This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Deep Imbalanced Regression via Hierarchical Classification Adjustment

Haipeng Xiong Angela Yao
National University of Singapore
haipeng,ayao@comp.nus.edu.sg

Abstract

Regression tasks in computer vision, such as age estimation or counting, are often formulated into classification by quantizing the target space into classes. Yet real-world data is often imbalanced – the majority of training samples lie in a head range of target values, while a minority of samples span a usually larger tail range. By selecting the class quantization, one can adjust imbalanced regression targets into balanced classification outputs, though there are trade-offs in balancing classification accuracy and quantization error. To improve regression performance over the entire range of data, we propose to construct hierarchical classifiers for solving imbalanced regression tasks. The fine-grained classifiers limit the quantization error while being modulated by the coarse predictions to ensure high accuracy. Standard hierarchical classification approaches, when applied to the regression problem, fail to ensure that predicted ranges remain consistent across the hierarchy. As such, we propose a range-preserving distillation process that effectively learns a single classifier from the set of hierarchical classifiers. Our novel hierarchical classification adjustment (HCA) for imbalanced regression shows superior results on three diverse tasks: age estimation, crowd counting and depth estimation. Code is available at `https://github.com/xhp-hust-2018-2011/HCA`.

1. Introduction

Data imbalance is a critical issue in deep learning. When learning from long-tail distributions, deep networks may be biased toward frequent head classes and perform poorly on tail classes. To ensure strong performance on the entire output space, many balancing strategies have been developed, such as reweighting [31, 44] and logit adjustment [24]. While these approaches improve tail performance, they invariably do so at the expense of head classes. Moreover, tail classes may also suffer from data insufficiency, leading to over-fitting on the limited training samples.

While many methods have been developed for imbalanced classification, few works have tackled the imbalanced regression problem [12, 26, 37]. The few existing works naively adopt imbalanced classification techniques directly into a regression setting, by extending reweighting [37] or logit adjustment [26] into the regression loss. However, such approaches do not address the inherent data imbalance at the heart of the problem. Therefore, they suffer similar drawbacks as the original long-tail classification techniques and also trade off head or tail performance.

In this work, we advocate for the reformulation of regression into classification. This is already done for many tasks in computer vision, such as depth estimation [6], age estimation [27] and crowd-counting [21], with good performance because classification is more tolerant to label noise [40]. Most commonly, the continuous output is quantized, and each bin is treated as a class. A key benefit that has been overlooked for the conversion is the ability to rebalance the class distribution. An imbalanced distribution of targets can be adjusted into a balanced one by applying a distribution specific quantization. For example, a long-tail distribution can be balanced by adopting a logarithmic quantization [9].

In the quantization process, the number of classes should be selected to ensure sufficient samples to avoid overfitting the minority class. However, the larger the interval, *i.e.* to ensure balanced and sufficient class samples, the greater the quantization error when recovering the target regression values. In practice, the number of classes and quantization scheme is chosen to trade off classification and quantization errors [6, 40] to minimize regression errors. With a single class quantization, it is impossible to ensure both balance (sufficiency) and small quantization error. A coarse quantization may produce an accurate classifier but suffer from extreme quantization errors; a fine-grained quantization limits quantization errors, but may not be so accurate.

Can we merge different classifiers such that we benefit from the higher performance of coarse classifiers while preserving the resolution of fine classifiers? In this work, we explore the *adjustment* of fine-grained classifiers with progressively coarser ones, where the output resolution is preserved while the activated range is adjusted to be more

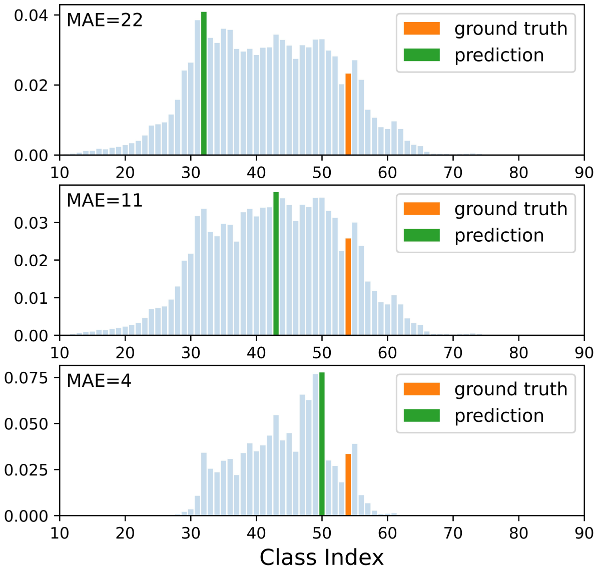


Figure 1. Hierarchical classification adjustment (HCA). The top plot shows normalized logits of the finest classifier H; progressing downwards, classifier H is adjusted by coarser classifiers ($1 \sim H-1$), which bring the prediction closer to the ground truth.

accurate. We refer to this procedure as Hierarchical Classification Adjustment, or HCA. HCA works with the logits of the classifier ensemble; as shown in Fig. 1, adding the coarser (but more accurate) logits progressively improves the accuracy. We also theoretically analyze the error of hierarchical classifiers under data insufficiency and imbalance and show why HCA is helpful in this case.

In addition, we propose to distill the entire hierarchical ensemble into a single classifier; we refer to this process as HCA-d. To ensure that estimated target ranges remain consistent in the distillation process (see detailed example in Fig. 2), we propose a range-preserving adjustment. Overall, HCA-d is simple but efficient, showing improvements over the whole range of the target space. Our contributions can be summarized as:

- A novel Hierarchical Classification Adjustment (HCA) that adjusts a fine classifier with an ensemble of progressively coarser classifiers over an imbalanced target range;
- A theoretical analysis of HCA for data insufficiency and imbalance which is empirically verified.
- A range-preserving distillation technique, HCA-d, which ensures consistent class (range) predictions predicted across the hierarchy of classifiers.
- HCA shows comparable or superior performance on imbalanced visual regression tasks, including age estimation, crowd counting and depth estimation.

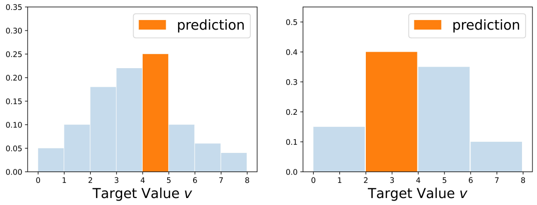


Figure 2. (Left) The prediction of an 8-class classifier. (Right) Downsampling logits by group-summing (eq. (10)) fails to preserve the range.

