Long-tailed Learning

Problem Definition

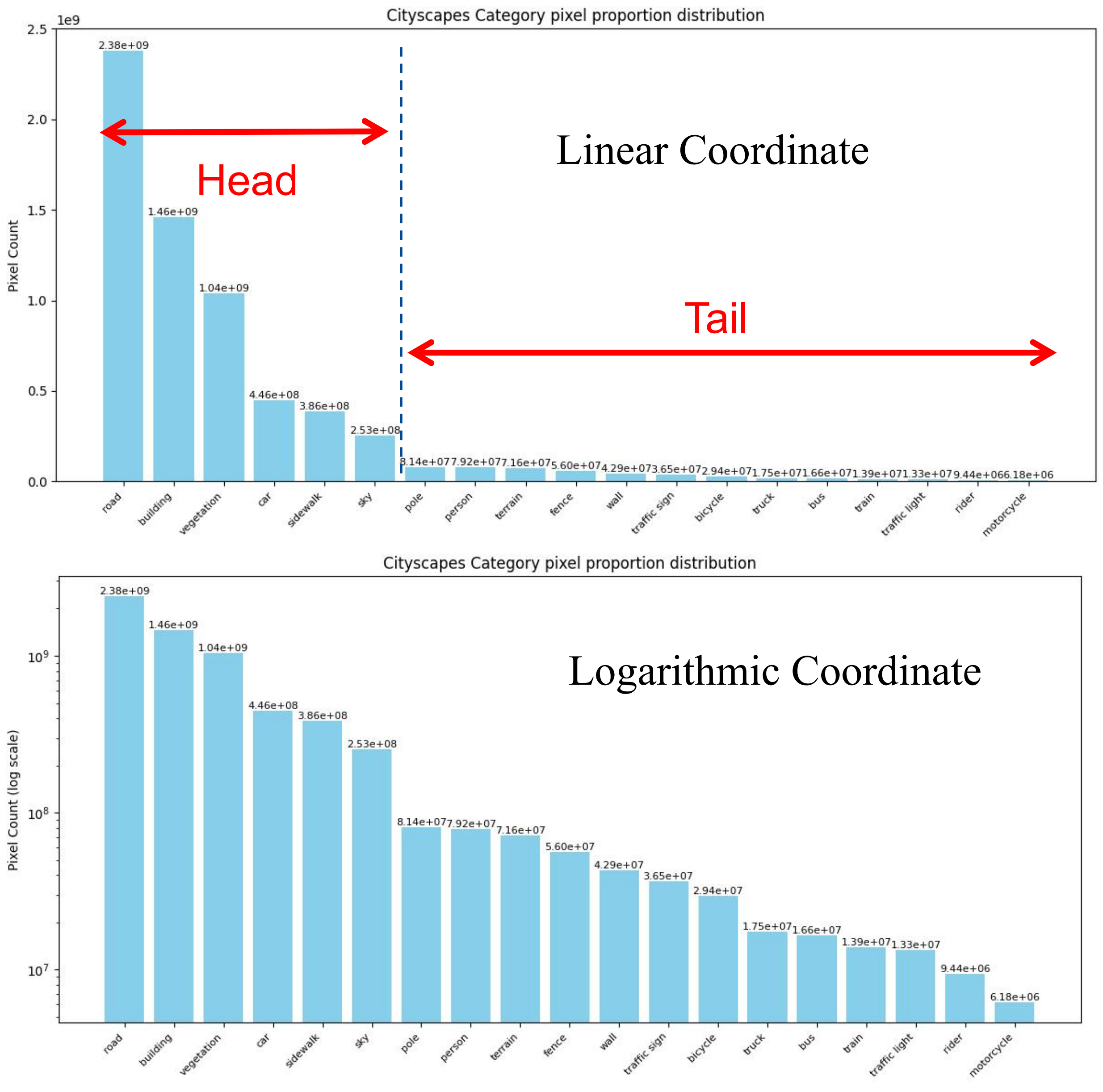
