Adaptive Hierarchical Representation Learning for Long-Tailed Object Detection

Banghuai Li
MEGVII Technology
libanghuai@gmail.com

Abstract

*General object detectors are always evaluated on hand-designed datasets, e.g., MS COCO and Pascal VOC, which tend to maintain balanced data distribution over different classes. However, it goes against the practical applications in the real world which suffer from a heavy class imbalance problem, known as the long-tailed object detection. In this paper, we propose a novel method, named **A**daptive **H**ierarchical **R**epresentation **L**earning **(AHRL)**, from a metric learning perspective to address long-tailed object detection. We visualize each learned class representation in the feature space, and observe that some classes, especially under-represented scarce classes, are prone to cluster with analogous ones due to the lack of discriminative representation. Inspired by this, we propose to split the whole feature space into a hierarchical structure and eliminate the problem in a coarse-to-fine way. AHRL contains a two-stage training paradigm. First, we train a normal baseline model and construct the hierarchical structure under the unsupervised clustering method. Then, we design an AHR loss that consists of two optimization objectives. On the one hand, AHR loss retains the hierarchical structure and keeps representation clusters away from each other. On the other hand, AHR loss adopts adaptive margins according to specific class pairs in the same cluster to further optimize locally. We conduct extensive experiments on the challenging LVIS dataset and AHRL outperforms all the existing state-of-the-art methods, with **29.1%** segmentation AP and **29.3%** box AP on LVIS v0.5 and **27.6%** segmentation AP and **28.7%** box AP on LVIS v1.0 based on ResNet-101. We hope our simple yet effective approach will serve as a solid baseline to help stimulate future research in long-tailed object detection. Code will be released soon.*

1. Introduction

The emerging of convolutional neural networks (CNNs) leads to prosperity in object detection. With effort of re-

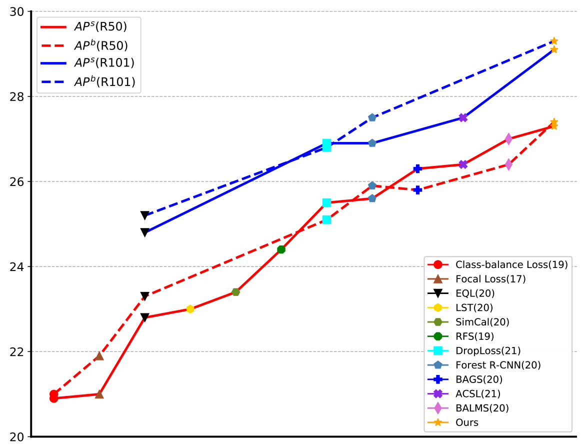
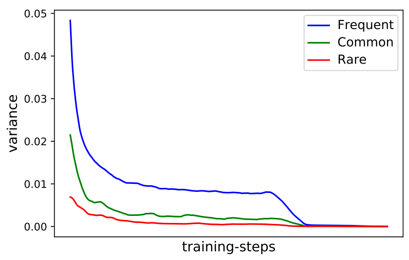


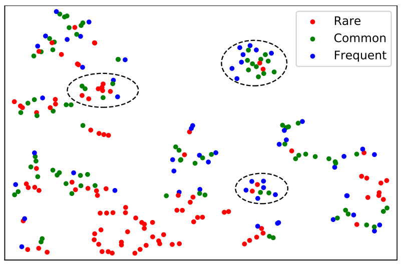
Figure 1. Comparisons between the state-of-the-art methods and our AHRL on LVIS v0.5 [8]. We report different task results (object detection and instance segmentation) on both ResNet-50(red) and ResNet-101(blue) backbones. AP^s stands for the segmentation AP, while AP^b means the box AP. Our proposed AHRL outperforms all the existing methods.

searchers, recent advances in object detection achieve encouraging results in manually balanced datasets, like Pascal VOC [5] and MS COCO [18]. However, in reality, we always need to face long-tailed distributed data [25], where head classes(classes with plenty of instances) and tailed/scarce classes(classes with few instances) significantly differ in the number of instances. Nevertheless, many traditional detection models are hard to take care of head classes and tailed classes in the same time, resulting in the desire for an adaptive solution.

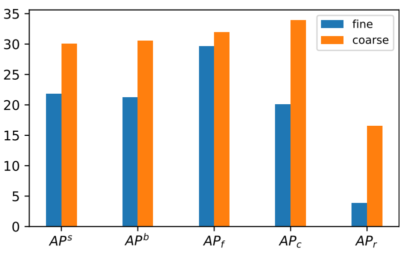
Different from long-tailed object recognition, an additional localization sub-task makes long-tailed object detection more challenging. Extreme imbalance of the instance number for each class still restricts its performance. Almost all the past works [3, 12, 15, 31, 35, 37] on long-tailed object detection reach a consensus that classifier is the major bottleneck for further improvements. As shown in Figure



(a) Variance of weights



(b) t-SNE visualization of class weights



(c) Results of coarse and fine classification

Figure 2. (a) The average variance for different frequent groups. (b) t-SNE visualization of classifier weights in Mask R-CNN. Red, green and blue points stand for the class weight/center of rare, common and frequent classes, respectively. And dotted ellipses mark some obvious clusters. (c) Results of coarse and fine classification. The blue bar represents the standard result of Mask R-CNN on LVIS v0.5, while orange bar represents the coarse result by ignoring misclassification in the same cluster.

