Distribution Alignment: A Unified Framework for Long-tail Visual Recognition

Songyang Zhang[1,3,5,*] Zeming Li[2] Shipeng Yan[1] Xuming He[1,4] Jian Sun[2]
[1]ShanghaiTech University [2]Megvii Technology [3] University of Chinese Academy of Sciences
[4]Shanghai Engineering Research Center of Intelligent Vision and Imaging
[5]Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Sciences
{zhangsy1, yanshp, hexm}@shanghaitech.edu.cn, {lizeming,sunjian}@megvii.com

Abstract

Despite the recent success of deep neural networks, it remains challenging to effectively model the long-tail class distribution in visual recognition tasks. To address this problem, we first investigate the performance bottleneck of the two-stage learning framework via ablative study. Motivated by our discovery, we propose a unified distribution alignment strategy for long-tail visual recognition. Specifically, we develop an adaptive calibration function that enables us to adjust the classification scores for each data point. We then introduce a generalized re-weight method in the two-stage learning to balance the class prior, which provides a flexible and unified solution to diverse scenarios in visual recognition tasks. We validate our method by extensive experiments on four tasks, including image classification, semantic segmentation, object detection, and instance segmentation. Our approach achieves the state-of-the-art results across all four recognition tasks with a simple and unified framework.

1. Introduction

While deep convolutional networks have achieved great successes in many vision tasks, it usually requires a large number of training examples for each visual category. More importantly, prior research mostly focuses on learning from a *balanced dataset* [23], where different object classes are approximately evenly distributed. However, for large-scale vision recognition tasks, partially due to the non-uniform distribution of natural object classes and varying annotation costs, we typically learn from datasets with a *long-tail* class label distribution. In such scenarios, the number of training instances per class varies significantly, from as few as one example for tail classes to hundreds or thousands for head classes [53, 28, 15, 56, 54, 40].

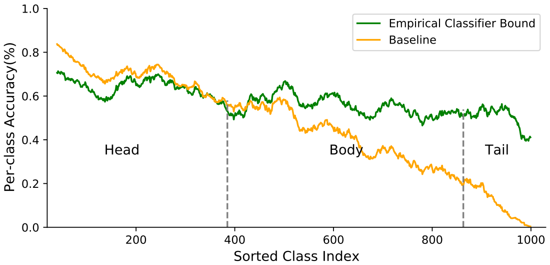

Figure 1: Per-class performance of the two-stage learning *baseline* and our *empirical classification bound* on ImageNet-LT val split. Two methods share the same representation while our bound setting retrains the classifier head with the balanced full dataset.

The intrinsic long-tail property of our visual data introduces a multitude of challenges for recognition in the wild [1], as a deep network model has to simultaneously cope with imbalanced annotations among the head and medium-sized classes, and few-shot learning in the tail classes. A naively learned model would be largely dominated by those few head classes while its performance is much degraded for many other tail classes.

Early works on re-balancing data distribution focus on learning one-stage models, which achieve limited successes due to lack of principled design in their strategies [2, 37, 3, 9, 28, 45]. More recent efforts aim to improve the long-tail prediction by decoupling the representation learning and classifier head learning [20, 30, 39, 42, 24]. However, such a two-stage strategy typically relies on heuristic design to adjust the decision boundary of the initially learned classifier head, which often requires tedious hyper-parameter tuning in practice. This severely limits its capacity to resolve the mismatch between *imbalanced training data distribution* and *balanced evaluation metrics*.

In this work, we first perform an ablative analysis on the two-stage learning strategy to shed light on its performance bottleneck. Specifically, our study estimates an ‘ideal’ clas-

[*] This work was done when Songyang Zhang was a research intern at Megvii Technology. This work was supported by Shanghai NSF Grant (No. 18ZR1425100), National Key R&D Program of China (No. 2017YFA0700800), and Beijing Academy of Artificial Intelligence (BAAI). Code is available: https://github.com/Megvii-BaseDetection/DisAlign

sification accuracy using a balanced dataset to retrain the classifier head while keeping the first-stage representation fixed. Interestingly, as shown in Fig. 6, we find a substantial gap between this ideal performance and the baseline network, which indicates that the first-stage learning with unbalanced data provides a good representation, but there is a large room for improvement in the second stage due to the *biased decision boundary* (See Sec. 3.1 for details).

Based on those findings, we propose a simple and yet effective two-stage learning scheme for long-tail visual recognition problems. Our approach focuses on improving the second-stage training of the classifier after learning a feature representation in a standard manner. To this end, we develop a *unified* distribution alignment strategy to calibrate the classifier output via matching it to a reference distribution of classes that favors the balanced prediction. Such an alignment strategy enables us to exploit the class prior and data input in a principled manner for learning class decision boundary, which eliminates the needs for tedious hyper-parameter tuning and can be easily applied to various visual recognition tasks.

Specifically, we develop a light-weight distribution alignment module for calibrating classification scores, which consists of two main components. In the first component, we introduce an adaptive calibration function that equips the class scores with an input-dependent, learnable magnitude and margin. This allows us to achieve a flexible and confidence-aware distribution alignment for each data point. Our second component explicitly incorporates a balanced class prior by employing a generalized re-weight design for the reference class distribution, which provides a unified strategy to cope with diverse scenarios of label imbalance in different visual recognition tasks.

We extensively validate our model on four typical visual recognition tasks, including image classification on three benchmarks (ImageNet-LT [28], iNaturalist [40] and Places365-LT [28]), semantic segmentation on ADE20k dataset [54], object detection and instance segmentation on LVIS dataset [15]. The empirical results and ablative study show our method consistently outperforms the state-of-the-art approaches on all the benchmarks. To summarize, the main contributions of our works are three-folds:

- We conduct an empirical study to investigate the performance bottleneck of long-tail recognition and reveal a critical gap caused by biased decision boundary.
- We develop a simple and effective distribution alignment strategy with a generalized re-weight method, which can be easily optimized for various long-tail recognition tasks without whistles and bells.
- Our models outperform previous work with a large margin and achieve state-of-the-art performance on long-tail image classification, semantic segmentation, object detection, and instance segmentation.

2. Related Works

One-stage Imbalance Learning To alleviate the adverse effect of the long-tail class distribution in visual recognition, prior work have extensively studied the one-stage methods, which either leverage the re-balancing ideas or explore knowledge transfer from head categories. The basic idea of resample-based methods is to over-sample the minority categories [4, 16] or to under-sample the frequent categories in the training process [11, 2]. Class-aware sampling [37] proposes to choose samples of each category with equal probabilities, which is widely used in vision tasks [28, 12]. Repeat factor sampling [29] is a smoothed sampling method conducting repeated sampling for tail categories, which demonstrates its efficacy in instance segmentation [15]. In addition, [41] proposes to increase the sampling rate for categories with low performance after each training epoch and balances the feature learning for under-privileged categories.

An alternative strategy is to re-weight the loss function in training. Class-level methods typically re-weight the standard loss with category-specific coefficients correlated with the sample distributions [19, 9, 3, 22, 21, 38]. Sample-level methods [26, 34] try to introduce a more fine-grained control of loss for imbalanced learning. Other work aim to enhance the representation or classifier head of tail categories by transferring knowledge from the head classes [45, 44, 28, 51, 8, 47, 46]. Nevertheless, these methods require designing a task specific network module or structure, which is usually non-trivial to be generalized to different vision tasks.

Two-stage Imbalance Learning More recent efforts aims to improve the long-tail prediction by decoupling the learning of representation and classifier head. Decouple [20] proposes an instance-balanced sampling scheme, which generates more generalizable representations and achieves strong performance after properly re-balancing the classifier heads. The similar idea is adopted in [42, 43, 24], which develop effective strategies for long-tail object detection tasks. [30, 39] improve the two-stage ideas by introducing a post-process to adjust the prediction score. However, such a two-stage strategy typically relies on heuristic design in order to adjust the decision boundary of initially learned classifiers and requires tedious hyper-parameter tuning in practice.

Visual Recognition Tasks Visual recognition community has witnessed significant progress with deep convolutional networks in recent years. In this study, we focus on four types of visual tasks, including image classification, object detection, semantic and instance segmentation, which have been actively studied in a large amount of prior work. For object detection, we consider the typical deep network architecture used in the R-CNN series method [14, 13, 35], which detects objects based on the region proposals. For instance segmentation, we take the Mask R-CNN [17] as our example, which extends the Faster R-CNN[35] by adding a branch for predicting the object masks in parallel with the existing branch for bounding box recognition. For the pixel-wise task, semantic segmentation, we use the FCN-based methods [36] and the widely-adopted encoder-decoder structures [7, 5, 6]. Despite those specific choices, we note that our strategy can be easily extended to other types of deep network methods for those visual recognition tasks.

3. Our Approach

Our goal is to address the problem of large-scale long-tail visual recognition, which typically has a large number of classes and severe class imbalance in its training data. To this end, we adopt a two-stage learning framework that first learns a feature representation and a classifier head from the unbalanced data, followed by a calibration stage that adjusts the classification scores. Inspired by our ablative study on existing two-stage methods, we propose a principled calibration method that aligns the model prediction with a reference class distribution favoring the balanced evaluation metrics. Our distribution alignment strategy is simple and yet effective, enabling us to tackle different types of large-scale long-tail visual recognition tasks in a unified framework.

