISTATS*, 2012.

[53] Y. Y. Boykov and M.-P. Jolly, “Interactive graph cuts for optimal boundary & region segmentation of objects in nd images,” in *ICCV*, 2001.

[54] C. Rother, V. Kolmogorov, and A. Blake, “" grabcut" interactive foreground extraction using iterated graph cuts,” *TOG*, 2004.

[55] P. Krähenbühl and V. Koltun, “Efficient inference in fully connected crfs with gaussian edge potentials,” *NIPS*, 2011.

[56] H. O. Song, R. Girshick, S. Jegelka, J. Mairal, Z. Harchaoui, and T. Darrell, “On learning to localize objects with minimal supervision,” in *ICML*, 2014.

[57] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, “Decaf: A deep convolutional activation feature for generic visual recognition,” in *ICML*, 2014.

[58] W. Ren, K. Huang, D. Tao, and T. Tan, “Weakly supervised large scale object localization with multiple instance learning and bag splitting,” *T-PAMI*, 2015.

[59] R. G. Cinbis, J. Verbeek, and C. Schmid, “Weakly supervised object localization with multi-fold multiple instance learning,” *T-PAMI*, 2016.

[60] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, “Is object localization for free?-weakly-supervised learning with convolutional neural networks,” in *CVPR*, 2015.

[61] A. J. Bency, H. Kwon, H. Lee, S. Karthikeyan, and B. Manjunath, "Weakly supervised localization using deep feature maps," in *ECCV*. Springer, 2016.

[62] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *CVPR*, 2016.

[63] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *ICCV*, 2017.

[64] X. Chen and A. Gupta, "Webly supervised learning of convolutional networks," in *ICCV*, 2015.

[65] D. Li, J.-B. Huang, Y. Li, S. Wang, and M.-H. Yang, "Weakly supervised object localization with progressive domain adaptation," in *CVPR*, 2016.

[66] M. Shi, H. Caesar, and V. Ferrari, "Weakly supervised object localization using things and stuff transfer," in *ICCV*, 2017.

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

[68] P. O. Pinheiro and R. Collobert, "From image-level to pixel-level labeling with convolutional networks," in *CVPR*, 2015.

[69] D. Pathak, P. Krahenbuhl, and T. Darrell, "Constrained convolutional neural networks for weakly supervised segmentation," in *ICCV*, 2015.

[70] G. Papandreou, L.-C. Chen, K. P. Murphy, and A. L. Yuille, "Weakly-and semi-supervised learning of a deep convolutional network for semantic image segmentation," in *ICCV*, 2015.

[71] A. Kolesnikov and C. H. Lampert, "Seed, expand and constrain: Three principles for weakly-supervised image segmentation," in *ECCV*, 2016.

[72] H. Bilen and A. Vedaldi, "Weakly supervised deep detection networks," in *CVPR*, 2016.

[73] A. Diba, V. Sharma, A. Pazandeh, H. Pirsiavash, and L. Van Gool, "Weakly supervised cascaded convolutional networks," in *CVPR*, 2017.

[74] P. Vernaza and M. Chandraker, "Learning random-walk label propagation for weakly-supervised semantic segmentation," in *CVPR*, 2017.

[75] K. K. Singh and Y. J. Lee, "Hide-and-seek: Forcing a network to be meticulous for weakly-supervised object and action localization," in *ICCV*, 2017.

[76] J. Ahn and S. Kwak, "Learning pixel-level semantic affinity with image-level supervision for weakly supervised semantic segmentation," in *CVPR*, 2018.

[77] P. Tang, X. Wang, X. Bai, and W. Liu, "Multiple instance detection network with online instance classifier refinement," in *CVPR*, 2017.

[78] W. Ge, S. Yang, and Y. Yu, "Multi-evidence filtering and fusion for multi-label classification, object detection and semantic segmentation based on weakly supervised learning," in *CVPR*, 2018.

[79] X. Zhang, J. Feng, H. Xiong, and Q. Tian, "Zigzag learning for weakly supervised object detection," in *CVPR*, 2018.

[80] P. Tang, X. Wang, A. Wang, Y. Yan, W. Liu, J. Huang, and A. Yuille, "Weakly supervised region proposal network and object detection," in *ECCV*, 2018.

