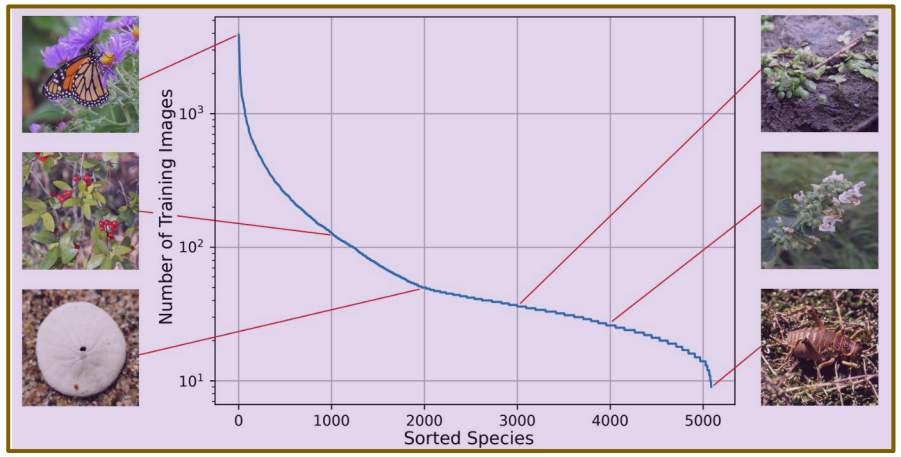
Trustworthy Machine Learning on Imbalance Data

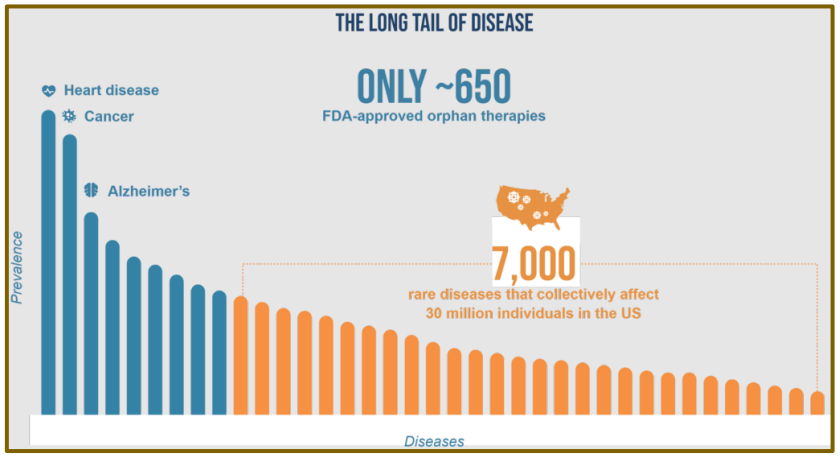
Jiangchao Yao

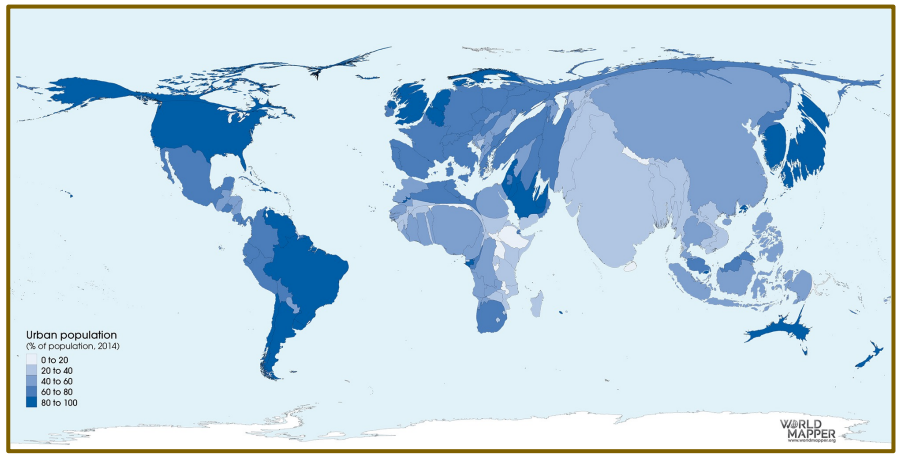
CMIC, Shanghai Jiao Tong University

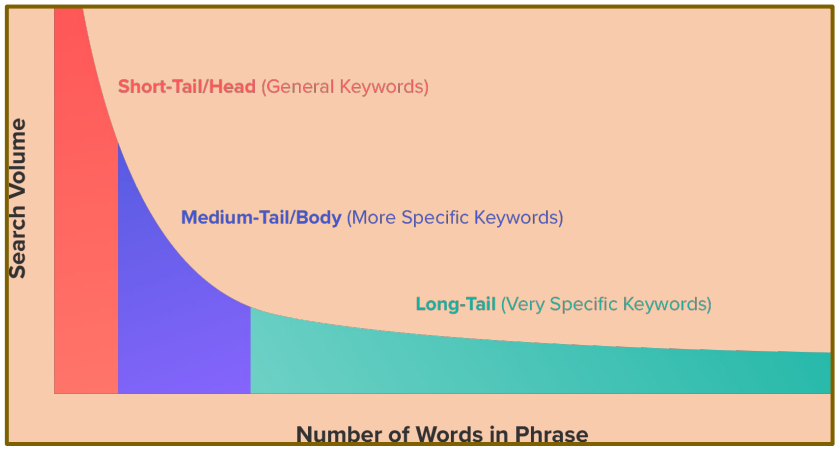
The Imbalance Nature of Data

Large-scale natural sources are very imbalance, usually following a long-tailed distribution.









[1] Van Horn et al. The iNaturalist Species Classification and Detection Dataset. CVPR 2018.

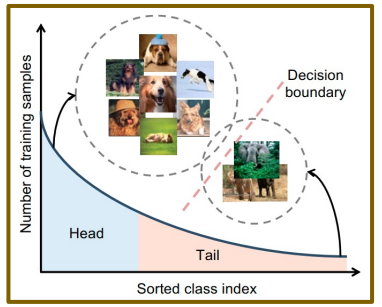
[2] Gregory et al. CXR-LT challenge. ICCV CVAMD 2023.

[3] https://worldmapper.org/maps/urban-population-relative-2014/

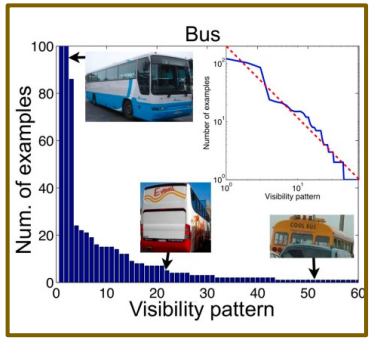
[4] https://seopressor.com/blog/short-tail-or-long-tail-keywords/

A direct decomposition on the risk minimization

$$\min_f R(f) = E_{P(x,y)}[\ell(f(x), y)] = \sum_{k=1}^{K} \boldsymbol{P(y=k)} E_{P(x|y=k)}[\ell(f(x), y)]$$



Minority ("generalized" conceptual) classes have weak importance for training, which will be easily ignored in the early phase especially for overparameterized DNNs [1] *(or namely, will be sacrificed first if it is not sufficient for the model to learn)*.



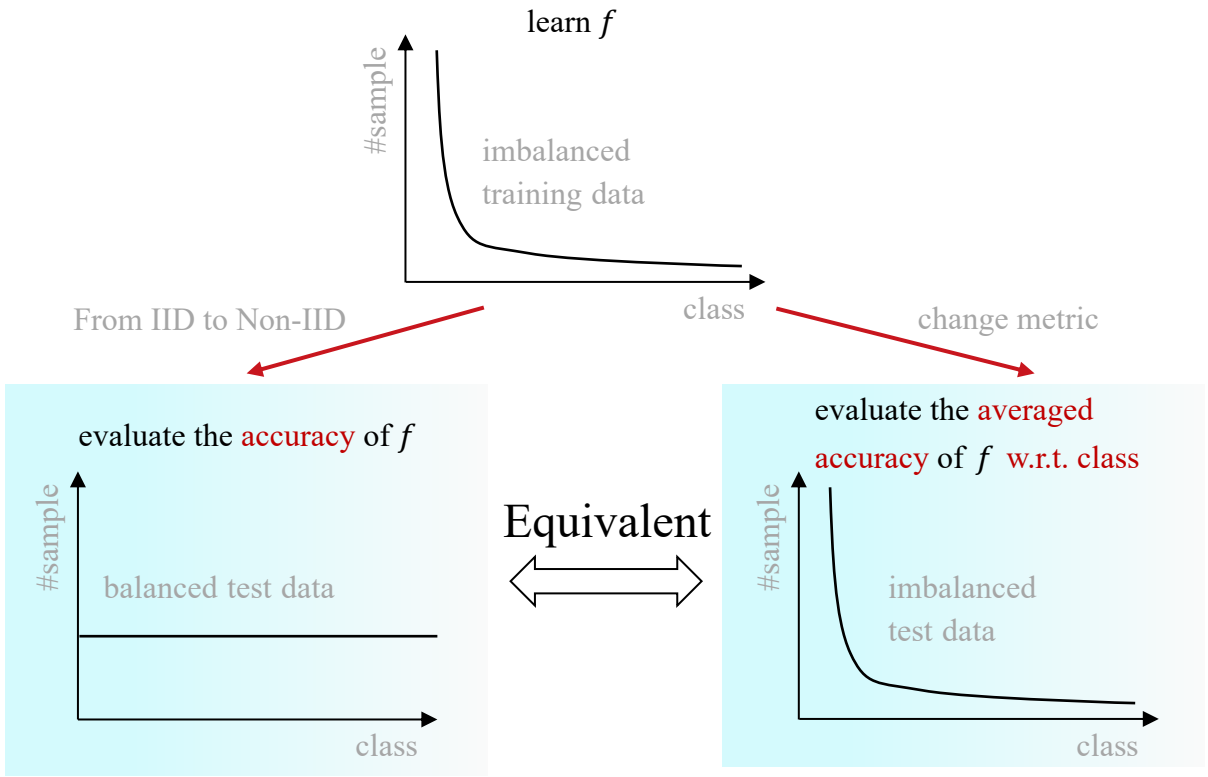
However, in real applications, the value of classes cannot be absolutely characterized by their quantity, and instead, sometimes **less is more** for sustainable long-term development.

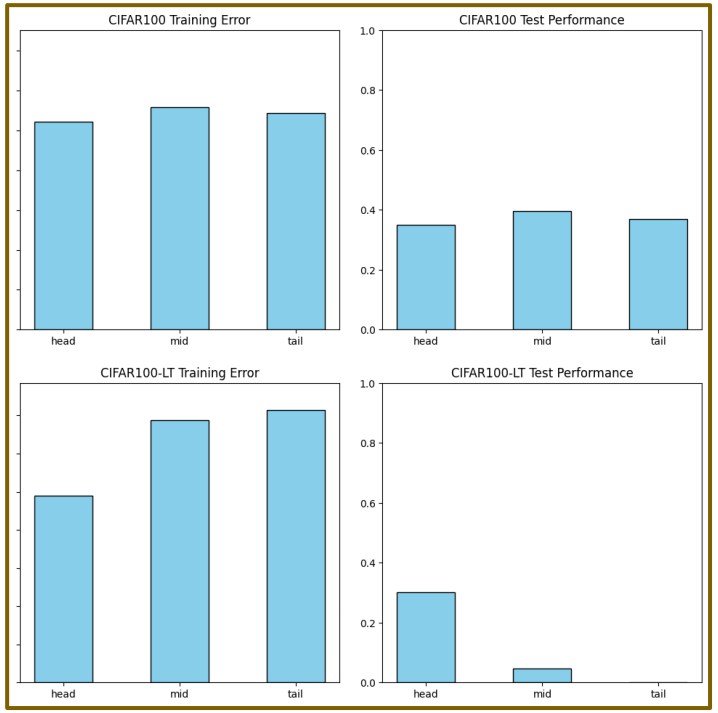
- **Fairness** w.r.t. diversity e.g., small populations of gender, race and consumers
- **Cost-sensitive scenarios** e.g., medical disease diagnosis and treatment (COVID-19)

[1] Vitaly Feldman. "Does Learning Require Memorization? A Short Tale about a Long Tail." SIGACT 2020.

Why the Resulted Imbalanced Learning is Special

A critical highlight on the evaluation, different from the ordinary IID learning





The change in evaluation metric induces **an statistical consistency problem** on applying conventional learning methods, that is,

What we design during training should be statistically consistent with what we pursue about the evaluation.

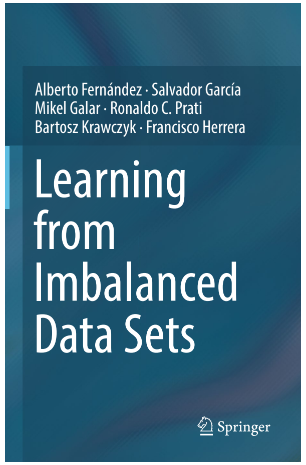
The Historical Development

The development of imbalance learning

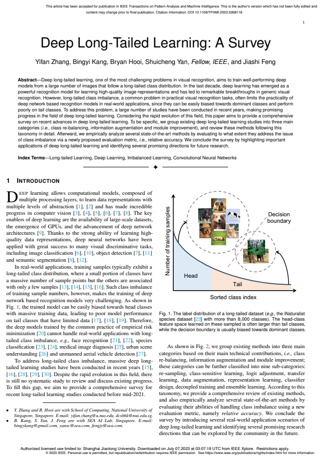
1998 — 2009 — 2018 — 2023

methods

Applications



Deep learning



[1] Karakoulas et al. Optimizing Classifers for Imbalanced Training Sets. NIPS 1998.

[2] He et al. Learning from Imbalanced Data. TKDE 2009.

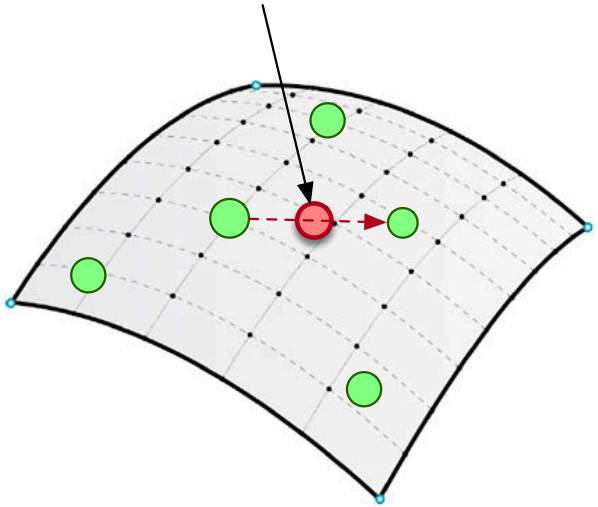
[3] Fernández et al. Learning from Imbalanced Data Sets. Springer, 2018.

[4] Zhang et al. Deep Long-tailed Learning: A Survey. TPAMI 2023.

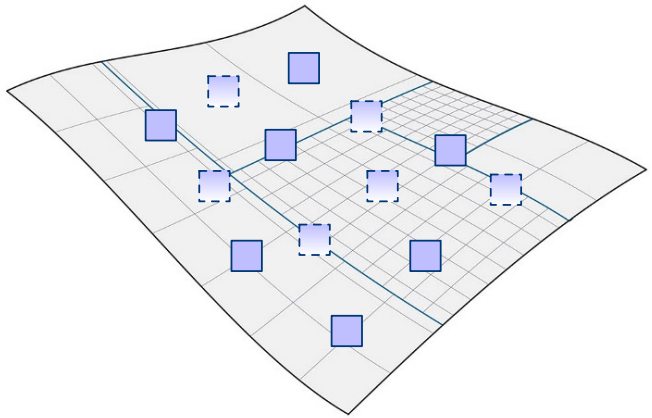
SMOTE: Synthetic Minority Over-sampling Technique

Motivation: Replication of the minority class does not cause its decision boundary to spread into the majority class region (but overfitting).

Interpolation on minority manifold



Under-sampling in the majority class



The main idea of SMOTE: augmentation for minority class by interpolation instead of over-sampling with replacement.

Interpolation is limited by the samples. Thus, SMOTE also always runs with the under-sampling for majority class.

Increase minority diversity and decrease majority diversity

Chawla, Nitesh V., et al. "SMOTE: Synthetic Minority Over-Sampling Technique." *JAIR 2002*.

Retrospection-II: Threshold-Moving Method

Threshold-Moving: adjust the prediction in a post-hoc manner.

Motivation: The over-confident prediction for majority or the low-confident prediction for minority can be calibrated after training.

THE THRESHOLD-MOVING ALGORITHM

Training phase:
1. Let S be the original training set.
2. Train a neural network from S.

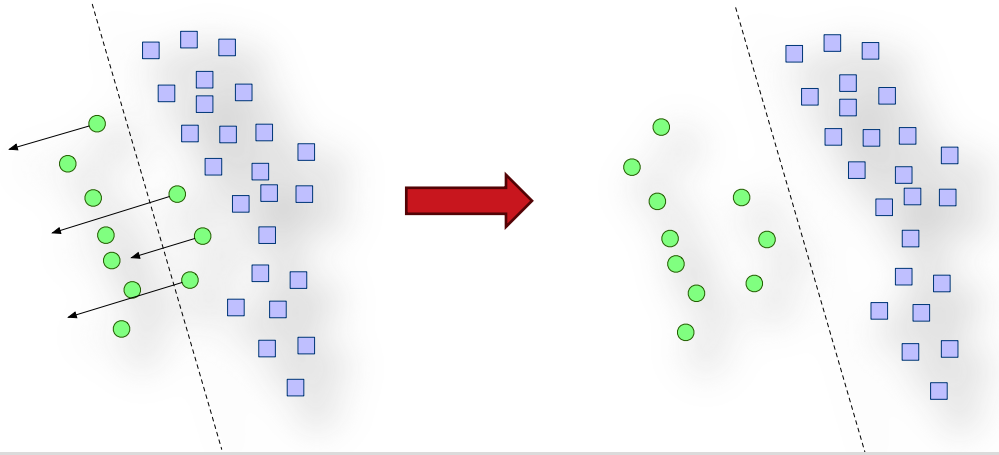
Test phase:
1. Generate real-value outputs with the trained neural network.
2. For every output, multiply it with the sum of the costs of misclassifying the corresponding class to other classes.
3. Return the class with the biggest output.

Moving function

$$\hat{p}_k = \frac{p_k * \sum_{k\prime=1}^{K} C[k][k']}{\eta}$$

where p_k is the probabilistic prediction, $C[k][k']$ is the cost mis-predicted from class k to k', and η is renormalization parameter.

Sampling methods might not always show promise in multi-class imbalance learning, but threshold-moving way does.



