A Systematic Review on Long-Tailed Learning

Chongsheng Zhang, *Senior Member, IEEE*, George Almpanidis, Gaojuan Fan, Binquan Deng, Yanbo Zhang, Ji Liu, Aouaidjia Kamel, Paolo Soda, and João Gama, *Fellow, IEEE*

***Abstract*— Long-tailed data are a special type of multiclass imbalanced data with a very large amount of minority/tail classes that have a very significant combined influence. Long-tailed learning (LTL) aims to build high-performance models on datasets with long-tailed distributions that can identify all the classes with high accuracy, in particular the minority/tail classes. It is a cutting-edge research direction that has attracted a remarkable amount of research effort in the past few years. In this article, we present a comprehensive survey of the latest advances in long-tailed visual learning. We first propose a new taxonomy for LTL, which consists of eight different dimensions, including data balancing, neural architecture, feature enrichment, logits adjustment, loss function, bells and whistles, network optimization, and posthoc processing techniques. Based on our proposed taxonomy, we present a systematic review of LTL methods, discussing their commonalities and alignable differences. We also analyze the differences between imbalance learning and LTL. Finally, we discuss prospects and future directions in this field.**

***Index Terms*— Deep imbalance learning, deep learning, imbalance learning, long-tailed data, long-tailed learning (LTL).**

I. Introduction

LONG-TAILED distributed data are essentially a special type of multiclass imbalanced data with a sufficiently large number of tail (minority) classes; moreover, the combined importance of the tail classes is very significant although each tail class by itself only has a small number of samples (sales). Long-tailed learning (hereafter referred to as LTL for short) [1], [2], [3] aims to build effective models for applications/tasks having long-tailed distributed data. Its main goal is to significantly improve the recognition accuracy on the tail classes or cases while maintaining the same or similar accuracy on the head (majority) classes or cases.

In recent years, researchers have proposed many LTL methods for object recognition [4], [5], [6], [7], [8] that can substantially improve the recognition/prediction accuracy on the tail classes/cases. Studies in [4], [8], [9], and [10] have shown promising results by disentangling deep feature representation learning and classifier training. There are also a large number of loss reweighting approaches that endow different weights to samples of the head and tail classes to adjust the decision boundary under the long-tailed settings [3], [11], [12], [13], [14], [15]. In object detection, a few LTL methods [16], [17], [18] have been designed to automatically locate rare objects or cases from images or videos. In image segmentation, researchers have developed methods [19], [20], [21] that can identify and segment the rare objects in images. In addition, specific data augmentation techniques for LTL have also been devised to mitigate the data scarcity problem in the tail classes [22], [23], [24], [25], [26].

However, due to the rapid development of this field, keeping pace with recent advances in LTL is becoming increasingly difficult. As such, a comprehensive survey of existing methodologies in this field is urgent and beneficial to the community. This motivates us to conduct an in-depth survey of recent advances in long-tailed visual learning to gain insights into their principles and technical aspects in a systematic way. To this end, we first collect articles on LTL published between 2016 and 2024 in leading computer vision and machine learning conferences, including CVPR, ICCV, ECCV, NeurIPS, ICML, ICLR, AAAI, and IJCAI, as well as prominent journals such as IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), IEEE Transactions on Image Processing (TIP), IEEE Transactions on Neural Networks and Learning Systems (TNNLS), and IEEE Transactions on Multimedia (TMM). Candidate articles were identified using queries "long AND tail⋆" and "imbalanc⋆ OR unbalanc⋆" on the titles of the articles in the above venues. Each article, along with its references, was checked to assess its relevance, and eventually ended up with a final set of publications, which is expected to cover most, if not all, of the important and latest works in LTL.

Moreover, in this work, we propose a novel taxonomy based on the inherent learning process that categorizes existing long-tailed visual learning approaches into eight groups, as depicted in Fig. 1, which are data balancing, neural architecture, feature enrichment, logits adjustment, loss function, bells and whistles, network optimization, and posthoc

Received 12 November 2023; revised 3 August 2024 and 28 November 2024; accepted 2 February 2025. Date of publication 26 February 2025; date of current version 6 August 2025. This work was supported in part by Henan Provincial Center for Outstanding Overseas Scientists under Grant GZS2025004, in part by Henan Provincial Science and Technology Project under Grant 232102211021, in part by Henan Provincial High-Level Talent International Training Program under Grant GCC2025010, in part by the Ministry of Education of China (MOE) Liberal Arts and Social Sciences Foundation under Grant 23YJAZH210, in part by the Major Program of National Social Science Foundation under Grant 23&ZD309, and in part by the National Natural Science Foundation of China under Grant 62250410371.
(Corresponding author: Gaojuan Fan.)

Chongsheng Zhang, George Almpanidis, Gaojuan Fan, Binquan Deng, Yanbo Zhang, and Aouaidjia Kamel are with Henan Key Laboratory of Big Data Analysis and Processing, Henan University, Kaifeng 47500, China (e-mail: cszhang@ieee.org; almpanidis@ieee.org; fangaojuan@henu.edu.cn; bqdeng@henu.edu.cn; zhangyanbo@henu.edu.cn; kamel@henu.edu.cn).

Ji Liu was with Baidu Inc., Beijing 100085, China. He is now with Hithink RoyalFlush Information Network Company Ltd., Beijing 100044, China (e-mail: jiliuwork@gmail.com).

Paolo Soda is with the Department of Engineering, University Campus Bio-Medico di Roma, 00128 Rome, Italy (e-mail: p.soda@unicampus.it).

João Gama is with the Laboratory of Artificial Intelligence and Decision Support, University of Porto, 4099-002 Porto, Portugal (e-mail: jgama@fep.up.pt).
Digital Object Identifier 10.1109/TNNLS.2025.3539314

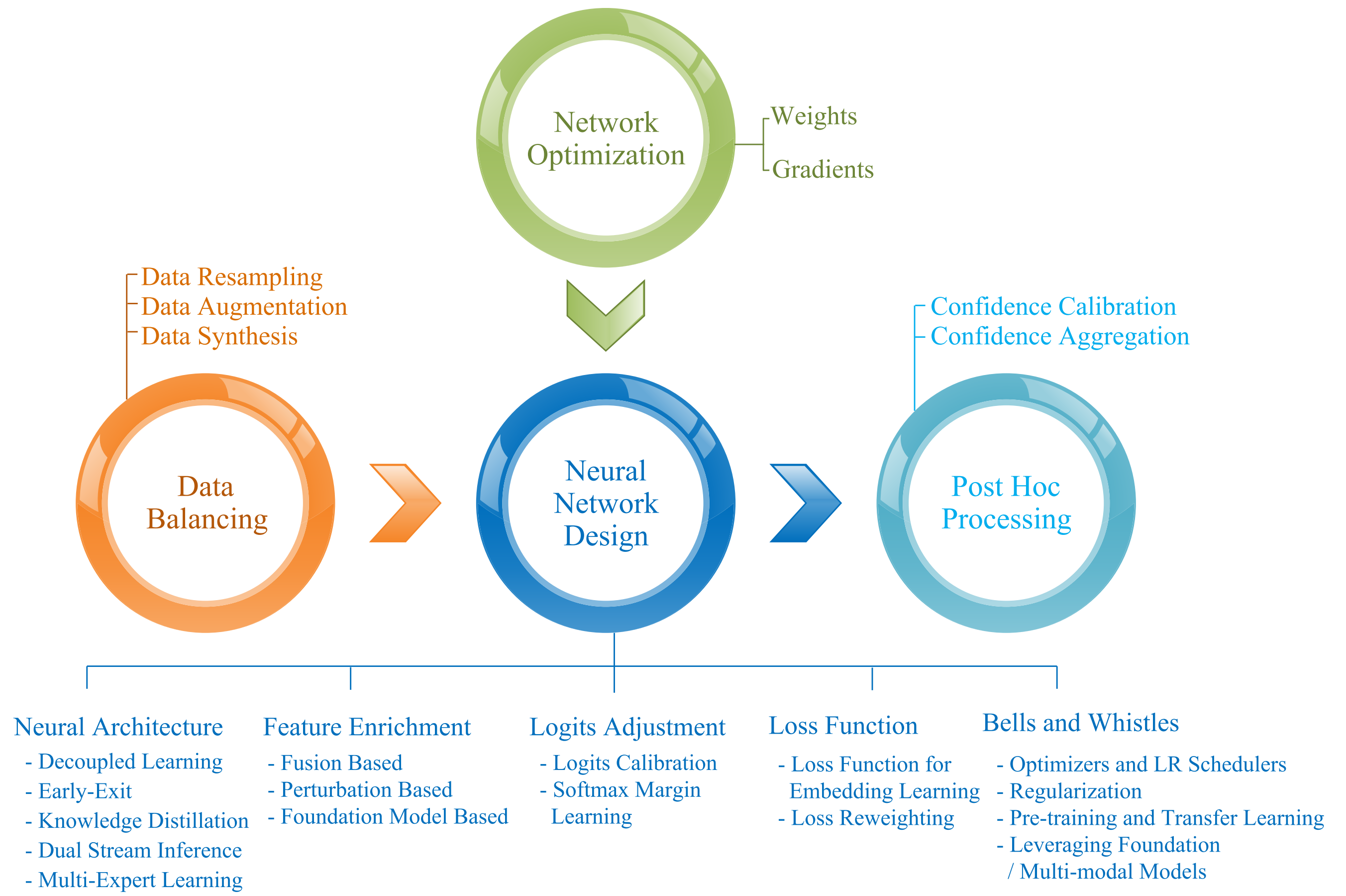

Fig. 1. Our proposed taxonomy for LTL.

processing approaches. Upon this new taxonomy, we present a comprehensive survey of state-of-the-art LTL approaches and discuss their ideas and characteristics.

Though a few recent articles [27], [28], [29] also provide literature reviews on LTL, we distinguish our survey from them by the following differences.

1) Existing reviews often employ the conventional taxonomy that categorizes existing LTL approaches into three types, which are the data resampling and augmentation category, the loss reweighting category, and the transfer learning category. However, such a taxonomy does not facilitate a comprehensive understanding of state-of-the-art LTL methods since it cannot sufficiently cover LTL's whole learning process. In this work, we propose a taxonomy that provides a more unified perspective on LTL based on its internal learning process, where we identify four major steps/modules with eight specific categories, as shown in Fig. 1. Using this new taxonomy, we provide a more up-to-date and insightful review of existing LTL methods, discussing their commonalities and alignable differences.
2) We make detailed comparisons between LTL and imbalance learning.
3) We include the latest advances in LTL and point out future directions in this field.

The main contributions of this work can be summarized as follows.

1) We propose a unified taxonomy with eight dimensions to characterize and organize existing LTL methodologies.
2) We present a comprehensive, taxonomy-guided survey of state-of-the-art long-tail visual learning approaches, focusing on recent advances and trends.
3) We compare LTL with imbalance learning and elucidate their connections and differences.
4) We summarize and analyze the results of different LTL methods in different downstream tasks, using the corresponding benchmark datasets.
5) Finally, we discuss a number of future research directions and trends that can be interesting to researchers in this field.

The remainder of this article is organized as follows. In Section II, we briefly introduce the related survey articles in LTL. In Section III, based on our proposed taxonomy, we present a comprehensive overview of state-of-the-art LTL methods. In Section IV, we provide detailed comparisons between imbalance learning and LTL. In Section V, we summarize and analyze the experimental results of different LTL methods in different downstream tasks. We discuss future directions and trends in Section VI, and conclude this article in Section VII.

II. Related Work

In the literature, there are a few related survey articles on LTL [27], [28], [29]. Zhang et al. [27] propose a taxonomy of three main categories, which are information augmentation, class rebalancing, and module improvement. Their first category consists of data augmentation and transfer learning techniques; the second category contains resampling, class-sensitive learning, and logit adjustment methods; and the third category consists of metric learning, classifier design, decoupled training, and ensemble learning approaches. Fu et al. [28] use a taxonomy for the training stage of LTL that consists of data augmentation, resampling, cost-sensitive loss, multiple experts, and transfer learning. Yang et al. [29] present a taxonomy of data processing, cost-sensitive weighting, decoupled learning, and other methods.

As can be seen from Fig. 1, our taxonomy is significantly different from the above survey articles. First of all, we divide the overall LTL process into four major modules based on the internal learning process of a neural network, including the (input) data preprocessing/balancing step, the neural network design (modeling) step, the (internal) network optimization step, and the posthoc processing step, which is more holistic and natural, and easier to understand and accept for those who are new to the LTL field.

Second, our taxonomy is more comprehensive, systematic, and in-depth.

1) Existing survey articles in LTL use the category "loss reweighting" or "cost-sensitive learning," which cannot cover other loss functions for the general embedding learning purpose, which we will address in Section III-E1.
2) Existing survey articles [28], [29] lack a systematic discussion on logit adjustment, which is a preliminary step ahead of loss function design, which we will address in Section III-D.
3) We categorize weights and gradients rebalancing approaches [13], [30], [31], [32], [33], [34], [35] into network optimization [36], which cannot be covered by taxonomies in existing survey articles.

III. LTL Methodologies

In Fig. 1 and Section I, we have introduced a new taxonomy for LTL. In this section, we will first provide the rational for building this taxonomy, and then introduce the techniques for each category.

Rationale for Our Taxonomy: When analyzing the internal learning process of an LTL model, we find that the workflow can be broken down into the following sequential steps.

1) *Data Preprocessing and Balancing:* Preprocess the input images and address the class imbalance in the dataset through data balancing techniques.
2) *Neural Network Architecture Design:* Choose an appropriate architecture tailored to the specific task and dataset.
3) *Feature Extraction and Enrichment:* The backbone of the neural network extracts features from input images. Due to limited training samples, feature representations for tail classes often require enrichment to ensure robustness and semantic richness.
4) *Logit Computation and Calibration:* The network's output layer generates logit values, which are raw, unnormalized predictions before applying activation and loss functions. Since such logits are often biased toward head classes, a calibration step is necessary to counter the imbalance.
5) *Loss Function Design:* Comprehensive loss functions play a crucial role in embedding learning, enabling the model to distinguish classes effectively within the feature space.
6) *Model Training With Backpropagation:* During training, backpropagation minimizes the loss function by computing gradients with respect to the network's weights. This process can be further optimized to improve LTL performance.
7) *Incorporating Various Strategies:* To enhance model performance, various strategies and techniques (sometimes referred to as "Bells and Whistles") can be employed in practice.
8) *Posttraining Calibration:* After model training is completed, during the prediction phase, the model's outputs can undergo further calibration to address the class imbalance and refine predictions.

Note that, the order of steps 1) and 2) can be interchanged. Based on the above rational, we propose our taxonomy for LTL. In Fig. 2, we also present representative methods for each category under our taxonomy.

A. Data Balancing

Data balancing aims to increase the volume and diversity of the training samples of the minority (tail) classes to make the training samples class-balanced. Along this line, there are three representative subcategories of approaches, which are: 1) data resampling, which balances the number of samples in different classes; 2) data augmentation, which expands the size of the training data via transformation operations; and 3) data synthesis, which generates synthetic samples using more advanced techniques such as generative adversarial networks (GANs), distribution estimation/transfer methods, and foundation models (such as GPT-4V and DALL-E [37]). Table I summarizes representative methods in each subcategory.

1) Data Resampling: In deep learning, classwise sampling and instancewise/uniform sampling are two frequently used methods for balancing the number of per-class training samples in the mini-batches. Shi et al. [38] show that classwise sampling can learn discriminative feature representations when the training samples are highly semantically related to their target labels; otherwise, uniform sampling is even better than class-balanced resampling. Huang et al. [1] first perform clustering within each class and then select samples through a quintuplet sampling scheme that enforces both intercluster and interclass margins. Dong et al. [39] propose to mine both the hard positives and hard negatives to improve the recognition performance on datasets with extremely imbalanced image attribute distributions.

