OBJECT DETECTION FRAMEWORKS FOR FULLY AUTOMATED PARTICLE PICKING IN CRYO-EM

ABBAS MASOUMZADEH TORK

A THESIS SUBMITTED TO
THE FACULTY OF GRADUATE STUDIES
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF SCIENCE

GRADUATE PROGRAM IN COMPUTER SCIENCE
YORK UNIVERSITY
TORONTO, ONTARIO

August 2019

ABSTRACT

Particle picking in cryo-EM is a form of object detection for noisy, low contrast, and out-of-focus microscopy images, taken of different (unknown) structures. This thesis presents a fully automated approach which, for the first time, explicitly considers training on multiple structures, while simultaneously learning both specialized models for each structure used for training and a generic model that can be applied to unseen structures. The presented architecture is fully convolutional and divided into two parts: (i) a portion which shares its weights across all structures and (ii) N+1 parallel sets of sub-architectures, N of which are specialized to the structures used for training and a generic model whose weights are tied to the layers for the specialized models. Experiments reveal improvements in multiple use cases over the-state-of-art and present additional possibilities to practitioners.
Keywords: Cryo-EM, particle picking, object detection, fully convolutional, dataset bias

Acknowledgments

First and foremost, I would like to thank my supervisor, Prof. Marcus Brubaker. Words cannot express my appreciation and gratitude for his advice, encouragement, support, and patience throughout my Master studies. I am sincerely grateful that he provided me with many opportunities to engage with the research communities of computer vision and Cryo-EM and to achieve research experience within and outside academia.

I would like to express my gratitude to Prof. Michael Brown for his invaluable comments and questions that added new perspectives to the thesis. I am thankful for his advice and encouragement regarding academic research and communication.

I am grateful to my friends at the Computational Vision and Imaging Lab, the Computer Vision Reading Group, and the EECS program at York who shared a happy, encouraging, and productive environment with me.

Last but not least, I would like to thank my parents and my brothers for their constant kindness and support, no matter the distance between us.

This research was financially supported by the Vision: Science to Applications (VISTA) program by Centre for Vision Research (CVR) thanks in part to funding from the Canada First Research Excellence Fund (CFREF).

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Electron cryomicroscopy (cryo-EM) is an experimental technique that captures images of biological samples at cryogenic temperatures using a transmission electron microscope. Single particle analysis of cryo-EM images is a set of computational procedures which aim to determine the 3D structure of single particles using 2D electron microscopy images (or micrographs) [1]. This study presents a novel approach to one of the first computational problems in single particle cryo-EM known as *particle picking*.

1.1 Problem Statement and Challenges

In particle picking the goal is to locate individual particles in a micrograph while avoiding contaminants, malformed particles and background regions. In other words, the input of the problem is a micrograph and the desired output is the coordinates of all particles in that micrograph image. Accurate detection of particles is necessary, as the presence of contaminating particles can complicate subsequent processing, degrade the resolution of the final estimated 3D structure or even cause the reconstruction process to fail entirely.

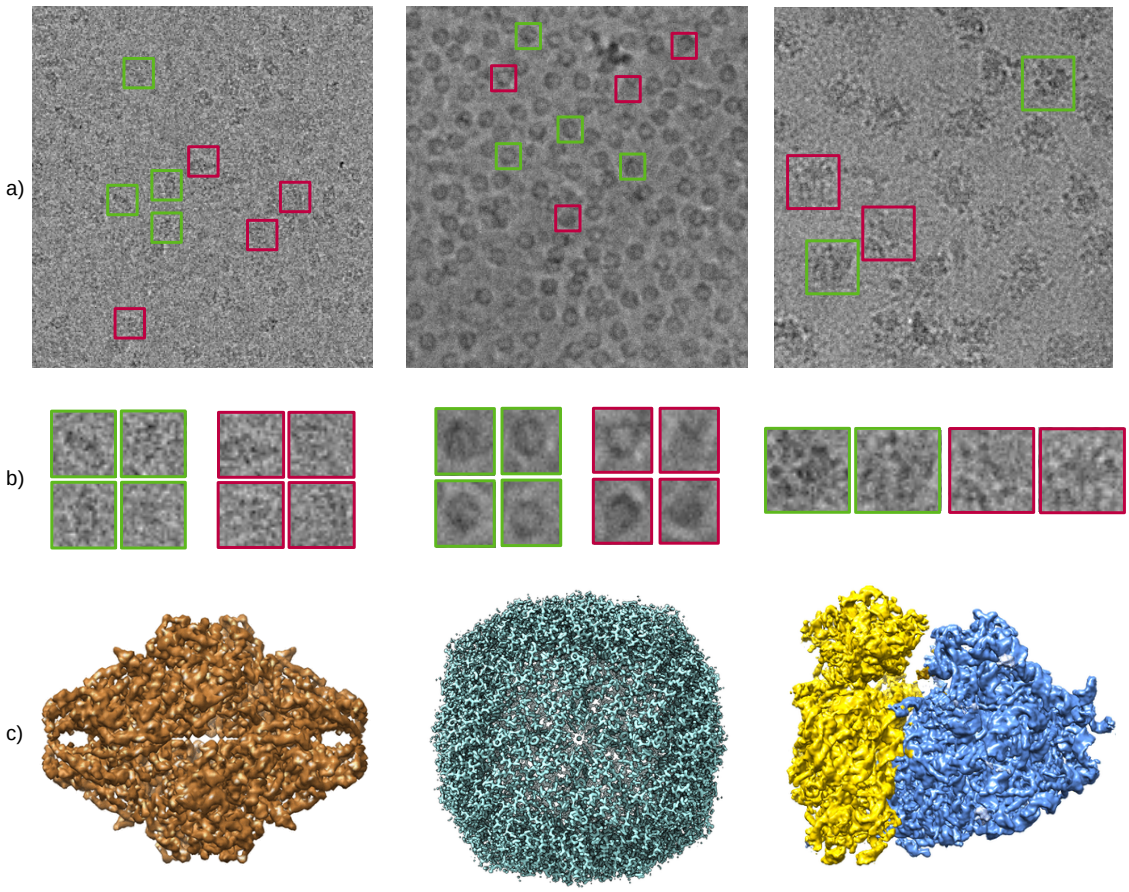

Figure 1.1: Visualization of Particle Picking. a) Micrograph patches of Beta-galactosidase (left), Apoferritin (middle), and Ribosome (right). b) Zoomed in boxes of a few correct particles (green) and corrupted particles (red). c) 3D reconstruction of the molecules using such particles. [2]–[4].

The picking task is challenging due to several factors, including high levels of noise, low contrast of particles, and variability of the appearance of an individual particle caused by changes in orientation and differences of structure between different particles. Figure 1.1 shows some sample micrographs, particle images and their corresponding 3D structures to illustrate the problem. In general, when performing particle picking for a new experiment, the appearance of particles in the case is unknown meaning that structure specific training data is unavailable. This has led previous researchers to attempt to use the appearance of other particles to train learning-based picking approaches by pooling data [5]–[10]. However, this can be problematic as different particles and datasets can have significantly different appearances and quantities of data leading to biases or degraded performance. Here we argue that this problem is analogous to the "dataset bias" problem which has been identified and considered in object recognition generally [11]–[13].

1.2 Summary of the Thesis

Motivated by the importance of high-quality 3D reconstruction of biological molecules and the role particle picking plays, we aim to leverage methods from computer vision to establish an objective comparative study on the topic and further improve the accuracy of fully automated particle picking.

In chapter 2, we review the literature related to fully automated particle picking in cryo-EM. We also look at the related background in computer vision, most notably deep learning based object detection methods and cross dataset studies in image recognition.

In chapter 3, We formulate particle picking as an object detection task and build off of modern object detection approaches, in particular the Single Shot Detector (SSD) approach [14]. However, unlike SSD we formulate the network architecture and learning problem

to represent and model the existence of particles from different datasets explicitly. The proposed approach consists of a network with a shared trunk and multiple heads, one head for each dataset and an additional head which can be used for zero-shot picking where the particles are of a previously unseen structure. We call this model HydraPicker.

In chapter 4, we consider the performance of HydraPicker in both a zero-shot and a few-shot setting (where limited training data of a new structure is available) which simulate the most important use cases for particle picking. Our results demonstrate the value of the new formulation, enabling performance improvements in both zero-shot and few-shot settings. We compare the proposed method directly against several recent learning-based particle picking methods in one of the most thorough experimental comparisons in the literature and establish HydraPicker as a new state-of-the-art for particle picking.

In chapter 5, we conclude this study with a discussion on major findings and a roadmap for future studies in multiple directions from particle picking to general computer vision.

1.3 Contributions

The main contributions of this study are: 1) Adaptation of deep learning architectures for image recognition and object detection, respectively ResNet and SSD, to specific properties of particle picking task to gain state-of-the-art accuracy in particle picking; 2) Explicit consideration of dataset bias in a deep learning framework for object detection in the context of particle picking problem as a proof of concept; 3) A framework for conducting comparative quantitative and qualitative study on practical use cases of the latest fully-automated learning-based particle picking approaches; 4) Several future directions for further studies both on the subject of particle picking and dataset bias in computer vision.

Chapter 2

Background

This chapter presents a literature review on fully-automated learning-based particle picking in cryo-EM. It also provides the related background in computer vision tasks including image recognition, object detection, and image segmentation.

Particle picking was traditionally done manually, through a time-consuming process where experts selected particles from hundreds or even thousands of micrographs. In cases where a low resolution model of the molecule or a related molecule is available template-based methods can be applied [15]–[18] but this limits the usefulness of the approach to effectively known structures. Such picking approaches are tedious, expensive and risks introducing biases into the process and fully automatic picking has always been a goal.

Many automatic approaches over the years have been tried including contrast enhancement [17] and difference of Gaussians [18]. However, results of such efforts have generally not been accurate enough to be used in a fully automated procedure. Instead these methods have often been used as part of semi-automatic methods where a high recall [1] automated method is used to select candidate particles which are shown to the experts to label [2], [19]

[1] The fraction of ground truth particles that are picked

and in some cases learning from this manual annotation to improve performance [20].

Recently, researchers have started to explore the use of deep learning in computer vision to improve the fully automated particle picking task. There are three main automated approaches to deal with objects in computer vision.

2.1 Recognition-Based Approach

Image recognition is defined as a classification problem in which it is assumed that the input is an image corresponding only to exactly one of the provided classes. It has been in focus as a main computer vision task, specially since the adaptation of deep neural networks to it in an approach known as AlexNet and its impressive accuracy in ImageNet challenge [21] which resulted in further popularity and adaptation of convolutional neural networks (CNNs) to many other applications.

Three main components used in such network architectures include convolutional, pooling, and fully connected layers. Convolutional layers in this case are usually defined as performing a 2D convolution with a 3×3 or 5×5 filter on each input channel of the layer followed by a rectification. Pooling layers usually choose the maximum or average value of every 3×3 or 5×5 patch of the input to the next layer which respectively helps to emphasize edges or smoothens the input while performing an up-scaling. Fully connected layers are used as the final layers of the networks to both flatten the outputs and perform a learning function considering all possible combinations of inputs.

Simonyan and Zisserman [22] won the ImageNet 2014 challenge by further exploring the network architectures. By introducing VGG-Net they considered having multiple 3×3 convolutional layers in between pooling layers, increasing the number of channels after each pooling layer and increasing the total depth of the network to 16-19 layers.

One of the main challenges of increasing the depth and consequently the computational power of CNNs was the ability to train them, as the flow of gradients would start to vanish in direct relation with the increased depth. He *et al.* [23] made an important contribution to the community by proposing shortcut connections in an architecture made of multiple similar components called residual blocks. These would allow the next layer to optimize an additive combination of the output of immediate previous block and an identity mapping coming from the output of a block before that. As a consequence, facilitating the gradient flow to deeper layers of the network, training architectures with hundreds of layers was made possible.

Recognition-Based Particle Picking The first particle picking approach enhanced by deep learning was DeepEM [5]. It utilized a simple CNN architecture based on AlexNet [21] to train a model that can pick particles from unseen images of the same dataset and an iterative process to improve picking performance from partially labelled data. DeepPicker [6] used a customized VGG-Net [22] architecture and trained on multiple molecules to try to create a more generic particle picking approach for unknown targets. However, training and testing was generally limited to a small number of datasets and performance indicated generally low precision which was somewhat improved with better data pre-processing steps, including micrograph sharpening and histogram equalization, in subsequent work [7].

