
南京航空航天大學

Nanjing University of Aeronautics and Astronautics

# Rethinking the Value of Labels for Improving Class-Imbalanced Learning

**Yuzhe Yang**
EECS
Massachusetts Institute of Technology
yuzhe@mit.edu

**Zhi Xu**
EECS
Massachusetts Institute of Technology
zhixu@mit.edu

***NIPS 2020***

# Class Imbalanced Learning



1. Re-sampling:
   - Under-sampling: RUS, ENN
   - Over-sampling: *SMOTE, ADASYN*
2. Re-weighting: *Focal-Loss, Effective Numbers, LDAM*
3. Ensemble: *SMOTEBoost, EasyEnsemble & BalanceCascade*
4. Transfer learning: *OLTR, Range-Loss, FTL*
5. Metric learning: *Contrastive-Learning*
6. Decoupling: *BBN, Classifier-Balancing*
7. Meta-Learning: *Meta-Weight-Net*

# Motivation – “the value of labels”



- *Positive*: Imbalanced labels are indeed **valuable**. Extra unlabeled data **benefits** imbalanced learning (Semi-supervised Manner)

- *Negative*: **Imbalanced labels** naturally impose **“label bias”** during learning, where the decision boundary can be significantly driven by the majority classes. (Self-supervised Manner)

*How to maximally exploit **the value of labels** to improve class-imbalanced learning?*

# Imbalanced Learning with Unlabeled Data



A mixture of two Gaussians (μ1 > μ2): $P_{XY}$

The optimal Bayes's classifier: $f(x) = \text{sign}(x - \frac{\mu_1+\mu_2}{2})$

Target: $\theta = \frac{\mu_1 + \mu_2}{2}$

Base classifier trained on imbalanced data: $f_B$

Imbalance in accuracy: $\Delta$

Pseudo-label is positive/negative $\tilde{n}_+ \quad \tilde{n}_-$

# Imbalanced Learning with Unlabeled Data



**Theorem 1.** *Consider the above setup. For any* $\delta > 0$*, with probability at least* $1 - 2e^{-\frac{2\delta^2}{9\sigma^2} \cdot \frac{\tilde{n}_+ \tilde{n}_-}{\tilde{n}_- + \tilde{n}_+}} - 2e^{-\frac{8\tilde{n}_+ \delta^2}{9(\mu_1 - \mu_2)^2}} - 2e^{-\frac{8\tilde{n}_- \delta^2}{9(\mu_1 - \mu_2)^2}}$ *our estimates* $\hat{\theta}$ *satisfies*

$$\left|\hat{\theta} - (\mu_1 + \mu_2)/2 - \Delta(\mu_1 - \mu_2)/2\right| \leq \delta.$$

*Interpretation.*

- *Training* ***data imbalance*** *affects the accuracy of our estimations*

- **Unlabeled data imbalance** affects the probability of obtaining such a good estimation.

# Semi-Supervised Imbalanced Learning Framework

(a) CIFAR-10-LT

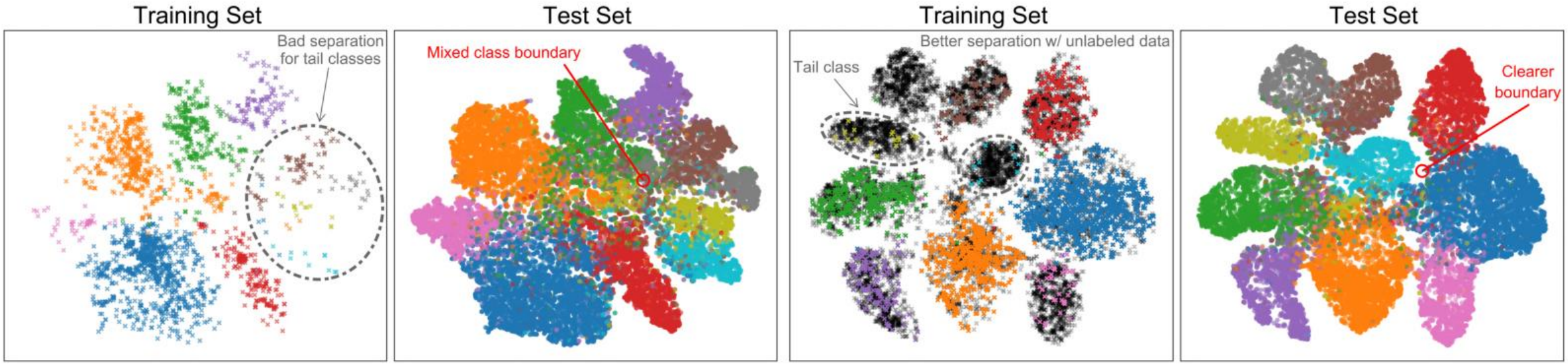
| Imbalance Ratio ($\rho$) | 100 | | | | 50 | | | | 10 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $\mathcal{D}_U$ Imbalance Ratio ($\rho_U$) | 1 | $\rho/2$ | $\rho$ | $2\rho$ | 1 | $\rho/2$ | $\rho$ | $2\rho$ | 1 | $\rho/2$ | $\rho$ | $2\rho$ |
| CE | 29.64 | | | | 25.19 | | | | 13.61 | | | |
| CE + $\mathcal{D}_U$ @5x | **17.48** | 18.42 | 18.74 | 20.06 | **16.79** | 16.88 | 18.36 | 19.94 | **10.22** | 10.48 | 10.86 | 11.04 |
| LDAM-DRW [7] | 22.97 | | | | 19.06 | | | | 11.84 | | | |
| LDAM-DRW + $\mathcal{D}_U$ @5x | **14.96** | 15.18 | 15.33 | 15.55 | **14.33** | 14.70 | 14.93 | 15.24 | 8.72 | **8.24** | 8.68 | 8.97 |