2) Data Augmentation: Data augmentation is a commonly used technique in deep learning for artificially expanding the training set to improve the accuracy of the models while avoiding overfitting. It includes basic image transformations

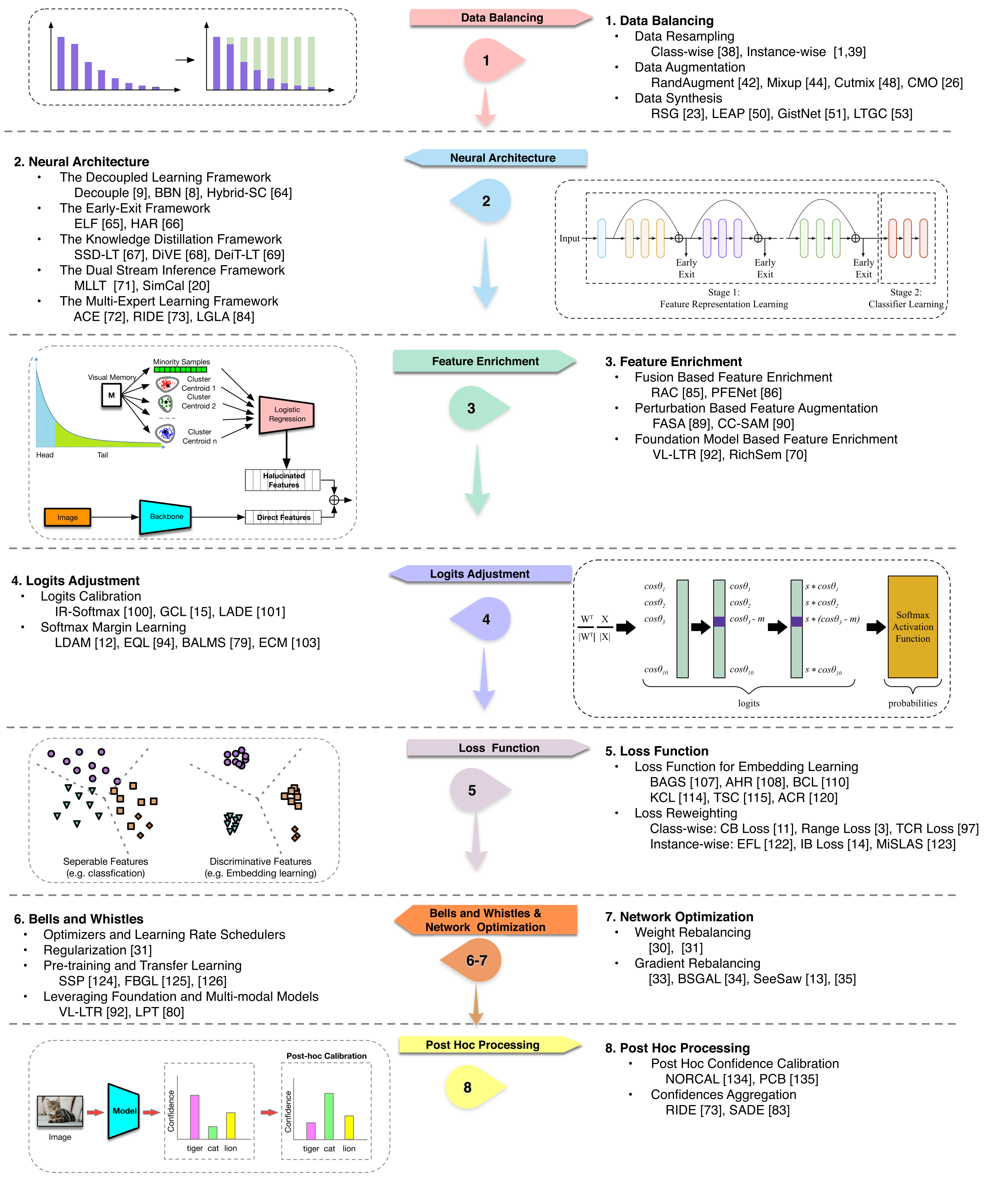

Fig. 2. New taxonomy for LTL, with representative methods in each category.

such as cropping and color space transformations, and more advanced approaches such as image mixing [40].

a) RandAugment and its variants: AutoAugment [41] and RandAugment [42] aim to find a group of transformation operations for data augmentation such that the deep model trained by a neural network can obtain better or the best accuracy on the target dataset. Unlike RandAugment, CUDA [43] finds specific augmentation operation sequences

TABLE I
SUMMARY OF DATA BALANCING APPROACHES TO LTL

Sub-Category	Representative Methods: Ref	Venue & Year	Rebalancing Strategy	Common Features of the Methods
Data Resampling	[38]	NeurIPS 2023	Class-wise sampling or Uniform sampling	Artificially balance data during model training: 1) undersampling sample with low frequency on the majority (head). 2) oversampling sample with high frequency on the minority (tail), or sample repeatedly on the minority class samples.
Data Augmentation	Rand Augment [42]	NeurIPS 2020	1) Image transformation 2) Employ memory bank to store minority class features 3) Borrow from external object-centric datasets	1) Object recognition perform data augmentation [42] based on image transformation, or interpolation [44], [48]. 2) Object detection use memory bank to store RoI feature and boxes over the past batches, or borrow and stitch external object-centric Image sets to create Scene Images, such as MosaiCOS [17].
	Mixup [44]	ICLR 2018		
	UniMix [24]	NeurIPS 2021		
	CutMix [48]	CVPR 2019		
	MetaSAug [22]	CVPR 2021		
	MosaiCOS [17]	ICCV 2021		
	CUDA [43]	ICLR 2023		
	DODA [49]	ICLR 2024		
Data Synthesis	RSG [23]	CVPR 2021	1) Cluster centroid estimation 2) Class-specific distribution variance estimation 3) Feature-level synthesis for minority samples 4) Using foundation models	Estimate the variance/covariance or the geometry of the data distribution of different classes in the feature space, then: 1) enforce the variance/covariance of a tail class to be the same as the averaged variance/covariance of the head classes. 2) transfer the geometry of the head classes to the tail classes. or use foundation models for image generation.
	M2m [25]	CVPR 2020		
	LEAP [50]	CVPR 2020		
	GistNet [51]	ICCV 2021		
	DisRobuLT [52]	ICCV 2021		
	LTGC [53]	CVPR 2024		

for different classes. They show that minority classes should be given shallow-degree sequential augmentation operations for curriculum learning.

b) Mixup and its variants: Mixup [44] is a data augmentation technique that generates synthetic images with soft labels by performing linear interpolations in both the raw input space and label space. The soft label describes the prior interclass relationship between the two input images. The main difference between Mixup and conventional transformation-based augmentation techniques lies in that Mixup alters both the raw input space and the label space when creating synthetic images. Remix [45] adapts Mixup to the long-tailed setting by assigning a soft label in favor of the minority class. Similarly, UniMix [24] extends Mixup by adopting a tail-favored Mixup factor and an inverse sampling strategy to encourage more occurrences of head–tail pairs. BEM [46] simultaneously considers classwise quantity and uncertainty in FixMatch-based [47] data/image mixing of unlabeled data with labeled data for improving the performance of semi-supervised LTL.

c) Cutmix and its variants: Cutmix [48] is another image data augmentation strategy that replaces a randomly selected region in an image with a patch from another image during data/image mixing. The soft label of the generated image can be obtained in the same way as Mixup, but the coefficients for different class labels should be in proportion to the areas of the patches in the generated images. Based on Cutmix, CMO [26] creates synthetic minority images by simply pasting the real minority samples (as the foreground) onto the majority samples (as the background) to transfer the rich contexts from majority samples to synthetic minority samples. Similarly, Shi et al. [38] propose a new augmentation method that first extracts the rich background images from the head classes using activation maps and then pastes the tail class images upon them to generate tail-class images with diverse contexts. Based on CMO, OTmix [54] considers the semantic distances between the foreground and background images/distributions in image mixing.

d) Other augmentation methods: ISDA [55] is a data augmentation method that produces diversified augmented samples by translating features along semantically meaningful directions (e.g., color and visual angles). MetaSAug [22] extends ISDA to the long-tailed setting by learning the semantic directions via meta-learning. M2m [25] iteratively modifies a majority sample until it can be confidently classified as a minority sample by a pretrained classifier. DODA [49] dynamically maintains a classwise data augmentation preference list by comparing the number of correctly predicted samples of each class with previous epochs.

e) Augmentation methods for long-tailed object detection: RIO [56] proposes to augment the minority classes in the current batch using object-centric region of interest (RoI) features and box coordinates stored in memory over the past batches. MosaicOS [17] generates pseudo-scene-centric images by stitching object-centric images (i.e., ImageNet) for long-tailed object detection. The difference between RIO and MosaicOS is that RIO uses a memory bank to reuse the object-centric RoI features while MosaicOS generates scene images by stitching several object-centric images into one image.

3) Data Synthesis: While data augmentation techniques generate synthetic images based on image transformations, data synthesis approaches utilize more complex techniques for image generation, including GANs and distribution estimation.

a) GAN-based data generation: Xie et al. [57] disentangle image features into class-agnostic features (class-specific) and class-generic features (class-independent) for one-shot image generation, then fuse the class-agnostic features from the input image and the class-generic features from the memory module, and send to the generator of a vanilla GAN for image generation. UTLO [58] is a GAN-based image generation method that devises a class-independent objective for low-resolution images and injects conditional information at higher resolution images, based on the discovery that information at the low and high resolutions tends to be class-independent and class-specific.

b) Distribution estimation-based data synthesis: When generating synthetic minority samples, RSG [23] forces the intraclass variation information of the tail classes to be the same as the head classes, while DisRobuLT [52] estimates the centroids of the tail classes using the distributional robustness theory. LEAP [50] uses the averaged variance value of all the head classes as the variance of the tail classes, whereas GistNet [51] learns the shared geometry parameters from the head classes and transfers them to the tail classes. Similarly, Wang et al. [59] propose a distribution calibration approach *LADC*, which uses the most similar head classes of each tail class to compute its mean and co-variance for inferring its distribution and then draws samples using the estimated distribution for each tail class.

c) Foundation model-based data generation: With the rise and prosperity of large pretrained models/foundation models, researchers attempt using such models to generate synthetic data. LTGC [53] leverages text-to-image (T2I) model [60] to generate diverse images for the tail classes and then refines them by measuring the cosine similarity between the image and text features extracted via CLIP [61].

B. Neural Architecture

Devising effective neural network structures/models, along with data balancing and loss function design, are the three preliminary approaches to improving the overall performance of a neural network. Along this line, related methods can be divided into subcategories of decoupled learning framework, early-exit framework, knowledge distillation framework, dual-stream inference framework, and multiexpert (ensemble) learning framework.

1) Decoupled Learning Framework: Decouple [9], as shown in Fig. 3, is a seminal work that reveals representation learning and classifier training should be carried out separately (i.e., decoupled) for improving long-tailed recognition. Upon it, Iscen et al. [62] propose to train several teacher models in the first stage to obtain more complementary features, which can be distilled into a student model. Zhang et al. [63] simultaneously use class-balanced sampling and conventional random sampling to construct mini-batches for the feature extraction step. DisAlign [10] conducts confidence calibration to adjust the decision boundary in the second stage.

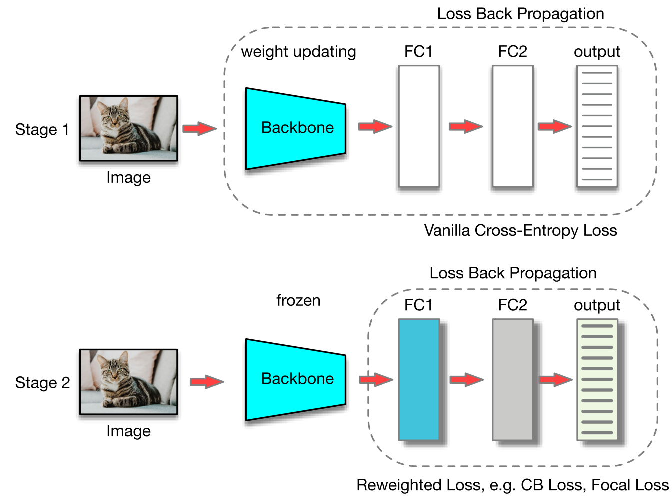

Fig. 3. Neural architecture for improving LTL performance.

The above methods have a clear boundary between the two decoupled learning stages, and thus, BBN [8] proposes to use an annealing factor to gradually transit between the two stages. Likewise, DB [4] devises a soft gate to switch between a conventional feature learning module and a tail-class favored feature learning module. Like BBN, Hybrid-SC [64] smoothly transits between the two branches via a cumulative learning adaptor, while using supervised contrastive learning in the first stage to learn better features.

2) Early-Exit Framework: The early exiting framework (ELF) [65] early-exits easy examples in the shallow layers of a neural network to focus on the hard examples in each class in the deeper layers. Similarly, HAR [66] augments a backbone network with auxiliary classifier branches, and samples that are confidently classified by one of the branches will early exit.

3) Knowledge Distillation Framework: SSD-LT [67] trains the self-distillation and classification heads for knowledge distillation for long-tailed recognition, under the decoupled learning framework. DiVE [68] uses knowledge distillation to transfer knowledge from the teacher models to a student model. DeiT-LT [69] distills knowledge from a convolutional neural network (CNN) teacher to a vision Transformer (ViT) backbone/student using out-of-distribution (OOD) images generated via Mixup and Cutmix. For long-tailed object detection, RichSem [70] adds a new branch to the detection framework for distilling semantics from CLIP, which can extract rich semantics for the bounding box patches.

4) Dual-Stream Inference Framework: MLLT [71] designs two branches for learning from the long-tailed and artificially balanced distributions, respectively, and adds a cross-branch consistency loss based on logit difference to enforce the two branches learn collaboratively. For long-tailed instance segmentation, SimCal [20] trains a vanilla classifier and a calibrated classifier on the class-balanced object proposals and original proposals obtained in the region proposal generation stage of Mask R-CNN, respectively, and then uses both of them in the object classification subtask to improve the performance on tail classes.

5) Multiexpert Learning Framework: Multiexpert/ensemble learning methods aim to utilize multiple models to achieve more accurate predictions. Along this line, ACE [72] trains cascade classifiers by iteratively eliminating samples of the current head class such that the medium-shot or tail classes will have chances to dominate a classifier. RIDE [73] trains multiple recognition models/experts on randomly sampled subsets and uses their logits mean to make predictions. CBD [62] trains multiple classification models on randomly sampled subsets and then distills the knowledge from these models into a single student model. LFME [74] differs from CBD in that its teacher models are trained over nonoverlapped partitions of classes and samples. BalPoE [75] is a multiexpert learning framework composed of logit-adjusted experts. SHIKE [76] aggregates the features in different experts with different layers of shallow features, due to their discovery that shallow features of a network can better represent the tail classes, and then averages the predictions from different experts as the final prediction. NCL [77] trains multiple experts on all the categories and the hard categories, respectively, and then adopts knowledge distillation to learn the multiple experts collaboratively. MDCS [78] uses the balanced Softmax loss [79] to control experts' focus on different categories to improve the diversity of the experts and performs self-distillation to reduce model variance. LPT [80] leverages the powerful discrimination ability of the large-scale pretrained vision models [61], [81] via shared prompt tuning to adapt the pretrained model to the target domain; it further learns group-specific prompt sets for each group of classes to enhance the learning performance.