2.2 Detection-Based Approach

More recently, advances in object detection architectures have been explored in particle picking as well. Unlike image recognition, object detection loosens the assumption of the input and allows it to contain one or multiple objects from any of the considered classes. It

further more requires the solution to have a pair of corresponding coordinates associated to each of the detected objects.

The first deep neural network approach to successfully address this task was introduced by Girshick *et al.* [24]. RCNN associates each input image to 2,000 proposed regions of interest given by selective search [25], which considers multiple hierarchical grouping strategies for the task. It then passes the proposed regions through a CNN to have a set of futures for each region. Finally, it uses class specific linear Support Vector Machines (SVMs), a category of kernel-based methods which had state-of-the-art results in classification tasks before the appearance of CNNs in the literature, to classify each region based on its features.

Main problems with RCNN were its slow and multi-stage training procedure and slow detection performance. To address these issues, Fast-RCNN [26] was introduced which was re-organized to first pass the whole image once through a CNN and then propose regions from the output feature-map. This approach combined with a multi-task training loss that considered both location and classification at the same time, resulted in both speed and accuracy improvements.

Faster-RCNN [27] further improved this line of research by introducing a trainable region proposal network to replace selective search and therefore train the whole network end-to-end. It is still known as one of the most accurate object detection approaches for natural images in the literature.

A faster approach in object detection, known as single-shot detector was developed in two concurrent but separate projects by Redmon *et al.* [28] and Liu *et al.* [14]. The main idea was to remove the region proposal section of the solution and try to detect objects using features learned directly over pre-defined bounding boxes of the input image. Their

first implementations were not as accurate as Faster-RCNN and therefore introduced as methods that drastically improve the speed of detection and can be used for real-time object detection in videos. Further improvements and customizations over these approaches have resulted in a range of solutions that allows the user to choose the architecture based on a trade-off between speed and accuracy [29].

Detection-Based Particle Picking Inspired by the advancements of object detection in generic computer vision, Xiao and Yang [8] considered Fast R-CNN [26] for particle picking. They decided to use a simplified region proposal method and some of the pre-processing introduced in [7]. They also introduced explicit labels for contaminants as distinct from background which helped reduce false positives. However, the method was only reported on three datasets and with no direct comparisons to existing techniques.

Zheng, Ni, and Zhao [30] followed the approach in [8] but instead of a fixed size sliding window, they used the Region Proposal Network (RPN) in Faster-RCNN [27] architecture. They also used previously discussed pre-processing steps and considered coefficients of 90 degrees rotations as augmentations. The performance was only reported on one dataset.

SPHIRE-crYOLO [9], customizes the You Only Look Once (YOLO) [28] architecture for particle picking which significantly improves picking speed while maintaining reasonable precision and recall rates. Direct comparison against other learning-based particle picking methods were not provided but showed improvements over a baseline semi-automated approach [31]. The model was trained on a large number of datasets, but when tested on a held-out target dataset comparing against a similar model specially trained for the target dataset, poorer performance was demonstrated, suggesting that even the much larger training set was not helpful enough for generalization.

2.3 Segmentation-Based Approach

A similar but conceptually lower level task in computer vision is semantic segmentation. It aims to assign a class label for each pixel in the image not considering the higher level concept of objects and their instances in an image. Ronneberger, Fischer, and Brox [32] have presented one of the most successful CNN architectures to address this task, U-Net. Its main idea is to compress the representation space of the image by a series of pooling and convolutional layers, before up-sampling those features back to the input image resolution. Therefore, it gives a same label to connected components in the pixel space.

Segmentation-Based Particle Picking In an approach called BoxNet, Tegunov and Cramer [10] formulated particle picking as a segmentation problem and used an architecture similar to U-Net [32]. This requires multiple post-processing steps to avoid picking from detected contaminated regions and to identify the final coordinates of the selected particles which can be particularly challenging in crowded micrographs. The study did not provide any quantitative comparisons.

2.4 Dataset Bias

The proposed particle picking approach is also motivated by the work on dataset bias in image recognition [11] in which one important obstacle for generalization of trained models is shown to be the bias towards the datasets used in training. Khosla *et al.* [12] considered training a shared SVM with explicit bias vectors for each training dataset and reported improvements in generalization over the traditional approach of training a single SVM using all datasets. Tommasi *et al.* [13] followed this research line and analyzed the effects

of multiple SVM solutions for domain-adaptation besides a specifically designed SVM to the task of cross-dataset generalization in image recognition [33]. They concluded that SVM cannot undo the damage of dataset bias on features learned through CNNs and seek a deep learning solution for future research directions.

2.5 Proposed Approach

The approach proposed here is most closely related to the detection-based approaches [8], [9] in that a detection-based training framework is used. However, we adapt a network architecture to the specific requirements of the picking task. Further, unlike existing approaches, we explicitly represent the fact that picked particles used in training come from multiple, different datasets corresponding to different structures. We can view the problem of particle picking on a new dataset as a form of dataset bias given only a finite (biased) sample of currently available datasets used in training. Unlike previous work on dataset bias [12], [13] with SVM, we introduce a deep learning architecture to concurrently learn specialized models for each dataset and an extra generalized model.

Beyond a novel technical approach and with the hope of encouraging the Cryo-EM community to form a standardized evaluation framework for particle picking, we further provide a detailed evaluation of our method on a range of datasets [10] and in realistic scenarios using conventional metrics in visual object detection and analyze the modelling decisions made. Further, we perform direct baseline comparisons to existing methods to demonstrate that HydraPicker represents the new state-of-the-art in particle picking.

Chapter 3

Technical Approach

Here we now outline the proposed HydraPicker method. The CNN architecture is modelled in two parts, a body which acts as a feature extractor and a set of dataset specific heads which are tied together by an enforced similarity to a "generalization" head which can be used for the zero-shot case, i.e. on datasets without labelled data.

3.1 Network Architecture

For the architecture of the body, we construct a variation of the ResNet architecture [23] which utilizes residual connections to improve training. However, the basic ResNet architecture was designed for natural images which have significantly different characteristics with cryo-EM micrographs. In particular, the high level of noise in particle picking suggests that the 3×3 filters in ResNet may be suboptimal as information must be aggregated over much larger spatial extents to perform effective detection. Larger filter sizes would be natural but quickly increases the computational costs and number of parameters that need to be estimated which can lead to slow training and overfitting. Instead, we replace

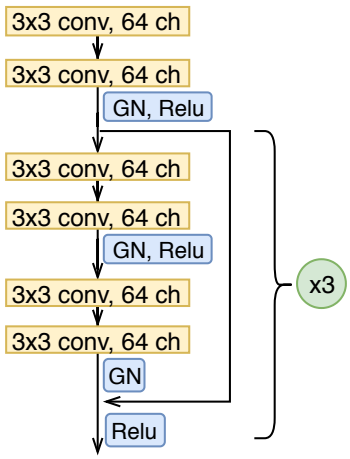

Figure 3.1: The shared recognition network. It consists of two stacked 3x3 convolutions, followed by three ResNet-like blocks of pairs of two stacked 3x3 convolutions with shortcut connections.

the single 3×3 convolutional layers with pairs of two 3×3, *without* an intervening non-linearity. This gives an effective filter size of 5×5 but with a reduced parameter count and computational requirements. For simplicity we use a consistent numbers of channels (64) throughout and consequently forego the 1×1 convolutional layer. Because we are using the body as a feature extractor for input into dataset specific heads, we remove the fully connected layers. This has the added benefit of ensuring that the architecture is fully convolutional. Finally, in order to handle smaller batch sizes during training we replace the batch normalization layers with group normalization layers [34] which divide channels into groups and normalize them separately. The complete architecture is shown in Figure 3.1.

The body forms a common feature extractor for multiple detector heads whose design (Figure 3.2) is derived from the single shot detector (SSD) framework [14]. SSD, like some previous approaches [28] operates by predicting detections and bounding boxes at a grid of

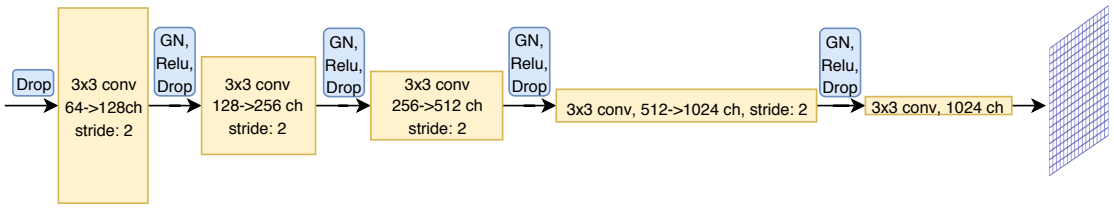

Figure 3.2: Architecture Diagram for Each of the SSD Heads. Each layer has half the resolution of the previous layer until it reaches the resolution of an assumed grid over the input. At the same time, each layer has twice the number of channels to allow more of the information to pass through. Output is an assumed grid over the image providing coordinates and classification probabilities for each box which represents 16×16 pixels.

anchor points. The network is trained using a focal classification loss [35]. This combination of SSD and the focal classification loss (called RetinaNet) has achieved state of the art performance for detection on natural images [35]. While SSD is a good starting point, we adapt the approach in several key ways to make it better suited for particle picking. First, in object detection there are many classes of objects which could be detected and so the output at each anchor point is a multi-class classification of which object is detected or no detection. In the case of particle picking there is only a single class (particle) or not particle. Second, objects in natural images can be at many different scales and with significant variations in aspect ratio and consequently the bounding box prediction at an anchor point includes not only an offset but also the size and aspect ratio of the bounding box. In the case of particle picking, because the images are orthographic projections, all particles of the same type will generally have the same size. Between different particles we assume that the input micrographs have been rescaled so that different particles share the same extent in pixels.[1] Further, bounding boxes for particle picking are square to simplify subsequent processing. Thus, in our adapted network the output at each anchor point for the bounding

[1]This is a relatively mild assumption in practice.

box needs only to include the offset.

The architecture itself consists of a sequence of 4 blocks consisting of 3×3 convolutions with a stride of 2, rectification, group normalization and dropout with the number of channels being 128, 256, 512, and 1024. The final layer is then a 3×3 convolutional layer with stride of 1 and with 3 outputs: 2 for the offset of the bounding box form the anchor and 1 for the (log) probability of a particle being detected at that anchor.

3.2 Training

The above architecture with a single head can be trained on a large number of micrographs and will work well on its own. However, as discussed, a major issue is the balance between different datasets which can have significantly different numbers of detected particles and the effective generalization of the approach as the number of datasets grows. Instead, HydraPicker uses a different head for each dataset that it is trained on, plus an additional head which generalizes picking on unseen datasets, *i.e.*, it is used for picking particles without training data. Inspired by [12], this generalization head is trained with an additional loss which encourages the weights of the dataset specific heads to be close to those of the generalization head. Conceptually, we can consider that there exists a general particle picking head which should work well on all datasets and dataset specific heads which are similar to this general head but with mild specializations for their specific datasets. Thus, the generalization head of HydraPicker is implicitly trained by requiring that it be similar to the dataset specific heads.

The is done by using the following loss function:

$$\ell_{\text{Hydra}} = \ell_{\text{loc}} + \lambda_{\text{cls}}\ell_{\text{cls}} + \lambda_{\text{bias}}\ell_{\text{bias}} \,, \tag{3.1}$$

where ℓ_{loc} is a localization loss which penalizes errors in the bounding box prediction, ℓ_{cls} is a classification loss which penalizes incorrect detections, ℓ_{bias} encourages the dataset specific heads to be close to the generalization head and λ_{cls} and λ_{bias} are hyperparameters which weight the losses. We discuss each part of this overall loss in turn next.

The localization loss is:

$$\ell_{\text{loc}}\left(\delta\mathbf{p}\,,\mathbf{p}\right) = \sum_{(i\,,j)\,\in M} ||(\mathbf{a}_i + \delta\mathbf{p}_i) - \mathbf{p}_j||_1\,, \tag{3.2}$$

where $\mathbf{a}_i$ is the location of the ith anchor, $\delta\mathbf{p}_i$ is the predicted offset of the bounding box at the ith anchor and $\mathbf{p}_j$ is the ground truth location of the bounding box for the jth detection. The sum is taken over the set M of particles in a micrograph and their corresponding anchor points. Formally $M = \{(i,j)|\text{IOU}[\text{box}(\mathbf{a}_i), \text{box}(\mathbf{p}_j)] > 0.6\}$ where IOU is the Intersection Over Union (or Jacquard index) between the anchor box and the particle bounding box and the threshold of 0.6 is selected to match previous approaches [9].

