# Supplementary Material for "Invariant Feature Learning for Generalized Long-Tailed Classification"

Kaihua Tang[1], Mingyuan Tao[2], Jiaxin Qi[1], Zhenguang Liu[3], Hanwang Zhang[1]

[1] Nanyang Technological University, Singapore
[2] Damo Academy, Alibaba Group, Hangzhou, China
[3] Zhejiang University, Hangzhou, China
kaihua.tang@ntu.edu.sg; juchen.tmy@alibaba-inc.com; jiaxin003@e.ntu.edu.sg
liuzhenguang2008@gmail.com; hanwangzhang@ntu.edu.sg

**Abstract.** This supplementary material includes: 1) implementation details of the proposed IFL and the baseline methods; 2) a detailed analysis of our problem formulation; 3) more evidences of the re-sampling strategy in the proposed environment construction; 4) more details of the dataset construction for the proposed ImageNet-GLT and MSCOCO-GLT benchmarks; 5) more experimental results of the proposed Invariant Feature Learning framework.

## A Implementation Details

For the proposed IFL, we first trained the model using the cross-entropy loss for 60 epochs, then we started to construct and update environments every 20 epochs. This is because the early epochs are learning generalized features [15,24], so constructing environments in early epochs won't benefit the robustness of the model too much. The trade-off parameter $\alpha$ between the classification loss and IFL metric loss is initialized as 0.0 and changed to 0.001 and 0.005 along with the environment construction. Besides, following the center loss [21], the mean feature is accumulated by an SGD optimizer with the fixed learning rate of 0.5. The pseudo code of the overall IFL algorithm is given in Algorithm 1.

For fair comparisons, we re-implemented all the investigated algorithms into the same GLT codebase that is publicly available on Github: https://github.com/KaihuaTang/Generalized-Long-Tailed-Benchmarks.pytorch. By default, all images are resized to $112 \times 112$ using Random Resized Crop and Random Horizontal Flip during training. We used ResNeXt-50 [23] as our backbone for all methods except for BBN [27], RIDE [20] and TADE [26]. All models were trained by the SGD optimizer with the batch size 256. The initial learning rate is 0.1 and the default learning rate decay strategy is Cosine Annealing scheduler [10] except for BBN [27], LDAM [2] and RIDE [20] that adopted the multi-step scheduler based on original settings in the corresponding papers. For all one-stage methods, results were reported as the performance of the model at epoch 100. For

two-stage cRT [7] and LWS [7], additional 10 epochs were required to fine-tune a balanced classifier.

**Algorithm 1** The proposed IFL algorithm

**Input:** the original training set $\{(x, y)\}$
**Initialize:** backbone $f(\cdot;\theta)$, classifier $g(\cdot;w)$
**for** N warm-up epochs **do**
  // optimizing the model from cross-entropy classification loss
  $\theta, w \in \arg\min_{\theta,w} L_{cls}(f(x;\theta), y; w)$
**end for**
**Initialize:** class centers $\{C_y\}$ for all classes
**repeat**
  $\{(x^{e_1}, y^{e_1})\}$, $\{(x^{e_2}, y^{e_2})\}$ = EnvConstruct($\{(x, y)\}$, $\theta$, $w$)
  **for** M epochs **do**
    // learning from $L_{cls}$ and $L_{IFL}$
    $\theta, w \in \arg\min_{\theta,w} \sum_{e\in\mathcal{E}} \sum_{i\in e}(L_{cls} + \alpha \cdot L_{IFL})$
    // update class center
    $\{C_y\} \leftarrow$ MovingAverage($\{C_y\}, \{(f(x^{e_1};\theta), y^{e_1})\}, \{(f(x^{e_2};\theta), y^{e_2})\}$)
  **end for**
**until** Converge
**Output:** backbone $f(\cdot;\theta)$, classifier $g(\cdot;w)$

## B Problem Formulation

In Section 3 of the original paper, we formulate the classification model $p(Y|X)$ as $p(Y|z_c, z_a)$ based on a common assumption [12] that any object image $X$ equals to a set of underlying class-specific components $z_c$ and varying attributes $z_a$, *i.e*, $X = (z_c, z_a)$, where $(z_c, z_a)$ can fully describe the entire the image $X$. Regarding the classification task, a robust feature should only be extracted from the underlying $z_c$, leaving non-robust $z_a$ out of the visual feature. That is to say, an ideal feature backbone $f(\cdot)$ should only respond to class-specific $z_c$: $\mathbf{z} = f(X) = h(z_c)$, where there exists a mapping function $h(\cdot)$ between the extracted feature $\mathbf{z}$ and the underlying robust class-specific components $z_c$. So, we can further convert the $p(Y|X)$ into the following formula using the Bayes theorem [18]:

$$\begin{aligned} p(Y=k|X=x) &= p(Y=k|z_c, z_a) \\ &= \frac{p(z_c, z_a|Y=k)}{p(z_c, z_a)} \cdot p(Y=k) \\ &= \frac{p(z_c|Y=k)}{p(z_c)} \cdot \underbrace{\frac{p(z_a|Y=k, z_c)}{p(z_a|z_c)}}_{attribute\ bias} \cdot \underbrace{p(Y=k)}_{class\ bias}, \end{aligned} \tag{1}$$

where $\frac{p(z_c|Y=k)}{p(z_c)}$ is a robust indicator of the class; $\frac{p(z_a|Y=k,z_c)}{p(z_a|z_c)}$ is a bias term introduced by the existence of imbalanced attributes; $p(Y=k)$ is a class bias reflecting the class distribution of the training data.

To better understand this formulation, we will provide a more detailed analysis, together with some interesting findings that can be derived from Eq. (1), in the following sub-sections.