Let $p_k = \frac{e^{z_k}}{\sum_{k\prime}^{K} e^{z_{k\prime}}}$ denote the class prediction. If we set $\sum_{k\prime=1}^{K} C[k][k'] = e^{-\tau \log \pi_k}$ where π_k is the class prior and τ is the temperature, the threshold-moving method recovers the popular **logit adjustment** method for long-tailed learning.

Majority classes have the smaller cost than minority classes, e.g., $e^{-\tau \log \pi_k}$ is monotonously decreasing.

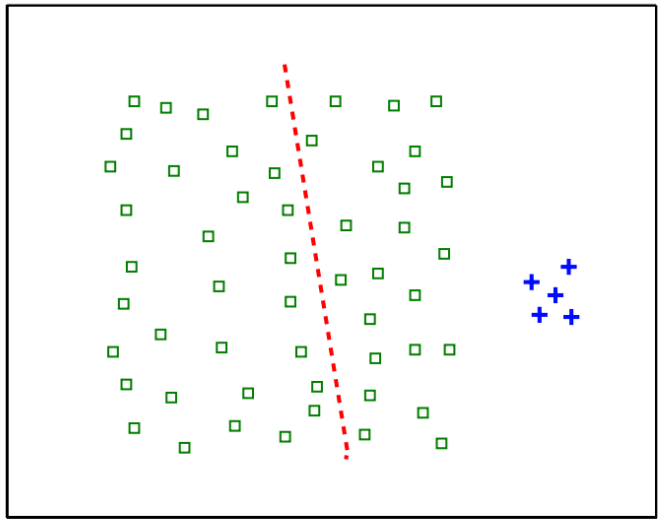
Zhou, Z. H., & Liu, X. Y. Training Cost-Sensitive Neural Networks with Methods Addressing the Class Imbalance Problem. *TKDE, 2005*.

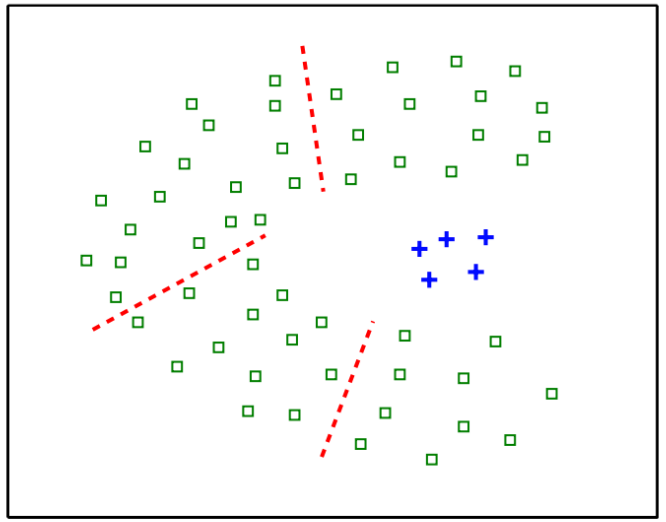
Retrospection-III: Pseudo Labeling

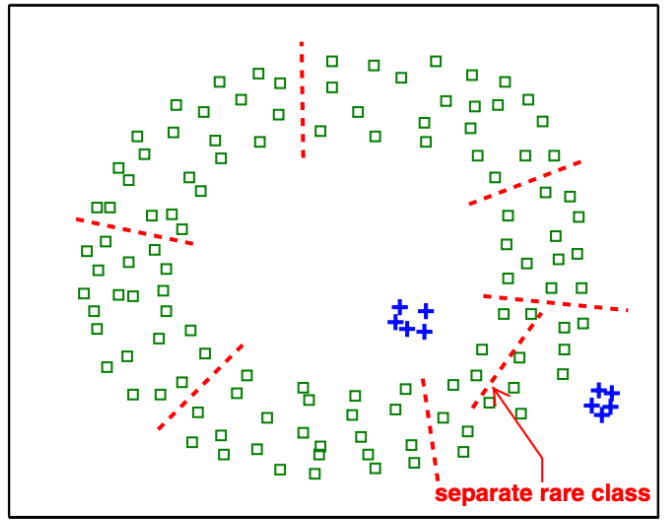
COG: Local Decomposition for Rare Class Analysis

Intuition: Quantity imbalance limits the learning pace of minority over majority. We can adjust the quantities by decomposition.

How to properly decompose the majority classes (or including minority classes) into subclasses to balance the training?







Phase I: local clustering
1. for class $i = 1$ to c // *"c" represents #classes*
2. clusterLabel(i) = Clustering$(\mathcal{D}(i), \boldsymbol{K}(i))$;
3. $\mathcal{D}(i)^*$ = changeLabel$(\mathcal{D}(i)$, clusterLabel$(i))$;
4. end for

Phase II: over-sampling (for COG-OS only)
5. for class $j = 1$ to c
6. $\mathcal{D}(j)^{**}$ = replicate$(\mathcal{D}(j)^*, \boldsymbol{r}(j))$
7. end for
8. $\mathcal{D}^{**} = \bigcup_{j=1}^{c}(\mathcal{D}(j)^{**})$;

Phase III: training
9. $\mathbb{M}$ = train$(\mathcal{D}^{**}, \mathbb{L})$;

Phase IV: predicting
10. $\boldsymbol{p}'$ = predict$(\mathcal{T}, \mathbb{M})$;
11. $\boldsymbol{p}$ = convertLabel$(\boldsymbol{p}')$;

Wu. J. J. et al. COG: Local Decomposition for Rare Class Analysis. *DMKD, 2010*.

On Statistical Consistency of Binary Classification with Balanced Accuracy

Motivation: The early ERM theory is developed for the instance-wise evaluation, but cannot guarantee the consistency for balanced measure.

$$\text{Accuracy} = \mathrm{E}_{p(x,y)}[h(x) = y] \quad \Rightarrow \quad \text{Balanced Accuracy} = \frac{\sum_{k\in\{-1,1\}} \mathrm{E}_{p(x|y=k)}[h(x)=k]}{2}$$

If we consider the balanced accuracy, how to modify the algorithm to satisfy the statistical consistency?

Theorem 3. *Let D be a probability distribution on $\mathcal{X}\times\{\pm 1\}$ satisfying Assumption A. Let $\widehat{p}_S$ denote any estimator of $p = \mathbf{P}(y=1)$ satisfying $\widehat{p}_S \in (0,1)$ and $\widehat{p}_S \xrightarrow{P} p$. Let $\widehat{\eta}_S : \mathcal{X}\to[0,1]$ denote any class probability estimator satisfying $\mathbf{E}_x\left[|\widehat{\eta}_S(x) - \eta(x)|^r\right] \xrightarrow{P} 0$ for some $r \geq 1$, and let $h_S(x) = \operatorname{sign}(\widehat{\eta}_S(x) - \widehat{p}_S)$. Then*

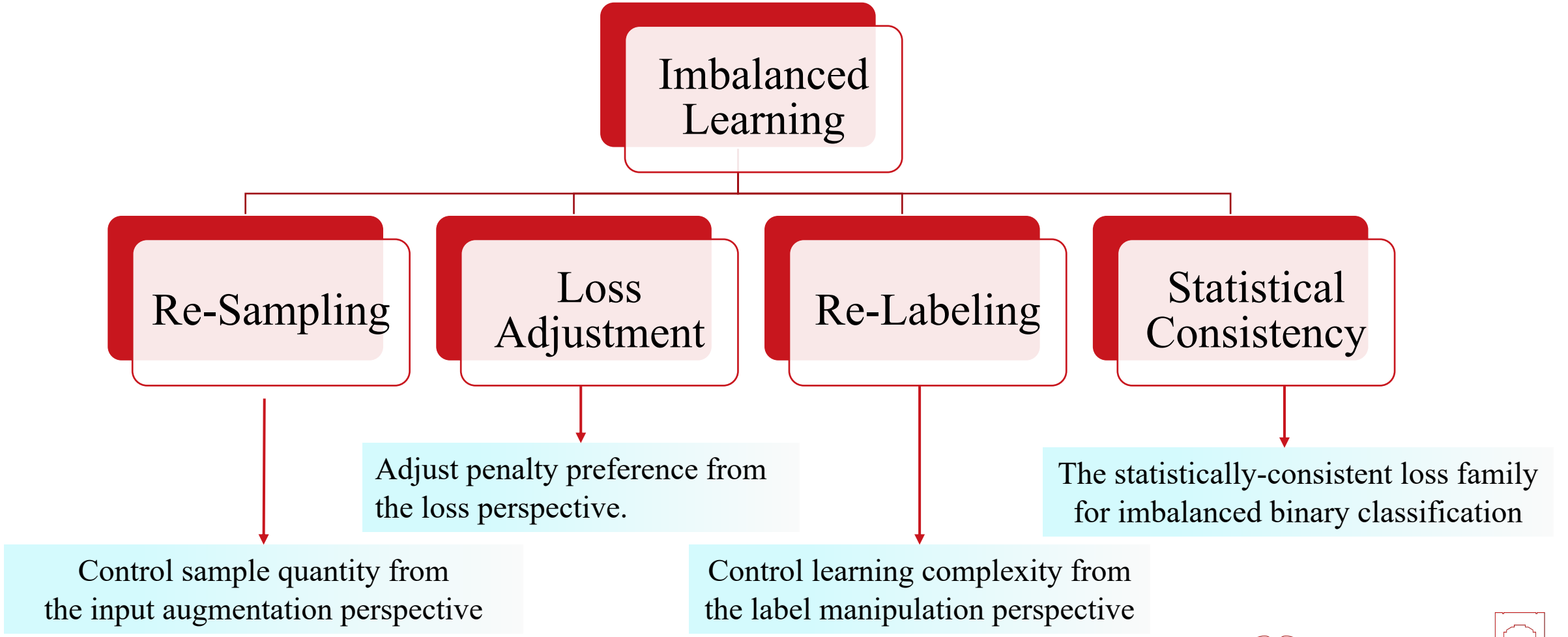
$$\operatorname{regret}_D^{\mathrm{AM}}[h_S] \xrightarrow{P} 0\,.$$

Algorithm 1 Plug-in with Empirical Threshold

1: **Input:** $S = ((x_1, y_1), \ldots, (x_n, y_n)) \in (\mathcal{X}\times\{\pm 1\})^n$
2: **Select:** (a) Proper (composite) loss $\ell : \{\pm 1\}\times\bar{\mathbb{R}}\to\bar{\mathbb{R}}_+$, with link function $\psi : [0,1]\to\bar{\mathbb{R}}$; (b) RKHS $\mathcal{F}_K$ with positive definite kernel $K : \mathcal{X}\times\mathcal{X}\to\mathbb{R}$; (c) regularization parameter $\lambda_n > 0$
3: $f_S \in \operatorname{argmin}_{f\in\mathcal{F}_K} \left\{\frac{1}{n}\sum_{i=1}^n \ell(y_i, f(x_i)) + \lambda_n \|f\|_K^2\right\}$
4: $\widehat{\eta}_S = \psi^{-1}\circ f_S$
5: $\widehat{p}_S =$ (as in Eq. (2))
6: **Output:** Classifier $h_S(x) = \operatorname{sign}(\widehat{\eta}_S(x) - \widehat{p}_S)$

Menon, Aditya, et al. "On the Statistical Consistency of Algorithms for Binary Classification under Class Imbalance." *ICML* 2013.

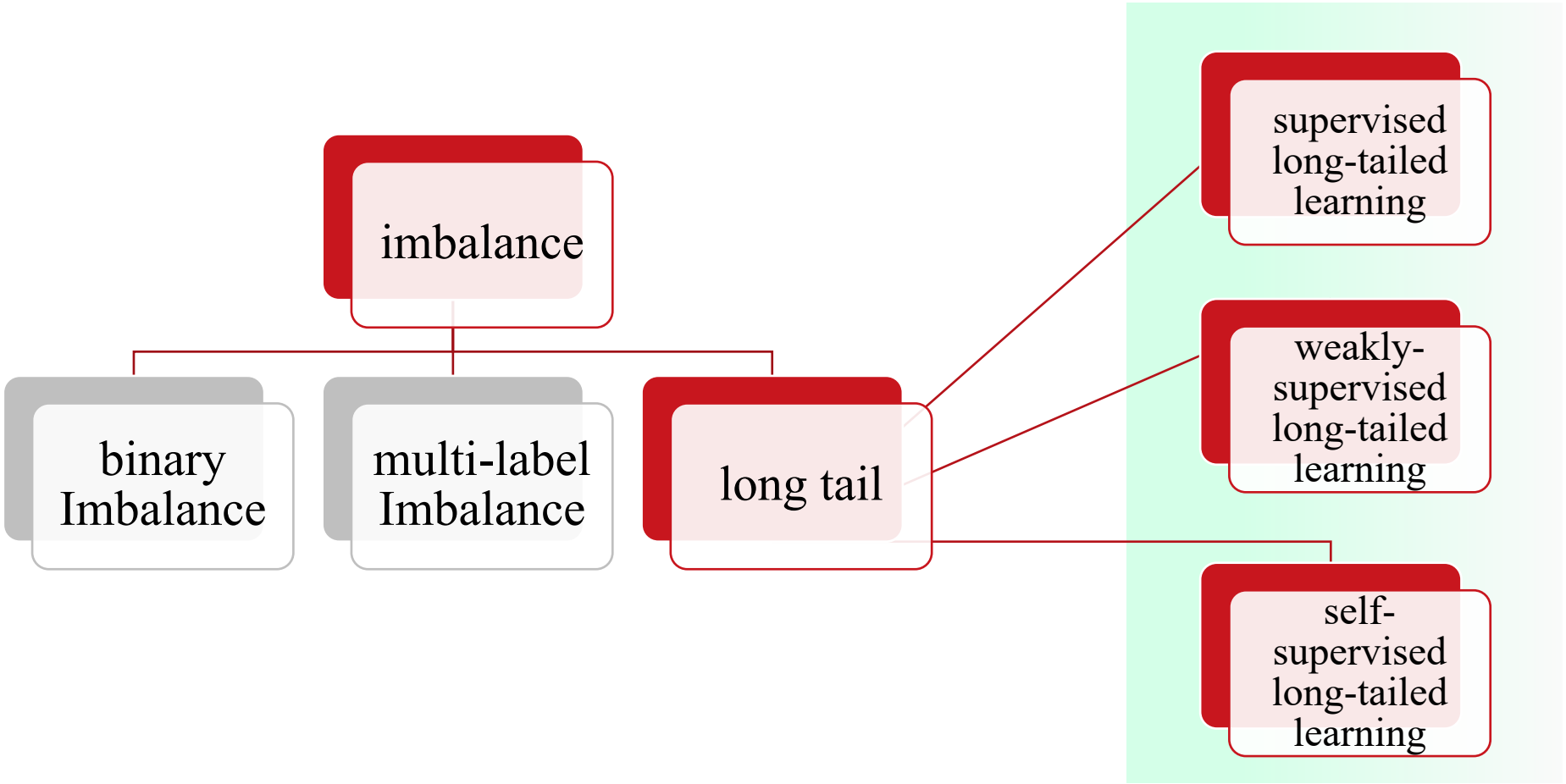
Summary of imbalanced learning in the early years



What is the new of this topic in the recent years?