For the classification loss, we use the focal classification loss [35]. Specifically,

$$\ell_{\text{cls}} = \sum_i -\frac{1}{2}\alpha_{c_i}(1-p_i)^\gamma \log p_i\,, \tag{3.3}$$

where c_i is the correct class at the ith location, p_i is the probability of the correct class at the ith location, α_c is a class-specific constant factor which accounts for imbalance between the particle and background classes, γ is a hyperparameter and the sum is taken over all detections. The focal loss is similar to a standard cross-entropy classification loss. However, the term $(1-p_i)^\gamma$ downweights detections where the probability of the correct class, p_i, is close to 1. That is, it downweights detections which are generally easy and allows learning to focus more on hard cases. As such, the focal loss can be considered a form of

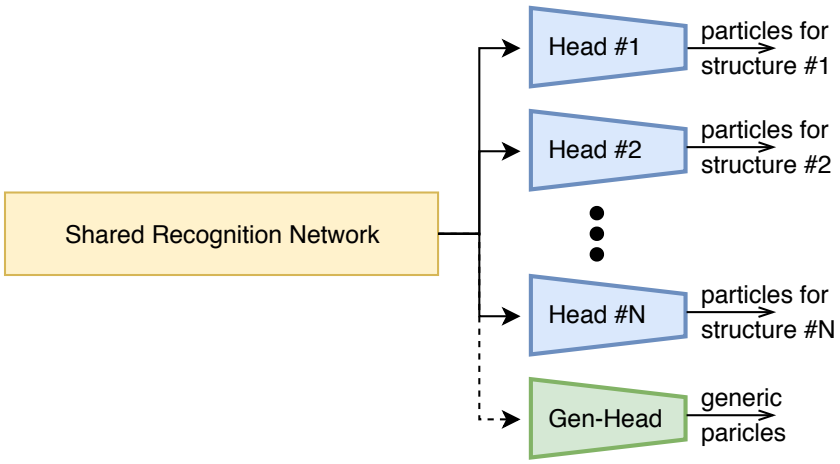

Figure 3.3: High-Level Diagram of HydraPicker. At training time on N datasets, all micrographs are passed through the shared portion of the architecture. However, each is only passed through the head assigned to its dataset resulting in a specialized model for that dataset. The generic head has its weights tied to all other heads. It both learns a generic model and acts as a regulator.

hard example mining. We use a value of $\gamma = 2$ which is typical.

The classification loss is evaluated using the correct head for each dataset, enabling both the shared body and the dataset specific heads to learn. To enable learning of the generalization head and to encourage the different, dataset specific heads to be similar to the generalization head we use a simple form of the generalization loss. Specifically,

$$\ell_{\text{bias}}(W_{S_1}, \ldots, W_{S_N}, W_G) = \frac{1}{N} \sum_{k=1}^{N} ||W_{S_k} - W_G||_2 \,, \tag{3.4}$$

where N is the number of datasets used in training, W_{S_i} is the set of weights for the ith dataset specific head and W_G is the set of weights for the generalization head.

Chapter 4

Experiments and results

4.1 Datasets

To evaluate HydraPicker, we made use of a collection of 37 datasets collected by Tegunov and Cramer [10]. These datasets are a mix of real and synthetic data from real structures all of which has been annotated per pixel. To choose a single pair of coordinates for each particle, we picked the central coordinates of each connected component in the annotation. Each dataset has between 4 to 103 micrographs with approximately 30 to 800 particles per micrograph. To account for scale variations all 37 datasets were re-scaled so that the target particles would have similar sizes. In order to have more representative augmentations during training, micrographs were padded with Gaussian noise with its mean and standard deviation chosen separately based on the background regions for each micrograph. For each dataset a small number of micrographs are randomly chosen as validation and test micrographs. Finally, we further split the data into 30 "source" datasets (table 4.1) and 7 "target" datasets (table 4.2). This split is used to test the performance of the methods on previously unseen target datasets. The 7 target datasets are chosen to have a diverse set of

Access Code	Sample	Synthetic	Phase Plate	Approximated Particle Size (in pixels)	Number of Micrographs
EMPIAR-10017	beta-galactosidase			22	84
EMPIAR-10077	80S ribosome			25	26
EMPIAR-10078	20S proteasome		✓	20	30
EMPIAR-10081	HCN1 channel			18	30
EMPIAR-10084	Haemoglobin		✓	8	15
EMPIAR-10089	TcdA1 in prepore state			24	24
EMPIAR-10097	Influenza Hemagglutinin			14	30
EMPIAR-10122	Apoferritin		✓	16	25
EMPIAR-10153	80S ribosome		✓	25	71
EMPIAR-10156	80S ribosome			31	21
gk_1	RNA Polymerase II complex			23	18
hh_2	RNA Polymerase II complex			21	36
lf_1	RNA Polymerase II complex			15	11
PDB-1sa0	Tubulin-Colchicine	✓		20	8
PDB-2gtl	Lumbricus Erythrocruorin	✓		13	8
PDB-3j9i	Thermoplasma acidophilum 20S proteasome	✓		18	8
PDB-4zor	S37P MS2 viral capsid	✓		17	8
PDB-5foj	Grapevine Fanleaf Virus complex with Nanobody	✓		8	5
PDB-5mmi	Chloroplast ribosome, large subunit	✓		24	8
PDB-5ngm	70S ribosome	✓		30	8
PDB-5w3s	TRPML3 ion channel	✓		13	8
PDB-5xwy	LbuCas13a-crRNA binary complex	✓		11	8
PDB-5y6p	Phycobilisome	✓		78	22
PDB-6az1	80S ribosome, small subunit	✓		32	8
PDB-6bco	TRPM4 in ATP bound state with short coiled coil	✓		15	8
PDB-6bcq	TRPM4 in ATP bound state with long coiled coil	✓		26	8
PDB-6bcx	mTORC1	✓		23	8
PDB-6bhu	Multidrug Resistance Protein 1 (MRP1)	✓		14	8
PDB-6bqv	Human TRPM4 ion channel	✓		17	8
ss_1	Viral polymerase			20	103

Table 4.1: Properties of source datasets. The information (except for particle size) was retrieved from [10], the Electron Microscopy Public Image Archive (EMPIAR) [36], and the Protein Data Bank (PDB) [37].

Access Code	Sample	Synthetic	Phase Plate	Approximated Particle Size (in pixels)	Number of Micrographs
PDB-2wri	70S ribosome	✓		24	8
PDB-4hhb	Human deoxyhaemoglobin	✓		7	4
PDB-5vy5	Rabbit muscle aldolase	✓		10	5
PDB-5w3l	Rhinovirus B14	✓		38	8
PDB-5xnl	Stacked PSII-LHCII supercomplex	✓		37	8
PDB-6b7n	Porcine delta coronavirus spike protein in the pre-fusion state	✓		14	8
PDB-6b44	CRISPR Csy surveillance complex with bound target dsDNA	✓		17	8

Table 4.2: Properties of target datasets. The information (except for particle size) was retrieved from [10] and the Protein Data Bank (PDB) [37].

structures. Besides, they are all chosen from the synthetic datasets to be confident of the picked ground truth coordinates, avoiding the possibility of human errors in their picking process.

4.2 Baselines

We compare our results on these datasets against two state-of-the-art learning-based particle picking methods, BoxNet [10] and crYOLO [9]. For these baseline methods we used published code to retrain the models using the same experimental setup as for HydraPicker.

For both methods we used the latest release but to ensure adherence to the experimental setup, we had to retrain them from scratch as the available models training sets overlapped with our held-out test and validation micrographs and target datasets.

We used the latest stable release of BoxNet and experimented with hyperparameters to find the best performing ones, finding an input pixel size of 5Å and remaining parameters set as default. We excluded any other pre-processing or CTF correction to compare the performance of all methods on exact same inputs. BoxNet allows users to label contamination in the training micrographs, a signal that isn't exploited by either crYOLO or HydraPicker.

Thus we compared against two versions of BoxNet, one which didn't use the contamination label (and hence labelled contamination as background) and another which masked the contamination. We refer to these two variants as BoxNet and BoxNet_mask respectively.

We trained crYOLO with a range of hyperparameters and selected an input resolution of 1024, batch-size of 6, anchor-size of 21, maximum 900 boxes per image, and the remaining parameters set as default.

4.3 Implementation

HydraPicker was implemented using the PyTorch deep learning framework [38]. For optimization, ADAM [39] was used with a cosine annealing scheduler with warm restarts every 40 epochs [40]. Micrographs were randomly rotated and cropped to a resolution of 368×368 and mini-batches of 4 were used where each mini-batch used micrographs from a single dataset only for better representation of smaller datasets during training. To improve training time and convergence, a single-head architecture was first trained as a generic particle picker for 5000 epochs and the best performing model was selected based on the loss on the validation data. Based on our experiments, validation accuracy would not improve significantly after 1200 epochs. However, due to the additional randomness given by the restarts, it was allowed to continue much longer to make sure any additional improvement in accuracy in the second phase of the training would be a result of the multi-head training. The resulting weights were used to initialize the training of the full multi-head architecture. Given that the network should already be in a neighbourhood of the optima, the Adam optimizer and the scheduler would not be much beneficial for this second phase of training. It could even slow down the training. Therefore, training was done only by 100 epochs using a SGD [41] optimizer with a momentum of 0.98. (A similar optimization setting as the first

phase for 1000 epochs was also tried but no significant improvement was found.)

4.4 Evaluation

Like in most detection tasks, particle picking is biased towards rejections rather than detections of true particles. As a consequence, measurements relying on true negatives such as accuracy and specificity are less informative [42]. Instead we propose evaluations based on a metric commonly used in the object detection literature, the precision-recall curve and the area under it, also known as the Average Precision (AP). For completeness, we also plot the ROC (Receiver Operating Characteristic) curve and compute the AUROC (Area Under the ROC curve). Following the literature [9] we use an IOU threshold of 0.6 between picked and true particle boxes to count as positive detection. In case of multiple detections for a single true particle, we select the one with maximum confidence as true positive. Furthermore, we illustrate the qualitative performance of different methods with a few sample micrographs and use green bounding boxes over ground truth locations and red bounding boxes over locations picked by a stated model. As all compared models give confidence probabilities per each picked location, to have less crowded and easier to interpret figures, we only show the bounding boxes with a higher confidence than a threshold. This threshold is chosen once for each method so that it would lead to a precision more than or equal to 0.8 based on the validation set accuracy of the 30 source datasets.

4.5 Generalization vs Specialization

To validate that training was successful we report results on the held-out test micrographs of the 30 source datasets. Two versions of HydraPicker are compared. HydraPicker_gen uses

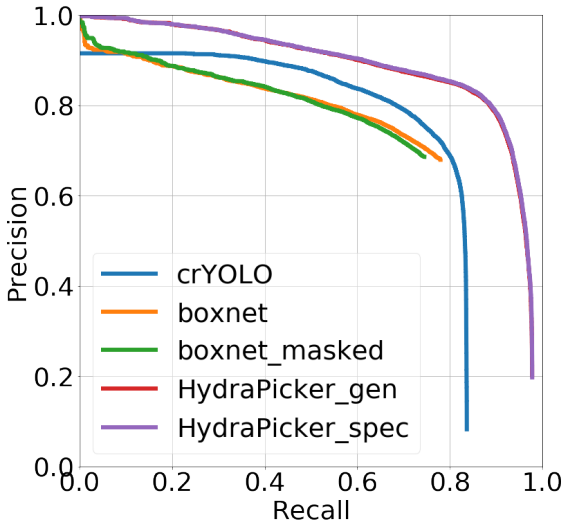

Figure 4.1: Precision-Recall Curves on Test Portions of Source Datasets

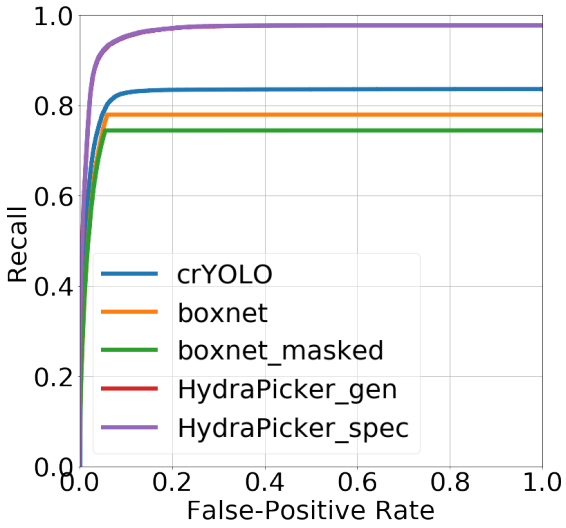

Figure 4.2: ROC curves on test portions of source datasets

Model	AP	AROC
crYOLO	0.718	0.822
BoxNet	0.650	0.766
BoxNet_mask	0.625	0.733
HydraPicker_gen	0.882	0.962
HydraPicker_spec	**0.884**	**0.963**

Table 4.3: Measurements on Test Portions of Source Datasets. Testing by the generic head of HydraPicker is distinguished from testing by the specialized heads for each dataset as ”gen” vs. ”spec”.

the generalization while HydraPicker_spec uses the specialized heads. The results, found in figures 4.1 and 4.2 and table 4.3, show that both versions of HydraPicker significantly outperform the baseline methods. Further, there is only a small improvement with the specialized head over the generalization head which suggests that the generalization head has been very effective in learning indirectly from the data. Finally, we see that there is a strong conservative approach to particle selection which prevents BoxNet from saturating recall. This is likely by design as avoiding bad picks is often considered more important than getting all particles in a micrograph. However, the precision-recall curves show that these methods still suffer from lower accuracy despite this preference.