(b) SVHN-LT

| Imbalance Ratio ($\rho$) | 100 | | | | 50 | | | | 10 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $\mathcal{D}_U$ Imbalance Ratio ($\rho_U$) | 1 | $\rho/2$ | $\rho$ | $2\rho$ | 1 | $\rho/2$ | $\rho$ | $2\rho$ | 1 | $\rho/2$ | $\rho$ | $2\rho$ |
| CE | 19.98 | | | | 17.50 | | | | 11.46 | | | |
| CE + $\mathcal{D}_U$ @5x | **13.02** | 13.73 | 14.65 | 15.04 | **13.07** | 13.36 | 13.16 | 14.54 | **10.01** | 10.20 | 10.06 | 10.71 |
| LDAM-DRW [7] | 16.66 | | | | 14.59 | | | | 10.27 | | | |
| LDAM-DRW + $\mathcal{D}_U$ @5x | **11.32** | 11.70 | 11.92 | 12.78 | **10.98** | 11.14 | 11.26 | 11.51 | 8.94 | 9.08 | **8.70** | 9.35 |

Loss function: $\mathcal{L}(\mathcal{D}_L, \theta) + \omega\mathcal{L}(\mathcal{D}_U, \theta)$

# Semi-Supervised Imbalanced Learning Framework




(a) Standard CE training (b) With unlabeled data $\mathcal{D}_U$ @5x (colored in black)

Figure 1: t-SNE visualization of training & test set on SVHN-LT. Using unlabeled data helps to shape clearer class boundaries and results in better class separation, especially for the tail classes.

# Semi-Supervised Imbalanced Learning Framework

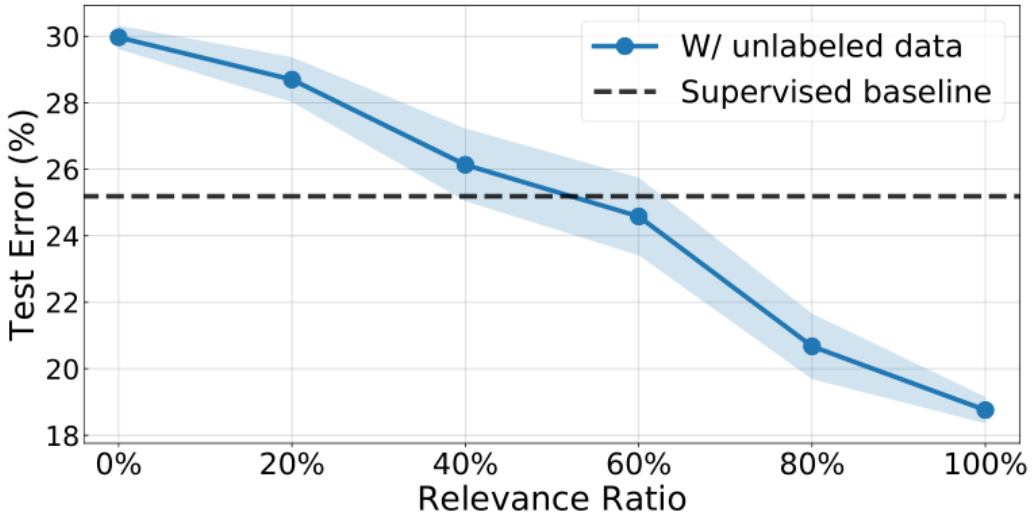



Figure 2: Test errors of different unlabeled data relevance ratios on CIFAR-10-LT with $\rho = 50$. We fix $\rho_U = 50$ for the relevant unlabeled data.

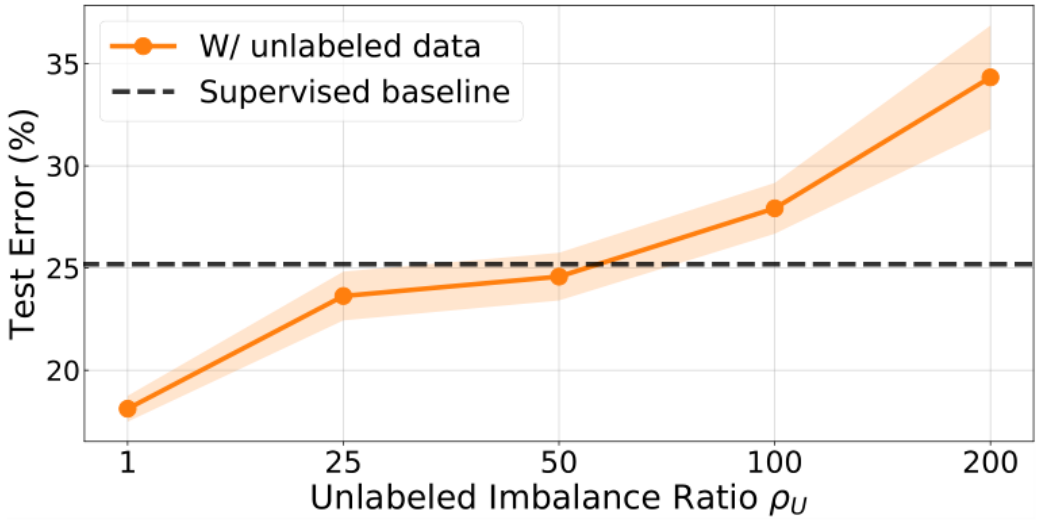



Figure 3: Test errors of different $\rho_U$ of relevant unlabeled data on CIFAR-10-LT with $\rho = 50$. We fix the unlabeled data relevance ratio as $60\%$.