### B.1 Details about Class-specific $z_c$

Since the underlying class-specific $z_c$ equals to the existence of multiple independent components, the formal definition of $\frac{p(z_c|Y=k)}{p(z_c)}$ in Eq. (1) is as follows:

$$\frac{p(z_c|Y=k)}{p(z_c)} = \prod_{c_i \in exist} \frac{p(z_{c_i}=1|Y=k)}{p(z_{c_i}=1)} \cdot \prod_{c_j \in non-exist} \frac{p(z_{c_j}=0|Y=k)}{p(z_{c_j}=0)}, \quad (2)$$

where the underlying class-specific components $z_c$ can be regarded as a 0/1 vector; $\{c_i\}$ are the existed components on the object; $\{c_j\}$ are the non-existed components on the object. An object $X$ belongs to class $Y=k$, if and only if **I)** all the essential components of class $Y=k$ exist, *e.g*, a human has to contain the existence of both $p(z_{head\ on\ object}=1|Y=human)=1$ and $p(z_{body\ on\ object}=1|Y=human)=1$, **II)** and the other irrelevant components don't exist, *e.g*, $p(z_{tail\ on\ object}=0|Y=human)=1$ and $p(z_{horn\ on\ object}=0|Y=human)=1$.

Note that the above formulation may raise two questions regarding the class-specific $z_c$: 1) the hierarchy of classification and 2) the partial occlusion.

**The Hierarchy of Classification:** One common question about the above Eq. (2) would be what if an object with all the essential components of a class also contains other irrelevant components, *e.g*, an object with both human head and human body also has 4 horse limbs[4]. This problem is raised from the hierarchy of classification. If an object from class $Y=k$ contains one or more unnecessary components, it usually means the object is actually from a sub-species of class $Y=k$, which is belong to $Y=k$ but with more fine-grained descriptions. In a valid multi-class classification dataset, a species and its sub-species won't simultaneously exist in the prediction vocabulary. Otherwise, it becomes a multi-label classification task. However, multiple sub-species from a common hidden hyper-species can be co-existed in one dataset, *e.g*, "shetland sheepdog" and "pug-dog" in ImageNet [17] are both from a hyper-species "dog". In this case, although these sub-species may have close templates, the class-specific components $z_c$ still satisfy the property of intra-class invariance. In fact, visualizations from the previous research [19] found that the cross-entropy loss will automatically learn discriminative components instead of some common components in this case, *e.g*, only the teeth of "warthog" will be learned as class-specific features, rather than the entire body, to increase the power of discrimination within the dataset.

**The Partial Occlusion:** Another question is about what if there are partial occlusions on the images. Does this break the intra-class invariance of class-specific components $z_c$? The answer is still NO. The recent study of Masked

[4] Centaur: a creature from Greek mythology that has both human upper body and horse lower body [22], which can be regarded as a sub-species of human.

Auto-Encoders [4] proves that deep-learning models can reconstruct the entire image from mere 25% patches due to the redundant visual information. However, what if the strong occlusion hurts the class-specific component $z_c$ by removing the entire component? During training, these images shouldn't exist in the training data in the first place, as they can not be labeled. During prediction, the corresponding image is no longer predictable and supposed to be detected as an outlier [1], which is beyond the scope of this paper. Therefore, if there are just some slight partial occlusions, models would still be capable of obtaining the original $z_c$, as we can naturally imagine the entire object from the redundant visual information.

### B.2 Details about the attributes $z_a$

When we look at Eq. (1), we may wonder if there exists certain attribute $z_{a_j}$ that is **class-independent**, *i.e*, $p(z_{a_j}|Y=k,z_c) = p(z_{a_j}|z_c)$. The answer is YES. So we can have a formal definition of $\frac{p(z_a|Y=k,z_c)}{p(z_a|z_c)}$ in Eq. (1) as follows:

$$\frac{p(z_a|Y=k,z_c)}{p(z_a|z_c)} = \prod_{a_i \in d-Y} \underbrace{\frac{p(z_{a_i}|Y=k,z_c)}{p(z_{a_i}|z_c)}}_{biased\ attributes} \cdot \prod_{a_j \in ind-Y} \underbrace{\frac{p(z_{a_j}|Y=k,z_c)}{p(z_{a_j}|z_c)}}_{benign\ attributes}, \tag{3}$$

where $d-Y$ and $ind-Y$ stand for class-dependent and class-independent, respectively; $\{a_i\}$ are class-dependent attributes; $\{a_j\}$ are class-independent attributes. It's easy to notice that all benign attributes have $p(z_{a_j}|Y=k,z_c) = p(z_{a_j}|z_c)$, so the $\prod_{a_j \in ind-Y} \frac{p(z_{a_j}|Y=k,z_c)}{p(z_{a_j}|z_c)}$ always equals to 1 and won't introduce any biases. In the original paper, we consider all attributes as a whole, *i.e*, a vector $z_a$, so we didn't differentiate the biased attributes and benign attributes. Their differences are explained in the following.

**Biased Attributes:** Nearly all semantic attributes are biased. For example, considering a class-specific component $z_{c_{fur}}$ as "fur" and $z_{a_{color}}$ is an attribute describing its "color", different animals (Class Y) may have different distributions for the color of fur. In summary, the biased attributes $\{a_i \in d-Y\}$ are either dependent on a class-specific component, *e.g*, the "color" or "texture" for a specific $z_{c_i}$, or dependent on multiple components at the same time, *e.g*, the "posture" of a human that depends on both head, body and four limbs. Those biased attributes not only cause the long-tailed prediction confidence within each class, but also create spurious correlations between a biased attribute $z_{a_i}$ and a specific class $Y=k$.