2. Related Works

Real-world data [29, 45] is commonly imbalanced. Most works [4, 14–16, 18, 22, 25, 31, 38, 44] focus on imbalanced classification, while only a few [12, 26, 32, 43] studied imbalanced settings in a regression.

Imbalanced Classification. Current research working with single classifiers tries to improve tail class performance via sample weighting [4, 31, 44], upsampling [15, 25] or adjusting class margins [19, 24, 46]. There is invariably a tradeoff which sacrifices the head for the tail.

While a single classifier may not be suitable, ensembling multiple classifiers can cover all the classes [14, 17, 18, 37, 42]. In an ensemble, a critical issue is ensuring individual learners' diversity. A classic approach to introduce diversity is bagging [3] - sampling with replacements for different training data partitions for each of the classifiers. More recent approaches focus on different strategies for sampling subsets [14], hard sample mining [18] and progressive splitting [42]. For deep imbalanced regression, it is convenient to get diverse classifiers by applying different quantization strategies without splitting the dataset.

From Imbalanced Classification to Regression Imbalanced regression [12, 26, 32, 43] is less explored than classification. Most works are inspired by classification techniques such as sample reweighting [43], logit adjustment [26]. Conversely, [21] and [36] choose distribution-aware quantization to transform imbalanced regression into a less imbalanced classification problem. We follow this latter strategy, but explore hierarchical classifiers, where each classifier trades off head and tail performances differently while the combination suits the whole range.

Hierarchical classification [2, 7, 10, 39] leverages the taxonomy or hierarchical class structure to ensure more semantically meaningful mistakes. For example, a poodle is better to be mis-classified as a dog instead of a cat. To learn the hierarchy, classifiers are trained together [2]. A key issue is how to align hierarchical outputs and propagate supervision from the coarse to the fine classifiers [2, 7, 10, 39]. The standard approaches [2, 10] treat classifier outputs after the softmax as posterior probabilities and sum them. Such a paradigm does not ensure consistent predictions. For regression, such inconsistencies adversely affect the learning and serve as the motivation for our proposed range-preserving distillation.

3. Hierarchical Classification Adjustment

We propose a Hierarchical Classification Adjustment (HCA) for imbalanced regression. It learns an ensemble of hierarchical classifiers and then leverages the set of predictions to improve the performance of few-shot ranges while maintaining the performance of many-shot ranges. In this section, we first describe how to set a single classifier for a regression problem (Sec. 3.1), then partition the continuous label space into an ensemble of discrete hierarchical classes (Sec. 3.2), followed by the hierarchical adjustment (Sec. 3.3) and distillation (Sec. 3.4). Finally, a theoretical analysis of the error of HCA is provided in Sec. 3.5.

3.1. A Vanilla Classifier for Continuous Targets

Consider a continuous dataset $D = \{x, v\}$, where x and v denote the input and target value, respectively. Let V_{min} and V_{max} denote the minimal and maximal values of v in the training set. Like [21, 41], we divide the target range $[V_{min}, V_{max}]$ into C intervals $(V_{min}, V_1]$, $(V_1, V_2]$, ..., $(V_{C-1}, V_{max}]$ and treat samples within each interval as samples belonging to classes $c = 1...C$. A standard classifier can be trained to estimate the interval index c based on feature representations of x. Consider an input sample x, represented by a feature $f \in \mathbb{R}^d$ extracted by network F:

$$f = F(x), \tag{1}$$

with a predicted class logit $\hat{p} \in \mathbb{R}^C$ where

$$\hat{p} = \text{Softmax}\{G(f)\} \tag{2}$$

and G is a mapping function with learnable weights. For learning F and G, we apply a cross-entropy loss L_{ce} with $\hat{p}$

$$L_{ce} = -\sum_{j=1}^{C} p[j] \times log(\hat{p}[j]). \tag{3}$$

where $p \in \mathbb{R}^C$ is the one-hot ground-truth. We can also apply label smoothing to p_i; the soft ordinal loss (SORD) [8] applies a Gaussian smoothing to ensure that ordinal relationships are partially preserved in the target classes.

After training, the predicted class can be determined by the maximum dimension of $\hat{p}$. The class is then mapped back to a representative regression value for evaluation, *e.g.* by considering the mean or median of samples belonging to the class interval.

3.2. Hierarchical Classifier Ensemble

Consider H classifiers; these classifiers are hierarchical, in that each covers a progressively coarser quantization. The finest quantization is designated the H-th classifier; its classes can be merged to form coarser quantization. For $h = 1$ to $H-1$, the classifier has $C_h = 2^h$ classes, where each class interval's range is determined to normalize the number of data samples per class. For example, for the first classifier ($h=1$), the two classes cover ranges $(V_{min}, V_{med}]$ and $[V_{med}, V_{max})$, where V_{med} is the value selected from V_i that is closest to the median; for $h = 2$, the 4 intervals are selected in V_i to cover quartiles of the data samples. Fig. 3 (a) shows an example of $H = 3$ hierarchical classifiers. We can observe that the label distribution of $1 \sim (H-1)$ hierarchical classifiers is more balanced than the H-th classifier.

The h-th classifier predicts $\hat{p}^h \in \mathbb{R}^{C_h}$ based on

$$\hat{p}^h = \text{Softmax}\{G_h(f)\}, \tag{4}$$

where G_h is a mapping function with learnable weights. Its cross entropy L^h_{ce} can be given as

$$L^h_{ce} = -\sum_{j=1}^{C_h} p^h[j] \times log(\hat{p}^h[j]), \tag{5}$$

where $p^h \in \mathbb{R}^{C_h}$ is the ground-truth for the h-th classifier. The overall loss for training feature network F and hierarchical classifiers G_h is the sum of all the cross-entropies:

$$L = \sum_{h=1}^{H} L^h_{ce}. \tag{6}$$

Note that we do not weight each L^h_{ce} differently since they have the same scale.