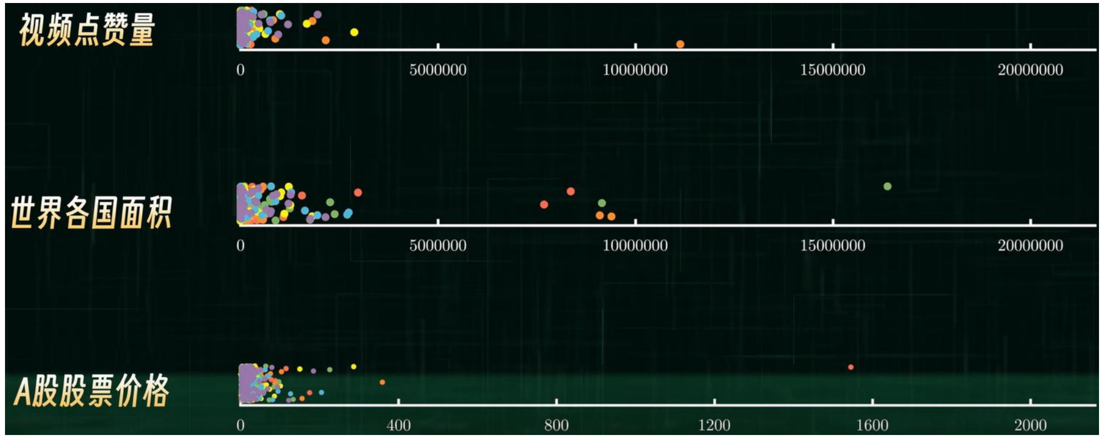
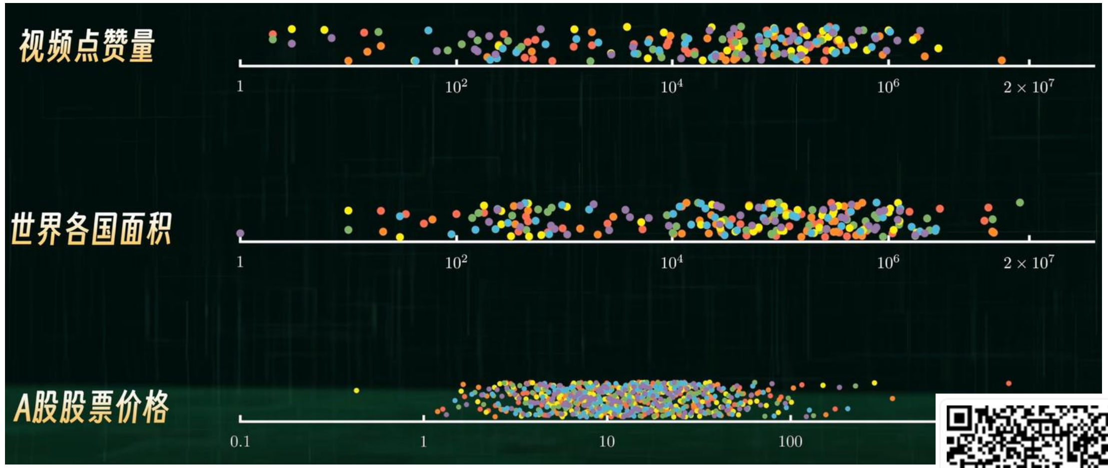


