Prime Sample Attention in Object Detection

Yuhang Cao1  Kai Chen1  Chen Change Loy2  Dahua Lin1

1CUHK - SenseTime Joint Lab, The Chinese University of Hong Kong
2Nanyang Technological University
{yhcao6,chenkaidev}@gmail.com  ccloy@ntu.edu.sg  dhlin@ie.cuhk.edu.hk

Abstract

It is a common paradigm in object detection frameworks to treat all samples equally and target at maximizing the performance on average. In this work, we revisit this paradigm through a careful study on how different samples contribute to the overall performance measured in terms of mAP. Our study suggests that the samples in each mini-batch are neither independent nor equally important, and therefore a better classifier on average does not necessarily result in higher mAP. Motivated by this study, we propose the notion of Prime Samples, those that play a key role in driving the detection performance. We further develop a simple yet effective sampling and learning strategy called PrIme Sample Attention (PISA) that directs the focus of the training process towards such samples. Our experiments demonstrate that it is often more effective to focus on prime samples than hard samples when training a detector.

1. Introduction

Modern object detection frameworks, including both single-stage [17, 15] and two-stage [8, 7, 20], usually adopt a region-based approach, where a detector is trained to classify and localize sampled regions. Therefore, the choice of region samples is critical to the success of an object detector. In practice, most of the samples are located in the background areas. Hence, simply feeding all the samples, or a random subset thereof, through a network and optimizing the average loss is obviously not a very effective strategy.

Recent studies [17, 21, 15] showed that focusing on difficult samples is an effective way to boost the performance of an object detector. A number of methods have been developed to implement this idea in various ways. Representative methods along this line include OHEM [21] and Focal Loss [15]. The former explicitly selects hard samples, i.e., those with high loss values; while the latter uses a reshaped loss function to reweight the samples, emphasizing the difficult ones.

Though simple and widely adopted, random sampling or hard mining are not necessarily the optimal sampling strategy in terms of training an effective detector. Particularly, a question remains open – what are the most important samples for training an object detector. In this work, we carry out a study on this issue with an aim to find a more effective way to sample/weight regions.

Our study reveals two significant aspects that need to be taken into consideration when designing a sampling strategy: (1) Samples should not be treated as independent nor as equally important. Region-based object detection is to select a small subset of bounding boxes out of a large number of candidates to cover all objects in an image. Hence, the decisions on different samples are competing with each other, instead of being independent (like in a classification task). In general, it is more advisable for a detector to yield high scores on one bounding box around each object while...
ensuring all objects of interest are sufficiently covered, instead of trying to produce high scores for all positive samples, i.e. those that substantially overlap with objects. Particularly, our study shows that focusing on those positive samples with the highest IoUs against the ground-truth objects is an effective way towards this goal. (2) The objectives of classification and localization are correlated. The observation that those samples that are precisely located around ground-truth objects are particularly important has a strong implication, that is, the objective of classification is closely related to that of localization. In particular, well-located samples need to be well classified with high confidences.

Inspired by the study, we propose Prime Sample Attention (PISA), a simple yet effective method to sample regions and learn object detectors, where we refer to those samples that play a more important role in achieving high detection performance as the prime samples. We define Hierarchical Local Rank (HLR) as a metric of importance. Specifically, we use IoU-HLR to rank positive samples and Score-HLR to rank negative samples in each mini-batch. This ranking strategy places positive samples with the highest IoUs around each object and negative samples with the highest scores in each cluster to the top of the ranked list, and directs the focus of the training process to them via a simple reweighting scheme. We also devise a classification-aware regression loss to jointly optimize the classification and regression branches. Particularly, this loss would suppress those samples with large regression loss, thus reinforcing attention to the prime samples.

We tested PISA with both two-stage and single-stage detection frameworks. On the MSCOCO [16] test-dev, with a strong backbone of ResNet-101-32x4d, PISA improves Faster R-CNN [20], Mask R-CNN [9], and RetinaNet [15] by 2.0%, 1.5%, 1.8%, respectively. For SSD, PISA achieves a gain of 2.1%.

Our main contributions mainly lie in three aspects: (1) Our study leads to a new insight into what samples are important for training an object detector, thus establishing the notion of prime samples. (2) We devise Hierarchical Local Rank (HLR) to rank the importance of samples, and on top of that an importance-based reweighting scheme. (3) We introduce a new loss called classification-aware regression loss that jointly optimizes both the classification and regression branches, which further reinforces attention to the prime samples.