**Benign Attributes:** Since we consider all attributes as a whole in the original paper, *i.e*, an underlying vector $z_a$, we don't differentiate benign attributes from the entire $z_a$. However, if we look as Eq. (3), an attribute $a_j \in ind-Y$ is totally benign as its distribution is independent of $Y=k$, *i.e*, $p(z_{a_j}|Y=k,z_c) = p(z_{a_j}|z_c)$, making $\frac{p(z_{a_j}|Y=k,z_c)}{p(z_{a_j}|z_c)} = 1$ become an unbiased term. In fact, **it fundamentally explains the effectiveness of the intuitive Data Augmentation method**, as the data augmentation introduces additional images

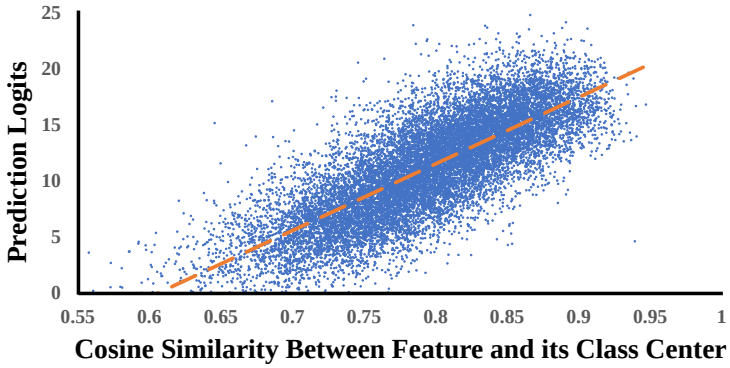



**Fig. 1.** The relationship between the prediction logits and the cosine similarity between image features and its class center in ImageNet-GLT. The direct proportion between them demonstrates why the prediction confidence, *i.e*, $p(Y = k|X)$, can be used to probe the intra-class variation, *i.e*, $\frac{p(z_a|Y=k,z_c)}{p(z_a|z_c)}$.

(at no cost) with benign attributes that are independent of their classes. There are two types of benign attributes regarding different types of data augmentation: **I)** the image-level attributes, that are independent of both $z_c$ and $Y$, are commonly used as a pre-processing augmentation in the data loader, as they can be easily generated from any given image, *e.g*, the rotation (Rotation), position (Crop), size (Resize), and color (ColorJitter) for the image; **II)** the component-related attributes, that are only independent of $Y$ but still depend on $z_c$, are data augmentation at the data collection stage, *e.g*, the viewpoint, as they are variations caused by the relative position between the camera and a specific component instead of the object class. These benign attributes can increase both the volume and the diversity of the classification dataset without any cost, which explains why the data augmentation method like RandAug [3] are so effective in our experiments.

## C Environment Construction

### C.1 Motivation

Since we don't impose the disentanglement assumption that perfect feature vectors $\mathbf{z} = [z_c; z_a]$ with separated $z_c$ and $z_a$ can be learned, we cannot easily eliminate $z_a$ through feature selection. Therefore, we can only implicitly prevent features from associating with $z_a$ during the backbone optimization. To achieve this, we use the prediction logits of images to sample diverse distributions of $\frac{p(z_a|Y=k,z_c)}{p(z_a|z_c)}$ based on the formulation Eq. (1), as $p(Y = k|X) \propto \frac{p(z_a|Y=k,z_c)}{p(z_a|z_c)}$ within a given class $Y = k$. If the feature backbone relies on the non-robust $z_a$,

the diverse distributions of $\frac{p(z_a|Y=k,z_c)}{p(z_a|z_c)}$ will make the model have unstable class centers across different environments. On the contrary, if the feature backbone relies more on the robust and invariant components $z_c$, the class centers should be more consistent against the environment change, which motivates the design of the proposed IFL.

### C.2 Re-sampling Strategy

The overall environment construction pipeline can be explained as follows: since $\frac{p(z_a|Y=k,z_c)}{p(z_a|z_c)}$ is only directly proportional to $p(Y=k|X)$ within a given class $Y=k$, the re-sampling strategy is conducted within each class independently. Then, for each environment, we merge the resultant subsets from each class that use the same re-sampling strategy, so the diverse environments are thus corresponding to different re-sampling strategy, *i.e*, different $\frac{p(z_a|Y=k,z_c)}{p(z_a|z_c)}$.

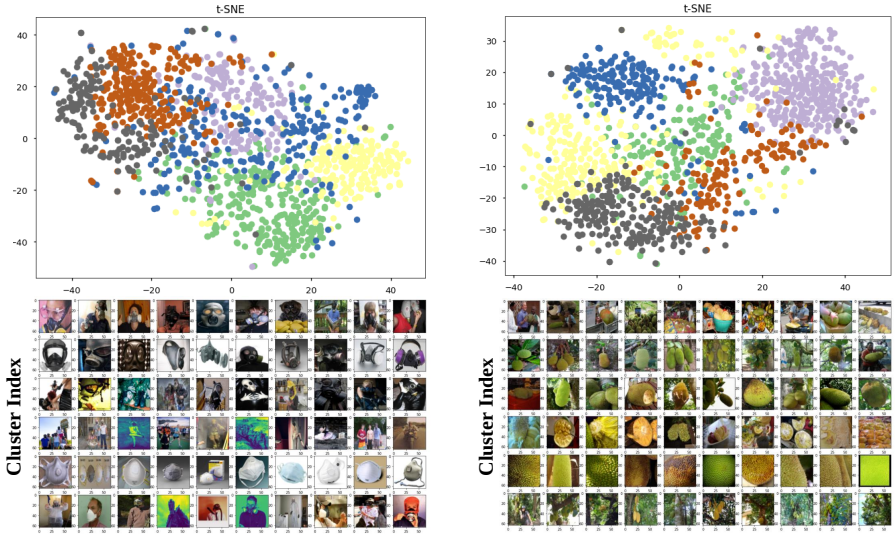
In this sub-section, we will provide additional evidences to support the re-sampling strategy in our environment construction method. To prove the direct correlation between $p(Y=k|X)$ and $\frac{p(z_a|Y=k,z_c)}{p(z_a|z_c)}$, we visualize the relationship between the prediction logits and the cosine similarity between image features and its class center in Fig. 1. First of all, $p(z_a|z_c)$ can be regarded as a fixed layout in the given dataset and $\frac{p(z_c|Y=k)}{p(z_c)}$ is invariant within a class, therefore, the intra-class variation is mainly caused by $p(z_a|Y=k,z_c)$ in a given class and dataset. Since the attribute $z_a$ is long-tailed, the prediction logits should also exhibit long-tailed distribution, *i.e*, most of the samples with few head attributes have relatively larger logits while few samples with most of the rare attributes have lower logits. The visualized Fig. 1 proved this assumption. As the class center is dominated by the head attributes due to their large population, the cosine similarity between the feature of an image and its class center actually indicates the rarity of the attribute vector $z_a$ for this image. The rarer the attributes are, the lower the prediction confidence is.