Below we start with a brief introduction to the long-tail classification and an empirical study of two-stage methods in Sec.3.1. We then describe our proposed distribution alignment strategy in Sec.3.2. Finally, we present a comparison with previous methods from the distribution match perspective in Sec.3.3.

3.1. Problem Setting and Empirical Study

We now introduce the problem setting of long-tail classification and review the two-stage learning framework for deep networks. Subsequently, we perform an empirical ablative study on a large-scale image classification task, which motivates our proposed approach.

Problem Definition The task of long-tail recognition aims to learn a classification model from a training dataset with long-tail class distribution. Formally, we denote the input as $\mathbf{I}$, and the target label space as $\mathcal{C} = \{c_1, \cdots, c_K\}$, where K is the number of classes. The classification model $\mathcal{M}$ defines a mapping from the input to the label space: $y = \mathcal{M}(\mathbf{I}; \Theta)$, where $y \in \mathcal{C}$ and Θ are its parameters. Our goal is to learn the model parameter from an imbalanced training dataset $\mathcal{D}_{tr}$ so that $\mathcal{M}$ achieves optimal performance on an evaluation dataset $\mathcal{D}_{eval}$ with respect to certain balanced metrics (*e.g.*, mean accuracy).

In the two-stage framework, we typically consider a deep network model $\mathcal{M}$ with two main components: a feature extractor network $f(\cdot)$ and a classifier head $h(\cdot)$. The feature extractor f first extracts an input representation $\mathbf{x}$, which is then fed into the classifier head h to compute class prediction scores $\mathbf{z}$ as follows:

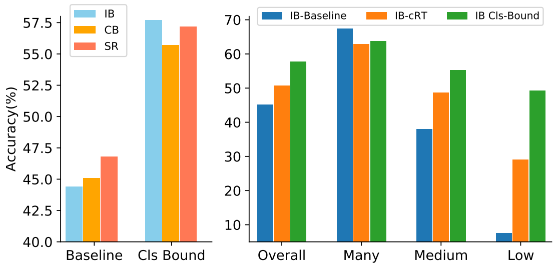

Figure 2: **Empirical analysis of the performance bottleneck.** *Left*: Baseline vs. ideal performance for representations learned with different sampling strategy. *Right*: Comparison of prior arts and ideal performance for the classifier head calibration. **Cls-Bound**: ideal performance bound. **IB**: instance-balanced sampling. **CB**: class-balanced sampling. **SR**: square-root sampling.

$$\mathbf{x} = f(\mathbf{I}, \theta_f) \in \mathbb{R}^d, \quad \mathbf{z} = h(\mathbf{x}, \theta_h) \in \mathbb{R}^K \tag{1}$$

where θ_f and θ_h are the parameter of $f(\cdot)$ and $h(\cdot)$, respectively. Here $\mathbf{z} = \{z_1, \cdots, z_K\}$ indicate the class prediction scores for K classes and the model predicts the class label by taking $y = \arg\max(\mathbf{z})$.

In this work, we instantiate the classifier head h as a linear classifier or a cosine similarity classifier as follows:

$$\text{Linear}: \quad z_j = \mathbf{w}_j^{\intercal}\mathbf{x} \tag{2}$$

$$\text{Cosine Similarity}: \quad z_j = s \cdot \frac{\mathbf{w}_j^{\intercal}\mathbf{x}}{||\mathbf{w}_j||||\mathbf{x}||} \tag{3}$$

where $\mathbf{w}_j \in \mathbb{R}^d$ is the parameter of j-th class and the s is a scale factor as in [33]. We note that the above formulation can be instantiated for multiple visual recognition tasks by changing the input $\mathbf{I}$: e.g., an image for image classification, an image with a pixel location for semantic segmentation, or an image with a bounding box for object detection.

Empirical Analysis on Performance Bound The two-stage learning method tackles the long-tail classification by decoupling the representation and the classifier head learning [20]. Specifically, it first learns the feature extractor f and classifier head h jointly, and then with the representation fixed, re-learns the classifier head with a class balancing strategy. While such design achieves certain success, an interesting question to ask is *which model component(s) impose a bottleneck on its balanced performance*. In the following, we attempt to address the question by exploiting the full set of the ImageNet dataset. Particularly, we follow the decoupling idea to conduct a series of ablative studies on two model components under an 'ideal' balanced setting.

We first investigate whether the feature representation learned on the imbalanced dataset is restrictive for the balanced performance. To this end, we start from learning the feature extractor on the imbalanced ImageNet-LT training set with several re-balancing strategies (*e.g.* instance-balanced, class-balanced, or square-root sampling). We then keep the representation fixed and re-train the classifier head with the ideal balanced ImageNet train set (excluding ImageNet-LT val set). Our results are shown in the left panel of Fig. 2, which indicate that *the first stage produces a strong feature representation that can potentially lead to large performance gain and the instance-based sampling achieves better overall results* (cf. [20]).

Moreover, we conduct an empirical study on the effectiveness of the recent decoupling method (*e.g.* cRT [20]) compared with the above 'ideal' classifier head learning. The right panel of Fig. 2 shows that there remains *a large performance gap* between the existing methods and the upper-bound. Those empirical results indicate that *the biased decision boundary in the feature space seems to be the performance bottleneck of the existing long-tail methods*. Consequently, a better strategy to address this problem would further improve the two-stage learning for the long-tail classification.

3.2. Distribution Alignment

To tackle the aforementioned issue, we now introduce a *unified* distribution alignment strategy to calibrate the classifier output via matching it to a reference distribution of classes that favors the balanced prediction. In this work, we adopt a two-stage learning scheme for all visual recognition tasks, which consists of a joint learning stage and a distribution calibration stage as follows.

1) Joint Learning Stage. The feature extractor $f(\cdot)$ and original classifier head (denoted as $h_o(\cdot)$ for clarity) are jointly learned on imbalanced $\mathcal{D}_{tr}$ with instance-balanced strategy in the first stage, where the original $h_o(\cdot)$ is severely biased due to the imbalanced data distribution.

2) Distribution Calibration Stage. For the second stage, the parameters of $f(\cdot)$ are frozen and we only focus on the classifier head to adjust the decision boundary. To this end, we introduce an *adaptive calibration function* (in Sec. 3.2.1) and a *distribution alignment strategy with generalized re-weighting* (in Sec. 3.2.2) to calibrate the class scores.

3.2.1 Adaptive Calibration Function

To learn the classifier head $h(\cdot)$ in the second stage, we propose an adaptive calibration strategy that fuses the original classifier head $h_o(\cdot)$ (parameters of $h_o(\cdot)$ are frozen) and a learned class prior in an input-dependent manner. As shown below, unlike previous work (*e.g.* cRT[20]), our design does not require a re-training of the classifier head from scratch and has much fewer free parameters. This enables us to reduce the adverse impact from the limited training data of the tail categories. Moreover, we introduce a flexible fusion mechanism capable of controlling the magnitude of calibration based on input features.

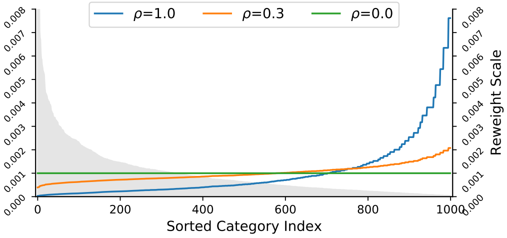

Figure 3: The category frequency r_i is plotted in gray color and the right axis denotes the re-weight coefficient at different scale ρ on ImageNet-LT dataset.

Specifically, denote the class scores from $h_o(\cdot)$ as $\mathbf{z}^o = [z_1^o, \cdots, z_K^o]$, we first introduce a class-specific linear transform to adjust the score as follows:

$$s_j = \alpha_j \cdot z_j^o + \beta_j, \quad \forall j \in \mathcal{C} \tag{4}$$

where α_j and β_j are the calibration parameters for each class, which will be learned from data. As mentioned above, we then define a confidence score function $\sigma(\mathbf{x})$ to adaptively combine the original and the transformed class scores:

$$\begin{aligned} \hat{z}_j &= \sigma(\mathbf{x}) \cdot s_j + (1 - \sigma(\mathbf{x})) \cdot z_j^o \qquad (5) \\ &= (1 + \sigma(\mathbf{x})\alpha_j) \cdot z_j^o + \sigma(\mathbf{x}) \cdot \beta_j \qquad (6) \end{aligned}$$

where the confidence score has a form of $g(\mathbf{v}^\intercal \mathbf{x})$, which is implemented as a linear layer followed by a non-linear activation function (*e.g.*, sigmoid function) for all input $\mathbf{x}$. The confidence $\sigma(\mathbf{x})$ controls how much calibration is needed for a specific input $\mathbf{x}$. Given the calibrated class scores, we finally define a prediction distribution for our model with the Softmax function:

$$p_m(y = j|\mathbf{x}) = \frac{\exp(\hat{z}_j)}{\sum_{k=1}^{C} \exp(\hat{z}_k)}. \tag{7}$$

3.2.2 Alignment with Generalized Re-weighting

Given a train dataset $\mathcal{D}_{tr} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$, we introduce a calibration strategy based on distribution alignment between our model prediction $p_m(\cdot)$ and a reference distribution of classes that favors the balanced prediction.

