Inverse Image Frequency for Long-Tailed Image Recognition

Konstantinos Panagiotis Alexandridis, Shan Luo, Member, IEEE, Anh Nguyen, Jiankang Deng, Member, IEEE, and Stefanos Zafeiriou, Member, IEEE

Abstract—The long-tailed distribution is a common phenomenon in the real world. Extracted large scale image datasets inevitably demonstrate the long-tailed property and models trained with imbalance data can obtain high performance for the over-represented categories, but struggle for the under-represented categories, leading to biased predictions and performance degradation. To address this challenge, we propose a novel de-biasing method named Inverse Image Frequency (IIF). IIF is a multiplicative margin adjustment transformation of the logits in the classification layer of a convolutional neural network. Our method achieves stronger performance than similar works and it is especially useful for downstream tasks such as long-tailed instance segmentation as it produces fewer false positive detections. Our extensive experiments show that IIF surpasses the state of the art on many long-tailed benchmarks such as ImageNet-LT, CIFAR-LT, Places-LT and LVIS, reaching 55.8% top-1 accuracy with ResNet50 on ImageNet-LT and 26.3% segmentation AP with MaskRCNN ResNet50 on LVIS. Code available at https://github.com/kostas1515/iif

Index Terms—Long tail, margin adjustment, image classification, instance segmentation, object detection.

I. INTRODUCTION

GREAT advancements have been made in the field of image recognition due to deep learning techniques and the use of massive parallel computer systems. As a result,

Manuscript received 12 October 2022; revised 31 July 2023; accepted 14 September 2023. Date of publication 12 October 2023; date of current version 24 October 2023. This work was supported in part by the Engineering and Physical Sciences Research Council (EPSRC) Centre for Doctoral Training in Distributed Algorithms under Grant EPS/S023445/1, in part by EPSRC ViTeac Project under Grant EP/T033517/2, in part by EPSRC GNOMON: Deep Generative Models in Non-Euclidean Spaces for Computer Vision and Graphics under Grant EP/X011364/1, in part by EPSRC DEFORM: Large Scale Shape Analysis of Deformable Models of Humans under Grant EP/S010203/1, in part by King’s College London N8EFS PhD. Studentship, in part by the University of Liverpool and Vision4ce, it also made use of the facilities of the N8 Centre of Excellence in Computationally Intensive Research provided and funded by the N8 research partnership and EPSRC [EP/1022167/1]. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Ioannis (Yiannis) Kompatirasaris. (Corresponding author: Shan Luo.)

Konstantinos Panagiotis Alexandridis and Shan Luo are with the Department of Engineering, King’s College London, WCR 2LS London, U.K., and also with the Department of Computer Science, University of Liverpool, L69 3BX Liverpool, U.K. (e-mail: konstantinos.alexandridis@kcl.ac.uk; shan.luo@kcl.ac.uk).

Anh Nguyen is with the Department of Computer Science, University of Liverpool, L69 3BX Liverpool, U.K. (e-mail: anh.nguyen@liverpool.ac.uk).

Jiankang Deng and Stefanos Zafeiriou are with the Department of Computing, Imperial College London, SW7 2AZ London, U.K. (e-mail: j.deng16@ic.ac.uk; s.zafeiriou@ic.ac.uk).

This article has supplementary downloadable material available at https://doi.org/10.1109/TIP.2023.3321461, provided by the authors. Digital Object Identifier 10.1109/TIP.2023.3321461

amazing technologies have been developed in the fields of automation, medicine, transportation and internet of things that make human life better. Most of these technologies use a large amount of data in order to train a deep convolutional neural network that solves the problem at hand. Even though this technique is highly efficient, it relies heavily on the availability of data. Models trained with curated balanced datasets like CIFAR [1], ImageNet [2] and COCO [3] achieve good performance in many image recognition tasks such as classification, object detection and instance segmentation. However, in the real world the data are rarely balanced, instead they follow a long-tailed distribution [4], i.e. the data are imbalanced and not uniform, resulting in a major performance degradation [5], [6], [7]. In essence, the models trained with long-tailed data can recognise the frequent (head) classes of the dataset but they fail to recognise the rare (tail) classes. As a consequence, the models that disregard the long-tailed nature of the problem, become unreliable and might raise serious concerns in critical scenarios (e.g. autonomous driving).

The cause of the performance drop in long-tailed datasets is classification imbalance [7], [10]. In detail, the frequent classes of those datasets dominate the training procedure and the network learns more about them and less about the rare classes. One way to solve this problem is to collect more samples from the rare categories so that in the end the data
distribution will be balanced. Unfortunately, this solution costs a lot of effort and it cannot address the issue completely, as the more samples one gathers, the more categories will appear making the annotation procedure intractable. For example, if one wants to gather more images of a rare class i.e., “remote control” object, then one should also annotate the “television” object and perhaps all other objects that appear inside the living room scene that will have a higher frequency than the “remote control”. This is a natural phenomenon of our physical world that the object frequencies follow the Zipf’s law [4], making the class distribution long-tailed.

Recent approaches tackle long-tailed classification by improving the classification layer of the model [8], [9], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20]. Margin adjustment techniques like [8], [9], [11], [12], [13], [20], are popular and intuitive classifier learning methods that have a strong theoretical foundation in label distribution shift and demonstrate performance.

However, most of margin adjustment techniques use an additive hand-crafted margin [8], [9], [11], and [20] that is suitable for image classification but falls short of downstream tasks such as long-tailed instance segmentation because of the background class. For example, given a background object proposal, the additive margin adjustment may force the rare class logits of the background proposal to become positive. Consequently, false positive detections will be produced by confusing the background class for a rare class as shown in the top branch of Figure 1. In contrast to this, our method uses a multiplicative margin, which only amplifies the original logits, without changing their sign, thus it does not produce false positives as shown in the bottom branch of Figure 1. This problem does not apply in long-tailed classification because all categories are foreground and the trade-off exists only between frequent and rare classes. In long-tailed instance segmentation however, a trade-off exists between both foreground and background classes and between frequent and rare categories.

To make this concrete, we use the TIDE toolkit [21] to measure the false positive detections and average precision (AP) [22] of popular margin-adjustment methods like Logit Adjustment [9] and Balanced Softmax [8]. TIDE has some limitations, i.e., its error metrics do not complement AP ($\Delta AP + AP \neq 1$). Nevertheless, it is useful for comparing the relative errors among models. TIDE breaks down the error into many types such as classification, localisation and miss-detection. In this analysis, we use only $\Delta AP_{FP}$ which is the $AP^50$ performance loss due to false positives. As shown in Figure 1, Softmax has low $AP$, because it fails to detect rare categories, i.e., it has low $AP_r$. Hand-crafted margin techniques like Logit Adjustment [9] and Balanced Softmax (BSCE) [8] boost the performance of rare classes but they make a lot of false positives as they have increased $\Delta AP_{FP}$ compared to Softmax.

There are a few ways to reduce false positives in long-tailed instance segmentation. Recent works [8], [12], [13], [23] calculate learnable margin transformations during two-stage learning. However, their margins cannot be easily explained and it requires additional training resources to learn them. Other works, disentangle foreground from background classes by introducing an objectness branch or use zero margin for the background class. However, it is difficult to find a suitable margin for the background class, as this depends on the architecture of the detector (i.e., two-stage vs one-stage) and it cannot be calculated from the dataset.

Motivated by this, we develop a strong dataset-dependent margin adjustment technique called Inverse Image Frequency (IIF). Our IIF uses a multiplicative adjustment, thus it reduces false positives compared to additive adjustment methods, as it only amplifies the original predictions keeping their sign unchanged, as shown in Figure 1, bottom branch. Moreover, our vanilla IIF method has the best instance segmentation performance and produces less false positives as it achieves lower $\Delta AP_{FP}$, compared to similar margin adjustment techniques like Balanced Softmax (BSCE) [8] and Logit Adjustment (Log. Adj.) [9].