Fig. Cityscapes Dataset pixel distribution

- 数据分布特点：通常近似于对数均匀分布





更多日常生活中长尾现象的解释：

【【毕导】这个定律，预言了你的人生进度条-哔哩哔哩】

https://b23.tv/7ixfn2o

手机扫码观看/分享

Problem Definition

- Motivation：
  - Learn a deep neural network model from a training dataset with a **long-tailed class distribution**.
- Tasks:
  - Image Classification, Detection and Segmentation; Visual Relation Learning, etc.
- Challenges:
  - Imbalanced data numbers across classes, biased to head class
  - Lack of tail-class samples

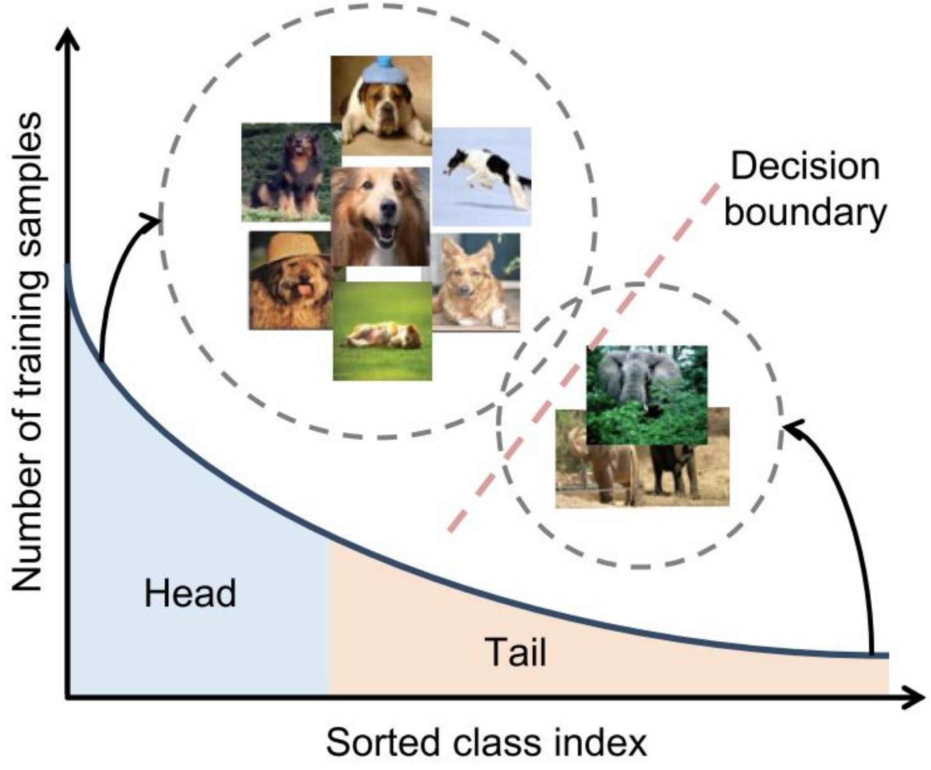


Fig. 1. The label distribution of a long-tailed dataset (e.g., the iNaturalist species dataset [23] with more than 8,000 classes). The head-class feature space learned on these sampled is often larger than tail classes, while the decision boundary is usually biased towards dominant classes.

Box output format :

Datasets

TABLE I
STATISTICS OF LONG-TAILED DATASETS

Task	Dataset	# classes	# training data	# test data
Cls.	ImageNet-LT [15]	1,000	115,846	50,000
	CIFAR100-LT [18]	100	50,000	10,000
	Places-LT [15]	365	62,500	36,500
	iNaturalist 2018 [23]	8,142	437,513	24,426
Det./Seg.	LVIS v0.5 [36]	1,230	57,000	20,000
	LVIS v1 [36]	1,203	100,000	19,800
Multi-label Cls.	VOC-LT [37]	20	1,142	4,952
	COCO-LT [37]	80	1,909	5,000
Video Cls.	VideoLT [38]	1,004	179,352	51,244

"Cls." indicates image classification; "det." represents object detection; "seg." means instance segmentation.