3.3. Hierarchical Classifier Adjustment (HCA)

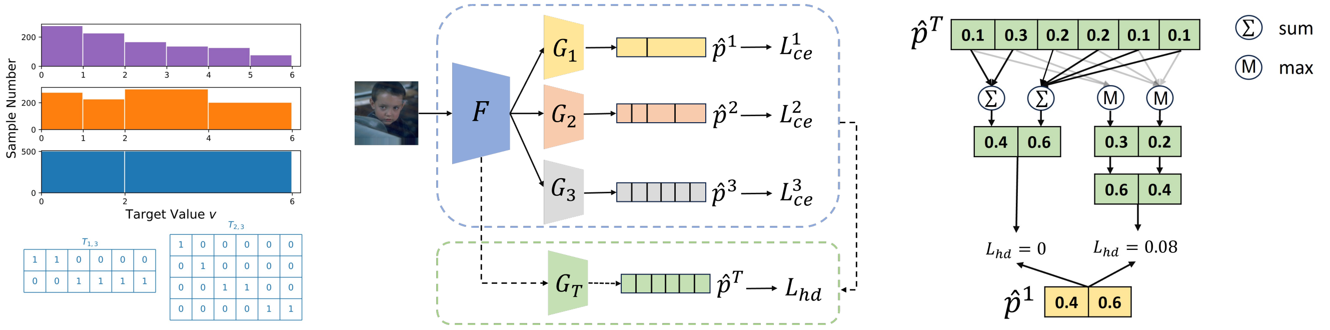
In the ensemble of classifiers learned by Eq. (6), classifier H has the finest quantization (and therefore the lowest quantization error) but is also the least accurate. In contrast, as the classifier gets progressively coarser, it gets more accurate, but also has higher quantization error (see Fig. 3). To merge these results, we can adjust the prediction of classifer H with the coarser classifiers $H - 1$ to 1.

From the hierarchical predictions $\hat{p}^h$, we can estimate an adjusted prediction through a summation operation

$$\hat{p}^a = \hat{p}^H + \sum_{h=1}^{H-1} T^T_{h,H} \cdot \hat{p}^h, \tag{7}$$

or a multiplication operation:

$$\hat{p}^m = log(\hat{p}^H) + \sum_{h=1}^{H-1} T^T_{h,H} \cdot log(\hat{p}^h). \tag{8}$$



(a) (b) (c)

Figure 3. An example of $H = 3$ hierarchical classifiers. (a) Class splitting and label distributions for hierarchical classifiers $1 \sim 3$ from bottom to top. We also visualize class transition matrices $T_{h,H}$ between the h-th and H-th classifiers, which are used to project hierarchical predictions to the finest dimension for merging (eq. (7) and eq. (8)). (b) Training of Hierarchical classifiers (dashed blue box). The dashed green box denotes the second-stage learning, which distillates classifier T from hierarchical predictions $\hat{p}^h$. (c) Distillation learning of classifier T. Hierarchical alignment by sum operation (eq. (10)) and max operation (eq. (13)) are compared. Here we only plot the distillation from $\hat{p}^1$, while from $\hat{p}^2$ and $\hat{p}^3$ are not visualized.

In Eqs. 7 and 8, $\hat{p}^a, \hat{p}^m$ are addition- and multiplication-adjusted predictions that keep the finest quantization as H-th classifier. $T_{h,H} \in \mathbb{R}^{C_h \times C_H}$ is the class mapping from h-th classifier to H-th classifier. If the u-th class in H-th classifier is the v-th class in the h-th classifier, then $T_{v,u} = 1$; otherwise $T_{v,u} = 0$. Fig. 3 (a) visualizes an example of $T_{h,H}$ for $H = 3$ hierarchical classifiers. Note that the multiplication merging in Eq. (8) has a similar form as logit adjustment, but here we use hierarchical prediction $\hat{p}^h$ to adjust $\hat{p}^H$ rather than the frequency of each class. The final class is recovered by taking a max over $\hat{p}^a$ or $\hat{p}^m$ for addition or multiplication adjustments respectively.

HCA, while proposed with the concept of adjusting the finest-quantized classifier with coarser ones, is effectively an ensembling approach, voting with either the logits (Eq. (7)) or log of the logits (Eq. (8)). However, such an ensembling approach cannot ensure that the adjusted or ensembled result $\hat{p}^a$ and $\hat{p}^m$ will predict a final class interval consistent with $\hat{p}^h$.

3.4. Range-Preserving Distillation (HCA-d)

In addition to the inconsistencies, the adjustment procedure, like other ensembling methods, is inefficient because it requires running H classifiers during testing. Alternatively, we propose to distill the ensemble of classifiers into a single adjusted classifier. The ensemble is learned during training in a first stage, frozen, and then distilled into a single classifier during a second stage; during inference, only the adjusted classifier is applied. Such an approach is motivated by hierarchical classification [2, 10], which also distills hierarchical classifiers, though they aim to learn hierarchy-aware features.

Consider a classifier T which predicts $\hat{p} \in R^d$ with a mapping function G_T:

$$\hat{p}^T = \text{Softmax}(G_T(f)), \tag{9}$$

where $\hat{p}^T$ distills the hierarchical information of $\hat{p}^h$. This can be achieved by adopting a Kullback–Leibler divergence loss between the softmax normalized logits $\hat{p}^T$ and $\hat{p}^h$. As $\hat{p}^T_i \in R^{C_H}$ and $\hat{p}^h_i \in R^{C_h}$ have different resolutions, they must be aligned before the distillation.

Previous works on hierarchical classification [2, 10] view $\hat{p}^T \in R^{C_H}$ as posterior probabilities and thus simply sum the corresponding dimensions in $\hat{p}^T$ to get a downsampled versions of $\overline{p}^{T,h} \in R^{C_h}$ to match with $\hat{p}^h$, *i.e.*

$$\overline{p}^{T,h}[j] = \sum_{k=1}^{C_H} T_{h,H}[j,k] \times \hat{p}^T[k]. \tag{10}$$