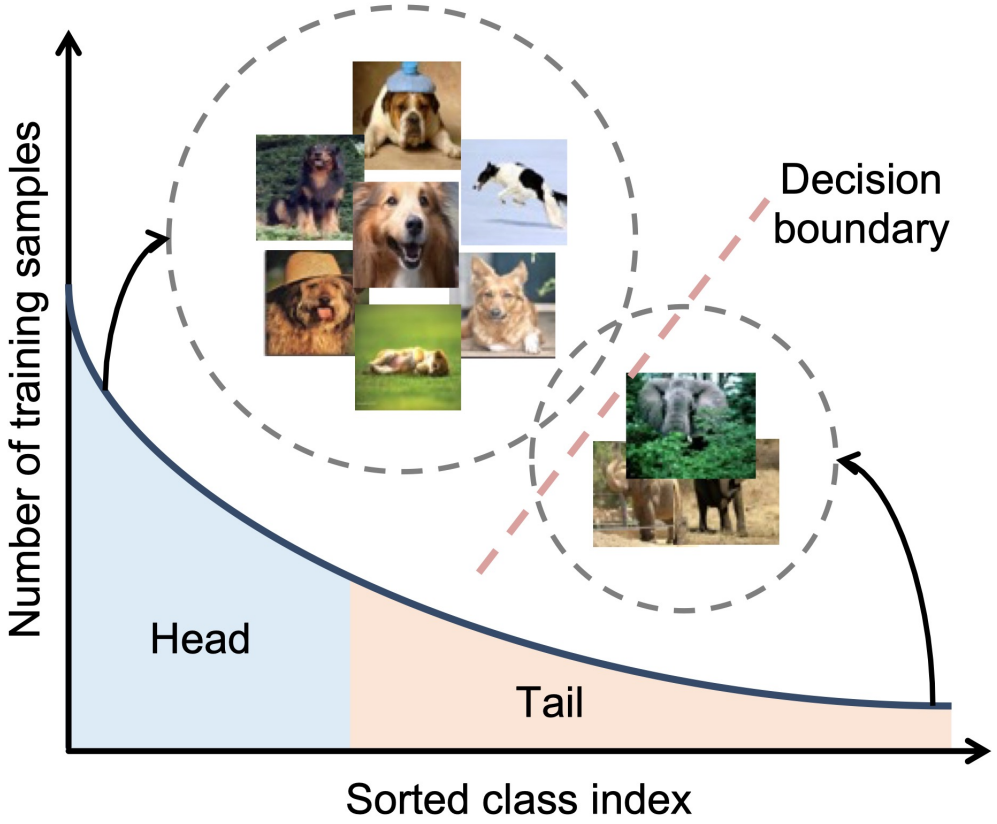
The Following Part in This Tutorial

The recent advances of imbalance learning powered by deep learning



Supervised Long-tailed Learning

It has been contributed with very broad explorations



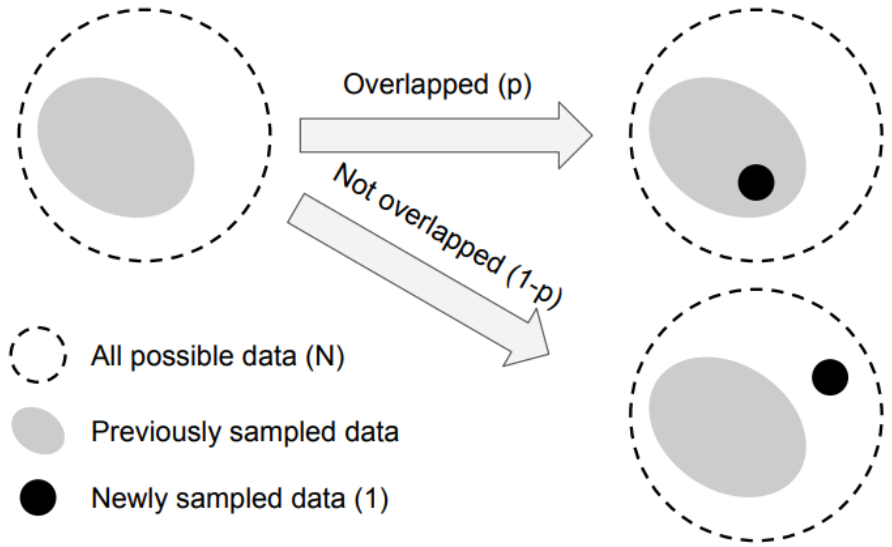
Method	Year	Class Re-balancing			Augmentation		Module Improvement				Target Aspect
		Re-sampling	CSL	LA	TL	Aug	RL	CD	DT	Ensemble	
LMLE [89]	2016						✓				feature
HFL [90]	2016						✓				feature
Focal loss [54]	2017		✓								objective
Range loss [21]	2017						✓				feature
CRL [50]	2017						✓				feature
MetaModelNet [91]	2017				✓						
DSTL [92]	2018				✓						
DCL [93]	2019	✓									sample
Meta-Weight-Net [94]	2019		✓								objective
LDAM [18]	2019		✓								objective
CB [16]	2019		✓								objective
UML [95]	2019		✓								feature
FTL [96]	2019				✓	✓					feature
Unequal-training [48]	2019						✓				feature
OLTR [15]	2019						✓				feature
Balanced Meta-Softmax [97]	2020	✓	✓								sample, objective
Decoupling [32]	2020	✓	✓				✓	✓	✓		feature, classifier
LST [98]	2020	✓			✓						sample
Domain adaptation [28]	2020		✓								objective
Equalization loss (ESQL) [19]	2020		✓								objective
DBM [22]	2020		✓								objective
Distribution-balanced loss [37]	2020		✓								objective
UNO-IC [99]	2020			✓							prediction
De-confound-TDE [45]	2020			✓				✓			prediction
M2m [100]	2020				✓	✓					sample
LEAP [49]	2020				✓	✓	✓				feature
OFA [101]	2020				✓	✓			✓		feature
SSP [102]	2020				✓		✓				feature
LFME [103]	2020				✓					✓	sample, model
IEM [104]	2020						✓				feature
Deep-RTC [105]	2020							✓			classifier
SimCal [34]	2020								✓	✓	sample, model
BBN [44]	2020									✓	sample, model
BAGS [56]	2020									✓	sample, model
VideoLT [38]	2021	✓									sample
LOCE [33]	2021	✓	✓								sample, objective
DARS [26]	2021	✓	✓		✓						sample, objective
CReST [106]	2021	✓			✓						sample
GIST [107]	2021	✓			✓			✓			classifier
FASA [58]	2021	✓				✓					feature
Equalization loss v2 [108]	2021		✓								objective
Seesaw loss [109]	2021		✓								objective
ACSL [110]	2021		✓								objective
IB [111]	2021		✓								objective
PML [51]	2021		✓								objective
VS [112]	2021		✓								objective
LADE [31]	2021		✓	✓							objective, prediction
RoBal [113]	2021		✓	✓				✓			objective, prediction
DisAlign [29]	2021		✓	✓					✓		objective, classifier
MiSLAS [114]	2021		✓			✓			✓		objective, feature, classifier
Logit adjustment [14]	2021			✓							prediction
Conceptual 12M [115]	2021				✓						
DiVE [116]	2021				✓						
MosaicOS [117]	2021				✓						
RSG [118]	2021				✓	✓					feature
SSD [119]	2021				✓				✓		
RIDE [17]	2021				✓					✓	model
MetaSAug [120]	2021					✓					sample
PaCo [121]	2021						✓				feature
DRO-LT [122]	2021						✓				feature
Unsupervised discovery [35]	2021						✓				feature
Hybrid [123]	2021						✓				feature
KCL [13]	2021						✓		✓		feature
DT2 [61]	2021								✓		feature, classifier
LTML [46]	2021									✓	sample, model
ACE [124]	2021									✓	sample, model
ResLT [125]	2021									✓	sample, model
SADE [30]	2021									✓	objective, model

[1] Zhang et al. "Deep Long-Tailed Learning: A Survey" TPAMI 2023.

Supervised Long-tailed Learning

loss re-weighting by effective number

➢ **Intuition:** Non-overlapping sample number, instead of the vanilla quantity number, playing the role of imbalance



➢ **Effective Number:** The effective number of examples is the expected volume of samples.

$$E_n = (1-\beta^n)/(1-\beta)$$

where $\beta = (N-1)/N$

$$\lim_{\beta \to 1} E_n = n$$

➢ **Class-Balanced Loss:** Training from imbalanced data by introducing a weighting factor that is inversely proportional to the effective number of samples.

The class-balanced loss term can be applied to a wide range of deep networks and loss functions.

[1] Cui et al. "Class-Balanced Loss Based on Effective Number of Samples" CVPR 2019.

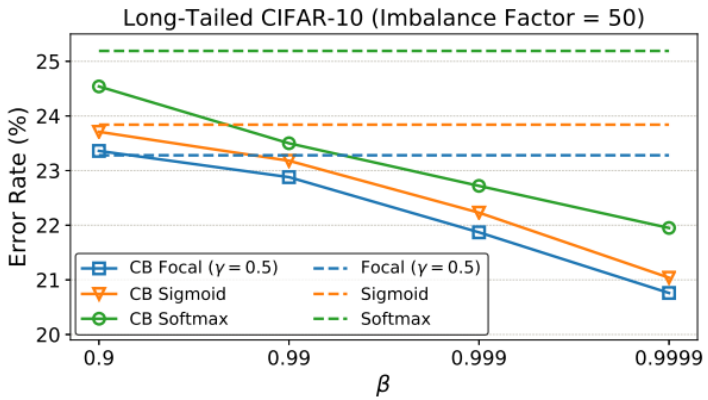
Supervised Long-tailed Learning

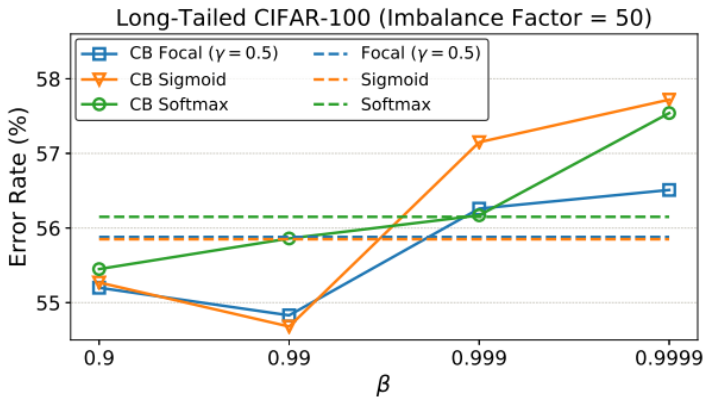
➢ **Class-Balanced Loss:** The class-balanced (CB) loss can be written as:

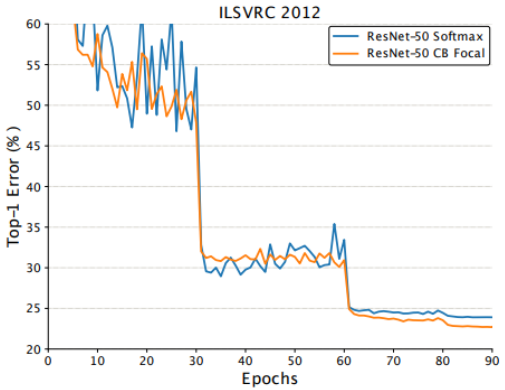
$$\mathrm{CB}(\mathbf{p}, y) = \frac{1}{E_{n_y}} \mathcal{L}(\mathbf{p}, y) = \frac{1-\beta}{1-\beta^{n_y}} \mathcal{L}(\mathbf{p}, y)$$

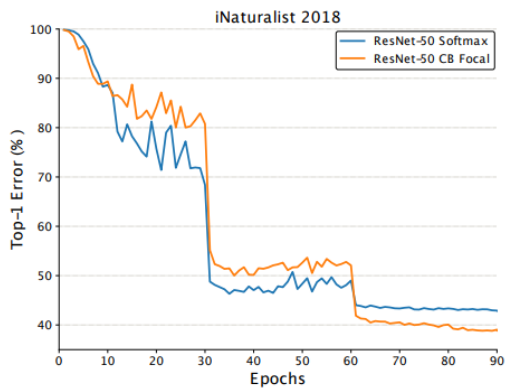
$$\mathrm{CB}_{\mathrm{softmax}}(\mathbf{z}, y) = -\frac{1-\beta}{1-\beta^{n_y}} \log\left(\frac{\exp(z_y)}{\sum_{j=1}^{C} \exp(z_j)}\right)$$

Class-Balanced loss can also combined with sigmoid cross-entropy loss, focal loss, etc.









Dataset Name	Long-Tailed CIFAR-10						Long-Tailed CIFAR-100					
Imbalance	200	100	50	20	10	1	200	100	50	20	10	1
Softmax	**34.32**	29.64	25.19	17.77	13.61	6.61	65.16	61.68	56.15	48.86	44.29	29.07
Sigmoid	34.51	**29.55**	23.84	**16.40**	**12.97**	**6.36**	64.39	**61.22**	55.85	48.57	44.73	**28.39**
Focal ($\gamma = 0.5$)	36.00	29.77	**23.28**	17.11	13.19	6.75	65.00	61.31	55.88	48.90	44.30	28.55
Focal ($\gamma = 1.0$)	34.71	29.62	23.29	17.24	13.34	6.60	**64.38**	61.59	**55.68**	**48.05**	**44.22**	28.85
Focal ($\gamma = 2.0$)	35.12	30.41	23.48	16.77	13.68	6.61	65.25	61.61	56.30	48.98	45.00	28.52
Class-Balanced	**31.11**	**25.43**	**20.73**	**15.64**	**12.51**	**6.36***	**63.77**	**60.40**	**54.68**	**47.41**	**42.01**	**28.39***
Loss Type	SM	Focal	Focal	SM	SGM	SGM	Focal	Focal	SGM	Focal	Focal	SGM
β	0.9999	0.9999	0.9999	0.9999	0.9999	-	0.9	0.9	0.99	0.99	0.999	-
γ	-	1.0	2.0	-	-	-	1.0	1.0	-	0.5	0.5	-

The proposed framework provides a non-parametric means of quantifying data overlap.

[1] Cui et al. "Class-Balanced Loss Based on Effective Number of Samples" CVPR 2019.

Class-wise margin calibration

- **Motivation:** Re-weighting and re-sampling often cause over-fitting. The authors propose to regularize the minority classes more strongly than the frequent classes to improve the generalization error of minority classes without sacrificing the model's ability to fit the frequent classes.

- **Class-distribution-aware margin trade-off:** Enforcing bigger margins can improve generalization, but may also hurt the margins of the frequent classes.

  The generalization error is proportional to the following:

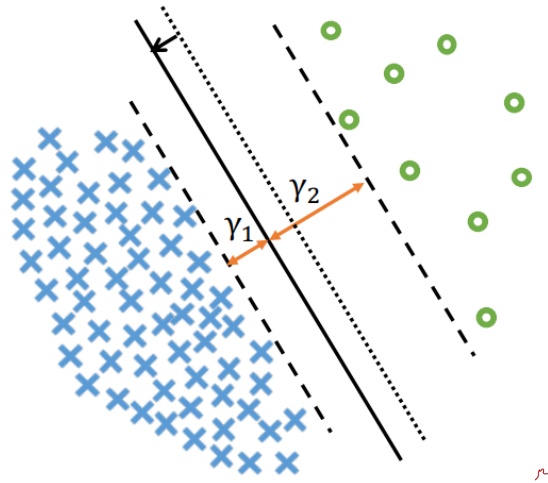
$$\frac{1}{\gamma_1\sqrt{n_1}} + \frac{1}{\gamma_2\sqrt{n_2}} \qquad \gamma_1 + \gamma_2 = \gamma$$

  For multi-class classification tasks, the margin of a sample is defined as:

$$\gamma(x, y) = f(x)_y - \max_{j \neq y} f(x)_j$$

The margin of a class is defined as the minimum value of the spacing between all its samples:

$$\gamma_j = \min_{i \in S_j} \gamma(x_i, y_i) \qquad \gamma_j = \frac{C}{n_j^{1/4}}$$



[1] Cao et al. "Learning Imbalanced Datasets with Label-Distribution-Aware Margin Loss" NeurIPS 2019.

Supervised Long-tailed Learning

➢ **LDAM:** The authors define their loss function as:

$$\mathcal{L}_{\text{LDAM-HG}}((x,y);f) = \max(\max_{j\neq y}\{z_j\} - z_y + \Delta_y, 0)$$

$$\text{where } \Delta_j = \frac{C}{n_j^{1/4}} \text{ for } j \in \{1,\dots,k\}$$

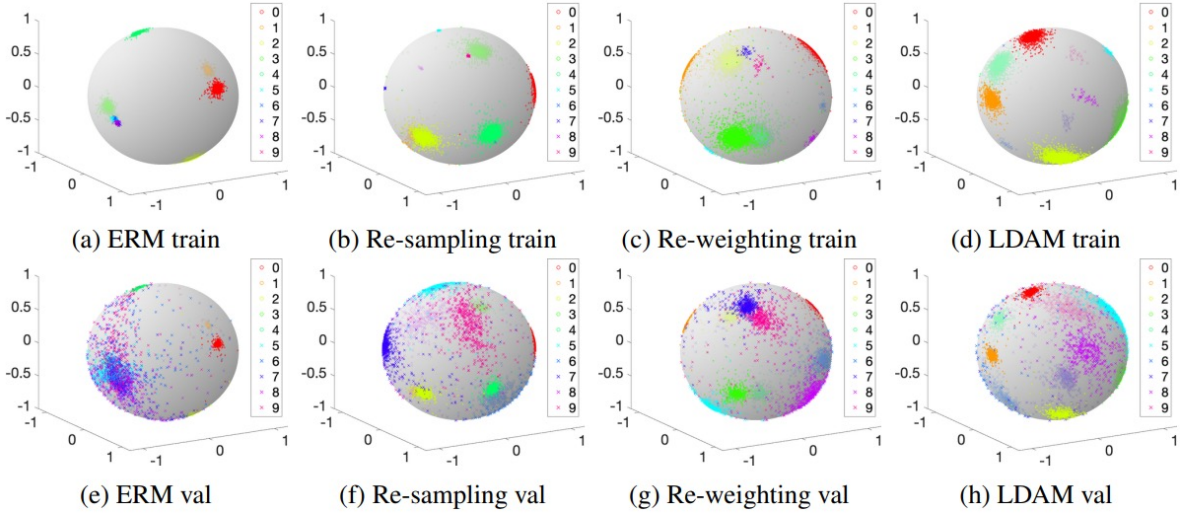
The smooth relaxation of the hinge loss is the following cross-entropy loss with enforced margins:

$$\mathcal{L}_{\text{LDAM}}((x,y);f) = -\log \frac{e^{z_y-\Delta_y}}{e^{z_y-\Delta_y} + \sum_{j\neq y} e^{z_j}}$$

$$\text{where } \Delta_j = \frac{C}{n_j^{1/4}} \text{ for } j \in \{1,\dots,k\}$$

Approach	Error on positive reviews	Error on negative reviews	Mean Error
ERM	2.86	70.78	36.82
RS	7.12	45.88	26.50
RW	5.20	42.12	23.66
LDAM-DRW	4.91	30.77	17.84

Dataset	Imbalanced CIFAR-10				Imbalanced CIFAR-100			
Imbalance Type	long-tailed		step		long-tailed		step	
Imbalance Ratio	100	10	100	10	100	10	100	10
ERM	29.64	13.61	36.70	17.50	61.68	44.30	61.45	45.37
Focal [Lin et al., 2017]	29.62	13.34	36.09	16.36	61.59	44.22	61.43	46.54
LDAM	26.65	13.04	33.42	15.00	60.40	43.09	60.42	43.73
CB RS	29.45	13.21	38.14	15.41	66.56	44.94	66.23	46.92
CB RW [Cui et al., 2019]	27.63	13.46	38.06	16.20	66.01	42.88	78.69	47.52
CB Focal [Cui et al., 2019]	25.43	12.90	39.73	16.54	63.98	42.01	80.24	49.98
HG-DRS	27.16	14.03	29.93	14.85	-	-	-	-
LDAM-HG-DRS	24.42	12.72	24.53	12.82	-	-	-	-
M-DRW	24.94	13.57	27.67	13.17	59.49	43.78	58.91	44.72
LDAM-DRW	**22.97**	**11.84**	**23.08**	**12.19**	**57.96**	**41.29**	**54.64**	**40.54**



(a) ERM train (b) Re-sampling train (c) Re-weighting train (d) LDAM train

(e) ERM val (f) Re-sampling val (g) Re-weighting val (h) LDAM val

[1] Cao et al. "Learning Imbalanced Datasets with Label-Distribution-Aware Margin Loss" NeurIPS 2019.

Class-wise logit adjustment

- **Motivation:** Design a consistent loss function that allows for a relatively elastic margin in the logit for head and tail.

- **Balanced error:** Under class imbalance, to measure balanced error:

$$\mathrm{BER}(f) \doteq \frac{1}{L}\sum_{y\in[L]} \mathbb{P}_{x|y}\left(y \notin \operatorname{argmax}_{y'\in\mathcal{Y}} f_{y'}(x)\right)$$

Under Bayes-optimal prediction, if $\mathbb{P}^{\mathrm{bal}}(y \mid x) \propto \mathbb{P}(y \mid x)/\mathbb{P}(y)$

Then

$$\operatorname{argmax}_{y\in[L]} \mathbb{P}^{\mathrm{bal}}(y \mid x) = \operatorname{argmax}_{y\in[L]} \exp(s^*_y(x))/\mathbb{P}(y) = \operatorname{argmax}_{y\in[L]} s^*_y(x) - \ln \mathbb{P}(y)$$

[1] Menon et al. ”Long-tailed Learning via Logit Adjustment" ICLR 2021.