Given the fact that the class distributions of test data are not necessarily uniform, DirMixE [82] uses the Monte Carlo method to estimate the mean and semivariance of a meta-distribution for capturing both the global and local variations to adapt to test-agnostic distributions, while SADE [83] trains three experts/models on uniform, long-tailed, and inversely long-tailed class distributions separately, then weighted aggregates them at the inference stage. LGLA [84] trains multiple models to extract more discriminative features for different subsets and uses logit adjustment strategy to enlarge the discrepancy among the models.

C. Feature Enrichment

The representation capability of deep learning-based models on the tail classes is generally limited, due to insufficient learning of features on the tail class samples. While directly obtaining more samples for enhancing the diversity of tail classes is often difficult, feature enrichment techniques aim to produce more diverse, robust, and semantically meaningful feature representations for the tail classes, using fusion-based approaches, perturbation-based approaches, large pretrained models/foundation models, etc.

1) Fusion-Based Feature Enrichment: As depicted in Fig. 4, RAC [85] uses an external memory module to store features of the tail samples and finds in the memory bank the Top-K most similar features of the same class for each input image, which are then fused with the original features.

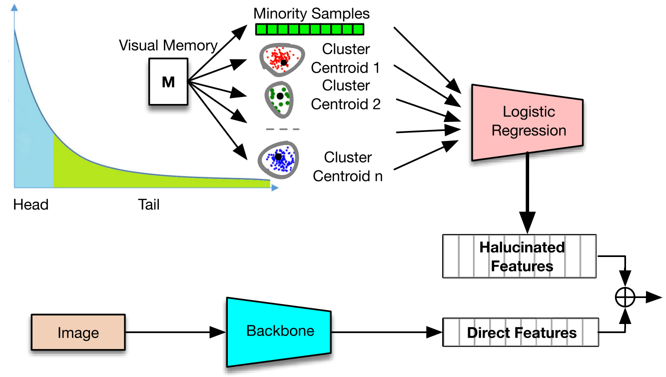

Fig. 4. Feature enrichment approaches to improving LTL performance.

Likewise, PFENet [86] enriches query features with support features to overcome the spatial inconsistency in few-shot segmentation. LOCE [18] utilizes a memory module to store features of the tail-class samples and proposes equilibrium loss to adjust the margins for the tail classes in long-tailed detection. OLTR [7] combines the features extracted by a conventional CNN pipeline and the ones stored in an additional memory module for generating features of the tail classes, which can transfer knowledge between head and tail classes. Li et al. [87] replace a portion of feature maps from tail classes with those belonging to head classes to enhance the diversity of tail-class representations. MFW [88] weakens features of the head classes by mixing them with features of other classes in the same mini-batches but without altering the corresponding labels to equalize the training progress across classes.

2) Perturbation-Based Feature Augmentation: For long-tailed instance segmentation, FASA [89] generates classwise virtual features by random perturbation over a Gaussian distribution on the real samples of each class. CC-SAM [90] adopts class-conditioned perturbations on the features in the last convolutional layer. Its idea is similar to LogitAdjust [91], but LogitAdjust performs perturbations over logit values in the output layer, whereas CC-SAM adds noises to the features in the last hidden layer.

3) Foundation Model-Based Feature Enrichment: VL-LTR [92] simultaneously utilizes the visual and linguistic representations extracted by CLIP [61] for long-tailed recognition. For long-tailed object detection, RichSem [70] adds a new branch to the detection framework for distilling knowledge from CLIP, which can extract rich visual and linguistic semantics to enhance the representations of the objects within the image.

D. Logits Adjustment

Logits adjustment is the preliminary step ahead of loss function design that aims to optimize the neural network's inherent learning process in deriving the logit values or calibrate the logit values in a posthoc manner, to obtain more optimal logit values before sending them to the loss functions. As a step in LTL's whole learning process, logits adjustment in LTL commonly needs to deal with the skewed or unrobust

TABLE II
SUMMARY OF LOGITS ADJUSTMENT METHODS AND LOSS REWEIGHTING METHODS FOR LTL

	Method & Ref.	Venue	Task	Loss Function	Notes/Remarks
Logits Adjustment	Balanced MSE [93]	CVPR 2022	Visual regression	$-\log\left(\frac{\exp(-\lVert p_i-y_i\rVert_2^2/\tau)}{\sum_{j=1,j\neq i}^{K}\exp(-\lVert p_j-y_j\rVert_2^2/\tau)}\right)$	y_i is the ground-truth value of class i in the one-hot vector, p_i is the predicted probability on class i, τ is the temperature hyper-parameter.
	LDAM Loss [12]	NeurIPS 2019	Object classification	$-\log\left(\frac{n_i^{\frac{1}{4}}\cdot\exp(z_i)}{\sum_{j=1}^{K} n_j^{\frac{1}{4}}\cdot\exp(z_j)}\right)$	It employs a data resampling method that estimates the optimal sampling rates for different classes, n_i is the number of samples in the class of the input sample.
	EQL Loss [94]	CVPR 2020	Object detection and recognition	$-\log\left(\frac{\exp(z_i)}{\sum_{j=1}^{K} w_j\cdot\exp(z_j)}\right)$	EQL Loss is short for Equalization Loss, w_j is a factor that aims to ignore or down-weight the gradients of other tail classes, while still keeping the gradients for the samples of the head classes.
	AutoBalance [95]	NeurIPS 2021	Object classification	$-\omega_i\cdot\log\left(\frac{\exp(s_i\cdot z_i+m_i)}{\sum_{j=1}^{K}\exp(s_j\cdot z_j+m_j)}\right)$	s_i, m_i are the multiplicative and additive margin parameters, ω_i is the class-wise reweighting factor for the group of classes having similar number of samples.
	LogitAdjust [91]	ICLR 2021	Object classification	$-\log\left(\frac{\exp(z_i-\tau\cdot m_i)}{\sum_{j=1}^{K}\exp(z_j-\tau\cdot m_j)}\right)$	m_i is a margin parameter for class i, τ is a hyper-parameter (temperature).
	C2AM [96]	CVPR 2022	Object detection	$-\log\left(\frac{\exp(s\cdot z_i)}{\exp(s\cdot z_i)+\sum_{j=1,j\neq i}^{K}\exp(s\cdot z_j+m_{ij})}\right)$	$z_i=\cos\theta_i$, $m_{ij}=\left\lvert\frac{\alpha}{\pi}\log\frac{\lVert w_i\rVert_2}{\lVert w_j\rVert_2}\right\rvert$, $\lVert w_i\rVert_2$ is the L2 norm of the corresponding weight vector for class i, α and s are hyper-parameters.
	GCL [15]	CVPR 2022	Object recognition	$-\log\left(\frac{\exp(s\cdot(z_i-m_i\cdot\epsilon_i))}{\sum_{j=1}^{K}\exp(s\cdot(z_j-m_j\cdot\epsilon_j))}\right)$	$z_i=\cos\theta_i$, which is the cosine of the angle between the logit vector and the corresponding weight vector for the class of the input sample. $m_i=\log\frac{n_{max}}{n_i}$. $\epsilon_i=\mathcal{N}(0,1)$, which is random decimal. s is a hyper-parameter.
Loss Reweighting	CB Loss [11]	CVPR 2019	Object classification	$-\frac{1-\beta}{1-\beta^{n_i}}\cdot\log\left(\frac{\exp(z_i)}{\sum_{j=1}^{K}\exp(z_j)}\right)$	$\beta=0.999$ or other values, n_i is the number of samples in the class which the input sample belongs to. CB Loss inversely reweights the loss of a sample by the number of samples in that class.
	Range Loss [3]	ICCV 2017	Face classification and recognition	$\alpha\cdot\sum_{i=1}\frac{2}{\frac{1}{d_1}+\frac{1}{d_2}}+\beta\cdot\lvert P-d_3\rvert-\lambda\cdot\log\frac{\exp(z_i)}{\sum_{j=1}^{K}\exp(z_j)}$	d_1, d_2 are the two largest intra-cluster distances of the input sample's class, d_3 is the shortest distance of the centres of any two classes, P is a hyperparameter.
	TCR Loss [97]	CVPR 2020	Object classification	$-(\omega_i+\varepsilon_i)\cdot\log\left(\frac{\exp(z_i)}{\sum_{j=1}^{K}\exp(z_j)}\right)$	TCR Loss is short for the two-component hybrid reweighting method. ω_i and ε_i are obtained via CB Loss [11] and L2RW loss [98], respectively.
	Focal Loss [99]	ICCV 2017	Object detection and recognition	$-(1-p_i)^r\cdot\log p_i,\ \ p_i=\frac{\exp(z_i)}{\sum_{j=1}^{K}\exp(z_j)}$	r is the focusing parameter, e.g. $r=2$. Difficult examples will be relatively upweighted $(1-p_i)^r$, while easy examples (those with less prediction loss value) will be relatively down-weighted.
	IB Loss [14]	ICCV 2021	Object classification (Instance-wise)	$-\frac{1}{\sum_{j=1}^{K}\lvert p_j-y_j\rvert\cdot\sum_{l=1}^{L}\lvert h_l\rvert}\cdot\log\left(\frac{\exp(z_i)}{\sum_{j=1}^{K}\exp(z_j)}\right)$	IB Loss is short for Influence-Balanced Loss, it contains a normal training step and a fine-tuning step. p_j is the predicted probability on the class j in the output layer, h is the logit vector.

logit values. In Table II, we summarize the formulas of representative methods in this category.

1) Logits Calibration:

a) Inherent logits calibration: When deriving logits from the last hidden layer to the output layer, IR-Softmax [100] sets the weights for each class as their corresponding class centers in the feature space to avoid the shift between the weights and their centers. *Balanced mse* [93] straightforwardly uses the mean square error on each class to replace the corresponding logit element in imbalance regression.

b) Posthoc logits calibration: GCL [15] perturbs the normalized logit with an additive margin that has a random disturbing parameter, while BLV [101] adds categorywise variations to the logits for long-tailed semantic segmentation, which are inversely proportional to the per-class sample size. CDMAD [102] finds that a classifier trained on long-tailed data produces skewed class probabilities on a solid color image, thus adjusting the classifier's predicted logits by subtracting its predicted logits on a white image.

2) Softmax Margin Learning: Softmax margin learning alters the logit values before passing them to the activation function. Therefore, it can also be considered as "posthoc logits calibration." LDAM [12] subtracts the logit element for the ground-truth class with classwise margin parameter, which is reversely proportional to the quadratic root of the number of samples in the class. Equalization loss (EQL) [94] removes the logit values of other tail classes in the denominator in the Softmax activation function. AutoBalance [95], BALMS [79], and LogitAdjust [91] modify the cross entropy loss by adding multiplicative and additive margin parameters. Similarly, C2AM [96] adds a margin for every nonground-truth class, based on the ratio between the weight norms for the ground-truth class and every other class. For long-tailed object detection, Cho and Krähenbühl [103] propose the effective class-margin (ECM) loss for margin-based classification.

E. Loss Function

Loss function design is one of the major approaches to embedding learning. Loss functions for LTL also need to balance the class distributions in feature space or expand the embedding space distribution of the tail classes.

1) Loss Function for Embedding Learning:

a) Supervised LTL: DLSA [104] adjusts the label space of the samples via deep clustering. The well-clustered samples are sent to a dedicated cluster-aided classifier, while the remaining samples are considered as outliers and sent to the next adjustment module to decrease the training difficulty. SuperDisco [105] discovers the super-class in a hierarchy of semantic abstraction and then refines the original features of a sample using the corresponding super-class features, based on their observation that the distribution of the super-classes in imbalanced data is more balanced than the original data. Similarly, for long-tailed regression, HCA [106] proposes a coarse-to-fine hierarchical classification strategy, which first uses coarse classifiers to provide the range and next selects the corresponding finer classifier in the range. BAGS [107] groups the classes by the number of instances and then trains a separate Softmax classifier for each group to reduce the suppression by the head classes. LST [21] also adopts the "divide-and-conquer" strategy for long-tailed segmentation, but devises a knowledge distillation loss, which measures the logit difference of the same class in neighboring parts to preserve the learned knowledge. AHR [108] adds a hierarchical representation (AHR) loss to the object detection loss for more discriminative embedding learning, which calculates both the coarse-grained classification loss for distinguishing each cluster clearly and the fine-grained classification loss within each cluster by adopting adaptive margins. GOL [109] utilizes the Gumbel activation function to replace Softmax for long-tailed object detection and segmentation.

SBCL [110] devises a triplet loss like contrastive loss for long-tailed recognition, which enforces a sample to be closer to samples of the same subclass than other samples in the same class, while being closer to samples of a different subclass in the loss function. GLMC [111], respectively, uses CMO [26] and Mixup [44] to create two synthetic images, and then uses both the cosine distances between the features of the two images and the classification loss on the two images to optimize the network. Wang et al. [112] quantify the dominance severity of the head classes in the representation space and formulate the interclass discrepancy and class-feature consistency losses to optimize the network.

Contrastive learning [113] has proven to be able to improve representation learning performance. Kang et al. [114] propose the K-positive contrastive loss (KCL) for LTL, which draws k instances from the same class to form the positive sample set for contrastive learning. Based upon KCL, TSC [115] adds a new loss term between the samples and the predefined optimal class centers to move the samples closer to the centers of their class and away from centers of other classes. The PaCo [116] loss introduces learnable classwise centers into the supervised contrastive learning process to make the probability that two samples being a true positive pair become more balanced across different classes. BCL [117] considers cluster compactness in the contrastive learning process.

b) Semisupervised LTL: CReST [118] iteratively samples pseudo-labeled data from the unlabeled set to expand the labeled set, and then retrains the semi-supervised learning model, due to their observation that semi-supervised learning methods obtain low recall but high precision performance on the minority samples. SimPro [119] is a simple probabilistic framework for long-tailed semi-supervised learning, which separates the estimation process into conditional and marginal distributions using the expectation–maximization algorithm. ACR [120] uses a dual-branch network to adaptively handle various class distributions, in which it derives the scaling parameter using the bidirectional KL divergence between the estimated class distribution and the anchor distributions.

2) Loss Reweighting: After obtaining the loss on each class or sample, loss reweighting approaches aim at adjusting the importance of their loss values through the weighting factors. Existing approaches along this line can be divided into classwise reweighting and instancewise reweighting methods. In Table II, we report the formulas of representative methods.

a) Classwise loss reweighting: Classwise loss reweighting methods assign the same weighting factor to instances of the same class or group. CB Loss [11] is a class-balanced loss function for long-tailed distributions that reweights the samples by the inverse class frequency, i.e., the weight/factor of a class is inversely proportional to the number of samples in the class. AREA [121] proposes to reweight the loss of each class based on the inverse Pearson correlation of the samples in the same class. Range Loss [3] is also a classwise reweighting method that consists of an intraclass loss and an interclass loss: the former is defined as the harmonic mean of the two largest intraclass distances, while the latter is the shortest distance between any two class centers. The learning to reweight (L2RW) method [98] is a meta-learning algorithm that assigns weights to a batch of samples based on their similarities in gradient directions. TCR [97] is a two-component reweighting loss function that simultaneously computes the classwise weights using the CB loss [11] and the conditional distribution between the source and target with L2RW [98].

b) Instancewise loss reweighting: The above methods assign the same weight to all samples belonging to the same class. In contrast, instancewise reweighting methods assign different weights to different samples based on their learning difficulty. Focal Loss [99] proposes to assign relatively larger weights to samples with high prediction loss, independent of their class labels. EFL [122] modifies Focal Loss by supplementing the focusing parameter with a new term that is the inverted ratio between the accumulated positive and negative gradients of the input sample's ground-truth class. IB Loss [14] reweights each sample differently using the designed influence function to alleviate overfitting in the decision boundary. MiSLAS [123] assigns different weights to the head and tail classes to handle different degrees of over-confidence, based on their observation that models trained on long-tailed datasets are more miscalibrated and over-confident than those trained on balanced datasets.