At the same time, it achieves strong performance on long-tailed image classification reaching 55.8% top-1 accuracy on ImageNet-LT when using ResNet50 [24] backbone surpassing the state-of-the-art methods by up to 3%. Moreover, it outperforms the state-of-the-art methods in the long-tailed instance segmentation LVIS benchmark [5] boosting the rare category performance by 17.5% compared to vanilla Softmax.

We describe our contributions as follows:

- We show that previous handcrafted margin adjustment techniques used in classification may produce false positives in long-tailed instance segmentation as a result of background class.
- We develop a robust margin adjustment method IIF that boosts the performance of rare categories and makes fewer false positive detections compared to other margin adjustment methods.
- We evaluate our IIF method on CIFAR10-LT, CIFAR100-LT, ImageNet-LT, Places-LT, LVISv1 and we show that it surpasses the state-of-the-art methods by a significant margin.

II. RELATED WORK

Long-tailed image recognition has received a lot of interest in recent years and many works have been developed, a summary of them can be found in these surveys [10], [25]. Many long-tailed datasets have been created for object classification [4], [20], scene classification [4], [17], species classification [26], faces recognition [27], [28], object detection [5], [29] and instance segmentation [5]. Recently, more datasets [30], [31], [32] and works [33], [34] were proposed that tackle both the long-tailed and domain adaptation problem simultaneously. The datasets are created either by extracting datasets from the wild, or by sub-sampling balanced datasets. These datasets can be characterised by their imbalance factor $\beta = n_{max}/n_{min}$ which is the ratio between the maximum and minimum class frequency on the training set. As shown in Table I, the most imbalanced dataset is LVIS [5] and the least imbalanced is CIFAR-LT [20]. Note that COCO [3] is artificially balanced in the sense that all classes have a large and diverse set of images. However, COCO has a larger imbalance factor than common long-tailed classification datasets as the class
frequencies are not uniform. This is due to the fact that COCO is a densely annotated scene-centric dataset and this makes it difficult to have totally balanced classes due to the Zipfian distribution. Regarding the testing set in these datasets, most of them adopt a balanced test set, but the performance is evaluated fairly on all categories. For the case of object detection, it is difficult to have balanced test set, due to scene-centric images. Despite that, when using mAP, every category is evaluated independently and has equal contribution to the final performance.

Many solutions have been developed inside the long-tailed paradigm and they can be categorised in representation learning and classifier learning techniques as shown in Table II.

### A. Representation Learning

A simple representation learning technique is to re-sample the data distribution [5], [13], [35]–[38], by either oversampling or downsampling the classes of datasets. Despite having satisfactory performance, oversampling requires additional computing resources and may cause overfitting for tail classes while undersampling does not exploit efficiently the available data and may cause underfitting for head classes.

Other representation learning techniques enhance the quality of the deep feature extractor by using contrastive learning and supervised learning [44], [45]. However, such methods require a laborious multi-stage training pipeline or the construction of multi-branch networks in order to combine the supervised and contrastive objectives effectively.

Some techniques enhance the feature extractor by generating rare category samples [40], [41], [42], [43]. However, generated features are usually perturbed versions of the old features thus they improve the quantity, rather than the quality of features. In addition to this, distillation methods [15], [39] have been proposed to efficiently exploit the representation quality of larger capacity models. These methods have good results, but they require additional training resources for learning the teacher models. Fusing methods use a two-branch network trained with random and oversampling strategy [47], [48] or learn ensemble models [46] that specialise in rare and frequent categories. They have shown good performance but it is at expense of additional training resources. Finally, data augmentation methods such as mixup [49], cutmix [50], label smoothing [51], [52] and AutoAugment [53], [54] improve the generalisation ability of the model for all classes.

### B. Classifier Learning

1) **Cost-Sensitive Learning**: [16], [17], [18], [19], [63] methods assign costs to samples according to the dataset’s distribution in order to balance the training and learn all classes. They can produce good results without the need of extra training resources but they require careful calibration, hyper-parameter tuning and they are difficult to design and optimise as the costs may be excessive and destabilise training.

2) **Gradient Balancing**: References [23], [55], [56], [57], [58] and [59] methods assign weights to the gradients produced by positive and negative samples, or use different activation functions for gradient balancing [60]. These techniques are most useful in long-tailed object detection and long-tailed instance segmentation as in such tasks the special background class magnifies the imbalance and increases the complexity of the task.

3) **Two-Stage Techniques**: References [7], [12], [13], [14], and [61] first optimise the model to classify the head classes and in the latter stage, they finetune or retrain it for the rare classes. This is achieved using re-sampling techniques, weight normalisation techniques or transfer learning so that in the end the model can classify both head and tail classes effectively. This technique can alleviate the bias of the classifiers and it is task agnostic. Nevertheless, it may require the construction of a complex pipeline and additional training resources.

4) **Margin Adjustment**: References [8], [9], [11]–[13], [20], and [62] alter the decision boundary of the classifier either during training or a posterior to shift the predicted distribution. The resulting classification boundary is closer to the head classes and further away from the tail classes and the feature space of head classes becomes smaller while the space of tail classes is enlarged. This way, during inference the adjusted classifier is less biased towards predicting the head classes.

The margin adjustment techniques produce good results, but they have limitations. For example, [8], [9], [11], and [20] use an additive adjustment for long-tailed image classification but this may produce many false positives in downstream tasks such as long-tailed instance segmentation, as they do not explicitly model the background class margin. Moreover, learnable margin transformation techniques [12], [13] require a two-stage strategy and therefore additional computing resources. They alleviate false positives in downstream tasks as they learn foreground and background category margins simultaneously but their margins are difficult to explain.

In contrast to these, our I1F uses dataset-dependent margins that are easy to explain and use in both long-tailed image classification and long-tailed instance segmentation.

---

**TABLE I**

| Dataset        | β | Train Distribution | # of Images |
|-----------------|---|--------------------|-------------|
| CIFAR-LT [20]  | 100 | Exponential       | 30K         |
| ImageNet-LT [4] | 256 | Pareto            | 186K        |
| Places-LT [4]   | 500 | Pareto            | 62.5K       |
| COCO [3]        | 1,325 | Balanced        | 118K        |
| LVISv1 [5]      | 50,552 | Long-tailed    | 99K         |

**TABLE II**

| Family          | Method                     | Reference |
|-----------------|----------------------------|-----------|
| Representation Learning | Re-sampling | [3], [13], [35]–[38] |
|                   | Distillation               | [15], [39] |
|                   | Feature Generation         | [40]–[43] |
|                   | Contrastive Learning       | [44], [45] |
|                   | Fusion                     | [46]–[48] |
|                   | Data Augmentation          | [49]–[54] |
| Classifier Learning | Cost-sensitive Learning | [16]–[19] |
|                   | Gradient Re-balancing      | [23], [55]–[60] |
|                   | Two-Stage Training         | [7], [12], [12]–[14], [61] |
|                   | Margin Adjustment          | [8], [9], [11]–[13], [20], [62] |
III. Preliminaries

\textbf{IIF} is closely related to ideas from label distribution shift, we follow a similar analysis as in [8], [9], and [12]. Let \( p_s(y|x) \) and \( p_t(y|x) \) be the source and target distributions respectively. By using the Bayes theorem, the source distribution can be written as:

\[
p_s(y|x) = \frac{p_s(x|y)p_s(y)}{p_s(x)}
\]

and the target distribution can be written as:

\[
p_t(y|x) = \frac{p_t(x|y)p_t(y)}{p_t(x)}
\]

If one assumes that the data generating functions are equal \( p_s(x|y) = p_t(x|y) \) then by dividing Eq. 1 and Eq. 2, one can rewrite the target distribution as:

\[
p_t(y|x) = \frac{c(x)p_t(y)}{p_t(x)}
\]

where \( c(x) = \frac{p_s(x)}{p_t(x)} \). During training \( p_s(y|x) \) is approximated by the model \( f_s(x; \theta) \) and a scorer function \( s(x) = e^t \):

\[
p_s(y|x) \propto e^{f_s(x; \theta)}
\]