In summary, Fig. 1 demonstrates the reason why we can use $p(Y=k|X)$ to probe the $\frac{p(z_a|Y=k,z_c)}{p(z_a|z_c)}$ and thus construct diverse environments and how bias the intra-class distribution is (caused by long-tailed attributes).

## D Dataset Construction

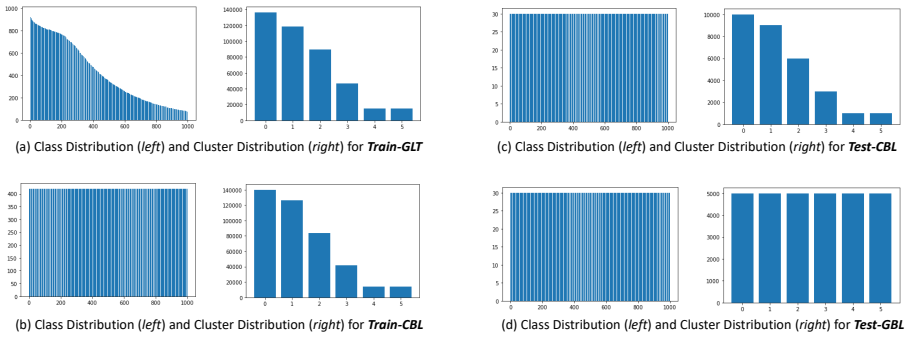
### D.1 ImageNet-GLT

The proposed ImageNet-GLT benchmark is a long-tailed subset generated from the original ImageNet [17] dataset. For Train-GLT, Train-CBL, and Test-CBL splits, we can directly follow the previous ImageNet-LT benchmark [9] to construct these splits. The tricky part is the Test-GBL that is used to evaluate the Attribute-wise Long Tail (ALT) protocol and Generalized Long Tail (GLT) protocol. To create the attribute-wise balanced and class-wise balanced evaluation environment (Test-GBL), we used the index of several feature clusters of each category as the pretext attribute labels.




(a) An Example of Clustered Images (b) An Example of Clustered Images

**Fig. 2.** Examples of feature clusters using KMeans (through t-SNE [6]) and the corresponding visualized images for each cluster in the original ImageNet [17].



(a) Class Distribution (*left*) and Cluster Distribution (*right*) for ***Train-GLT***

(c) Class Distribution (*left*) and Cluster Distribution (*right*) for ***Test-CBL***

(b) Class Distribution (*left*) and Cluster Distribution (*right*) for ***Train-CBL***

(d) Class Distribution (*left*) and Cluster Distribution (*right*) for ***Test-GBL***

**Fig. 3.** Class distributions and cluster distributions for each split of ImageNet-GLT benchmark. Note that clusters may represent different attribute layouts in each classes, so ImageNet-GLT actually has $6 \times 1000$ pretext attributes rather than 6, *i.e*, each column in the cluster distribution stands for 1000 pretext attributes having the same frequency.

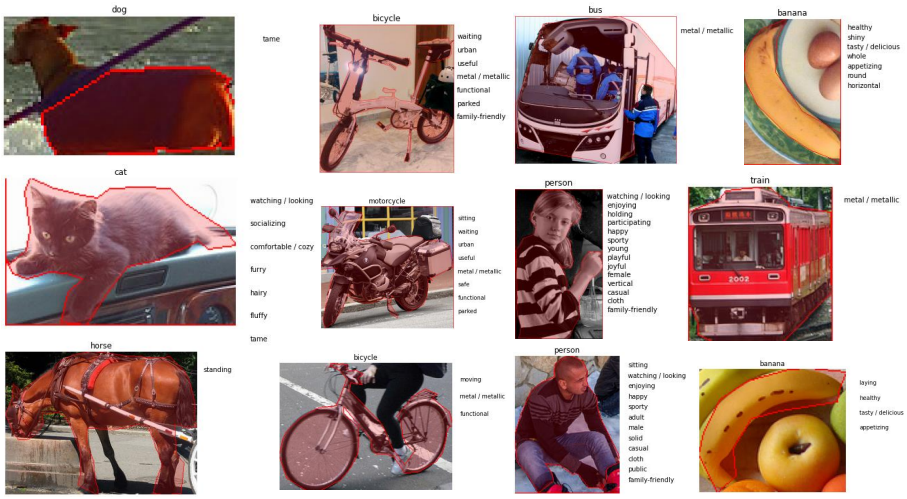



**Fig. 4.** Examples of objects from MSCOCO-Attribute [14] dataset. Note that the object attribute won't be released in the proposed MSCOCO-GLT

To begin with, we applied a pre-trained ResNet-50 [5] backbone provided by PyTorch [13] to extract a 2048-dimensional feature vector for all images in the ImageNet dataset. Then, we ran the KMeans algorithm to generate 6 clusters for each category. As we can see from Figure 2, the corresponding visualized images demonstrate that the feature clusters within each category naturally separate images by different types of attributes, *e.g*, materials, backgrounds, *etc*.