We also look at the results in a more detailed view, per dataset, for this task (table B.1 in the appendix). We first compare the results using the generalization head versus the specialized heads and assume that a relative difference of 0.005 to be significant for either AP or AROC between the two types of heads. Only 7 out of 30 datasets are significantly different. Five of which (EMPIAR-10084, gk_1, PDB-5foj, PDB-5xwy, and ss_1) have their picking improved using the specialized heads. Two other (PDB-2gtl and PDB-6bcq) were picked better by the generalized head. The fact that the results for the two datasets are worsened is obviously unfavorable. One way to address this issue would be to explicitly consider dataset specific loss terms comparing the generalization head with the specialized heads, instead of a single bias loss term as in Equation 3.4. Moreover, looking at AP among all datasets, it is evident that PDB-5foj has been a hard dataset for this task. While HydraPicker gives an AP in the range of 0.57 and 0.59, the AP for all other methods are in the range of 0.11 to 0.21. The dataset's original particle size is the smallest (8 pixels). Therefore, it could be a result of scaling issues that the performance on this dataset is worse.

To better understand the situation, we draw the Precision-Recall and ROC curves for

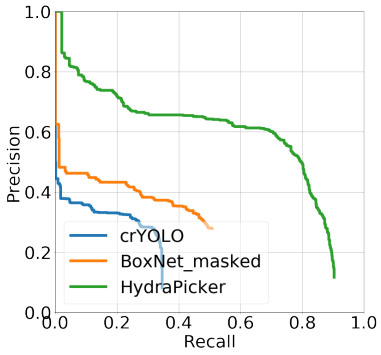

Figure 4.3: Precision-Recall Curves on PDB-5foj

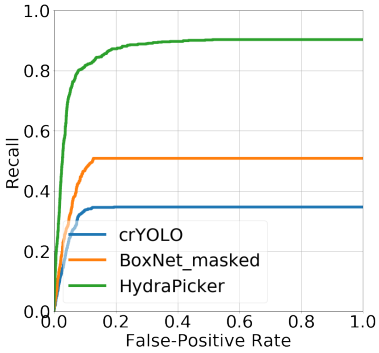

Figure 4.4: ROC Curves on PDB-5foj

Model	AP	AROC
crYOLO	0.116	0.333
BoxNet_mask	0.207	0.486
HydraPicker	**0.586**	**0.872**

Table 4.4: Measurements on PDB-5foj

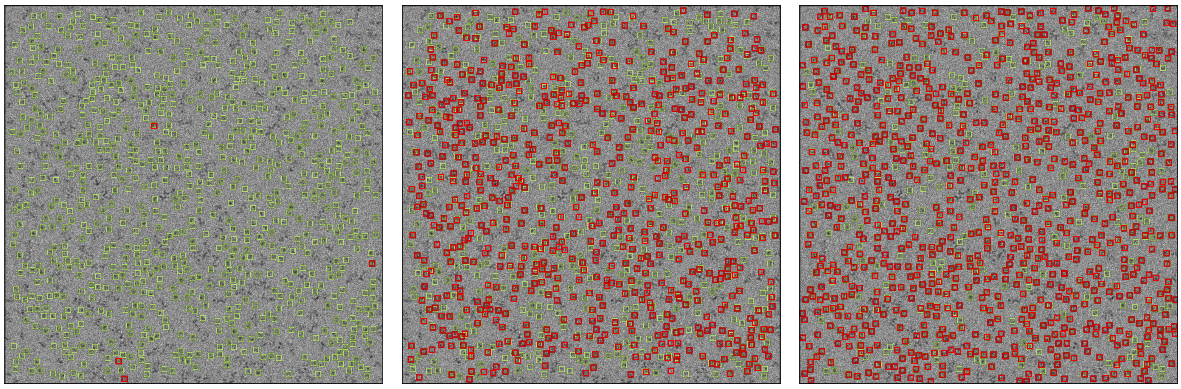

Figure 4.5: A Sample of a Micrograph with Particle Pickings from PDB-5foj. The ground truth is depicted in green. From left to right, the results for crYOLO, BoxNet_mask, and HydraPicker are depicted in red. Here we only show the picks with a confidence higher than a set threshold.

this dataset exclusively. We also illustrate a sample micrograph from its held-out test set and how crYOLO, BoxNet_mask, and HydraPicker pick particles in it (figures 4.3, 4.4, and 4.5 and table 4.4). The reason behind having so few picked particles by crYOLO in figure 4.5 is that we chose the confidence threshold for picked particles based on the aggregated results on the validation set of all 30 datasets. Although BoxNet_mask has picked a fairly large amount of particles as well, too many of them are false positives resulting in a much lower AP and AROC. There are many corrupted particles, which are not marked as ground truth, in the dataset. Such a large number of corrupted particles is unusual among the source datasets and has resulted in poor performance by all methods.

Moreover, for EMPIAR-10078 dataset, crYOLO has performed better than HydraPicker, both in AP and AROc. We look at the exclusive results on this dataset (figures 4.6, 4.7, and 4.8 and table 4.4). It is evident that almost all the particles around the border of the micrographs are marked as ground truth in the dataset. However, they are very similar to ground truth particles and HydraPicker picks most of them, resulting false positives. A post-processing step which filters out detections near the borders could be added as an option to meet the user's preference in such databases.

4.6 Zero-Shot Picking

To test the performance of HydraPicker on previously unseen datasets, we apply the generalization head on the 7 target datasets which were never used in training. This can be thought of as a "zero-shot" learning scenario as the particles in the target datasets are unseen. For comparison, we also trained 7 different single-head HydraPicker models on the training portions of the target datasets and report these under the "7 models" title. Results

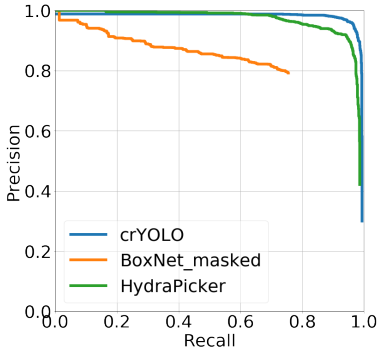

Figure 4.6: Precision-Recall Curves on EMPIAR-10078

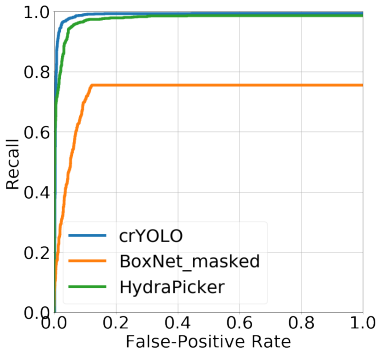

Figure 4.7: ROC Curves on EMPIAR-10078

Model	AP	AROC
crYOLO	0.116	0.333
BoxNet_mask	0.207	0.486
HydraPicker	**0.586**	**0.872**

Table 4.5: Measurements on EMPIAR-10078

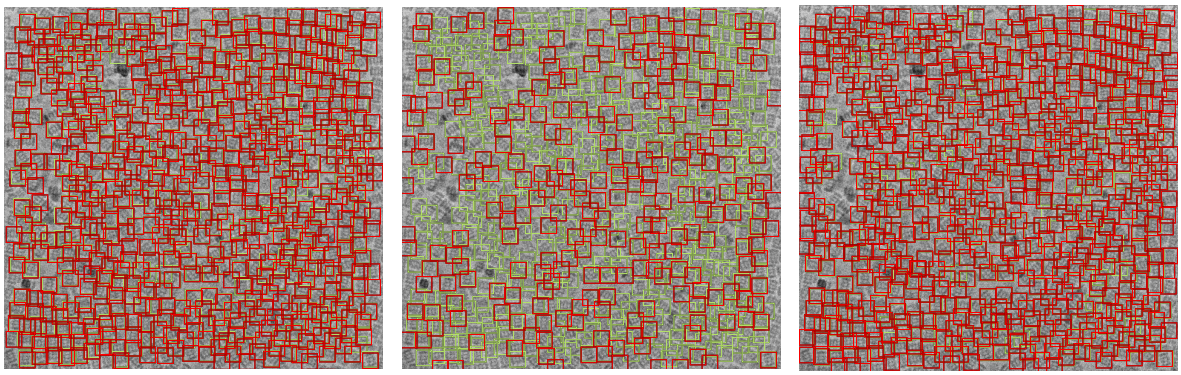

Figure 4.8: A Sample of a Micrograph with Particle Pickings from EMPIAR-10078. The ground truth is depicted in green. From left to right, the results for crYOLO, BoxNet_mask, and HydraPicker are depicted in red. Here we only show the picks with a confidence higher than a set threshold.

can be seen in figures 4.9 and 4.10 and table 4.6. Again, HydraPicker significantly outperformed the baselines on this task. Further, note that HydraPicker's generic picking head performed almost identically to the "7 models" case, *despite never having seen any of the datasets in training*.

Moreover, we look at the per dataset results on this task (table B.2). Compared to crYOLO and both versions of BoxNet, HydraPicker is more accurate for almost all datasets. The only exception is crYOLO on PDB-5w3l.

To further investigate the reason behind HydraPicker's lower precision on this dataset, we illustrate the results and some sample micrographs exclusively from this dataset (figures 4.11, 4.12, and 4.13 and table 4.7). One thing that could be problematic for HydraPicker on this dataset is the existence of many overlapping particles in the ground truth. The reason could be that the hyperparameters like the size of the bounding box are chosen based on the validation subset of source datasets while this situation is less evident in source datasets.

Furthermore, compared to the 7 models directly trained on target datasets, HydraPicker performs slightly better than six of them. Only in PDB-4hhb, HydraPicker performs worse. Although it is absolutely unfair to expect HydraPicker to generalize as well as a model directly trained on a target, we would like to have an explanation. By looking at the target dataset properties (table 4.2) we learn that the original particle size (7 pixels) for this dataset is smaller and outside the range of all 30 source datasets. This would result in more scaling effects in the image than the network is trained to tolerate. As extrapolation is a hard task for neural networks, this behavior is not far from our expectations.