Formally, denote the reference distribution as $p_r(y|\mathbf{x})$, we aim to minimize the expected KL-divergence between

Method	**Align Type**	**Top-1 Accuracy@R-50**				**Top-1 Accuracy@X-50**			
		Average	**Many**	**Medium**	**Few**	**Average**	**Many**	**Medium**	**Few**
Baseline[20]	-	41.6	64.0	33.8	5.8	44.4	65.9	37.5	7.7
Baseline*	-	48.4	68.4	41.7	15.2	49.2	68.9	42.8	15.6
NCM[20]	Hand-Craft	44.3	53.1	42.3	26.5	47.3	56.6	45.3	28.1
τ-Norm[20]		46.7	56.6	44.2	27.4	49.4	59.1	46.9	30.7
Logit Adjust(post)[30]		50.4	-	-	-	-	-	-	-
Deconfound*[39]		-	-	-	-	51.8	62.7	48.8	31.6
cRT[20]	Learnable	47.3	58.8	44.0	26.1	49.6	61.8	46.2	27.4
cRT*[20]		-	-	-	-	49.7	60.4	46.8	29.3
LWS[20]		47.7	57.1	45.2	29.3	49.9	60.2	47.2	30.3
DisAlign		51.3	59.9	49.9	31.8	52.6	61.5	50.7	33.1
DisAlign*		**52.9**	61.3	52.2	31.4	**53.4**	62.7	52.1	31.4

Table 1: **Quantitative results on ImageNet-LT.** $*$ denotes the model uses cosine classifier. **R-50** and **X-50** means the ResNet-50 and ResNeXt-50, respectively.

$p_r(y|\mathbf{x})$ and the model prediction $p_m(y|\mathbf{x})$ as follows:

$$\mathcal{L} = \mathbb{E}_{\mathcal{D}_{tr}}\left[\mathcal{KL}(p_r(y|\mathbf{x})||p_m(y|\mathbf{x}))\right] \tag{8}$$

$$\approx -\frac{1}{N}\sum_{i=1}^{N}\left[\sum_{y\in\mathcal{C}} p_r(y|\mathbf{x}_i)\log(p_m(y|\mathbf{x}_i))\right] + C \tag{9}$$

where the expectation is approximated by an empirical average on $\mathcal{D}_{tr}$ and C is a constant.

In this work, we adopt a re-weighting approach [9] and introduce a generalized re-weight strategy for the alignment in order to exploit the class prior. Formally, we represent the reference distribution as a weighted empirical distribution on the training set,

$$p_r(y = c|\mathbf{x}_i) = w_c \cdot \delta_c(y_i), \quad \forall c \in \mathcal{C} \tag{10}$$

where w_c is the class weight, and $\delta_c(y_i)$ is the Kronecker delta function(equals 1 if $y_i = c$, otherwise equals 0). We then define the reference weight based on the empirical class frequencies $\mathbf{r} = [r_1, \cdots, r_K]$ on the training set:

$$w_c = \frac{(1/r_c)^\rho}{\sum_{k=1}^{K}(1/r_k)^\rho}, \quad \forall c \in \mathcal{C} \tag{11}$$

where ρ is a scale hyper-parameter to provide more flexibility in encoding class prior. Note that our scheme reduces to the instance-balance re-weight method with $\rho = 0$, and to the class-balanced re-weight method with $\rho = 1$. We illustrate the curve of re-weight coefficients based on ImageNet-LT dataset in Fig. 3.

3.3. Connection with Recent Work

Below we discuss the connections between our proposed distribution alignment strategy and recent two-stage methods. Detailed comparison is reported in Tab. 2. Notably, Logit Adjustment[30] and Deconfound[39] introduce a hand-craft margin to adjust the distribution while keep the magnitude as 1.0, and incorporate the class prior directly in r_i or $\mathbf{w}_i$ without re-training. LWS[20] and τ-normalized[20] try to achieve a similar goal by learning a magnitude scale and discarding the margin adjustment.

Method	**Align Method**			
	Type	Balance	Magnitude	Margin
Joint	-	-	-	-
LWS[20]	L	CB-RS	α_j	0
τ-Normalized[20]	H	CB-RS	$1/\|\|\mathbf{w}_j\|\|^\tau$	0
Logit Adjust[30]	H	-	1.0	$-\lambda\log(r_j)$
Deconfound*[39]	H	-	1.0	$-\lambda d(\mathbf{x},\mathbf{e})\mathbf{w}_j^\intercal\mathbf{e}$
DisAlign	L	G-RW	$1+\sigma(\mathbf{x})\alpha_j$	$\sigma(\mathbf{x})\beta_j$
DisAlign*	L	G-RW	$1+\sigma(\mathbf{x})\alpha_j$	$\sigma(\mathbf{x})\beta_j$

Table 2: **Comparison with related methods.** $*$ denotes cosine classifier, **L**: learnable, **H**:hand-craft, **CB-RS**: class-balanced resampling, **G-RW**: generalized re-weight, r_j: class frequency for the j-th class, λ: hypper-parameter, $\mathbf{e}$: mean feature of training data, $d(\cdot)$: cosine distance.

All these methods can be considered as the special cases of our DisAlign approach, which provides a unified and simple form to model the distribution mismatch in a learnable way. Moreover, the resample based strategy is not easy to be applied for the instance-level (object detection/instance segmentation) or pixel-level (semantic segmentation) tasks, our generalized re-weight provides an alternative solution to incorporate the class prior in a simple and effective manner. Experimental results in Sec. 4 also demonstrate the strength of our method compared with the aforementioned works.

4. Experiments

In this section, we conduct a series of experiments to validate the effectiveness of our method. Below we present our experimental analysis and ablation study on the image

Method	ResNet-50		ResNet-152	
	90 E	200 E	90 E	200 E
LDAM[3]	68.0	-	-	-
Baseline	61.7	65.8	65.0	69.0
Baseline*	64.8	66.2	67.3	69.0
cRT[20]	65.2	68.2	68.5	71.2
τ-norm[20]	65.6	69.3	68.8	72.5
LWS[20]	65.9	69.5	69.1	72.1
BBN[52]	66.3	69.6	-	-
DisAlign	67.8	**70.6**	71.3	**74.1**
DisAlign*	**69.5**	70.2	**71.7**	72.8

Table 3: **Average accuracy on iNaturalist-2018.** $*$ denotes the cosine classifier.

Method	ResNet-152			
	Average	Many	Medium	Few
Focal Loss[28]	34.6	41.1	34.8	22.4
Range Loss[28]	35.1	41.1	35.4	23.2
OLTR[28]	35.9	44.7	37.0	25.3
Feature Aug[8]	36.4	42.8	37.5	22.7
Baseline	30.2	**45.7**	27.3	8.2
NCM	36.4	40.4	37.1	27.3
cRT	36.7	42.0	37.6	24.9
LWS	37.6	40.6	39.1	28.6
τ-norm	37.9	37.8	40.7	**31.8**
DisAlign	**39.3**	40.4	**42.4**	30.1

Table 4: **Results on Place365-LT with ResNet-152.**

classification task in Sec. 4.1, followed by our results on semantic segmentation task in Sec. 4.2. In addition, we further evaluate our methods on object detection and instance segmentation tasks in Sec. 4.3.

4.1. Image Classification

Experimental Details To demonstrate our methods, we conduct experiments on three large-scale long-tail datasets, including ImageNet-LT [28], iNaturalist 2018 [40], and Places-LT [28]. We follow the experimental setting and implementation of [20] [1]. For the ImageNet-LT dataset, we report performance with ResNet/ResNeXt-{50,101,152} as backbone, and mainly use ResNet-50 for ablation study. For iNaturalist 2018 and Places-LT, our comparisons are performed under the settings of ResNet-{50,101,152}.

Comparison with previous methods 1) ImageNet-LT. We present the quantitative results for ImageNet-LT in Tab. 10. Our approach achieves **52.9%** in per-class average accuracy based on ResNet-50 backbone and **53.4%** based on ResNeXt-50, which outperform the state-of-the-art methods by a significant margin of **2.5%** and **1.6%**, respectively. **2) iNaturalist.** In Tab. 13, our method DisAlign with cosine classifier achieves **69.5%** per-class average accuracy using ResNet-50 backbone and 90 epochs of training, surpassing the prior art LDAM by a large margin at **1.5%**. It also shows that our performance can be further improved with larger backbone and/or more training epochs. **3) Places-LT.** In Tab. 14, we show the experimental results under the same setting as [20] on Places-LT. Our method achieves **39.3%** per-class average accuracy based on ResNet-152, with a notable performance gain at **1.4%** over the prior methods. We also report the detailed performance of these three datasets with ResNet-{50,101,152} in the supplementary materials.

GR	MT	MG	Average	Many	Medium	Few
✗	✗	✗	41.6	64.0	33.8	5.8
✓	✓	✗	50.1	60.4	48.0	28.8
✓	✗	✓	49.9	63.9	46.9	21.2
✓	✓	✓	51.3	59.9	49.9	31.8

Table 5: **Ablation study of DisAlign. GR** means the generalized reweight strategy. **MT** means the learnable magnitude parameter $(1+\sigma(\mathbf{x})\alpha)$ and **MG** is the learnable margin parameter $\sigma(\mathbf{x})\beta$.

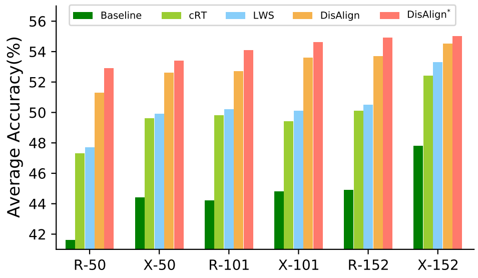

Figure 4: **Performance of DisAlign with different backbone on ImageNet-LT.** Detailed results will be reported in supplementary material.