Note: ImageNet, CIFAR100 and Places365 are Sampled from Pareto distributions, and iNaturalist is a real-world long-tailed dataset

Evaluation Metrics

- To evaluate how well class imbalance is resolved:
  - The model performance on all classes
  - The performance on class subsets (i.e., head, middle and tail classes)
- Evaluation Metrics:
  - For balanced test sets: Top-1 Accuracy / Error Rate
  - For imbalanced test sets: mean Average Precision (mAP) / macro accuracy (treat each class equally)

Relationships With Related Tasks

- Class-imbalanced learning
  - LT Learning is a subset of Class-imbalanced learning
  - Differences: Long-tailed class distribution; the number of class; tail class samples.
- Few-shot learning
  - Train models from a limited number of labeled samples (e.g., 1 or 5) per class.
  - A sub-task of long-tailed learning.

Advanced long-tailed learning methods

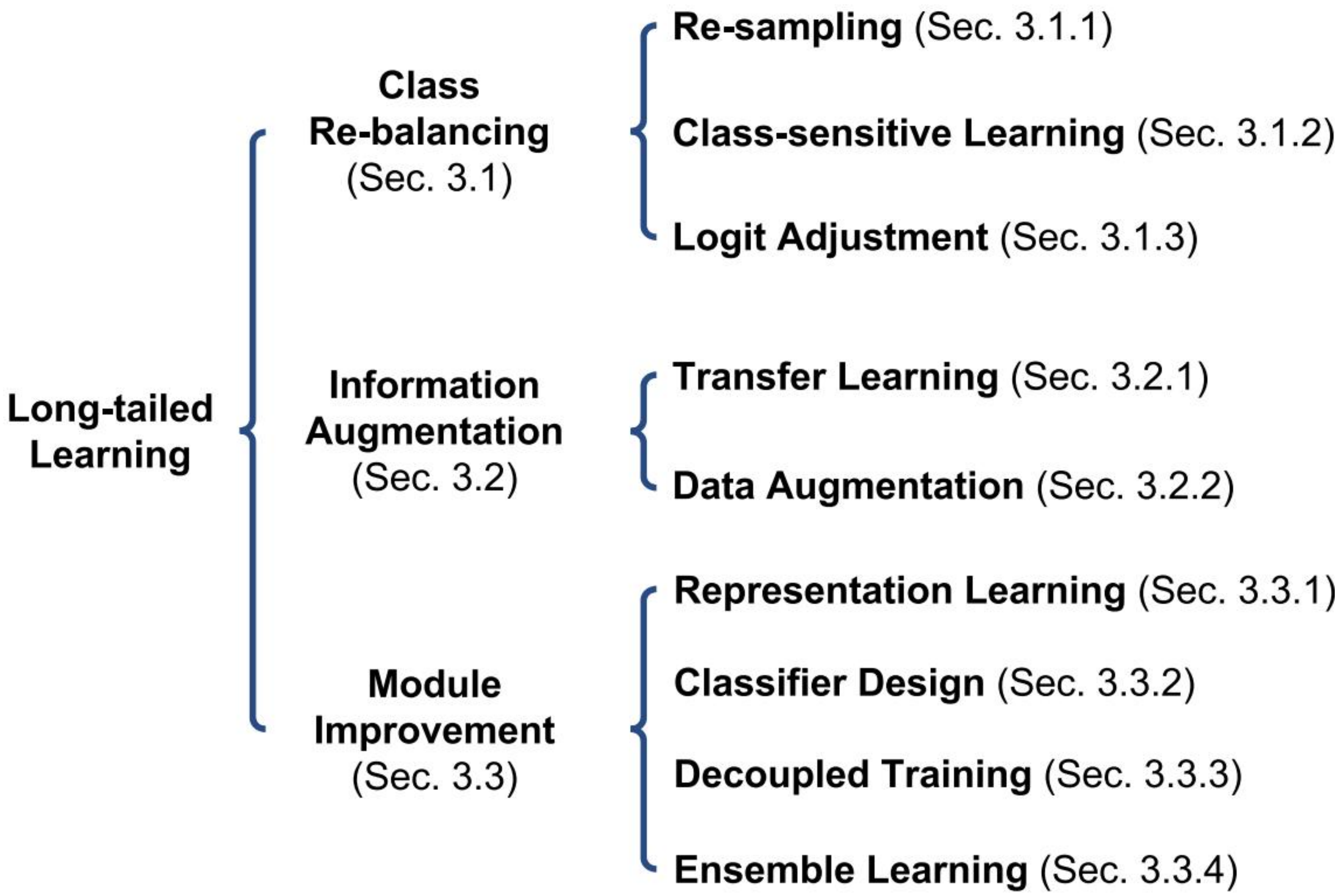


Fig. 2. Taxonomy of existing deep long-tailed learning methods.