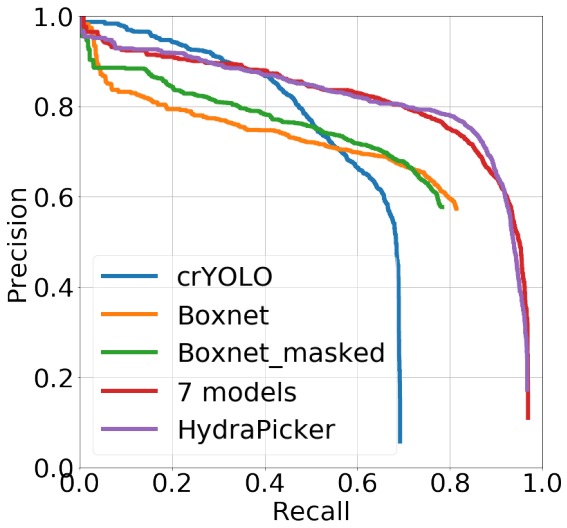

Figure 4.9: Precision-Recall Curves for Zero-Shot Picking

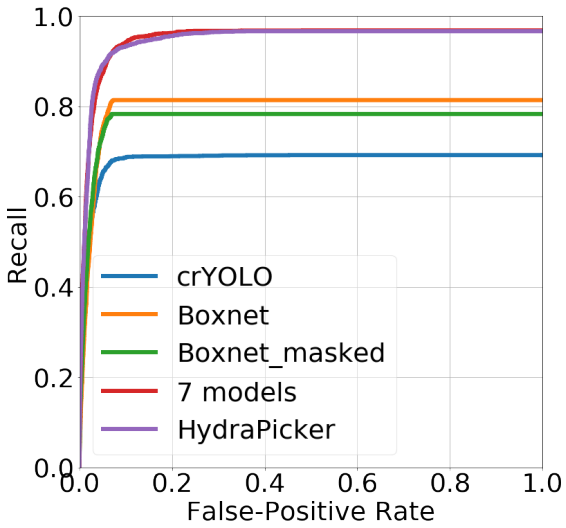

Figure 4.10: ROC Curves for Zero-Shot Picking

Model	AP	AROC
crYOLO	0.584	0.682
BoxNet	0.611	0.797
BoxNet_mask	0.613	0.769
HydraPicker	**0.803**	**0.947**
7 models	0.802	0.949

Table 4.6: Measurements for Zero-Shot Picking. Tests using 7 independent single-head HydraPicker models trained on few micrographs of target datasets are also provided as "7 models".

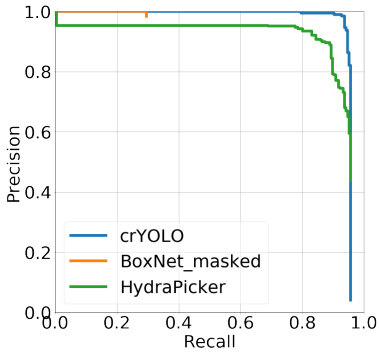

Figure 4.11: Precision-Recall Curves for Zero-Shot Picking on PDB-5w3l

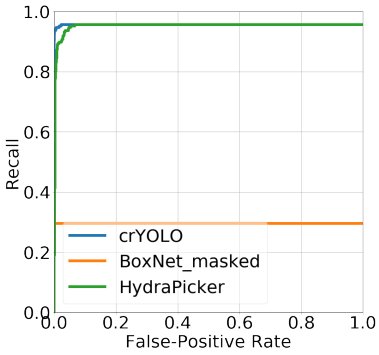

Figure 4.12: ROC Curves for Zero-Shot Picking on PDB-5w3l

Model	AP	AROC
crYOLO	**0.953**	**0.956**
BoxNet_mask	0.296	0.296
HydraPicker	0.894	0.951

Table 4.7: Measurements for Zero-Shot Picking on PDB-5w3l

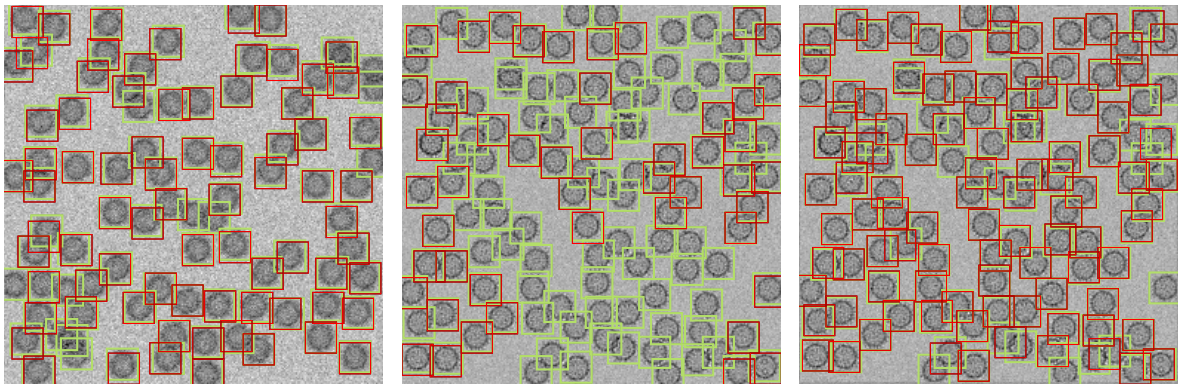

Figure 4.13: A Sample of a Micrograph with Particle Pickings from PDB-5w3l. The ground truth is depicted in green. From left to right, the results for crYOLO, BoxNet_mask, and HydraPicker are depicted in red. Here we only show the picks with a confidence higher than a set threshold.

4.7 Few-Shot Picking

We further explore the case where there is a small amount of training data available for a new dataset. In this case, we train new dataset specific heads for the target datasets using the same training procedure, except we freeze the weights of the body and the generalization head. This can be thought of as a "few-shot" learning scenario as only a small number of particles in the target datasets are used for training.

For other methods we similarly fine-tuned their models using the training data of the target datasets. we tried multiple times both re-training the whole model and . For crYOLO, by testing on the validation set, whole model re-training was chosen as it performed better than fine-tuning on just a few of the last layers. The results are shown in figures 4.14 and 4.15 and table 4.8. HydraPicker performs better in all measurements. It also outperforms the 7 single-head trained models indicating that picking on the target datasets are benefiting from the additional information available from a larger set of source datasets.

Comparing the per dataset results of HydraPicker in the few-shot scenario versus the previous zero-shot scenario (tables B.3 and B.2 in the appendix), we can confirm that improvement is achieved for all datasets and that it has performed better than all the 7 models directly trained on the target datasets. This means regardless of the target structure, if the zero-shot performance of HydraPicker is not good enough for an application, it is possible to improve the results by fine tuning on a few of the target micrographs and to outperform training from scratch on those few micrographs. However, comparing the results for other methods, we notice that their accuracy on three datasets (PDB-5w3l, PDB-5xnl, and PDB-6b44) have generally decreased after fine-tuning. This means for other methods these datasets are hard to fine-tune. According to table 4.2 in two of these datasets, the original particle sizes (37 and 38 pixels) were much larger than the average size among source

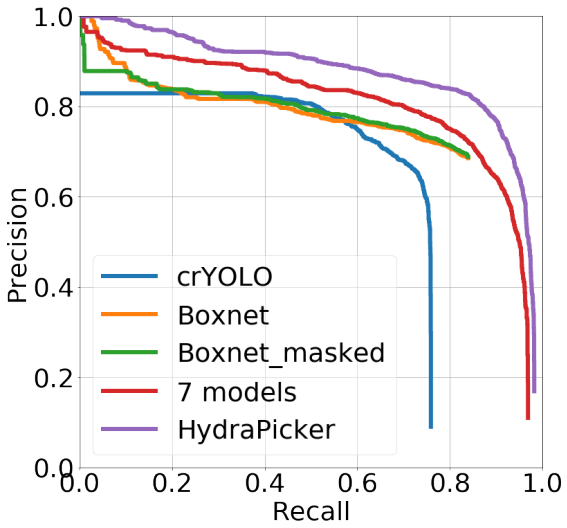

Figure 4.14: Precision-Recall Curves for Few-Shot Picking

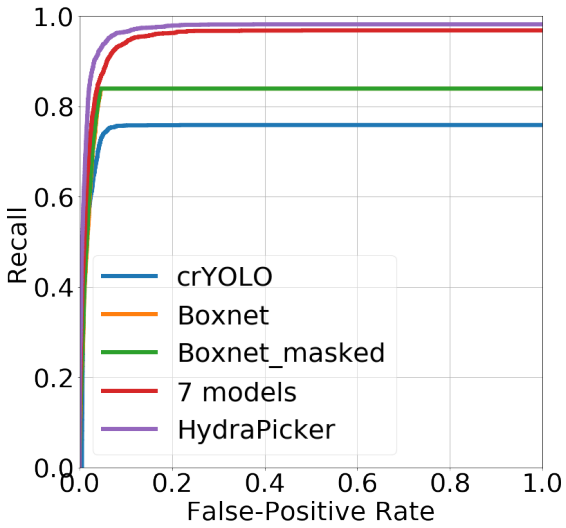

Figure 4.15: ROC Curves for Few-Shot Picking

Model	AP	AROC
crYOLO	0.599	0.746
BoxNet	0.676	0.826
BoxNet_mask	0.676	0.827
HydraPicker	**0.870**	**0.969**
7 models	0.802	0.949

Table 4.8: Measurements for Few-Shot Picking. Tests using seven independent single-head HydraPicker models trained on few micrographs of target datasets are also provided as ”7 models”.

Dataset Access Code	Multi-head AP	Multi-head AROC	Single-head AP	Single-head AROC
PDB-2wri	**0.949**	**0.997**	0.931	0.995
PDB-4hhb	**0.785**	**0.955**	0.775	0.947
PDB-5vy5	**0.733**	**0.93**	0.728	0.924
PDB-5w3l	**0.964**	0.975	0.947	**0.982**
PDB-5xnl	0.986	**0.999**	**0.99**	**0.999**
PDB-6b7n	**0.909**	**0.99**	0.9	0.985
PDB-6b44	0.936	0.982	**0.939**	**0.983**
Average	**0.895**	**0.975**	0.887	0.974

Table 4.9: Measurements for multi-head vs single-head. The access codes indicate the Protein Data Bank (PDB) [37] structure used to simulate the micrographs by [10].

datasets (21 pixels). Since crYOLO fine-tuned worse on both of them, we could assume it cannot tolerate scaling effects when fine-tuning. In appendix A, we have included per dataset results on all 7 datasets for few-shot learning along with sample micrographs.

4.8 Multi-Head vs Single-Head

Finally, we analyze the contribution of multiple heads. As discussed in the beginning of this section, we trained a single-head architecture which we provide a comparison to in this experiment. We fine tune this model on the target datasets and compare it against against the multi-head architecture. To have a more detailed insight on the results, we look at the AP and AROC per target dataset as well as on average. As seen in Table 4.9, in 5 out of 7 target datasets the multi-head model outperforms the single-head model in both AP and AROC measures. The average improvement is small, about 1% in AP, but the results indicate overall that the multi-head model improves over the single head model.

Chapter 5

Conclusion and Future Work

This work has presented HydraPicker, a new method for particle picking in single-particle cryo-EM. The proposed method consists of a customized CNN architecture tailored for the particle picking problem and taking into account the differences of datasets in particle picking data through the use of multiple, dataset specific heads. The architecture is trained using a variation of a focal loss combined with a new term which allows for the training of a general, non-dataset specific head. Beyond a new architecture and training loss, we establish a rigorous testing framework for particle picking methods and compare HydraPicker against state-of-the-art particle picking methods. Our results demonstrate that HydraPicker significantly outperforms existing methods in both zero shot and few shot detection scenarios both in terms of accuracy and consistency among multiple datasets.

In terms of future work, we believe there are several promising directions. First, the formulation used here could also be used to handle the general problem of dataset bias in tasks like recognition, detection, and segmentation. Second, there are a number of modelling decisions which could yield performance improvements. To come up with the proposed HydraPicker, we ran multiple rounds of architecture search. One of the main constraints

that we considered was to avoid usage of more than 11 GB of GPU memory (typical for high-end consumer GPUs at the time) when training the full multi-head model. This limited the flexibility of search over the architecture for detector heads as we needed one copy per dataset in GPU. It also limited the depth of the shared recognition network. Further architecture search without hardware restrictions or even the application of automated search methods [43] is a promising future direction. Third, other choices for ℓ_{bias} may work better. The simple Euclidean norm that we used was effective, but others may yield better performance. In particular, there may yet be room for more improvements in the few-shot case. Fourth, there are a number of other problem specific characteristics which could be used. We excluded the pre-processing and post-processing steps that should further improve the picking accuracy but make objective comparisons more complicated as they may affect each method differently. For instance, explicit handling of the microscope's contrast transfer function can help detection methods generalize over a range of imaging conditions. Fifth, computational and memory complexity and consequently an empirical comparison on the speed of each method should also be considered. However, objective comparison of such matters would require a unified software implementations of all competing methods.