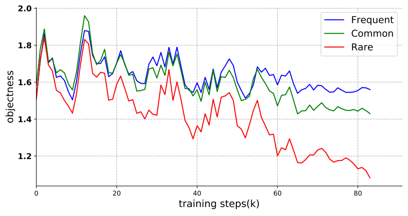
2a, we calculate the variance of the classification weight for each class during the model training and take average according to their frequency groups, i.e., rare, common, and frequent in LVIS v0.5 [8]. Head classes dominate the model optimization due to the more diverse samples, while tailed classes are seldom tackled because of the heavy data imbalance. Thus, it always leads to unsatisfactory performance. Following long-tailed object recognition, early attempts in long-tailed object detection exploit data re-sampling [3, 8] and loss re-weighting [7,14,23,29,31,35] strategies to solve this problem. By data re-sampling, a more balanced dataset is given to the model, preventing the bias to head classes to some extent. Compared with directly balancing dataset, loss re-weighting approaches elaborately modify the weight to adapt to the long-tailed scene. However, these methods suffer from overfitting to the limited data, and the overall performance is always sensitive to the re-weighting or re-sampling hyperparameters.

In this work, we present a simple yet effective method, named ***Adaptive Hierarchical Representation Learning (AHRL)***, from a metric learning perspective to address the long-tailed object detection problem. As shown in Figure 2b, we take Mask R-CNN [10] as an example model to train on LVIS v0.5 [8] dataset and utilize t-SNE [33] to visualize each class weight. Each dot in Figure 2b stands for a specific class center, and we select 247 out of 1230 classes for better illustration. Moreover, rare, common, and frequent classes are marked in red, green, and blue, respectively (detailed class information for those dots can be found in our supplementary materials). We can find an interesting phenomenon that some classes, especially under-represented scarce classes, are prone to cluster with analogous ones due to the lack of discriminative representation. Thus, these classes tend to be misclassified and result in poor performance. Go a step further. We adopt K-Means to group all the class centers into 50 clusters and ignore the misclassification in the same cluster to re-evaluate the performance. As depicted in Figure 2c, we distinguish this evaluation method and the standard one as *coarse* and *fine* classification results, respectively, and we observe that coarse results are much better than fine results, especially for scarce classes, which also verifies our assumption above. This discovery opens up room to optimize the long-tailed object detection performance and inspires us to handle this tough problem in a coarse-to-fine way.

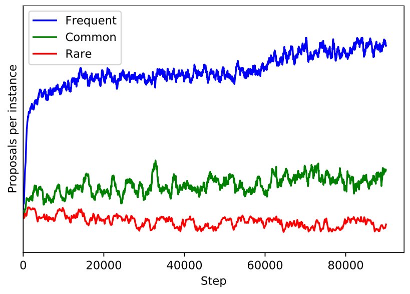
Motivated by the observation above, we resort to a coarse-to-fine strategy to tackle this problem and design a two-stage training procedure **AHRL** from a hierarchical representation learning perspective. In the first stage, we follow standard settings in [8,10,31] to train a typical baseline model, i.e., Mask R-CNN. Then, we adopt unsupervised clustering algorithms, i.e., K-Means, to build the hierarchical feature space based on the classification weights of the pre-trained model. In the second stage, we propose a novel loss function, named **A**daptive **H**ierarchical **R**epresentation loss (or **AHR** loss), to implement our coarse-to-fine design. AHR loss involves two optimization objectives, one for coarse-grained classification and the other one for fine-grained classification. On the one hand, AHR loss retains the constructed hierarchical structure and prompts all clusters to repel each other. On the other hand, AHR loss adopts dynamic and adaptive margins according to the specific relationship between each class pair in the same cluster, the more similar pairs they are and the larger margins between them are performed during the whole training process, to make indistinguishable classes more discriminative. We conduct extensive experiments on LVIS dataset and achieve new state-of-the-art results with both ResNet-50 [11] and ResNet-101 [11] backbones, as shown in Figure 1.

To sum up, the contributions of this work are as follow:

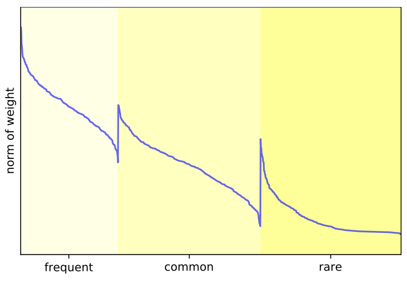
- We delve deep into the long-tailed object detection problem and present a strong baseline to help ease future research, which already beats the most state-of-the-art methods.



(a) Objectness score



(b) Proposals per instance



(c) Norm of class weights

Figure 3. (a) Average objectness score per instance in RPN during training. All boxes are filtered by IOU threshold and matched with corresponding ground-truth to get their labels and frequency. Different frequency groups are marked by different colors. (b) Proposals per instance. We monitor the average proposals per instance for different frequency groups during the model training. (c) Magnitude of weight vectors for different classes. Different background colors stand for different frequency groups. The norms of weights are sorted in descending order in every frequency group.

- We present a simple and effective approach, named Adaptive Hierarchical Representation Learning(AHRL), from a metric learning perspective to eliminate long-tailed object detection in a coarse-to-fine way. A novel AHR loss is also proposed to make AHRL work better.
- Compared with other existing state-of-the-art methods, our proposed method outperforms them and achieves a new state-of-the-art performance on LVIS benchmarks with various backbones.

2. Related Work

General Object Detection and Instance Segmentation. The rise of deep learning improves the performance of object detection in recent years. These deep learning-based frameworks can be divided into two categories. One-stage approaches [17, 20, 24] chase faster inference speed, while two-stage frameworks [6, 27] prefer a higher accuracy. With the appearance of Mask R-CNN [10], the gap between object detection and instance segmentation disappeared by adding an extra segmentation branch upon Faster R-CNN [27].