Methods: Class Re-balancing

Class Re-balancing (Sec. 3.1)

- **Re-sampling** (Sec. 3.1.1)
- **Class-sensitive Learning** (Sec. 3.1.2)
- **Logit Adjustment** (Sec. 3.1.3)

- Summary
  - **Goal**: Re-balancing classes.
  - **Drawback**: (1) Improve tail-class performance at the cost of lower head-class performance; (2) cannot handle lacking tail class information.

Methods: Class Re-balancing (1)

Class Re-balancing (Sec. 3.1)
- **Re-sampling** (Sec. 3.1.1)
- **Class-sensitive Learning** (Sec. 3.1.2)
- **Logit Adjustment** (Sec. 3.1.3)

- Re-sampling
  - **Targeted Issue**: Random sampling samples more head-class samples than tail-class samples in each sample mini-batch.
  - **Method**: Rebalance classes by adjusting the number of samples per class in each sample batch for model training.
    - (1) The label frequencies of different classes are known a priori;
    - (2) The statistics of model training to guide re-sampling - preferred for real applications.
  - Address the class imbalance issue at the **sample level**

Methods: Class Re-balancing (2)

Class Re-balancing (Sec. 3.1)
- **Re-sampling** (Sec. 3.1.1)
- **Class-sensitive Learning** (Sec. 3.1.2)
- **Logit Adjustment** (Sec. 3.1.3)

- Class-Sensitive Learning
  - **Targeted Issue**: Conventional Softmax CE Loss ignores the class imbalance in data sizes and tends to generate <u>**uneven gradients**</u> for different classes.
  - **Method**: Adjust the training loss values for various classes to re-balance the uneven training effects caused by the imbalance issue.
    - (1) Re-weighting: ;
    - (2) Re-margining: .
  - Resolve the class imbalance issue at the objective level.

Methods: Class Re-balancing (3)

Class Re-balancing (Sec. 3.1)
- **Re-sampling** (Sec. 3.1.1)
- **Class-sensitive Learning** (Sec. 3.1.2)
- **Logit Adjustment** (Sec. 3.1.3)

- Logits Adjustment
  - **Targeted Issue**: Logit adjustment is <u>*Fisher consistent*</u> to minimize the average per-class error.
  - **Method**: Resolve the class imbalance by adjusting the prediction logits of a class-biased deep model.
    - (1) Post-adjust the predictions of biased deep models: If the training label frequencies are known;
    - (2) Learn an adaptive calibration function: training label frequencies are unkown.
  - Address the class imbalance at the **Prediction Level**.

Methods: Information Augmentation

Information Augmentation (Sec. 3.2)

- **Transfer Learning** (Sec. 3.2.1)
- **Data Augmentation** (Sec. 3.2.2)

- Summary
  - **Goal**: Introduce additional information into model training, so that the model performance can be improved for long-tailed learning.
  - **Drawback**: (1) ignore the class imbalance and inevitably augment more head-class samples than tail-class samples.

Methods: Information Augmentation (1)

Information Augmentation (Sec. 3.2)
- **Transfer Learning** (Sec. 3.2.1)
- **Data Augmentation** (Sec. 3.2.2)

- Transfer Learning
  - **Targeted Issue**: .
  - **Method**: Transfer the knowledge from a source domain (e.g., datasets) to enhance model training on a target domain.
    - (1) Model pre-training: fine-tunes the model on a more class-balanced training subset;
    - (2) Knowledge distillation: Train a student model based on the outputs of a well-trained teacher model;
    - (3) Head-to-tail model transfer: ;
    - (4) Self-training: learn well-performing models from a small number of labeled samples and massive unlabeled samples.
  - Address the class imbalance at the **Prediction Level**.
  - Advantages: Most of them can be used together.