Finally, we believe it would be beneficial to establish a larger collection of datasets and standard testing procedures for particle picking methods. The progress on the problem of particle picking has been unclear, in part because of a lack of consistent and comparable testing methodology. This study attempts to address this in part by establishing a general methodology and directly comparing against previous approaches. However, more work remains to be done by collecting a larger set of datasets and providing a set of consistent and meaningful evaluation metrics. To encourage further comparisons, we will release the code for our method, our comparison methodology and the dataset splits.

References

[1] J. L. Milne, M. J. Borgnia, A. Bartesaghi, E. E. Tran, L. A. Earl, D. M. Schauder, J. Lengyel, J. Pierson, A. Patwardhan, and S. Subramaniam, "Cryo-electron microscopy–a primer for the non-microscopist," *The Federation of European Biochemical Societies (FEBS) Journal*, vol. 280, no. 1, pp. 28–45, 2013.

[2] S. H. Scheres, "Semi-automated selection of cryo-em particles in relion-1.3," *Journal of Structural Biology*, vol. 189, no. 2, pp. 114–122, 2015.

[3] Cellular Structure and 3D Bioimaging Team EMBL-EBI. (2018). EMPIAR-10216, [Online]. Available: `http://dx.doi.org/10.6019/EMPIAR-10216`.

[4] O. von Loeffelholz, G. Papai, R. Danev, A. G. Myasnikov, S. K. Natchiar, I. Hazemann, J. F. Menetret, and B. P. Klaholz, "Volta phase plate data collection facilitates image processing and cryo-em structure determination," *Journal of Structural Biology*, vol. 202, no. 3, pp. 191–199, 2018.

[5] Y. Zhu, Q. Ouyang, and Y. Mao, "A deep convolutional neural network approach to single-particle recognition in cryo-electron microscopy," *BioMed Central (BMC) Bioinformatics*, vol. 18, no. 1, p. 348, 2017.

[6] F. Wang, H. Gong, G. Liu, M. Li, C. Yan, T. Xia, X. Li, and J. Zeng, "Deeppicker: A deep learning approach for fully automated particle picking in cryo-em," *Journal of Structural Biology*, vol. 195, no. 3, pp. 325–336, 2016.

[7] T. Da, J. Ding, L. Yang, and G. Chirikjian, "A method for fully automated particle picking in cryo-electron microscopy based on a cnn," in *ACM Conference on Bioinformatics, Computational Biology, and Health Informatics (ACM BCB)*, 2018.

[8] Y. Xiao and G. Yang, "A fast method for particle picking in cryo-electron micrographs based on fast r-cnn," in *International Conference on Applied Mathematics and Computer Science (ICAMCS)*, 2017.

[9] T. Wagner, F. Merino, M. Stabrin, T. Moriya, C. Antoni, A. Apelbaum, P. Hagel, O. Sitsel, T. Raisch, D. Prumbaum, *et al.*, "Sphire-cryolo is a fast and accurate fully automated particle picker for cryo-em," *Communications Biology*, vol. 2, no. 1, p. 218, 2019.

[10] D. Tegunov and P. Cramer, "Real-time cryo-EM data pre-processing with warp," *bioRxiv*, 2018. DOI: 10.1101/338558.

[11] A. Torralba and A. A. Efros, "Unbiased look at dataset bias," in *CVPR*, 2011.

[12] A. Khosla, T. Zhou, T. Malisiewicz, A. A. Efros, and A. Torralba, "Undoing the damage of dataset bias," in *ECCV*, 2012.

[13] T. Tommasi, N. Patricia, B. Caputo, and T. Tuytelaars, "A deeper look at dataset bias," in *Domain Adaptation in Computer Vision Applications*. 2017, pp. 37–55.

[14] 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.

[15] N. Volkmann, "An approach to automated particle picking from electron micrographs based on reduced representation templates," *Journal of Structural Biology*, vol. 145, no. 1-2, pp. 152–156, 2004.

[16] B. Rath and J. Frank, "Fast automatic particle picking from cryo-electron micrographs using a locally normalized cross-correlation function: A case study," *Journal of Structural Biology*, vol. 145, no. 1-2, pp. 84–90, 2004.

[17] P. U. Adiga, R. Malladi, W. Baxter, and R. M. Glaeser, "A binary segmentation approach for boxing ribosome particles in cryo em micrographs," *Journal of Structural Biology*, vol. 145, no. 1-2, pp. 142–151, 2004.

[18] N. Voss, C. Yoshioka, M. Radermacher, C. Potter, and B. Carragher, "Dog picker and tiltpicker: Software tools to facilitate particle selection in single particle electron microscopy," *Journal of Structural Biology*, vol. 166, no. 2, pp. 205–213, 2009.

[19] J. Vargas, V. Abrishami, R. Marabini, J. de la Rosa-Trevín, A. Zaldivar, J. Carazo, and C. Sorzano, "Particle quality assessment and sorting for automatic and semiautomatic particle-picking techniques," *Journal of Structural Biology*, vol. 183, no. 3, pp. 342–353, 2013.

[20] J. Zhao, M. A. Brubaker, and J. L. Rubinstein, "Tmacs: A hybrid template matching and classification system for partially-automated particle selection," *Journal of Structural Biology*, vol. 181, no. 3, pp. 234–242, 2013.

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

[22] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *ICLR*, 2015.

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

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

[25] J. R. Uijlings, K. E. Van De Sande, T. Gevers, and A. W. Smeulders, "Selective search for object recognition," *IJCV*, vol. 104, no. 2, pp. 154–171, 2013.

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

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

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

[29] 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.

[30] F. Zheng, F. Ni, and L. Zhao, "Localization and recognition of single particle image in microscopy micrographs based on region based convolutional neural networks," in *International Conference of Pioneering Computer Scientists, Engineers and Educators (ICPCSEE)*, 2018.

[31] G. Tang, L. Peng, P. R. Baldwin, D. S. Mann, W. Jiang, I. Rees, and S. J. Ludtke, "Eman2: An extensible image processing suite for electron microscopy," *Journal of Structural Biology*, vol. 157, no. 1, pp. 38–46, 2007.

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

[33] T. Tommasi and T. Tuytelaars, "A testbed for cross-dataset analysis," in *ECCV*, 2014.

[34] Y. Wu and K. He, "Group normalization," in *ECCV*, 2018.

[35] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," *IEEE TPAMI*, 2018.

[36] A. Iudin, P. K. Korir, J. Salavert-Torres, G. J. Kleywegt, and A. Patwardhan, "Empiar: A public archive for raw electron microscopy image data," *Nature Methods*, vol. 13, no. 5, p. 387, 2016.

[37] H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne, "The protein data bank," *Nucleic Acids Research*, vol. 28, no. 1, pp. 235–242, 2000.

[38] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," in *NeurIPS, Workshop on Automatic Differentiation (AutiDiff)*, 2017.

[39] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *ICLR*, 2015.

[40] I. Loshchilov and F. Hutter, "Sgdr: Stochastic gradient descent with warm restarts," *Learning*, vol. 10, p. 3, 2016.

[41] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, "On the importance of initialization and momentum in deep learning," in *ICML*, 2013.

[42] R. Langlois and J. Frank, "A clarification of the terms used in comparing semi-automated particle selection algorithms in cryo-em," *Journal of Structural Biology*, vol. 175, no. 3, pp. 348–352, 2011.

[43] B. Zoph and Q. V. Le, "Neural architecture search with reinforcement learning," in *ICLR*, 2017.

Appendix A

Per Dataset Results for Few-Shot Picking with Sample Micrographs

Here we have provided Precision-Recall curves, Receiver Operating Characteristics (ROC) curves, and their related measurements for HydraPicker, crYOLO, and BoxNet_mask. We only included BoxNet_mask as it performed slightly better than its unmasked variant in a few-shot learning scenario on all 7 target datasets. For each method, We looked at the Precision-Recall curves on the validation set of all the 30 source datasets and chose a confidence threshold for which the precision stays above 0.8. The reason we looked at validation sets of source datasets rather than target datasets is the assumption that the labeled data for few-shot picking is limited and all of it is used for fine-tuning and a threshold based on training set of target datasets should not generalize well to testing sets. In the provided figures we used those thresholds to pick particles from sample micrographs of testing sets.

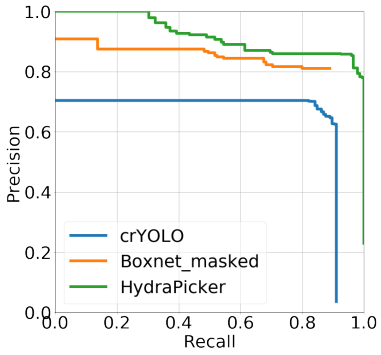

Figure A.1: Precision-Recall Curves for Few-Shot Picking on PDB-2wri

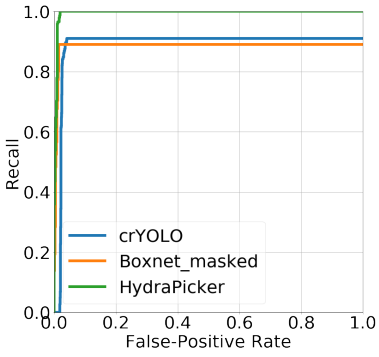

Figure A.2: ROC Curves for Few-Shot Picking on PDB-2wri

Model	AP	AROC
crYOLO	0.638	0.889
BoxNet_mask	0.765	0.884
HydraPicker	**0.921**	**0.995**

Table A.1: Measurements for Few-Shot Picking on PDB-2wri

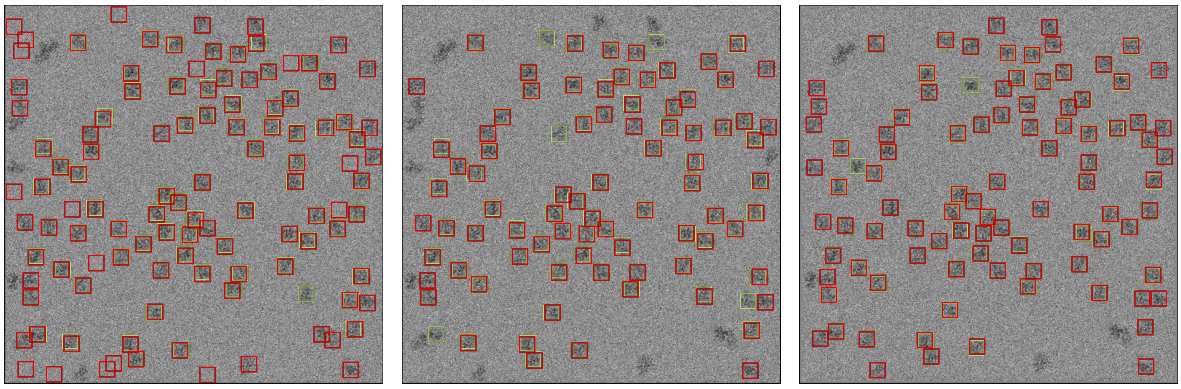

Figure A.3: A Sample of a Micrograph with Particle Pickings from PDB-2wri. The ground truth is depicted in green. From left to right, the results for fine-tuned models of crYOLO, BoxNet_mask, and HydraPicker are depicted in red. Here we only show the picks with a confidence higher than a set threshold.