After aligning $\hat{p}^T$ with the individual $\hat{p}^h$, we can apply the Kullback–Leibler (KL) divergence between $\hat{p}^h_i$ and $\overline{p}^{T,h}_i$:

$$L^h_{\text{hd}} = \text{KL}\{\hat{p}^h || \overline{p}^{T,h}\}, \tag{11}$$

and an overall hierarchical distillation by summing over all the classifiers:

$$L_{\text{hd}} = \sum_{h=1}^{H} L^h_{\text{hd}}. \tag{12}$$

The hierarchical loss in Eq. (11) is not range-preserving when we choose Eq. (10) as the hierarchical alignment. As indicated in Fig. 3 (c), $L_{hd} = 0$ does not indicate the class predicted by $\hat{p}^T$ is within the range of classes predicted by $\hat{p}^h$. We can adjust Eq. (10) to be range-preserving by considering the maximum of $T_{h,H}[j,k]$:

$$\ddot{p}^{T,h}[j] = \max_{k=1,...,C_H} T_{h,H}[j,k] \times \hat{p}^T[k]. \tag{13}$$

and then $\ddot{p}^{T,h}$ is normalized to get $\overline{p}^{T,h} \in R^{C_h}$

$$\overline{p}^{T,h}[j] = \frac{\ddot{p}^{T,h}[j]}{\sum_{l=1}^{C_h} \ddot{p}^{T,h}[l]}. \tag{14}$$

Proposition 1 (Range-Preserving Alignment)**.** *Let $v = argmax_j \overline{p}^{T,h}[j]$, $u = argmax_k \hat{p}^T[k]$. If $\overline{p}^{T,h}$ is computed by eqs.* (13) *and* (14)*, then $T_{h,H}[v, u] = 1$, which indicates the class predicted by $\hat{p}^T$ is within the range of that predicted by $\overline{p}^{T,h}$.*

3.5. Error Analysis

From a theoretical perspective, it is possible to show that the upper bound of MAE for HCA is lower than that of a vanilla classifier. We sketch the case below for a simple case of two classifiers but is easy to extend the result to H classifiers by induction.

Consider hierarchical classifiers G_1 and G_2. Classifier G_1 has C_1 balanced classes, with $n_{1,i} = \frac{N}{C_1}$ samples for i-th class ; G_2 has $C_2 = 2C_1$ imbalanced classes, with $n_{2,j}$ samples for j-th class. Note that i-th class of G_1 correspond to $(2i - 1)$-th and $2i$-th classes in G_2. We first show the upper bound of classification error is related to the sample number per class in Prop. 2 and then compare the MAE of HCA to a vanilla classifier in Prop. 3.

Definition 1. *Following [5], the margin of i-th class of G_h is defined as $\gamma_i^h = \min_{y^h=i} \max_{l \neq y} \hat{p}^h[y^h] - \hat{p}^h[l]$, where y^h is the ground-truth for G_h.*

Definition 2. *Let $Pr(\hat{y}^h = j|y^h = i)$ denote the probability of i-th class in h-th classifier being mis-classified as j-th class by G_h. The classification error of G_h on the i-th class is defined as $L_i^h = \sum_{j \neq i} Pr(\hat{y}^h = j|y^h = i)$.*

Proposition 2 (Generalization Error Bound [5])**.** *With probability $1 - \frac{1}{N^5}$, L_i^h is upper bounded by Δ_i^h:*

$$L_i^h \lesssim \Delta_i^h = \frac{1}{\gamma_i^h}\sqrt{\frac{C(G_h)}{n_{h,i}} + \frac{\log(N)}{\sqrt{n_{h,i}}}} \propto \frac{1}{\sqrt{n_{h,i}}}, \tag{15}$$

where "$\propto$" denotes being proportional to, $C(G_h)$ is some proper complexity measure of function G_h, such as [1, 11], and we use $\lesssim$ to hide some constant factors.

Prop. 2 suggests that few-shot classes tend to have larger error bounds than many-shot classes because the upper bound Δ_i^h is proportional to $\frac{1}{\sqrt{n_{h,i}}}$.

Proposition 3 (MAE of HCA)**.** *If U_2 and U_{HCA} denote the upper bounds of the mean absolute error (MAE) of G_2 and HCA, then we have*

$$U_2 - U_{HCA} \propto \sum_{i=1}^{C_1} (\Delta_{2i-1}^2 + \Delta_{2i}^2 - 2\Delta_i^1) > 0, \tag{16}$$

$$\frac{\Delta_{2i-1}^2 + \Delta_{2i}^2}{2\Delta_i^1} > \frac{\eta_i}{2} \geq \sqrt{2}, \tag{17}$$

where "$\propto$" denotes being proportional to, $\eta_i = \sqrt{1 + r_i} + \sqrt{1 + \frac{1}{r_i}}$ and $r_i = \frac{n_{2,2i-1}}{n_{2,2i}}$.

Prop. 3 eq. (16) implies that the MAE bound of HCA is smaller than G_2, and the reduction of MAE is proportional to the difference in classification error.

Remarks on Data Sufficiency: i) When the data is sufficient ($n_{h,i} \to \infty$), the upper bounds Δ_i^h for a given classifier, as given in Eq. (15) approaches zero. Therefore, each of the Δ_i^h terms on the RHS of Eq. (16) will progressively shrink, *i.e.* $(\Delta_{2i-1}^2 + \Delta_{2i}^2) - 2\Delta_i^1$ becomes smaller, resulting in a limited gap between U_2 and U_{HCA} (eq. (16)).

ii) The converse is true for Δ_i^h when the data is limited and the gap between U_2 and U_{HCA} will become more prominent, as eq. (17) suggests that RHS of eq. (16) is larger than $\sum_{i=1}^{C_1} 2(\sqrt{2} - 1)\Delta_i^1$. Moreover, as per eq. (16) and (17), the more imbalanced the data (the larger r_i), the larger the difference between U_2 and U_{HCA}.

4. Experiment and Discussion

4.1. Implementation Details

We conduct experiments on three imbalanced regression tasks: IMDB-WIKI-DIR [43] for age estimation, SHTech [45] for crowd counting, and NYUDv2-DIR [43] for depth estimation. For the age and depth datasets, we follow the same ResNet50 [13] backbone for feature extraction and training setting as [43]. For SHTech, we use VGG16 [30] backbone and the same training setting as [40]. The finest class numbers C_H are 121 for ages and 100 for depth and counting. Following [36], we choose the mean values of samples fallen in each class interval. For age estimation, we adopt linear intervals with length 1, since ages increase with step 1; while for counting or depth estimation tasks, we choose log-spaced intervals as per [21, 40] for fair comparison. Since $C_H \leq 2^7$ for all datasets, we set H as 7 for all datasets. For G_h $(h = 1, ..., H)$, we adopt one linear layer, which maps features $f \in R^d$ to outputs $\hat{p}^h \in R^{C_h}$. For G_T, a linear layer is also feasible, while a non-linear mapping is more adequate to distill the hierarchical information. Specifically, we adopt two fully connected layers with hidden dimensions $\frac{d}{4}$ and softplus activation. Detailed experiments of G_T can be found in the supplementary.