Long-tailed Recognition. Common methods towards long-tailed recognition can be summarized as follows. 1) **Data re-sampling.** It is the most intuitive way by duplicating tailed samples [8, 9] or under-sampling head samples [4] to deal with the long-tailed distribution. [38] goes a step further by changing the ratio of head and tailed classes over time. But all of them cannot avoid under-fitting in head classes or over-fitting in tailed classes. 2) **Data augmentation.** Generating or synthesizing new samples is always used to enlarge the limited dataset. Recent studies [1, 2, 19] manage to create fake samples for tailed classes to address long-tailed distribution. However, these methods are limited to the diversity of tailed classes. 3) **Loss re-weighting.** Instead of modifying input, modifying loss function directly is also a popular way to settle down this problem. Recently, several works [30, 31, 35] seek many ways to adapt the loss weight for both head and tailed classes to prevent severe and frequent punishment to tailed classes.

Long-tailed Object Detection. As long-tailed recognition becomes mature, researchers start to pay attention to long-tailed detection. Meanwhile, Facebook start a long-tailed detection challenge with dataset LVIS [8]. EQL loss [31] easily decreases the times to suppress punishment to tailed classes to conquer this problem. Following EQL, ACSL [35] prevents tailed classes from suppressing of head classes and preserves the discrimination between similar classes. Besides focusing on loss function, some methods deliberately design the last layer in the classifier. Forest R-CNN [37] constructs a classification forest with different prior knowledge to incorporates relations. BAGS [15] uses a cascade-like softmax layer to alleviate the difference between head classes and tailed classes in quantity. These structures avoid the imbalance between reward and punishment in a specific part of the model. Moreover, some adaptive methods [26, 32] provide in long-tailed classification still have an fantastic result in long-tailed object detection.

In this paper, we tackle the long-tailed object detection problem from the metric learning perspective. By splitting the whole feature space into the hierarchical structure, AHRL can handle the tough problem in a divide-and-conquer way and achieves superior results. It should be noted that Forest R-CNN [37] adopts an analogous hierarchical split method. However, it achieves this by adding a separate classification branch to distinguish parent classes, which results in a severe inconsistency between parent and fine-grained classification as these two branches are projected to different feature spaces. On the contrary, our proposed AHRL adopts unsupervised clustering algorithms based on fine-grained classes to construct the hierarchical structure and optimize both coarse-grained and fine-grained classes at the same time.

3. Proposed Method

In this section, we first introduce a strong baseline model to help ease future research in long-tailed object detection and further verify the effectiveness of our proposed method. Then, we discuss our proposed method AHRL, followed by the details of the AHR loss.

3.1. Preliminary and A Strong Baseline

In the past works, the naive Mask R-CNN [10] is invariably adopted as the baseline model to conduct experiments and verify the superiority of their proposed methods. However, with the development of modern deep learning methods, some intuitive and mature techniques can boost naive baseline performance to a certain extent. This section delves deep into the long-tailed object detection problem and presents a strong baseline based on naive Mask R-CNN, named *baseline++* for simplicity. Baseline++ serves as the baseline model to further verify the effectiveness of our proposed method. Its details are described as follows.

Proposal Oversampling. As shown in Figure 3a and Figure 3b, we observe a clear gap between tailed classes and head classes on the average objectness score during the model training. The proposals of tailed classes tend to achieve lower objectness score and be filtered out before the ROI head in Mask R-CNN. Figure 3b can well illustrate the phenomenon, average proposals per instance for tailed classes are much smaller than head classes, which results in less optimization for tailed classes. According to these findings, we directly double the maximum number of tailed class proposals allowed to keep after non-max suppression(NMS), bringing more foreground proposals for tailed classes.

Cosine Similarity Classifier. Generally speaking, the fully-connected (FC) layer is the default choice to implement classifiers in most object detectors. However, the fully-connected layer will result in an obvious bias towards head classes, when it comes to the long-tailed object detection problem. Supposing $\mathcal{F}$ is the feature extractor, $W^c \in \mathbb{R}^{d\times k}$ is the final classification weight matrix, k is the number of whole classes, and $W^c = [w_1^c, w_2^c, ..., w_k^c]$, where $w_i^c \in \mathbb{R}^d$ is the corresponding classification weight vector for i-th class. When given an input sample vector x, we can get the raw classification score s_i^c by dot-product operation:

$$\begin{aligned} s_i^c &= \mathcal{F}(x)^T \cdot w_i^c \\ &= \|\mathcal{F}(x)\| \, \|w_i^c\| \cos(\theta) \end{aligned} \tag{1}$$

We can find that all things being equal, weight vectors with larger magnitude tend to yield higher scores. As shown in Figure 3c, we take Mask R-CNN [10] as an example and observe that weight vectors of head classes have a much larger magnitude comparing with tailed classes, which results in prediction preference towards head classes. Inspired by [36], we adopt a cosine similarity classifier to replace the original linear classifier to reduce the intra-class variance, which is defined as follows:

$$s_i^c = \lambda_c \cdot \frac{\mathcal{F}(x)^T \cdot w_i^c}{\|\mathcal{F}(x)\| \, \|w_i^c\|} \tag{2}$$

where $\mathcal{F}(x)$ is the feature of a given proposal, w_i^c is the weights for class i and λ_c is the scaling factor.

Moreover, compared with weights in a fully-connected classifier, weights from a cosine classifier, without preference and bias, can respond to the relationship among classes better, which can lay the foundation for subsequent excellent clustering results.