2. Related Work

Region-based object detectors. Region-based object detectors transform the task of object detection into a bounding box classification and regression problem. Contemporary approaches mostly fall into two categories, i.e., the two-stage and single-stage detection paradigms. Two-stage detectors such as R-CNN [8], Fast R-CNN [7] and Faster R-CNN [20] first generate a set of candidate proposals, then randomly sample a small batch of proposals from all the candidates. These proposals are classified into foreground classes or background, and their locations are refined by regression. There are also some recent improvements [5, 14, 9, 11, 1, 3, 3] along this paradigm. In contrast, single-stage detectors like SSD [17] and RetinaNet [15] directly predict class scores and box offsets from anchors, without the region proposal step. Other variants include [25, 13, 26, 27]. The proposed PISA is not designed for any specific detectors but can be easily applied to both paradigms.

Sampling strategies in object detection. The most widely adopted sampling scheme in object detection is random sampling, that is, to randomly select some samples from all candidates. Since negative samples are usually much more than positive ones, a fixed ratio may be set for positive and negative samples during sampling, like in [7, 20]. Another popular idea is to sample hard samples that have larger losses; this strategy can lead to better optimization for classifiers. The principle of hard mining is proposed in early detection work [23, 6] and also adopted by more recent methods [17, 8, 21] in the deep learning era. Libra R-CNN [18] proposes IoU-balanced Sampling as an approximation of hard negative mining. Focal Loss [15] applies different loss weights to samples, which can be seen as a soft version of sampling. GHM [13] further improves FL by down-weighting the gradient contribution of outliers. AP loss [2] and DR loss [19] introduce a new perspective that converts the classification task into a ranking task. However, the goal of hard mining and ranking loss is to boost the average performance of a classifier and alleviate the imbalance of training samples; they do not investigate the difference between detection and classification. Different from that, PISA can achieve a biased performance on different samples. According to our study in Sec. 3, we find that prime samples are not necessarily the hard ones, which is opposite to hard mining.

Relation between samples Unlike conventional detectors that predict all samples independently, He et al. [11] propose an attention module adapted from the natural language processing field to model relations between objects. Though it is effective, all samples are still treated equally and relations are learned implicitly, without understanding what exactly the relations are. In PISA, samples are attended differently according to their importance.

Improvement of NMS with localization confidence IoU-Net [12] proposes to use localization confidence instead of classification scores for NMS. It adds an extra branch to predict the IoU of samples and use the localization confidence, i.e., predicted IoU, for NMS. There are some major differences between IoU-Net and our method. First, IoU-
Net aims to yield higher scores for proposals with higher predicted IoUs. In this work, we find that a high IoU does not necessarily mean being important for training. Particularly, the relative ranking among proposals around the objects also play a crucial role. Second, our goal is not to improve the NMS and we do not exploit an additional branch to predict the localization confidence, but investigate the sample importance and propose to pay more attention to prime samples with the importance-based reweighting, as well as a new loss to correlate the training of two branches.

3. Prime Samples

In this section, we introduce the concept of Prime Samples, namely those that have greater influence on the performance of object detection. Specifically, we carry out a study on the importance of different samples by revisiting how they affect mAP, the major performance metric for object detection. Our study shows that the importance of each sample depends on how its IoU or score compares to that of the others that overlap with the same object. Therefore, we propose HLR (IoU-HLR and Score-HLR), a new ranking strategy, as a quantitative way to assess the importance.

A Revisit of mAP. mAP is a widely adopted metric for assessing the performance of an object detector, computed as follows. Given an image with annotated ground-truths, each bounding box will be marked as true positive (TP) when: (i) the IoU between this bounding box and its nearest ground truth is greater than a threshold $\theta$, and (ii) there are no other boxes with higher scores that is also a TP of the same ground truth. All other bounding boxes are considered as false positives (FP). Then, the recall is defined as the fraction of ground-truths that are covered by TPs, and the precision is defined as the fraction of resulted bounding boxes that are TPs. On a testing dataset, one can obtain a precision-recall curve by varying the threshold $\theta$, usually ranging from 0.5 to 0.95, and compute the average precision (AP) for each class as the area under the curve. Then mAP is defined as the mean of the AP values over all classes.