Supervised Long-tailed Learning

➢ **The logit adjusted softmax cross-entropy**

$$\ell(y, f(x)) = -\log \frac{e^{f_y(x)+\tau\cdot\log\pi_y}}{\sum_{y'\in[L]} e^{f_{y'}(x)+\tau\cdot\log\pi_{y'}}} = \log\left[1+\sum\nolimits_{y'\neq y}\left(\frac{\pi_{y'}}{\pi_y}\right)^{\tau}\cdot e^{(f_{y'}(x)-f_y(x))}\right]$$

$$w_1^\top\Phi(x)/\pi_1 < w_2^\top\Phi(x)/\pi_2 \nRightarrow \exp(w_1^\top\Phi(x))/\pi_1 < \exp(w_2^\top\Phi(x))/\pi_2.$$

➢ **Post-hoc logit adjustment**

$$\operatorname{argmax}_{y\in[L]} \exp(w_y^\top\Phi(x))/\pi_y^\tau = \operatorname{argmax}_{y\in[L]} f_y(x) - \tau\cdot\log\pi_y$$

[1] Menon et al. ”Long-tailed Learning via Logit Adjustment" ICLR 2021.

➢ **An remarkable point on the statistical consistency of long-tailed multi-class classification**

$$\ell(y, f(x)) = \alpha_y \cdot \log\left[1 + \sum\nolimits_{y' \neq y} e^{\Delta_{yy'}} \cdot e^{(f_{y'}(x) - f_y(x))}\right]$$

Theorem 1. *For any $\delta \in \mathbb{R}_+^L$, the pairwise loss in (11) is Fisher consistent with weights and margins*

$$\alpha_y = \delta_y / \mathbb{P}(y) \qquad \Delta_{yy'} = \log\left(\delta_{y'} / \delta_y\right).$$

Letting $\delta_y = \pi_y$, we immediately deduce that the logit-adjusted loss of (10) is consistent, *provided* our π_y is a consistent estimate of $\mathbb{P}(y)$. Similarly, $\delta_y = 1$ recovers the classic result that the balanced loss is consistent. While Theorem 1 only provides a sufficient condition in multi-class setting, one can provide a necessary and sufficient condition that rules out other choices of Δ in the binary case.

[1] Menon et al. ”Long-tailed Learning via Logit Adjustment" ICLR 2021.

Dynamic adjustment based on a fine-grained generalization bound

Proposition 3 (Data-Dependent Bound for the VS Loss). *Given the function set $\mathcal{F}$ and the VS loss L_{VS}, for any $\delta \in (0,1)$, with probability at least $1-\delta$ over the training set $\mathcal{S}$, the following generalization bound holds for all $f \in \mathcal{F}$:*

$$\mathcal{R}_{bal}^{L}(f) \lesssim \Phi(L_{VS}, \delta) + \frac{\hat{\mathfrak{C}}_{\mathcal{S}}(\mathcal{F})}{C\pi_C} \sum_{y=1}^{C} \alpha_y \tilde{\beta}_y \sqrt{\pi_y} \left[1 - softmax\left(\beta_y B_y(f) + \Delta_y\right)\right].$$

$$L_{\text{VS}}(f(\boldsymbol{x}), y) = -\alpha_y \log\left(\frac{e^{\beta_y f(\boldsymbol{x})_y + \Delta_y}}{\sum_{y'} e^{\beta_{y'} f(\boldsymbol{x})_{y'} + \Delta_{y'}}}\right).$$

Algorithm 1: Principled Learning Algorithm induced by the Theoretical Insights

Require: Training set $\mathcal{S} = \{(x_i, y_i)\}_{i=1}^{N}$ and a model f parameterized by Θ.

Initialize the model parameters Θ randomly.
for $t = 1, 2, \cdots, T$ **do**
  $\mathcal{B} \leftarrow$ SampleMiniBatch$(\mathcal{S}, m)$ ▷ A mini-batch of m samples
  if $t < T_0$ **then**
    Set $\alpha = 1, \beta_y, \Delta_y$ ▷ Adjust logits during the initial phase
  else
    Set $\alpha_y \propto \pi_y^{-\nu}, \beta_y = 1, \Delta_y, \nu > 0$ ▷ TLA and ADRW
  end if
  $L(f, \mathcal{B}) \leftarrow \frac{1}{m} \sum_{(\boldsymbol{x},y)\in\mathcal{B}} L_{\text{VS}}(f(\boldsymbol{x}), y)$ ▷ Calculate the loss
  $\Theta \leftarrow \Theta - \eta \nabla_{\Theta} L(f, \mathcal{B})$ ▷ One SGD step
  Optional: anneal the learning rate η. ▷ Required when $t = T_0$
end for

[1] Wang et al. "A Unified Generalization Analysis of Re-Weighting and Logit-Adjustment for Imbalanced Learning" NeurIPS 2023.

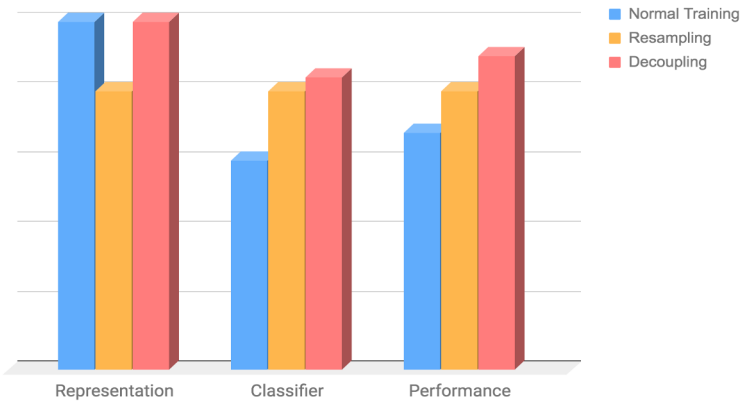
Supervised Long-tailed Learning

Both representation and classifier matter

➢ **Motivation:** Representation and classifier are in the different learning pace, and can be treated differently during training.

➢ **The problem behind long-tail:**

Classification performance =
Representation Quality + Classifier Quality



➢ **Ways of learning classifiers:**

Classifier Re-training (cRT)

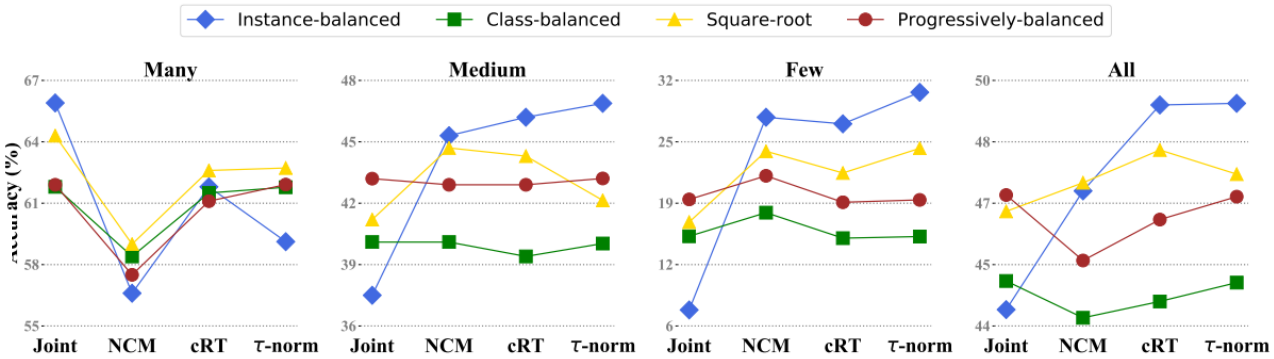
Nearest Class Mean classifier (NCM)

τ -normalized classifier (τ -normalized)

$$\widetilde{w_i} = \frac{w_i}{||w_i||^\tau}$$

Learnable weight scaling (LWS)

$$\widetilde{w_i} = f_i * w_i, \text{where } f_i = \frac{1}{||w_i||^\tau}$$



[1] Kang et al. " Decoupling Representation and Classifier for Long-Tailed Recognition " ICLR 2020.

Supervised Long-tailed Learning

The special constant classifier geometry

- **Motivation:** In the training on an imbalanced dataset, the classifier vectors of minor classes will be merged, termed as minority collapse, which breaks up the ETF structure and deteriorates the performance on test data.

- **Simplex Equiangular Tight Frame:**

$$\mathbf{M} = \sqrt{\frac{K}{K-1}}\mathbf{U}\left(\mathbf{I}_K - \frac{1}{K}\mathbf{1}_K\mathbf{1}_K^T\right)$$

$$\mathbf{m}_i^T\mathbf{m}_j = \frac{K}{K-1}\delta_{i,j} - \frac{1}{K-1}, \forall i,j \in [1,K]$$

- **ETF Classifier:**

$$\min_{\mathbf{H}} \quad \frac{1}{N}\sum_{k=1}^{K}\sum_{i=1}^{n_k}\mathcal{L}_{CE}(\mathbf{h}_{k,i}, \mathbf{W}^*)$$

$$\mathbf{w}_k^{*T}\mathbf{w}_{k'}^* = E_W\left(\frac{K}{K-1}\delta_{k,k'} - \frac{1}{K-1}\right), \forall k,k' \in [1,K]$$

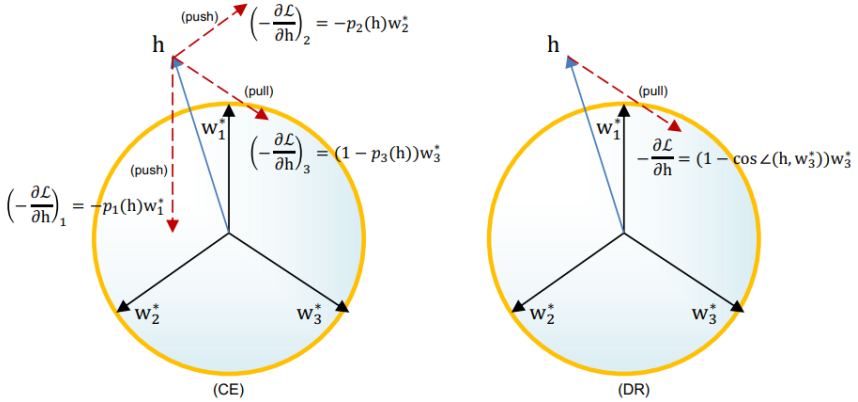
[1] Yang et al. "Inducing Neural Collapse in Imbalanced Learning: Do We Really Need a Learnable Classifier at the End of Deep Neural Network?" NeurIPS 2022.

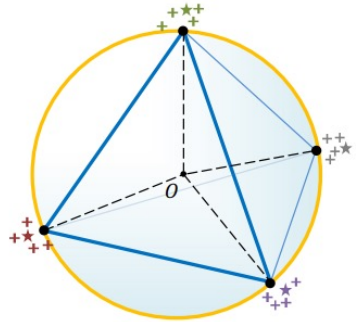
Supervised Long-tailed Learning

➢ **Dot-Regression Loss:**

Squared loss function:

$$\mathcal{L}_{DR}(\mathbf{h}, \mathbf{W}^*) = \frac{1}{2\sqrt{E_W E_H}} \left(\mathbf{w}_c^{*T} \mathbf{h} - \sqrt{E_W E_H} \right)^2 \qquad \frac{\partial \mathcal{L}_{DR}}{\partial \mathbf{h}} = -\left(1 - \cos \angle(\mathbf{h}, \mathbf{w}_c^*)\right) \mathbf{w}_c^*$$





Epoch	Methods	Acc. (%)
90	Learnable Classifier + CE	34.6
	ETF Classifier + DR	41.8
120	Learnable Classifier + CE	41.9
	ETF Classifier + DR	43.2
150	Learnable Classifier + CE	42.5
	ETF Classifier + DR	43.8
180	Learnable Classifier + CE	44.3
	ETF Classifier + DR	44.7

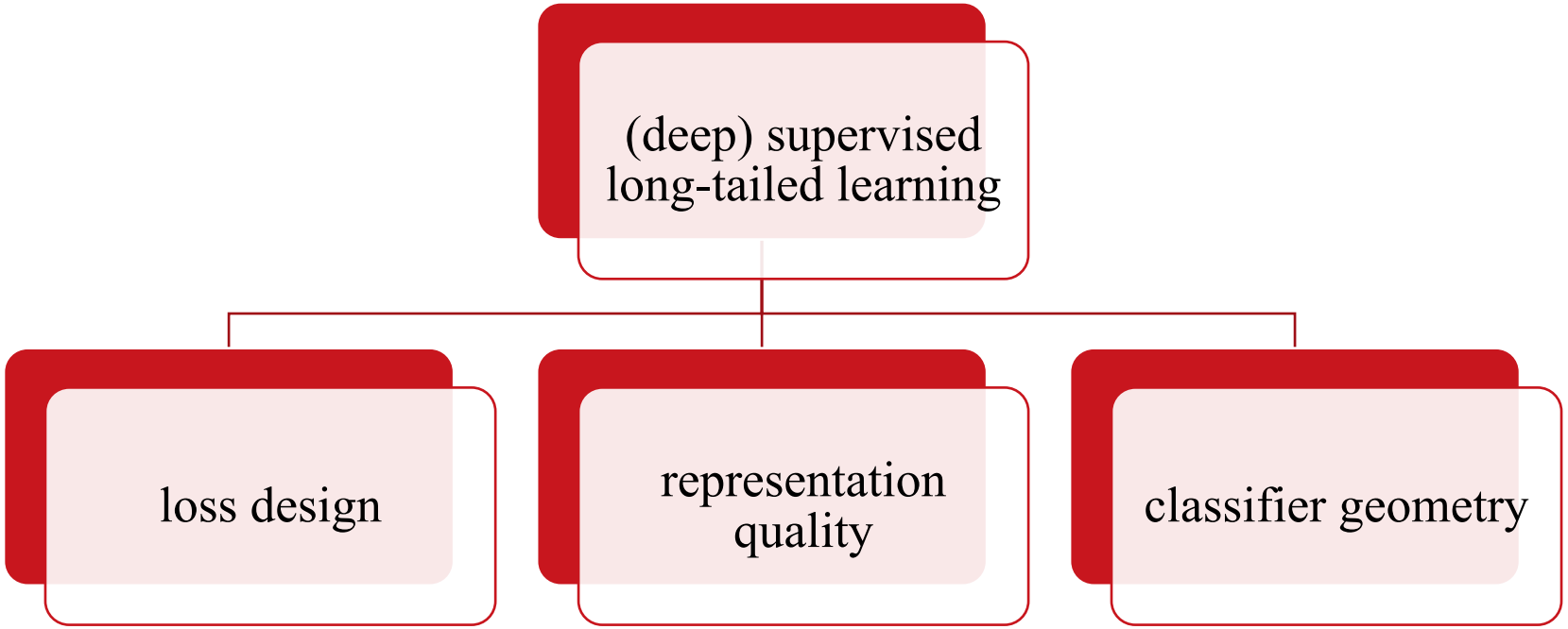
Backbone	Methods	Acc. (%)
ResNet-34	Learnable Classifier + CE	82.2
	ETF Classifier + DR	83.0
ResNet-50	Learnable Classifier + CE	85.5
	ETF Classifier + DR	86.1
ResNet-101	Learnable Classifier + CE	86.2
	ETF Classifier + DR	87.0

Methods	CIFAR-10 [17]			CIFAR-100 [17]			SVHN [25]			STL-10 [3]		
	0.005	0.01	0.02	0.005	0.01	0.02	0.005	0.01	0.02	0.005	0.01	0.02
ResNet												
Learnable Classifier + CE	67.3	72.8	78.6	38.7	43.0	48.1	40.5	40.9	49.3	33.1	37.9	38.8
ETF Classifier + DR	71.9	76.5	81.0	40.9	45.3	50.4	42.8	45.7	49.8	33.5	37.2	37.9
Improvements	**+4.6**	**+3.7**	**+2.4**	**+2.2**	**+2.3**	**+2.3**	**+2.3**	**+4.8**	**+0.5**	**+0.4**	-0.7	-0.9
DenseNet												
Learnable Classifier + CE	71.1	77.7	84.1	40.3	43.8	49.8	39.7	40.5	46.4	38.5	41.2	44.9
ETF Classifier + DR	72.9	78.5	83.4	40.1	44.0	49.7	40.5	44.8	48.4	39.5	42.9	46.3
Improvements	**+1.8**	**+0.8**	-0.7	-0.2	**+0.2**	-0.1	**+0.8**	**+4.3**	**+2.0**	**+1.0**	**+1.7**	**+1.4**

[1] Yang et al. "Inducing Neural Collapse in Imbalanced Learning: Do We Really Need a Learnable Classifier at the End of Deep Neural Network?" NeurIPS 2022.

Supervised Long-tailed Learning

Summary: supervised long-tailed learning in the context of deep learning, should take more factors (loss, representation etc.) into account to improve the performance of imbalanced learning.



Weakly-Supervised Long-tailed Learning

Weak supervision

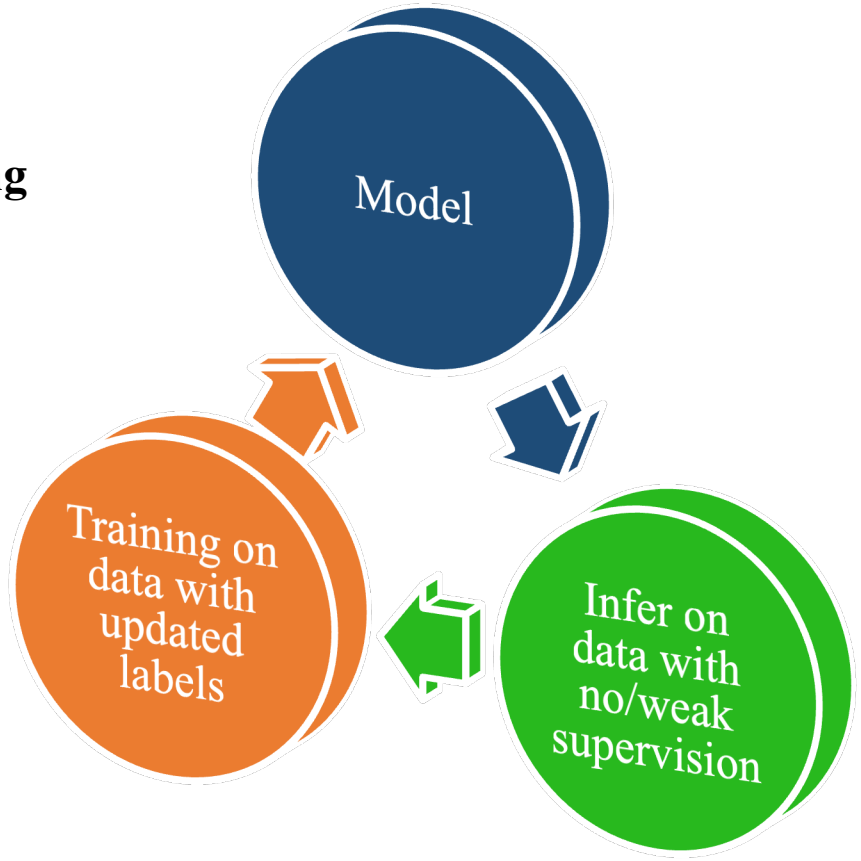
- Incomplete supervision: **semi-supervised learning**
- Inexact supervision: **partial label learning**, **multi-instance learning**
- Inaccurate supervision: **label-noise learning**

Target: *low labeling cost* but *high accuracy*.