[81] Y. Zhang, Y. Bai, M. Ding, Y. Li, and B. Ghanem, "W2f: A weakly-supervised to fully-supervised framework for object detection," in *CVPR*, 2018.

[82] C. R. Rao, "Diversity and dissimilarity coefficients: a unified approach," *Theoretical population biology*, 1982.

[83] D. Bouchacourt, M. P. Kumar, and S. Nowozin, "Disco nets: Dissimilarity coefficient networks," in *NIPS*, 2016.

[84] D. Bouchacourt, "Task-oriented learning of structured probability distributions," Ph.D. dissertation, University of Oxford, 2017.

[85] H. Jhuang, J. Gall, S. Zuffi, C. Schmid, and M. J. Black, "Towards understanding action recognition," in *ICCV*, 2013.

[86] M. P. Kumar, B. Packer, and D. Koller, "Self-paced learning for latent variable models," in *NIPS*, 2010.

[87] A. Toshev and C. Szegedy, "Deeppose: Human pose estimation via deep neural networks," in *CVPR*, 2014.

[88] M. Andriluka, S. Roth, and B. Schiele, "Pictorial structures revisited: People detection and articulated pose estimation," in *CVPR*, 2009.

[89] L. Bourdev and J. Malik, "Poselets: Body part detectors trained using 3d human pose annotations," in *ICCV*, 2009.

[90] V. Ferrari, M. Marin-Jimenez, and A. Zisserman, "Progressive search space reduction for human pose estimation," in *CVPR*, 2008.

[91] S. Johnson and M. Everingham, "Learning effective human pose estimation from inaccurate annotation," in *CVPR*, 2011.

[92] L. Ladicky, P. H. Torr, and A. Zisserman, "Human pose estimation using a joint pixel-wise and part-wise formulation," in *CVPR*, 2013.

[93] L. Pishchulin, M. Andriluka, P. Gehler, and B. Schiele, "Strong appearance and expressive spatial models for human pose estimation," in *CVPR*, 2013.

[94] D. Ramanan, "Learning to parse images of articulated bodies," in *NIPS*, 2006.

[95] B. Sapp and B. Taskar, "Modec: Multimodal decomposable models for human pose estimation," in *CVPR*, 2013.

[96] Y. Yang and D. Ramanan, "Articulated human detection with flexible mixtures of parts," *IEEE Transactions on PAMI*, 2013.

[97] J. J. Tompson, A. Jain, Y. LeCun, and C. Bregler, "Joint training of a convolutional network and a graphical model for human pose estimation," in *NIPS*, 2014.

[98] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh, "Convolutional pose machines," in *CVPR*, 2016.

[99] X. Chu, W. Yang, W. Ouyang, C. Ma, A. L. Yuille, and X. Wang, "Multi-context attention for human pose estimation," in *CVPR*, 2017.

[100] W. Yang, S. Li, W. Ouyang, H. Li, and X. Wang, "Learning feature pyramids for human pose estimation," in *ICCV*, 2017.

[101] I. Lillo, J. Carlos Niebles, and A. Soto, "A hierarchical pose-based approach to complex action understanding using dictionaries of actionlets and motion poselets," in *CVPR*, 2016.

[102] C. Thurau and V. Hlavác, "Pose primitive based human action recognition in videos or still images," in *CVPR*, 2008.

[103] R. Vemulapalli, F. Arrate, and R. Chellappa, "Human action recognition by representing 3d skeletons as points in a lie group," in *CVPR*, 2014.

[104] R. Vemulapalli and R. Chellapa, "Rolling rotations for recognizing human actions from 3d skeletal data," in *CVPR*, 2016.

[105] A. Yao, J. Gall, and L. Van Gool, "Coupled action recognition and pose estimation from multiple views," *IJCV*, 2012.

[106] U. Iqbal, M. Garbade, and J. Gall, "Pose for action-action for pose," in *2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017)*, 2017.

[107] K. Raja, I. Laptev, P. Pérez, and L. Oisel, "Joint pose estimation and action recognition in image graphs," in *ICIP*, 2011.