The way mAP works reveals two criteria on which positive samples are more important for an object detector. (1) Among all bounding boxes that overlap with a ground-truth object, the one with the highest IoU is the most important as its IoU value directly influences the recall. (2) Across all bounding boxes with the highest IoUs for different objects, the ones with higher IoUs are more important, because they are the last ones that fall below the IoU threshold $\theta$ as $\theta$ increases and thus have great impact on the overall precision.

A Revisit of False Positives. One of the main sources of false positives is misclassifying negative samples as positive, such misclassification is harmful to the precision and will decrease the mAP. However, not all misclassified samples have direct influence on the final results. During inference, if there are multiple negative samples that heavily overlap with each other, only the one with the highest score remains while others are discarded after Non-Maximum Suppression (NMS). In this way, if a negative sample is close to another one with higher score, then the negative sample becomes less important even if its score may also be high because it will not be kept in the final results. We can learn which negative samples are important. (1) Among all negative samples within a local region, the one with the highest score is the most important. (2) Across all highest-score samples in different regions, the ones with higher scores are more important, because they are the first ones that decrease the precision.

Hierarchical Local Rank (HLR). Based on the analysis above, we propose IoU Hierarchical Local Rank (IoU-HLR) and Score Hierarchical Local Rank (Score-HLR) to rank the importance of positive and negative samples in a mini-batch. This rank is computed in a hierarchical manner, which reflects the relation both locally (around each ground truth object or some local regions) and globally (over the whole image or mini-batch). Notably, We compute IoU-HLR and Score-HLR based on the final located position of samples, other than the coordinates before regression, since mAP is evaluated based on the regressed samples.

As shown in Figure 3, to compute IoU-HLR, we first divide all samples into different groups, based on their nearest ground truth objects. Next, we sort the samples within each group in descending order by their IoU with the ground truth, and get the IoU Local Rank (IoU-LR). Subsequently, we take samples with the same IoU-LR and sort them in descending order. Specifically, all top 1 IoU-LR samples are collected and sorted, followed by top 2, top 3, and so on. These two steps result in the ranking of all samples, that is the IoU-HLR. IoU-HLR follows the two criteria mentioned...
ative samples and then follow the same steps as computing minimum score over all foreground classes as the score of negative samples. We adopt the maximum score over all foreground classes as the score of negative samples and then follow the same steps as computing IoU-HLR, as shown in Figure 3.

We plot the distributions of random, hard and prime samples in Figure 4, with the IoU vs. classification loss. It is observed that hard positive samples tend to have high classification losses and scatter over a wider range along the IoU axis, while prime positive samples tend to have high IoUs and low classification losses. Hard negative samples tend to have high classification losses and high IoUs, while prime negative samples also cover some low loss samples and have a more diverged IoU distribution. This suggests that these two categories of samples are of essentially different characteristics.

4. Learn Detectors via Prime Sample Attention

The aim of object detection is not to obtain a better classification accuracy on average, but to achieve a performance as good possible on prime samples in the set, as discussed in Sec 3. Nevertheless, this is nontrivial. If we just use top IoU-HLR samples for training like what OHEM does, the mAP will drop significantly because most prime samples are easy ones and cannot provide enough gradients to optimize the classifier.

In this work, we propose PrIme Sample Attention, a simple and effective sampling and learning strategy that pays more attention to prime samples. PISA consists of two components: Importance-based Sample Reweighting (ISR) and Classification-Aware Regression Loss (CARL). With the proposed method, the training process is biased on prime samples rather than evenly treat all ones. Firstly, the loss weight of prime samples are larger than the others, so that the classifier tends to be more accurate on these samples. Secondly, the classifier and regressor are learned with a joint objective, thus scores of positive prime samples get boosted relative to unimportant ones.

4.1. Importance-based Sample Reweighting

Given the same classifier, the distribution of performance usually matches the distribution of training samples. If part of the samples occur more frequently in the training data, a better classification accuracy on those samples is supposed to be achieved. Hard sampling and soft sampling are two
different ways to change the training data distribution. Hard sampling selects a subset of samples from all candidates to train a model, while soft sampling assigns different weights for all samples. Hard sampling can be seen as a special case of soft sampling, where each sample is assigned a loss weight of either 0 or 1.