Pseudo Labeling

- Generate more precise annotations for weakly supervised data

Transfer weakly-supervised learning → supervised learning

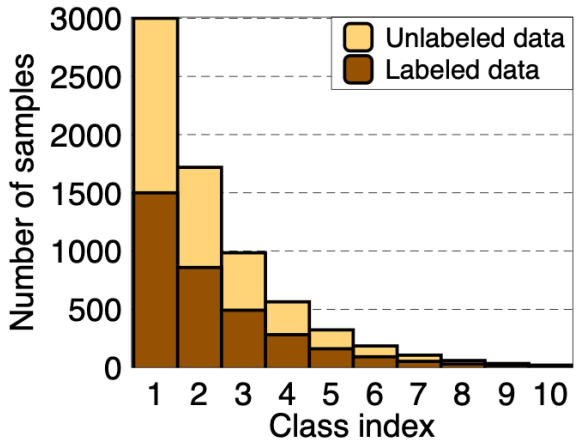


Pseudo Labeling

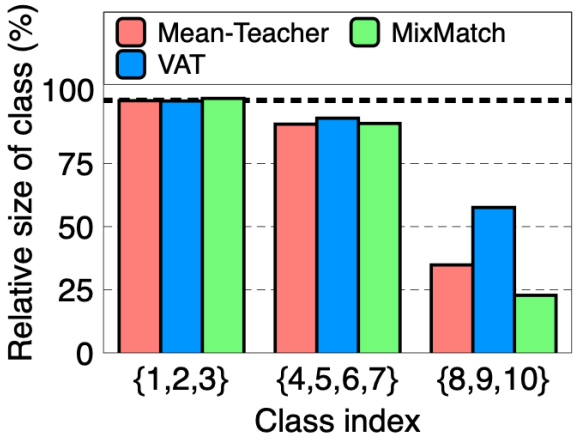
Zhou, Z. H. (2018). A brief introduction to weakly supervised learning. *National science review*, *5*(1), 44-53.

When long-tailed distribution meets weak supervision

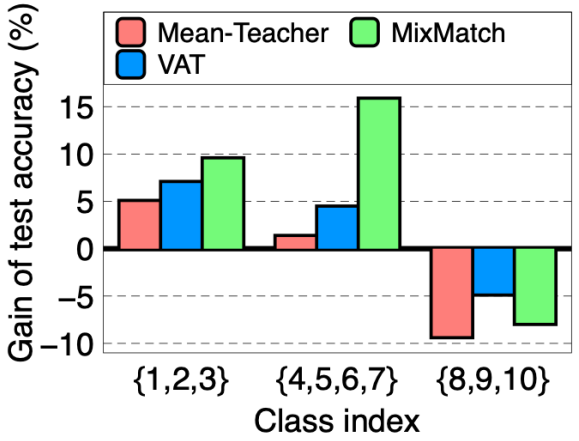
- Positive feedback property of pseudo-labeling: head get better and more



(a) Imbalanced class distribution



(b) Bias of pseudo-labels



(c) Accuracy gain from SSL

[1]

- Distributional corrections need to go deeper into training and pseudo-labeling

[1]Kim, J., Hur, Y., Park, S., et al. Distribution aligning refinery of pseudo-label for imbalanced semi-supervised learning. *NeurIPS*. 2020.

Long-tailed Semi-Supervised Learning

- Motivation: Label propagation on unlabeled data biased toward the majority classes.
- Main idea: Refine the original, biased pseudo-labels so that their distribution can match the true class distribution of unlabeled data, while constrain the refined pseudo-labels to be close from the original ones.
- Formulation: An optimization problem for constructing refined pseudo-labels: minimizing the distortion from the original pseudo-labels, while matching the true class distribution.

$$\text{minimize} \quad \sum_{m=1}^{M} w_m D_{KL}(\hat{y}_m \,||\, \hat{y}_m^{\texttt{unlabeled}})$$

$$\text{subject to} \quad \sum_{m=1}^{M} \hat{y}_m(k) = M_k,\ \forall k,\ \sum_{k=1}^{K} \hat{y}_m(k) = 1,\ \forall m,\ \hat{y}_m(k) \in [0,1],\ \forall m,k$$

Algorithm 1 `DualCoordinateAscent`: Coordinate ascent algorithm for dual of (1)

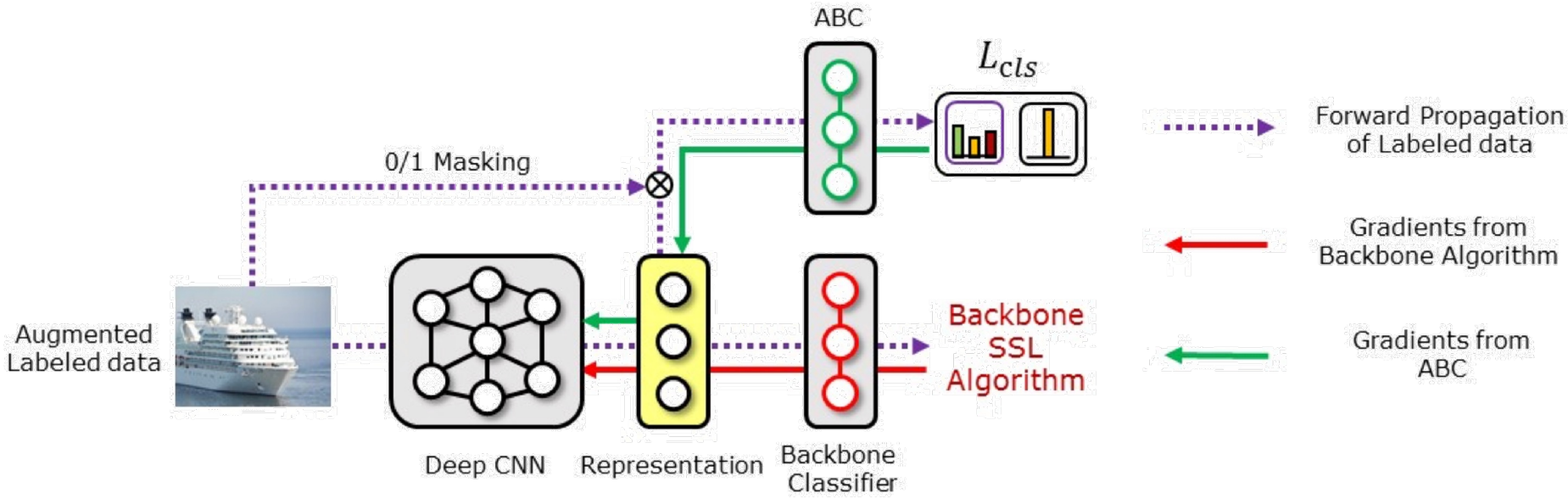
Require: $\{\hat{y}_m^{\texttt{unlabeled}}\}_{m=1}^M, \{w_m\}_{m=1}^M, \{M_k\}_{k=1}^K, T$
Ensure: The unique solution of (1)

$\hat{y}_m^0 \leftarrow y_m^{\texttt{unlabeled}},\ \alpha_m^0 \leftarrow 1,\ \beta_k^0 \leftarrow 1, \quad \forall m,k$
for $t = 1$ **to** T **do**
  if t is odd or $t = T$ **then**
    $\beta_k^t \leftarrow \beta_k^{t-1}, \quad \alpha_m^t \leftarrow \left(\sum_{k=1}^K \hat{y}_m^0(k)(\beta_k^{t-1})^{\frac{1}{w_m}}\right)^{-1}, \quad \forall m,k,$
  else
    $\alpha_m^t \leftarrow \alpha_m^{t-1}, \quad \beta_k^t \leftarrow \texttt{Solve}_{Z\geq 0}\left(\sum_{m=1}^M \hat{y}_m^0(k)\alpha_m^{t-1} Z^{\frac{1}{w_m}} - M_k\right), \quad \forall m,k$
  end if
end for
$\hat{y}_m^{\texttt{out}}(k) \leftarrow \hat{y}_m^0(k)\alpha_m^T(\beta_k^T)^{\frac{1}{w_m}}, \quad \forall m,k$

Kim, J., Hur, Y., Park, S., et al. Distribution aligning refinery of pseudo-label for imbalanced semi-supervised learning. *NeurIPS*, 2020.

ABC: Auxiliary Balanced Classifier [1]

- Motivation: high-quality representations can be learned even the classifier is biased [2].
- Train an Auxiliary Balanced Classifier (ABC) by resampling a balanced subset while using the high-quality representations learned from all data.



[1] Lee, H., Shin, S., & Kim, H.. Abc: Auxiliary balanced classifier for class-imbalanced semi-supervised learning. *NeurIPS*. 2021.
[2] Kang, B., Xie, S., Rohrbach, M., et al. Decoupling Representation and Classifier for Long-Tailed Recognition. *ICLR*. 2020.

Long-tailed Partial Label Learning

- Dataset $\mathcal{X} = \{(x_i, y_i, S_i): i \in (1, 2, 3, \dots, N)\}$, where any x_i is associated with a candidate label set S_i and its ground truth $y_i \in S_i$ is invisible.

Parker: Morning, Flash.
Thompson: Good morning, Parker.

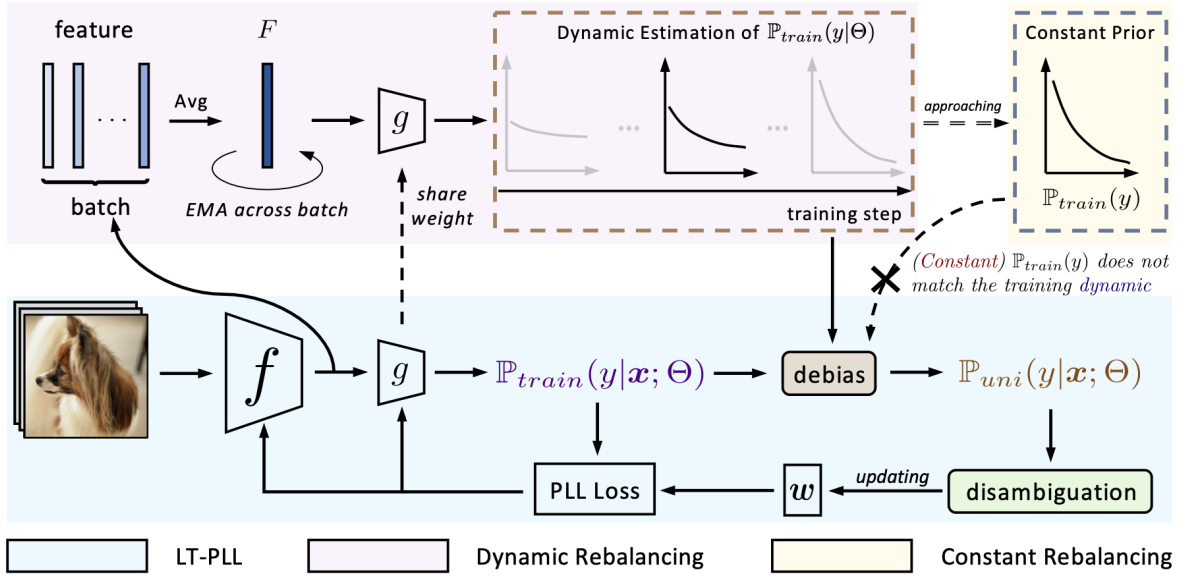
The History of LeBron James and Stephen Curry's Rivalrous Friendship

Annotator 1: Korat
Annotator 2: Russian Blue

- The sample number of L classes in descending order
  - $N_1 \geq N_2 \geq \cdots \geq N_L$, Imbalance ratio $\gamma = \frac{N_1}{N_L} \gg 1$
- Main challenges:
  - Tail samples cannot be correctly recognized even in training
  - No available class prior

RECORDS: Rebalancing for dynamic bias

- Observation: even after applying an oracle class distribution prior in the training, existing long-tailed techniques underperform in LT-PLL and even fail in some cases.
  - Existing long-tailed techniques: leverage a constant class distribution prior to rebalance the training and does not consider the dynamic of label disambiguation.
- A dynamic rebalancing mechanism friendly to the training dynamic



Constant Rebalancing

$$z^y_{uni}(\boldsymbol{x}) = z^y(\boldsymbol{x}) - \log \mathbb{P}_{train}(y|\Theta) = z^y(\boldsymbol{x}) - \log \mathrm{softmax}(g^y(F;\boldsymbol{W})).$$

Match the training dynamic

Dynamic Rebalancing

$$z^y_{uni}(\boldsymbol{x}) = z^y(\boldsymbol{x}) - \log \mathbb{P}_{train}(y|\Theta) = z^y(\boldsymbol{x}) - \log \mathrm{softmax}(g^y(F;\boldsymbol{W})).$$

Hong, F., Yao, J., Zhou, Z., Zhang, Y., & Wang, Y. Long-Tailed Partial Label Learning via Dynamic Rebalancing. *ICLR*. 2023.

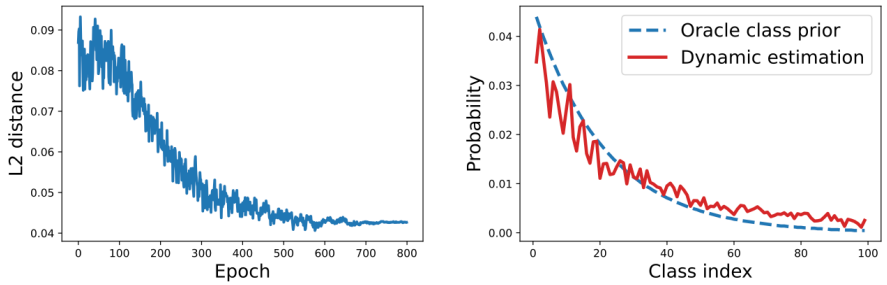
RECORDS: Rebalancing for dynamic bias

➢ Theory: alongside the label disambiguation, the dynamic estimation can progressively approach to the oracle class distribution.

Proposition 1. *Let* $\eta = \sup_{(\boldsymbol{x},y)\in\mathcal{X}\times\mathcal{Y},y_j\in\mathcal{Y},y_j\neq y}\mathbb{P}_{S|(\boldsymbol{x},y)}(y_j \in S)$ *denote the ambiguity degree,* d_H *be the Natarajan dimension of the hypothesis space* H*,* $\tilde{h} = h_{\tilde{\Theta}}$ *be the optimal classifier on the basis of the label disambiguation, where* $\tilde{\Theta} = \arg\min_{\Theta} R^{\mathcal{D}_{train}}(\Theta)$*. If the small ambiguity degree condition (Cour et al., 2011a; Liu & Dietterich, 2014)) satisfies, namely,* $\eta \in [0, 1)$*, then for* $\forall\delta > 0$*, the* L_2 *distance between* $\mathbb{P}_{train}(y)$ *and* $\mathbb{P}_{train}(y|\tilde{\Theta})$ *given* $\tilde{h}$ *is bounded as*

$$L_2\left(\tilde{h}\right) < \frac{4}{(\ln 2 - \ln(1+\eta))N}(d_H(\ln 2N + 2\ln C) - \ln\delta + \ln 2)$$

with probability at least $1-\delta$*, where* N *is the sample number and* C *is the category number.*



(a) L_2 distance during training (b) Estimated class distribution

	CIFAR-10-LT						CIFAR-100-LT						PASCAL VOC
Imbalance ratio ρ	50			100			50			100			
Ambiguity q	0.3	0.5	0.7	0.3	0.5	0.7	0.03	0.05	0.07	0.03	0.05	0.07	
LW	71.31	39.69	25.23	66.08	41.2	26.59	37.34	31.59	30.6	33.78	30.69	27.91	20.73
CAVL	57.77	41.68	21.32	54.28	39.47	20.67	23.01	21.29	14.23	31.65	28.77	13.65	18.33
PRODEN	74.22	56.25	45.11	65.72	49.23	42.26	43.02	40.15	39.1	38.66	36.23	35.52	24.47
CORR	77.53	58.35	45.66	69.12	50.96	42.87	45.95	42.83	41.72	41.79	38.47	36.96	26.72
SoLar	83.88	76.55	54.61	75.38	70.63	53.15	47.93	46.85	45.1	42.51	41.71	39.15	56.49
LW+RECORDS	77.22	59.88	43.98	72.2	59.87	43.63	38.16	33.6	31.53	35.33	31.55	28.8	55.19
CAVL+RECORDS	68.93	63.59	44.61	66.65	59.62	41.88	46.13	41.93	34.01	41.33	35.79	31.29	55.03
PRODEN+RECORDS	81.23	78.82	68.03	73.92	66.21	55.73	48.12	46.01	44.19	42.98	41.03	40.25	54.78
CORR+RECORDS	**84.25**	**82.5**	**71.24**	**79.79**	**74.07**	**62.25**	**52.08**	**50.58**	**47.91**	**46.57**	**45.22**	**44.73**	**58.45**

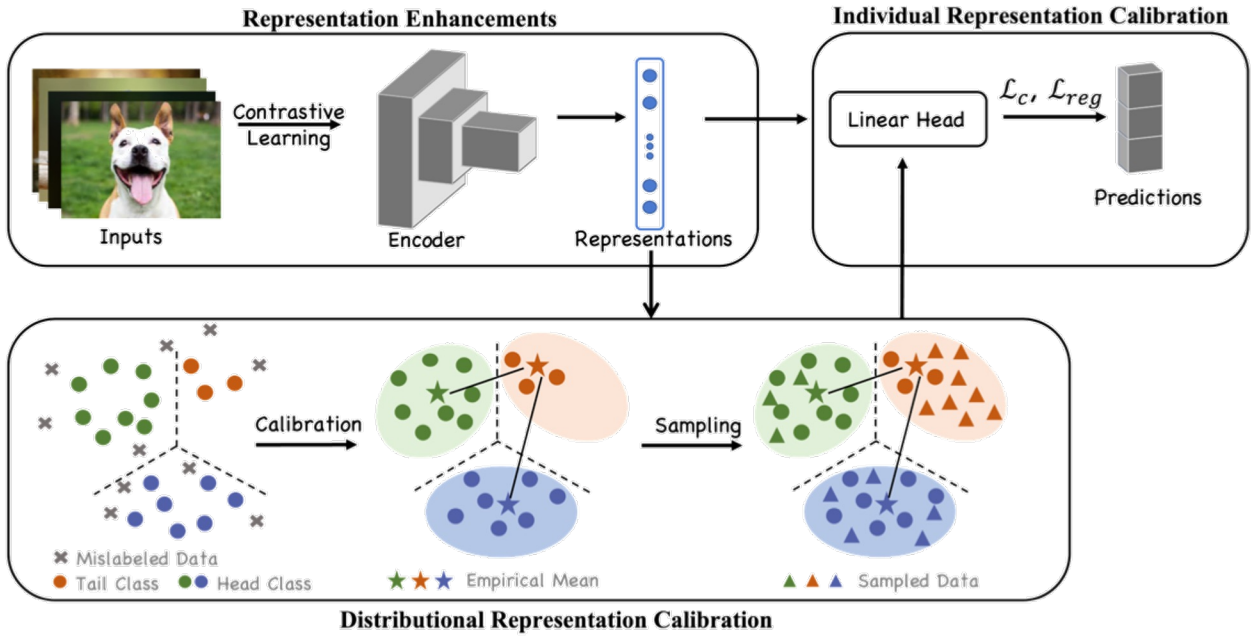
Hong, F., Yao, J., Zhou, Z., Zhang, Y., & Wang, Y. Long-Tailed Partial Label Learning via Dynamic Rebalancing. *ICLR*. 2023.