Ablation Study 1) Different Backbone: We validate our method on different types of backbone networks, ranging from ResNet-{50,101,152} to ResNeXt-{50, 101, 152}, reported in Fig. 4. Our method achieves **54.9%** with ResNet-152, and **55.0%** with ResNeXt-152. It's worth noting that even when adopting stronger backbones, the gain of DisAlign compared to the state-of-the-art methods is still significant. This demonstrates that our DisAlign is complementary to the capacity of backbone networks. **2) Model Components:** We conduct a series of ablation studies to evaluate the importance of each component used in our DisAlign method. Tab. 5 summarizes the results of our ablation experiments, in which we compare our full model with several partial model settings. From the table, we find the learnable magnitude has a significant improvement compared with

[1] Detailed configuration and results are provided in the supplementary materials.

Framework	Method	B	Mean IoU(%)				Mean Accuracy(%)			
			Average	Head	Body	Tail	Average	Head	Body	Tail
FCN[36]	Baseline	R-50	38.1	64.6	40.0	29.6	46.3	78.6	49.3	35.4
	DisAlign		40.1(+2.0)	65.0(+0.4)	42.8(+2.8)	31.3(+1.7)	51.4(+5.1)	78.6(+0.0)	56.1(+6.8)	40.6(+5.2)
	Baseline	R-101	41.4	67.0	43.3	33.2	50.2	80.6	52.9	40.1
	DisAlign		43.7(+2.3)	67.4(+0.4)	46.1(+2.8)	35.7(+2.5)	55.9(+5.7)	80.6(+0.0)	59.7(+6.8)	46.4(+6.3)
	Baseline	S-101	46.2	67.6	48.0	39.1	57.3	79.4	61.7	48.2
	DisAlign		46.9(+0.7)	67.7(+0.1)	48.2(+0.2)	40.3(+1.2)	60.1(+2.8)	79.7(+0.3)	64.2(+2.5)	51.9(+3.7)
DeepLabV3$^+$[7]	Baseline	R-50	44.9	67.7	48.3	36.4	55.0	80.1	60.8	44.1
	DisAlign		45.7(+0.8)	67.7(+0.0)	48.6(+0.3)	37.8(+1.4)	57.3(+2.3)	80.8(+0.7)	63.0(+2.2)	46.9(+2.8)
	Baseline	R-101	46.4	68.7	49.0	38.4	56.7	80.9	61.5	46.7
	DisAlign		47.1(+0.7)	68.7(+0.0)	49.4(+0.4)	39.6(+1.2)	59.5(+2.8)	81.4(+0.5)	64.2(+2.7)	50.3(+3.6)
	Baseline	S-101	47.3	69.0	49.7	39.7	58.1	80.8	63.4	48.2
	DisAlign		47.8(+0.5)	68.9(-0.1)	49.8(+0.1)	40.7(+1.0)	60.1(+2.0)	81.0(+0.2)	65.5(+2.1)	52.0(+3.8)

Table 6: **Performance of semantic segmentation on ADE-20K:** All baseline models are trained with an image size of 512×512 and 160K iteration in total. **B** is backbone network(R-50, R-101, S-101 denote ResNet-50, ResNet-101 and ResNeSt-101, respectively).

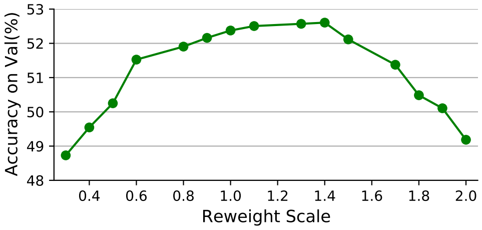

Figure 5: **Effects of the different generalized re-weight scale.** Performance is reported on ImageNet-LT val split.

baseline and the learnable margin also achieves competitive results at 49.9%, which demonstrate the effectiveness of individual modules in our design. **3) Generalized Re-weight Scale** We also investigate the influence of the generalized re-weight scale on the validation set of ImageNet-LT and plot the accuracy-scale curve in Fig. 5. It is evident that adjusting generalized reweight is able to achieve significant performance improvement. Moreover, we find the setting of $\rho > 1$ is able to outperform the class-balanced re-weight ($\rho = 1$), which indicates that the generalized re-weight is more effective in coping with long-tail distributions.

B	Method	Mask R-CNN		Cascade R-CNN	
		$\mathbf{AP}_{bbox}$	$\mathbf{AP}_{mask}$	$\mathbf{AP}_{bbox}$	$\mathbf{AP}_{mask}$
R-50	Baseline	20.8	21.2	25.2	23.0
	DisAlign	23.9(+3.1)	24.2(+3.0)	28.7(+3.5)	26.1(+3.1)
	Baseline*	22.8	23.8	28.8	26.2
	DisAlign*	25.6(+2.8)	26.3(+2.5)	32.2(+3.4)	29.4(+3.2)
R-101	Baseline	22.2	22.6	26.1	24.0
	DisAlign	25.6(+3.4)	25.8(+3.2)	29.7(+3.6)	27.3(+3.3)
	Baseline*	24.5	25.1	30.4	28.1
	DisAlign*	27.5(+3.0)	28.2(+3.1)	33.7(+3.3)	30.9(+2.8)
X-101	Baseline	24.5	25.0	28.4	26.1
	DisAlign	26.8(+2.3)	27.4(+2.4)	31.3(+2.9)	28.7(+2.6)
	Baseline*	26.9	27.7	32.6	29.8
	DisAlign*	29.5(+2.6)	30.0(+2.3)	34.7(+2.1)	31.8(+2.0)

Table 7: **Results on LVIS v0.5 dataset with different backbones and different architectures.** The results are reported based on the Detectron2[48, 55] framework. We refer the reader to the supplementary material for the detailed comparison with the state of art.

4.2. Semantic Semgnetaion on ADE20k Dataset

To further validate our method, we apply DisAlign strategy to segmentation networks and report our performance on the semantic segmentation benchmark, ADE20k [54].

Dataset and Implementation Details Follow a similar protocol as in image classification, we divide the 150 categories into 3 subsets according to the percentage of pixels in every category over the entire dataset. Specifically, we define three disjoint subsets as follows: *head classes* (each with more than 1.0% of total pixels), *body classes* (each with a percentage ranging from 0.1% to 1% of total pixels) and *tail classes* (each with less than 0.1% of total pixels). [2]

Quantitative Results We evaluate our method using two widely-adopted segmentation models (FCN [36] and DeepLabV3$^+$ [7]) based on different backbone networks, ranging from ResNet-50, ResNet-101 to the latest ResNeSt-101, and report the performance in Tab. 6. Our method achieves **2.0** and **2.3** improvement in mIoU using FCN-8s with ResNet-50 and ResNet-101, respectively. The performance on the body and tail are improved significantly. Moreover, our method outperforms the baseline with large margin at **5.7** in mean accuracy with ResNet-101 backbone. Even with a stronger backbone: **ResNeSt-101** [50], our method also achieves **0.7** mIoU and **2.8** improvement in mean accu-

[2] The complete list of the split is reported in supplementary material.

Pre-Train	**Method**	**BBox AP**				**Mask AP**			
		AP$_{bbox}$	**AP**$^{r}_{bbox}$	**AP**$^{c}_{bbox}$	**AP**$^{f}_{bbox}$	**AP**$_{mask}$	**AP**$^{r}_{mask}$	**AP**$^{c}_{mask}$	**AP**$^{f}_{mask}$
ImageNet	Baseline	20.8	3.3	19.5	29.4	21.2	3.7	21.6	28.4
	Baseline*	22.8	10.2	21.1	30.1	23.8	11.5	23.7	28.9
	Focal Loss[26]	21.9	-	-	-	21.0	9.3	21.0	25.8
	SimCal[42]	22.6	13.7	20.6	28.7	23.4	16.4	22.5	27.2
	LST[18]	22.6	-	-	-	23.0	-	-	-
	RFS[15]	23.6	12.8	22.3	29.4	24.3	14.6	24.0	28.5
	EQL[38]	23.3	-	-	-	22.8	11.3	24.7	25.1
	DisAlign	23.9	7.5	25.0	29.1	24.3	8.5	26.3	28.1
	DisAlign*	**25.6**	13.7	25.6	30.5	**26.3**	14.9	27.6	29.2
COCO	Baseline	22.8	2.6	21.8	32.0	23.9	2.8	23.4	30.5
	Baseline*	25.0	10.2	23.9	32.3	25.3	11.0	25.5	30.7
	GroupSoftmax[24]	25.8	15.0	25.5	30.4	26.3	18.0	26.9	28.7
	DisAlign	25.5	8.2	26.3	32.4	25.7	9.4	27.6	29.7
	DisAlign*	**27.6**	14.8	27.9	32.4	**27.9**	16.2	29.3	30.8

Table 8: **Comparison with the-state-of-art on LVIS with Mask-R-CNN-FPN(ResNet-50 backbone).** All results are evaluated on the LVIS v0.5 validation set with the score threshold at 0.0001. ($*$ denotes cosine classifier for bbox classification.)