By using Eq. 3 and Eq. 4 one can compensate for label distribution shift using the following Equation:

\[
p_t(y|x) \propto \frac{1}{c(x)} p_t(y)\exp(f_s(x; \theta)) = e^{f_s(x; \theta) + \log(p_t(y)) - \log(p_s(y)) - \log(c(x))}
\]

During inference, one is interested to predict a single class \( \tilde{y} \) and this is usually achieved by taking the maximum value of Eq. 5:

\[
\tilde{y} = \arg \max_y (f_s(x; \theta) + \log(p_t(y)) - \log(p_s(y)) - \log(c(x)))
\]

Moreover, one can simplify Eq. 6 by eliminating \( c(x) \) because it is invariant to \( \arg \max_y \) as follows:

\[
\tilde{y} = \arg \max_y (f_s(x; \theta) + \log(p_t(y)) - \log(p_s(y)))
\]

Using Eq. 7 one can solve the label distribution shift problem. However, in the real world, \( p_s(y) \) and \( p_t(y) \) may be unknown. Luckily, one can still solve the label shift problem by estimating \( p_t(y) \) and \( p_s(y) \) from the data.

A. Training-Set Distribution

First, even though \( p_s(y) \) is unknown, one has access to a training set \( D \) that is sampled uniformly from the source distribution \( s \). Thus, instead of calculating \( p_s(y) \) one can use \( p_D(y) \), which is the class distribution on the training set. Accordingly, as \(|D|\) grows larger then one can be more certain that \( p_D(y) \) will be a good estimate of \( p_s(y) \).

B. Test-Set Distribution

Generally, \( p_t(y) \) can be any arbitrary distribution and when \( p_t(y) \neq p_s(y) \) there exists label shift. If the label shift is unknown, i.e. \( p_t(y) \) is not known, then it can be estimated using the model’s predictions as suggested in [64]. In the case of long-tailed image recognition, the target distribution is uniform because the test set is balanced. The reason is that, in long-tailed visual benchmarks, every category is evaluated fairly and it contributes equally to the final performance [4], [8], [9], [20]. Therefore, \( p_t(y) = \frac{1}{C} \) where \( C \) is the total number of classes in the dataset.

To this end, we can rewrite Eq. 7 as:

\[
\tilde{y} = \arg \max_y (f_s(x; \theta) - \log(p_D(y)))
\]

In essence, Eq. 8 suggests that one can compensate for this label distribution shift by translating the model’s output \( f_s(x; \theta) \) by the training set’s class probability:

\[
z' = z - \log(p_D(y))
\]

C. Limitations

Despite that, in downstream tasks like instance segmentation or object detection, there is also the special background class \( b \), that depends on the model’s configuration, i.e., one-stage detectors [65], [66] have a different estimate of \( b \) than two-stage detectors [67], [68] that use region proposals. The background class is usually handled by predicting C+1 categories using softmax, but in this way, Eq. 9 is not directly applicable as \( p_D(y = b) \) cannot be easily calculated. Additionally, if a bad background probability estimate is used, this may cause false positives and deteriorate the model’s performance.

Some works [12], [13] alleviate this problem by learning the foreground and background class margins during two-stage learning but these margins are difficult to explain, and may cause concerns in safety critical applications. Other works like [23] use an objectness branch to reduce false positives. They predict two extra logits that determine whether the sample belongs to foreground and background respectively. This technique disentangles the classification task to two sub-tasks, i.e. background and foreground prediction; in this way, foreground class margins can be applied easily to foreground samples. However, the usage of objectness branch hurts the model’s Fixed-AP performance [69], as it only improves the cross category rankings as suggested by [69]. Recently, Hsieh et al. [57] studied the background category problem and propose DropLoss, a loss that assigns weights to background gradients in an adaptive manner. However, they utilised a gradient re-balancing method which is different from margin adjustment techniques.

For these reasons, we develop \textbf{IIF} using dataset-dependent margins that are easy to explain and use in long-tailed classification and long-tailed instance segmentation. Our \textbf{IIF} alleviates for label distribution shift in long-tailed benchmarks. At the same time it uses a multiplicative adjustment, that keeps the original sign of the predictions unchanged thus it reduces the false positive detections compared to other additive margin adjustment methods.
IV. METHODOLOGY

A. Inverse Image Frequency

Inverse Image Frequency (IIF) is inspired by Inverse Document Frequency (IDF). IDF is an important heuristic method that reweights textual terms according to their relevance and it has been extensively used in text retrieval tasks [70], [71], [72], [73]. IDF reweights a term according to the number of documents the term appears in the corpus. In our work, instead of measuring the number of documents where a term appears, we measure the number of images where an object appears.

In detail, given a set of training images \( D \) sampled from the source distribution \( s \), Image Frequency \( IF(y, D) \) of a class \( y \in \mathbb{N} \) is computed as the number of images in which an object \( o_y \) appears:

\[
IF(y, D) = |\{image \in D : o_y \in image\}|
\]

The class probability \( p_D(y) \) of class \( y \) is defined as:

\[
p_D(y) = \frac{IF(y, D)}{K}
\]

where \( K = \sum_{y=1}^{C} IF(y, D) \) and \( C \) is the total number of classes in \( D \). Next, IIF is measured by taking the logarithm of the inverse of \( p(y) \), i.e.,

\[
IIF(y) = \log\left(\frac{K}{IF(y)}\right) = -\log(p(y))
\]

Next, one can transform the logits \( z \) of the classification layer using the IIF transformation:

\[
z_{IIF} = -z \log(p(y))
\]

This feature transformation is similar to the IDF feature transformation whose justification is explained in [74].

The use of the logarithm is convenient because it links the probability space \((0, 1)\) to real space enhancing the compatibility of the predicted logits \( z \) and IIF weights. Other link functions are discussed in Table III.

When IIF is multiplied with the logits of the classification layer, it redistributes the weights across different classes. The weights of IIF are larger for the rare classes than the frequent classes thus, it can be used to remove the frequent category bias and alleviate class imbalance.

The Eq. 13 resembles Eq. 9. Its difference is that instead of additive adjustment, it performs multiplicative adjustment. The multiplicative adjustment benefits both long-tailed classification and long-tailed instance segmentation as it alleviates class imbalance and it makes fewer false positive detections since it maintains the sign of the original predictions intact. If the detector predicts logits that are negative for one background region, then an additive adjustment may force them inside the detection threshold, making the model overconfident and producing false detections. In contrast to that, the multiplicative adjustment will only amplify the logits, in other words, it will not affect their sign and keep background predictions outside the detection threshold as shown in the bottom branch of Figure 1.

1) Connection to Softmax: In practice, neural networks typically produce a probabilistic vector \( q \) by using a softmax output layer \( \sigma \). This converts the logit \( z_i \) for each class \( i \) into a probability \( q_i \), by comparing \( z_i \) with the other logits \( q_{\sigma} = \frac{\exp(z_i)}{\sum_{j=1}^{C} \exp(z_j)} \). The dominant prediction \( q_i \) can be found by computing \( \arg\max_{i \in C} z_i \), and this holds true because all \( z_i \) are activated by the same strictly increasing activation function \( f(x) = e^x \). Changing the base of the activation function from \( e \) to any \( \alpha \in \mathbb{R} \) would not affect the ranking and this is fair for balanced datasets. For imbalanced datasets, we can change the base of the activation function for each \( z_i \) according to the inverse class probability i.e., \( f^*(x) = (\frac{x}{\sum x})^\alpha \) and compensate for imbalance. In this way, we allow logits that correspond to rare classes to get easily activated. We can achieve this by applying IIF transformation Eq. 13:

\[
q_{IIF,i} = \frac{\exp(z_i \log(\frac{1}{p(i)}))}{\sum_{j=1}^{C} \exp(z_j \log(\frac{1}{p(j)}))} = \left(\frac{1}{p(i)}\right)^z_i \frac{1}{\sum_{j=1}^{C} \left(\frac{1}{p(j)}\right)^z_j}
\]

Note that, here the class index starts from 1 to C, but in the case of instance segmentation there exist a background class \( b \) that is usually encoded as the “0” class. In this case, there could be \( C+1 \) classes in softmax and the index starts from 0 to C.