Other Effective Attempts. According to our discussion in the *related work* section, we adopt EQL [31] as the loss re-weighting method and GIoU [28] is utilized to replace the default Smooth L1 loss for more accurate bounding box regression. Besides, we also try several simple data augmentation methods to increase the data diversity. Due to the space limit, please refer to our *Appendix* for more detailed description about these attempts.

3.2. Adaptive Hierarchical Representation Learning

Motivated by our findings in Section 1, we design a simple yet effective method, named ***A**daptive **H**ierarchical **R**epresentation **L**earning* (**AHRL**), from a metric learning perspective. An overview of AHRL's pipeline can be found in Figure 4. AHRL contains a two-stage training paradigm. In the first stage, we follow standard settings in [10] to train a normal baseline++ model, illustrated in Section 3.1. Then, we construct the hierarchical feature space based on the classification nodes of pre-trained baseline++. An intuitive way to achieve this is with the help of modern clustering algorithms. In this section, without loss of generality, we only cluster these classification nodes for once and divide the whole classification feature space into two levels for better illustration. Assuming we get n clusters finally, e.g., $\mathbb{P} = \{\mathcal{P}_1, \mathcal{P}_2, ..., \mathcal{P}_n\}$, and the cluster representation w_i^p is defined as the mean of each classification node w_j^c in cluster $\mathcal{P}_i$:

$$w_i^p = \frac{\sum_{j\in\mathcal{P}_i} w_j^c}{\|\mathcal{P}_i\|} \tag{3}$$

where $\|\mathcal{P}_i\|$ equals to the number of nodes in set $\mathcal{P}_i$.

When it comes to clustering algorithms, there are two typical choices we can take. One is based on unsupervised clustering methods like K-Means to aggregate similar classification nodes, which is deemed to utilize visual information to some extend. The other one is based on lexical infor-

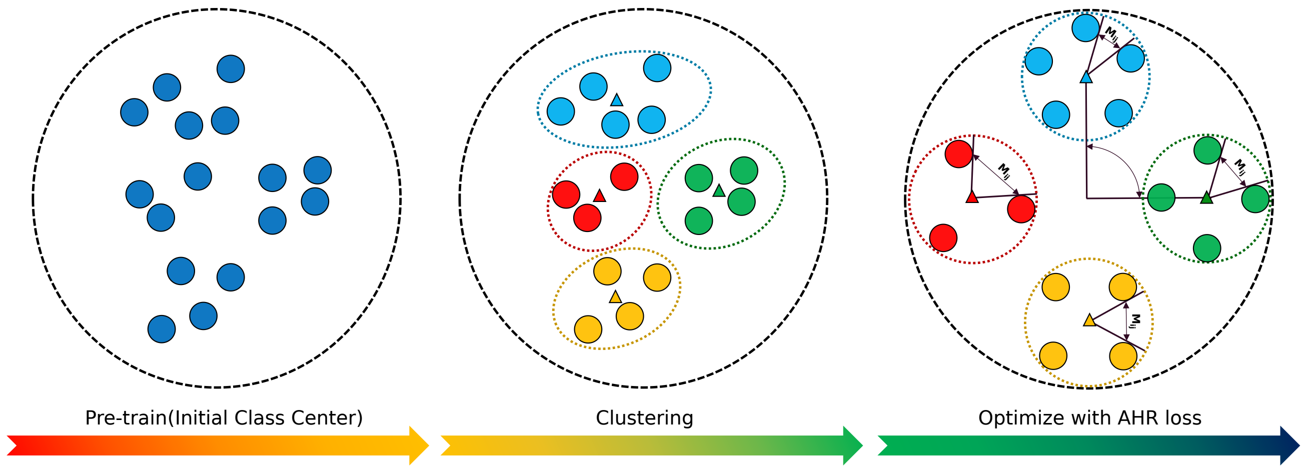


Figure 4. Overview of our adaptive hierarchical representation learning. In the first stage, we train a simple baseline model and each blue dot in the first circle represents a specific class center. Next, clustering algorithms, i.e., K-Means, are adopted to construct the hierarchical structure and each cluster representation (triangle in the second circle) is defined as the mean of each node in the same cluster. Finally, we fine-tune the model and optimize it with our AHR loss. M_{ij} is the adaptive margin between each class pair.

Table 1. Ablation for each component in our proposed strong baseline. AP_r, AP_c and AP_f denote segmentation average precision for rare, common and frequent classes, respectively. $*$ indicates that the reported result is based on our own implementation.

Method	EQL*	GIoU	Proposal Oversample	Data Augmentation	Cosine Similarity Head	AP^b	AP^s	AP_r	AP_c	AP_f
Mask R-CNN* [8]	✗	✗	✗	✗	✗	23.6	24.2	14.0	24.2	28.3
Baseline++ (ours)	✓	✗	✗	✗	✗	25.5	25.9	17.1	27.1	28.0
	✓	✓	✗	✗	✗	25.8	26.2	17.8	27.1	28.5
	✓	✓	✓	✗	✗	26.1	26.4	17.7	27.6	28.4
	✓	✓	✓	✓	✗	26.5	26.7	17.8	28.0	28.6
	✓	✓	✓	✓	✓	**26.7**	**26.8**	**17.9**	**28.2**	**28.7**

mation, e.g., WordNet [21], to provide an intuitive hierarchical structure. However, lexical information is not always consistent with visual characteristics. For example, *seagull* and *plane* are totally different classes in WordNet [21], but they look pretty similar from visual perspectives. We make a detailed comparison between these two methods in Section 4.2.