Backbone	**Method**	**BBox AP**				**Mask AP**			
		AP$_{bbox}$	**AP**$^{r}_{bbox}$	**AP**$^{c}_{bbox}$	**AP**$^{f}_{bbox}$	**AP**$_{mask}$	**AP**$^{r}_{mask}$	**AP**$^{c}_{mask}$	**AP**$^{f}_{mask}$
ResNet-50	Baseline*	26.5	8.7	25.0	36.0	23.5	8.1	22.4	31.5
	DisAlign*	30.5	17.9	30.1	36.5	27.0	15.7	27.0	31.9
ResNet-101	De-confound[39]	25.8	-	-	-	23.5	5.2	22.7	32.3
	De-confound TDE[39]	30.0	-	-	-	27.1	16.0	26.9	32.1
	Baseline*	28.9	11.8	27.7	37.8	25.6	10.5	24.9	33.0
	DisAlign*	32.7	20.5	32.8	38.1	28.9	18.0	29.3	33.3
ResNeXt-101	Baseline*	30.7	14.2	29.3	39.6	27.3	13.0	26.4	34.6
	DisAlign*	33.7	21.4	33.1	39.7	29.7	18.4	29.7	34.7

Table 9: **Results on LVIS v1.0 dataset with Cascade R-CNN.** * denotes cosine classifier head.

racy, where the tail categories have a performance gain of **1.2** in mIoU and **3.7** in mean accuracy. We further validate our method using DeepLabV3$^+$, which is a more powerful semantic segmentation model. Our DisAlign improves the performance of DeepLabV3$^+$ by a margin of **0.5** based on ResNeSt-101, which achieves the new state-of-the-art (**47.8** in mIoU) on the ADE20k dataset.

4.3. Object Detection and Instance Segmentation

Experimental Configuration We conduct experiments on LVIS [15] dataset. For evaluation, we use a COCO-style average precision (AP) metric that averages over categories and different box/mask IoU threshold [27].

Quantitative Results and Ablation Study We first compare our method with recent work and report quantitative results in Tab. 8. We find our DisAlign with cosine classifier head achieves **25.6** in AP$_{bbox}$, and **26.3** in AP$_{mask}$ when applied to the Mask R-CNN+FPN with the ImageNet pre-trained ResNet-50 backbone. Moreover, our strategy can be further improved to achieve **27.6** in AP$_{bbox}$ and **27.9** in AP$_{mask}$ based on the COCO pre-trained model. In both cases, our method is able to maintain the performance of the frequent (also called *head*) categories, and gain significant improvement on common (also called *body*) and rare (also called *tail*) categories. We also report performance with more power detection framework (*e.g.*Cascade R-CNN) and stronger backbones (*e.g.* ResNet-50/101, and ResNeXt-101) in Tab. 7 and Tab. 9. It is worth noting that even with the stronger backbones or frameworks, the performance gain of our DisAlign over the baseline is still significant.

5. Conclusion

In this paper, we have presented a unified two-stage learning strategy for the large-scale long-tail visual recognition tasks. To tackle the biased label prediction, we develop a confidence-aware distribution alignment method to calibrate initial classification predictions. In particular, we design a generalized re-weight scheme to leverage the category prior for the alignment process. Extensive experiments show that our method outperforms previous works with a large margin on a variety of visual recognition tasks(image classification, semantic segmentation, and object detection/segmentation).

References

[1] Samy Bengio. Sharing representations for long tail computer vision problems. In *Proceedings of the ACM on International Conference on Multimodal Interaction*, 2015. 1

[2] Mateusz Buda, Atsuto Maki, and Maciej A Mazurowski. A systematic study of the class imbalance problem in convolutional neural networks. *Neural Networks*, 2018. 1, 2

[3] Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. In *Advances in Neural Information Processing Systems(NeurIPS)*, 2019. 1, 2, 6

[4] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 2002. 2

[5] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence(TPAMI)*, 2018. 3

[6] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint*, 2017. 3

[7] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European Conference on Computer Vision(ECCV)*, 2018. 3, 7, 13, 14

[8] Peng Chu, Xiao Bian, Shaopeng Liu, and Haibin Ling. Feature space augmentation for long-tailed data. In *Proceedings of the European Conference on Computer Vision(ECCV)*, 2020. 2, 6

[9] 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(CVPR)*, 2019. 1, 2, 5

[10] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR)*, 2009. 11

[11] 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*, 2003. 2

[12] Yuan Gao, Xingyuan Bu, Yang Hu, Hui Shen, Ti Bai, Xubin Li, and Shilei Wen. Solution for large-scale hierarchical object detection datasets with incomplete annotation and data imbalance. *arXiv preprint*, 2018. 2

[13] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision(ICCV)*, 2015. 2

[14] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR)*, 2014. 2

[15] 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(CVPR)*, 2019. 1, 2, 8, 13

[16] 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*, 2005. 2

[17] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision(ICCV)*, 2017. 2

[18] 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(CVPR)*, 2020. 8

[19] Chen Huang, Yining Li, Chen Change Loy, and Xiaoou Tang. Learning deep representation for imbalanced classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR)*, 2016. 2

[20] Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. Decoupling representation and classifier for long-tailed recognition. *International Conference on Learning Representations(ICLR)*, 2020. 1, 2, 3, 4, 5, 6, 11

[21] Salman Khan, Munawar Hayat, Syed Waqas Zamir, Jianbing Shen, and Ling Shao. Striking the right balance with uncertainty. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR)*, 2019. 2

[22] Salman H Khan, Munawar Hayat, Mohammed Bennamoun, Ferdous A Sohel, and Roberto Togneri. Cost-sensitive learning of deep feature representations from imbalanced data. *IEEE Transactions on Neural Networks and Learning Systems(TNNLS)*, 2017. 2

[23] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems(NeurIPS)*, 2012. 1

[24] 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(CVPR)*, 2020. 1, 2, 8

[25] 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/CVF Conference on Computer Vision and Pattern Recognition(CVPR)*, 2017. 13

[26] 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(ICCV)*, 2017. 2, 8

[27] 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 *Proceedings of the European Conference on Computer Vision(ECCV)*, 2014. 8, 13

[28] Ziwei Liu, Zhongqi Miao, Xiaohang Zhan, Jiayun Wang, Boqing Gong, and Stella X Yu. Large-scale long-tailed recognition in an open world. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR)*, 2019. 1, 2, 6, 11

[29] Dhruv Mahajan, Ross Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens van der Maaten. Exploring the limits of weakly supervised pretraining. In *Proceedings of the European Conference on Computer Vision(ECCV)*, 2018. 2

[30] Aditya Krishna Menon, Sadeep Jayasumana, Ankit Singh Rawat, Himanshu Jain, Andreas Veit, and Sanjiv Kumar. Long-tail learning via logit adjustment. *arXiv preprint*, 2020. 1, 2, 5

[31] Open MMLab. Mmsegmentation. https://github.com/open-mmlab/mmsegmentation, 2020. 12

[32] 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. In *Advances in Neural Information Processing Systems(NeurIPS)*, 2019. 11

[33] Hang Qi, Matthew Brown, and David G Lowe. Low-shot learning with imprinted weights. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR)*, 2018. 3

[34] Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. Learning to reweight examples for robust deep learning. In *International Conference on Machine Learning(ICML)*, 2018. 2

[35] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems(NeurIPS)*, 2015. 2, 3

[36] Evan Shelhamer, Jonathan Long, and Trevor Darrell. Fully convolutional networks for semantic segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence(TPAMI)*, 2017. 3, 7, 13, 14

[37] Li Shen, Zhouchen Lin, and Qingming Huang. Relay backpropagation for effective learning of deep convolutional neural networks. In *Proceedings of the European Conference on Computer Vision(ECCV)*, 2016. 1, 2

[38] 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)*, 2020. 2, 8

[39] Kaihua Tang, Jianqiang Huang, and Hanwang Zhang. Long-tailed classification by keeping the good and removing the bad momentum causal effect. In *Advances in Neural Information Processing Systems(NeurIPS)*, 2020. 1, 2, 5, 8

[40] Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alex Shepard, Hartwig Adam, Pietro Perona, and Serge Belongie. The inaturalist species classification and detection dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR)*, 2018. 1, 2, 6, 11

[41] Hao Wang, Qilong Wang, Fan Yang, Weiqi Zhang, and Wangmeng Zuo. Data augmentation for object detection via progressive and selective instance-switching, 2019. 2

[42] 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 *Proceedings of the European Conference on Computer Vision(ECCV)*, 2020. 1, 2, 8

[43] Xin Wang, Thomas E. Huang, Trevor Darrell, Joseph E Gonzalez, and Fisher Yu. Frustratingly simple few-shot object detection. In *International Conference on Machine Learning(ICML)*, 2020. 2

[44] Yu-Xiong Wang, Ross Girshick, Martial Hebert, and Bharath Hariharan. Low-shot learning from imaginary data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR)*, 2018. 2

[45] Yu-Xiong Wang, Deva Ramanan, and Martial Hebert. Learning to model the tail. In *Advances in Neural Information Processing Systems(NeurIPS)*, 2017. 1, 2

[46] Jialian Wu, Chunluan Zhou, Qian Zhang, Ming Yang, and Junsong Yuan. Self-mimic learning for small-scale pedestrian detection. In *Proceedings of the 28th ACM International Conference on Multimedia(ACM MM)*, 2020. 2