[108] B. Xiaohan Nie, C. Xiong, and S.-C. Zhu, "Joint action recognition and pose estimation from video," in *CVPR*, 2015.

[109] T.-H. Yu, T.-K. Kim, and R. Cipolla, "Real-time action recognition by spatiotemporal semantic and structural forests." in *BMVC*, 2010.

[110] D. Bouchacourt, S. Nowozin, and M. Pawan Kumar, "Entropy-based latent structured output prediction," in *ICCV*, 2015.

[111] W. Ping, Q. Liu, and A. T. Ihler, "Marginal structured svm with hidden variables." in *ICML*, 2014.

[112] A. Schwing, T. Hazan, M. Pollefeys, and R. Urtasun, "Efficient structured prediction with latent variables for general graphical models," in *ICML*, 2012.

[113] T. Durand, N. Thome, and M. Cord, "Weldon: Weakly supervised learning of deep convolutional neural networks," in *CVPR*, 2016.

[114] D. Pathak, E. Shelhamer, J. Long, and T. Darrell, "Fully convolutional multi-class multiple instance learning," in *ICLR-W*, 2014.

[115] P. Tokmakov, K. Alahari, and C. Schmid, "Weakly-supervised semantic segmentation using motion cues," in *ECCV*, 2016.

[116] V. Premachandran, D. Tarlow, and D. Batra, "Empirical minimum bayes risk prediction: How to extract an extra few% performance from vision models with just three more parameters," in *CVPR*, 2014.

[117] C. Bishop, "Pattern recognition and machine learning (information science and statistics), 1st edn. 2006. corr. 2nd printing edn," 2007.

[118] V. Belagiannis, C. Rupprecht, G. Carneiro, and N. Navab, "Robust optimization for deep regression," in *ICCV*, 2015.

[119] J. Dai, Y. Li, K. He, and J. Sun, "R-FCN: Object detection via region-based fully convolutional networks," in *NIPS*, 2016.

[120] R. Girshick, "Fast R-CNN," in *ICCV*, 2015.

[121] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *CVPR*, 2014.

[122] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in *ECCV*, 2016.

[123] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *CVPR*, 2016.

[124] H. Bilen and A. Vedaldi, "Weakly supervised deep detection networks," in *CVPR*, 2016.

[125] A. Diba, V. Sharma, A. Pazandeh, H. Pirsiavash, and L. Van Gool, "Weakly supervised cascaded convolutional networks," in *CVPR*, 2017.

[126] Z. Jie, Y. Wei, X. Jin, J. Feng, and W. Liu, "Deep self-taught learning for weakly supervised object localization," in *CVPR*, 2017.

[127] M. Gao, A. Li, R. Yu, V. I. Morariu, and L. S. Davis, "C-wsl: Count-guided weakly supervised localization," in *ECCV*, 2018.

[128] B. Lai and X. Gong, "Saliency guided end-to-end learning for weakly supervised object detection," in *IJCAI*, 2017.

[129] D. Li, J.-B. Huang, Y. Li, S. Wang, and M.-H. Yang, "Weakly supervised object localization with progressive domain adaptation," in *CVPR*, 2016.

[130] S. Li, X. Zhu, Q. Huang, H. Xu, and C.-C. J. Kuo, "Multiple instance curriculum learning for weakly supervised object detection," in *BMVC*, 2017.

[131] P. Tang, X. Wang, X. Bai, and W. Liu, "Multiple instance detection network with online instance classifier refinement," in *CVPR*, 2017.

[132] P. Tang, X. Wang, A. Wang, Y. Yan, W. Liu, J. Huang, and A. Yuille, "Weakly supervised region proposal network and object detection," in *ECCV*, 2018.

[133] X. Zhang, J. Feng, H. Xiong, and Q. Tian, "Zigzag learning for weakly supervised object detection," in *CVPR*, 2018.

[134] X. Zhang, Y. Yang, and J. Feng, "ML-Locnet: Improving object localization with multi-view learning network," in *ECCV*, 2018.

[135] Y. Zhang, Y. Bai, M. Ding, Y. Li, and B. Ghanem, "W2F: A weakly-supervised to fully-supervised framework for object detection," in *CVPR*, 2018.

[136] A. Arun, C. V. Jawahar, and M. P. Kumar, "Learning human poses from actions," in *BMVC*, 2018.