F. Bells and Whistles

"Bells and Whistles" in deep learning refers to various techniques, training strategies, and practices that go beyond neural architectures and loss functions but are also essential or even critical for improving model performance. Common "bells and whistles" include network training strategies such as advanced optimizer, learning rate (LR) schedulers, regularization, and commonly adopted techniques in deep learning such as data augmentation, pretraining and transfer learning, knowledge distillation, and ensemble learning for improving the performance of deep learning models.

There are different types of optimization algorithms used in neural networks [36], in which stochastic gradient descent (SGD) is the major method used for optimizing the network parameters (weights and biases) in LTL. LR schedulers are methods that adjust the LR during network training, such as linear decay, step decay (such as CosineAnnealing), time-based decay, and exponential decay to optimize model performance over time. Regularization techniques in deep learning are methods used to prevent overfitting and improve the generalization ability of neural networks. Commonly used regularization techniques include L2 regularization (weight decay) [31], [32], dropout, batch normalization, and label smoothing (which replaces hard labels with smoothed distributions to reduce the model's confidence and encourage it to be more robust and generalize better) [123].

Since data augmentation and synthesis, knowledge distillation, and ensemble (multiexpert) learning methods have already been addressed in Sections III-A and III-B, we only present a few pretraining and transfer learning techniques and foundation models-based approaches.

a) Pretraining and transfer learning for LTL: SSP [124] introduces self-supervised pretraining into class-imbalanced learning to yield good network initialization, which can alleviate the label bias issue in imbalanced datasets and enhance LTL performance. FBGL [125] uses both contrastive pretraining and normalization for improving the feature extractor and classifier for long-tailed recognition, which consists of a new balanced contrastive loss to improve long-tailed pretraining and a generalized normalization method for the classifier (feature and weight vectors). Dong et al. [126] propose a simple yet effective stepwise learning framework based on deformable DETR [127] that combines fine-tuning and knowledge transfer for long-tailed object detection. In specific, after selecting representative examplars for the head and tails classes, it first pretrains a model on all the classes, next learns a head-class model via fine-tuning, and then transfers knowledge from the head-class model to the final model.

b) Leveraging large foundation models and multimodal models for LTL: Lately, there has been a popular trend in utilizing the powerful feature representation capability of foundation models [81], [128], [129], [130] and pretrained multimodal models [61], [92] to improve the LTL performance, which can also be considered as a "trick" for LTL. For instance, integration of textual and visual features enhances the model's ability to discriminate between classes with fewer examples [70], [92] or enables the model to understand and reason about the semantic relationships between concepts depicted in images and described in texts [80].

Finally, we note that these strategies and techniques are often collectively used to make LTL models more effective.

G. Network Optimization

Network optimization is the internal mechanism for updating and optimizing the huge amount of learnable parameters in a neural network. In LTL, network optimization involves techniques in updating the weights, and the gradients with respect to the loss functions and weights. Existing approaches can be divided into weights rebalancing and gradients rebalancing approaches.

We clarify that "network optimization" refers to techniques that optimize or utilize the weights and gradients of the network for internal network optimization, while "Bells and Whistles" in Section III-F refer to different external training strategies, choices, techniques, and tricks for obtaining better learning performance.

1) Weights Rebalancing: Kim and Kim [30] show an obvious correlation between sampling frequency and the norm of the weight vectors, and propose to multiply the weight vector of each class with the ratio between the number of samples in the most frequent class and the current class to adjust the decision boundary in LTL. Alshammari et al. [31] also reveal that, for long-tailed data, a normally trained deep learning model has imbalanced norms of classifier weights, with the majority classes having dominating weights. They propose to balance network weights using standard regularization techniques such as weight decay (L2 regularization) and notably improve the long-tailed recognition performance. Hasegawa and Sato [32] discuss why weight balancing [31] is effective in two-stage LTL. It reveals that weight decay helps achieve better feature extraction, while class-balanced loss enables implicit logit adjustment in the training. Curvature-balanced regularizer (CR) [131] uses the sum of the logarithm of the inverse of the maximum normalized Gaussian curvature mean of each class as a regularizer to enable the model to learn curvature-balanced manifolds.

2) Gradients Rebalancing: Zhang et al. [33] reveal that long-tailed data suffer from a gradient distortion problem, in which the overall gradient is shifted toward the head gradient; it, thus, proposes a dual-phase long-tailed recognition method, which first updates model parameters using the gradient on the head classes only, and next grows the classifier on the added tail classes. Zhu et al. [34] propose BSGAL, which is a gradient-based generated data contribution estimation method for long-tailed instance segmentation that calculates the loss and gradient of the model on a batch of data with and without augmentation to actively accept or reject the generated data. SeeSaw Loss [13] dynamically rebalances the gradients of positive and negative samples for each category with two complementary factors to reduce punishments on tail categories and increase the penalty of misclassified instances.

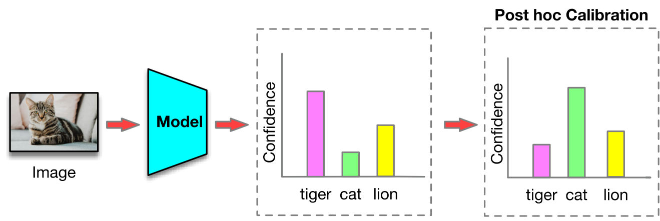

Fig. 5. Posthoc processing methods for LTL.

For long-tailed class-incremental learning, He [35] proposes to separately reweight the gradients for the new tasks and learned tasks, based on the accumulated magnitude of gradients and the volume of the lost training data of each class, respectively.

Overall, tuning the above factors is crucial for the internal optimization of the neural network to achieve high model accuracy and prevent overfitting in LTL.

H. Posthoc Processing

Posthoc processing methods modify the outputs of the trained models to tackle the imbalance problem at the inference stage (rather than in the model training process), as demonstrated in Fig. 5, which is simple but sometimes useful technique/step for improving the model's prediction performance in long-tailed distributions. Expected calibration error (ECE) [132] is often used to measure how well the predicted probabilities match the ground-truth probabilities, which first partitions the confidence scores into several equal-size bins, and then computes the difference between the accuracy and the averaged confidence in each bin.

1) Posthoc Confidence Calibration: Temperature scaling (TS) [133] uses a single scalar parameter T to rescale the logits of a trained neural network, which will then be transformed into calibrated confidence scores using the Softmax activation function. However, this approach has yet to be validated in long-tailed settings.

For confidence calibration in long-tailed object detection, NORCAL [134] multiplies the prediction (output) scores with a vector of factors that are inversely proportional to the training sample size of each class.

For confidence calibration in long-tailed object segmentation, the pairwise class balance (PCB) loss [135] postcalibrates a sample's prediction probability on each class by multiplying the corresponding pairwise misclassification bias obtained from the confusion matrix.

2) Confidence Aggregation: When a neural network has multiple heads for inference, e.g., the dual streams inference framework and the multiexpert learning framework addressed in Sections III-B4 and III-B5, these methods commonly use the logits mean [73] or the weighted aggregation of the logits [83], as well as the confidence mean to make the final prediction.

Discussion: It is worth noting that these eight categories/dimensions of methods are orthogonal and complementary. Each category of methods has its own perspective and strength. Data balancing (in particular, the data resampling and data augmentation methods), loss function, and posthoc processing techniques are relatively easy/light to implement and have proven to be very useful in practice, whereas neural architecture, feature enrichment, and bells and whistles approaches typically need lots of test and trials, i.e., with higher empirical study costs. Finally, logits adjustment and (internal) network optimization approaches require deeper understanding of the neural network mechanisms. In theory, we can apply all of the eight categories of approaches in a streamline; this way, we can extend from single-perspective, single-dimension LTL methods to multiperspective, multidimension, and more comprehensive LTL approaches to yield significant performance.

IV. Comparisons Between LTL and Imbalance Learning

When referring to long-tailed data, one may neglect the fact that long-tailed data are a special type of imbalanced data, and LTL essentially belongs to imbalance learning [136], [137], [138].

There are two major differences between long-tailed data and imbalanced data. First, in long-tailed data, there are a significantly large number of tail classes, while in "ordinary" imbalanced data, their amount may be as few as two (even only one). Second, since long-tailed data have a very long tail, the overall trend/pattern of event frequency/sample size per class should decrease exponentially from the head classes to the tail classes, but the subpattern/trend among the tail classes should be a relatively flat curve with a slowly decreasing pattern, whereas ordinary imbalanced data do not necessarily have this property.

Imbalance learning is a long-standing research direction in machine learning. For relational/tabular data, its features are its attribute values, that is, features are already available before imbalance learning; thus, techniques for feature representation are not essential in imbalance learning. But for image data, feature representation learning is a prerequisite step. Traditional computer vision methods use hand-crafted feature extraction methods, which are often model-driven and independent of data distributions. After the feature extraction step, if the resulting features are class-imbalanced, they rely on imbalance learning algorithms to solve the imbalance problem. Early approaches such as [139] and [140] fall into this category. Due to the advantages of deep learning in representation learning, researchers started exploring the use of deep learning for tacking the class-imbalanced problem. However, such deep learning methods also bring new problems, in particular, their feature representation learning becomes class-biased on long-tailed data. To address such new challenges, researchers came up with the new term "LTL" and proposed various LTL techniques.

In the following, we will mainly compare the data balancing and loss functions/cost-sensitive learning methods, as well as the multiexpert/ensemble learning methods adopted in both imbalance learning and LTL.

A. Data Balancing

1) Data Resampling: In machine learning, oversampling/upsampling the minority classes and undersampling/downsampling the majority classes are two commonly used strategies for handling class imbalance [136]. In LTL, only random sampling is utilized, either at class-level or instance-level, referred to as classwise sampling and instancewise sampling, respectively. However, in imbalance learning, more advanced undersampling and oversampling techniques have been proposed.

a) Undersampling techniques in imbalance learning: In condensed nearest neighbor rule [141], for each sample from the majority set, if it cannot be correctly classified by nearest neighbor rule, it is kept; otherwise, it is discarded. Zhu et al. [142] propose an intuitive geometric space partition-based method for imbalance classification, which iteratively uses a new hyperplane classifier to cut the current geometric data space into two partitions and then removes the partition that only contains the majority samples.

b) Oversampling techniques in imbalance learning: ROS [143] is the most straightforward oversampling technique, which balances the data by replicating the minority class samples. MWMOTE [144] first identifies the most important and hard-to-learn minority class samples, and then assigns weights to them according to their Euclidean distance to the nearest majority class samples.

2) Data Augmentation: For image data, in LTL, Mixup [44], ReMix [45], and UniMix [24] create synthetic samples with interpolated feature vectors and labels out of two original samples.

For relational/tabular data, since the features are already available before imbalance learning, it is easy and common to use linear interpolation algorithms to generate synthetic minority samples between two real minority samples of the same class. SMOTE [145] creates synthetic minority samples by randomly selecting a new point on the line segment between a pair of neighboring minority samples. Borderline SMOTE [146] extends SMOTE by emphasizing the borderline samples that have both majority and minority class points in their neighborhood and ignoring the rest of the minority samples. ADASYN [147] creates synthetic minority samples according to data density, where a comparatively larger number of synthetic samples are created (using SMOTE) in regions of low density of the minority class than in higher density regions. SMOTE-RSB [148] first uses SMOTE to generate synthetic instances for the minority classes, and then employs the rough set theory as a cleaning method to select only the synthetic minority examples that belong to the lower approximation of their class.

In general, the idea of Mixup, ReMix, and UniMix is similar to SMOTE and its variants, but the former interpolates the label space as well, while SMOTE and its variants only handle samples of the same minority classes, but without modifying the labels of the synthetic samples.

3) Data Synthesis: In LTL, representative data synthesis methods include (RSG) [23], LEAP [50] and GistNet [51], GAN [58], and foundation model-based [53] methods. In comparison, in imbalance learning, GANs are commonly utilized to generate synthetic tabular data, such as TableGAN [149], CTGAN [150], medGAN [151], and QAST [152].

TableGAN [149] was the first attempt to synthesize tables/relational databases using GANs. It adopts the deep convolutional GAN (DCGAN) architecture, but adds a new neural network module called "classifier" and a corresponding loss function to measure the discrepancy between the label of a generated record and the label predicted by the classifier for this record. CTGAN [150] models the distribution of the values with a few Gaussian distributions (modes) for each continuous attribute. For discrete attributes, CTGAN introduces a conditional generator and training-by-sampling method to ensure that each category of a discrete attribute gets a fair chance to be included during sampling. By this way, it can simultaneously generate a mix of discrete and continuous attributes. Medical GAN (medGAN) [151] uses auto-encoders to learn the salient features of discrete attributes and GAN to generate realistic samples in the embedding space. To address the usability of GAN-generated samples in imbalanced learning, QAST [152] designs a semantic pseudo-labeling module to control the quality of the generated features, and then calibrates their corresponding semantic labels using a classifier committee consisting of multiple pretrained shallow classifiers. The above approaches only generate synthetic examples for the minority classes. Jing et al. [153] propose using GAN to generate a subset of the majority of samples that follow the same distribution as the original dataset. Next, they construct a few balanced subsets, each containing all the minority samples and an equal-sized subset of GAN-generated majority samples. Then, upon each decomposed two-class subset, they use deep metric learning to build a two-class classification model. These models are then integrated to make final predictions. Recently, a diffusion model-based [154] imbalance learning method, namely, TabDDPM [155], has been proposed for generating rational samples.

B. Loss Function

1) Loss Function for Embedding Learning: The idea behind embedding learning is to map complicated (often high-dimensional) metric spaces into easier (often low-dimensional) metric spaces, where interclass and intraclass distances can be optimized simultaneously. In LTL, there are many specially designed embedding learning approaches, such as [114], [117], [118], and [156].

In imbalance learning over relational data, many traditional embedding learning methods have been proposed to transform the learning space. IML [157] is an imbalance learning method that aims to find a more stable neighborhood space for the testing data by performing metric learning on the data and training sample selection according to the testing data iteratively until the selected training samples in two adjacent iterations are the same. MLFP [158] expands the decision boundaries around the positive samples and learns better metric space using a triplet-based cost function on the false negative and false positive samples.

In recent years, many deep imbalance learning approaches have also been proposed, which use neural networks to transform the metric space to learn better embeddings. Khan et al. [159] propose a cost-sensitive (CoSen) neural network method to learn feature representations for both the majority and minority classes. Dong et al. [160] address multilabel imbalance classification by formulating a class rectification loss regularization, which imposes an additional batchwise class balancing on top of the cross-entropy loss. GAMO [161] is an end-to-end deep oversampling model for imbalance classification, which effectively integrates sample generation with classifier training.

2) Loss Reweighting: In LTL, loss-level reweighting methods assign different weights to the loss values of the samples, which include classwise reweighting algorithms such as CB Loss [11] and L2RW [98], and instancewise reweighting algorithms such as IB Loss [14], as illustrated in Section III-E2.

In imbalance learning, there are also a few cost-sensitive learning approaches that modify the internal cost function by assigning different weights to different samples based on their difficulty. Such methods usually assign a higher cost to minority classes to boost the importance of the minority classes during the learning process [162]. To adapt support vector machine (SVM) to imbalanced data, cost-sensitive SVM [163] adds a class-specific classification penalty in its cost function.

