HydraPicker: Fully Automated Particle Picking in Cryo-EM by Utilizing Dataset Bias in Single Shot Detection

Abbas Masoumzadeh
abbasmz@eecs.yorku.ca
Marcus Brubaker
mab@eecs.yorku.ca

York University
Toronto, Canada

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 paper 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.

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) [13]. This paper presents a novel approach to one of the first computational problems in single particle cryo-EM known as *particle picking*.

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 a set containing 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. 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 shows some sample micrographs, particle images and their corresponding 3D structures to illustrate the problem.

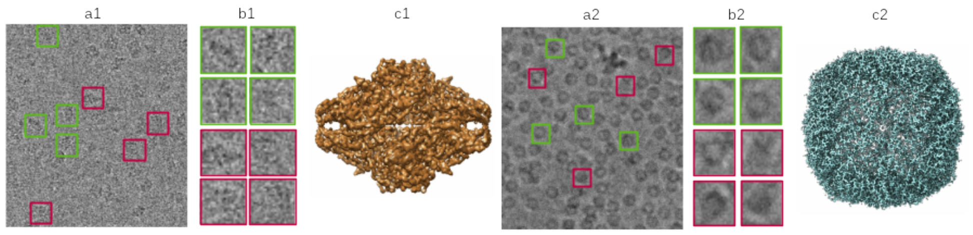

Figure 1: Hardness of particle picking. a) Micrograph patches of Beta-galactosidase (left) and Apoferritin (right). b) Zoomed in boxes of correct particles (green) and most probably corrupted particles, as not picked by practitioners (red). c) 3D reconstruction of the particles. [23, 24].

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 [3, 22, 30, 31, 33, 35]. 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 [6, 25, 26].

In this paper 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 [11]. 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 and consider its performance in both a zero-shot setting 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.

2 Background and Related Work

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 [1, 15, 28, 29] but this limits the usefulness of the approach to effectively known structures. This process is 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 [1] and using difference of Gaussians to detect particles over a specific range of sizes [29]. 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

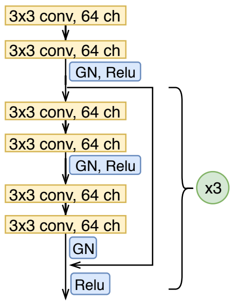

Figure 2: The shared recognition network. It consists of two stacked 3x3 convolutions, followed by three ResNet-like blocks of pairs of two stacked 3x3 convolutions with shortcut connections.

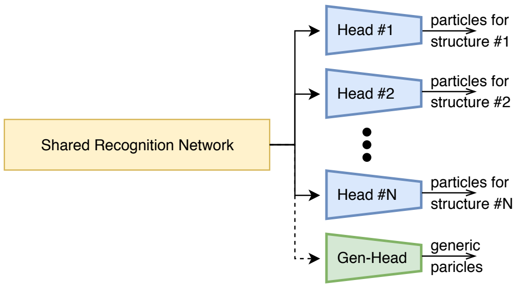
Shared Recognition Network
Head #1
particles for structure #1
Head #2
particles for structure #2
Head #N
particles for structure #N
Gen-Head
generic paricles

Figure 3: The 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.

recall automated method is used to select candidate particles which are shown to the experts to label [18, 27] and in some cases learning from this manual annotation to improve performance [34].

Others have also explored the use of deep learning architectures traditionally used for object recognition to improve the fully automated particle picking task. DeepEM [35] utilized a simple CNN architecture based on AlexNet [8] 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 [31] used a customized VGG-Net [19] 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 preprocessing in subsequent work [3].

Most recently, advances in object detection architectures have been explored in particle picking. Xiao and Yang [33] used the Fast R-CNN [4] architecture with a simplified region proposal method and some of the preprocessing introduced in [3]. 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. SPHIRE-crYOLO [30], customizes the You Only Look Once (YOLO) [16] architecture for particle picking which significantly improves detection 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 [21]. The model was trained on large number of datasets, but when compared against a similar model specially trained for a single target dataset, poorer performance was demonstrated, suggesting that the larger training set was hurting performance. Finally, in an approach called BoxNet, Tegunov and Cramer [22] formulated particle picking as a segmentation problem and used an architecture similar to U-Net [17]. 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.