To make fewer modifications and fit existing frameworks, we propose a soft sampling strategy named Importance-based Sample Reweighting (ISR), which assigns different loss weights to samples according to their importance. ISR consists of positive sample reweighting and negative sample reweighting, denoted as ISR-P and ISR-N, respectively. We adopt IoU-HLR as the importance measurement for positive samples and Score-HLR for negative samples. Given the importance measurement, the remaining question is how to map the importance to an appropriate loss weight.

We first transform the rank to a real value with a linear mapping. According to its definition, HLR is computed separately within each class (N foreground classes and 1 background class). For class $j$, supposing there are $n_j$ samples in total with their corresponding HLR $\{r_1, r_2, \ldots, r_{n_j}\}$, where $0 \leq r_i \leq n_j - 1$, we use a linear function to transform each $r_i$ to $u_i$ as shown in Equ. 1. Here $u_i$ denotes the importance value of the $i$-th sample of class $j$. $n_{\text{max}}$ denotes the max value of $n_j$ over all classes, which ensures that the samples at the same rank of different classes will be assigned the same $u_i$.

$$u_i = \frac{n_{\text{max}} - r_i}{n_{\text{max}}}$$ (1)

A monotone increasing function is needed to further cast the sample importance $u_i$ to a loss weight $w_i$. Here we adopt an exponential form as Equ. 2, where $\gamma$ is the degree factor indicating how much preference will be given to important samples and $\beta$ is a bias that decides the minimum sample weight.

$$w_i = ((1 - \beta)u_i + \beta)^\gamma$$ (2)

With the proposed reweighting scheme, the cross-entropy classification loss can be rewritten as Equ. 3, where $n$ and $m$ are the numbers of positive and negative samples; $s$ and $\hat{s}$ denote the predicted score and classification target, respectively. Note that simply adding loss weights will change the total value of losses and the ratio between the loss of positive and negative samples, so we normalize $w$ to $w'$ in order to keep the total loss unchanged.

$$L_{\text{cls}} = \sum_{i=1}^{n} w'_i CE(s_i, \hat{s}_i) + \sum_{j=1}^{m} w'_j CE(s_j, \hat{s}_j)$$

$$w'_i = \frac{w_i \sum_{i=1}^{n} CE(s_i, \hat{s}_i)}{\sum_{i=1}^{n} w_i CE(s_i, \hat{s}_i)}$$

$$w'_j = \frac{w_j \sum_{j=1}^{m} CE(s_j, \hat{s}_j)}{\sum_{j=1}^{m} w_j CE(s_j, \hat{s}_j)}$$ (3)

4.2. Classification-Aware Regression Loss

Re-weighting the classification loss is a straightforward way to focus on prime samples. Besides that, we develop another method to highlight the prime samples, motivated by the earlier discussion that classification and localization is correlated. We propose to jointly optimize the two branches with a Classification-Aware Regression Loss (CARL). CARL can boost the scores of prime samples while suppressing the scores of other samples. The regression quality determines the importance of a sample and we expect the classifier to output higher scores for important samples. The optimization of two branches should be correlated rather than being independent.

Our solution is to add a classification-aware regression loss, so that gradients are propagated from the regression branch to the classification branch. To this end, we propose CARL as shown in Equ. 4. $p_i$ denotes the predicted probability of the corresponding ground truth class and $d_i$ denotes the output regression offset. We use an exponential function to transform the $p_i$ to $v_i$, and then rescale it according to the average value of all samples. $L$ is the commonly
used smooth L1 loss.

\[ L_{carl} = \sum_{i=1}^{n} c_i \mathcal{L}(d_i, \hat{d}_i) \]

\[ c_i = \frac{1}{n} \sum_{i=1}^{n} v_i \]

\[ v_i = ((1 - b)p_i + b)^h \]

It is obvious that the gradient of \( c_i \) is proportional to the original regression loss \( \mathcal{L}(d_i, \hat{d}_i) \). In the supplementary, we prove that there is a positive correlation between \( \mathcal{L}(d_i, \hat{d}_i) \) and the gradient of \( p_i \). Namely, samples with greater regression loss will receive large gradients for the classification scores, which means stronger suppression on the classification scores. In another view, \( \mathcal{L}(d_i, \hat{d}_i) \) reflects the localization quality of sample \( i \), thus can be seen as an estimation of IoU and further seen as an estimation of IoU-HLR. Approximately, top-ranked samples have low regression loss, thus the gradients of classification scores are smaller. With CARL, the classification branch gets supervised by the regression loss. The scores of unimportant samples are greatly suppressed, while the attention to prime samples are reinforced.