We first train the hierarchical classifiers with the summed cross-entropy loss in Eq. (6). We can then apply learning-free HCA results, HCA-add and HCA-mul with Eq. (7) and Eq. (8), respectively. For range-preserving HCA (HCA-d), classifiers $1 \sim H$ and feature extraction network F are fixed. Only classifier T is trained with L_{hd} in Eq. (12) for additional 20% epochs of stage 1 until convergence.

4.2. Ablation Studies

We first do ablation studies to verify some factors of HCA, including hierarchical class settings and two variants of HCA. IMDB-WIKI-DIR [43] and SHTech Part A (SHA) [45] datasets are chosen for ablation studies. Mean absolute error (MAE) and its balanced version bMAE [26] are adopted as evaluation metrics for SHA and IMDB-WIKI-DIR, respectively. Lower MAE and bMAE denote better performance.

i) **Hierarchical Class Settings** Combining extra classifiers could improve a single vanilla classifier, but could we just duplicate the vanilla classifier at the finest level rather than setting hierarchical classifiers? Besides, how about splitting hierarchical classifiers that equalize the interval length of each class rather than equalize sample numbers? We compare hierarchical class settings in Table 1. Compared with a single classifier, assembling duplicated classifiers can be helpful (overall bMAE from 13.50 to 13.42), but the improvement is limited compared to that of HCA. Moreover, it is more beneficial to split hierarchical classes by equaling the sample number within each class rather than equaling the length of class intervals.

Configuration	IMDB-WIKI-DIR				SHA
	All	Many	Med.	Few	
Single CLS	13.58	7.13	13.95	33.21	58.2
Same CLSs	13.42	7.10	14.38	32.22	57.9
E-Num HCA-d	**12.70**	**7.00**	13.18	29.94	**53.7**
E-Len HCA-d	12.77	7.23	**12.92**	**29.77**	56.2

Table 1. Comparison of various (hierarchical) class settings. "Same CLSs" means H classifiers adopts the same class splitting as the H-th classifier. "E-Num" means equaling the number of samples within each class during hierarchical class splitting, while "E-Len" will equal the length of each class interval.

ii) **Comparing two variants of HCA** Learning-free HCA and range-preserving HCA (HCA-d) are compared in Table 2. It can be observed: all variants of HCAs are clearly better than a single classifier or ensemble same classifiers in all shots; HCA-d is better than HCA-add and HCA-mul, suggesting that learning-free HCA cannot fully explore the hierarchical information in $\hat{p}_h$ and an explicit hierarchical distillation learning is more beneficial.

4.3. Analysis of HCA

i) **Quantization error of coarse classifiers:** Fig. 4 (a) compares the classifiers individually versus their quantization error. The coarse classifiers $(1 - 3)$ perform worse than the vanilla H-th classifier due to the quantization error of representing the entire interval with one value.

ii) **Coarse classifiers provide better range estimation; fine classifiers mitigate quantization errors**: In Fig. 4 (c), a coarse h-th classifier and the finest H-th classifier are combined in a coarse-to-fine manner. Specifically, we first get a coarse range prediction from the h-th classifier and then select a finer class within this coarse range according to $\hat{p}^H$ predicted by the H-th classifier. It can be observed that merging coarse predictions will significantly decrease the error of the H-th classifier, suggesting coarse classifiers provide better range estimation than the finest H-th classifier. Meanwhile, selecting a finer class within the coarse range will decrease the bMAE of coarse classifiers $(1 \sim 3)$, implying that combining fine classifiers can mitigate quantization error in coarse classifiers.

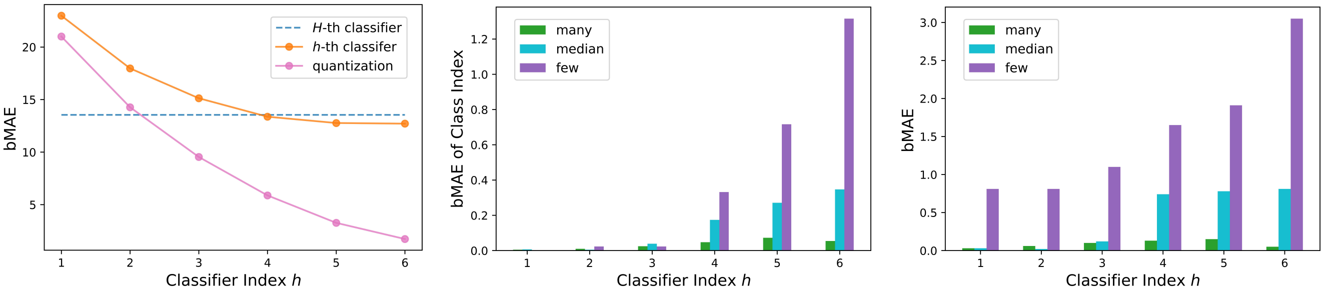
Combine	IMDB-WIKI-DIR				SHA
	All	Many	Med.	Few	
Single CLS	13.58	7.13	13.95	33.21	58.2
Same CLSs	13.42	7.10	14.38	32.22	57.9
Average	14.85	7.18	17.83	36.24	106.1
HCA-add	12.86	**6.98**	**13.15**	30.80	55.9
HCA-mul	12.89	7.00	13.36	30.74	54.7
HCA-d	**12.70**	7.00	13.18	**29.94**	**53.7**

Table 2. Comparison of two hierarchical adjustment approaches.

iii) **Range-preserving distillation is key for successful HCA**: Sec. 3.3 showed that the summation alignment of Eq. (10) is not range preserving. Table 3 experimentally compare summation and range-preserving alignment (13); using sum alignment harms HCA in all shots, while the range-preserving alignment benefits vanilla classification. Fig. 5 shows the percentage of inconsistent samples for each classifier head when using Eq. (10). The inconsistency increases with the number of classes; this is directly explained by the decreased maximum value of prediction $\hat{p}^i$ in finer classifiers. Ideally, if the maximum value of $\hat{p}^h$ is 1, then the sum of $\hat{p}^h$ will not change the range predicted by $\hat{p}^h$; however, the maximum value of $\hat{p}^h$ will be much less than 1 in regression by finer classifiers. As such, the sum operation in Eq. (10) cannot ensure consistent ranges across the hierarchy, as shown in Fig. 2.