Methods: Information Augmentation (2)

Information Augmentation (Sec. 3.2)
- **Transfer Learning** (Sec. 3.2.1)
- **Data Augmentation** (Sec. 3.2.2)

- Data Augmentation
  - **Targeted Issue**: Tail-class samples have much smaller intra-class variance than head-class samples, leading to biased feature spaces and decision boundaries.
  - **Method**: Enhance the size and quality of datasets by applying pre-defined transformations to each data / feature for model training.
    - (1) Head-to-tail transfer augmentation: Transfer the knowledge from head classes to augment tail-class samples;
    - (2) Non-transfer augmentation: improve or design conventional data augmentation methods.
  - Address the class imbalance at the **Sample or Feature Levels**.

Methods: Module Improvement

- Summary
  - **Goal**: Seek to address long-tailed problems by improving network modules.
  - **Advantages**: Thanks to the aggregation of multiple experts, are able to achieve better long-tailed performance without sacrificing the performance on any class subsets.

Module Improvement (Sec. 3.3)

- **Representation Learning** (Sec. 3.3.1)
- **Classifier Design** (Sec. 3.3.2)
- **Decoupled Training** (Sec. 3.3.3)
- **Ensemble Learning** (Sec. 3.3.4)

Methods: Module Improvement (1)

- Representation Learning
  - **Targeted Issue**: Improves the feature extractor.
  - **Method**: Three main paradigms:
    - (1) Metric learning: Designing task-specific distance metrics for establishing similarity or dissimilarity between data;
    - (2) Prototype learning: Learn class-specific feature prototypes to enhance long-tailed learning performance.
    - (3) Sequential training: Learn data representation in a continual way.
  - Address the class imbalance at the **Feature Levels**.

Module Improvement (Sec. 3.3)

- **Representation Learning** (Sec. 3.3.1)
- **Classifier Design** (Sec. 3.3.2)
- **Decoupled Training** (Sec. 3.3.3)
- **Ensemble Learning** (Sec. 3.3.4)

Methods: Module Improvement (2)(3)

- Classifier Design
  - **Targeted Issue**: Larger classifier weight norms for head classes than tail classes.
  - Address the class imbalance at the **Classifier Levels**.
- Decoupled Training
  - Method: Decouples the learning procedure into representation learning and classifier training.
  - Address the class imbalance at the **Feature and Classifier Levels**.

Module Improvement (Sec. 3.3)

- **Representation Learning** (Sec. 3.3.1)
- **Classifier Design** (Sec. 3.3.2)
- **Decoupled Training** (Sec. 3.3.3)
- **Ensemble Learning** (Sec. 3.3.4)

Methods: Module Improvement (4)

- **Ensemble Learning**
  - **Method**: Generate and combine multiple network modules.
  - Address the class imbalance at the **Model Level**.