5. Experiments

5.1. Experimental Setting

Dataset and evaluation metric. We conduct experiments on the challenging MS COCO 2017 dataset [16]. It consists of two subsets: the train split with 118k images and val split with 5k images. We use the train split for training and report the performance on val and test-dev. The standard COCO-style AP metric is adopted, which averages mAP of IoUs from 0.5 to 0.95 with an interval of 0.05.

Implementation details. We implement our methods based on MMDetection [4]. ResNet-50 [10], ResNeXt-101-32x4d [24], and VGG16 [22] are adopted as backbones in our experiments. Unless otherwise specified, we follow the default setting in MMDetection, and detailed settings are described in the supplementary material.

5.2. Results

Overall results. We evaluate the proposed PISA on both two-stage and single-stage detectors, on two popular benchmarks. We use the same hyper-parameters of PISA for all backbones and datasets. The results on MS COCO dataset are shown in Table 1. PISA achieves consistent mAP improvements on all detectors with different backbones, indicating its effectiveness and generality. Specifically, it improves Faster R-CNN, Mask R-CNN and RetinaNet by 2.1%, 1.8% and 1.4%, respectively, with a ResNet-50 backbone. Even with a strong backbone like ResNeXt-101-32x4d, similar improvements are observed. On SSD300 and SSD512, the gain is 2.0% and 2.1%, respectively. As shown on Table 2, PISA adds a computational overhead of 0.07 0.14 s/iter for training, but there is no additional parameters so the inference time remains the same as that of the baseline.

On the PASCAL VOC dataset, PISA also outperforms the baselines, as shown in Table 3. PISA not only brings performance gains under the VOC evaluation metric that uses 0.5 as the IoU threshold, but performs significant better under the COCO metric that uses the average of multiple IoU thresholds. This implies that PISA is especially beneficial to high IoU metrics and makes more accurate prediction on precisely located samples.

Comparison of different sampling methods. To investigate the effects of different sampling methods, we apply random sampling (R), hard mining (H), and PISA (P) on positive and negative samples. Hard mining here refers to the OHEM 1:3 variant adopted in [15], which fixes the ratio of positive and negative samples to 1:3. In this way, we can apply different sampling methods to positive and negative samples, which allows a more detailed study. We also evaluate the original OHEM implementation (denoted as H*), which forward all 2000 proposals and select 512 samples with the highest loss, without limitation of positive sample ratio. Faster R-CNN is adopted as the baseline method. As shown in Table 4, PISA outperforms random sampling and hard mining in all cases. We find that hard mining is effective when applied to negative samples, but hampers the performance when applied to positive samples. For positive samples, PISA achieves 1.6% and 2.0% higher mAP than random sampling and hard mining, respectively. For negative samples, PISA surpasses them by 0.9% and 0.4%, respectively. When applying to both positive and negative samples, PISA leads to 2.1%, 1.7% and 1.3% improvements compared to random sampling, hard mining and OHEM, respectively. It is noted that the gain mainly originates from the AP of high IoU thresholds, such as AP15. This indicates that attending prime samples helps the classifier to be more accurate on samples with high IoUs. We demonstrate some qualitative results of PISA and the baseline in Figure 5. PISA results in less false positives and higher scores for positive prime samples.

5.3. Analysis

We perform a thorough study on each component of PISA and explain how it works compared with random sampling and hard mining.

Component Analysis. Table 5 shows the effects of each component of PISA. We can learn that ISR-P, ISR-N and CARL improve the mAP by 0.7%, 0.9%, and 1.0%, respectively. ISR (ISR-P + ISR-N) boosts mAP by 1.5%. Applying PISA only to positive samples (ISR-P + CARL) increases mAP by 1.6%. With all 3 components, PISA achieves a
total gain of 2.1%.

Ablation experiments of hyper-parameters. For both ISR and CARL, we use an exponential transformation function of Equ. 2 and 2 hyper-parameters ($\gamma_P$, $\beta_P$ for ISR-P, $\gamma_N$, $\beta_N$ for ISR-N, and $k$, $b$ for CARL) are introduced. The exponential factor $\gamma$ or $k$ controls the steepness of the curve, while the constant factor $\beta$ or $b$ affects the minimum value.