Equation 14 has two beneficial properties. First, it maintains the property that \( \sum_{i=1}^{C} q_{IIF,i} = 1 \), this means that \( q_{IIF,i} \) is a probabilistic vector, the proof is provided in supplementary material. Secondly, re-balancing occurs naturally, as logits \( z_i \) corresponding to frequent classes i.e., \( p(i) \to 1 \) will not contribute as much in softmax because they will be irrelevant, \( (\frac{1}{p(i)})^{z_i} \to 1, \forall z_i \). On the other hand, logits \( z_i \) corresponding to rare classes i.e., \( p(i) \to 0 \), will determine the final outcome of softmax \( (\frac{1}{p(i)})^{z_i} \to +\infty \). To make the second point concrete, one can consider an extreme example of binary classification where \( p(y = 0) = 0.999999 \) and \( p(y = 1) = 0.000001 \). IIF will significantly downgrade \( z_0 \) rendering it irrelevant and it will make \( z_1 \) the dominant factor in softmax. In other words, IIF will make softmax more sensitive to \( z_1 \) than \( z_0 \) which is the class that matters most in this hypothetical example. Compared to previous works that perform additive adjustment, IIF makes stronger adjustment, because of the multiplicative function, and it enlarges the rare class probabilities with a faster rate than the additive case.

2) Variants: Moreover, one can define IIF variants by using different log bases or different link functions in order to transform the probability space into the real space. The motivation is that the imbalance factor changes according to the dataset thus, it may be beneficial to use different variants that provide stronger debiasing effects.

In Table III, some basic variants are summarised and in Figure 2 their behaviour is illustrated.

- The raw IIF is the most straightforward way to transform the probabilities into weights. The different log-bases can be used to control the magnitude of the weights when dealing with very low probabilities.
The smooth IIIF has similar behaviour to raw IIIF, but it has the advantage of handling zero image frequency values, thus it can be used either on the full training dataset D or on the mini-batch d in online fashion using the mini-batch statistics.

- The relative IIIF uses the inverse logit link function and it has a larger range of values than the smooth or raw IIIF. It is a symmetrical function around 0.5 and it is useful when modelling binary events. In the long tail scenario, usually most class probabilities are below 0.5 thus the relative link will produce only positive weights and will have similar behaviour with the raw IIIF.

- The Normit IIIF assumes that the data follow Gaussian Distribution. It is similar in properties to the relative IIIF, it is also symmetrical around 0.5, but it has a smoother slope.

- The Gombit IIIF assumes that the data follow Gompertz Distribution. It uses an asymmetrical link function that puts more emphasis to small probability events as it produces increasingly larger positive weights compared to high probability events. In other words, the growth rate for the response value is larger as the probability gets smaller.

In addition to Inverse Image Frequency, one can calculate Inverse Object Frequency IOF by counting object instances instead of images. In tasks such as image classification, IIIF and IOF will produce the same result as objects and images have a one-to-one relationship. For other tasks such as instance segmentation, multiple objects can coexist in a single image thus the two methods are different and they produce different weights.

3) IIIF Cross Entropy: IIIF can be integrated during training by optimising the IIIF Cross Entropy loss. Let \( \mathbf{y} \) be the ground truth one-hot encoded vector of class \( y \) then by using Eq. 14 the loss is:

\[
CE_{IIIF}(q, \mathbf{y}) = -\sum_{i=1}^{C} y_i \log(q_{IIIF,i})
\]

\[
= -\log \left( \frac{\exp(z_y \log \left( \frac{1}{p(y)} \right))}{\sum_{j=1}^{C} \exp(z_j \log \left( \frac{1}{p(j)} \right))} \right)
\]

\[
= -z_y \log \left( \frac{1}{p(y)} \right) + \log \left( \sum_{j=1}^{C} \left( \frac{1}{p(j)} \right)^z_j \right) \quad (15)
\]

It can be seen that when the class probability is higher i.e., \( p(y) \rightarrow 1 \), then there is loss only for the negative classes and the network does not receive any information about the target class \( y \). On the other hand, when the class probability is low, i.e., \( p(y) \rightarrow 0 \), then the positive class \( y \) dominates the loss, forcing the model to focus on the rare class. In the end, this allows the model to learn more about the categories whose class probability is low.

For long-tailed image classification, all classes are foreground and IIIF can be used without modifications. That is not the case for long-tailed instance segmentation as there exist background samples. In instance segmentation, many models encode the background samples as the “0” class. Thus they predict a logit vector \( z = [z_0, z_1, \ldots, z_C] \). To apply our IIIF in this case, we set the background weight as 1, i.e., \( IIIF = [1, -\log(p(1)), \ldots, -\log(p(C))] \), to keep the background object’s estimation unaltered and only change the foreground objects’ estimations.

Using IIIF Cross Entropy Loss Eq. 15, the gradient is shown in Eq. 16. The proof is provided in appendix.

\[
\frac{\partial CE_{IIIF}}{\partial z_i} = \begin{cases} 
-\log(p(i))(q_{IIIF,i} - 1) & \text{if } i = y \\
-\log(p(i))q_{IIIF,i} & \text{otherwise}
\end{cases} \quad (16)
\]

It can be seen that the positive gradient, i.e., when \( i = y \) will be larger in magnitude when the class probability \( p(i) \) of the target class is low. This will encourage the model to learn more about the rare classes of the dataset. Additionally, the negative gradients i.e., when \( i \neq y \), will be weighted according to their class probabilities. This means that negative gradients occurring from frequent categories will be suppressed. In the end, using IIIF the model becomes more sensitive to rare classes as their gradients will be upweighted.

4) Post-Process IIIF: Moreover, IIIF can be applied during inference using Eq. 14 as a post-processing method. If post-processing IIIF is applied then it is no longer necessary to use Eq. 15. Instead, vanilla Cross Entropy can be used to train a model and only during inference Eq. 14 can be injected into the model’s output in order to de-bias the predictions. In conclusion, all IIIF strategies can be illustrated in Fig. 3.

B. IIIF Cross Entropy Versus Cost Sensitive Learning

IIIF Cross Entropy re-weights the gradient of each sample \( i \) according to its Inverse Image Frequency \( -\log(p(i)) \), as shown in Eq. 13. This differs from Cost-Sensitive Learning.
method (CSL) that re-weights all samples based on scalar weight $\alpha_i$. In principle, CSL applies weight multiplication to the loss rather than the logits, more details on CSL can be found in this work [19]. The gradient of CSL for a sample $i$ is:

$$\frac{\partial L_{CSL}}{\partial z_i} = \begin{cases} \alpha_i (q_i - 1) & \text{if } i = y \\ \alpha_i q_i & \text{otherwise} \end{cases}$$

To better understand how CSL differs from IIF, we can set $\alpha_i = -\log(p(y))$ and assume that $q_i = q_{IIF,i}$. The positive gradient of CSL, (i.e. $i = y$), will be the same with IIF whereas the negative will differ. In CSL, the negative gradient will be multiplied by the scalar $-\log(p(y))$ which is the weight of the target class, whereas in IIF it is multiplied by its class respective weight $-\log(p(i))$. The latter balances negative gradients more efficiently and suggests that positive and negative gradients should not be valued the same.

In practice, CSL might be unstable during the early phases of training because of the imbalance between positive and negative gradients which is magnified by the weight $\alpha_i$. Consequently, it requires careful hyperparameter tuning to balance the dynamics in mini-batch training. IIF on the other hand, suppresses the imbalance caused by the negative gradients as it re-weights them based on their class probabilities.