RCAL: Representation Calibration

- Motivation: The representations of unsupervised contrastive learning are not influenced by corrupted labels and thus naturally robust.

- Main spirit: Decoupling the imbalance from the noise labels with the help of self-supervised representation learning methods.

- Based upon the achieved representations, the calibration can be performed. Distributional representation calibration estimate the robust feature distribution considering data imbalance and perform balanced sampling. Individual representation calibration constrain the distance between supervised training representations and pre-training representations.



Zhang, M., Zhao, X., Yao, J., et al. When Noisy Labels Meet Long Tail Dilemmas: A Representation Calibration Method. *ICCV*. 2023.

RCAL: Representation Calibration

Algorithm 1 Algorithm of the proposed method RCAL

Require: the training dataset $\tilde{\mathcal{S}} = \{(\boldsymbol{x}_i, \tilde{y}_i)\}_{i=1}^n$, regularization strength β, scalar temperature τ, confidence weight γ, the pre-training epochs T_p, max epochs T_m.

1: **for** $t = 1, ..., T_p$ **do**
2: **Pre-train** the encoder network f with MoCo [20].
3: **end for**
4: **Extract** deep representations of instances with $\boldsymbol{z} = f(\boldsymbol{x})$.
5: **for** $c = 1, ..., K$ **do**
6: **Perform** the LOF algorithm for the c-th class and obtain preserved examples $\tilde{\mathcal{S}}_c'$.
7: **Build** the multivariate Gaussian distribution $\mathcal{N}(f(\boldsymbol{x})|\hat{\boldsymbol{\mu}}_c, \hat{\boldsymbol{\Sigma}}_c)$ for c-th class using $\tilde{\mathcal{S}}_c'$.
8: **end for**
9: **Calibrate** the multivariate Gaussian distributions of tail classes with the statistics of head classes.
10: **Sample** data points from achieved multivariate Gaussian distributions of all classes.
11: **for** $t = T_p + 1, ..., T_m$ **do**
12: **Add** distance constraints between learned representations and representations brought by contrastive learning.
13: **Adopt** the mixup technology to original examples.
14: **Train** the encoder f and the linear head h simultaneously on the training dataset and sample data points with the training loss in Eq. (2).
15: **end for**
16: **return** The robust classifier $h(f(\boldsymbol{x}))$ for testing.

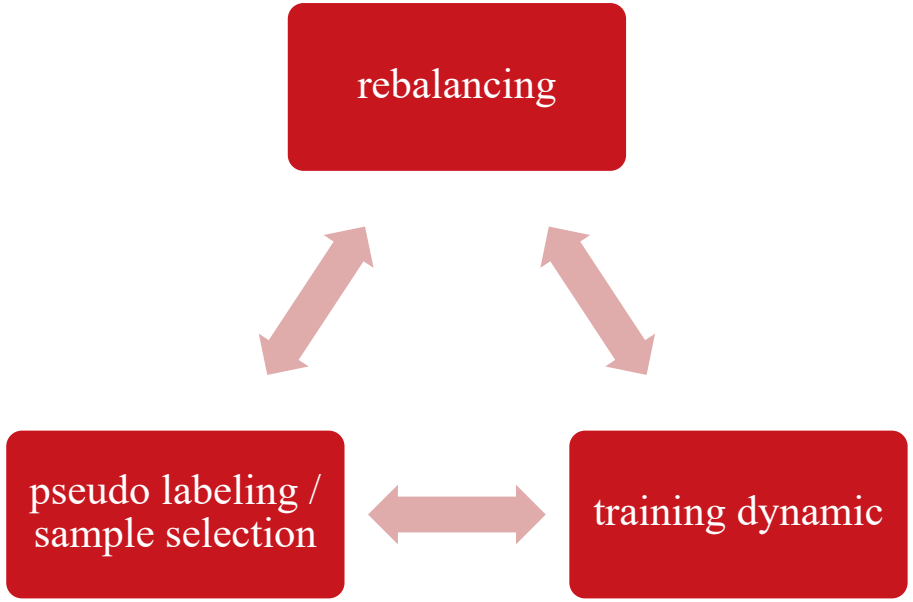
Dataset	Imbalance Ratio	10					100				
	Noise Rate	0.1	0.2	0.3	0.4	0.5	0.1	0.2	0.3	0.4	0.5
CIFAR-10	ERM	80.41	75.61	71.94	70.13	63.25	64.41	62.17	52.94	48.11	38.71
	ERM-DRW	81.72	77.61	71.94	70.13	63.25	66.74	62.17	52.94	48.11	38.71
	LDAM	84.59	82.37	77.48	71.41	60.30	71.46	66.26	58.34	46.64	36.66
	LDAM-DRW	85.94	83.73	80.20	74.87	67.93	76.58	72.28	66.68	57.51	43.23
	CRT	80.22	76.15	74.17	70.05	64.15	61.54	59.52	54.05	50.12	36.73
	NCM	82.33	74.73	74.76	68.43	64.82	68.09	66.25	60.91	55.47	42.61
	MiSLAS	87.58	85.21	83.39	76.16	72.46	75.62	71.48	67.90	62.04	54.54
	Co-teaching	80.30	78.54	68.71	57.10	46.77	55.58	50.29	38.01	30.75	22.85
	CDR	81.68	78.09	73.86	68.12	62.24	60.47	55.34	46.32	42.51	32.44
	Sel-CL+	86.47	85.11	84.41	80.35	77.27	72.31	71.02	65.70	61.37	56.21
	HAR-DRW	84.09	82.43	80.41	77.43	67.39	70.81	67.88	48.59	54.23	42.80
	RoLT	85.68	85.43	83.50	80.92	78.96	73.02	71.20	66.53	57.86	48.98
	RoLT-DRW	86.24	85.49	84.11	81.99	80.05	76.22	74.92	71.08	63.61	55.06
	RCAL (Ours)	88.09	86.46	84.58	83.43	80.80	78.60	75.81	72.76	69.78	65.05

Dataset	CIFAR-10				CIFAR-100			
Imbalance Ratio	10		100		10		100	
Noise Rate	0.2	0.4	0.2	0.4	0.2	0.4	0.2	0.4
RCAL	**86.46**	**83.43**	**75.81**	**69.78**	**54.85**	**48.91**	**39.85**	**33.36**
RCAL w/o Mixup	84.08	79.27	72.47	64.83	51.22	45.53	36.78	30.85
RCAL w/o Mixup, REG	83.23	78.12	67.49	58.27	48.74	42.15	34 31	27.14
RCAL w/o Mixup, REG, DC	80.40	74.37	64.02	54.61	47.01	40.85	32.27	25.42
RCAL w/o Mixup, REG, DC, CL	75.61	70.13	62.17	48.11	43.27	32.94	26.21	17.91

Zhang, M., Zhao, X., Yao, J., et al. When Noisy Labels Meet Long Tail Dilemmas: A Representation Calibration Method. *ICCV*. 2023.

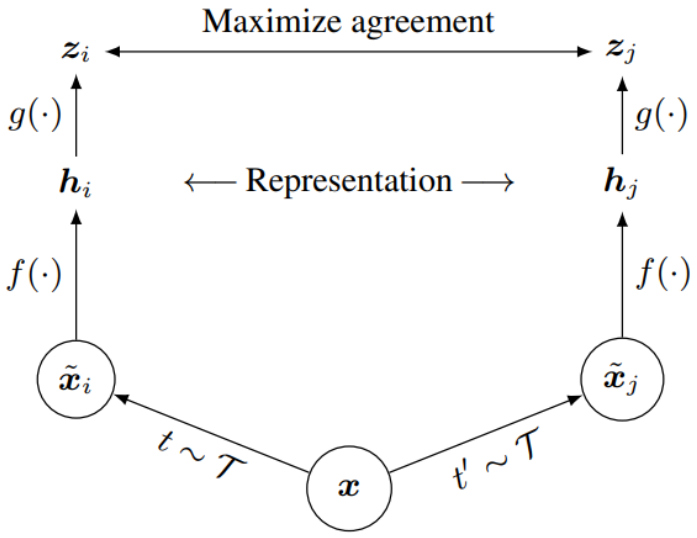
Summary

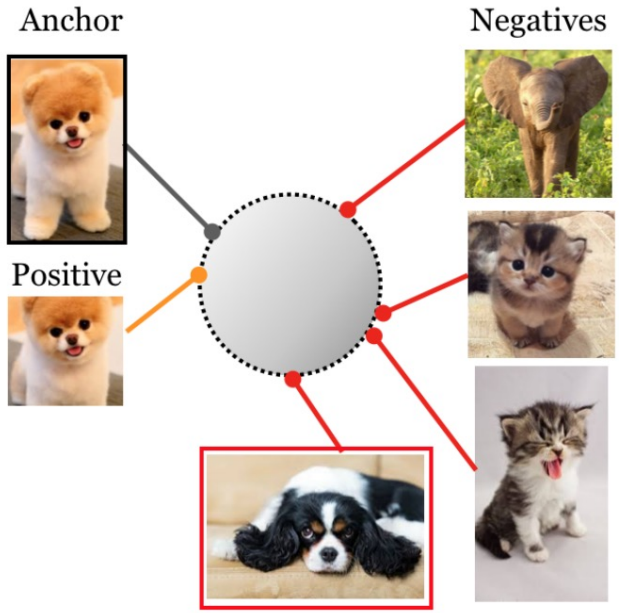
- Weak Supervision can exacerbate the long-tail effect, impairing the performance of minority classes.
- It is critical to construct a friendly mechanism to decouple *pseudo labeling / sample selection* and *rebalancing*.



Self-supervised learning

- The idea of contrastive learning: push augmented views of the same image closer, and embeddings to random pairs of images far apart

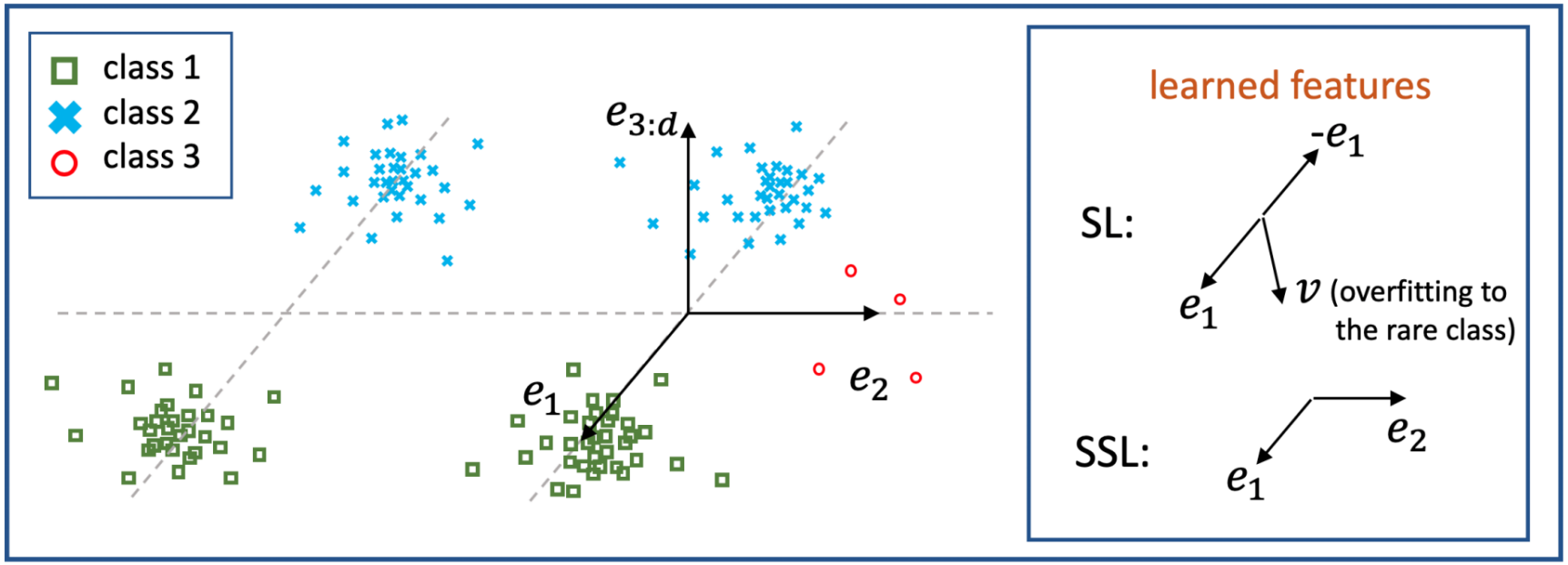




[1] Chen et al. "A simple framework for contrastive learning of visual representations." ICML 2020.
[2] Khosla et al. "Supervised contrastive learning." NeurIPS 2020.

Self-Supervised Long-tailed Learning

Is self-supervised learning more robust to data imbalance?



- Supervised learning (SL) only extracts features that are useful for predicting labels (e_1)

- Self-supervised learning (SSL) learns task-irrelevant features regardless of the labels, which enables richer and more robust representation (e_1, e_2)

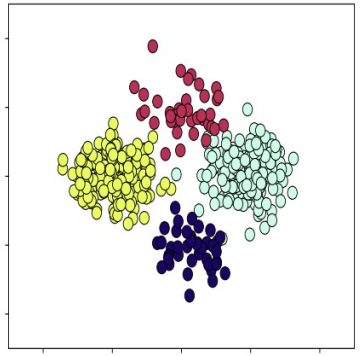
[1] Liu et al. "Self-supervised Learning is More Robust to Dataset Imbalance." ICLR 2021.

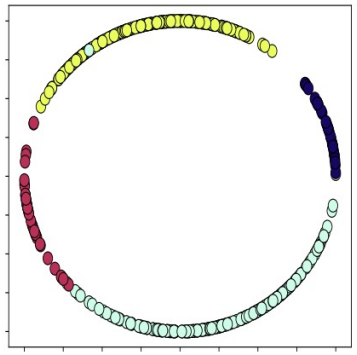
Self-supervised learning still suffers from data imbalance

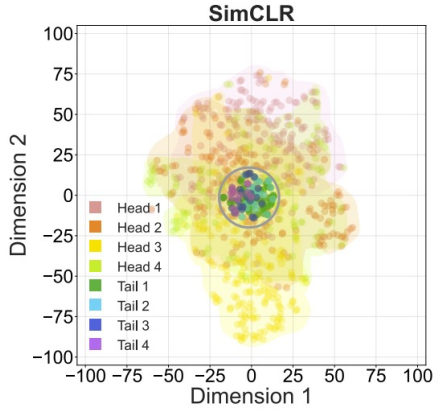
- Performance degeneration: Linear probing on imbalanced data (D_i) and balanced data (D_b) with same data amount

Dataset	Subset	*Many*	*Medium*	*Few*	All
CIFAR10	D_b	77.14 ± 4.64	74.25 ± 6.54	71.47 ± 7.55	74.57 ± 0.65
	D_i	76.07 ± 3.88	67.97 ± 5.84	54.21 ± 10.24	67.08 ± 2.15
CIFAR100	D_b	25.48 ± 1.74	25.16 ± 3.07	24.01 ± 1.23	24.89 ± 0.99
	D_i	30.72 ± 2.01	21.93 ± 2.61	15.99 ± 1.51	22.96 ± 0.43

- Representation learning disparity: head classes dominate the feature regime but tail classes passively collapse







[1] Jiang et al. "Self-damaging contrastive learning." ICML 2021.

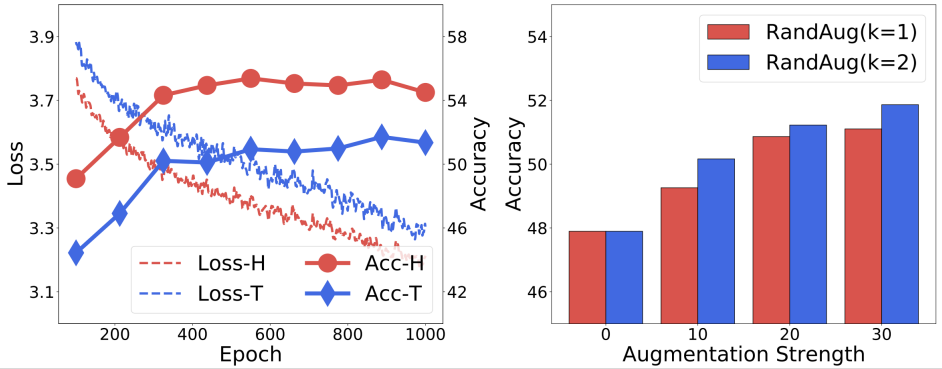
[2] Zhou et al. "Combating Representation Learning Disparity with Geometric Harmonization." NeurIPS 2023.