[47] Tz-Ying Wu and Pedro Morgado. Solving long-tailed recognition with deep realistic taxonomic classiï¬ er. In *European Conference on Computer Vision(ECCV)*, 2020. 2

[48] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. https://github.com/facebookresearch/detectron2, 2019. 7, 13

[49] Hang Zhang, Kristin Dana, Jianping Shi, Zhongyue Zhang, Xiaogang Wang, Ambrish Tyagi, and Amit Agrawal. Context encoding for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR)*, 2018. 12

[50] Hang Zhang, Chongruo Wu, Zhongyue Zhang, Yi Zhu, Zhi Zhang, Haibin Lin, Yue Sun, Tong He, Jonas Muller, R. Manmatha, Mu Li, and Alexander Smola. Resnest: Split-attention networks. *arXiv preprint*, 2020. 7, 12

[51] Yaoyao Zhong, Weihong Deng, Mei Wang, Jiani Hu, Jianteng Peng, Xunqiang Tao, and Yaohai Huang. Unequal-training for deep face recognition with long-tailed noisy data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, 2019. 2

[52] 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(CVPR)*, 2020. 6

[53] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE transactions on pattern analysis and machine intelligence(TPAMI)*, 2017. 1

[54] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR)*, 2017. 1, 2, 7, 14

[55] Benjin Zhu*, Feng Wang*, Jianfeng Wang, Siwei Yang, Jianhu Chen, and Zeming Li. cvpods: All-in-one toolbox for computer vision research, 2020. 7

[56] Xiangxin Zhu, Dragomir Anguelov, and Deva Ramanan. Capturing long-tail distributions of object subcategories. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR)*, 2014. 1

A. Experiments of Image Classification

In this section, we first introduce the dataset and evaluation metrics for image classification task in Sec.A.1. Then the training configuration will be detailed in Sec.A.2, followed by results on three benchmarks in Sec.A.3.

A.1. Dataset and Evaluation Metrics

Datasets To demonstrate our methods, we conduct experiments on three large-scale long-tailed datasets, including Places-LT[28], ImageNet-LT[28], and iNaturalist 2018[40]. Places-LT and ImageNet-LT are artificially generated by sampling a subset from their balanced versions (Places-365[28] and ImageNet-2012[10]) following the *Parento distribution*. iNaturalist 2018 is a real-world, naturally long-tailed dataset, consisting of samples from 8,142 species.

Evaluation Metrics We report the class-balanced average *Top-1* accuracy on the corresponding validation/test set, and also calculate the accuracy of three disjoint subsets, 'Many', 'Medium' and 'Few', which are defined according to the amount of training data per class [20].

A.2. Training Configuration

Configuration Detail Following [20], we use PyTorch[32] framework for all experiments. For *ImageNet-LT*, we report performance with ResNet-{50,101,152} and ResNeXt-{50,101,152} and mainly use ResNet-50 for ablation study. For *iNaturalist 2018*, performance is reported with ResNet-{50,101,152}. For *Places-LT*, ResNet-152 is used as backbone and we pre-train it on the full ImageNet-2012 dataset.

We use the SGD optimizer with momentum 0.9, batch size 256, cosine learning rate schedule gradually decaying from 0.1 to 0, and image resolution 224×224. For the joint learning stage, the backbone network and original classifier head are jointly trained with 90 epochs for ImageNet-LT, and 90/200 epochs for iNaturalist-2018. For the Places-LT dataset, the models are trained with 30 epochs with the all layers frozen expect the last ResNet block in the first stage.

Implementation of Our Method In the second distribution alignment stage, we restart the learning rate and train it for 10/30 epochs as [20] while keeping the backbone network and original classifier head fixed(10 epochs for ImageNet-LT and Places-LT, 30 epochs for iNaturalist-2018). For all three datasets, we set the generalized re-weight scale $\rho = 1.2$ for dot-product classifier head, $\rho = 1.5$ for cosine normalized classifier head. The α and β are initialized with 1.0 and 0.0, respectively.

A.3. Detailed Experimental Results

ImageNet-LT. We present the detailed quantitative results for ImageNet-LT in Table 10.

iNaturalist and Places-LT. To further demonstrate our method, we conduct experiments on two extra large-scale long-tail benchmarks and report the performance in Table 13 and Table 14.

A.4. Ablation Study

Influence of Model Components We report an ablation study of the two main components of our method with ResNeXt-50 in Tab. 11, which shows that both adaptive calibration and generalized re-weighting(G-RW) contribute to the performance improvement of our approach.

Analysis of the Calibration We plot the learned magnitude and margin according to the class sizes below. They share a similar trend, in which the tail/body classes have larger value than head. Thus our calibration alleviates the bias in the original prediction by boosting the tail scores.

Confidence Score We study confidence-based calibration in the table below, which shows that the input-aware calibration outperforms the input-agnostic counterpart and the baselines using only magnitude or margin. We also observe that the example whose biased prediction probability is low on its ground-truth class tends to be improved with higher confidence.

B. Experiments of Semantic Segmentation

Similar to image classification, the large-scale semantic segmentation task still suffers from the long-tail data distribution. To further validate the effectiveness of our method, we also apply DisAlign on large-scale semantic segmentation benchmark: ADE-20k.

B.1. Dataset and Evaluation

Dataset. ADE20K dataset is a scene parsing benchmark, which contains 150 stuff/object categories. The dataset includes 20K/2K/3K images for training, validation, and testing. Compared with the image classification[28], the imbalance of ADE20K is more serve than the image classification, which has an imbalance ratio of **788**(Max/Min). Follow the similar protocol in image classification, we divide the 150

Backbone	Method	ResNet				ResNeXt			
		Average	Many	Medium	Few	Average	Many	Medium	Few
*-50	Baseline	41.6	64.0	33.8	5.8	44.4	65.9	37.5	7.7
	Baseline*	48.4	68.4	41.7	15.2	49.2	68.9	42.8	15.6
	DisAlign	51.3	59.9	49.9	31.8	52.6	61.5	50.7	33.1
	DisAlign*	52.9	61.3	52.2	31.4	53.4	62.7	52.1	31.4
*-101	Baseline	44.2	66.6	36.8	7.1	44.8	66.2	37.8	8.6
	Baseline*	49.5	69.3	43.1	15.9	50.0	69.9	43.7	15.9
	DisAlign	52.7	61.7	51.1	32.4	53.6	63.3	51.2	34.6
	DisAlign*	54.1	63.2	53.1	31.9	54.6	64.7	53.0	31.7
*-152	Baseline	44.9	66.9	37.7	7.7	47.8	69.1	41.4	10.4
	Baseline*	50.2	70.1	43.9	16.1	50.5	70.0	44.4	16.5
	DisAlign	53.7	62.8	51.9	34.2	54.5	64.5	52.0	34.7
	DisAlign*	54.8	63.9	53.9	32.5	55.0	65.1	53.3	32.2

Table 10: **Top-1 Accuracy on ImageNet-LT test set.** All models use the feature extractor and original classifier head trained with 90 epoch in joint learning stage, $*$ denotes the model uses cosine classifier head.

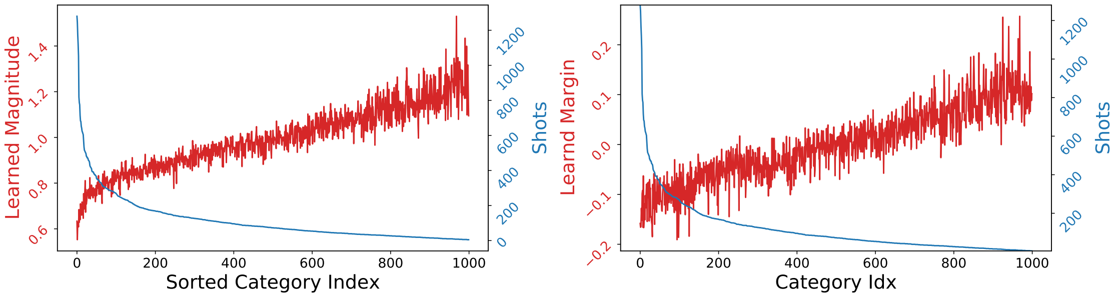

Figure 6: **Analysis of the Calibration.** We use model trained on ImageNet-LT with ResNeXt-50 for analysis.

Method	Calibration	G-RW	Top-1 Acc
Baseline*	-	-	49.2
cRT*	✗	✗	49.7
-	✗	✓	51.9
DisAlign*	✓	✓	53.4

Table 11: **Influence of Model Components.** Backbone is ResNeXt-50, $*$ means cosine classifier.

Generalized Re-weighting		✓	✓	✓	✓	✓	✓
Magnitude(w/o Confidence)		✓				✓	
Magnitude			✓				✓
Margin(w/o Confidence)				✓		✓	
Margin					✓		✓
Average Accuracy	41.6	49.9	50.1	49.6	49.9	51.0	51.3

Table 12: **Ablation of the Confidence Score.** We extend the Tab.5(main paper) to analyze the influence of confidence score.

categories into 3 groups according to the ratio of pixel number over the whole dataset. Specifically, three disjoint subsets are: *head classes*(classes each with a ratio over 1.0%), *body classes*(classes each with a ratio ranging from 0.1% to 1%) and *tail classes*(classes under a ratio of 0.1%), the complete list of the split is reported in Tab.16.