C. Connection to Other Works

IIF has a similar idea to recent successful margin adjustment techniques shown in Table IV as all of these methods reweight the logits based on either probabilities, image frequencies or learnable weights. The important detail of IIF is that it does multiplicative adjustment, thus it makes fewer false positives than additive handcrafted margin adjustment techniques for downstream tasks. Moreover, since it is a dataset-dependent method, it is easier to interpret and justify than other learnable margin adjustment approaches. In addition to this, our method can also be compared to calibration techniques such as Platt Scaling [75], Temperature Scaling [62], [76] or NorCal [77]. In comparison to [75], IIF reweights the classification logits based on dataset statistics rather than learnable parameters, in contrast to [76] it uses class specific weights instead of a single global temperature and different from [62], [77] IIF does not require additional hyperparameters.

V. LONG-TAILED CLASSIFICATION EXPERIMENTS

A. Datasets and Evaluation

In long-tailed image classification, we use CIFAR10-LT and CIFAR100-LT with exponential imbalance ratio 100 as in [20], ImageNet-LT [4] and Places-LT [4] following the common long-tailed classification protocol. These datasets show a significant label shift as they have long-tailed train distribution and balanced test distribution. The balanced test distribution is artificially constructed so that the model’s performance can be fairly evaluated on each class. These datasets have the characteristics described in subsection III-A and III-B and our method can alleviate their shift from long-tailed distribution to balanced distribution. To measure the performance of IIF we use top-1 accuracy following the common evaluation protocol.

B. Implementation Details

We have observed that the standard implementation is suboptimal and can be significantly enhanced. Therefore, we create Squeeze-and-Excitation (SE) [78] ResNets to increase the capacity of our representation models. We choose this attention mechanism as it adds minimal complexity and has good performance. We use an SE-ResNet32 with reduction factor $r = 4$ for CIFAR-LT, SE-ResNet50 and SE-ResNeXt50-4 x 32 with $r = 16$ for ImageNet-LT. For Places-LT, we pre-train a SE-ResNet152 with $r = 16$ on full ImageNet and then we finetune it according to [4]. For all SE modules we use the Average squeeze operator and the Sigmoid excitation operator and all linear layers have the same dimensions as in [78] implementation. The ResNet implementation follows official Pytorch implementation [79]. All models are trained using Pytorch framework and 4 Nvidia V100 GPUs.

1) Longer Training: We have observed that training for more epochs improves the performance of the representation model. For CIFAR-LT datasets we use a batch size of 64 and a training schedule of 400 epochs, a learning rate of 0.1 with learning rate decay at epoch 360 and 380. For ImageNet-LT, the model is trained for 200 epochs using a batch size of 256, a learning rate of 0.2 and cosine learning schedule. For Places-LT [4] dataset we use an ImageNet pre-trained ResNet152 backbone. Then, we finetune its last residual block and classifier for 30 epochs using batch size 256, learning rate 0.1 and cosine scheduler.

2) Regularisation and Augmentations: We use cosine classifier with scale $s = 16$ for ImageNet-LT and CIFAR-LT and learnable scale for Places-LT. Moreover, we use Mixup [49] with the factor of 0.2 for all datasets. Regarding augmentations, we use the optimal AutoAugment [53] policies.
for CIFAR-LT and ImageNet-LT and RandAugment [54] for Places-LT. In addition to this, we observed that the recommended weight decay used in CIFAR-LT [20] and ImageNet-LT [13] is suboptimal for our model. After conducting a grid search, we found that the value 1e-4 works well for all datasets and improves the performance. Our findings confirm that weight decay tuning is important and should not be overlooked as also mentioned in [80].

3) Two-Stage Strategy: Inspired by [13], we perform experiments using the two-stage strategy when training the models with IIF. We use random sampling in all stages. In the first stage we pre-train the models using Softmax Cross-Entropy and in the second stage, we retrain only the classifier’s weights using IIF. For ImageNet-LT, we use a learning rate of 2e-5 and we train the classifier for 5 epochs; for Places-LT, we use 1e-5 and we train for 10 epochs; and for CIFAR-LT, we use a learning rate of 1e-4 and we train the classifier for 20 epochs.

C. Classifier Learning Using IIF

1) Training Strategies: We start our analysis by studying strategies to improve classification using IIF. We explore IIF as decoupled strategy, as a post-processing method and as a cost-sensitive learning method.

Table V suggests that using IIF with decoupled training achieves the best performance, reaching 84.1% in CIFAR10-LT, 48.9% in CIFAR100-LT and 56.0% in ImageNet-LT. The decoupled training is better than end-to-end training because the representations are learned more efficiently with Softmax Cross Entropy rather than other techniques as described in [13]. After learning the representations, IIF can be used to retrain only the classifier and remove the frequent category bias from the model. Moreover, IIF has good results when used as a post-processing method. Under this setup, the model is first trained with Softmax and only during inference the IIF weights are injected. This technique achieves slightly worse results than decoupled IIF, because it does not involve re-training the classifier. However, it does not cost additional computing resources and it is useful when there are computing limitations. In particular for CIFAR datasets, post-processing IIF drops the performance by −0.9% while for ImageNet-LT it achieves the same result compared to the decoupling strategy. This is due to the fact that, the CIFAR-LT datasets have larger variance than ImageNet-LT thus the decoupling strategy allows the model to explore better solutions and achieve slightly better results.

In the end, decoupled-IIF is best as it improves the performance in CIFAR10-LT by 5.5%, in CIFAR100-LT by 5.9 and in ImageNet-LT by 3.8% compared to Softmax. Regarding the datasets, the best performance boost is observed in CIFAR100-LT, because this dataset has larger vocabulary than CIFAR10-LT and it is less complex than ImageNet-LT.

2) IIF Variants: Next, we explore the IIF variants listed in Table III with respect to the post-processing strategy and the decoupled training strategy. Starting from the post-processing IIF strategy, as Table VI indicates, the best variant for CIFAR10-LT and CIFAR100-LT datasets is smooth IIF that improves the performance by 5.4% and 5.3% respectively. For ImageNet-LT the best variants are the raw, gombit and relative IIF as they boost performance by 3.8%. Other variants produce similar results for ImageNet-LT, except for Normit IIF. This is because ImageNet-LT has a large vocabulary and the majority of its class probabilities are within a specific range of values that cause similar re-weighting for most IIF variants.

In the end, smooth IIF is the best choice as it generalises better than other variants and achieves the best performance in both small and large vocabulary datasets under various imbalance factors.

Notice that the variants raw, base2 and base10 have the same performance (i.e. 83.2%, 48.0%, 56.0% for CIFAR10-LT, CIFAR100-LT and ImageNet-LT respectively) under the post-processing strategy. That’s because they produce exactly the same rankings, however, when using the decoupled training strategy, they have slightly different results due to different optimisation.

To illustrate this, we use the decoupled IIF strategy with random sampling. As Table VII suggests, smooth IIF has the best performance for CIFAR10-LT as it boosts the performance by 6.0%. Regarding CIFAR100-LT, the gombit IIF has the best performance as it surpasses Softmax by 6.0%. Finally, in ImageNet-LT the raw, the relative and the base10 have the best performances boosting the accuracy by 3.8%.