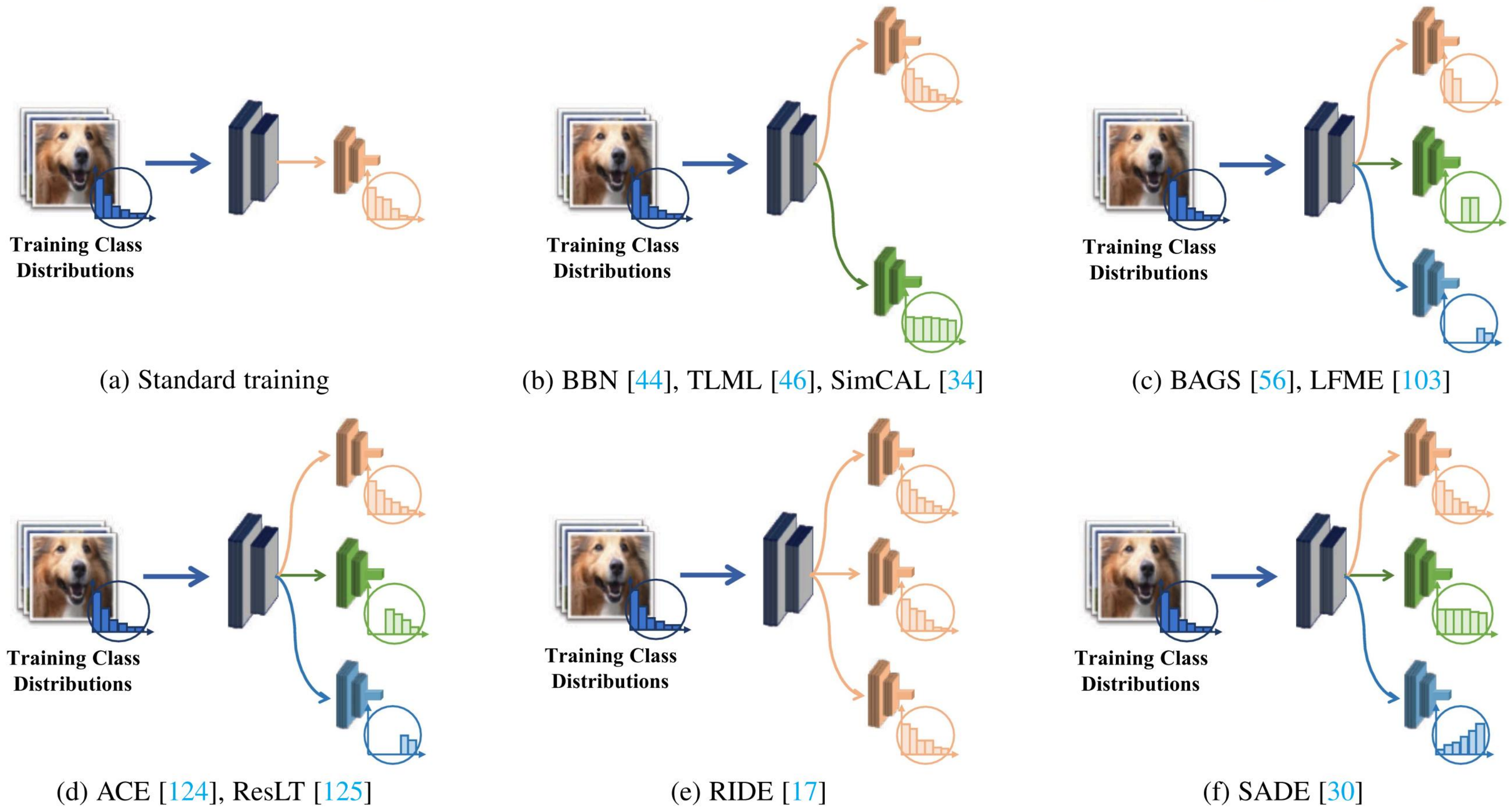


Fig. 3. Illustrations of existing ensemble-based long-tailed methods. Compared to standard training (a), the trained experts by ensemble-based methods (b-f) may have different expertise, e.g., being skilled in different class distributions or different class subsets (indicated by different colors). For example, BBN and SimCAL train two experts for simulating the original long-tailed and uniform distributions so that they can address the two distributions well. BAGS, LFME, ACE, and ResLT train multiple experts by sampling class subsets, so that different experts can particularly handle different sets of classes. SADE directly trains multiple experts to separately simulate long-tailed, uniform and inverse long-tailed class distributions from a stationary long-tailed distribution, which enables it to handle test sets with agnostic class distributions based on self-supervised aggregation.