Evaluation. For the evaluation metric, we use the mean intersection of union(mIoU) and mean pixel accuracy(mAcc). We also report the mIoU and mAcc of each group(head, body and tail) for clarity.

B.2. Training Configuration

We implement our method based on MMSegmentation toolkit[31]. In the joint learning training phase, we set the learning rate to 0.01 initially, which gradually decreases to 0 by following the 'poly' strategy as [49]. The images are cropped to 512 × 512 and augmented with randomly scaling(from 0.5 to 2.0) and flipping. ResNet-50, ResNet-101 and ResNeSt-101[50] are used as the backbone. For the evaluation metric, we use the mean intersection of union(mIoU)

Backbone	Method	90 Epoch				200 Epoch			
		Average	Many	Medium	Few	Average	Many	Medium	Few
ResNet-50	Baseline	61.7	72.2	63.0	57.2	65.8	75.7	66.9	61.7
	Baseline*	64.8	75.8	66.6	59.7	66.2	77.3	68.3	60.7
	DisAlign	67.8	64.1	68.5	67.9	70.6	69.0	71.1	70.2
	DisAlign*	69.5	61.6	70.8	69.9	70.2	68.0	71.3	69.4
ResNet-101	Baseline	64.6	75.9	66.0	59.9	67.3	75.5	68.9	63.2
	Baseline*	66.4	76.8	68.5	61.1	68.0	78.9	69.7	63.0
	DisAlign	70.0	68.3	70.4	69.9	72.9	73.0	73.5	72.1
	DisAlign*	70.8	65.4	72.2	70.4	71.9	69.3	72.6	71.8
ResNet-152	Baseline	65.0	75.2	66.3	60.7	69.0	78.2	70.6	64.7
	Baseline*	67.3	77.8	69.4	61.8	69.0	78.5	71.0	64.0
	DisAlign	71.3	70.7	71.8	70.8	74.1	74.9	74.4	73.5
	DisAlign*	71.7	67.1	73.0	71.3	72.8	70.6	73.6	72.3

Table 13: Top-1 Accuracy on iNaturalist 2018 with different backbones(ResNet-$\{50,101,152\}$) and different training epochs(90 & 200), $*$ denotes the model uses cosine classifier head.

Backbone	Method	Top-1 Accuracy			
		Average	Many	Medium	Few
R-50	Baseline	29.2	45.3	25.5	8.0
	DisAlign	37.8	39.3	40.7	28.5
R-101	Baseline	30.2	46.1	26.9	8.4
	DisAlign	38.5	39.1	42.0	29.1
R-152	Baseline	30.2	45.7	27.3	8.2
	DisAlign	39.3	40.4	42.4	30.1

Table 14: Top-1 Accuracy on Places-LT with different backbones(ResNet-$\{50,101,152\}$).

and mean pixel accuracy(mAcc). All models are trained with 160k iterations with a batch size of 32 based on 8 V100 GPUs. In the DisAlign stage, we follow a similar protocol as stage-1 and only training the model with 8k iterations. We set $\rho = 0.3$ for all experiments.

B.3. Quantitative Results

We evaluate our method with two state-of-the-art segmentation models(FCN[36] and DeepLabV3+[7])based on different backbone networks, ranging from ResNet-50, ResNet-101 to the latest ResNeSt-101, and report the performance in Tab.15.

C. Experiments on LVIS Dataset

C.1. Dataset and Evaluation Protocol

Dataset. LVIS v0.5[15] dataset is a benchmark dataset for research on large vocabulary object detection and instance segmentation, which contains 56K images over 1230 categories for training, 5K images for validation. This challenging dataset is an appropriate benchmark to study the large-scale long-tail problem, where the categories can be binned into three types similar with ImageNet-LT: *rare*(1-10 training images), *common*(11-100 training images), and *frequent*($>$ 100 training images).

Evaluation Protocol. We evaluate our method on LVIS for object detection and instance segmentation. For evaluation, we use a COCO-style average precision(AP) metric that averages over categories and different box/mask intersection over union(IoU) threshold[27]. All standard LVIS evaluation metrics including AP, AP^r, AP^c, AP^f for box bounding boxes and segmentation masks. Subscripts 'r', 'c', and 'f' refer to rare, common and frequent category subsets.

C.2. Training Configuration

Experimental Details. We train our models for object detection and instance segmentation based on Detecron2[48], which is implemented in PyTorch. Unless specified, we use the ResNet backbone(pre-trained on ImageNet) with FPN[25]. Following the training procedure in [15], we resize the images so that the shorter side is 800 pixels. All baseline experiments are conducted on 8 GPUs with 2 images per GPU for 90K iterations, with a learning rate of 0.02 which is decreased by 10 at the 60K and 80K iteration. We use SGD with a weight decay of 0.0001 and momentum of 0.9. Scale jitter is applied for all experiments in default same with [15].

For the DisAlign, we freeze all network parameters and learn the magnitude and margin for extra 9K iterations with a learning rate of 0.02. Generalized re-weight is only used for fore-ground categories. Generalized re-weight scale ρ is set to 0.8 for all experiments.

Framework	B	Method	Aug	Mean IoU				Mean Accuracy			
				Average	Head	Body	Tail	Average	Head	Body	Tail
FCN[36]	R-50	Baseline	✗	36.1	62.5	38.1	27.6	45.4	76.9	48.8	34.5
		DisAlign	✗	37.5(+1.4)	62.6(+0.1)	40.2(+2.1)	28.8(+1.2)	49.9(+4.5)	76.7(-0.2)	54.9(+6.1)	39.0(+4.5)
		Baseline	✓	38.1	64.6	40.0	29.6	46.3	78.6	49.3	35.4
		DisAlign	✓	40.1(+2.0)	65.0(+0.4)	42.8(+2.8)	31.3(+1.7)	51.4(+5.1)	78.6(+0.0)	56.1(+6.8)	40.6(+5.2)
	R-101	Baseline	✗	39.9	65.3	42.0	31.7	49.6	79.1	52.6	39.6
		DisAlign	✗	41.8(+1.9)	65.5(+0.2)	44.1(+2.1)	33.7(+2.0)	54.7(+5.1)	79.0(-0.1)	58.6(+6.0)	45.2(+5.6)
		Baseline	✓	41.4	67.0	43.3	33.2	50.2	80.6	52.9	40.1
		DisAlign	✓	43.7(+2.3)	67.4(+0.4)	46.1(+2.8)	35.7(+2.5)	55.9(+5.7)	80.6(+0.0)	59.7(+6.8)	46.4(+6.3)
	S-101	Baseline	✗	45.6	66.6	47.5	38.6	57.8	78.8	62.1	48.9
		DisAlign	✗	46.2(+0.6)	66.6(+0.0)	48.0(+0.4)	39.4(+0.8)	60.3(+2.5)	79.1(+0.3)	64.9(+2.8)	51.7(+2.8)
		Baseline	✓	46.2	67.6	48.0	39.1	57.3	79.4	61.7	48.2
		DisAlign	✓	46.9(+0.7)	67.7(+0.1)	48.2(+0.2)	40.3(+1.2)	60.1(+2.8)	79.7(+0.3)	64.2(+2.5)	51.9(+3.7)
DeepLabV3$^+$[7]	R-50	Baseline	✗	43.9	66.6	47.1	35.6	54.9	79.4	60.3	44.5
		DisAlign	✗	44.4(+0.5)	66.6(+0.0)	47.2(+0.1)	36.5(+0.9)	57.2(+2.3)	79.8(+0.4)	62.3(+2.0)	47.5(+3.0)
		Baseline	✓	44.9	67.7	48.3	36.4	55.0	80.1	60.8	44.1
		DisAlign	✓	45.7(+0.8)	67.7(+0.0)	48.6(+0.3)	37.8(+1.4)	57.3(+2.3)	80.8(+0.7)	63.0(+2.2)	46.9(+2.8)
	R-101	Baseline	✗	45.5	67.6	48.2	37.6	56.4	80.1	61.2	46.6
		DisAlign	✗	46.0(+0.5)	67.6(+0.0)	48.4(+0.2)	38.5(+0.9)	59.1(+2.7)	80.5(+0.4)	63.8(+2.6)	49.9(+3.3)
		Baseline	✓	46.4	68.7	49.0	38.4	56.7	80.9	61.5	46.7
		DisAlign	✓	47.1(+0.7)	68.7(+0.0)	49.4(+0.4)	39.6(+1.2)	59.5(+2.8)	81.4(+0.5)	64.2(+2.7)	50.3(+3.6)
	S-101	Baseline	✗	46.5	68.0	49.1	38.8	58.1	80.1	63.4	48.5
		DisAlign	✗	46.9(+0.4)	67.8(-0.2)	49.2(+0.1)	39.6(+0.8)	60.7(+2.6)	80.5(+0.4)	65.5(+2.1)	51.9(+3.4)
		Baseline	✓	47.3	69.0	49.7	39.7	58.1	80.8	63.4	48.2
		DisAlign	✓	47.8(+0.5)	68.9(-0.1)	49.8(+0.1)	40.7(+1.0)	60.1(+2.0)	81.0(+0.2)	65.5(+2.1)	52.0(+3.8)

Table 15: **Results on ADE-20K:** All baseline models are trained with a image size of 512x512 and 160K iteration in total. **Aug** denotes multi-scale is used for inference.