Under the decoupling strategy, we notice that for datasets CIFAR100-LT and ImageNet-LT all variants except for normit IIF produce similar results and their differences in performance are marginal. This is because the class probabilities in these datasets are within a small range of values that produce similar weights when using the aforementioned IIF variants.

| Method         | Strategy     | CIFAR10-LT | CIFAR100-LT | ImageNet-LT |
|----------------|--------------|------------|-------------|-------------|
| Softmax        | End-to-End   | 78.6       | 43.0        | 52.2        |
| IIF            | Post-Process | 83.2       | 48.0        | 56.0        |
| IIF            | Decoupled    | 84.1       | 48.9        | 56.0        |

| Variant        | CIFAR10-LT   | CIFAR100-LT | ImageNet-LT |
|----------------|--------------|-------------|-------------|
| Raw            | 84.1         | 48.9        | 56.0        |
| Smooth         | 84.6         | 48.8        | 55.8        |
| Rel            | 81.2         | 48.8        | 56.0        |
| Gombit         | 84.4         | 48.9        | 56.0        |
| Normit         | 82.9         | 49.0        | 55.9        |
| Base2          | 84.1         | 48.9        | 55.9        |
| Base10         | 84.4         | 48.9        | 56.0        |
| Raw/Rel        | 84.1         | 48.9        | 56.0        |

| Variant        | CIFAR10-LT   | CIFAR100-LT | ImageNet-LT |
|----------------|--------------|-------------|-------------|
| Raw            | 84.1         | 48.9        | 56.0        |
| Smooth         | 84.6         | 48.8        | 55.8        |
| Rel            | 81.2         | 48.8        | 56.0        |
| Gombit         | 84.4         | 48.9        | 56.0        |
| Normit         | 82.9         | 49.0        | 55.9        |
| Base2          | 84.1         | 48.9        | 55.9        |
| Base10         | 84.4         | 48.9        | 56.0        |
In the end, the smooth \( \text{IIF} \) is the best variant as it achieves the best performance in CIFAR10-LT and generalises well to both CIFAR100-LT and ImageNet-LT.

In conclusion, we use decoupled strategy as it produces better results than the post-processing \( \text{IIF} \) strategy. Regarding the variants, we use smooth \( \text{IIF} \) because it provides good performance and it generalises better than other \( \text{IIF} \) variants in all datasets and strategies.

**D. Comparison With Other Methods**

Long-tailed image classification has been advancing rapidly during the recent years and diverse solutions have been proposed. We compare our method against many families of methods such as:

- **Two Stage Methods.** We show the efficacy of our \( \text{IIF} \) by comparing it to other two stage methods such as DisAlign [12], LWS [13], cRT [13] and MiSLAS [81].

- **Self-supervised.** We highlight the simplicity and stronger performance of \( \text{IIF} \) against self-supervised methods such as Hybrid SC [44] and DRO-LT [45].

- **Higher Capacity Models.** \( \text{IIF} \) is additionally compared against higher capacity models like ensemble RIDE [46], knowledge distilled CBD [15] and DiVE [39] and two branch network BBN [47].

- **Margin Adjustment.** Finally, \( \text{IIF} \) is compared with other margin adjustment techniques like Balanced Softmax [8], LADE [11] and Logit Adjustment [9].

In summary, for our models we use smooth \( \text{IIF} \) with decoupled strategy as this produces the best performance.

We compare \( \text{IIF} \) in common long-tailed classification benchmarks such as CIFAR-LT, ImageNet-LT and Places-LT and we show that \( \text{IIF} \) surpasses the state-of-the-art.

1) *ImageNet-LT:* Our method has on average better top-1 accuracy than all state-of-the-art methods on ImageNet-LT as shown in Table VIII. Our \( \text{IIF} \) significantly surpasses the best two-stage DisAlign method by 2.9% on average accuracy using ResNet50 and by 2.8% using ResNeXt50. Secondly, it overcomes the best margin adjustment LADE method by 3.2% using ResNeXt50 under a similar training budget. Additionally, it outperforms higher capacity models like ensemble RIDE by 1.4% and self-supervised models like DRO-LT by 2.3% using ResNet50. Furthermore, it outperforms knowledge distilled models like CBD by 4.2% using ResNet50 and DiVE by 3.1% using ResNeXt50, having a more straightforward training pipeline.

2) *CIFAR-LT:* Our method shows great performance on the CIFAR-LT datasets as well, highlighting its generalisation ability. As Table IX suggests, \( \text{IIF} \) surpasses the best margin adjustment method LADE [11] by 3.4% on CIFAR100-LT. Moreover, it overcomes the best two-stage MisLAS by 2.5% on CIFAR10-LT and by 1.8% on CIFAR100-LT. Furthermore, it outperforms self-supervised methods like the Hybrid SC method [44] by 3.2% on CIFAR10-LT and by 2.1% on CIFAR100-LT. Finally, it is better than ensemble methods like RIDE by 1.8% on CIFAR100-LT, using a single model.

3) *Places-LT:* Finally, in Table X the results on Places-LT are displayed. \( \text{IIF} \) outperforms all other methods on average accuracy, achieving 40.2% top-1 accuracy. It achieves an absolute 9.1% increase compared to Softmax and 4.3% increase compared to OLTR [4]. Additionally, it surpasses the margin adjustment LADE method, by an overall 1.4% in top-1 accuracy and the two-stage DisAlign by 0.9%.

4) *Comparison Against LWS:* Our \( \text{IIF} \) significantly surpasses the LWS method in both ImageNet-LT and Places-LT datasets. LWS [13] uses multiplicative adjustment as shown in Table IV, like our \( \text{IIF} \). However, the class margins in LWS need to be learned in two stage training, whereas in \( \text{IIF} \), the margins can be injected during inference using Post-Process \( \text{IIF} \), without the need for classifier retraining. Furthermore, the margins of \( \text{IIF} \) are easier to explain as they are calculated.
VI. LONG-TAILED INSTANCE SEGMENTATION

In the previous section, we showed that IIF can achieve good performance in long-tailed classification. In this section, we show that IIF can generalise to downstream tasks such as long-tailed instance segmentation.

A. Experiment Setup

1) Dataset: We use LVIS version 1 (LVISv1) which contains 99k images for training and 19.8k images for validation. LVISv1 is a heavily class imbalanced dataset that contains 1,203 categories that are grouped according to their image frequency into rare categories (those with 1-10 images in the dataset), common categories (11 to 100 images in the dataset) and frequent categories (those with > 100 images in the dataset). We report our results using average mask precision AP, average mask precision for rare AP_r, common AP_c and frequent categories AP_f and average box precision AP_b. The imbalance factor of LVISv1 is shown in Table I.

2) Implementation Details: We use a plethora of architectures such as MaskRCNN [68], Cascade MaskRCNN [83] and Hybrid Task Cascade [84]. For our intermediate experiments, we use the 1x schedule in order to reduce the computational time and still showcase the performance of IIF. When we compare against the state-of-the-art we use a longer training schedule that is 2x and standard model enhancements such as Cosine Classifier [12] and Normalised Mask [23]. We also use RFS [5] as our sampling policy and FASA [42] as our augmentation policy and we train all models using the MMdetection framework [85].

B. IIF in Long-Tailed Instance Segmentation

We analyse IIF in conjunction to training strategies, sampling strategies, IIF variants and model architectures. Unless specified, for all experiments we use MaskRCNN with ResNet50 as our main architecture.

1) Training Strategies: The task of long-tailed instance segmentation is different and more complex than long-tailed classification as it contains the special background class, it has a larger imbalance factor as shown in Table I and the target distribution is not uniform but long-tailed. For this reason, we examine two strategies of applying IIF: either end-to-end training or decoupled strategy. As shown in Table XI the best strategy is end-to-end training, as this gives the best mask AP and box AP_b. In detail, the 12-epoch schedule IIF boosts mask AP by 4.6% and box AP by 2.6% while the 24-epoch schedule increases mask AP by 3.4% and box AP 2.5% compared to Softmax. The decoupled strategy also increases the performance by 3.1% in mask AP and by 1.9% in box AP compared to Softmax trained for 12-epochs. However, decoupling strategy costs more training resources and achieves lower performance than the end-to-end training. To this end, end-to-end training is better for long-tailed instance segmentation in contrast to long-tailed classification where decoupled training works best. The reason is that the long-tail instance segmentation task is a finetuning task which typically uses a backbone pretrained on ImageNet-1K. Thus, the network has already learned good representations and IIF can be used end-to-end to finetune the model in the downstream task. Also, end-to-end training is preferable because it converges faster than decoupled training as shown in Table XI. Finally, end-to-end IIF training is a superior strategy because it reduces not only the foreground imbalance but also the foreground to background imbalance, allowing the model to distinguish rare categories from the background.