# Imbalanced Learning from Self-Supervision

A mixture of two Gaussians (μ1 = μ2):

$$X|Y=+1 \sim \mathcal{N}(0, \sigma_1^2 \mathbf{I}_d)$$

$$X|Y=-1 \sim \mathcal{N}(0, \beta\sigma_1^2 \mathbf{I}_d)$$

Linear classifiers:

$$f(x) = \text{sign}(\langle \theta, \text{feature} \rangle + b)$$

Standard error probability:

$$\text{err}_f = \mathbb{P}_{(X,Y)\sim P_{XY}}\left(f(X) \neq Y\right)$$

Learned representation:

$$Z = k_1 ||X||_2^2 + k_2$$

**Theorem 2.** *Let $\Phi$ be the CDF of $\mathcal{N}(0,1)$. For any linear classifier of the form $f(X) = sign\big(\langle \theta, X \rangle + b\big)$ where $b > 0$, the error probability satisfies:* $\text{err}_f = p_+ \Phi\left(-\frac{b}{||\theta||_2 \sigma_1}\right) + p_- \Phi\left(\frac{b}{||\theta||_2 \sqrt{\beta}\sigma_1}\right) \geq \frac{1}{4}$.

# Imbalanced Learning from Self-Supervision

$$f_{ss}(X) = \text{sign}(-Z + b),\ b = \frac{1}{2}\left(\frac{\sum_{i=1}^{N} \mathbf{1}_{\{Y_i=+1\}} Z_i}{N_+} + \frac{\sum_{i=1}^{N} \mathbf{1}_{\{Y_i=-1\}} Z_i}{N_-}\right)$$

**Theorem 3.** *Consider the linear classifier with self-supervised learning, $f_{ss}$. For any $\delta \in \left(0, \frac{\beta-1}{\beta+1}\right)$ we have that with probability at least $1 - 2e^{-N_- d\delta^2/8} - 2e^{-N_+ d\delta^2/8}$, the classifier satisfies*

$$\text{err}_{f_{ss}} \leq \begin{cases} p_+ e^{-d\cdot\frac{(\beta-1-(1+\beta)\delta)^2}{32}} + p_- e^{-d\cdot\frac{(\beta-1-(1+\beta)\delta)^2}{32\beta^2}}, & \text{if } \delta \in \left[\frac{\beta-3}{\beta+1}, \frac{\beta-1}{\beta+1}\right); \\ p_+ e^{-d\cdot\frac{(\beta-1-(1+\beta)\delta)}{16}} + p_- e^{-d\cdot\frac{(\beta-1-(1+\beta)\delta)^2}{32\beta^2}}, & \text{if } \delta \in \left(0, \frac{\beta-3}{\beta+1}\right). \end{cases}$$

*Interpretation.*

- *With high probability, we obtain a **satisfying classifier fss(x)**, whose error probability decays exponentially on the dimension d.*
- **Training data imbalance** affects our probability of obtaining such a satisfying classifier.

# Self-Supervised Imbalanced Learning Framework



Table 2: Top-1 test error rates (%) of ResNet-32 on long-tailed CIFAR-10 and CIFAR-100. Using SSP, we consistently improve different imbalanced learning techniques, and achieve the best performance.

| Dataset | CIFAR-10-LT | | | CIFAR-100-LT | | |
|---|---|---|---|---|---|---|
| Imbalance Ratio ($\rho$) | 100 | 50 | 10 | 100 | 50 | 10 |
| CE | 29.64 | 25.19 | 13.61 | 61.68 | 56.15 | 44.29 |
| CB-CE [11] | 27.63 | 21.95 | 13.23 | 61.44 | 55.45 | 42.88 |
| CB-CE + ***SSP*** | **23.47** | **19.60** | **11.57** | **56.94** | **52.91** | **41.94** |
| Focal [32] | 29.62 | 23.29 | 13.34 | 61.59 | 55.68 | 44.22 |
| CB-Focal [11] | 25.43 | 20.73 | 12.90 | 60.40 | 54.83 | 42.01 |
| CB-Focal + ***SSP*** | **22.90** | **18.74** | **11.75** | **57.03** | **53.12** | **41.16** |
| CE-DRW [7] | 24.94 | 21.10 | 13.57 | 59.49 | 55.31 | 43.78 |
| CE-DRS [7] | 25.53 | 21.39 | 13.73 | 59.62 | 55.46 | 43.95 |
| CE-DRW + ***SSP*** | **23.04** | **19.93** | **12.66** | **57.21** | **53.57** | **41.77** |
| LDAM [7] | 26.65 | 23.18 | 13.04 | 60.40 | 55.03 | 43.09 |
| LDAM-DRW [7] | 22.97 | 19.06 | 11.84 | 57.96 | 53.85 | 41.29 |
| LDAM-DRW + ***SSP*** | **22.17** | **17.87** | **11.47** | **56.57** | **52.89** | **41.09** |

# Self-Supervised Imbalanced Learning Framework



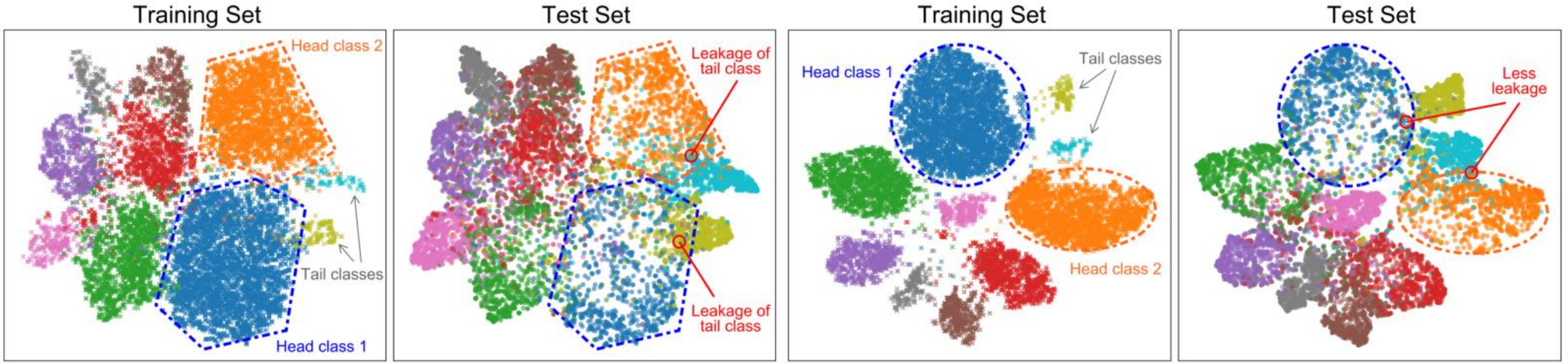
Table 3: Top-1 test error rates (%) on ImageNet-LT. † denotes results reproduced with authors' code.