Finally, in the second stage, we keep the hierarchical structure and eliminate long-tailed object detection in a coarse-to-fine way through the AHR loss, which is described in the next section.

3.3. Adaptive Hierarchical Representation Loss

Adaptive **H**ierarchical **R**epresentation (**AHR**) loss consists of two optimization objectives, $\mathcal{L}_{coarse}$ and $\mathcal{L}_{fine}$. On the one hand, $\mathcal{L}_{coarse}$ retains the hierarchical structure and prompts all clusters to repel each other. On the other hand, $\mathcal{L}_{fine}$ adopts adaptive margins according to the specific relationship between each class pair in the same cluster to further optimize locally. Thus, the overall formulation of AHR loss can be defined as follows:

$$\mathcal{L}_{AHR} = \mathcal{L}_{fine} + \lambda \mathcal{L}_{coarse} \tag{4}$$

where λ is the hyperparameters to balance the scale of $\mathcal{L}_{coarse}$.

More specifically, $\mathcal{L}_{coarse}$ serves as a coarse-grained classification loss to distinguish each cluster clearly, and we adopt a simple cross-entropy loss to achieve this goal:

$$L_{coarse} = -\sum_i p_i log\sigma(s_i^p) + (1-p_i)log(1-\sigma(s_i^p)) \tag{5}$$

where:

$$s_i^p = \lambda_p \cdot \frac{\mathcal{F}(x)^T \cdot w_i^p}{\|\mathcal{F}(x)\| \, \|w_i^p\|} \tag{6}$$

$$p_i = \begin{cases} 1, & x \in \mathcal{P}_{\pi_i} \\ 0, & x \notin \mathcal{P}_{\pi_i} \end{cases} \tag{7}$$

where w_i^p is the weight for cluster i, which is defined in Eq. (3), π_i is the cluster index corresponding to class i, σ is the $Sigmoid$ operation, λ_p is the scaling factor for cluster predictions.

$\mathcal{L}_{coarse}$ only focuses on coarse-grained cluster classification. Thus $\mathcal{L}_{fine}$ is necessary for further fine-grained classification in the single cluster. It is worth mentioning that our proposed $\mathcal{L}_{fine}$ adopts adaptive margin mechanism to make each class more discriminative:

$$\mathcal{L}_{fine} = -\sum_i (y_i log \sigma_{ad}(s^c_{i,y_i}) + (1-y_i)\sum_j log(1-\sigma_{ad}(s^c_{i,j}))) \tag{8}$$

where:

$$\sigma_{ad}(s^c_{i,j}) = \frac{1}{1+e^{-(s^c_{i,j}+M_{y_i,j})}} \tag{9}$$

where y_i is the ground-truth label of i-th proposal, $s^c_{i,j}$ is the raw score of j-th class for proposal i and M is a matrix measuring the specific margin values between each class pair. This paper adopts the cosine similarity between each class pair to reflect the margin between them. As we have discussed in Section 3.1, the classifier weights of our proposed method are all normalized. Thus the M can be directly defined as follows:

$$M_{i,j} = \begin{cases} 0, & i = j \\ \lambda_m max(0, (w^c_i \cdot (w^c_j)^T)) & i \neq j \end{cases} \tag{10}$$

where λ_m is a hyperparameter to control the degree of mutual exclusion, which is set as 2 by default. Moreover, because the purpose of $\mathcal{L}_{fine}$ is to optimize classification performance locally in the single cluster, M is restricted to kick in for those classes in the same cluster:

$$I_{i,j} = \begin{cases} 1, & \pi_i = \pi_j \\ 0, & \pi_i \neq \pi_j \end{cases} \tag{11}$$

$$M = M \cdot I \tag{12}$$

It is noteworthy that M is calculated dynamically during the model training until all the class nodes achieve an optimal status.

Consequently, $\mathcal{L}_{AHR}$, the combination of $\mathcal{L}_{coarse}$ and $\mathcal{L}_{fine}$, works in a coarse-to-fine way to effectively address the long-tailed object detection problem. Besides, $\mathcal{L}_{AHR}$ is easy to extend to $Softmax$ version, and we implement $\mathcal{L}_{AHR}$ in this paper based on $Sigmoid$ for simplicity.

3.4. Training Objective

In the first stage, the base detector is trained with a standard Mask R-CNN [10] i.e., a typical loss $\mathcal{L}_{rpn}$ to improve the qualification of foreground proposals, a EQL [31] loss and GIoU loss for box classification and box regression respectively in ROI head. In the second stage, $\mathcal{L}_{fine}$ is adopted to our proposed baseline++ , and it can be reformulated based on [31] and Eq. (8):

$$\begin{aligned}\mathcal{L}_{fine} = -\sum_i (y_i log \sigma_{ad}(s^c_{i,y_i}) \quad + \\ \sum_j E(r) T_{\lambda_r}(f_i)(1-y_i) log(1-\sigma_{ad}(s^c_{i,j})))\end{aligned} \tag{13}$$

Finally, the overall objective function in the second stage is as follow, and λ is set as 1 by default:

$$\begin{aligned}\mathbb{L} =& \mathcal{L}_{rpn} + \mathcal{L}_{reg} + \mathcal{L}_{AHR} \\ =& \mathcal{L}_{rpn} + \mathcal{L}_{GIoU} + \mathcal{L}_{fine} + \lambda \mathcal{L}_{coarse}\end{aligned} \tag{14}$$

4. Experiments

4.1. Experiment Setup

Datasets. Large Vocabulary Instance Segmentation(LVIS) dataset,a large long-tailed vocabulary dataset in long-tailed detection, consists of 1230 categories in v0.5 and 1203 categories in v1.0. Since LVIS is a federated dataset [8], a few annotations are missing and few annotations are ambiguous. All categories are officially divided into three groups: frequent(more than 100 images), common(10 to 100 images), and rare(less than 10 images). Following the official guideline, we train our model on the train set and evaluate the result on the val set. Besides widely-used AP across IoU threshold from 0.5 to 0.95, AP for frequent(AP_f), common(AP_c), rare(AP_r) groups will be reported respectively for both object detection and instance segmentation results.