To justify the influence of second-stage training, we add a "CLS+GT sup" baseline (see Table 3), which uses the ground-truth labels to train the classifier T. Yet this baseline does not improve over the vanilla classification, indicating that hierarchical distillation rather than extra training stages is helpful for imbalanced regression.

iv) **Influence of imbalanced vs insufficient data?** In imbalanced regression tasks, like age estimation and counting, the "Few" range is also a low-shot (≤ 20 samples per class). HCA is helpful in this imbalanced and also insufficient setting. However, is it still effective if the dataset is imbalanced but has sufficient samples in the "Few" range? Moreover, is HCA applicable to balanced regression? To verify the influence of imbalanced and insufficient data samples, we resample the IMDB-WIKI-DIR dataset to create these sce-



(a) Quantization and Prediction Error

(b) Individual Classifier Error

(c) Adjusted Classifier Error

Figure 4. Analysis of hierarchical classifiers on IMDB-WIK-DIR dataset [43]. (a) Comparison between quantization error and bMAE of the h-th classifier; for bMAE, lower is better. (b) Decrements of bMAE of the class index in each hierarchical level of classes. We report the decrements from the value of the vanilla H-th classifier. (c) Decrements of bMAE when adjusting the H-th classifier with a coarse h-th classifier. Specifically, a coarse h-th classifier provides the range, and then the finer class is selected in this range according to the prediction of the H-th classifier. We have subtracted the value of the vanilla H-th classifier.

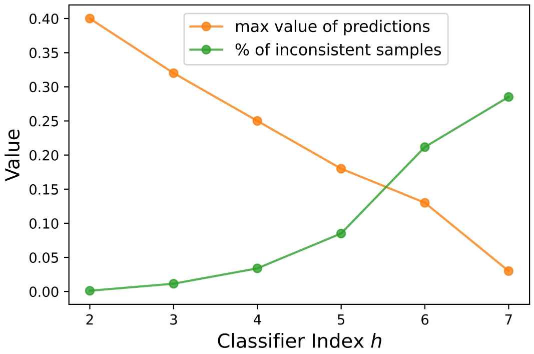


Figure 5. Percentage of inconsistent samples for each hierarchical classifier when downsampled by eq. (10). The maximum value of hierarchical predictions $\hat{p}_h$ is also visualized.

Combine	IMDB-WIKI-DIR				SHA
	All	Many	Med.	Few	
CLS	13.58	7.13	13.95	33.21	58.2
CLS+GT sup	13.64	7.20	14.94	32.54	57.0
HCA sum (10)	27.08	15.14	38.57	55.11	150.3
HCA max (13)	**12.70**	**7.00**	**13.18**	**29.94**	**53.7**

Table 3. Comparison of the summation and ranging preserving alignments of hierarchical predictions. "CLS+GT sup" denotes using the ground-truth labels to supervise the classifier T.

narios. We first generate balanced subsets with M samples per age and ages ranging from 20 to 49. M can be 1000, 100 and 10 to cover the sufficient to insufficient data scenarios. Then for imbalanced subsets, we take 20 $\sim$ 34 ages as many and 35 $\sim$ 49 ages as "Few", while keeping the ratios between Many to Few as 19. To make a fair comparison, we keep the total sample number the same among balanced and imbalanced subsets, thus having 1900 : 100, 190 : 100 and 19 : 1 imbalanced subsets covering sufficient and insufficient cases. Table 4 presents quantitative results. We can observe that: i) HCA does not show significant improvement when the training set is balanced or imbalanced but with sufficient samples (1900 : 100), in accordance with Sec. 3.5 "Remark i)"; ii) HCA outperforms vanilla classification or regression by a clear margin when the training set is insufficient, and the difference is more prominent when data imbalance is also encountered, in accordance with Sec. 3.5 "Remark ii)".

4.4. Comparison with SOTA on Regression Tasks

SHTech Dataset SHTech [45] is a crowd-counting dataset, which presents severe imbalanced distribution [21, 40, 41]. For the two subsets, Part A features crowded scenes captured in arbitrary camera views, while Part B has relatively sparse scenes captured by surveillance cameras. We follow the same network setting as [40], where 100 logarithm classes are adopted for C_H. Mean absolute error (MAE) and rooted mean square error are adopted as evaluation metrics; for both, lower errors are better. Quantitative results are presented in Table 6. It can be observed that Hierarchical classification shows the best performance and improves plain classification by a large margin.

IMDB-WIKI-DIR Dataset IMDB-WIKI-DIR [43] is a large age estimation dataset; it is an imbalanced subset sampled from IMDB-WIKI [28]. There are $192k$ / $11k$ / $11k$ training / validation / testing samples.

We choose three baselines of classification, they are: i) vanilla classification, which is H-th classifier of HCA; ii) classification with label distribution smoothing (LDS) [43], which re-weight samples with inverse class frequency; iii) classification with label distribution smoothing (LDS) and ranksim [12] regularization. ranksim [12] regularizes feature space to have the same ordering as label space. Their HCA counterparts are also included.

Table 5 presents the quantitative results. From Table 5, we can observe that: i) HCA shows clear im-

Configuration	Balanced Subsets			Imbalanced Subsets		
	1000:1000	100:100	10:10	1900:100	190:10	19:1
Regression	6.00±0.10	7.50±0.04	7.56±0.07	6.78±0.04	7.68±0.05	7.74±0.12
CLS	6.09±0.03	7.63±0.05	7.61±0.07	6.78±0.03	7.74±0.12	7.90±0.07
HCA-d	6.06±0.04	7.53±0.03	7.53±0.03	6.72±0.04	7.54±0.03	7.54±0.05

Table 4. Comparison on subsampled balanced and imbalanced subsets of IMDB-WIKI-DIR. Each method is repeated for 5 times.