2) Sampling Strategies: Next, we examine sampling strategies, in particular, oversampling and random sampling. In contrast to long-tailed classification, the oversampling strategy is essential to the performance of long-tailed instance segmentation methods and many works use it [8], [23], [57], [59], [77]. Similar to these works, we explore RFS sampling [5], and random sampling and we compare them with end-to-end training and decoupling strategy. As shown in Table XII, the best sampling strategy is RFS [5] used in End-to-End (E2E) for 24 epochs as this has the best overall AP. In detail, it achieves 22.9% in overall mask AP and 23.5% in overall box AP. It also increases the AP_r by 1.0% and AP_b by 1.6% compared to random sampling used in End-to-End training for
24 epochs. However, this technique reduces the performance of frequent categories slightly by 0.4% compared to end-to-end random sampling, as also noted by [5]. Moreover, the end-to-end 12-epoch RFS schedule achieves the best AP, adding a further boost of 1.7%, but at the same time it lowers the performance of the frequent categories 1.6%, compared to the end-to-end 24-epoch RFS schedule. This indicates that there exists a trade-off for frequent and rare categories that depends on the training schedule, i.e. the longer schedule may be suboptimal for the rare categories but it benefits frequent categories and vice versa. Regarding the decoupling strategy, we use three different sampling combinations for the two stage training; random sampling for both stages, random sampling first and RFS secondly and finally RFS for both stages. All decoupling strategies require more training resources and have worst performance than training end-to-end. This is because, in long-tailed instance segmentation, the backbone is already pretrained on Imagenet-1K, thus the decoupled strategy converges slower than the end-to-end training. In the end, we adopt the end-to-end 24-epoch schedule as this has the best overall performance.

3) Extension to Deeper Architectures: We show the generalisability of 1IF by applying it to the popular instance segmentation models such as MaskRCNN, Cascade MaskRCNN and Hybrid Task Cascade using 1x schedule and random sampling. As shown in Table XIII, 1IF improves the performance of all models significantly. Furthermore, the gains in performance become larger as models become deeper linearly, which indicates that our method can generalise well to larger architectures. 1IF increases MaskRCNN ResNet50 by 4.6%, MaskRCNN ResNet101 by 4.6%, MaskRCNN ResNeXt101 by 4.9%, Cascade MaskRCNN ResNet101 by 5.4% and Hybrid Task Cascade ResNet101 by 5.6% in overall mask AP. Moreover, 1IF increases the performance of all categories, both head and tail, for all architectures. This is due to the fact that even frequent categories may have lower expectations compared to the dominant background class, especially for the edge locations of an image. 1IF can alleviate such imbalance and increase the performance for all categories, thus it is a robust method for long-tailed instance segmentation.

4) 1IF Variants: We conduct an extensive ablation study of different 1IF variants in Table XIV. All 1IF variants significantly improve the baseline in both mask and box AP and the best variant is base10 1OF.

The base10 1OF achieves 20.0% overall box AP and 19.9% overall mask AP and it boosts the detection performance by 4.5% and segmentation performance by 5.7% for rare categories. There are other variants that achieve better segmentation performance for rare categories like the relative 1IF that boosts performance by 7.3%. However, in the task of long-tailed instance segmentation it is better to opt for a variant that achieves high bounding box performance, as this enables the mask AP to improve further by combining this technique with other sampling strategies and methods. In our experiments we have observed that, during MaskRCNN inference, the bounding box performance, determines the segmentation performance, thus the bounding box performance is the bottleneck. For this reason, we use base10 1OF to achieve the best possible box AP and this enables the creation of better models as we show in the following section.

5) 1IF Enhancements: We use the base10 1OF variant and end-to-end training. Moreover, we use standard techniques that have been previously used by other state-of-the-art such as Normalisation Mask [23], Cosine classifier [12], RFS [5] and FASA [42]. Additionally, we use a stricter Non-maximum suppression threshold that is 0.3, mask threshold of 0.4 and a longer training schedule that is 2x. Starting from the Softmax model, we replace the Dot-product Classifier with Cosine Classifier following [12], this adds 1.4% in mask AP. Next, we adopt a Normalisation Mask [23] that further increases

| LVISv1.0 | Box AP | Mask AP |
|---------|-------|---------|
| Variant | Method | AP | AP0 | AP50 | AP75 | APs | APt | AP | AP0 | AP50 | AP75 | APs | APt |
| Baseline | Softmax | 16.1 | 66.3 | 16.9 | 0.4 | 10.5 | 29.2 | 13.2 | 24.4 | 16.1 | 0.5 | 10.6 | 26.9 |
| Raw | | 19.5 | 34.7 | 18.3 | 6.6 | 15.4 | 29.7 | 19.8 | 32.3 | 20.7 | 7.5 | 17.7 | 27.6 |
| Smooth | | 19.0 | 34.3 | 18.0 | 5.4 | 14.9 | 29.6 | 19.5 | 32.0 | 20.3 | 6.8 | 17.3 | 27.6 |
| Relative | 1IF | 19.7 | **48.8** | 18.9 | **6.8** | 15.5 | 30.0 | **19.9** | **32.4** | 20.8 | 7.3 | 17.8 | **27.9** |
| Base2 | | 19.0 | 34.3 | 17.7 | 6.4 | 14.5 | 29.5 | 19.5 | 31.9 | 20.4 | 7.6 | 17.0 | 27.6 |
| Base10 | | 19.5 | 33.2 | 19.7 | 3.2 | 16.6 | 29.9 | 19.2 | 30.9 | 20.2 | 4.2 | 17.7 | 27.4 |
| Normit | | 19.3 | 33.1 | 19.5 | 2.5 | 16.3 | **30.1** | 19.0 | 30.7 | 19.9 | 3.7 | 17.2 | 27.7 |
| Gombit | | 19.7 | 34.7 | 19.1 | 5.9 | 16.0 | 29.9 | 19.8 | 32.3 | 20.7 | 6.9 | 17.9 | 27.7 |

| TABLE XV ABLATION STUDY OF COMPONENTS USED WITH 1IF |
|-------------|--------|--------|--------|---------|--------|
| Cos. Cls. | Norm. M | PASA | RFS | base10-1OF | AP |
| ✓ | ✓ | ✓ | ✓ | ✓ | 18.7 |
| ✓ | ✓ | ✓ | ✓ | ✓ | 20.1 |
| ✓ | ✓ | ✓ | ✓ | ✓ | 20.8 |
| ✓ | ✓ | ✓ | ✓ | ✓ | 23.3 |
| ✓ | ✓ | ✓ | ✓ | ✓ | 25.0 |
| ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | 24.1 |
| ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | 26.3 |
the performance by 0.7%. Recently, Zang et al. proposed FASA [42] which is a novel feature augmentation technique. Using FASA we further improve the model by 2.5%.

Using IIF in addition to these methods, the model’s performance is further increased by 0.8%. If we adopt RFS [5] as our sampling strategy, this further increments the performance by 1.7% compared to FASA. Finally, using base10 IOF, a stricter NMS threshold of 0.3 and mask threshold of 0.4, we further increase performance by 1.3% achieving 26.3% in overall mask AP.

C. Comparison to Other Methods

We compare our IIF method against the state-of-the-art in Table XVI. Using ResNet50, our method has the best overall segmentation performance, surpassing EQLv2 [56] by 0.8% and NorCal [77] by 1.1% in overall AP. Furthermore, our method achieves the best AP in common and frequent categories. Also, it increases the AP by 7.6%, APb by 17.5%, APc by 9.0%, APi by 1.6% and APb by 6.3% compared to vanilla Softmax. We further investigate the performance of ResNet50-RSB [88] which is a ResNet50 backbone pretrained with better augmentations and regularisations. As the compared methods did not use this backbone, we have reproduced them for fair comparison. Using ResNet50-RSB [88], IIF surpasses the state-of-the-art in overall mask and box performance reaching 27.4% in both metrics. Also, it outperforms NorCal [77] by 0.3% and RFS [5] by 1.7%. Moreover, it achieves the best performance in rare, common and frequent categories reaching 19.4%, 26.9% and 32.1% respectively.