When performing ablation study on hyper-parameters of ISR-P, ISR-N or CARL, we do not involve other components. A larger $\gamma$ and a smaller $\beta$ mean a larger gap between prime samples and unimportant samples, so that we focus more on prime samples. The opposite case means we pay more equal attention to all samples. Through a coarse search, we adopt $\gamma_P = 2.0$, $\beta_P = 0.0$, $\gamma_N = 0.5$, $\beta_N = 0.0$ for ISR, and $k = 1.0$, $b = 0.2$ for CARL. We also observe that the performance is not very sensitive to those hyper-parameters.

**What samples do different sampling strategies prefer?** To understand how ISR works, we study the sample distribution of different sampling strategies from the aspects of IoU and loss. Sample weights are taken into account when obtaining the distribution. Results are shown in Figure 6. For positive samples, we learn that samples selected by hard positive samples are preferred by ISR-P, ISR-N, or CARL.

For ISR-P, ISR-N, or CARL, we do not involve other components. A larger $\gamma$ and a smaller $\beta$ mean a larger gap between prime samples and unimportant samples, so that we focus more on prime samples. The opposite case means we pay more equal attention to all samples. Through a coarse search, we adopt $\gamma_P = 2.0$, $\beta_P = 0.0$, $\gamma_N = 0.5$, $\beta_N = 0.0$ for ISR, and $k = 1.0$, $b = 0.2$ for CARL. We also observe that the performance is not very sensitive to those hyper-parameters.

**What samples do different sampling strategies prefer?** To understand how ISR works, we study the sample distribution of different sampling strategies from the aspects of IoU and loss. Sample weights are taken into account when obtaining the distribution. Results are shown in Figure 6. For positive samples, we learn that samples selected by hard positive samples are preferred by ISR-P, ISR-N, or CARL.
mling and PISA diverge from each other. Hard samples have high losses and low IoUs, while prime samples come with high IoUs and low losses, indicating that prime samples tend to be easier for classifiers. For negative samples, PISA presents an intermediate preference between random sampling and hard mining. Unlike random sampling that focuses more on low IoU and easy samples, and hard mining that attends to relatively high IoU and hard samples, PISA maintains the diversity of samples.

**How does ISR affect classification scores?** ISR assigns larger weights to prime samples, but does it achieve the biased classification performance as expected? In Figure 7, we plot the score distribution of positive and negative samples w.r.t. different HLRs. For positive samples, the scores of top-ranked samples are higher than those of baseline, while that of lower-ranked samples are lower. The result demonstrates that ISR-P biases the classifier, thus boosting the prime samples while suppressing others. For negative samples, the scores of all samples are lower than those of the baseline, especially for top-ranked samples. This implies that ISR-N has a strong suppression for false positives.

**How does CARL affect classification scores?** CARL correlates the classification and localization branches by introducing the classification scores to the regression loss. The gradient will suppress the scores of samples with lower regression quality, but highlight the prime samples that are localized more accurately. Figure 8 shows the scores of samples of different IoUs. Compared with the FPN baseline, CARL boosts scores of high IoU samples but decreases scores of low IoU samples as expected.

**Is IoU-HLR better than other metrics?** The results prove that for positive sampling, IoU-HLR is an effective importance metric while loss is not, but is it better than others? We test other metrics for ISR, including loss rank, IoU, and IoU before regression (denoted as IoU*). The results are shown in Table 7, which suggests (1) the performance is more related to IoU instead of loss, and (2) using the locations after regression is important, and (3) IoU-HLR is better than IoU. These results match our intuition and analysis in Sec. 3.

**6. Conclusion**

We study on what are the most important samples for training an object detector and establishing the notion of prime samples. We present PrIme Sample Attention (PISA), a simple and effective sampling and learning strategy to highlight important samples. On both MS COCO and PASCAL VOC datasets, PISA achieves consistent improvements over random sampling and hard mining counterparts.

**Acknowledgement.** This work is partially supported by the SenseTime Collaborative Grant on Large-scale Multimodality Analysis (CUHK Agreement No. TS1610626 & No. TS1712093), the General Research Fund (GRF) of Hong Kong (No. 14236516 & No. 14203518), Singapore MOE AcRF Tier 1 (2018-T1-002-056), NTU SUG, and NTU NAP.
References