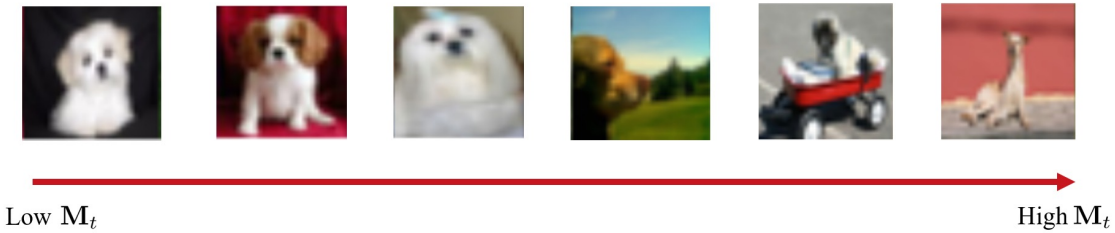
Data-based Self-Supervised Long-tailed Learning

BCL: Boosted Contrastive Learning

- Motivation I: Memorization effect still holds under long-tailed distribution.
- Motivation II: Stronger information discrepancy motivates tail samples mining

- **Challenge:** how to **detect tail data** and how to construct the **desired information discrepancy**





- Motivated from the observation that *learning speed-based proxy* shows strong correlation with the memorization score[1], BCL extends the memorization estimation to *self-supervised learning*.

$$\mathcal{L}^m_{i,0} = \mathcal{L}_{i,0}, \quad \mathcal{L}^m_{i,t} = \beta \mathcal{L}^m_{i,t-1} + (1-\beta)\mathcal{L}_{i,t} \qquad \mathbf{M}_{i,t} = \frac{1}{2}\left(\frac{\mathcal{L}^m_{i,t} - \bar{\mathcal{L}}^m_t}{\max\left\{\left|\mathcal{L}^m_{i,t} - \bar{\mathcal{L}}^m_t\right|\right\}_{i=0,\ldots,N}} + 1\right)$$

$$\Psi(x_i; \mathcal{A}, \mathbf{M}_i) = a_1(x_i) \circ \ldots \circ a_k(x_i),$$

$$a_j(x_i) = \begin{cases} A_j(x_i; \mathbf{M}_i\zeta) & u \sim \mathcal{U}(0,1) \;\&\; u < \mathbf{M}_i \\ x_i & \text{otherwise} \end{cases} \qquad \mathcal{L}_{\mathrm{BCL}} = \frac{1}{N}\sum_{i=1}^{N} -\log \frac{\exp\left(\frac{f(\Psi(x_i))^\top f(\Psi(x_i^+))}{\tau}\right)}{\sum_{x'_i \in X'} \exp\left(\frac{f(\Psi(x_i))^\top f(\Psi(x'_i))}{\tau}\right)}$$

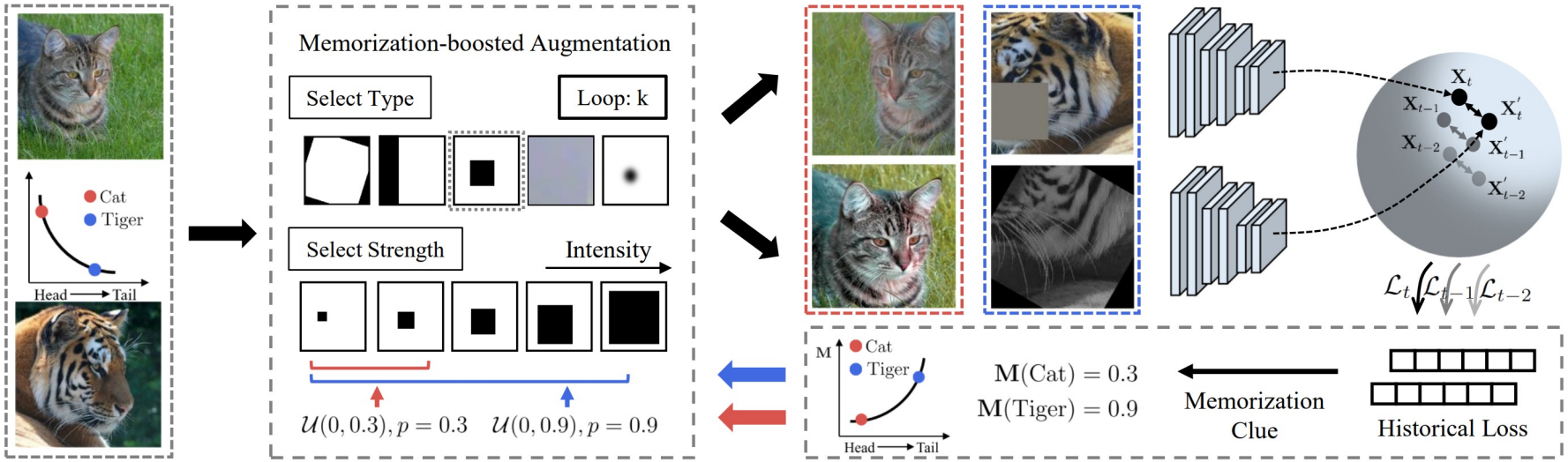
Adaptively assigns the appropriate augmentation strength for the individual sample according to the feedback from the memorization clues

[1] Jiang et al. “Characterizing structural regularities of labeled data in overparameterized models.” ICML 2021

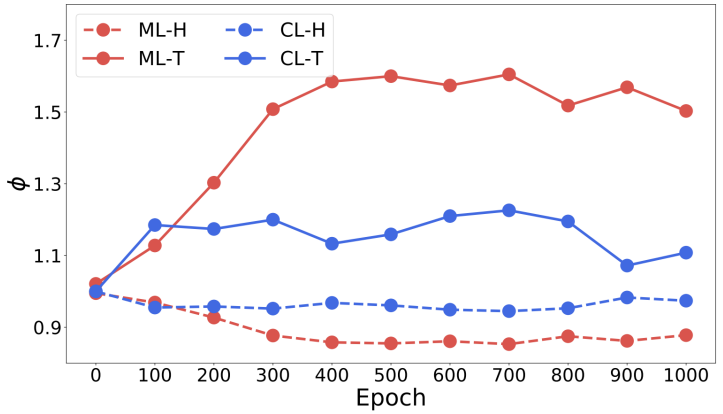
[2] Zhou et al. “Contrastive learning with boosted memorization. " ICML 2022.

Data-based Self-Supervised Long-tailed Learning

BCL: Boosted Contrastive Learning



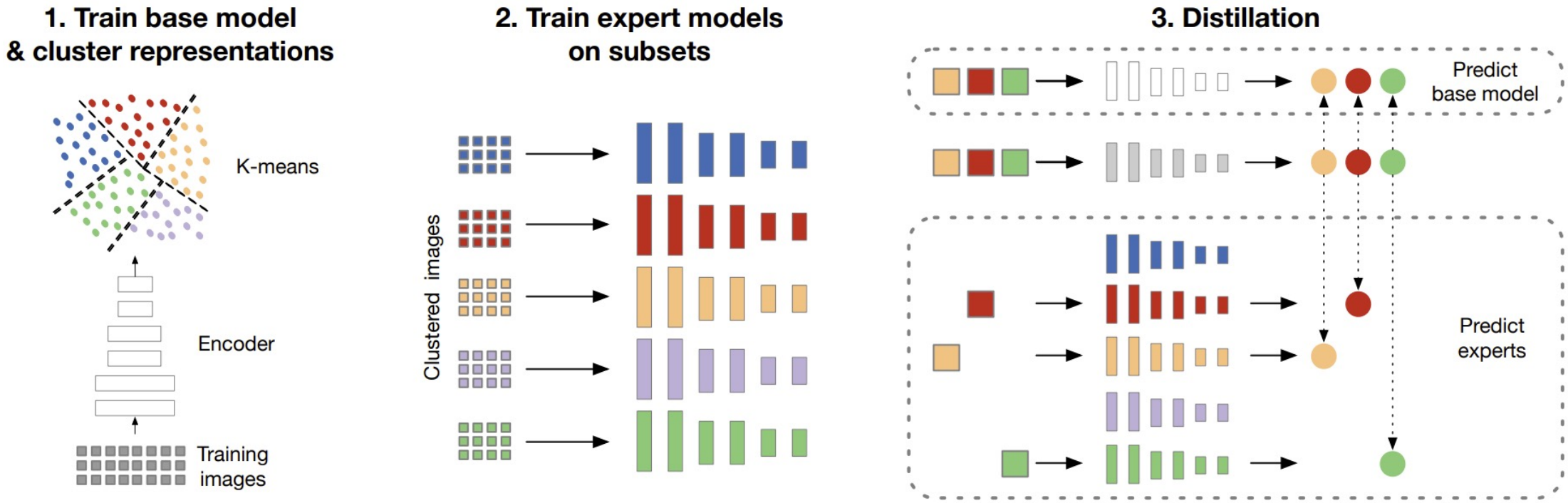
- Calculate memorization scores based on historical statistics to detect tail.
- Construct instance-wise augmentations to enhance representation learning.



[1] Zhou et al. "Contrastive learning with boosted memorization. " ICML 2022.

DnC: Divide and Contrast: Self-Supervised Learning from Uncurated Data

➢ **Motivation:** Conquer-and-Divide Training to isolate the negative effect on tail classes during training.



[1] Tian et al. "Divide and contrast: Self-supervised learning from uncurated data." ICCV 2021.

DnC: Divide and Contrast

Transfer learning experiments

		Food-101	CIFAR10	CIFAR100	Birdsnap	SUN397	Cars	Aircraft	VOC2007	DTD	Pets	Caltech-101	Flowers	Average
YFCC	BYOL-5k	69.1	85.8	66.8	**35.5**	64.1	50.1	**51.9**	82.5	74.5	74.0	**87.6**	95.8	69.8
	MoCLR-5k	68.4	87.6	69.7	30.5	63.9	41.0	46.7	82.4	76.2	68.5	86.0	93.0	67.8
	DnC-4.5k	**72.1**	**88.0**	**71.1**	**35.5**	**67.2**	**52.6**	49.2	**83.7**	**76.5**	**75.9**	87.0	**97.8**	**71.4**
JFT-300M	BYOL-5k	73.3	89.8	72.4	38.2	61.8	64.4	**54.4**	81.3	75.5	77.0	90.1	94.3	72.7
	MoCLR-5k	72.8	90.7	72.5	33.8	62.2	60.6	50.9	81.9	75.3	75.8	89.5	93.8	71.7
	DnC-4.5k	**78.7**	**91.7**	**74.9**	**42.1**	**65.0**	**75.3**	54.1	**83.1**	**76.6**	**86.1**	**90.2**	**98.2**	**76.3**

		COCO detection			COCO instance seg.			PASCAL seg.	NYU v2 depth estimation				
		AP^{bb}	AP^{bb}_{50}	AP^{bb}_{75}	AP^{mk}	AP^{mk}_{50}	AP^{mk}_{75}	mIoU	<1.25	$<1.25^2$	$<1.25^3$	rms↓	rel↓
ImageNet Super.		39.5	60.1	43.3	35.4	56.9	38.1	74.4	81.1	95.3	98.8	0.573	0.127
YFCC	BYOL-5k	41.1	62.0	45.1	36.6	58.6	38.9	75.5	83.5	96.4	99.0	0.558	0.130
	MoCLR-5k	40.8	61.7	44.8	36.6	58.5	39.0	75.1	**86.7**	**97.4**	**99.3**	**0.503**	**0.117**
	DnC-4.5k	**41.5**	**62.5**	**45.6**	**37.0**	**59.3**	**39.6**	**76.6**	86.2	97.2	**99.3**	0.512	0.121
JFT-300M	BYOL-5k	40.6	61.2	44.3	36.2	58.1	38.8	75.8	84.4	96.5	99.0	0.544	0.129
	MoCLR-5k	41.1	62.0	45.4	36.9	58.9	39.5	76.1	**86.3**	**97.2**	99.3	0.513	0.120
	DnC-4.5k	**41.7**	**62.5**	**45.9**	**37.2**	**59.3**	**39.8**	**76.9**	86.1	**97.2**	**99.4**	**0.509**	**0.119**

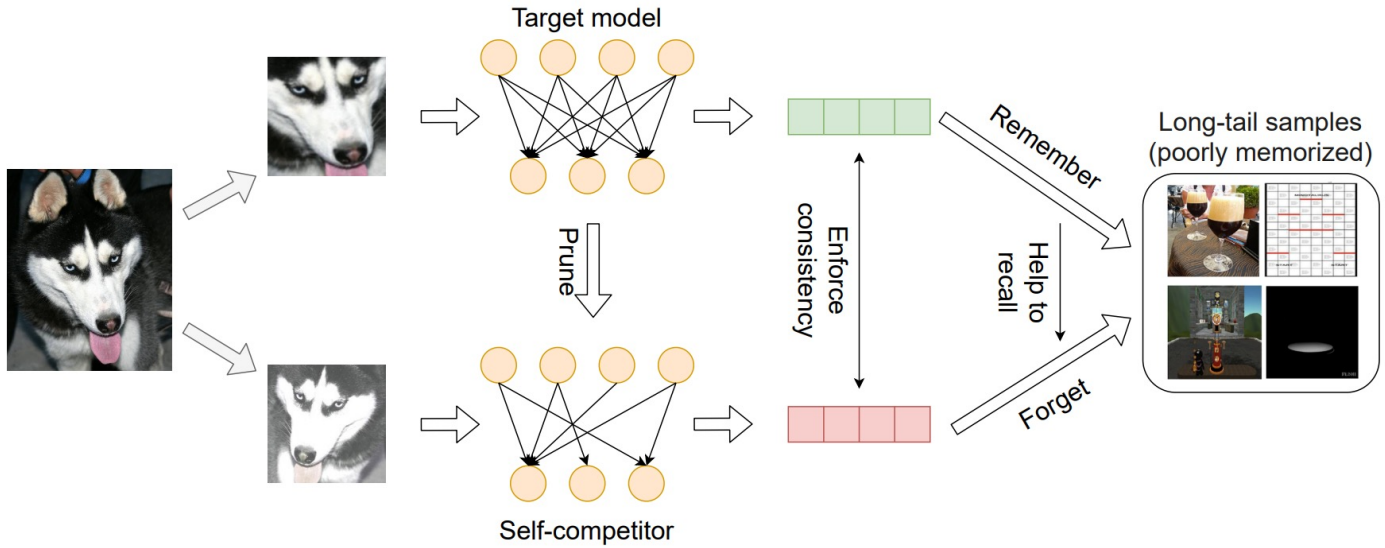
Method	Arch	pre-training # epochs	ImageNet Top-1 Acc	Places 365 Top-1 Acc
Concurrent work trained on IG 1B images:				
SEER [32]	R-50	≈1,000	61.6	-
	R-101	≈1,000	65.8	-
Pre-training on YFCC100M:				
MoCLR	R-50	1,000	65.1	53.2
BYOL	R-50	1,000	65.3	52.9
MoCLR	R-50	3,000	65.7	53.2
BYOL	R-50	3,000	66.6	52.9
DnC	R-50	3,000	**67.8**	**54.1**
MoCLR	R-50	5,000	66.1	53.5
BYOL	R-50	5,000	67.0	53.2
DnC	R-50	4,500	**68.5**	**54.4**
Pre-training on JFT-300M:				
MoCLR	R-50	1,000	66.6	52.1
BYOL	R-50	1,000	67.0	51.9
DnC	R-50	1,000	**67.9**	**52.5**
MoCLR	R-50	3,000	67.4	52.5
BYOL	R-50	3,000	67.6	52.4
DnC	R-50	3,000	**69.8**	**53.3**
MoCLR	R-50	5,000	67.6	52.4
BYOL	R-50	5,000	67.9	52.4
DnC	R-50	4,500	**70.7**	**53.5**
With larger ResNet:				
MoCLR	R-200x2	3,000	74.2	54.6
DnC	R-200x2	3,000	**77.3**	**56.2**

[1] Tian et al. "Divide and contrast: Self-supervised learning from uncurated data." ICCV 2021.

SDCLR: Self-damaging Contrastive Learning

- **Intuition:** The sensitivity of head and tail samples to the model pruning, are very different, which helps us to anchor and promote the training of tail samples.

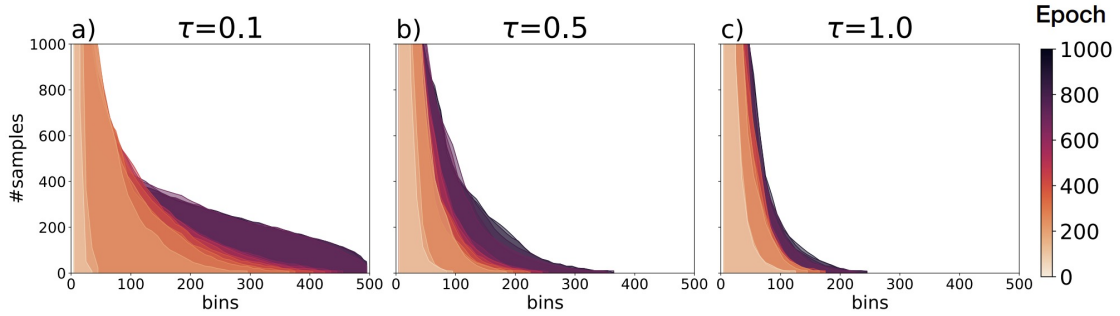
- **Pruning identified exemplars (PIE)** systematically investigates the model output changes introduced by pruning and finds that certain examples are particularly sensitive to sparsity. They are high likely to be rare and atypical samples, which probably comes from tail classes.



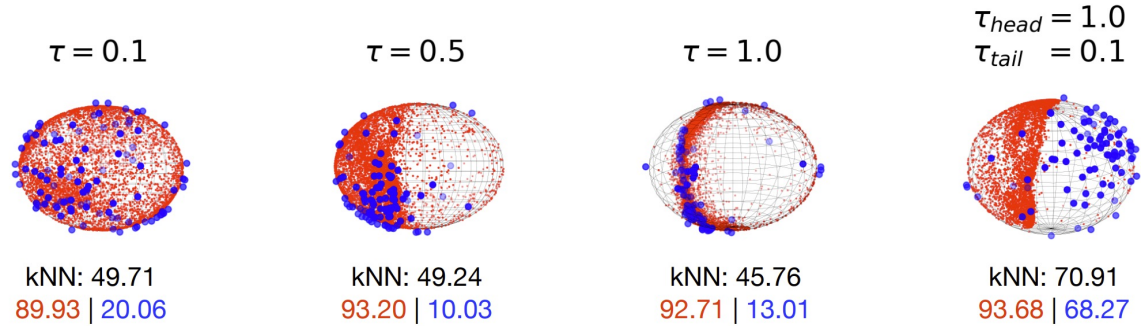
[1] Jiang et al. "Self-damaging contrastive learning." ICML 2021.

TS: Temperature Schedules for contrastive learning

- **Observation:** TS investigates the role of the temperature parameter τ in the contrastive loss, and find that a large τ emphasizes group-wise discrimination, whereas a small τ leads to a higher degree of instance discrimination.



Coverage of embedding space during training. For small τ the representations are more uniformly distributed.



Representations of a head and a tail class. Red: head class and blue: tail class. Small τ = 0.1 promotes uniformity, while large τ = 1.0 creates dense clusters. Tail classes benefit from instance discrimination.