| Method | ResNet-10 | ResNet-50 |
|---|---|---|
| CE (Uniform) | 65.2 | 61.6 |
| CE (Uniform) + ***SSP*** | **64.1** | **54.4** |
| CE (Balanced) | 62.9 | 59.7 |
| CE (Balanced) + ***SSP*** | **61.6** | **52.4** |
| OLTR [33] | 64.4 | 62.6† |
| OLTR + ***SSP*** | **62.3** | **53.9** |
| cRT [25] | 58.2 | 52.7 |
| cRT + ***SSP*** | **56.8** | **48.7** |

Table 4: Top-1 test error rates (%) on iNaturalist 2018. † denotes results reproduced with authors' code.

| Method | ResNet-50 |
|---|---|
| CE (Uniform) | 39.3 |
| CE (Uniform) + ***SSP*** | **35.6** |
| CE (Balanced) | 36.5 |
| CE (Balanced) + ***SSP*** | **34.1** |
| LDAM-DRW [7] | 35.4† |
| LDAM-DRW + ***SSP*** | **33.7** |
| cRT [25] | 34.8 |
| cRT + ***SSP*** | **31.9** |

# Self-Supervised Imbalanced Learning Framework




(a) Standard CE training (b) Standard CE training with SSP

Figure 4: t-SNE visualization of training & test set on CIFAR-10-LT. Using SSP helps mitigate the tail classes leakage during testing, which results in better learned boundaries and representations.


南京航空航天大學

Nanjing University of Aeronautics and Astronautics

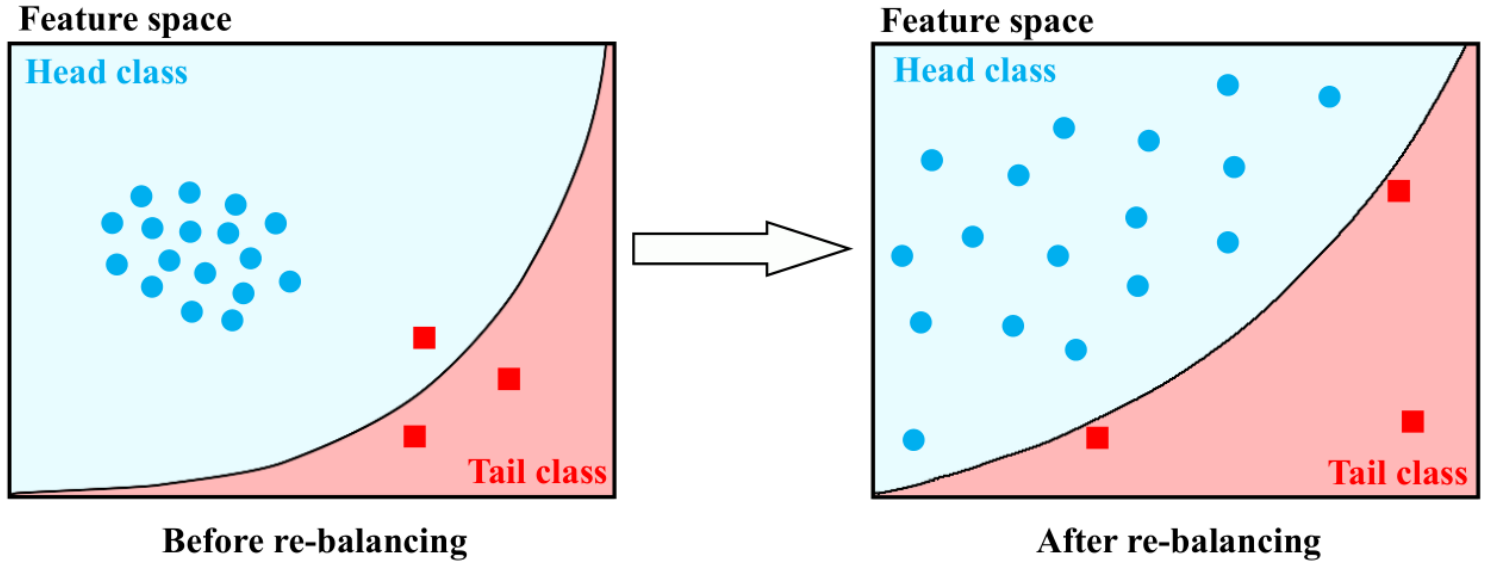
# BBN: Bilateral-Branch Network with Cumulative Learning for Long-Tailed Visual Recognition

Boyan Zhou[1] Quan Cui[1,2] Xiu-Shen Wei[1*] Zhao-Min Chen[1,3]

[1]Megvii Technology [2]Waseda University [3]Nanjing University

*CVPR 2020*

# Intuition




- *Class re-balancing strategies is to significantly* ***promote classifier learning*** *but will unexpectedly* ***damage the representative ability*** *of the learned deep features.*
- After re-balancing, the decision boundary tends to accurately classify the tail data. However, the **intra-class distribution** of each class becomes more **separable**.