We notice that IIF generally improves all categories, which is different from long-tailed image classification where there is a performance trade-off between rare and frequent categories. This is because in long-tailed instance segmentation the trade-off is not only between foreground but also between background and foreground categories. In long-tailed segmentation, the background samples dominate the training process and render all foreground classes as the minority. Thus, using IIF all categories can benefit resulting in the general performance boost.

D. Model Analysis

Inspired by [13] we analyse the weight norms of the classification layer of MaskRCNN trained with our method. As seen in Figure 5, IIF produces a more balanced weight norm distribution compared to Softmax. In this way, it removes classification bias by increasing the norms associated with rare classes and decreasing the norms associated with frequent classes. Lastly, we compare instance segmentation results of our method against Softmax, shown in five images from LVIS validation set using MaskRCNN in Figure 4. Our IIF model recognises correctly the rare classes like the parrot, owl, horse-carriage and giant panda, in contrast to vanilla Softmax, that either classifies them as the common classes bird, polar bear or does not recognise them at all. However, not all rare categories can be correctly recognised with IIF as our method did not detect the eagle in the last image. Nevertheless, IIF shows promising results as it predicted a more interesting and rare class that is duck instead of the common class bird, probably because the context around the image is water. This shows that this method can be further improved by explicitly modeling the context around the images or by capturing the relationship between objects inside the image.

TABLE XVI

| Method       | Backbone            | AP   | APb  | APc  | APi  | APb  |
|--------------|---------------------|------|------|------|------|------|
| Softmax      |                     | 18.7 | 1.1  | 16.2 | 29.2 | 19.5 |
| EQL [55]     |                     | 21.6 | 3.8  | 21.7 | 29.2 | 22.5 |
| DropLoss [57]|                     | 22.3 | 12.4 | 22.3 | 26.5 | 22.9 |
| Forest-RCNN  | [87]                | 23.2 | 14.2 | 22.7 | 27.7 | 24.6 |
| RFS† [5]     | ResNet-50-FPN       | 23.7 | 13.3 | 23.0 | 29.0 | 24.7 |
| FASA [42]    |                     | 24.1 | 17.3 | 22.9 | 28.5 | -     |
| DisAlign [12]|                     | 24.2 | 13.2 | 23.8 | 29.3 | 24.7 |
| NorCal [77]  |                     | 25.2 | 19.3 | 24.2 | 29.0 | 26.1 |
| EQLv2 [56]   |                     | 25.5 | 17.7 | 24.3 | 26.7 | 26.1 |
| IIF† (ours)  |                     | 26.3 | 18.6 | 25.2 | 30.8 | 25.8 |

| Softmax      |                     | 23.4 | 8.4  | 22.5 | 30.8 | 23.1 |
| EQL† [55]    |                     | 23.9 | 14.0 | 23.4 | 28.9 | 236  |
| RFS† [5]     | ResNet-50-FPN (RSB) | 25.4 | 13.0 | 25.5 | 30.9 | 24.9 |
| FASA† [42]   |                     | 25.5 | 14.3 | 25.2 | 30.7 | 24.9 |
| DropLoss† [57]|                     | 25.7 | 14.4 | 26.6 | 29.7 | 25.1 |
| NorCal† [77] |                     | 27.1 | 18.4 | 26.6 | 31.5 | 26.8 |
| IIF (ours)   |                     | 27.4 | 19.4 | 26.8 | 31.5 | 27.4 |

VII. DISCUSSIONS

As presented in the above experiments, IIF has proven to be a robust method that can be used in many long-tailed tasks such as long-tailed classification and long-tailed instance segmentation to boost the performance of rare categories. Moreover, it generalises well to many backbones and architectures and therefore it can be a valuable component to long-tailed methods.

As shown in classification, IIF can be used either as a post-processing strategy or as decoupled strategy. The decoupling strategy has slightly better performance than the post-processing strategy but it costs additional training. After exploring many IIF variants, we showed that IIF is robust and we used the smooth IIF variant as this produced the best performance. On the other hand, IIF uses weight multiplication. This may be disadvantageous as it makes the
Fig. 4. MaskRCNN-ResNet50 detections on LVISv1 validation set using Softmax versus our IIF method. IIF can correctly detect rare classes such as parrot, owl, horse-carriage and giant panda in contrast to Softmax method. However, both methods fail to detect the rare class eagle in the last image.

Fig. 5. Visualisation of MaskRCNN classifier’s weight norms on LVIS using Softmax and IIF. IIF produces a more balanced weight-norm distribution in comparison to Softmax, thus it reduces the frequent category bias.

In this work, we proposed the novel Inverse Image Frequency (IIF) to address the long-tailed problem that is a common issue in most real-world datasets. Our method reweights the classification logits of the deep model to improve the recognition performance of the rare classes in the dataset. We investigated IIF with many training strategies and variations on four classification datasets, one instance segmentation dataset and one object detection dataset. We showed that decoupled smooth IIF works the best in the classification task; the end-to-end base-10 IOF works the best in the long-tailed instance segmentation task. Our IIF models can largely improve the rare category performance and surpass the state-of-the-art by a large margin (e.g., ∼ 3.0% on ImageNet-LT compared to similar methods and ∼ 0.7% on LVIS in overall performance), thus it can serve as a valuable component in the long-tailed recognition methodology. Our models can be used in a variety of applications such as autonomous vehicles, Internet of Things and medical applications where data follow a long-tailed distribution. In the future, we will expand IIF to other tasks such as semantic segmentation and few-shot learning and explore optimal sampling strategies to further boost the performance of rare classes.

REFERENCES

[1] A. Krizhevsky and G. Hinton, “Learning multiple layers of features from tiny images,” M.S. thesis, Univ. Toronto, Toronto, ON, Canada, 2009.
Anh Nguyen received the Ph.D. degree from the Italian Institute of Technology (IIT), Italy. Previously, he was with the Hamlyn Centre for Robotic Surgery, Imperial College London, Australian National University, Inria, and The University of Adelaide. He is currently a Lecturer (an Assistant Professor) and directs the Smart Robotic Laboratory, Department of Computer Science, University of Liverpool. He is an Associate Editor of *IEEE Transactions on Medical Robotics and Bionics*. His research goal is to develop methods that enable robots to see and act like humans.

Jiankang Deng (Member, IEEE) received the Ph.D. degree from Imperial College London (ICL), supervised by Prof. Stefanos Zafeiriou and funded by the Imperial President’s Ph.D. Scholarships. He is currently an Honorary Lecturer with the Department of Computing, Imperial College London. His research interests include deep learning-based face analysis, including detection, alignment, reconstruction, recognition, and generation. He has achieved more than 9K citations to his research work. He is one of the main contributors to the widely used open-source platform insightface.

Stefanos Zafeiriou (Member, IEEE) is currently a Professor in machine learning and computer vision with the Department of Computing, Imperial College London, London, U.K., and an EPSRC Early Career Research Fellow. He served as an Associate Editor and a Guest Editor in various journals, including *IEEE Transactions on Pattern Analysis and Machine Intelligence* and *IJCV*. He has been a guest editor of more than eight journal special issues and co-organized over 16 workshops/special sessions on specialized computer vision topics in top venues. He has coauthored more than 100 journal articles mainly on novel statistical machine learning methodologies applied to computer vision problems, such as 2-D/3-D face analysis, deformable object fitting and tracking, shape from shading, and human behavior analysis, published in the most prestigious journals in his field of research, such as *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *IJCV*, and many papers in top conferences, such as CVPR, ICCV, ECCV, ICML, NeurIPS, and ICLR. His students frequently receive very prestigious and highly competitive fellowships, such as the Google Fellowship x2, the Intel Fellowship, and the Qualcomm Fellowship x4. He has more than 28K citations to his work, an H-index of 72. He was the General Chair of BMVC 2017.