Implementation Details. We use Mask R-CNN [10] as our base detector and ResNet-50 [11] with a Feature Pyramid Network [16] as the backbone. We use 8 GPUs with a batch size 16 for training. Our model is trained using stochastic gradient descent(SGD) with momentum 0.9 and weight decay 0.0001 for 90k steps, with an initial learning rate of 0.02, which is decay to 0.002 and 0.0002 at 60k and 80k respectively. We adopt a class-specific branch for both mask and bounding box regression. The threshold of the prediction score is set to be 0.05. We follow [36] to set λ_c and λ_p as 20 in our experiments, respectively. We set λ to 1 to balance the scale of the losses. Following [31], λ_r is set to be 1.76×10^{-3}.

4.2. Ablation Studies

In this section, we take Mask R-CNN [10] based on ResNet-50 [11] as the baseline model to perform ablation studies on LVIS v0.5 [8] unless otherwise specified.

Ablation for each component in Baseline++. We follow standard settings in [8, 15, 31, 37] and adopt [8] equipped with the Repeat Factor Sampling (RFS) method

Table 2. Performance comparisons with the state-of-the-art methods on LVIS v0.5 [8]. ResNet-50 and ResNet-101 are adopted as the backbones respectively for fair comparisons. $*$ indicates the reported result is based on its official implementation under Pytorch [22] framework.

Method	Conference	Backbone	AP^b	AP^s	AP_r	AP_c	AP_f
Class-balanced Loss [3]	*CVPR 2019*	ResNet-50-FPN	21.0	20.9	8.2	21.2	25.7
Focal Loss [17]	*ICCV 2017*	ResNet-50-FPN	21.9	21.0	9.3	21.0	25.8
EQL [31]	*CVPR 2020*	ResNet-50-FPN	23.3	22.8	11.3	24.7	25.1
RFS [8]	*CVPR 2019*	ResNet-50-FPN	-	24.4	14.5	24.3	28.4
LST [13]	*CVPR 2020*	ResNet-50-FPN	-	23.0	-	-	-
SimCal [34]	*ECCV 2020*	ResNet-50-FPN	-	23.4	16.4	22.5	27.2
Forest R-CNN [37]	*ACMMM 2020*	ResNet-50-FPN	25.9	25.6	18.3	26.4	27.6
BAGS [15]	*CVPR 2020*	ResNet-50-FPN	25.8	26.3	18.0	26.9	28.7
BALMS* [26]	*NeurIPS 2020*	ResNet-50-FPN	26.4	27.0	17.3	28.1	29.5
DropLoss [12]	*AAAI 2021*	ResNet-50-FPN	25.1	25.5	13.2	27.9	27.3
ACSL [35]	*CVPR 2021*	ResNet-50-FPN	-	26.4	18.6	26.4	29.4
EQL [31]	*CVPR 2020*	ResNet-101-FPN	25.2	24.8	14.6	26.7	26.4
Forest R-CNN [37]	*ACMMM 2020*	ResNet-101-FPN	27.5	26.9	20.1	27.9	28.3
DropLoss [12]	*AAAI 2021*	ResNet-101-FPN	26.8	26.9	14.8	29.7	28.3
ACSL [35]	*CVPR 2021*	ResNet-101-FPN	-	27.5	19.3	27.6	**30.7**
AHRL(ours)	N/A	ResNet-50-FPN	27.4	27.3	17.5	29.0	29.1
AHRL(ours)	N/A	ResNet-101-FPN	**29.3**	**29.1**	**21.3**	**30.7**	30.3

Table 3. Comparisons between various clustering strategies.

Method	Clusters	AP^b	AP^s	AP_r	AP_c	AP_f
WordNet	108	26.7	26.9	16.7	28.3	29.2
K-Means	108	26.9	27.0	16.9	28.6	28.9
K-Means	200	**27.4**	**27.3**	**17.5**	**29.0**	29.1
K-Means	400	27.1	26.8	16.1	28.5	**29.3**

Table 4. Comparisons between our proposed method and the baseline Mask R-CNN based on various backbones.

Method	Backbone	AP^b	AP^s	AP_r	AP_c	AP_f
Mask R-CNN [10]	ResNet-50-FPN	23.6	24.2	14.0	24.2	28.3
AHRL(ours)	ResNet-50-FPN	**27.4**	**27.3**	**17.5**	**29.0**	**29.1**
Mask R-CNN [10]	ResNet-101-FPN	26.0	26.2	18.0	26.3	29.4
AHRL(ours)	ResNet-101-FPN	**29.3**	**29.2**	**21.3**	**30.7**	**30.3**

as our baseline model. In Section 3.1, we propose an effective strong baseline baseline++ and Table 1 shows that each component of baseline++ can effectively promote the overall performance.