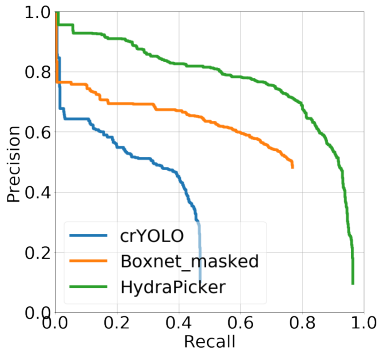

Figure A.4: Precision-Recall Curves for Few-Shot Picking on PDB-4hhb

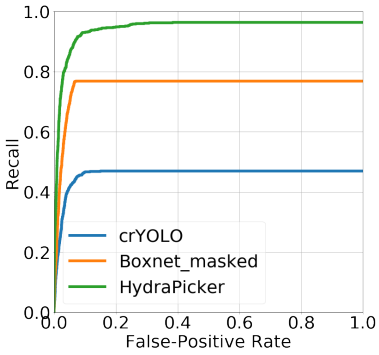

Figure A.5: ROC Curves for Few-Shot Picking on PDB-4hhb

Model	AP	AROC
crYOLO	0.256	0.458
BoxNet_mask	0.506	0.752
HydraPicker	**0.758**	**0.942**

Table A.2: Measurements for Few-Shot Picking on PDB-4hhb

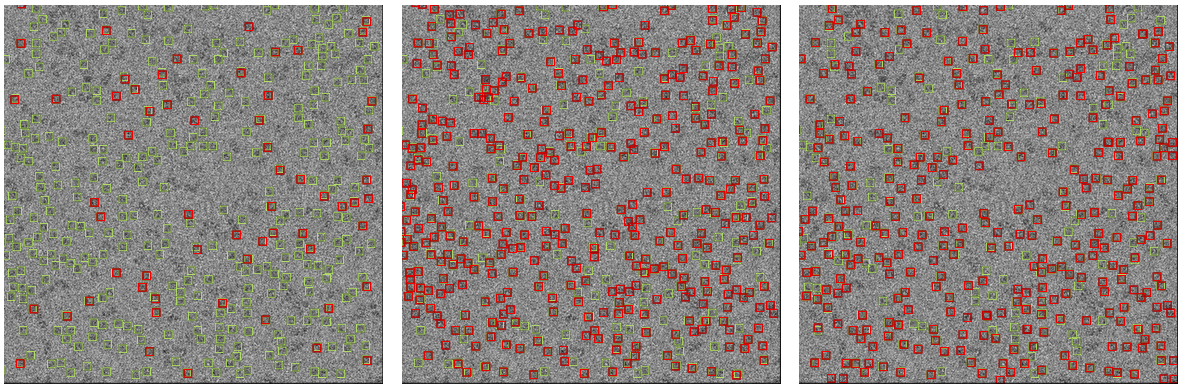

Figure A.6: A Sample of a Micrograph with Particle Pickings from PDB-4hhb. The ground truth is depicted in green. From left to right, the results for fine-tuned models of crYOLO, BoxNet_mask, and HydraPicker are depicted in red. Here we only show the picks with a confidence higher than a set threshold.

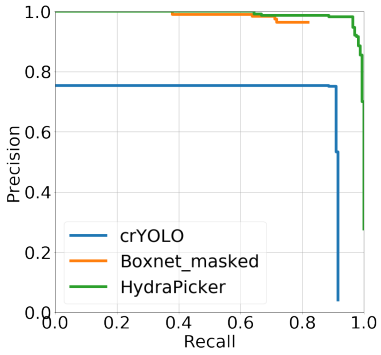

Figure A.7: Precision-Recall Curves for Few-Shot Picking on PDB-5xnl

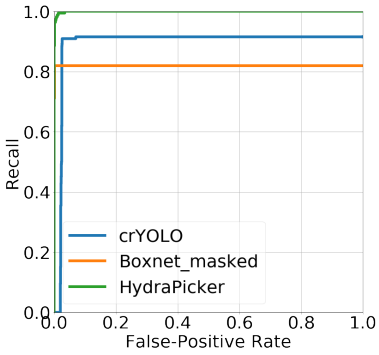

Figure A.8: ROC Curves for Few-Shot Picking on PDB-5xnl

Model	AP	AROC
crYOLO	0.689	0.894
BoxNet_mask	0.812	0.819
HydraPicker	**0.991**	**0.999**

Table A.3: Measurements for Few-Shot Picking on PDB-5xnl

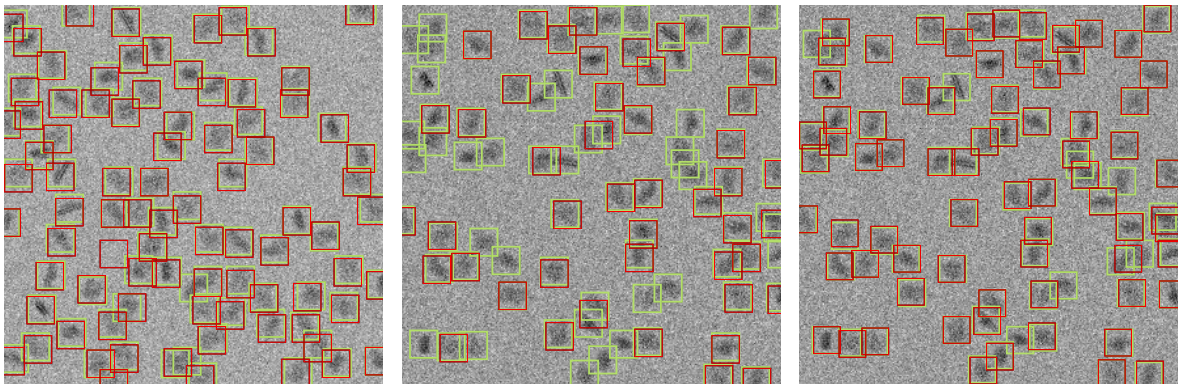

Figure A.9: A Sample of a Micrograph with Particle Pickings from PDB-5xnl. The ground truth is depicted in green. From left to right, the results for fine-tuned models of crYOLO, BoxNet_mask, and HydraPicker are depicted in red. Here we only show the picks with a confidence higher than a set threshold.

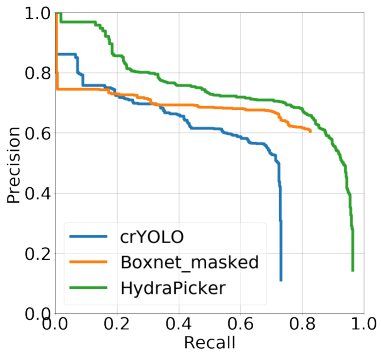

Figure A.10: Precision-Recall Curves for Few-Shot Picking on PDB-5vy5

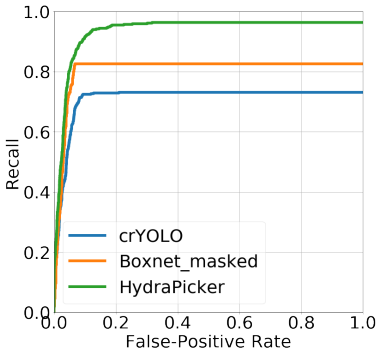

Figure A.11: ROC Curves for Few-Shot Picking on PDB-5vy5

Model	AP	AROC
crYOLO	0.493	0.709
BoxNet_mask	0.576	0.804
HydraPicker	**0.731**	**0.933**

Table A.4: Measurements for Few-Shot Picking on PDB-5vy5

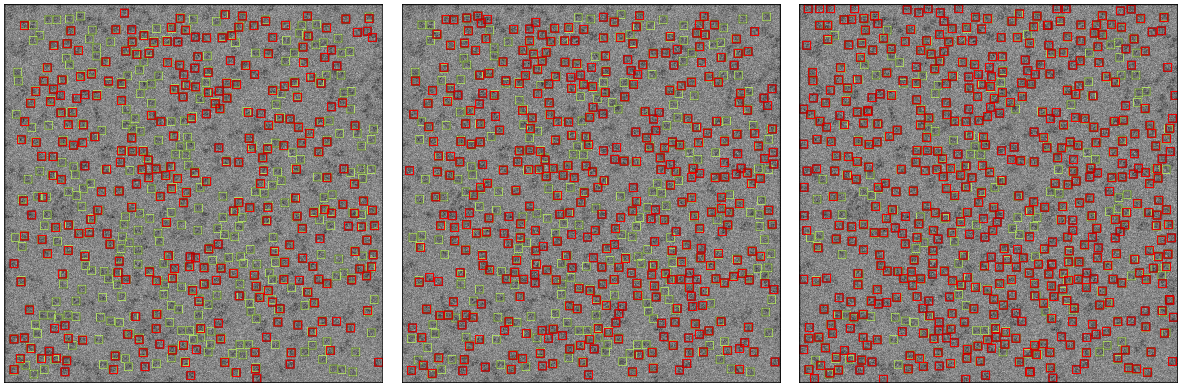

Figure A.12: A Sample of a Micrograph with Particle Pickings from PDB-5vy5. The ground truth is depicted in green. From left to right, the results for fine-tuned models of crYOLO, BoxNet_mask, and HydraPicker are depicted in red. Here we only show the picks with a confidence higher than a set threshold.

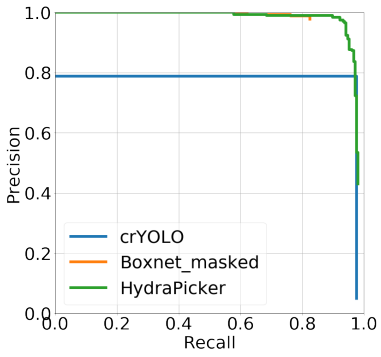

Figure A.13: Precision-Recall Curves for Few-Shot Picking on PDB-5w3l

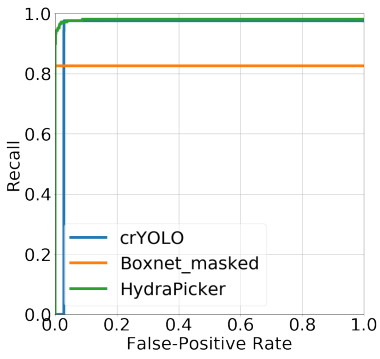

Figure A.14: ROC Curves for Few-Shot Picking on PDB-5w3l

Model	AP	AROC
crYOLO	0.769	0.948
BoxNet_mask	0.824	0.825
HydraPicker	**0.970**	**0.979**

Table A.5: Measurements for Few-Shot Picking on PDB-5w3l

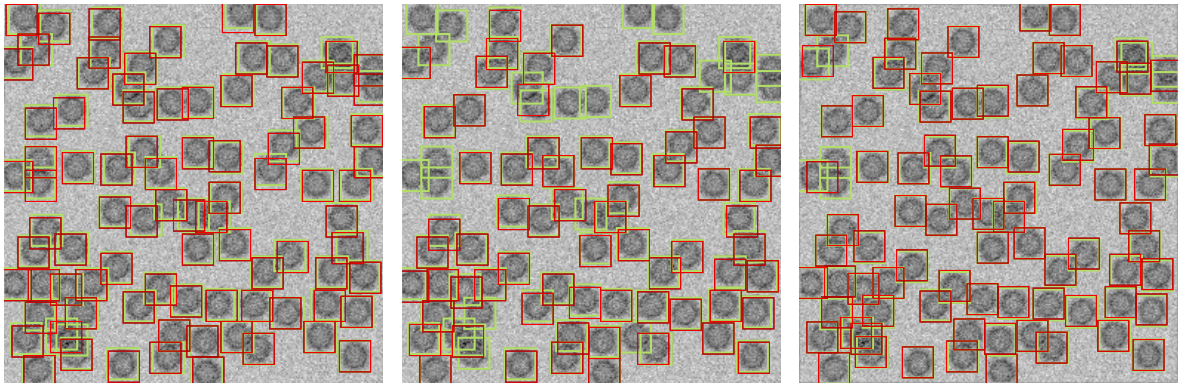

Figure A.15: A Sample of a Micrograph with Particle Pickings from PDB-5w3l. The ground truth is depicted in green. From left to right, the results for crYOLO, BoxNet_mask, and HydraPicker are depicted in red. Here we only show the picks with a confidence higher than a set threshold.