Category	Ratio	Group	Category	Ratio	Group	Category	Ratio	Group	Category	Ratio	Group	Category	Ratio	Group
'wall'	0.1576,	Head	'armchair'	0.0044,	Body	'river'	0.0015,	Body	'airplane'	0.0007,	Tail	'food'	0.0005,	Tail
'building'	0.1072,	Head	'seat'	0.0044,	Body	'bridge'	0.0015,	Body	'dirt track'	0.0007,	Tail	'step'	0.0004,	Tail
'sky'	0.0878,	Head	'fence'	0.0033,	Body	'bookcase'	0.0014,	Body	'apparel'	0.0007,	Tail	'tank'	0.0004,	Tail
'floor'	0.0621,	Head	'desk'	0.0031,	Body	'blind'	0.0014,	Body	'pole'	0.0006,	Tail	'trade name'	0.0004,	Tail
'tree'	0.048,	Head	'rock'	0.003,	Body	'coffee table'	0.0014,	Body	'land'	0.0006,	Tail	'microwave'	0.0004,	Tail
'ceiling'	0.045,	Head	'wardrobe'	0.0027,	Body	'toilet'	0.0014,	Body	'bannister'	0.0006,	Tail	'pot'	0.0004,	Tail
'road'	0.0398,	Head	'lamp'	0.0026,	Body	'flower'	0.0014,	Body	'escalator'	0.0006,	Tail	'animal'	0.0004,	Tail
'bed'	0.0231,	Head	'bathtub'	0.0024,	Body	'book'	0.0013,	Body	'ottoman'	0.0006,	Tail	'bicycle'	0.0004,	Tail
'windowpane'	0.0198,	Head	'railing'	0.0024,	Body	'hill'	0.0013,	Body	'bottle'	0.0006,	Tail	'lake'	0.0004,	Tail
'grass'	0.0183,	Head	'cushion'	0.0023,	Body	'bench'	0.0013,	Body	'buffet'	0.0006,	Tail	'dishwasher'	0.0004,	Tail
'cabinet'	0.0181,	Head	'base'	0.0023,	Body	'countertop'	0.0012,	Body	'poster'	0.0006,	Tail	'screen'	0.0004,	Tail
'sidewalk'	0.0166,	Head	'box'	0.0022,	Body	'stove'	0.0012,	Body	'stage'	0.0006,	Tail	'blanket'	0.0004,	Tail
'person'	0.016,	Head	'column'	0.0022,	Body	'palm'	0.0012,	Body	'van'	0.0006,	Tail	'sculpture'	0.0004,	Tail
'earth'	0.0151,	Head	'signboard'	0.002,	Body	'kitchen island'	0.0012,	Body	'ship'	0.0006,	Tail	'hood'	0.0004,	Tail
'door'	0.0118,	Head	'chest of drawers'	0.0019,	Body	'computer'	0.0011,	Body	'fountain'	0.0005,	Tail	'sconce'	0.0003,	Tail
'table'	0.011,	Head	'counter'	0.0019,	Body	'swivel chair'	0.001,	Tail	'conveyer belt'	0.0005,	Tail	'vase'	0.0003,	Tail
'mountain'	0.0109,	Head	'sand'	0.0018,	Body	'boat'	0.0009,	Tail	'canopy'	0.0005,	Tail	'traffic light'	0.0003,	Tail
'plant'	0.0104,	Head	'sink'	0.0018,	Body	'bar'	0.0009,	Tail	'washer'	0.0005,	Tail	'tray'	0.0003,	Tail
'curtain'	0.0104,	Head	'skyscraper'	0.0018,	Body	'arcade machine'	0.0009,	Tail	'plaything'	0.0005,	Tail	'ashcan'	0.0003,	Tail
'chair'	0.0103,	Head	'fireplace'	0.0018,	Body	'hovel'	0.0009,	Tail	'swimming pool'	0.0005,	Tail	'fan'	0.0003,	Tail
'car'	0.0098,	Body	'refrigerator'	0.0018,	Body	'bus'	0.0009,	Tail	'stool'	0.0005,	Tail	'pier'	0.0003,	Tail
'water'	0.0074,	Body	'grandstand'	0.0018,	Body	'towel'	0.0008,	Tail	'barrel'	0.0005,	Tail	'crt screen'	0.0003,	Tail
'painting'	0.0067,	Body	'path'	0.0018,	Body	'light'	0.0008,	Tail	'basket'	0.0005,	Tail	'plate'	0.0003,	Tail
'sofa'	0.0065,	Body	'stairs'	0.0017,	Body	'truck'	0.0008,	Tail	'waterfall'	0.0005,	Tail	'monitor'	0.0003,	Tail
'shelf'	0.0061,	Body	'runway'	0.0017,	Body	'tower'	0.0008,	Tail	'tent'	0.0005,	Tail	'bulletin board'	0.0003,	Tail
'house'	0.006,	Body	'case'	0.0017,	Body	'chandelier'	0.0008,	Tail	'bag'	0.0005,	Tail	'shower'	0.0003,	Tail
'sea'	0.0053,	Body	'pool table'	0.0017,	Body	'awning'	0.0007,	Tail	'minibike'	0.0005,	Tail	'radiator'	0.0003,	Tail
'mirror'	0.0052,	Body	'pillow'	0.0017,	Body	'streetlight'	0.0007,	Tail	'cradle'	0.0005,	Tail	'glass'	0.0002,	Tail
'rug'	0.0046,	Body	'screen door'	0.0015,	Body	'booth'	0.0007,	Tail	'oven'	0.0005,	Tail	'clock'	0.0002,	Tail
'field'	0.004	Body	'stairway'	0.0015	Body	'television receiver'	0.0007	Tail	'ball'	0.0005	Tail	'flag'	0.0002	Tail

Table 16: **Splits of ADE-20K:** The ratio of each category is reported according to [54].

Backbone	**Method**	**BBox AP**				**Mask AP**			
		$\mathbf{AP}_{bbox}$	$\mathbf{AP}^{r}_{bbox}$	$\mathbf{AP}^{c}_{bbox}$	$\mathbf{AP}^{f}_{bbox}$	$\mathbf{AP}_{mask}$	$\mathbf{AP}^{r}_{mask}$	$\mathbf{AP}^{c}_{mask}$	$\mathbf{AP}^{f}_{mask}$
ResNet-50	Baseline	20.8	3.3	19.5	29.4	21.2	3.7	21.6	28.4
	DisAlgin	23.9	7.5	25.0	29.1	24.2	8.5	26.2	28.0
	Baseline*	22.8	10.3	21.1	30.1	23.8	11.5	23.7	28.9
	DisAlgin*	25.6	13.7	25.6	30.5	26.3	14.9	27.6	29.2
ResNet-101	Baseline	22.2	2.6	21.1	31.6	22.6	2.7	22.8	30.2
	DisAlgin	25.6	9.0	26.5	30.9	25.8	10.3	27.6	29.6
	Baseline*	24.5	10.1	23.2	31.8	25.1	11.2	25.2	30.4
	DisAlgin*	27.5	15.9	27.6	32.0	28.2	17.8	29.7	30.5
ResNeXt-101	Baseline	24.5	3.9	24.1	33.1	25.0	4.2	26.3	31.8
	DisAlgin	26.8	8.8	27.6	33.0	27.4	11.0	29.3	31.6
	Baseline*	26.9	12.1	26.1	33.8	27.7	15.2	28.2	32.2
	DisAlgin*	29.5	17.7	29.5	33.8	30.0	19.6	31.5	32.3

Table 17: **Results on LVIS v0.5 dataset with Mask R-CNN.** * denotes the model use cosine classifier head.

Backbone	**Method**	**BBox AP**				**Mask AP**			
		$\mathbf{AP}_{bbox}$	$\mathbf{AP}^{r}_{bbox}$	$\mathbf{AP}^{c}_{bbox}$	$\mathbf{AP}^{f}_{bbox}$	$\mathbf{AP}_{mask}$	$\mathbf{AP}^{r}_{mask}$	$\mathbf{AP}^{c}_{mask}$	$\mathbf{AP}^{f}_{mask}$
ResNet-50	Baseline	25.2	3.7	24.3	34.8	23.0	3.5	23.0	30.8
	DisAlgin	28.7	9.0	30.2	34.6	26.1	8.4	28.1	30.7
	Baseline*	28.8	15.4	28.2	34.9	26.2	13.6	26.3	31.1
	DisAlgin*	32.2	21.6	33.3	35.2	29.4	19.4	30.9	31.4
ResNet-101	Baseline	26.1	3.4	25.4	35.9	24.0	3.3	24.2	32.0
	DisAlgin	29.7	8.1	31.7	35.8	27.3	7.8	29.7	32.0
	Baseline*	30.4	15.5	30.3	36.5	28.1	13.9	29.2	32.4
	DisAlgin*	33.7	22.1	34.9	36.9	30.9	19.0	33.2	32.8
ResNeXt-101	Baseline	28.4	4.6	28.6	37.5	26.1	4.6	27.2	33.4
	DisAlgin	31.3	9.5	33.2	37.7	28.7	9.0	31.1	33.6
	Baseline*	32.6	18.5	32.8	37.9	29.8	16.9	30.9	33.7
	DisAlgin*	34.7	24.6	35.3	38.1	31.8	22.0	33.2	33.9

Table 18: **Results on LVIS v0.5 dataset with Cascade R-CNN.** * denotes the model use cosine classifier head.

C.3. Quantitative Results

We report the detailed results in Table.17 and Tab.18.