General object detectors always suffer from heavy class imbalance in long-tailed object detection problem, and EQL [31] alleviates this by ignoring the suppression to tails when they act as the negative samples. Finally, it can achieve around **1.9%** segmentation **A**verage **P**recision (AP) and **2.0%** box AP gains in our implementation. GIoU [28], a more advanced IoU-based regression loss, achieves about **0.3%** segmentation AP and **0.3%** box AP gains against the default Smooth L1 loss in our settings. Unlike the common data sampling methods [3, 8], our proposed proposal oversampling for rare classes can eliminate the class imbalance problem more intrinsically, and it achieves **0.2%** segmentation AP and **0.3%** box AP improvements, respectively.

As we all know, it is almost common sense to perform data augmentation on scarce classes to ease class imbalance. In this paper, we try several simple data augmentation methods and find random cropping and color jitter can contribute to the performance, which achieves **0.3%** segmentation AP and **0.4%** box AP improvements, respectively. In addition, we replace the final fully-connected layer with the cosine similarity head for the classification and it achieves about **0.2%** box AP and **0.1%** segmentation gains, which keeps consistent with our discussion about the magnitude of weight vectors in Section 3.1. In summary, our proposed strong baseline achieves about **2.8%** segmentation AP and **2.2%** box AP improvements comparing with the original baseline in [8].

Effectiveness of our proposed method. We adopt two typical backbones, i.e., ResNet-50 [11] and ResNet-101 [11], to implement AHRL based on Mask-RCNN [10] to verify the effectiveness of our method. Table 4 shows the detailed comparisons. We can find that AHRL outperforms the baseline model by a large margin, whether it is based on ResNet-50 or ResNet-101. Concretely, AHRL achieves about **3.1%** segmentation AP and **3.8%** box AP gains with

Table 5. Comparisons between different training strategies.

Method	Backbone	AP^b	AP^s	AP_r	AP_c	AP_f
Baseline++	ResNet-50-FPN	26.7	26.8	17.9	28.2	28.7
Baseline++†	ResNet-50-FPN	26.8	27.0	15.4	29.0	**29.2**

ResNet-50, while it also achieves about **3.0%** segmentation AP and **3.3%** box AP gains with ResNet-101. This experiment proves that AHRL can work very well with various backbones and achieve promising results. We randomly sample several images from LVIS v0.5 to intuitively depict the effect of our AHRL, visualization results can be found in our appendix.

Different clustering strategies. Clustering algorithms play an important role in our proposed AHRL. In this section, we conduct extensive experiments on unsupervised K-Means and WordNet [21]. As shown in Table 3, we follow WordNet settings in [37] and group all the classification nodes into 108 clusters. We can find that the overall performance of K-Means is slightly better than WordNet under the same settings, which keeps consistent with our discussion in Section 3.2. Moreover, AHRL achieves optimal results when we group all the classes into 200 clusters. We have to emphasize that we do not pay much attention to fine-tuning the cluster hyperparameter otherwise we believe AHRL can achieve further improvements.

Discussion about our training paradigm. As we have described in Section 3.2, our proposed AHRL involves a two-stage training paradigm. To eliminate the doubt that whether the gain is brought by the 2x training time, in the second stage, we follow the same settings in AHRL to fine-tune the pre-trained model without any extra modifications, and we mark the result in Table 5 as baseline++†. We observe that baseline++† achieves comparable performance with the pre-trained model. 2x training time leads to even worse prediction bias towards head classes. It is noteworthy that we strictly share the same settings between the two-stage training paradigm, e.g., learning rate, batch size, etc. Thus, we attribute it to the influence of different initial statuses.So far, we can conclude that the improvements brought by AHRL benefit from our novel design instead of the training time.

4.3. Comparisons with State-of-the-art Methods

As shown in Table 2 and Table 6, we compare our proposed method with all the published state-of-the-art methods. It is obvious that AHRL achieves superior performance and sets up a new state-of-the-art record on both LVIS v0.5 [8] and LVIS v1.0 [8] dataset. Moreover, it is worth mentioning that our proposed AHRL is free to boost long-tailed object detection performance without any extra inference cost. Due to the space limit, detailed results of LVIS v1.0 for each sub-category are reported in our supplementary materials.

Table 6. Performance comparisons with the state-of-the-art methods on LVIS v1.0 [8].

Method	Backbone	AP^b	AP^s
Mask R-CNN [10]	ResNet-50-FPN	20.0	19.2
EQL [31]	ResNet-50-FPN	22.5	21.6
BAGS [15]	ResNet-50-FPN	23.7	23.1
DropLoss [12]	ResNet-50-FPN	22.9	22.3
Mask R-CNN [10]	ResNet-101-FPN	21.7	20.8
EQL [31]	ResNet-101-FPN	24.2	22.9
BAGS [15]	ResNet-101-FPN	26.5	25.8
AHRL(ours)	ResNet-50-FPN	26.4	25.7
AHRL(ours)	ResNet-101-FPN	**28.7**	**27.6**

5. Conclusions

In this paper, we propose a novel yet effective method from a metric learning perspective to address the long-tailed object detection problem. Our proposed AHRL splits the whole classification feature space into a hierarchical structure and eliminates this tough problem in a coarse-to-fine way. More specifically, AHRL builds the hierarchical structure based on the classification weights of the pre-trained model in the first stage, then AHR loss retains the hierarchical structure and prompts all clusters to repel each other. In addition, according to the relationship between each class pair, an adaptive and dynamic margin mechanism is designed to make similar classes more discriminative. We conduct extensive experiments to verify the effectiveness of our proposed method, and we achieve a new state-of-the-art result on the challenging LVIS dataset based on various backbones without bells and whistles.