Afterwards, we smoothed the attribute distribution of each category by ensuring the frequencies of Top-2, Medium-2, Bottom-2 clusters to be 70%, 20%, and 10% for all categories. Note that the underlying attribute distribution is naturally long-tailed. However, since we only used a limited number of clusters, *i.e*, 6 clusters, to approximate the numerous realistic attributes, those clusters could be relatively too balanced for some categories, so we force the distribution of pretext attributes to be the same for all categories.

We also list all the class distributions and cluster distributions for all four split, Train-GLT, Train-CBL, Test-CBL, Test-GBL, of ImageNet-GLT benchmark in Fig. 3. Note that each column in the cluster distribution stands for 1000 pretext attributes for all 1000 classes that have the same frequency, as clusters may represent different attribute layouts in each classes.

### D.2 MSCOCO-GLT

The proposed MSCOCO-GLT benchmark is a long-tailed subset generated from the MSCOCO-Attribute [14,8], where we cropped each object as individual images. Fig. 4 shows several examples of objects from MSCOCO-Attribute [14]

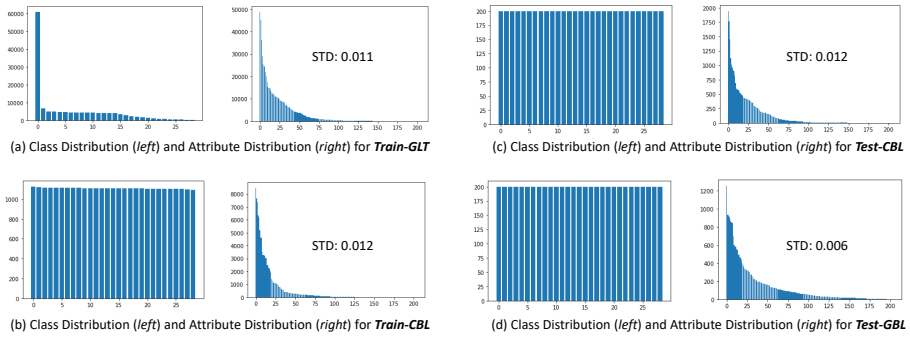



**Fig. 5.** Class distributions and attribute distributions for each split of MSCOCO-GLT benchmark, where the most frequent category is the "person". Although we cannot strictly balance the attribute distribution in Test-GBL split due to the fact that both head attributes and tail attributes can co-occur in one object, the selected Test-GBL has lower standard deviation of attributes than other splits.

**Table 1. Evaluation of GLT Protocol on ImageNet-GLT**: This is a supplementary table of the Table 1 in original paper. $\text{Many}_A$, $\text{Medium}_A$, $\text{Few}_A$ indicate the head (Top-2 Clusters), medium (Medium-2 Clusters), and tail (Bottom-2 Clusters) attributes. The (Top-1) Accuracy and Precision of previous LT algorithms and their IFL variants are reported

| | Setting | Generalized Long Tail (GLT) Protocol | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | Test Splits | $\text{Many}_A$ | | $\text{Medium}_A$ | | $\text{Few}_A$ | | Overall | |
| | Evaluation Metric | Accuracy | Precision | Accuracy | Precision | Accuracy | Precision | Accuracy | Precision |
| Re-balance | Baseline | 47.36 | 51.84 | 33.45 | 36.05 | 23.44 | 25.66 | 34.75 | 40.65 |
| | cRT [7] | 51.21 | 50.48 | 36.25 | 35.03 | 25.24 | 24.62 | 37.57 | 37.51 |
| | LWS [7] | 51.95 | 51.14 | 36.77 | 35.60 | 25.09 | 24.70 | 37.94 | 38.01 |
| | Deconfound-TDE [19] | 51.03 | 49.52 | 36.11 | 34.25 | 25.53 | 24.47 | 37.56 | 37.00 |
| | BLSoftmax [16] | 50.58 | 50.21 | 35.74 | 35.17 | 24.96 | 25.46 | 37.09 | 38.08 |
| | Logit-Adj [11] | 52.25 | 51.08 | 36.12 | 34.92 | 25.04 | 24.71 | 37.80 | 37.56 |
| | BBN [27] | 51.92 | 53.17 | 36.24 | 36.26 | 25.56 | 26.07 | 37.91 | 41.77 |
| | LDAM [2] | 52.33 | 51.91 | 37.09 | 36.07 | 26.19 | 25.56 | 38.54 | 39.08 |
| | (ours) Baseline + IFL | 51.16 | **56.64** | 36.50 | 39.88 | 26.21 | 29.00 | 37.96 | **44.47** |
| | (ours) cRT + IFL | 53.53 | 54.93 | 38.15 | 38.21 | 27.13 | 27.66 | 39.60 | 41.65 |
| | (ours) LWS + IFL | 53.31 | 55.42 | 38.21 | 39.16 | 27.40 | 28.46 | 39.64 | 42.45 |
| | (ours) BLSoftmax + IFL | 53.68 | 55.35 | 39.05 | **40.67** | 27.49 | 29.09 | 40.08 | 43.48 |
| | (ours) Logit-Adj + IFL | **54.90** | 55.19 | **39.09** | 39.93 | **27.57** | **29.27** | **40.52** | 42.28 |
| Augment | Mixup [25] | 43.46 | 47.05 | 30.23 | 30.63 | 20.96 | 22.73 | 31.55 | 37.44 |
| | RandAug [3] | 51.63 | 55.21 | 36.96 | 39.95 | 26.14 | 27.84 | 38.24 | 44.74 |
| | (ours) Mixup + IFL | 57.10 | 61.66 | 41.46 | 45.04 | 30.43 | 33.42 | 43.00 | 49.25 |
| | (ours) RandAug + IFL | **58.67** | **62.42** | **44.06** | **47.36** | **31.97** | **34.67** | **44.90** | **50.47** |
| Ensemble | TADE [26] | 55.52 | 56.35 | 40.86 | 41.23 | 28.85 | 30.01 | 41.75 | 44.15 |
| | RIDE [20] | 57.49 | 56.24 | 41.57 | 39.96 | 29.94 | 29.18 | 43.00 | 43.32 |
| | (ours) TADE + IFL | 56.99 | 57.44 | 42.55 | 42.39 | 30.86 | 31.55 | 43.47 | 45.17 |
| | (ours) RIDE + IFL | **59.24** | **59.87** | **44.98** | **44.80** | **32.69** | **32.55** | **45.64** | **47.14** |