HydraPicker is most closely related to the detection-based approaches [30, 33] in that a detection-based training framework is used. However, we use a network architecture which

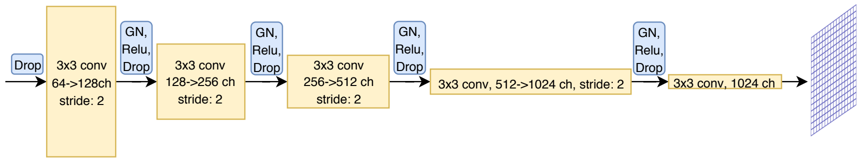
Drop
3x3 conv 64->128ch stride: 2
GN, Relu, Drop
3x3 conv 128->256 ch stride: 2
GN, Relu, Drop
3x3 conv 256->512 ch stride: 2
GN, Relu, Drop
3x3 conv, 512->1024 ch, stride: 2
GN, Relu, Drop
3x3 conv, 1024 ch

Figure 4: The 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.

has been customized to the picking task. Further, unlike existing approaches, we explicitly represent the fact that particle detections used in training come from multiple, different datasets corresponding to different structures. The specific approach is motivated by the work on dataset bias [26] as 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.

Beyond a novel technical approach, we further provide a detailed evaluation of our method on a range of datasets [22] and in realistic scenarios to evaluate 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.

3 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 on datasets without labelled data.

3.1 Network Architecture

For the architecture of the body, we construct a variation of the ResNet architecture [5] which utilizes residual connections to improve training. However, the basic ResNet architecture was designed for images which have significantly different characteristics than 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 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 remove 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 [32] which divide channels into groups and normalize them separately. The complete architecture is shown in Figure 2.

The body forms a common feature extractor for multiple detection heads whose design (Figure 4) is derived from the single shot detection (SSD) framework [11]. SSD, like some previous approaches [16] operates by predicting detections and bounding boxes at a grid of anchor points. The network is trained using a focal classification loss [10]. This combination of SSD and the focal classification loss (called RetinaNet) has achieved state of the art performance for detection on natural images [10].

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 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, ReLU, 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. For instance, smaller datasets and less likely imaging conditions are likely to be ignored as the amount of available data grow. Instead, HydraPicker uses a different head for each dataset which is specialized to that dataset, plus an additional head which generalizes picking on unseen datasets, *i.e.*, it is used for picking particles with no corresponding training data. Inspired by [6], 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

$$\ell_{\text{Hydra}} = \ell_{\text{loc}} + \lambda_{\text{cls}} \ell_{\text{cls}} + \lambda_{\text{bias}} \ell_{\text{bias}} \qquad (1)$$

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

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:

$$\ell_{\text{loc}}(\delta\mathbf{p},\mathbf{p}) = \sum_{(i,j)\in M} ||(\mathbf{a}_i + \delta\mathbf{p}_i) - \mathbf{p}_j||_1 \tag{2}$$

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 [30].

For the classification loss, we use the focal classification loss [10]. Specifically,

$$\ell_{\text{cls}} = \sum_i -\frac{1}{2}\alpha_{c_i}(1-p_i)^\gamma \log p_i \tag{3}$$

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

The classification and localization losses are evaluated using only the assigned head for samples from each dataset, enabling both the shared body and the dataset specific heads to learn. To enable learning of the generalization head, as training samples have no direct effect on it, 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,

$$\ell_{\text{bias}}(W_{S_1},\ldots,W_{S_N},W_G) = \frac{1}{N}\sum_{k=1}^{N} ||W_{S_k} - W_G||_2 \tag{4}$$

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.

4 Experiments

HydraPicker was implemented using the PyTorch deep learning framework [14]. For optimization, ADAM [7] was used with a cosine annealing scheduler with warm restarts every 40 epochs [12]. In order to have more representative augmentations during training, micrographs were padded to be at least 520×520 pixels using Gaussian noise with its mean and standard deviation chosen separately based on the background regions for each micrograph. 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.To improve training time and convergence, a single-head architecture was first trained as a generic particle picker for 5000 epochs and its weights were used to initialize the training of the full multi-head architecture. Its training was done for 100 epochs using a SGD [20] optimizer with momentum. The best performing model was selected based on the loss on the validation data. Below we discuss the datasets, evaluation and some of our results. For more results, please see the Supplemental Material.

Datasets To evaluate HydraPicker, we made use of a collection of 37 datasets collected by Tegunov and Cramer [22]. These datasets are a mix of real data which has been annotated and synthetic datasets from real structures. 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. 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 and 7 "target" datasets. This split is used to test the performance of the methods on previously unseen target datasets.

Baselines We compare our results on these datasets against two state-of-the-art learning-based particle picking methods, BoxNet [22] and crYOLO [30]. For both methods we used their latest release but to ensure adherence to the same experimental setup as for HydraPicker, we had to retrain them from scratch. Because the provided models had training sets that overlapped with our held-out test and validation micrographs and target datasets.

We experimented with hyperparameters of BoxNet to find the best performing ones, finding an input pixel size of 5Å and remaining parameters set as default. No other pre-processing or CTF correction was done. 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.

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 [9]. 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 [30] 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.