Methods	IMDB-WIKI-DIR bMAE↓				NYUDv2-DIR					
	All	Many	Med.	Few	MAE↓	RMSE↓	AbsRel↓	δ_1 ↑	δ_2 ↑	δ_3 ↑
Regression [43]	13.92	7.32	15.93	32.78	1.004	1.486	**0.179**	0.678	0.908	0.975
Regression+LDS [43]	13.37	7.55	13.96	30.92	0.968	1.387	0.188	0.672	0.907	**0.976**
Regression+LDS+ranksim [12]	12.83	7.00	13.28	30.51	0.931	1.389	0.183	0.699	0.905	0.969
Balanced MSE [26]	12.66	7.65	12.68	28.14	0.922	**1.279**	0.219	0.695	0.878	0.947
CLS	13.58	7.13	13.95	33.21	1.011	1.512	0.184	0.678	0.906	0.958
HCA-d	12.70	7.00	13.18	29.94	0.987	1.475	0.181	0.689	0.915	0.961
CLS+LDS	12.85	7.31	13.40	29.54	0.924	1.383	0.181	0.711	0.909	0.965
HCA-d+LDS	12.42	7.28	12.47	28.24	0.911	1.367	**0.179**	0.714	0.911	0.966
CLS+LDS+ranksim	12.33	**6.70**	13.16	29.10	0.904	1.335	0.182	**0.715**	0.916	0.972
HCA-d+LDS+ranksim	**11.92**	6.88	**11.67**	**27.72**	**0.895**	1.321	0.180	**0.715**	**0.919**	0.972

Table 5. Comparison on IMDB-WIKI-DIR and NYUDv2-DIR Dataset. Detailed results can be found in the supplementary.

	SHA		SHB	
	MAE↓	RMSE↓	MAE↓	RMSE↓
CSRNet [20]	68.2	115.0	10.6	16.0
BL [23]	62.8	101.8	7.7	12.7
MNA [33]	61.9	99.6	7.4	11.3
OT [35]	59.7	95.7	7.4	11.8
GL [34]	61.3	95.4	7.3	11.7
Regression [40]	65.4	103.3	10.7	19.5
DC-regression [40]	60.7	101.0	7.1	**11.0**
CLS	58.2	96.7	7.0	11.8
HCA-d	**53.7**	**87.8**	**6.8**	11.8

Table 6. Comparison on SHTech dataset [45].

provement in bMAE over naive classification baselines. Specifically, HCA-d can improve all the shots for "CLS" and "CLS+LDS" baselines, while for strong baseline "CLS+LDS+ranksim", since the baseline results are already saturated for the many-shot, there is still a slight trade-off between many and few-shot (many-shot bMAE increases from 6.70 to 6.88). ii) HCA outperforms its regression baselines and other regression approaches. Noted that Balanced MSE [26] is a logit adjustment version for regression, it improves the few/medium-shot performances via significantly harming the many-shot (bMAE from 7.32 to 7.56), while for HCA-d, many-shot performance is roughly maintained or improved.

NYUDv2-DIR Dataset NYUDv2-DIR [43] is an imbalanced version sampled from the NYU Depth Dataset V2 [29]. The depth values range from 0 to 10 meters, which are divided into 100 logarithm classes for C_H. Mean absolute error (MAE), rooted mean square error (RMSE), relative absolute error (RelAbs), δ_1, δ_2 and δ_1 are adopted as evaluation metrics. Noted that all classes in NYUDv2-DIR have more than 10^7 samples, which should be all categorized as many-shot classes according to the criteria in IMDB-WIKI-DIR [43] ($>$ 100 samples). We report the overall results in Table 5 and detailed results can be found in the supplementary. We can observe that HCA shows improvements to its classification baselines and it is also comparable to or better than other regression methods. Noted that the improvement of HCA to CLS in NYUDv2-DIR is small. It is because NYUDv2-DIR is imbalanced but with sufficient samples per class, thus HCA does not improve much. This result is also in accordance with the theoretical analysis in Sec. 3.5 "Remark i)" and experiments in Table 4.

5. Conclusion

This paper proposes a hierarchical classification adjustment (HCA) for imbalanced regression. HCA leverages hierarchical class predictions to adjust the vanilla classifiers and improves the regression performance in the whole target space without introducing extra quantization errors. On imbalanced regression tasks including age estimation, crowd counting and depth estimation, HCA shows superior results to regression or vanilla classification approaches. HCA is extremely helpful in imbalanced and insufficient scenarios; while it is also helpful in balanced and sufficient scenarios.

Acknowledgment. This research is supported by the Ministry of Education, Singapore, under its MOE Academic Research Fund Tier 2 (STEM RIE2025 MOE-T2EP20220-0015).

References

[1] Peter L Bartlett, Dylan J Foster, and Matus J Telgarsky. Spectrally-normalized margin bounds for neural networks. *NeurIPS*, 30, 2017.

[2] Luca Bertinetto, Romain Mueller, Konstantinos Tertikas, Sina Samangooei, and Nicholas A Lord. Making better mistakes: Leveraging class hierarchies with deep networks. In *CVPR*, pages 12506–12515, 2020.

[3] Leo Breiman. Bagging predictors. *Machine learning*, 24: 123–140, 1996.

[4] Jonathon Byrd and Zachary Lipton. What is the effect of importance weighting in deep learning? In *ICML*, pages 872–881. PMLR, 2019.

[5] Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. *NeurIPS*, 32, 2019.

[6] Yuanzhouhan Cao, Zifeng Wu, and Chunhua Shen. Estimating depth from monocular images as classification using deep fully convolutional residual networks. *IEEE TCSVT*, 28 (11):3174–3182, 2017.

[7] Dongliang Chang, Kaiyue Pang, Yixiao Zheng, Zhanyu Ma, Yi-Zhe Song, and Jun Guo. Your" flamingo" is my" bird": fine-grained, or not. In *CVPR*, pages 11476–11485, 2021.

[8] Raul Diaz and Amit Marathe. Soft labels for ordinal regression. In *CVPR*, pages 4738–4747, 2019.

[9] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep ordinal regression network for monocular depth estimation. In *CVPR*, pages 2002–2011, 2018.

[10] Ashima Garg, Depanshu Sani, and Saket Anand. Learning hierarchy aware features for reducing mistake severity. In *ECCV*, pages 252–267. Springer, 2022.

[11] Noah Golowich, Alexander Rakhlin, and Ohad Shamir. Size-independent sample complexity of neural networks. In *COLT*, pages 297–299. PMLR, 2018.

[12] Yu Gong, Greg Mori, and Fred Tung. RankSim: Ranking similarity regularization for deep imbalanced regression. In *ICML*, pages 7634–7649, 2022.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[14] Shohei Hido, Hisashi Kashima, and Yutaka Takahashi. Roughly balanced bagging for imbalanced data. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 2(5-6):412–426, 2009.