6. Broad Impact

Our contributions focus on the hierarchical representation learning for long-tailed object detection, which can be extended to other computer vision tasks. Also, it may provide new ideas for follow-up research. It therefore has the potential to advance both the beneficial and harmful applications of object detectors, such as autonomous vehicles, intelligent video surveillance, robotics and so on. As for ethical aspects and future societal consequences, this technology can bring harmful or beneficial effects to the society, which depends on the citizens who have evil or pure motivation and who can make good use of this technological progress.

References

[1] Hsin-Ping Chou, Shih-Chieh Chang, Jia-Yu Pan, Wei Wei, and Da-Cheng Juan. Remix: Rebalanced mixup. *arXiv preprint arXiv:2007.03943*, 2020.

[2] Peng Chu, Xiao Bian, Shaopeng Liu, and Haibin Ling. Feature space augmentation for long-tailed data. In *European Conf. on Computer Vision (ECCV)*, 2020.

[3] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9268–9277, 2019.

[4] Chris Drummond, Robert C Holte, et al. C4. 5, class imbalance, and cost sensitivity: why under-sampling beats over-sampling. In *Workshop on learning from imbalanced datasets II*, volume 11, pages 1–8. Citeseer, 2003.

[5] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010.

[6] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

[7] Samantha Guerriero, Barbara Caputo, and Thomas Mensink. Deepncm: Deep nearest class mean classifiers, 2018.

[8] Agrim Gupta, Piotr Dollar, and Ross Girshick. Lvis: A dataset for large vocabulary instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5356–5364, 2019.

[9] Hui Han, Wen-Yuan Wang, and Bing-Huan Mao. Borderline-smote: a new over-sampling method in imbalanced data sets learning. In *International conference on intelligent computing*, pages 878–887. Springer, 2005.

[10] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[12] Ting-I Hsieh, Esther Robb, Hwann-Tzong Chen, and Jia-Bin Huang. Droploss for long-tail instance segmentation. In *Proceedings of the Workshop on Artificial Intelligence Safety 2021 (SafeAI 2021) co-located with the Thirty-Fifth AAAI Conference on Artificial Intelligence (AAAI 2021), Virtual, February 8, 2021*, 2021.

[13] Xinting Hu, Yi Jiang, Kaihua Tang, Jingyuan Chen, Chunyan Miao, and Hanwang Zhang. Learning to segment the tail. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14045–14054, 2020.

[14] Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. Decoupling representation and classifier for long-tailed recognition.

[15] Yu Li, Tao Wang, Bingyi Kang, Sheng Tang, Chunfeng Wang, Jintao Li, and Jiashi Feng. Overcoming classifier imbalance for long-tail object detection with balanced group softmax. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10991–11000, 2020.

[16] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.

[17] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

[18] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[19] Jialun Liu, Yifan Sun, Chuchu Han, Zhaopeng Dou, and Wenhui Li. Deep representation learning on long-tailed data: A learnable embedding augmentation perspective. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2970–2979, 2020.

[20] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016.

[21] George A. Miller. Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41, Nov. 1995.

[22] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *arXiv preprint arXiv:1912.01703*, 2019.

[23] S. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert. icarl: Incremental classifier and representation learning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5533–5542, 2017.

[24] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[25] William J Reed. The pareto, zipf and other power laws. *Economics Letters*, 74(1):15–19, 2001.

[26] Jiawei Ren, Cunjun Yu, Shunan Sheng, Xiao Ma, Haiyu Zhao, Shuai Yi, and Hongsheng Li. Balanced meta-softmax for long-tailed visual recognition. *arXiv preprint arXiv:2007.10740*, 2020.

[27] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *arXiv preprint arXiv:1506.01497*, 2015.

[28] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[29] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In I. Guyon, U. V. Luxburg,

S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[30] Jingru Tan, Xin Lu, Gang Zhang, Changqing Yin, and Quanquan Li. Equalization loss v2: A new gradient balance approach for long-tailed object detection. *arXiv e-prints*, pages arXiv–2012, 2020.

[31] Jingru Tan, Changbao Wang, Buyu Li, Quanquan Li, Wanli Ouyang, Changqing Yin, and Junjie Yan. Equalization loss for long-tailed object recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[32] Kaihua Tang, Jianqiang Huang, and Hanwang Zhang. Long-tailed classification by keeping the good and removing the bad momentum causal effect. In *NeurIPS*, 2020.

[33] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[34] Tao Wang, Yu Li, Bingyi Kang, Junnan Li, Junhao Liew, Sheng Tang, Steven Hoi, and Jiashi Feng. The devil is in classification: A simple framework for long-tail instance segmentation. In *European conference on computer vision*, pages 728–744. Springer, 2020.

[35] Tong Wang, Yousong Zhu, Chaoyang Zhao, Wei Zeng, Jinqiao Wang, and Ming Tang. Adaptive class suppression loss for long-tail object detection, 2021.

[36] Xin Wang, Thomas E Huang, Trevor Darrell, Joseph E Gonzalez, and Fisher Yu. Frustratingly simple few-shot object detection. *arXiv preprint arXiv:2003.06957*, 2020.

[37] Jialian Wu, Liangchen Song, Tiancai Wang, Qian Zhang, and Junsong Yuan. Forest r-cnn: Large-vocabulary long-tailed object detection and instance segmentation. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 1570–1578, 2020.

[38] Boyan Zhou, Quan Cui, Xiu-Shen Wei, and Zhao-Min Chen. Bbn: Bilateral-branch network with cumulative learning for long-tailed visual recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9719–9728, 2020.