Overall, the general ideas for loss-level reweighting in LTL and cost-sensitive learning in imbalance learning are very similar, i.e., they all give higher weights to samples of the minority classes or the misclassified samples.

C. Multiexpert Learning

Multiexpert learning (ensemble learning) techniques are applicable to both LTL and imbalance learning. For instance, the idea of ACE [72] to train cascaded learning models for LTL is similar to that of BalanceCascade [164] and AdaCost [165] for imbalance learning.

There are also a few differences, for instance: 1) knowledge distillation-based ensemble techniques have been adopted in LTL [62], [74], [83], but not in imbalance learning and 2) dichotomy-based ensemble technique for multiclass imbalance learning [166], [167] is not needed in LTL since cross-entropy loss-based LTL classifiers are naturally multiclass classifiers.

D. Evaluation Metrics

In terms of evaluation metrics, existing LTL methods commonly assume that the test sets are class-balanced, that is, all the classes have the same number of test instances in the testing/inference stage. Under such an assumption, only the Top-1 accuracy/overall accuracy metric is reported. However, we note that, in real-world applications, the test sets might also be long-tailed, or at least nonuniformly distributed. In contrast, in imbalance learning, the test sets are often considered to be imbalanced, and various evaluation metrics have been proposed [136], [168], including the overall accuracy, G-mean, and Macro-F1.

Du and Wu [169] argue that the average accuracy metric used in LTL incurs little penalty on classes with very low per-class recalls. They, thus, propose three new metrics for evaluating long-tailed recognition algorithms: the geometric mean and harmonic mean of the per-class recall (referred to as G-mean and H-mean for short), and the lowest per-class recall. They also propose the geometric mean loss (GML), which essentially calculates the logarithm of the averaged prediction probability for samples of each class.

As a short summary, when the test set distributions are skewed, major evaluation metrics including overall accuracy, G-mean, Macro-F1, and the lowest per-class recall should be adopted in LTL and imbalance learning to comprehensively assess the performance of the algorithms.

V. Performance Analysis

A. Datasets

For long-tailed recognition, benchmark datasets include CIFAR-100-LT [12], ImageNet-LT [7], Places-LT [7], and iNaturalist 2018 [170]. For long-tailed object detection and instance segmentation, LVIS [171] is the primary real-world dataset. LVIS v0.5 consists of 1230 categories, whereas LVIS v1.0 contains 1203 categories, with 100k images and 19.8k images for the training and test sets, respectively.

B. Evaluation Metrics

For long-tailed recognition, existing LTL methods all assume that the test set is uniformly distributed, i.e., the sample size for each class in the test set is equal; therefore, only the overall accuracy metric is needed for performance evaluation. Besides, these methods often separately report the specific performance on three super groups, which are "many-shot" (if a class has >100 instances), "medium-shot" (20–100 instances), and "few-shot" (<20 instances).

For long-tailed detection and segmentation, the main evaluation metric is average precision (AP). For the latter, separate performance on three super groups are also reported, which are AP_f for frequent classes (having >100 instances), AP_c for common class (having 11–100 instances), and AP_r for rare classes (having <11 instances).

C. Performance Comparison

1) Long-Tailed Recognition: Table III summarizes the performance of representative long-tailed classification algorithms on four benchmark datasets CIFAR-100-LT (with two different imbalance ratios (IRs) which are 50 and 100), ImageNet-LT, Places-LT, and iNaturalist 2018. It can be observed that, under the CNN backbone (ResNet), MDCS [78], LGLA [84], and SHIKE [76] achieve the best overall performance on all the four benchmark datasets. Besides, NCL [77], SADE [83], RIDE [73], and ACE [72] also obtain remarkable performance, all of which adopt the multiexpert learning framework, due to its effectiveness. While under the Transformer backbone [172], due to the utilization of foundation models, LTGC [53] and VL-LTR [92] achieve significantly better performance than CNN-based methods; in particular, their performance improvements over MDCS [78] and LGLA [84] on ImageNet-LT are near 20% in terms of overall accuracy, which is very substantial.

Besides, we see that the performance of loss reweighting methods tends to saturate since all the possible forms have been exhaustively examined, as can be seen from Table II.

In Table IV, we report the experimental settings of representative LTL algorithms on the iNaturalist 2018 dataset.

TABLE III

PERFORMANCE OF REPRESENTATIVE LONG-TAILED RECOGNITION METHODS ON THE BENCHMARK DATASETS. A, D, M, L, W, AND N ARE SHORT FOR NEURAL ARCHITECTURE AND TRAINING STRATEGY TUNING, DATA-LEVEL REBALANCING, MULTIEXPERT LEARNING (WHICH IS A SUBCATEGORY UNDER A), LOGITS CALIBRATION AND MARGIN ADJUSTMENT, LOSS REWEIGHTING, AND NETWORK OPTIMIZATION, RESPECTIVELY

Method	Year	Venue	CIFAR-100-LT			ImageNet-LT					Places-LT		iNaturalist 2018		Type
			Epoch	IR=100	IR=50	Epoch	Many	Medium	Few	Overall	Epoch	Overall	Epoch	Overall	
CNN Backbone (ResNet)			ResNet-32			ResNet-50					ResNet-152		ResNet-50		
TCR Loss [97]	2020	*CVPR*	200	44.1	49.2	90	-	-	-	48.0	30	30.8	90	67.6	W
cRT [9]	2020	*ICLR*	200	43.8	-	100	58.8	44.0	26.1	47.3	30	36.7	100	62.5	A
BBN [8]	2020	*CVPR*	200	42.6	47.0	-	-	-	-	-	30	-	200	69.6	A
LFME [74]	2020	*ECCV*	200	43.8	-	-	-	-	-	-	30	36.2	90	69.6	M
MetaSAug [22]	2021	*CVPR*	200	48.0	52.3	90	-	-	-	47.4	30	-	100	68.8	D
IB Loss [14]	2021	*ICCV*	200	42.1	46.2	-	-	-	-	-	30	-	90	65.4	W
LogitAdjust [91]	2021	*ICLR*	200	44.1	-	90	-	-	-	48.8	30	-	90	68.4	L
LADE [173]	2021	*CVPR*	200	45.4	50.5	90	62.3	49.3	31.2	51.9	30	38.8	90	70.0	L
DisRobuLT [52]	2021	*ICCV*	200	47.3	57.6	90	64.0	49.8	33.1	53.5	30	-	90	69.7	L
DisAlign [10]	2021	*CVPR*	-	-	-	90	61.3	52.2	31.4	52.9	30	39.3	90	70.2	A
DiVE [68]	2021	*ICCV*	200	45.4	51.1	90	64.1	50.4	31.5	53.1	30	-	90	71.7	A
SSD [67]	2021	*ICCV*	200	46.0	50.5	135	66.8	53.1	35.4	56.0	30	-	90	71.5	A
PaCo [116]	2021	*ICCV*	400	52.0	56.0	400	65.0	55.7	38.2	57.0	30	41.2	400	73.2	A
Hybrid [64]	2021	*CVPR*	200	46.7	51.9	-	-	-	-	-	30	-	90	68.1	M
RIDE [73]	2021	*ICLR*	200	48.0	-	100	66.2	51.7	34.9	54.9	30	40.3	100	72.2	M
ACE [72]	2021	*ICCV*	400	49.4	50.7	100	-	-	-	54.7	30	-	100	72.9	M
SAFA [174]	2022	*ECCV*	200	46.0	50.0	-	-	-	-	-	30	41.5	90	69.8	D
CMO (+RIDE) [26]	2022	*CVPR*	200	50.0	-	100	66.4	53.9	35.6	56.2	30	-	200	72.8	D
TSC [115]	2022	*CVPR*	200	43.8	47.4	100	63.5	49.7	30.4	52.4	30	-	100	69.7	L
DOC [112]	2022	*ECCV*	200	-	-	90	64.2	51.4	31.8	53.7	30	-	100	71.0	L
GCL [15]	2022	*CVPR*	200	48.7	53.6	100	-	-	-	54.9	100	40.6	100	72.0	L
DLSA (+RIDE) [104]	2022	*ECCV*	200	-	-	-	**67.8**	54.5	38.8	57.5	30	-	100	72.8	A
BCL [117]	2022	*CVPR*	200	51.9	56.6	90	-	-	-	56.0	30	-	100	71.8	M
SADE [83]	2022	*NIPS*	200	49.4	-	100	66.5	**57.0**	43.5	58.8	30	40.9	200	72.9	M
NCL [77]	2022	*CVPR*	400	54.2	58.2	400	-	-	-	59.5	30	41.8	400	74.9	M
CUDA (+RIDE) [43]	2023	*ICLR*	200	50.7	53.7	100	65.9	51.7	34.9	54.7	30	-	100	72.4	D
AREA [121]	2023	*ICCV*	200	48.8	51.8	120	-	-	-	49.5	30	-	200	68.3	W
SBCL [110]	2023	*ICCV*	200	44.9	48.7	90	63.8	51.3	31.2	53.4	30	-	100	70.8	L
CR (+RIDE) [131]	2023	*CVPR*	200	49.8	53.7	-	-	-	-	-	30	-	100	73.5	L
CC-SAM [90]	2023	*CVPR*	200	50.8	53.9	200	61.4	49.5	37.1	52.4	30	40.6	200	70.9	A
SuperDisco [105]	2023	*CVPR*	200	53.8	58.3	100	66.1	53.3	37.1	57.1	30	40.3	100	73.6	A
GLMC [111]	2023	*CVPR*	200	57.1	**62.3**	-	-	-	-	-	30	-	100	-	A
BalPoE [75]	2023	*CVPR*	200	52.0	56.3	180	66.0	56.7	**43.6**	58.5	30	-	100	75.0	M
SHIKE [76]	2023	*CVPR*	200	56.3	59.8	220	-	-	-	59.7	30	41.9	220	75.4	M
MDCS [78]	2023	*ICCV*	400	56.1	60.1	400	-	-	-	**60.7**	30	**42.4**	400	75.6	M
LGLA [84]	2023	*ICCV*	400	**57.2**	61.6	180	-	-	-	59.7	30	42.0	400	**76.2**	M
Dual-Phase [33]	2024	*TNNLS*	200	44.79	49.32	-	64.6	48.3	22.1	51	60	39.8	200	69.8	N
LADC [59]	2024	*TNNLS*	430	50.79	54.93	230	-	-	-	52.6	-	-	130	69.33	D
DODA(+RIDE) [49]	2024	*ICLR*	200	50.2	52.6	100	66.6	51.9	35.9	55.8	-	-	100	73.7	D
Transformer Backbone (ViT)			ViT-B			ViT-B					ViT-B		ViT-B		
VL-LTR [92]	2022	*ECCV*	-	-	-	-	**84.5**	**74.6**	59.3	77.2	-	50.1	-	76.8	M
RAC [85]	2022	*CVPR*	-	-	-	-	-	-	-	-	-	47.2	-	80.2	M
LPT [80]	2022	*ICLR*	-	**89.1**	-	-	-	-	-	-	-	50.1	-	76.1	M
LTGC [53]	2024	*CVPR*	-	-	-	-	-	-	**70.5**	**80.6**	-	**54.1**	-	**82.5**	M

TABLE IV

EXPERIMENTAL SETTINGS OF REPRESENTATIVE LONG-TAILED RECOGNITION ALGORITHMS ON THE iNATURALIST 2018 DATASET, INCLUDING THE TRAINING SETTINGS AND TRAINING STRATEGIES/TRICKS, i.e., BELLS AND WHISTLES. "lr", "lr" SCHEDULER, AND "lr DECAY" DENOTE THE LR, THE WAY OF ADJUSTING IT DOWNWARD AFTER CERTAIN EPOCHS, AS WELL AS THE LR DECAY HYPERPARAMETER, RESPECTIVELY. "WEIGHT DECAY" REPRESENTS THE REGULARIZATION/WEIGHT DECAY STRATEGY TO MAKE THE WEIGHTS SMALLER, WHERE L2 REGULARIZATION IS COMMONLY ADOPTED. "BASIC" DENOTES METHODS THAT ADOPT BASIC TRANSFORMATION-BASED AUGMENTATION (SUCH AS ROTATION AND RESIZE), WHILE RandAugment AND CUTMIX DENOTE METHODS THAT ADOPT MORE ADVANCED DATA AUGMENTATION TECHNIQUES FOR IMPROVING THE LTL PERFORMANCE. "BS" AND LDAM ARE SHORT FOR BALANCED SOFTMAX LOSS [79] AND LABEL-DISTRIBUTION-AWARE MARGIN LOSS [12], RESPECTIVELY. THE OPTIMIZER USED FOR ALL METHODS IS SGD, WITH THE MOMENTUM FACTOR SET TO 0.9

Method	Year	Venue	Training Settings (Hyper-Parameters)					Bells and Whistles					
			epoch	batch size	lr	lr scheduler	lr decay	weight decay	knowledge distillation	pre-training	data augmentation	extra loss	MoE
MetaSAug [22]	2021	*CVPR*	100	64	0.01	linear	5×10^{-4}	L2	-	-	basic	-	-
PaCo [116]	2021	*ICCV*	400	128	0.02	cosine	1×10^{-4}	L2	-	MoCo	RandAugment	-	-
RIDE [73]	2021	*ICLR*	100	512	0.2	linear	2×10^{-4}	L2	-	-	basic	LDAM	yes
GCL [15]	2022	*CVPR*	100	512	0.1	cosine	1×10^{-4}	L2	-	-	Mixup	-	-
BCL [117]	2022	*CVPR*	100	256	0.2	cosine	1×10^{-4}	L2	-	SimCLR	RandAugment	-	yes
SADE [83]	2022	*NeurIPS*	200	512	0.2	linear	2×10^{-4}	L2	-	-	basic	BS	yes
CMO [26]	2022	*CVPR*	200	128	0.1	linear	2×10^{-4}	L2	-	-	CutMix	LDAM	-
CUDA [43]	2023	*ICLR*	100	128	0.1	linear	2×10^{-4}	L2	-	-	basic	LDAM	-
CC-SAM [90]	2023	*CVPR*	200	256	0.1	linear	2×10^{-4}	L2	-	-	basic	BS	-
SHIKE [76]	2023	*CVPR*	200+20	128	0.025	linear	5×10^{-4}	L2	logit-distill	-	RandAugment	BS	yes
MDCS [78]	2023	*ICCV*	400	512	0.2	linear	5×10^{-4}	L2	self-distill	-	RandAugment	BS	yes
LGLA [84]	2023	*ICCV*	400	512	0.2	cosine	2×10^{-4}	L2	-	-	RandAugment	-	yes
DODA(+RIDE) [49]	2024	*ICLR*	100	512	0.1	linear	2×10^{-4}	L2	-	-	basic	LDAM	yes
LTGC [53]	2024	*CVPR*	-	-	-	-	-	-	-	CLIP (ViT-B)	GPT-4V & DALL-E Mixup	-	-

Overall, on ImageNet-LT and iNaturalist 2018, the state-of-the-art performance is only 82.5% in terms of overall accuracy, while the best overall performance is only 54.1% on Places-LT, which can hardly satisfy the requirements of real-world applications. Therefore, there is still a large room for performance improvement.