dataset. Note that the object attribute won't be released in the proposed MSCOCO-GLT.

Different from the ImageNet-GLT, we adopt real attribute annotations to construct the MSCOCO-GLT. However, as we can see from Fig. 4, a single object can be labelled with multiple attributes, causing the tail attribute co-occurring with the head in one image. Therefore, it's impossible to strictly balance the attribute distribution in the Test-GBL, so we proposed the Algorithm 2 to generate the Test-GBL for MSCOCO-GLT. Specifically, we need to iteratively look for samples to minimize the standard deviation of the overall attribute distribution within each category, so the output subset is the most balanced subset in terms of the attribute distribution in the entire dataset.

As to the Train-GLT, Train-CBL, and Test-CBL, we can use the same strategy as the previous sub-section to directly sample from each category. Note that the long-tailed distribution in the MSCOCO-GLT is more severe than ImageNet-GLT, as it only has 29 classes but the single class "person" possess over 40% of the training data. Therefore, the class-wise balanced Train-CBL has much smaller size of data.

We also list all the class distributions and attribute distributions for all four split, Train-GLT, Train-CBL, Test-CBL, Test-GBL, of MSCOCO-GLT benchmark in Fig. 5. Here we only show the Test-GBL for CLT and GLT protocols. Test-GBL for ALT protocols has small size due to the cost of class-wise data re-balance for Train-CBL, but the class and attribute distributions are still the same as Fig. 5 (d). It also worth noting that although the attribute distribution is not strictly balanced in Fig. 5 (d), it has the lowest attribute STD, making the Test-GBL more balanced in attributes than other splits.

**Algorithm 2** Generating Test-GBL for MSCOCO-GLT

**Input**: $\{(x, y, z)\}$ is a set of object $x$ with corresponding label $y$ and attribute $z$
**Parameter**: $N$ is the number of selected images for each category
**Initialize**: **Test-GBL** $= \{\}$
**for** $i$ in **Y do**
  //Y is the set of category
  $count = 0$
  $z_{dist} = [0, 0, 0, ...]$
  **while** $count < N$ **do**
    $Min_{STD} = MAX$
    $Cancidate = None$
    **for** $(x, y, z)$ `in` $\{(x, y, z)\}$ **do**
      **if** $y = i$ **then**
        $temp_{dist} = z_{dist} + z$
        $temp_{dist} = Norm(temp_{dist})$
        $temp_{STD} = STD(temp_{dist})$
        **if** $temp_{STD} < Min_{STD}$ **then**
          $Min_{STD} = temp_{STD}$
          $Cancidate = (x, y)$
          $z_{candidate} = z$
        **end if**
      **end if**
    **end for**
    **Test-GBL**.append($Candidate$)
    $z_{dist} = z_{dist} + z_{candidate}$
    $count+ = 1$
  **end while**
**end for**
**Output**: **Test-GBL**

## E Experimental Results

Due to the limited space, we simplified the Table 1 and Table 3 in the original paper by skipping the results of some detailed splits, *e.g*, $\text{Many}_A$, $\text{Medium}_A$, $\text{Few}_A$ in GLT protocol for Table 1 and $\text{Many}_C$, $\text{Medium}_C$, $\text{Few}_C$ for Table 3. Therefore, we complete the corresponding parts in this supplementary material.

As we can see from Table 1, the detailed $\text{Many}_A$, $\text{Medium}_A$, $\text{Few}_A$ of GLT protocol are quite similar to those in the ALT protocol, so the conclusions drawn from the Table 2 of the original still hold at here. We can notice that the effects caused by the attribute-wise imbalance is different from those caused by class-wise imbalance, as there are no precision-accuracy trade-off in $\text{Many}_A$, $\text{Medium}_A$, $\text{Few}_A$. It further proves that why the previous trends [26] of improving both head and tail categories are actually trying to solve the GLT with attribute-wise imbalance, because extracting more attribute-invariant features can benefit both head and tail categories.