[15] Yan Hong, Jianfu Zhang, Zhongyi Sun, and Ke Yan. Safa: Sample-adaptive feature augmentation for long-tailed image classification. In *ECCV*, pages 587–603. Springer, 2022.

[16] Bingyi Kang, Yu Li, Sa Xie, Zehuan Yuan, and Jiashi Feng. Exploring balanced feature spaces for representation learning. In *ICLR*, 2021.

[17] Taghi M Khoshgoftaar, Jason Van Hulse, and Amri Napolitano. Comparing boosting and bagging techniques with noisy and imbalanced data. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 41(3): 552–568, 2010.

[18] Jun Li, Zichang Tan, Jun Wan, Zhen Lei, and Guodong Guo. Nested collaborative learning for long-tailed visual recognition. In *CVPR*, pages 6949–6958, 2022.

[19] Mengke Li, Yiu-ming Cheung, and Yang Lu. Long-tailed visual recognition via gaussian clouded logit adjustment. In *CVPR*, pages 6929–6938, 2022.

[20] Yuhong Li, Xiaofan Zhang, and Deming Chen. Csrnet: Dilated convolutional neural networks for understanding the highly congested scenes. In *CVPR*, pages 1091–1100, 2018.

[21] Liang Liu, Hao Lu, Haipeng Xiong, Ke Xian, Zhiguo Cao, and Chunhua. Shen. Counting objects by blockwise classification. *IEEE TCSVT*, 30(10):3513–3527, 2019.

[22] Xu-Ying Liu, Jianxin Wu, and Zhi-Hua Zhou. Exploratory undersampling for class-imbalance learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 39(2):539–550, 2008.

[23] Zhiheng Ma, Xing Wei, Xiaopeng Hong, and Yihong Gong. Bayesian loss for crowd count estimation with point supervision. In *ICCV*, pages 6142–6151, 2019.

[24] Aditya Krishna Menon, Sadeep Jayasumana, Ankit Singh Rawat, Himanshu Jain, Andreas Veit, and Sanjiv Kumar. Long-tail learning via logit adjustment. In *ICLR*, 2021.

[25] Seulki Park, Youngkyu Hong, Byeongho Heo, Sangdoo Yun, and Jin Young Choi. The majority can help the minority: Context-rich minority oversampling for long-tailed classification. In *CVPR*, pages 6887–6896, 2022.

[26] Jiawei Ren, Mingyuan Zhang, Cunjun Yu, and Ziwei Liu. Balanced mse for imbalanced visual regression. In *CVPR*, 2022.

[27] Rasmus Rothe, Radu Timofte, and Luc Van Gool. Dex: Deep expectation of apparent age from a single image. In *ICCVW*, pages 10–15, 2015.

[28] Rasmus Rothe, Radu Timofte, and Luc Van Gool. Deep expectation of real and apparent age from a single image without facial landmarks. *IJCV*, 126(2-4):144–157, 2018.

[29] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, pages 746–760, 2012.

[30] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[31] Saptarshi Sinha, Hiroki Ohashi, and Katsuyuki Nakamura. Class-wise difficulty-balanced loss for solving class-imbalance. In *ACCV*, 2020.

[32] Michael Steininger, Konstantin Kobs, Padraig Davidson, Anna Krause, and Andreas Hotho. Density-based weighting for imbalanced regression. *Machine Learning*, 110:2187–2211, 2021.

[33] Jia Wan and Antoni Chan. Modeling noisy annotations for crowd counting. *NeurIPS*, 33, 2020.

[34] Jia Wan, Ziquan Liu, and Antoni B Chan. A generalized loss function for crowd counting and localization. In *CVPR*, pages 1974–1983, 2021.

[35] Boyu Wang, Huidong Liu, Dimitris Samara, and Minh Hoai. Distribution matching for crowd counting. In *NeurIPS*, 2020.

[36] Changan Wang, Qingyu Song, Boshen Zhang, Yabiao Wang, Ying Tai, Xuyi Hu, Chengjie Wang, Jilin Li, Jiayi Ma, and

Yang Wu. Uniformity in heterogeneity: Diving deep into count interval partition for crowd counting. In *ICCV*, pages 3234–3242, 2021.

[37] Xudong Wang, Long Lian, Zhongqi Miao, Ziwei Liu, and Stella Yu. Long-tailed recognition by routing diverse distribution-aware experts. In *ICLR*, 2021.

[38] Yu-Xiong Wang, Deva Ramanan, and Martial Hebert. Learning to model the tail. In *NeurIPS*. Curran Associates, Inc., 2017.

[39] Hui Wu, Michele Merler, Rosario Uceda-Sosa, and John R Smith. Learning to make better mistakes: Semantics-aware visual food recognition. In *ACM MM*, pages 172–176, 2016.

[40] Haipeng Xiong and Angela Yao. Discrete-constrained regression for local counting models. In *ECCV*, pages 621–636. Springer, 2022.

[41] Haipeng Xiong, Hao Lu, Chengxin Liu, Liang Liu, Zhiguo Cao, and Chunhua Shen. From open set to closed set: Counting objects by spatial divide-and-conquer. In *ICCV*, pages 8362–8371, 2019.

[42] Yue Xu, Yong-Lu Li, Jiefeng Li, and Cewu Lu. Constructing balance from imbalance for long-tailed image recognition. In *ECCV*, pages 38–56. Springer, 2022.

[43] Yuzhe Yang, Kaiwen Zha, Ying-Cong Chen, Hao Wang, and Dina Katabi. Delving into deep imbalanced regression. In *ICML*, 2021.

[44] Sihao Yu, Jiafeng Guo, Ruqing Zhang, Yixing Fan, Zizhen Wang, and Xueqi Cheng. A re-balancing strategy for class-imbalanced classification based on instance difficulty. In *CVPR*, pages 70–79, 2022.

[45] Yingying Zhang, Desen Zhou, Siqin Chen, Shenghua Gao, and Yi Ma. Single-image crowd counting via multi-column convolutional neural network. In *CVPR*, pages 589–597, 2016.

[46] Zhisheng Zhong, Jiequan Cui, Shu Liu, and Jiaya Jia. Improving calibration for long-tailed recognition. In *CVPR*, pages 16489–16498, 2021.