- **Temperature Schedules (TS):** alternates between an upper τ and a lower τ bound at a fixed period length T

$$\tau_{\cos}(t) = (\tau_+ - \tau_-) \times (1 + \cos(2\pi\, t/T))/2 + \tau_-$$

method	CIFAR-10-LT kNN@10	CIFAR-10-LT FS LP	CIFAR-10-LT LS LP	CIFAR-100-LT kNN@10	CIFAR-100-LT FS LP	CIFAR-100-LT LT LP	ImageNet-100-LT kNN@10	ImageNet-100-LT FS LP	ImageNet-100-LT LT LP
SimCLR	60.19	68.29	61.68	28.12	25.70	31.20	38.00	42.64	44.82
SDCLR	60.74	71.03	64.99	29.22	27.28	**34.23**	37.36	42.74	46.40
SimCLR + TS	**62.91**	**71.86**	**65.03**	**30.06**	**28.89**	33.28	**38.86**	**45.18**	**47.26**

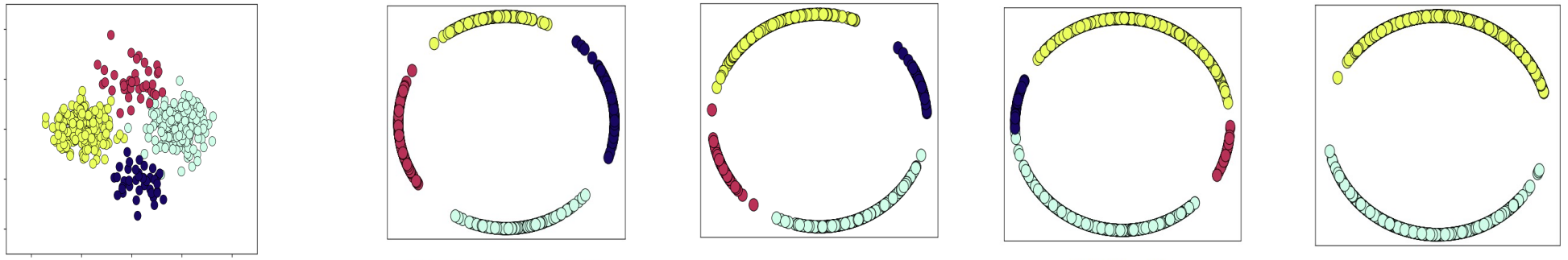
[1] Kukleva et al. "Temperature Schedules for self-supervised contrastive methods on long-tail data." ICLR 2023.

GH: Geometric Harmonization

➢ Why the conventional contrastive learning underperforms in self-supervised long-tailed context?

Conventional contrastive loss motivates *sample-level uniformity*, that is biased towards the head classes.

➢ Geometric Harmonization aims at achieves *category-level uniformity*, *i.e.*, equal allocation *w.r.t.* classes



(a) R=1(Balanced) (b) R=4 (c) R=16 (d) R=64

Contrastive learning causes severer representation learning disparity when enlarging the imbalance ratios.

➢ Challenges I: No guarantee for the desired category-level uniformity

➢ Challenges II: The latent true labels are not available, while the estimated labels are noisy

Geometric Uniform Structure

$$\mathbf{M}_i^\top \cdot \mathbf{M}_j = C, \quad \forall i, j \in \{1, 2, \ldots, K\}, \ i \neq j,$$

Any two vectors in $\mathbf{M}$ have the same angle, namely, the unit space are equally partitioned by the vectors.

Surrogate Label Allocation

$$\min_{\hat{\mathbf{Q}}=[\hat{\boldsymbol{q}}_1,\ldots,\hat{\boldsymbol{q}}_N]} \mathcal{L}_{\mathrm{GH}} = -\frac{1}{|\mathcal{D}|} \sum_{\boldsymbol{x}_i \sim \mathcal{D}} \hat{\boldsymbol{q}}_i \log \boldsymbol{q}_i,$$

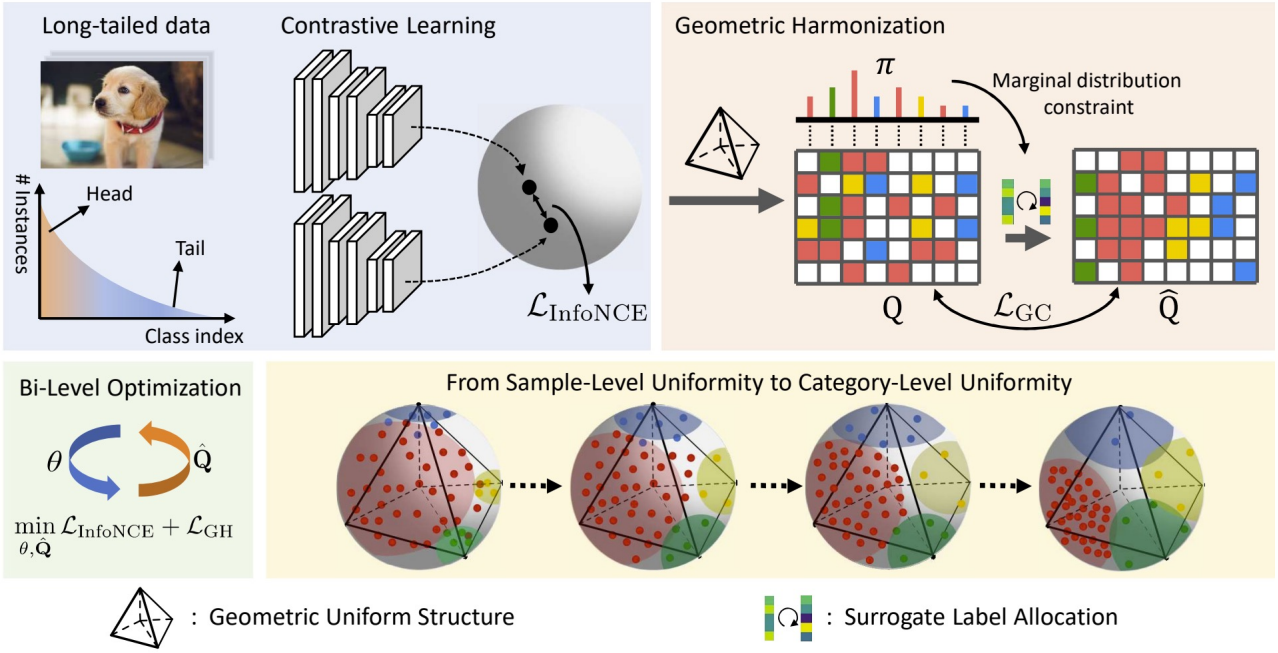
$$\text{s.t. } \hat{\mathbf{Q}} \cdot \mathbb{1}_N = N \cdot \boldsymbol{\pi}, \ \hat{\mathbf{Q}}^\top \cdot \mathbb{1}_K = \mathbb{1}_N,$$

Overall objective

$$\min_{\theta, \hat{\mathbf{Q}}} \mathcal{L} = \mathcal{L}_{\mathrm{InfoNCE}} + w_{\mathrm{GH}} \mathcal{L}_{\mathrm{GH}},$$

[1] Zhou et al. "Combating Representation Learning Disparity with Geometric Harmonization." NeurIPS 2023.

GH: Geometric Harmonization

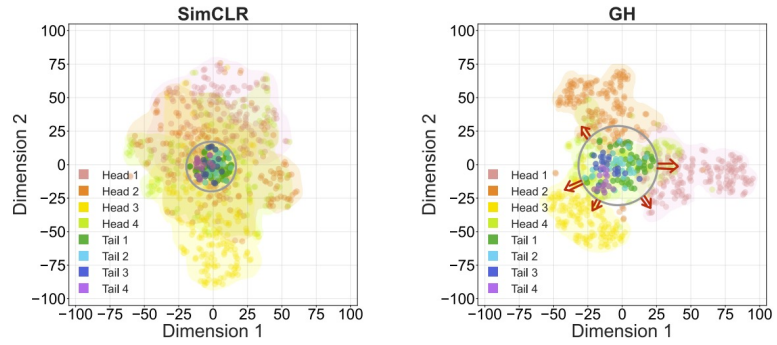


Dataset		SimCLR	+GH	Focal	+GH	SDCLR	+GH	DnC	+GH	BCL	+GH	Improv.
CIFAR-R100	Many	54.97	57.38	54.24	57.01	57.32	57.44	55.41	57.56	59.15	59.50	+1.56
	Med	49.39	52.27	49.58	52.93	50.70	52.85	51.30	53.74	54.82	55.73	+2.35
	Few	47.67	52.12	49.21	51.74	50.45	54.06	50.76	53.26	55.30	57.67	+3.09
	Std	3.82	2.99	2.80	2.76	3.90	2.38	2.54	2.36	2.37	1.89	-0.61
	Avg	50.72	53.96	51.04	53.92	52.87	54.81	52.52	54.88	56.45	57.65	+2.32
CIFAR-R50	Many	56.00	58.88	55.40	57.97	57.50	58.47	56.03	59.04	59.44	60.82	+2.16
	Med	50.48	53.00	51.14	53.55	51.85	53.88	52.68	55.05	54.73	57.58	+2.44
	Few	50.12	54.27	50.02	53.58	52.15	53.58	50.83	54.81	57.30	58.55	+2.87
	Std	3.30	3.09	2.84	2.54	3.18	2.74	2.64	2.38	2.36	1.66	-0.38
	Avg	52.24	55.42	52.22	55.06	53.87	55.34	53.21	56.33	57.18	59.00	+2.49
CIFAR-R10	Many	57.85	59.26	58.18	60.06	58.47	59.21	59.82	61.09	60.41	61.41	+1.26
	Med	55.06	56.91	55.82	56.79	54.79	56.06	56.67	58.33	57.15	59.27	+1.57
	Few	54.03	55.85	54.64	57.24	52.97	55.58	56.21	57.33	59.76	60.30	+1.74
	Std	1.98	1.75	1.80	1.77	2.80	1.97	1.96	1.95	1.73	1.07	-0.35
	Avg	55.67	57.36	56.23	58.05	55.44	56.97	57.59	58.94	59.12	60.34	+1.52
ImageNet-LT	Many	41.69	41.53	42.04	42.55	40.87	41.92	41.70	42.19	42.92	43.22	+0.44
	Med	33.96	36.35	35.02	36.75	33.71	36.53	34.68	36.63	35.89	38.16	+2.23
	Few	31.82	35.84	33.32	36.28	32.07	36.04	33.58	35.86	33.93	36.96	+3.25
	Std	5.19	3.15	4.62	3.49	4.68	3.26	4.41	3.45	4.73	3.32	-1.39
	Avg	36.65	38.28	37.49	38.92	36.25	38.53	37.23	38.67	38.33	39.95	+1.68
Places-LT	Many	31.98	32.46	31.69	32.40	32.17	32.78	32.07	32.51	32.69	33.22	+0.55
	Med	34.05	35.03	34.33	35.14	34.71	35.60	34.51	35.55	35.37	36.00	+0.87
	Few	35.63	36.14	35.73	36.49	35.69	36.18	35.84	35.91	37.18	37.62	+0.45
	Std	1.83	1.89	2.05	2.08	1.82	1.82	1.91	1.87	2.26	2.23	0.00
	Avg	33.61	34.33	33.65	34.42	33.99	34.70	33.90	34.52	34.76	35.32	+0.68

Theorem 3.4. *(Optimal state for $\mathcal{L}$) Given Eq. (4) under the proper optimization strategy, when it arrives at the category-level uniformity (Definition 3.3) defined on the geometric uniform structure* $\mathbf{M}$ *(Definition 3.1), we will achieve the minimum of the overall loss $\mathcal{L}^*$ as*

$$\mathcal{L}^* = -2\sum_{l=1}^{L} \boldsymbol{\pi}_l^{\boldsymbol{y}} \log\left(1/\left(1+(K-1)\exp(C-1)\right)\right) + \log\left(J/L\right), \tag{5}$$

where J denotes the size of the collection of the negative samples and $\boldsymbol{\pi}^{\boldsymbol{y}}$ refers to the marginal distribution of the latent ground-truth labels $\boldsymbol{y}$.



[1] Zhou et al. "Combating Representation Learning Disparity with Geometric Harmonization." NeurIPS 2023.

rwSAM: Reweighted Sharpness Aware Minimization

- **Motivation:** Many prior works on imbalanced supervised learning regularize the rare classes more strongly, motivated by the observation that the rare classes suffer from more overfitting. The frequent classes have much smaller pre-training generalization gap than the rare classes.

- **Reweighted SAM:** A data-dependent regularizer that can have different effects on rare and frequent examples. SAM improves model generalization by penalizing loss sharpness.

$$\min_{\phi} \widehat{L}(\phi + \epsilon_w(\phi)),$$
$$\text{where} \quad \epsilon_w(\phi) = \arg\max_{\|\epsilon\|<\rho} \epsilon^\top \nabla_\phi \widehat{L}_w(\phi).$$

The rwSAM objective re-weights the regularization-related terms with the estimated density by kernel density estimation:

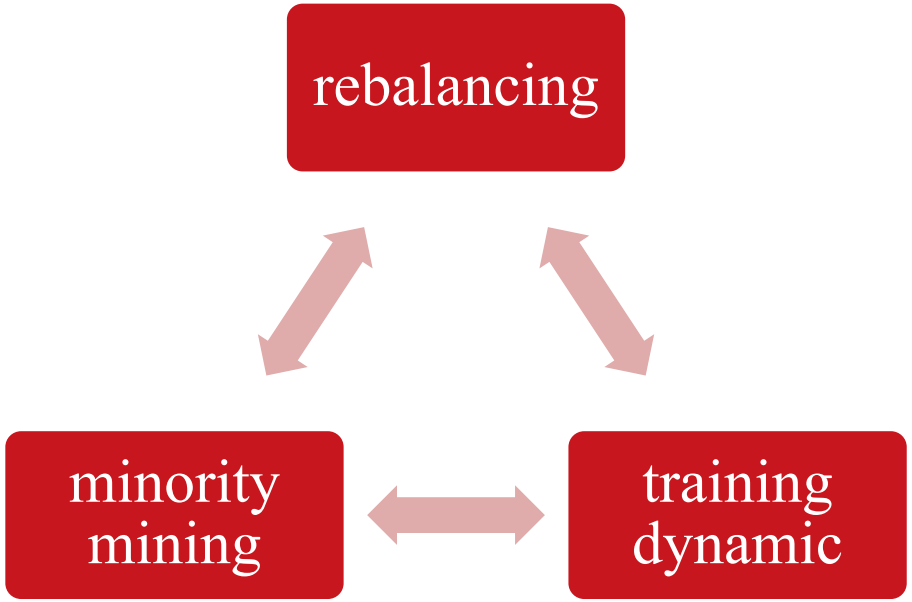
$$w_i = \left(\frac{1}{n}\sum_{j=1}^{n} K(f_\phi(x_i) - f_\phi(x_j), h)\right)^{-\alpha}$$

(b) ImageNet, OOD	Target dataset				
Method	CUB	Cars	Aircrafts	Pets	Avg.
MoCo v2	69.9 ± 0.7	88.4 ± 0.4	82.9 ± 0.6	80.1 ± 0.6	80.3
MoCo v2+SAM	69.9 ± 0.5	88.8 ± 0.5	83.4 ± 0.4	81.5 ± 0.8	80.9
MoCo v2, balanced	69.8 ± 0.5	88.6 ± 0.4	82.7 ± 0.5	80.0 ± 0.4	80.2
MoCo v2+rwSAM	70.3 ± 0.7	88.7 ± 0.3	84.9 ± 0.6	81.7 ± 0.4	81.4
SimSiam	70.0 ± 0.3	87.0 ± 0.6	81.5 ± 0.7	83.8 ± 0.5	80.6
SimSiam, balanced	70.5 ± 0.8	87.9 ± 0.7	81.8 ± 0.7	82.7 ± 0.4	80.7
SimSiam+rwSAM	70.7 ± 0.8	88.4 ± 0.6	82.6 ± 0.6	84.0 ± 0.4	81.4

[1] Liu et al. "Self-supervised Learning is More Robust to Dataset Imbalance." ICLR 2021.

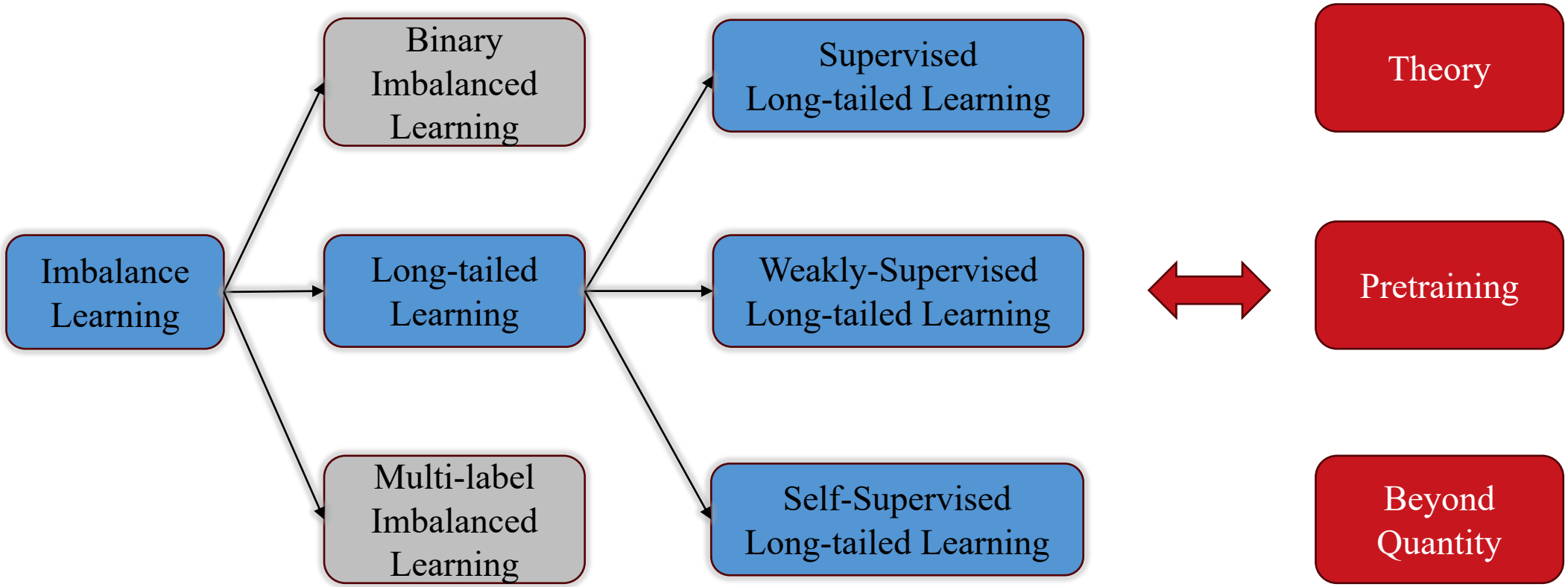
Summary

- Self-supervised learning is **more robust** to data imbalance **than the supervised counterpart**
- However, self-supervised learning still suffers from the long-tailed distribution, resulting in performance degeneration and **representation learning disparity**



Summary

Still require more efforts on this way



Tutorial website: https://tmlr-group.github.io/tutorials/aaai2024.html

Thank you

Q & A