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 the generalization while HydraPicker_spec uses the specialized heads. The results, found in Figures 5 and 6 and Table 1, 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

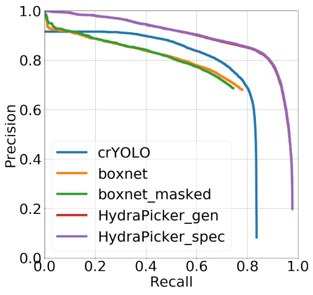

Figure 5: Precision-Recall curves on test portions of source datasets

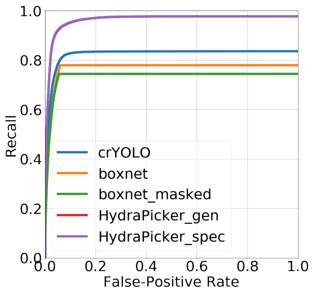

Figure 6: ROC curves on test portions of source datasets

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 1: 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".

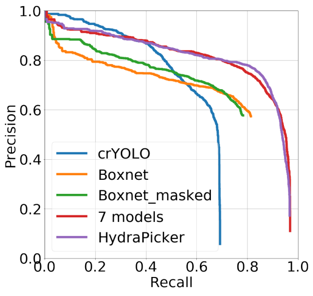

Figure 7: Precision-Recall curves for zero-shot picking

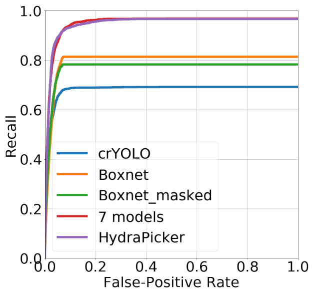

Figure 8: ROC curves for zero-shot picking

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 2: 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".

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 them 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.

Zero-Shot Picking To test the performance of HydraPicker on previously unseen datasets, we apply the generalization head on the seven 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 seven different single-head HydraPicker models on the training portions of the target datasets and report these under the “7 models” title. Results can be seen in Figures 7 and 8 and Table 2. 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*. Note that similarly for real use case in CryoEM, data labeling is time consuming and unfavorable for the practitioners. Therefore, the total size of the training subset of the target datasets is smaller than that of the source datasets.

Few-Shot Picking We also 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

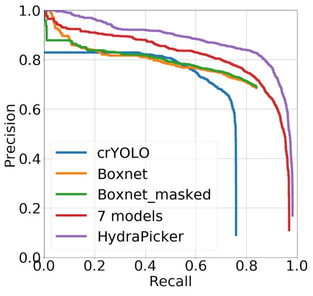

Figure 9: Precision-Recall curves for few-shot picking

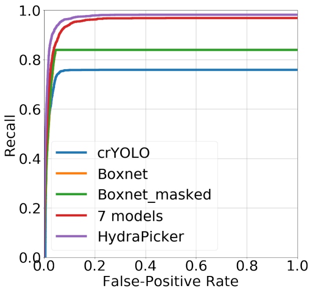

Figure 10: ROC curves for few-shot picking

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 3: Measurements for few-shot picking. Tests using 7 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: Measurements for multi-head vs single-head. The access codes indicate the Protein Data Bank (PDB) [2] structure used to simulate the micrographs by [22].

a small number of particles in the target datasets are used for training. For BoxNet and crYOLO we similarly fine-tuned their models using the training data of the target datasets. The results are shown in Figures 9 and 10 and Table 3. Again, 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.

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, 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.

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.

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 other tasks like recognition, detection, and segmentation. Second, there are a number of modelling decisions which could yield performance improvements. For instance, further architecture search or even the application of automated search methods [36] is a promising direction. In addition, 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. Third, there are a number of other problem specific characteristics which could be used. For instance, explicit handling of the microscope's contrast transfer function can help detection methods generalize over a range of imaging conditions.

Finally, we believe it would be beneficial to establish larger collections 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 paper 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 make the code for our method, our comparison methodology and the dataset splits available.

Acknowledgements This research was undertaken as part of the Vision: Science to Applications program, thanks in part to funding from the Canada First Research Excellence Fund.

References

[1] PS Umesh Adiga, Ravi Malladi, William Baxter, and Robert M Glaeser. A binary segmentation approach for boxing ribosome particles in cryo em micrographs. *Journal of Structural Biology*, 145(1-2):142–151, 2004.

[2] Helen M Berman, John Westbrook, Zukang Feng, Gary Gilliland, Talapady N Bhat, Helge Weissig, Ilya N Shindyalov, and Philip E Bourne. The protein data bank. *Nucleic Acids Research*, 28(1):235–242, 2000.

[3] Ting Da, Jianzhong Ding, Liang Yang, and Gregory 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.