2) Long-Tailed Object Detection and Segmentation: Table V reports the performance of representative long-tailed object detection and segmentation methods on the LVIS v1.0 benchmark dataset. We see that BSGAL [34], RichSem [70], Stepwise [126], PCB [135], ECM [103], and C2AM [96] achieve the best long-tailed object detection performance, while BSGAL [34], GOL [109], ECM [103], PCB [135], C2AM [96], and NORCAL [134] are among the best performers in long-tailed segmentation. We would like to note that Stepwise [126] is implemented based upon Deformable DETR [127], which is essentially a CNN–Transformer combined detection framework, while the other methods are commonly based on Mask R-CNN with ResNet as the backbone.

Overall, the state-of-the-art performance is below 35.4% in both long-tailed object detection and segmentation tasks, and the segmentation performance on rare classes is even lower, which is less than 25.4%. Therefore, there is considerable room for future research.

VI. FUTURE DIRECTIONS

While numerous approaches have been proposed to address the long tail issue, many research opportunities remain.

A. Federated LTL

Federated learning enables multiple clients to learn a global model collaboratively without transmitting the local private data on each client to a centralized server [178]. The problem becomes more challenging when the training data across the clients are both heterogeneous and long-tailed.

Fairness refers to whether a model can perform equally for different categories. Although there are a few preliminary studies [179], fairness within a federated learning system in long-tailed settings needs further studies.

B. Long-Tailed OOD Detection

OOD samples are samples that do not match the training data distribution, regardless of whether they belong to known categories or entirely new (open-set) categories. Wang et al. [180] point that it is sometimes challenging to distinguish between the in-distribution tail class samples and the OOD samples that belong to entirely new categories. To address this issue, Wei et al. [181] expand the in-distribution class space by introducing multiple abstention classes and adopt mixture of experts for classification. It also augments the context-limited tail classes using both the head and OOD classes via Cutmix. At test time, the sum of probabilities for the k abstention classes is used for OOD detection. Liu et al. [7] also investigate the open-set long-tailed recognition problem, in which a long-tailed distribution also contains samples of new/unknown/open-set categories. They propose to simultaneously handle long-tailed classification, few-shot

TABLE V
PERFORMANCE OF REPRESENTATIVE LONG-TAILED OBJECT DETECTION AND SEGMENTATION METHODS ON LVIS v1.0

Method	Year	Venue	Seg.				Det.
			AP	AP_r	AP_c	AP_f	AP^b
BAGS [107]	2020	*CVPR*	23.1	13.1	22.5	28.2	25.8
FASA [89]	2021	*ICCV*	24.1	17.3	22.9	28.5	-
RIO [56]	2021	*ICML*	23.7	15.2	22.5	28.8	24.1
DisAlign [10]	2021	*CVPR*	24.3	8.5	26.3	28.1	23.9
MOSAICOS [17]	2021	*ICCV*	24.5	18.2	23.0	28.8	25.0
RS [175]	2021	*ICCV*	25.2	16.8	24.3	29.9	25.9
EQLv2 [176]	2021	*CVPR*	25.5	17.7	24.3	29.9	25.9
Seesaw [13]	2021	*CVPR*	26.4	19.6	26.1	29.8	27.4
LOCE [18]	2021	*ICCV*	26.6	18.5	26.2	30.7	27.4
NORCAL [134]	2021	*NeurIPS*	26.8	23.9	25.8	29.1	27.8
FreeSeg [177]	2022	*ECCV*	25.2	20.2	23.8	28.9	26.0
AHRL [108]	2022	*CVPR*	25.7	-	-	-	26.4
C2AM [96]	2022	*CVPR*	27.2	16.6	27.2	31.9	27.9
PCB [135]	2022	*CVPR*	27.2	19.0	27.1	30.9	28.1
ECM [103]	2022	*ECCV*	27.4	19.7	27.0	31.1	27.9
GOL [109]	2022	*ECCV*	27.7	21.4	27.7	30.4	27.5
Step-wise [126]	2023	*ICCV*	-	-	-	-	28.7
RichSem [70]	2023	*NIPS*	-	-	-	-	30.6
BSGAL [34]	2024	*ICML*	**31.6**	**25.4**	**30.6**	**35.4**	**35.4**

learning, and open-set recognition (OOD detection) in a unified framework.

C. Active Learning for Long-Tailed Distributions

Active learning approaches iteratively select the most informative samples from underrepresented classes for annotation and training to optimize the use of limited labeled data to improve model performance over time. Choi et al. [182] incorporate class imbalance into active learning by considering both class imbalance and labeling cost. Yang et al. [183] propose a progressive sampling mechanism that explores active learning of OOD samples for enhancing the performance in both long-tailed classification and OOD detection.

D. Long-Tailed Class Incremental Learning

Liu et al. [184] add a layer of learnable scaling weights in the decoupled learning framework to integrate the outputs of the classifier heads for class-incremental learning. Wang et al. [185] propose to use subprotypes to discover and store data distributions for subtasks with long-tailed distributions and adopt the knowledge distillation technique to avoid catastrophic forgetting when training a new task.

E. Long-Tailed Domain Generalization

Domain generalization aims to learn from multiple training domains to generalize to previously unseen target domains. Su et al. [186] introduce the maximum square loss, which has a linearly increasing gradient to address the long-tailed distribution issue in domain generalization. While Su et al. [186] focus on single-domain imbalance, Yang et al. [187] investigate the multiple-domain LTL problem, which aims to learn from multidomain imbalanced data and generalize to all domains. It proposes a loss for bounding the transferability statistics based on the contrastive loss over the positive and negative cross-domain pairs.

F. Long-Tailed Adversarial Training

Adversarial training is a defense strategy aimed at enhancing the robustness of machine learning models against adversarial attacks by incorporating adversarial examples into the training process. Wu et al. [188] investigate the adversarial vulnerability in long-tailed recognition and propose RoBal, which includes a scale-invariant cosine classifier and a two-stage LTL framework. Yue et al. [189] reveal that adversarial training under long-tailed distributions suffers from robust overfitting, i.e., the long-tailed model overfits to current adversarial samples. They show that simply combining balanced Softmax loss [79] and data augmentation can simultaneously alleviate robust overfitting and improve model robustness against adversarial attacks.

G. Diverse Applications With Long Tail Concerns

Long-tailed data widely exist in real-world domains, e.g., defect identification [190], remote sensing [191], nutrition ingredients analysis [192], etc. While we can apply LTL techniques in these domains, such real-world applications often pose new challenges to be addressed, which will advance the LTL techniques, in turn.

VII. CONCLUSION

In this article, we propose a new taxonomy for LTL that divides existing techniques into eight categories, which are data balancing, neural architecture, feature enrichment, logits adjustment, loss function, bells and whistles, network optimization, and posthoc processing techniques. Based on this taxonomy, we systematically review LTL methods in each category, including the latest advances. We also examine the distinctions between imbalance learning and LTL approaches. We summarize the experimental results of representative methods and show that there is still a very large room for performance improvements. In future work, we will design a holistic LTL framework that can integrate all eight different categories of approaches to obtain significantly improved LTL performance.

REFERENCES

[1] C. Huang, Y. Li, C. C. Loy, and X. Tang, "Learning deep representation for imbalanced classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 5375–5384.
[2] W. Ouyang, X. Wang, C. Zhang, and X. Yang, "Factors in finetuning deep model for object detection with long-tail distribution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 864–873.
[3] X. Zhang, Z. Fang, Y. Wen, Z. Li, and Y. Qiao, "Range loss for deep face recognition with long-tailed training data," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 5419–5428.
[4] D. Cao, X. Zhu, X. Huang, J. Guo, and Z. Lei, "Domain balancing: Face recognition on long-tailed domains," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 5671–5679.

[5] Y. Zhong et al., "Unequal-training for deep face recognition with long-tailed noisy data," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 7812–7821.
[6] B. Li et al., "Dynamic class queue for large scale face recognition in the wild," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 3762–3771.
[7] Z. Liu, Z. Miao, X. Zhan, J. Wang, B. Gong, and S. X. Yu, "Large-scale long-tailed recognition in an open world," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 2537–2546.
[8] B. Zhou, Q. Cui, X.-S. Wei, and Z.-M. Chen, "BBN: Bilateral-branch network with cumulative learning for long-tailed visual recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 9719–9728.
[9] B. Kang et al., "Decoupling representation and classifier for long-tailed recognition," in *Proc. 8th Int. Conf. Learn. Represent.*, Jan. 2019, pp. 1–11.
[10] S. Zhang, Z. Li, S. Yan, X. He, and J. Sun, "Distribution alignment: A unified framework for long-tail visual recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 2361–2370.
[11] Y. Cui, M. Jia, T.-Y. Lin, Y. Song, and S. Belongie, "Class-balanced loss based on effective number of samples," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 9268–9277.
[12] K. Cao, C. Wei, A. Gaidon, N. Arechiga, and T. Ma, "Learning imbalanced datasets with label-distribution-aware margin loss," in *Proc. NIPS*, 2019, pp. 1565–1576.
[13] J. Wang et al., "Seesaw loss for long-tailed instance segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 9695–9704.
[14] S. Park, J. Lim, Y. Jeon, and J. Y. Choi, "Influence-balanced loss for imbalanced visual classification," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Jun. 2021, pp. 735–744.
[15] M. Li, Y.-M. Cheung, and Y. Lu, "Long-tailed visual recognition via Gaussian clouded logit adjustment," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 6929–6938.
[16] T. Wang, Y. Zhu, C. Zhao, W. Zeng, J. Wang, and M. Tang, "Adaptive class suppression loss for long-tail object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2021, pp. 3103–3112.
[17] C. Zhang et al., "MosaicOS: A simple and effective use of object-centric images for long-tailed object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 407–417.
[18] C. Feng, Y. Zhong, and W. Huang, "Exploring classification equilibrium in long-tailed object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Oct. 2021, pp. 3417–3426.
[19] Z. Weng, M. G. Ogut, S. Limonchik, and S. Yeung, "Unsupervised discovery of the long-tail in instance segmentation using hierarchical self-supervision," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 2603–2612.
[20] T. Wang et al., "The devil is in classification: A simple framework for long-tail instance segmentation," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 728–744.
[21] X. Hu, Y. Jiang, K. Tang, J. Chen, C. Miao, and H. Zhang, "Learning to segment the tail," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 14042–14051.
[22] S. Li, K. Gong, C. H. Liu, Y. Wang, F. Qiao, and X. Cheng, "MetaSAug: Meta semantic augmentation for long-tailed visual recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 5212–5221.
[23] J. Wang, T. Lukasiewicz, X. Hu, J. Cai, and Z. Xu, "RSG: A simple but effective module for learning imbalanced datasets," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2021, pp. 3784–3793.
[24] Z. Xu, Z. Chai, and C. Yuan, "Towards calibrated model for long-tailed visual recognition from prior perspective," in *Proc. Annu. Conf. Neural Inf. Process. Syst.*, Jan. 2021, pp. 7139–7152.
[25] J. Kim, J. Jeong, and J. Shin, "M2M: Imbalanced classification via major-to-minor translation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2020, pp. 13896–13905.
[26] S. Park, Y. Hong, B. Heo, S. Yun, and J. Y. Choi, "The majority can help the minority: Context-rich minority oversampling for long-tailed classification," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 6877–6886.
[27] Y. Zhang, B. Kang, B. Hooi, S. Yan, and J. Feng, "Deep long-tailed learning: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 9, pp. 10795–10816, Sep. 2023.
[28] Y. Fu et al., "Long-tailed visual recognition with deep models: A methodological survey and evaluation," *Neurocomputing*, vol. 509, pp. 290–309, Oct. 2022.
[29] L. Yang, H. Jiang, Q. Song, and J. Guo, "A survey on long-tailed visual recognition," *Int. J. Comput. Vis.*, vol. 130, no. 7, pp. 1837–1872, Jul. 2022.
[30] B. Kim and J. Kim, "Adjusting decision boundary for class imbalanced learning," *IEEE Access*, vol. 8, pp. 81674–81685, 2020.
[31] S. Alshammari, Y.-X. Wang, D. Ramanan, and S. Kong, "Long- tailed recognition via weight balancing," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 6887–6897.
[32] N. Hasegawa and I. Sato, "Exploring weight balancing on long-tailed recognition problem," in *Proc. 12th Int. Conf. Learn. Represent. (ICLR)*, Jan. 2023, pp. 1–10.
[33] H. Zhang, L. Zhu, X. Wang, and Y. Yang, "Divide and retain: A dual-phase modeling for long-tailed visual recognition," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 10, pp. 13538–13549, Oct. 2024.
[34] M. Zhu et al., "Generative active learning for long-tailed instance segmentation," in *Proc. Int. Conf. Mach. Learn. (ICML)*, Jun. 2024, pp. 1–12.
[35] J. He, "Gradient reweighting: Towards imbalanced class-incremental learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2024, pp. 16668–16677.
[36] S. Ruder, "An overview of gradient descent optimization algorithms," 2016, *arXiv:1609.04747*.
[37] J. Betker et al. (2023). *Improving Image Generation With Better Captions*. Accessed: Jul. 15, 2024. [Online]. Available: https://cdn.openai.com/papers/dall-e-3.pdf
[38] J.-X. Shi, W. Tong, Y. Xiang, and Y.-F. Li, "How re-sampling helps for long-tail learning?" in *Proc. Annu. Conf. Neural Inf. Process. Syst. (NeurIPS)*, Jan. 2023, pp. 1–16.
[39] Q. Dong, S. Gong, and X. Zhu, "Class rectification hard mining for imbalanced deep learning," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 1869–1878.
[40] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *J. Big Data*, vol. 6, no. 1, pp. 1–48, Dec. 2019.
[41] E. D. Cubuk, B. Zoph, D. Mané, V. Vasudevan, and Q. V. Le, "AutoAugment: Learning augmentation strategies from data," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 113–123.
[42] E. D. Cubuk, B. Zoph, J. Shlens, and Q. V. Le, "RandAugment: Practical automated data augmentation with a reduced search space," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2020, pp. 18613–18624.
[43] S. Ahn, J. Ko, and S.-Y. Yun, "CUDA: Curriculum of data augmentation for long-tailed recognition," in *Proc. 11th Int. Conf. Learn. Represent. (ICLR)*, Jan. 2023, pp. 1–20.
[44] H. Zhang, M. Cissé, Y. Dauphin, and D. López-Paz, "Mixup: Beyond empirical risk minimization," in *Proc. 6th Int. Conf. Learn. Represent.*, Jan. 2017, pp. 1–7.
[45] H.-P. Chou, S.-C. Chang, J.-Y. Pan, W. Wei, and D.-C. Juan, "Remix: Rebalanced mixup," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 95–110.
[46] H. Zheng, L. Zhou, H. Li, J. Su, X. Wei, and X. Xu, "BEM: Balanced and entropy-based mix for long-tailed semi-supervised learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2024, pp. 22893–22903.
[47] K. Sohn et al., "FixMatch: Simplifying semi-supervised learning with consistency and confidence," in *Proc. Annu. Conf. Neural Inf. Process. Syst. (NeurIPS), Jan. 2020.*, Jan. 2020, pp. 1–9.
[48] S. Yun, D. Han, S. Chun, S. J. Oh, Y. Yoo, and J. Choe, "CutMix: Regularization strategy to train strong classifiers with localizable features," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6022–6031.
[49] B. Wang et al., "Kill two birds with one stone: Rethinking data augmentation for deep long-tailed learning," in *Proc. 12th Int. Conf. Learn. Represent. (ICLR)*, 2024, pp. 1–8.
[50] J. Liu, Y. Sun, C. Han, Z. Dou, and W. Li, "Deep representation learning on long-tailed data: A learnable embedding augmentation perspective," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2020, pp. 2970–2979.
[51] B. Liu, H. Li, H. Kang, G. Hua, and N. Vasconcelos, "GistNet: A geometric structure transfer network for long-tailed recognition," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 8189–8198.