# Intuition

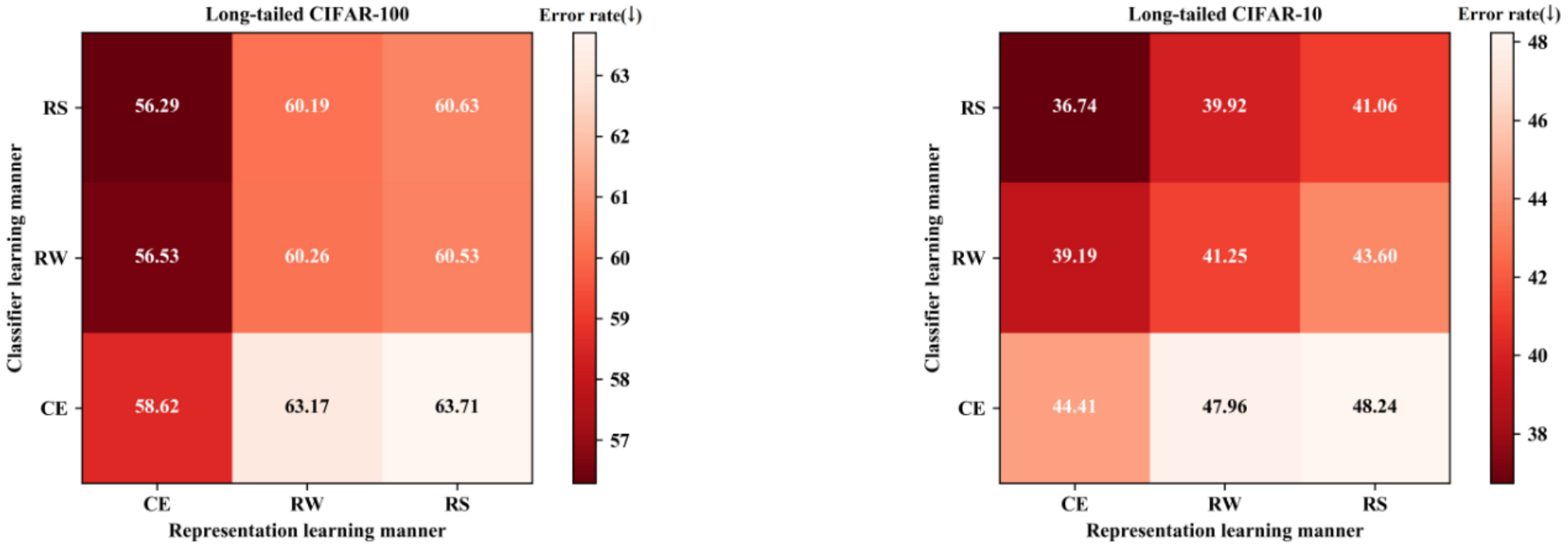
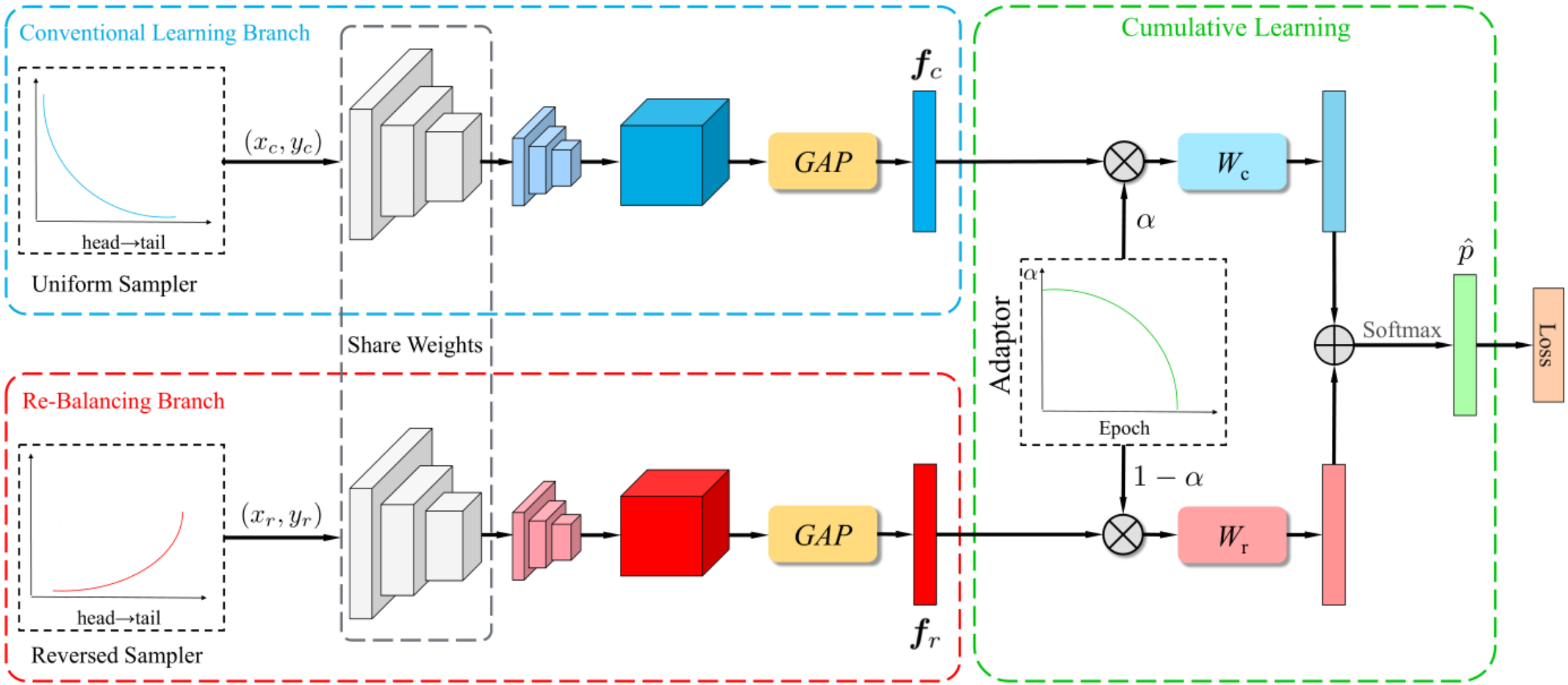



Figure 2. Top-1 error rates of different manners for representation learning and classifier learning on two long-tailed datasets CIFAR-100-IR50 and CIFAR-10-IR50 [3]. "CE" (Cross-Entropy), "RW" (Re-Weighting) and "RS" (Re-Sampling) are the conducted learning manners. As observed, when fixing the representation (comparing error rates of three blocks in the vertical direction), the error rates of classifiers trained with RW/RS are reasonably lower than CE. While, when fixing the classifier (comparing error rates in the horizontal direction), the representations trained with CE surprisingly get lower error rates than those with RW/RS. Experimental details can be found in Section 3.