[137] R. G. Cinbis, J. Verbeek, and C. Schmid, "Weakly supervised object localization with multi-fold multiple instance learning," *TPAMI*, 2017.

[138] H. O. Song, Y. J. Lee, S. Jegelka, and T. Darrell, "Weakly-supervised discovery of visual pattern configurations," *NIPS*, 2014.

[139] C. Wang, W. Ren, K. Huang, and T. Tan, "Weakly supervised object localization with latent category learning," in *ECCV*, 2014.

[140] P. Siva, C. Russell, and T. Xiang, "In defence of negative mining for annotating weakly labelled data," in *ECCV*, 2012.

[141] P. Siva and T. Xiang, "Weakly supervised object detector learning with model drift detection," in *ICCV*, 2011.

[142] W. Ge, S. Yang, and Y. Yu, "Multi-evidence filtering and fusion for multi-label classification, object detection and semantic segmentation based on weakly supervised learning," in *CVPR*, 2018.

[143] J. Wang, J. Yao, Y. Zhang, and R. Zhang, "Collaborative learning for weakly supervised object detection," in *IJCAI*, 2018.

[144] Z. Ren, Z. Yu, X. Yang, M.-Y. Liu, A. G. Schwing, and J. Kautz, "Ufo 2: A unified framework towards omni-supervised object detection," in *ECCV*, 2020.

[145] P. Chen, X. Yu, X. Han, N. Hassan, K. Wang, J. Li, J. Zhao, H. Shi, Z. Han, and Q. Ye, "Point-to-box network for accurate object detection via single point supervision," in *ECCV*, 2022.

[146] P. Tang, X. Wang, S. Bai, W. Shen, X. Bai, W. Liu, and A. Yuille, "Pcl: Proposal cluster learning for weakly supervised object detection," *TPAMI*, 2018.

[147] Z. Zeng, B. Liu, J. Fu, H. Chao, and L. Zhang, "Wsod2: Learning bottom-up and top-down objectness distillation for weakly-supervised object detection," in *ICCV*, 2019.

[148] Y. Gao, B. Liu, N. Guo, X. Ye, F. Wan, H. You, and D. Fan, "C-midn: Coupled multiple instance detection network with segmentation guidance for weakly supervised object detection," in *ICCV*, 2019.

[149] Z. Ren, Z. Yu, X. Yang, M.-Y. Liu, Y. J. Lee, A. G. Schwing, and J. Kautz, "Instance-aware, context-focused, and memory-efficient weakly supervised object detection," in *CVPR*, 2020.

[150] J. Seo, W. Bae, D. J. Sutherland, J. Noh, and D. Kim, "Object discovery via contrastive learning for weakly supervised object detection," in *ECCV*, 2022.

[151] Y. Yin, J. Deng, W. Zhou, L. Li, and H. Li, "Cyclic-bootstrap labeling for weakly supervised object detection," in *ICCV*, 2023.

[152] J. R. Uijlings, K. E. Van De Sande, T. Gevers, and A. W. Smeulders, "Selective search for object recognition," *IJCV*, 2013.

[153] P.-T. Jiang, C.-B. Zhang, Q. Hou, M.-M. Cheng, and Y. Wei, "Layercam: Exploring hierarchical class activation maps for localization," *TIP*, 2021.

[154] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, and K. Murphy, "Speed/accuracy trade-offs for modern convolutional object detectors," in *CVPR*, 2017.

[155] T. Deselaers, B. Alexe, and V. Ferrari, "Weakly supervised localization and learning with generic knowledge," *IJCV*, 2012.

[156] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014.

[157] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016.

[158] X. Zhou, R. Girdhar, A. Joulin, P. Krähenbühl, and I. Misra, "Detecting twenty-thousand classes using image-level supervision," in *ECCV*, 2022.

[159] J. Lin, Y. Shen, B. Wang, S. Lin, K. Li, and L. Cao, "Weakly supervised open-vocabulary object detection," in *AAAI*, 2024.

[160] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo *et al.*, "Segment anything," in *CVPR*, 2023.