**Table 2. Evaluation of CLT and GLT Protocols on MSCOCO-GLT**: This is a supplementary table of the Table 3 in original paper. (Top-1) Accuracy | Precision of previous LT algorithms and their variants equipped with the proposed IFL are reported. All methods are re-implemented under the same codebase with ResNext-50 backbone to ensure fair comparisons

| | Methods | Class-wise Long Tail (CLT) Protocol | | | | | | | | Generalized Long Tail (GLT) Protocol | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | < **Accuracy** \| **Precision** > | $\text{Many}_C$ | | $\text{Medium}_C$ | | $\text{Few}_C$ | | Overall | | $\text{Many}_C$ | | $\text{Medium}_C$ | | $\text{Few}_C$ | | Overall | |
| Re-balance | Baseline | 80.86 | 71.21 | 74.79 | 76.35 | 51.83 | 87.01 | 72.34 | 76.61 | 74.41 | 65.54 | 66.58 | 69.81 | 38.75 | 81.08 | 63.79 | 70.52 |
| | cRT [7] | 80.18 | 71.18 | 75.71 | 75.43 | 57.50 | 85.19 | 73.64 | 75.84 | 73.00 | 64.34 | 67.42 | 68.00 | 44.00 | 76.32 | 64.69 | 68.33 |
| | LWS [7] | 80.50 | 69.85 | 74.46 | 75.21 | 54.42 | 87.20 | 72.60 | 75.66 | 73.55 | 63.57 | 66.08 | 67.57 | 40.42 | 80.87 | 63.60 | 68.81 |
| | Deconfound-TDE [19] | 78.36 | 72.64 | 77.96 | 73.01 | 57.08 | 82.83 | 73.79 | 74.90 | 72.14 | 65.99 | 71.04 | 66.68 | 45.00 | 75.29 | 66.07 | 68.20 |
| | BLSoftmax [16] | 80.41 | 73.53 | 77.38 | 70.53 | 48.92 | 87.86 | 72.64 | 75.25 | 72.32 | 65.88 | 70.63 | 64.21 | 35.83 | 82.30 | 64.07 | 68.59 |
| | Logit-Adj [11] | 80.36 | 74.03 | 78.00 | 77.16 | 61.58 | 81.54 | 75.50 | 76.88 | 73.64 | 66.14 | 68.75 | 69.16 | 47.33 | 70.79 | 66.17 | 68.35 |
| | BBN [27] | 82.32 | 70.35 | 78.33 | 77.35 | 48.58 | **90.22** | 73.69 | 77.35 | 75.82 | 62.96 | 69.21 | 69.59 | 34.25 | 84.70 | 64.48 | 70.20 |
| | LDAM [2] | 82.05 | 75.43 | 80.04 | 74.36 | 54.75 | 88.54 | 75.57 | 77.70 | 76.05 | **69.93** | 72.96 | 66.85 | 39.75 | 79.81 | 67.26 | 70.70 |
| | (ours) Baseline + IFL | 82.09 | 72.40 | 77.50 | 79.29 | 53.67 | 90.03 | 74.31 | 78.90 | 76.09 | 66.53 | 68.08 | **71.80** | 40.00 | 83.57 | 65.31 | **72.24** |
| | (ours) cRT + IFL | 82.55 | 73.93 | 78.58 | 79.34 | 59.83 | 88.13 | 76.21 | 79.11 | **76.77** | 67.28 | 69.25 | 70.25 | 44.08 | 80.94 | 66.90 | 71.34 |
| | (ours) LWS + IFL | 82.27 | 73.69 | 78.29 | 79.39 | 59.83 | 88.82 | 75.98 | **79.18** | 76.73 | 67.30 | 68.58 | 70.65 | 43.83 | 80.88 | 66.55 | 71.49 |
| | (ours) BLSoftmax + IFL | 81.77 | 72.79 | 78.25 | 74.53 | 49.92 | 90.07 | 73.72 | 77.08 | 74.73 | 66.54 | 69.71 | 65.39 | 36.58 | **85.58** | 64.76 | 70.00 |
| | (ours) Logit-Adj + IFL | **82.77** | **75.42** | **79.25** | **79.61** | **62.67** | 84.78 | **77.16** | 79.09 | 75.59 | 68.16 | **69.83** | 70.73 | **48.17** | 72.77 | **67.53** | 70.18 |
| Augment | Mixup [25] | 81.86 | 72.96 | 76.58 | 80.00 | 55.50 | 86.19 | 74.22 | 78.61 | 75.41 | 67.13 | 67.13 | 71.47 | 39.00 | 77.79 | 64.45 | 71.13 |
| | RandAug [3] | 83.27 | 74.70 | 80.04 | 80.76 | 58.50 | 87.62 | 76.81 | 79.88 | 77.18 | 68.07 | 71.46 | 72.68 | 42.83 | 81.40 | 67.71 | 72.73 |
| | (ours) Mixup + IFL | 84.59 | **75.20** | 80.17 | **83.01** | **59.42** | **91.38** | 77.55 | **81.78** | 78.73 | **69.93** | **72.21** | **75.50** | **43.92** | 82.52 | **68.83** | **74.84** |
| | (ours) RandAug + IFL | **84.86** | **75.20** | **81.42** | 82.12 | 57.17 | 89.85 | **77.71** | 81.10 | **78.82** | 69.19 | 71.42 | 73.88 | 42.08 | **82.93** | 68.16 | 73.97 |
| Ensemble | TADE [20] | 83.77 | 76.29 | 81.46 | 75.19 | 51.92 | 90.80 | 76.22 | 78.84 | 75.68 | 68.78 | 73.83 | 66.27 | 37.33 | 85.58 | 66.98 | 71.22 |
| | RIDE [20] | 83.82 | 77.49 | **84.00** | 77.38 | 56.75 | 91.44 | 78.29 | 80.33 | **77.09** | **70.22** | **74.54** | 68.50 | 41.08 | 83.23 | 68.59 | 72.20 |
| | (ours) TADE + IFL | 83.91 | 75.78 | 81.63 | 75.74 | 52.83 | **92.14** | 76.53 | 79.15 | 76.82 | 68.55 | 73.88 | 68.13 | 37.08 | **88.10** | 67.38 | 72.42 |
| | (ours) RIDE + IFL | **85.05** | **78.36** | 83.21 | **78.68** | **58.83** | 89.04 | **78.86** | **80.70** | **77.09** | 69.64 | **74.54** | **70.75** | **43.50** | 81.59 | **69.09** | **72.57** |