# BBN framework



# Algorithm



**Algorithm 1** Learning algorithm of our proposed BBN

**Require :** Training Dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n}$; $\texttt{UniformSampler}(\cdot)$ denotes obtaining a sample from $\mathcal{D}$ selected by a uniform sampler; $\texttt{ReversedSampler}(\cdot)$ denotes obtaining a sample by a reversed sampler; $\mathcal{F}_{cnn}(\cdot;\cdot)$ denotes extracting the feature representation from a CNN; $\theta_c$ and $\theta_r$ denote the model parameters of the conventional learning and re-balancing branch; $\boldsymbol{W}_c$ and $\boldsymbol{W}_r$ present the classifiers' weights (*i.e.*, last fully connected layers) of the conventional learning and re-balancing branch.

**for** $T = 1$ to $T_{max}$ **do**
  $\alpha \leftarrow 1 - \left(\frac{T}{T_{max}}\right)^2$
  $(\mathbf{x}_c, y_c) \leftarrow \texttt{UniformSampler}(\mathcal{D})$
  $(\mathbf{x}_r, y_r) \leftarrow \texttt{ReversedSampler}(\mathcal{D})$
  $\boldsymbol{f}_c \leftarrow \mathcal{F}_{cnn}(\mathbf{x}_c; \theta_c)$
  $\boldsymbol{f}_r \leftarrow \mathcal{F}_{cnn}(\mathbf{x}_r; \theta_r)$
  $\mathbf{z} \leftarrow \alpha \boldsymbol{W}_c^\top \boldsymbol{f}_c + (1-\alpha)\boldsymbol{W}_r^\top \boldsymbol{f}_r$
  $\hat{\boldsymbol{p}} \leftarrow \texttt{Softmax}(\boldsymbol{z})$
  $\mathcal{L} \leftarrow \alpha E(\hat{\boldsymbol{p}}, y_c) + (1-\alpha) E(\hat{\boldsymbol{p}}, y_r)$
  Update model parameters by minimizing $\mathcal{L}$
**end for**

# Experiments



Table 1. Top-1 error rates of ResNet-32 on long-tailed CIFAR-10 and CIFAR-100. (Best results are marked in bold.)

| Dataset | Long-tailed CIFAR-10 | | | Long-tailed CIFAR-100 | | |
|---|---|---|---|---|---|---|
| Imbalance ratio | 100 | 50 | 10 | 100 | 50 | 10 |
| CE | 29.64 | 25.19 | 13.61 | 61.68 | 56.15 | 44.29 |
| Focal [17] | 29.62 | 23.28 | 13.34 | 61.59 | 55.68 | 44.22 |
| Mixup [35] | 26.94 | 22.18 | 12.90 | 60.46 | 55.01 | 41.98 |
| Manifold Mixup [29] | 27.04 | 22.05 | 12.97 | 61.75 | 56.91 | 43.45 |
| Manifold Mixup (two samplers) | 26.90 | 20.79 | 13.17 | 63.19 | 57.95 | 43.54 |
| CE-DRW [3] | 23.66 | 20.03 | 12.44 | 58.49 | 54.71 | 41.88 |
| CE-DRS [3] | 24.39 | 20.19 | 12.62 | 58.39 | 54.52 | 41.89 |
| CB-Focal [5] | 25.43 | 20.73 | 12.90 | 60.40 | 54.83 | 42.01 |
| LDAM-DRW [3] | 22.97 | 18.97 | 11.84 | 57.96 | 53.38 | 41.29 |
| Our BBN | **20.18** | **17.82** | **11.68** | **57.44** | **52.98** | **40.88** |


南京航空航天大學

Nanjing University of Aeronautics and Astronautics

# Self-paced Ensemble for Highly Imbalanced Massive Data Classification

Zhining Liu*†, Wei Cao‡, Zhifeng Gao‡, Jiang Bian‡, Hechang Chen*†, Yi Chang*† and Tie-Yan Liu‡

*School of Artificial Intelligence, Jilin University

†Key Lab. of Symbolic Computation and Knowledge Engineering of MOE, Jilin University