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

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

[6] Aditya Khosla, Tinghui Zhou, Tomasz Malisiewicz, Alexei A Efros, and Antonio Torralba. Undoing the damage of dataset bias. In *ECCV*, 2012.

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

[8] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NeurIPS*, 2012.

[9] Robert Langlois and Joachim Frank. A clarification of the terms used in comparing semi-automated particle selection algorithms in cryo-em. *Journal of structural biology*, 175(3):348–352, 2011.

[10] Tsung-Yi Lin, Priyal Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. *IEEE TPAMI*, 2018.

[11] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, C. Y. Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *ECCV*, 2016.

[12] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *Learning*, 10:3, 2016.

[13] Jacqueline LS Milne, Mario J Borgnia, Alberto Bartesaghi, Erin EH Tran, Lesley A Earl, David M Schauder, Jeffrey Lengyel, Jason Pierson, Ardan Patwardhan, and Sriram Subramaniam. Cryo-electron microscopy–a primer for the non-microscopist. *The Federation of European Biochemical Societies (FEBS) Journal*, 280(1):28–45, 2013.

[14] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *NeurIPS, Workshop on Automatic Differentiation (AutiDiff)*, 2017.

[15] BK 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*, 145(1-2):84–90, 2004.

[16] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *CVPR*, 2016.

[17] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 2015.

[18] Sjors HW Scheres. Semi-automated selection of cryo-em particles in relion-1.3. *Journal of Structural Biology*, 189(2):114–122, 2015.

[19] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

[20] Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In *ICML*, 2013.

[21] Guang Tang, Liwei Peng, Philip R Baldwin, Deepinder S Mann, Wen Jiang, Ian Rees, and Steven J Ludtke. Eman2: an extensible image processing suite for electron microscopy. *Journal of Structural Biology*, 157(1):38–46, 2007.

[22] Dimitry Tegunov and Patrick Cramer. Real-time cryo-EM data pre-processing with warp. *bioRxiv*, 2018. doi: 10.1101/338558.

[23] Cellular Structure and 3D Bioimaging Team EMBL-EBI. EMPIAR-10017, 2015. URL http://dx.doi.org/10.6019/EMPIAR-10017.

[24] Cellular Structure and 3D Bioimaging Team EMBL-EBI. EMPIAR-10216, 2018. URL http://dx.doi.org/10.6019/EMPIAR-10216.

[25] Tatiana Tommasi, Novi Patricia, Barbara Caputo, and Tinne Tuytelaars. *A Deeper Look at Dataset Bias*, pages 37–55. 2017.

[26] Antonio Torralba and Alexei A Efros. Unbiased look at dataset bias. In *CVPR*, 2011.

[27] J. Vargas, V. Abrishami, R. Marabini, JM de la Rosa-Trevín, A. Zaldivar, JM. Carazo, and COS Sorzano. Particle quality assessment and sorting for automatic and semiautomatic particle-picking techniques. *Journal of Structural Biology*, 183(3):342–353, 2013.

[28] Niels Volkmann. An approach to automated particle picking from electron micrographs based on reduced representation templates. *Journal of Structural Biology*, 145(1-2): 152–156, 2004.

[29] NR Voss, CK Yoshioka, M. Radermacher, CS Potter, and B. Carragher. Dog picker and tiltpicker: software tools to facilitate particle selection in single particle electron microscopy. *Journal of Structural Biology*, 166(2):205–213, 2009.

[30] Thorsten Wagner, Felipe Merino, Markus Stabrin, Toshio Moriya, Claudia Antoni, Amir Apelbaum, Philine Hagel, Oleg Sitsel, Tobias Raisch, Daniel Prumbaum, et al. Sphire-cryolo is a fast and accurate fully automated particle picker for cryo-em. *Communications Biology*, 2(1):218, 2019.

[31] Feng Wang, Huichao Gong, Gaochao Liu, Meijing Li, Chuangye Yan, Tian Xia, Xueming Li, and Jianyang Zeng. Deeppicker: A deep learning approach for fully automated particle picking in cryo-em. *Journal of Structural Biology*, 195(3):325–336, 2016.

[32] Yuxin Wu and Kaiming He. Group normalization. In *ECCV*, 2018.

[33] Yifan Xiao and Guangwen 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.

[34] Jianhua Zhao, Marcus A Brubaker, and John L Rubinstein. Tmacs: A hybrid template matching and classification system for partially-automated particle selection. *Journal of Structural Biology*, 181(3):234–242, 2013.

[35] Yanan Zhu, Qi Ouyang, and Youdong Mao. A deep convolutional neural network approach to single-particle recognition in cryo-electron microscopy. *BMC Bioinformatics*, 18(1):348, 2017.

[36] Barret Zoph and Quoc V. Le. Neural architecture search with reinforcement learning. In *ICLR*, 2017.