In Table 2, we completed the $\text{Many}_C$, $\text{Medium}_C$, $\text{Few}_C$ for both CLT and GLT protocols in MSCOCO-GLT, where $\text{Many}_C$ contains Top-1 to Top-11 frequent categories, $\text{Medium}_C$ has Top-12 to Top-23 categories, and the rest belongs to $\text{Few}_C$. The proposed variants of IFL still outperform the corresponding LT methods in most cases. But we notice that the previous strong GLT baseline RandAug + IFL is now worse than another GLT baseline Mixup + IFL. We believe that this is probably caused by the weird distribution in Fig. 5 (a), where one single super head class "person" contains over 40% of the entire data, so how to learn better boundaries between "person" class and other classes becomes the biggest problem, making the Random Augmentation [3] less effective than Mixup [25], because the latter can ensure the boundaries having linear transition between classes.

## References


1. Abdar, M., Pourpanah, F., Hussain, S., Rezazadegan, D., Liu, L., Ghavamzadeh, M., Fieguth, P., Cao, X., Khosravi, A., Acharya, U.R., et al.: A review of uncertainty quantification in deep learning: Techniques, applications and challenges. Information Fusion (2021) 4
2. Cao, K., Wei, C., Gaidon, A., Arechiga, N., Ma, T.: Learning imbalanced datasets with label-distribution-aware margin loss. NeurIPS (2019) 1, 9, 12
3. Cubuk, E.D., Zoph, B., Shlens, J., Le, Q.V.: Randaugment: Practical automated data augmentation with a reduced search space. In: CVPR Workshops (2020) 5, 9, 12
4. He, K., Chen, X., Xie, S., Li, Y., Dollár, P., Girshick, R.: Masked autoencoders are scalable vision learners. arXiv:2111.06377 (2021) 4
5. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016) 8

6. Hinton, G., Roweis, S.T.: Stochastic neighbor embedding. In: NeurIPS (2002) 7
7. Kang, B., Xie, S., Rohrbach, M., Yan, Z., Gordo, A., Feng, J., Kalantidis, Y.: Decoupling representation and classifier for long-tailed recognition. In: ICLR (2020) 2, 9, 12
8. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: ECCV. pp. 740–755. Springer (2014) 8
9. Liu, Z., Miao, Z., Zhan, X., Wang, J., Gong, B., Yu, S.X.: Large-scale long-tailed recognition in an open world. In: CVPR (2019) 6
10. Loshchilov, I., Hutter, F.: Sgdr: Stochastic gradient descent with warm restarts. arXiv preprint arXiv:1608.03983 (2016) 1
11. Menon, A.K., Jayasumana, S., Rawat, A.S., Jain, H., Veit, A., Kumar, S.: Long-tail learning via logit adjustment. ICLR (2020) 9, 12
12. Mirza, M., Osindero, S.: Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784 (2014) 2
13. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S.: Pytorch: An imperative style, high-performance deep learning library. In: Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., Garnett, R. (eds.) Advances in Neural Information Processing Systems 32, pp. 8024–8035. Curran Associates, Inc. (2019), http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf 8
14. Patterson, G., Hays, J.: Coco attributes: Attributes for people, animals, and objects. In: ECCV. Springer (2016) 8
15. Prechelt, L.: Early stopping-but when? In: Neural Networks: Tricks of the trade, pp. 55–69. Springer (1998) 1
16. Ren, J., Yu, C., Sheng, S., Ma, X., Zhao, H., Yi, S., Li, H.: Balanced meta-softmax for long-tailed visual recognition. NeurIPS (2020) 9, 12
17. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: Imagenet large scale visual recognition challenge. International journal of computer vision (2015) 3, 6, 7
18. Stone, J.V.: Bayes' rule: a tutorial introduction to bayesian analysis. Sebtel Press (2013) 2
19. Tang, K., Huang, J., Zhang, H.: Long-tailed classification by keeping the good and removing the bad momentum causal effect. NeurIPS (2020) 3, 9, 12
20. Wang, X., Lian, L., Miao, Z., Liu, Z., Yu, S.X.: Long-tailed recognition by routing diverse distribution-aware experts. ICLR (2020) 1, 9, 12
21. Wen, Y., Zhang, K., Li, Z., Qiao, Y.: A discriminative feature learning approach for deep face recognition. In: ECCV (2016) 1
22. Wikipedia: Centaur (2022), https://en.wikipedia.org/wiki/Centaur 3
23. Xie, S., Girshick, R., Dollár, P., Tu, Z., He, K.: Aggregated residual transformations for deep neural networks. In: CVPR (2017) 1
24. Zhang, C., Bengio, S., Hardt, M., Recht, B., Vinyals, O.: Understanding deep learning (still) requires rethinking generalization. Communications of the ACM (2021) 1
25. Zhang, H., Cisse, M., Dauphin, Y.N., Lopez-Paz, D.: mixup: Beyond empirical risk minimization. In: ICLR (2018) 9, 12

26. Zhang, Y., Hooi, B., Hong, L., Feng, J.: Test-agnostic long-tailed recognition by test-time aggregating diverse experts with self-supervision. ICCV (2021) 1, 9, 11, 12
27. Zhou, B., Cui, Q., Wei, X.S., Chen, Z.M.: Bbn: Bilateral-branch network with cumulative learning for long-tailed visual recognition. In: CVPR (2020) 1, 9, 12