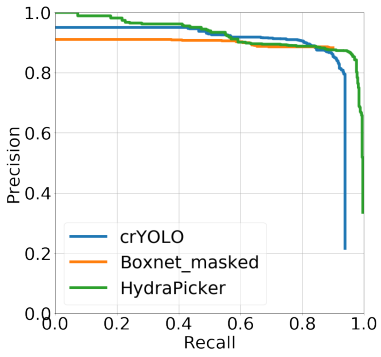

Figure A.16: Precision-Recall Curves for Few-Shot Picking on PDB-6b44

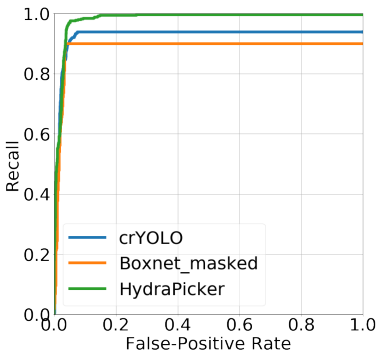

Figure A.17: ROC Curves for Few-Shot Picking on PDB-6b44

Model	AP	AROC
crYOLO	0.870	0.924
BoxNet_mask	0.812	0.884
HydraPicker	**0.926**	**0.979**

Table A.6: Measurements for Few-Shot Picking on PDB-6b44

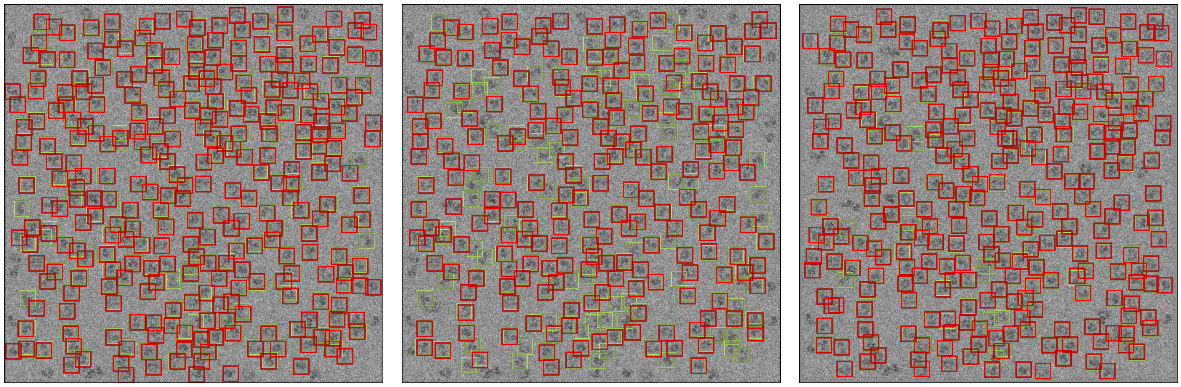

Figure A.18: A Sample of a Micrograph with Particle Pickings from PDB-6b44. The ground truth is depicted in green. From left to right, the results for fine-tuned models of crYOLO, BoxNet_mask, and HydraPicker are depicted in red. Here we only show the picks with a confidence higher than a set threshold.

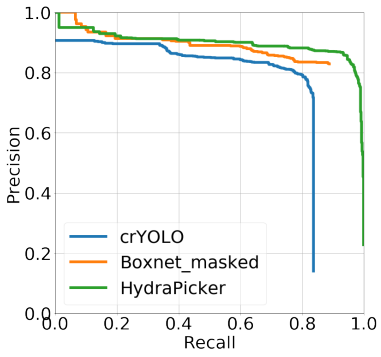

Figure A.19: Precision-Recall Curves for Few-Shot Picking on PDB-6b7n

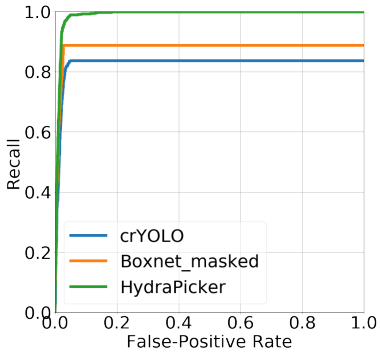

Figure A.20: ROC Curves for Few-Shot Picking on PDB-6b7n

Model	AP	AROC
crYOLO	0.721	0.826
BoxNet_mask	0.797	0.878
HydraPicker	**0.898**	**0.987**

Table A.7: Measurements for Few-Shot Picking on PDB-6b7n

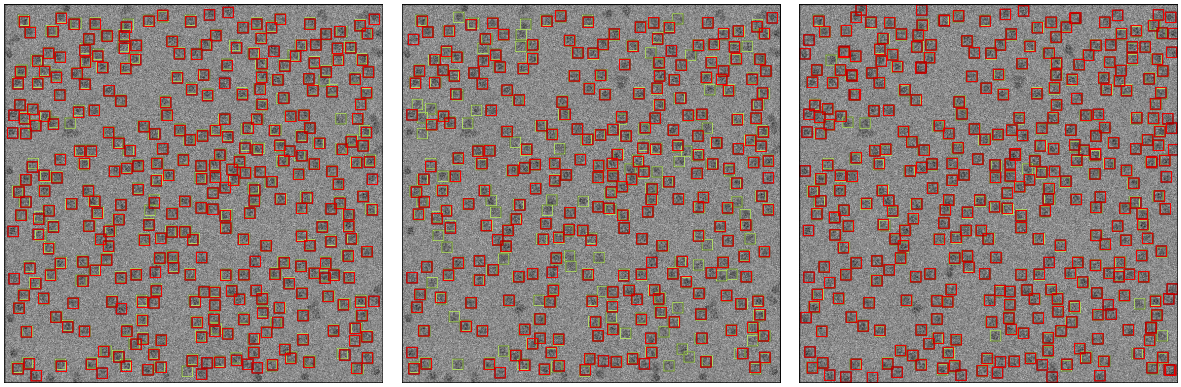

Figure A.21: A Sample of a Micrograph with Particle Pickings from PDB-6b7n. The ground truth is depicted in green. From left to right, the results for fine-tuned models of crYOLO, BoxNet_mask, and HydraPicker are depicted in red. Here we only show the picks with a confidence higher than a set threshold.

Appendix B

Per Dataset Measurements on All Datasets and Methods

Here we have provided per dataset measurements on source datasets and on target datasets, both in zero-shot picking and few-shot picking for all the methods discussed in this study.

Access Code	Gen_AP	Gen_AROC	Spec_AP	Spec_AROC	crYOLO_AP	crYOLO_AROC	BoxNet_AP	BoxNet_AROC	BoxNet_mask_AP	BoxNet_mask_AROC
EMPIAR-10017	*0.93*	**0.986**	**0.932**	**0.986**	0.816	0.883	0.842	0.931	0.809	0.896
EMPIAR-10077	*0.858*	**0.98**	**0.859**	**0.98**	0.714	0.845	0.521	0.721	0.524	0.701
EMPIAR-10078	*0.962*	*0.973*	*0.962*	*0.973*	**0.977**	**0.986**	0.71	0.792	0.666	0.724
EMPIAR-10081	*0.95*	**0.994**	**0.952**	**0.994**	0.945	0.973	0.872	0.89	0.858	0.87
EMPIAR-10084	*0.71*	*0.91*	**0.749**	**0.922**	0.431	0.692	0.558	0.707	0.462	0.615
EMPIAR-10089	**0.98**	**0.995**	**0.98**	**0.995**	0.876	0.927	0.467	0.638	0.461	0.623
EMPIAR-10097	*0.753*	**0.978**	**0.756**	**0.978**	0.623	0.824	0.615	0.863	0.66	0.878
EMPIAR-10122	**0.918**	**0.988**	**0.918**	**0.988**	0.854	0.934	0.72	0.835	0.466	0.631
EMPIAR-10153	**0.973**	**0.998**	**0.973**	**0.998**	0.831	0.948	0.841	0.945	0.818	0.935
EMPIAR-10156	**0.887**	**0.965**	*0.885*	**0.965**	0.794	0.887	0.684	0.859	0.636	0.727
gk_1	*0.846*	*0.976*	**0.853**	**0.977**	0.762	0.908	0.602	0.775	0.57	0.693
hh_2	*0.878*	**0.961**	0.877	**0.961**	**0.882**	0.939	0.695	0.719	0.669	0.687
lf_1	**0.922**	**0.975**	*0.92*	*0.973*	0.8	0.868	0.739	0.828	0.708	0.796
PDB-1sa0	**0.913**	**0.98**	*0.912*	**0.98**	0.681	0.813	0.564	0.68	0.635	0.722
PDB-2gtl	**0.774**	**0.939**	*0.753*	*0.929*	0.389	0.62	0.69	0.828	0.629	0.782
PDB-3j9i	**0.923**	**0.985**	**0.923**	**0.985**	0.878	0.97	0.877	0.933	0.893	0.942
PDB-4zor	*0.99*	**0.992**	**0.991**	**0.992**	0.972	0.975	0.763	0.768	0.523	0.539
PDB-5foj	*0.574*	*0.865*	**0.586**	**0.872**	0.116	0.333	0.136	0.409	0.207	0.486
PDB-5mmi	**1**	**1**	*0.999*	**1**	0.836	0.89	0.712	0.789	0.764	0.796
PDB-5ngm	*0.968*	**0.974**	**0.969**	**0.974**	0.826	0.88	0.825	0.851	0.8	0.817
PDB-5w3s	**0.824**	**0.971**	*0.821*	*0.969*	0.591	0.781	0.783	0.925	0.814	0.951
PDB-5xwy	*0.751*	*0.922*	**0.755**	**0.924**	0.373	0.565	0.53	0.687	0.563	0.708
PDB-5y6p	**0.949**	**0.958**	**0.949**	**0.958**	0.853	0.888	0.605	0.625	0.472	0.472
PDB-6az1	*0.999*	**1**	**1**	**1**	0.914	0.941	0.828	0.844	0.881	0.891
PDB-6bco	*0.839*	*0.979*	**0.84**	**0.981**	0.587	0.793	0.768	0.909	0.706	0.839
PDB-6bcq	**0.879**	**0.892**	*0.872*	*0.888*	0.78	0.81	0.442	0.485	0.493	0.531
PDB-6bcx	*0.963*	**0.988**	**0.964**	**0.988**	0.822	0.864	0.761	0.818	0.722	0.777
PDB-6bhu	**0.851**	**0.952**	**0.851**	**0.952**	0.566	0.724	0.597	0.72	0.622	0.739
PDB-6bqv	*0.878*	**0.952**	**0.88**	**0.952**	0.861	0.944	0.731	0.784	0.739	0.791
ss_1	*0.816*	*0.932*	**0.822**	**0.933**	0.757	0.864	0.509	0.602	0.444	0.519

Table B.1: Per Dataset Measurements on Source Datasets. Results for HydraPicker using the generalized head and the specialized heads are shown separately as “Gen” and “Spec”.

Access Code	HydraPicker_AP	HydraPicker_AROC	crYOLO_AP	crYOLO_AROC	BoxNet_AP	boxnet_AROC	boxnet_mask_AP	boxnet_mask_AROC	7-models_AP	7-models_AROC
PDB-2wri	**0.869**	**0.992**	*0.858*	*0.931*	0.759	0.857	0.651	0.721	0.856	0.991
PDB-4hhb	**0.591**	**0.883**	0.132	0.339	0.382	*0.709*	*0.414*	0.704	0.73	0.929
PDB-5vy5	**0.7**	**0.923**	0.334	0.57	0.491	0.746	*0.555*	*0.791*	0.636	0.885
PDB-5w3l	*0.894*	*0.951*	**0.953**	**0.956**	0.861	0.864	0.296	0.296	0.873	0.949
PDB-5xnl	**0.976**	**0.992**	*0.9*	*0.943*	0.716	0.74	0.827	0.842	0.975	0.998
PDB-6b7n	**0.894**	**0.983**	0.62	0.76	0.741	0.842	*0.806*	*0.875*	0.853	0.963
PDB-6b44	**0.905**	**0.976**	0.844	*0.912*	*0.851*	0.904	0.846	0.903	0.864	0.973

Table B.2: Per Dataset Measurements for Zero-Shot Picking. Tests using seven independent single-head HydraPicker models trained on few micrographs of target datasets are also provided as "7 models".

Access Code	HydraPicker_AP	HydraPicker_AROC	crYOLO_AP	crYOLO_AROC	BoxNet_AP	BoxNet_AROC	BoxNet_mask_AP	BoxNet_mask_AROC	7-models_AP	7-models_AROC
PDB-2wri	**0.921**	**0.995**	0.638	0.889	*0.829*	*0.926*	0.765	0.884	0.856	0.991
PDB-4hhb	**0.758**	**0.942**	0.256	0.458	*0.524*	*0.77*	0.506	0.752	0.73	0.929
PDB-5vy5	**0.731**	**0.933**	0.493	0.709	0.536	0.794	*0.576*	*0.804*	0.636	0.885
PDB-5w3l	**0.97**	**0.979**	0.769	0.948	0.783	0.786	*0.824*	*0.825*	0.873	0.949
PDB-5xnl	**0.991**	**0.999**	0.689	0.894	*0.816*	*0.842*	0.812	0.819	0.975	0.998
PDB-6b7n	**0.898**	**0.987**	0.721	0.826	0.788	0.867	*0.797*	*0.878*	0.853	0.963
PDB-6b44	**0.926**	**0.979**	*0.87*	*0.924*	0.795	0.87	0.812	0.884	0.864	0.973

Table B.3: Per Dataset Measurements for Few-Shot Picking. Tests using seven independent single-head HydraPicker models trained on few micrographs of target datasets are also provided as "7 models".