‡Microsoft Research

znliu19@mails.jlu.edu.cn,

{weicao, zhgao, jiang.bian, tyliu}@microsoft.com,

chenhc14@mails.jlu.edu.cn, yichang@jlu.edu.cn

***ICDE 2020***

# Classification hardness

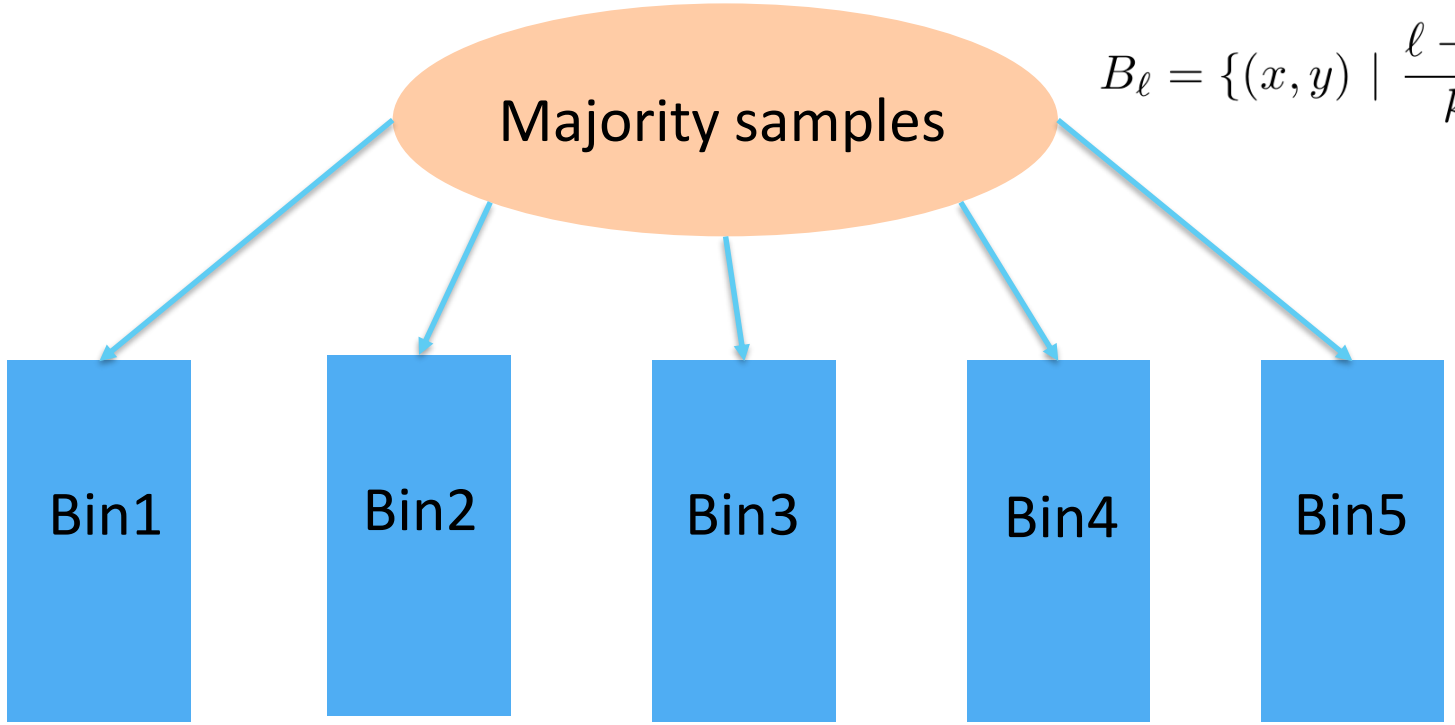
Definition: We use the symbol **H** to denote the **classification hardness** function, where H can be any “decomposable” error function.

Advantages:

- It fills the gap between the **imbalance ratio** and the **task difficulty**.
- The classification hardness also fills the gap between **data sampling strategy** and the **classifiers’ capacity**.

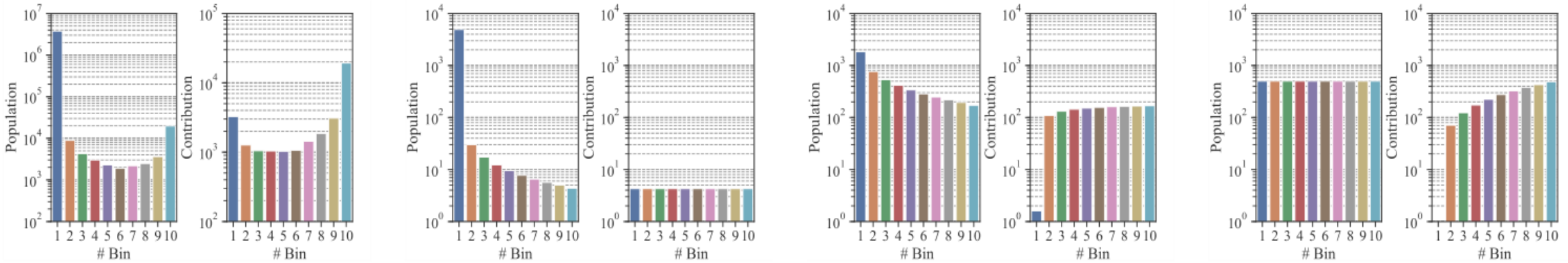
# Self-paced Under-sampling

*Hardness Harmonize*: We split all the majority samples into **k bins** according to their **hardness values(H)**. Each bin indicates a specific hardness level. We then under-sample the majority instances into a balanced dataset by keeping the total hardness contribution in each bin as the same.




$$B_\ell = \{(x, y) \mid \frac{\ell - 1}{k} \leq \mathcal{H}(x, y, F) < \frac{\ell}{k}\} \; w.l.o.g. \; \mathcal{H} \in [0, 1]$$

# Self-paced Under-sampling




(a) Original majority set $\mathcal{N}$ (b) $\alpha = 0$ (c) $\alpha = 0.1$ (d) $\alpha \to \infty$

Fig. 3. An example to visualize how self-paced factor $\alpha$ controls the self-paced under-sampling. The left part of each subfigure shows the number of samples in each bin, the right part shows the hardness contribution from each bin. Subfigure (a) is the distribution over all majority instances. (b)(c)(d) are the distribution over subsets under-sampled by our mechanism when $\alpha = 0$, $\alpha = 0.1$, and $\alpha \to \infty$, respectively. Note that the y-axis uses log scale since the number of samples within different hardness bins can differ by orders of magnitude.

We use Bℓ to denote the ℓ-th bin: $B_\ell = \{(x, y) \mid \frac{\ell - 1}{k} \leq \mathcal{H}(x, y, F) < \frac{\ell}{k}\}$ $w.l.o.g.$ $\mathcal{H} \in [0, 1]$

# Algorithm



**Algorithm 1:** Self-paced Ensemble

**Input:** Training set $\mathcal{D}$, hardness function $\mathcal{H}$, base classifier $f$, number of base classifiers $n$, number of bins $k$,

**Initialize:** $\mathcal{P} \Leftarrow$ minority in $\mathcal{D}$, $\mathcal{N} \Leftarrow$ majority in $\mathcal{D}$

Train classifier $f_0$ using random under-sample majority subsets $\mathcal{N}_0'$ and $\mathcal{P}$, where $|\mathcal{N}_0'| = |\mathcal{P}|$.