[161] L.-C. Chen, A. Hermans, G. Papandreou, F. Schroff, P. Wang, and H. Adam, "Masklab: Instance segmentation by refining object detection with semantic and direction features," in *CVPR*, 2018.

[162] Y. Li, H. Qi, J. Dai, X. Ji, and Y. Wei, "Fully convolutional instance-aware semantic segmentation," in *CVPR*, 2017.

[163] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," in *CVPR*, 2018.

[164] D. Novotny, S. Albanie, D. Larlus, and A. Vedaldi, "Semi-convolutional operators for instance segmentation," in *ECCV*, 2018.

[165] C.-C. Hsu, K.-J. Hsu, C.-C. Tsai, Y.-Y. Lin, and Y.-Y. Chuang, "Weakly supervised instance segmentation using the bounding box tightness prior," in *NeurIPS*, 2019.

[166] J. Ahn, S. Cho, and S. Kwak, "Weakly supervised learning of instance segmentation with inter-pixel relations," in *CVPR*, 2019.

[167] H. Cholakkal, G. Sun, F. S. Khan, and L. Shao, "Object counting and instance segmentation with image-level supervision," in *CVPR*, 2019.

[168] W. Ge, S. Guo, W. Huang, and M. R. Scott, "Label-PEnet: Sequential label propagation and enhancement networks for weakly supervised instance segmentation," in *ICCV*, 2019.

[169] I. H. Laradji, D. Vazquez, and M. Schmidt, "Where are the masks: Instance segmentation with image-level supervision," *BMVC*, 2019.

[170] Y. Zhou, Y. Zhu, Q. Ye, Q. Qiu, and J. Jiao, "Weakly supervised instance segmentation using class peak response," in *CVPR*, 2018.

[171] Y. Zhu, Y. Zhou, H. Xu, Q. Ye, D. Doermann, and J. Jiao, "Learning instance activation maps for weakly supervised instance segmentation," in *CVPR*, 2019.

[172] Z. Huang, X. Wang, J. Wang, W. Liu, and J. Wang, "Weakly-supervised semantic segmentation network with deep seeded region growing," in *CVPR*, 2018.

[173] S. J. Oh, R. Benenson, A. Khoreva, Z. Akata, M. Fritz, and B. Schiele, "Exploiting saliency for object segmentation from image level labels," in *CVPR*, 2017.

[174] D. Lin, J. Dai, J. Jia, K. He, and J. Sun, "Scribblesup: Scribble-supervised convolutional networks for semantic segmentation," in *CVPR*, 2016.

[175] J. Dai, K. He, and J. Sun, "Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation," in *CVPR*, 2015.

[176] G. Papandreou, L.-C. Chen, K. Murphy, and A. Yuille, "Weakly-and semi-supervised learning of a dcnn for semantic image segmentation," in *ICCV*, 2015.

[177] Y. Wei, J. Feng, X. Liang, M.-M. Cheng, Y. Zhao, and S. Yan, "Object region mining with adversarial erasing: A simple classification to semantic segmentation approach," in *CVPR*, 2017.

[178] P. Arbeláez, J. Pont-Tuset, J. Barron, F. Marques, and J. Malik, "Multiscale combinatorial grouping," in *CVPR*, 2014.

[179] S. Kohl, B. Romera-Paredes, C. Meyer, J. De Fauw, J. R. Ledsam, K. Maier-Hein, S. A. Eslami, D. J. Rezende, and O. Ronneberger, "A probabilistic u-net for segmentation of ambiguous images," in *NIPS*, 2018.

[180] T. Hazan, J. Keshet, and D. A. McAllester, "Direct loss minimization for structured prediction," in *NIPS*, 2010.

[181] Y. Song, A. Schwing, R. Urtasun *et al.*, "Training deep neural networks via direct loss minimization," in *ICML*, 2016.

[182] G. Lorberbom, A. Gane, T. Jaakkola, and T. Hazan, "Direct optimization through argmax for discrete variational auto-encoder," in *NeurIPS*, 2019.

[183] B. Hariharan, P. Arbeláez, L. Bourdev, S. Maji, and J. Malik, "Semantic contours from inverse detectors," in *ICCV*, 2011.

[184] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik, "Simultaneous detection and segmentation," in *ECCV*, 2014.