[52] D. Samuel and G. Chechik, "Distributional robustness loss for long-tail learning," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 9475–9484.

[53] Q. Zhao, Y. Dai, H. Li, W. Hu, F. Zhang, and J. Liu, "LTGC: Long-tail recognition via leveraging LLMs-driven generated content," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2024, pp. 19510–19520.

[54] J. Gao, H. Zhao, Z. Li, and D. Guo, "Enhancing minority classes by mixing: An adaptative optimal transport approach for long-tailed classification," in *Proc. Annu. Conf. Neural Inf. Process. Syst. (NeurIPS)*, 2023, pp. 1–11.

[55] Y. Wang, X. Pan, S. Song, H. Zhang, C. Wu, and G. Huang, "Implicit semantic data augmentation for deep networks," in *Proc. Annu. Conf. Neural Inf. Process. Syst.*, Jan. 2019, pp. 12614–12623.

[56] N. Chang, Z. Yu, Y.-X. Wang, A. Anandkumar, S. Fidler, and J. Alvarez, "Image-level or object-level? A tale of two resampling strategies for long-tailed detection," in *Proc. 38th Int. Conf. Mach. Learn.*, vol. 139, Jun. 2021, pp. 1463–1472.

[57] Y. Xie, Y. Fu, Y. Tai, Y. Cao, J. Zhu, and C. Wang, "Learning to memorize feature hallucination for one-shot image generation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 9120–9129.

[58] S. Khorram, M. Jiang, M. Shahbazi, M. H. Danesh, and L. Fuxin, "Taming the tail in class-conditional GANs: Knowledge sharing via unconditional training at lower resolutions," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2024, pp. 7580–7590.

[59] C. Wang et al., "Label-aware distribution calibration for long-tailed classification," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 5, pp. 6963–6975, May 2024.

[60] C. Zhang, C. Zhang, M. Zhang, I. So Kweon, and J. Kim, "Text-to-image diffusion models in generative AI: A survey," 2023, *arXiv:2303.07909*.

[61] A. Radford et al., "Learning transferable visual models from natural language supervision," in *Proc. Int. Conf. Mach. Learn.*, vol. 139, 2021, pp. 8748–8763.

[62] A. Iscen, A. Araujo, B. Gong, and C. Schmid, "Class-balanced distillation for long-tailed visual recognition," in *Proc. 32nd Brit. Mach. Vis. Conf.*, Jan. 2021, pp. 165–178.

[63] J. Zhang, L. Liu, P. Wang, and C. Shen, "To balance or not to balance: An embarrassingly simple approach for learning with long-tailed distributions," 2019, *arXiv:1912.04486*.

[64] P. Wang, K. Han, X.-S. Wei, L. Zhang, and L. Wang, "Contrastive learning based hybrid networks for long-tailed image classification," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 943–952.

[65] R. Duggal, S. Freitas, S. Dhamnani, D. Horng Chau, and J. Sun, "ELF: An early-exiting framework for long-tailed classification," 2020, *arXiv:2006.11979*.

[66] R. Duggal, S. Freitas, S. Dhamnani, D. H. Chau, and J. Sun, "HAR: Hardness aware reweighting for imbalanced datasets," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2021, pp. 735–745.

[67] T. Li, L. Wang, and G. Wu, "Self supervision to distillation for long-tailed visual recognition," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Oct. 2021, pp. 630–639.

[68] Y.-Y. He, J. Wu, and X.-S. Wei, "Distilling virtual examples for long-tailed recognition," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 235–244.

[69] H. Rangwani, P. Mondal, P. Mondal, M. Mishra, A. R. Asokan, and R. V. Babu, "DeiT-LT: Distillation strikes back for vision transformer training on long-tailed datasets," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2024, pp. 23396–23406.

[70] L. Meng et al., "Learning from rich semantics and coarse locations for long-tailed object detection," in *Proc. Annu. Conf. Neural Inf. Process. Syst. (NeurIPS)*, Jan. 2023, pp. 1–16.

[71] H. Guo and S. Wang, "Long-tailed multi-label visual recognition by collaborative training on uniform and re-balanced samplings," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 15089–15098.

[72] J. Cai, Y. Wang, and J. Hwang, "ACE: Ally complementary experts for solving long-tailed recognition in one-shot," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 112–121.

[73] X. Wang, L. Lian, Z. Miao, Z. Liu, and S. X. Yu, "Long-tailed recognition by routing diverse distribution-aware experts," in *Proc. 9th Int. Conf. Learn. Represent.*, Jan. 2020, pp. 1–7.

[74] L. Xiang, G. Ding, and J. Han, "Learning from multiple experts: Self-paced knowledge distillation for long-tailed classification," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 247–263.

[75] E. S. Aimar, A. Jonnarth, M. Felsberg, and M. Kuhlmann, "Balanced product of calibrated experts for long-tailed recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 19967–19977.

[76] Y. Jin, M. Li, Y. Lu, Y.-M. Cheung, and H. Wang, "Long-tailed visual recognition via self-heterogeneous integration with knowledge excavation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 23695–23704.

[77] J. Li, Z. Tan, J. Wan, Z. Lei, and G. Guo, "Nested collaborative learning for long-tailed visual recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 6939–6948.

[78] Q. Zhao, C. Jiang, W. Hu, F. Zhang, and J. Liu, "MDCS: More diverse experts with consistency self-distillation for long-tailed recognition," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2023, pp. 11563–11574.

[79] J. Ren et al., "Balanced meta-softmax for long-tailed visual recognition," in *Proc. NIPS*, 2020, pp. 4175–4186.

[80] B. Dong, P. Zhou, S. Yan, and W. Zuo, "LPT: Long-tailed prompt tuning for image classification," in *Proc. 11th Int. Conf. Learn. Represent. (ICLR)*, Jan. 2022, pp. 1–17.

[81] J. Zhang, J. Huang, S. Jin, and S. Lu, "Vision-language models for vision tasks: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 46, no. 8, pp. 5625–5644, Aug. 2024.

[82] Z. Yang et al., "Harnessing hierarchical label distribution variations in test agnostic long-tail recognition," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2024, pp. 1–20.

[83] Y. Zhang, B. Hooi, L. Hong, and J. Feng, "Self-supervised aggregation of diverse experts for test-agnostic long-tailed recognition," in *Proc. Annu. Conf. Neural Inf. Process. Syst.*, Jan. 2021, pp. 34077–34090.

[84] Y. Tao et al., "Local and global logit adjustments for long-tailed learning," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2023, pp. 11749–11758.

[85] A. Long et al., "Retrieval augmented classification for long-tail visual recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 6949–6959.

[86] Z. Tian, H. Zhao, M. Shu, Z. Yang, R. Li, and J. Jia, "Prior guided feature enrichment network for few-shot segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 2, pp. 1050–1065, Feb. 2022.

[87] M. Li, Z. Hu, Y. Lu, W. Lan, Y. Cheung, and H. Huang, "Feature fusion from head to tail for long-tailed visual recognition," in *Proc. 28th AAAI Conf. Artif. Intell. (AAAI)*, vol. 38, Mar. 2024, pp. 13581–13589.

[88] H.-J. Ye, D.-C. Zhan, and W.-L. Chao, "Procrustean training for imbalanced deep learning," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 92–102.

[89] Y. Zang, C. Huang, and C. C. Loy, "FASA: Feature augmentation and sampling adaptation for long-tailed instance segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Oct. 2021, pp. 3457–3466.

[90] Z. Zhou, L. Li, P. Zhao, P.-A. Heng, and W. Gong, "Class-conditional sharpness-aware minimization for deep long-tailed recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 3499–3509.

[91] A. K. Menon, S. Jayasumana, A. S. Rawat, H. Jain, A. Veit, and S. Kumar, "Long-tail learning via logit adjustment," in *Proc. 9th Int. Conf. Learn. Represent.*, Jan. 2020, pp. 1–8.

[92] C. Tian, W. Wang, X. Zhu, J. Dai, and Y. Qiao, "VL-LTR: Learning class-wise visual-linguistic representation for long-tailed visual recognition," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2022, pp. 73–91.

[93] J. Ren, M. Zhang, C. Yu, and Z. Liu, "Balanced MSE for imbalanced visual regression," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2022, pp. 7926–7935.

[94] J. Tan et al., "Equalization loss for long-tailed object recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 11662–11671.

[95] M. Li, X. Zhang, C. Thrampoulidis, J. Chen, and S. Oymak, "AutoBalance: Optimized loss functions for imbalanced data," in *Proc. Annu. Conf. Neural Inf. Process. Syst.*, Jan. 2022, pp. 3163–3177.

[96] T. Wang et al., "C2AM loss: Chasing a better decision boundary for long-tail object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 6970–6979.

[97] M. A. Jamal, M. Brown, M.-H. Yang, L. Wang, and B. Gong, "Rethinking class-balanced methods for long-tailed visual recognition from a domain adaptation perspective," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 7610–7619.

[98] M. Ren, W. Zeng, B. Yang, and R. Urtasun, "Learning to reweight examples for robust deep learning," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 4334–4343.

[99] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2999–3007.

[100] H. Zhu, Y. Yuan, G. Hu, X. Wu, and N. M. Robertson, "Imbalance robust softmax for deep embedding learning," in *Proc. Asian Conf. Comput. Vis.*, 2020, pp. 274–291.

[101] Y. Wang et al., "Balancing logit variation for long-tailed semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 19561–19573.

[102] H. Lee and H. Kim, "CDMAD: Class-distribution-mismatch-aware debiasing for class-imbalanced semi-supervised learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2024, pp. 23891–23900.

[103] J. H. Cho and P. Krähenbühl, "Long-tail detection with effective class-margins," in *Proc. Eur. Conf. Comput. Vis.*, Jan. 2022, pp. 698–714.

[104] Y. Xu, Y.-L. Li, J. Li, and C. Lu, "Constructing balance from imbalance for long-tailed image recognition," in *Proc. Eur. Conf. Comput. Vis.*, Jan. 2022, pp. 38–56.

[105] Y. Du, J. Shen, X. Zhen, and C. G. Snoek, "Superdisco: Super-class discovery improves visual recognition for the long-tail," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2023, pp. 19944–19954.

[106] H. Xiong and A. Yao, "Deep imbalanced regression via hierarchical classification adjustment," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2024, pp. 23721–23730.

[107] Y. Li et al., "Overcoming classifier imbalance for long-tail object detection with balanced group softmax," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2020, pp. 10991–11000.

[108] B. Li, "Adaptive hierarchical representation learning for long-tailed object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 2303–2312.

[109] K. P. Alexandridis, J. Deng, A. Nguyen, and S. Luo, "Long-tailed instance segmentation using Gumbel optimized loss," in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 353–369.

[110] C. Hou, J. Zhang, H. Wang, and T. Zhou, "Subclass-balancing contrastive learning for long-tailed recognition," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Oct. 2023, pp. 5395–5407.

[111] F. Du, P. Yang, Q. Jia, F. Nan, X. Chen, and Y. Yang, "Global and local mixture consistency cumulative learning for long-tailed visual recognitions," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 15814–15823.

[112] H. Wang, S. Fu, X. He, H. Fang, Z. Liu, and H. Hu, "Towards calibrated hyper-sphere representation via distribution overlap coefficient for long-tailed learning," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2022, pp. 179–196.

[113] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 1597–1607.

[114] B. Kang, Y. Li, S. Xie, Z. Yuan, and J. Feng, "Exploring balanced feature spaces for representation learning," in *Proc. 9th Int. Conf. Learn. Representat.*, May 2021, pp. 1–15.

[115] T. Li et al., "Targeted supervised contrastive learning for long-tailed recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 6908–6918.

[116] J. Cui, Z. Zhong, S. Liu, B. Yu, and J. Jia, "Parametric contrastive learning," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 695–704.

[117] J. Zhu, Z. Wang, J. Chen, Y.-P.-P. Chen, and Y.-G. Jiang, "Balanced contrastive learning for long-tailed visual recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 6908–6917.

[118] C. Wei, K. Sohn, and C. Mellina, "CReST: A class-rebalancing self-training framework for imbalanced semi-supervised learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2021, pp. 10857–10866.

[119] C. Du, Y. Han, and G. Huang, "SimPro: A simple probabilistic framework towards realistic long-tailed semi-supervised learning," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2024, pp. 1–7.

[120] T. Wei and K. Gan, "Towards realistic long-tailed semi-supervised learning: Consistency is all you need," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 3469–3478.

[121] X. Chen et al., "AREA: Adaptive reweighting via effective area for long-tailed classification," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2023, pp. 19277–19287.

[122] B. Li et al., "Equalized focal loss for dense long-tailed object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 6990–6999.

[123] Z. Zhong, J. Cui, S. Liu, and J. Jia, "Improving calibration for long-tailed recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2021, pp. 16489–16498.

[124] Y. Yang and Z. Xu, "Rethinking the value of labels for improving class-imbalanced learning," in *Proc. Annu. Conf. Neural Inf. Process. Syst.*, Jan. 2020, pp. 19290–19301.

[125] N. Kang, H. Chang, B. Ma, and S. Shan, "A comprehensive framework for long-tailed learning via pretraining and normalization," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 3, pp. 3437–3449, Mar. 2024.

[126] N. Dong, Y. Zhang, M. Ding, and G. H. Lee, "Boosting long-tailed object detection via step-wise learning on smooth-tail data," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2023, pp. 6917–6926.

[127] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, "Deformable DETR: Deformable transformers for end-to-end object detection," in *Proc. 9th Int. Conf. Learn. Represent. (ICLR)*, Oct. 2020, pp. 1–18.

[128] B. Min et al., "Recent advances in natural language processing via large pre-trained language models: A survey," *ACM Comput. Surveys*, vol. 56, no. 2, pp. 1–40, Feb. 2024.

[129] J. Li, T. Tang, W. X. Zhao, J.-Y. Nie, and J.-R. Wen, "Pre-trained language models for text generation: A survey," *ACM Comput. Surveys*, vol. 56, no. 9, pp. 1–39, Oct. 2024.

[130] L. Hu, Z. Liu, Z. Zhao, L. Hou, L. Nie, and J. Li, "A survey of knowledge enhanced pre-trained language models," *IEEE Trans. Knowl. Data Eng.*, vol. 36, no. 4, pp. 1413–1430, Apr. 2024.

[131] Y. Ma, L. Jiao, F. Liu, S. Yang, X. Liu, and L. Li, "Curvature-balanced feature manifold learning for long-tailed classification," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 15824–15835.

[132] M. P. Naeini, F. Gregory Cooper, and M. Hauskrecht, "Obtaining well calibrated probabilities using Bayesian binning," in *Proc. 29th AAAI Conf. Artif. Intell.*, 2015, pp. 2901–2907.

[133] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in *Proc. 34th Intl. Conf. Mach. Learn.*, 2017, pp. 1321–1330.

[134] T.-Y. Pan et al., "On model calibration for long-tailed object detection and instance segmentation," in *Proc. Annu. Conf. Neural Inf. Process. Syst.*, Jan. 2021, pp. 2529–2542.

[135] Y.-Y. He, P. Zhang, X.-S. Wei, X. Zhang, and J. Sun, "Relieving long-tailed instance segmentation via pairwise class balance," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 6990–6999.

[136] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 9, pp. 1263–1284, Sep. 2009.

[137] B. Krawczyk, "Learning from imbalanced data: Open challenges and future directions," *Prog. Artif. Intell.*, vol. 5, no. 4, pp. 221–232, Nov. 2016.

[138] H. Kaur, H. S. Pannu, and A. K. Malhi, "A systematic review on imbalanced data challenges in machine learning: Applications and solutions," *ACM Comput. Surv.*, vol. 52, no. 4, pp. 1–36, Aug. 2019.

[139] G. Wu and E. Y. Chang, "Adaptive feature-space conformal transformation for imbalanced-data learning," in *Proc. 20th Int. Conf. Mach. Learn.*, Aug. 2003, pp. 816–823.

[140] E. Y. Chang, B. Li, G. Wu, and K. Goh, "Statistical learning for effective visual information retrieval," in *Proc. Int. Conf. Image Process.*, vol. 2, 2003, pp. 609–612.

[141] P. Hart, "The condensed nearest neighbor rule," *IEEE Trans. Inf. Theory*, vol. IT-14, no. 3, pp. 515–516, May 1968.

[142] Z. Zhu, Z. Wang, D. Li, Y. Zhu, and W. Du, "Geometric structural ensemble learning for imbalanced problems," *IEEE Trans. Cybern.*, vol. 50, no. 4, pp. 1617–1629, Apr. 2020.

[143] G. E. A. P. A. Batista, R. C. Prati, and M. C. Monard, "A study of the behavior of several methods for balancing machine learning training data," *ACM SIGKDD Explor. Newslett.*, vol. 6, no. 1, pp. 20–29, Jun. 2004.

[144] S. Barua, Md. M. Islam, X. Yao, and K. Murase, "MWMOTE-majority weighted minority oversampling technique for imbalanced data set learning," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 2, pp. 405–425, Feb. 2014.

[145] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, Jun. 2002.

[146] H. Han, W. Y. Wang, and B. H. Mao, "Borderline-SMOTE: A new over-sampling method in imbalanced data sets learning," in *Proc. Int. Conf. Intell. Comput.*, vol. 3644, Aug. 2005, pp. 878–887.

[147] H. He, Y. Bai, E. A. Garcia, and S. Li, "ADASYN: Adaptive synthetic sampling approach for imbalanced learning," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, Jun. 2008, pp. 1322–1328.

[148] E. Ramentol, Y. Caballero, R. Bello, and F. Herrera, "SMOTE-RSB: A hybrid preprocessing approach based on oversampling and undersampling for high imbalanced data-sets using SMOTE and rough sets theory," *Knowl. Inf. Syst.*, vol. 33, no. 2, pp. 245–265, Nov. 2012.

[149] N. Park, M. Mohammadi, K. Gorde, S. Jajodia, H. Park, and Y. Kim, "Data synthesis based on generative adversarial networks," *Proc. VLDB Endowment*, vol. 11, no. 10, pp. 1071–1083, Jun. 2018.

[150] L. Xu, M. Skoularidou, A. Cuesta-Infante, and K. Veeramachaneni, "Modeling tabular data using conditional GAN," in *Proc. Annu. Conf. Neural Inf. Process. Syst.*, Jan. 2019, pp. 7333–7343.

[151] E. Choi, S. Biswal, B. Malin, J. Duke, W. F. Stewart, and J. Sun, "Generating multi-label discrete patient records using generative adversarial networks," in *Proc. Mach. Learn. Healthcare Conf.*, 2017, pp. 286–305.

[152] C. Zhang, Y. Hou, K. Chen, S. Cao, G. Fan, and J. Liu, "Quality-aware self-training on differentiable synthesis of rare relational data," in *Proc. Thirty-Seventh AAAI Conf. Artif. Intell.*, Jun. 2023, vol. 37, no. 5, pp. 6602–6611.

[153] X.-Y. Jing et al., "Multiset feature learning for highly imbalanced data classification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 1, pp. 139–156, Jan. 2021.

[154] J. Ho, A. N. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *Proc. Annu. Conf. Neural Inf. Process. Syst. (NeurIPS)*, Jan. 2020, pp. 1–12.

[155] A. Kotelnikov, D. Baranchuk, I. Rubachev, and A. Babenko, "Tab-DDPM: Modelling tabular data with diffusion models," in *Proc. Int. Conf. Mach. Learn. (ICML)*, Jan. 2022, pp. 17564–17579.

[156] L. Feng, H. Wang, B. Jin, H. Li, M. Xue, and L. Wang, "Learning a distance metric by balancing KL-divergence for imbalanced datasets," *IEEE Trans. Syst. Man, Cybern., Syst.*, vol. 49, no. 12, pp. 2384–2395, Dec. 2019.

[157] N. Wang, X. Zhao, Y. Jiang, and Y. Gao, "Iterative metric learning for imbalance data classification," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, Jul. 2018, pp. 2805–2811.

[158] R. Viola, R. Emonet, A. Habrard, G. Metzler, and M. Sebban, "Learning from few positives: A provably accurate metric learning algorithm to deal with imbalanced data," in *Proc. 29th Int. Joint Conf. Artif. Intell.*, Jul. 2020, pp. 2155–2161.

[159] S. H. Khan, M. Hayat, M. Bennamoun, F. A. Sohel, and R. Togneri, "Cost-sensitive learning of deep feature representations from imbalanced data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 8, pp. 3573–3587, Feb. 2017.

[160] Q. Dong, S. Gong, and X. Zhu, "Imbalanced deep learning by minority class incremental rectification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 6, pp. 1367–1381, Jun. 2019.

[161] S. S. Mullick, S. Datta, and S. Das, "Generative adversarial minority oversampling," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1695–1704.

[162] Z.-H. Zhou and X.-Y. Liu, "On multi-class cost-sensitive learning," *Comput. Intell.*, vol. 26, no. 3, pp. 232–257, 2010.

[163] T. Lee, K. B. Lee, and C. O. Kim, "Performance of machine learning algorithms for class-imbalanced process fault detection problems," *IEEE Trans. Semicond. Manuf.*, vol. 29, no. 4, pp. 436–445, Nov. 2016.

[164] X.-Y. Liu, J. Wu, and Z.-H. Zhou, "Exploratory undersampling for class-imbalance learning," *IEEE Trans. Syst. Man, Cybern., B, Cybern.*, vol. 39, no. 2, pp. 539–550, Apr. 2009.

[165] W. Fan, S. Stolfo, J. Zhang, and P. Chan, "AdaCost: Misclassification cost-sensitive boosting," in *Proc. 16th Int. Conf. Mach. Learn.*, 1999, pp. 97–105.

[166] J. Bi and C. Zhang, "An empirical comparison on state-of-the-art multi-class imbalance learning algorithms and a new diversified ensemble learning scheme," *Knowl.-Based Syst.*, vol. 158, pp. 81–93, Oct. 2018.

[167] Y. L. Murphey, H. Wang, G. Ou, and L. A. Feldkamp, "OAHO: An effective algorithm for multi-class learning from imbalanced data," in *Proc. Int. Joint Conf. Neural Netw.*, Aug. 2007, pp. 406–411.

[168] T. Raeder, G. Forman, and N. V. Chawla, "Learning from imbalanced data: Evaluation matters," in *Data Mining: Foundations and Intelligent Paradigms: Volume 1: Clustering, Association and Classification*. Berlin, Germany: Springer-Verlag, 2012, pp. 315–331.

[169] Y. Du and J. Wu, "No one left behind: Improving the worst categories in long-tailed learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 15804–15813.

[170] G. Van Horn et al., "The iNaturalist species classification and detection dataset," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8769–8778.

[171] A. Gupta, P. Dollár, and R. Girshick, "LVIS: A dataset for large vocabulary instance segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 5356–5364.

[172] A. Dosovitskiy et al., "An image is worth 16×16 words: Transformers for image recognition at scale," in *Proc. 9th Int. Conf. Learn. Represent. (ICLR)*, Jan. 2020, pp. 1–16.

[173] Y. Hong, S. Han, K. Choi, S. Seo, B. Kim, and B. Chang, "Disentangling label distribution for long-tailed visual recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 6622–6632.

[174] Y. Hong, J. Zhang, Z. Sun, and K. Yan, "SAFA: Sample-adaptive feature augmentation for long-tailed image classification," in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 587–603.

[175] K. Oksuz, B. C. Cam, E. Akbas, and S. Kalkan, "Rank & sort loss for object detection and instance segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 2989–2998.

[176] J. Tan, X. Lu, G. Zhang, C. Yin, and Q. Li, "Equalization loss v2: A new gradient balance approach for long-tailed object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 1685–1694.

[177] C. Zhang, T.-Y. Pan, T. Chen, J. Zhong, W. Fu, and W. Chao, "Learning with free object segments for long-tailed instance segmentation," in *Proc. Eur. Conf. Comput. Vis.*, Jan. 2022, pp. 655–672.

[178] X. Shang, Y. Lu, G. Huang, and H. Wang, "Federated learning on heterogeneous and long-tailed data via classifier re-training with federated features," in *Proc. 31st Int. Joint Conf. Artif. Intell.*, Jul. 2022, pp. 2218–2224.

[179] Z. Wang, X. Fan, J. Qi, C. Wen, C. Wang, and R. Yu, "Federated learning with fair averaging," in *Proc. 13th Int. Joint Conf. Artif. Intell.*, Aug. 2021, pp. 1615–1623.

[180] H. Wang et al., "Partial and asymmetric contrastive learning for out-of-distribution detection in long-tailed recognition," in *Proc. Int. Conf. Mach. Learn.*, Jan. 2022, pp. 23446–23458.

[181] W. Tong, B.-L. Wang, and M.-L. Zhang, "EAT: Towards long-tailed out-of-distribution detection," in *Proc. 28th AAAI Conf. Artif. Intell. (AAAI)*, vol. 38, Mar. 2024, pp. 15787–15795.

[182] J. Choi et al., "VaB-AL: Incorporating class imbalance and difficulty with variational Bayes for active learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 6745–6754.

[183] Y. Yang, Y. Zhang, X. Song, and Y. Xu, "Not all out-of-distribution data are harmful to open-set active learning," in *Proc. Annu. Conf. Neural Inf. Process. Syst. (NeurIPS)*, 2023, pp. 1–15.

[184] X. Liu, Y.-S. Hu, X.-S. Cao, A. D. Bagdanov, K. Li, and M.-M. Cheng, "Long-tailed class incremental learning," in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 495–512.

[185] X. Wang, X. Yang, J. Yin, K. Wei, and C. Deng, "Long-tail class incremental learning via independent SUb-prototype construction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2024, pp. 28598–28607.

[186] H. Su et al., "Sharpness-aware model-agnostic long-tailed domain generalization," in *Proc. 28th AAAI Conf. Artif. Intell. (AAAI)*, vol. 38, Mar. 2024, pp. 15091–15099.

[187] Y. Yang, H. Wang, and D. Katabi, "On multi-domain long-tailed recognition, imbalanced domain generalization and beyond," in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 57–75.

[188] T. Wu, Z. Liu, Q. Huang, Y. Wang, and D. Lin, "Adversarial robustness under long-tailed distribution," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 8659–8668.

[189] X. Yue, N. Mou, Q. Wang, and L. Zhao, "Revisiting adversarial training under long-tailed distributions," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2024, pp. 24492–24501.

[190] C.-H. Ho, K.-C. Peng, and N. Vasconcelos, "Long-tailed anomaly detection with learnable class names," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2024, pp. 12435–12446.

[191] W. Zhao, Z. Zhang, J. Liu, Y. Liu, Y. He, and H. Lu, "Center-wise feature consistency learning for long-tailed remote sensing object recognition," *IEEE Trans. Geosci. Remote Sens.*, vol. 62, 2024, Art. no. 5620011.

[192] J. Gao, J. Chen, H. Fu, and Y.-G. Jiang, "Dynamic mixup for multi-label long-tailed food ingredient recognition," *IEEE Trans. Multimedia*, vol. 25, pp. 4764–4773, 2023.

Chongsheng Zhang (Senior Member, IEEE) received the Ph.D. degree from INRIA, Sophia Antiplolis, France, in 2012.

Since then, he has been working with Henan University, Kaifeng, China, where he is currently a Professor and the Leader of the Data Science and Artificial Intelligence Team. He has published more than 70 articles in peer-reviewed journals and conferences, including KDD, AAAI, IJCAI, and IEEE Transactions on Neural Networks and Learning Systems. He has authored eight books and holds 20 Chinese patents. His research interests include long-tailed learning, optical character recognition (OCR), and digital paleography.

George Almpanidis received the Ph.D. degree in informatics from the Aristotle University of Thessaloniki, Thessaloniki, Greece, in 2008.

He was an Associate Professor with Henan University, Kaifeng, China. He has published 21 articles in peer-reviewed journals and conferences, including KDD and IEEE Transactions on Audio, Speech and Language Processing. His research interests include machine learning, information retrieval, and speech processing.

Gaojuan Fan received the Ph.D. degree in information network from Nanjing University of Posts and Telecommunications, Nanjing, China, in 2011.

Since then, she has been working with Henan University, Kaifeng, China, where she is currently an Associate Professor. She has led or participated in several national research projects, and has published more than 20 articles in peer-reviewed journals and conferences. Her research interests include the Internet of Things and data science.

Binquan Deng is currently pursuing the master's degree with the School of Computer and Information Engineering, Henan University, Kaifeng, China, under the supervision of Prof. Chongsheng Zhang.

He has published a first-author paper on open-set long-tailed learning in *Neural Networks Journal*. His research interests include computer vision and long-tailed learning.

Yanbo Zhang received the Ph.D. degree in communication and information systems from East China Normal University, Shanghai, China, in 2009.

Since then, he has been working with Henan University, Kaifeng, China, where he is currently a Full Professor. He has published more than 20 articles in peer-reviewed journals. His research interests include signal processing and artificial intelligence.

Ji Liu received the joint Ph.D. degree from INRIA, Sophia Antipolis, France, and the University of Montpellier, Montpellier, France, in 2016.

He was with Baidu Research, Beijing, China. He is currently a Researcher with Hithink RoyalFlush Information Network Company Ltd., Beijing, China. He has published more than 40 articles in peer-reviewed journals and conferences, including NeurIPS, CVPR, ICCV, ICML, and EMNLP. His research interests include federated learning and distributed machine learning.

Aouaidjia Kamel received the Ph.D. degree in computer science from Shanghai Jiao Tong University, Shanghai, China, in 2019.

He is currently a Post-Doctoral Researcher with the School of Computer and Information Engineering, Henan University, Kaifeng, China, under the supervision of Prof. Chongsheng Zhang. He has published ten articles, including four first-author journal articles at IEEE Transactions on Systems, Man, and Cybernetics: Systems, IEEE Transactions on Multimedia, and *Information Sciences*. His current research interests include computer vision and action recognition.

Paolo Soda is currently a Full Professor of computer science and computer engineering with the University Campus Bio-Medico di Roma, Rome, Italy. He has published more than 180 articles in peer-reviewed journals and conferences. His research interests include artificial intelligence (AI) and machine learning, with applications to medical image processing and analysis.

Dr. Soda has been chairing the IEEE International Technical Committee for Computational Life Sciences, since 2017.

João Gama (Fellow, IEEE) received the Ph.D. degree in computer science from the University of Porto, Porto, Portugal, in 2000.

He is currenrly a Professor with the University of Porto. He has published more than 400 articles in major international conferences and journals. His research interests include machine learning and data mining.

Dr. Gama is an EurAI Fellow. He served as the Chair/Co-Chair for the ECML 2005, ECML PKDD 2015, ECML PKDD 2025, and PAKDD 2022 international conferences.