**for** *i=1 to n* **do**

  Ensemble $F_i(x) = \frac{1}{i}\sum_{j=0}^{i-1} f_j(x)$

  Cut majority set into $k$ bins w.r.t. $\mathcal{H}(x, y, F_i)$: $B_1, B_2, \cdots, B_k$

  Average hardness contribution in $\ell$-th bin: $h_\ell = \sum_{s \in B_\ell} \mathcal{H}(x_s, y_s, F_i)/|B_\ell|, \ \forall \ell = 1, \ldots, k$

  Update self-paced factor $\alpha = tan(\frac{i\pi}{2n})$

  Unnormalized sampling weight of $\ell$-th bin: $p_\ell = \frac{1}{h_\ell + \alpha}, \forall\ \ell = 1, \ldots, k$

  Under-sample from $\ell$-th bin with $\frac{p_\ell}{\sum_m p_m} \cdot |\mathcal{P}|$ samples

  Train $f_i$ using newly under-sampled subset

**end**

**return** *final ensemble* $F(x) = \frac{1}{n}\sum_{m=1}^{n} f_m(x)$

# Experiments



TABLE II
GENERALIZED PERFORMANCE (AUCPRC) ON CHECKER BOARD DATASET.

| Model | Hyper | RandUnder | Clean | SMOTE | Easy$_{10}$ | Cascade$_{10}$ | SPE$_{10}$ |
|---|---|---|---|---|---|---|---|
| KNN | k_neighbors=5 | 0.281±0.003 | 0.382±0.000 | 0.271±0.003 | 0.411±0.003 | 0.409±0.005 | **0.498**±0.004 |
| DT | max_depth=10 | 0.236±0.010 | 0.365±0.001 | 0.299±0.007 | 0.463±0.009 | 0.376±0.052 | **0.566**±0.011 |
| MLP | hidden_unit=128 | 0.562±0.017 | 0.138±0.035 | 0.615±0.009 | 0.610±0.004 | 0.582±0.005 | **0.656**±0.005 |
| SVM | C=1000 | 0.306±0.003 | 0.405±0.000 | 0.324±0.002 | 0.386±0.001 | 0.456±0.010 | **0.518**±0.004 |
| AdaBoost$_{10}$ | n_estimator=10 | 0.226±0.019 | 0.362±0.000 | 0.297±0.004 | 0.487±0.017 | 0.391±0.013 | **0.570**±0.008 |
| Bagging$_{10}$ | n_estimator=10 | 0.273±0.002 | 0.401±0.000 | 0.316±0.003 | 0.436±0.004 | 0.389±0.007 | **0.568**±0.005 |
| RandForest$_{10}$ | n_estimator=10 | 0.260±0.004 | 0.229±0.000 | 0.306±0.011 | 0.454±0.005 | 0.402±0.012 | **0.572**±0.003 |
| GBDT$_{10}$ | boost_rounds=10 | 0.553±0.015 | 0.602±0.000 | 0.591±0.008 | 0.645±0.006 | 0.648±0.009 | **0.680**±0.003 |

# Experiments

TABLE IV
GENERALIZED PERFORMANCE ON 5 REAL-WORLD DATASETS.

| Dataset | Model | Metric | RandUnder | Clean | SMOTE | Easy$_{10}$ | Cascade$_{10}$ | SPE$_{10}$ |
|---|---|---|---|---|---|---|---|---|
| Credit Fraud | KNN | AUCPRC | 0.052±0.002 | 0.677±0.000 | 0.352±0.000 | 0.162±0.012 | 0.676±0.015 | **0.752**±0.018 |
| | | F1 | 0.112±0.007 | 0.821±0.000 | 0.559±0.001 | 0.250±0.020 | 0.792±0.023 | **0.843**±0.016 |
| | | GM | 0.228±0.009 | 0.822±0.000 | 0.593±0.001 | 0.399±0.025 | 0.810±0.001 | **0.852**±0.002 |
| | | MCC | 0.222±0.014 | 0.822±0.000 | 0.592±0.001 | 0.650±0.004 | 0.815±0.006 | **0.855**±0.006 |
| | DT | AUCPRC | 0.014±0.001 | 0.598±0.013 | 0.088±0.011 | 0.339±0.039 | 0.592±0.029 | **0.783**±0.015 |
| | | F1 | 0.032±0.002 | 0.767±0.004 | 0.177±0.006 | 0.478±0.021 | 0.737±0.023 | **0.838**±0.021 |
| | | GM | 0.119±0.003 | 0.778±0.006 | 0.303±0.017 | 0.548±0.048 | 0.749±0.011 | **0.843**±0.007 |
| | | MCC | 0.124±0.001 | 0.780±0.008 | 0.310±0.003 | 0.409±0.015 | 0.778±0.049 | **0.831**±0.008 |
| | MLP | AUCPRC | 0.225±0.050 | 0.001±0.000 | 0.527±0.017 | 0.605±0.016 | 0.738±0.009 | **0.747**±0.011 |
| | | F1 | 0.388±0.047 | 0.003±0.000 | 0.725±0.013 | 0.762±0.023 | 0.803±0.004 | **0.811**±0.010 |
| | | GM | 0.494±0.040 | 0.040±0.000 | 0.665±0.060 | 0.748±0.019 | 0.806±0.007 | **0.828**±0.003 |
| | | MCC | 0.178±0.008 | 0.000±0.000 | 0.718±0.006 | 0.705±0.004 | 0.744±0.046 | **0.826**±0.008 |
| KDDCUP (DOS vs. PRB) | AdaBoost$_{10}$ | AUCPRC | 0.930±0.012 | - - - | - - - | 0.995±0.002 | **1.000**±0.000 | **1.000**±0.000 |
| | | F1 | 0.962±0.001 | - - - | - - - | 0.997±0.000 | **0.999**±0.000 | **0.999**±0.000 |
| | | GM | 0.964±0.001 | - - - | - - - | 0.997±0.001 | 0.998±0.000 | **0.999**±0.000 |
| | | MCC | 0.956±0.004 | - - - | - - - | 0.992±0.001 | 0.993±0.003 | **0.999**±0.000 |
| KDDCUP (DOS vs. R2L) | AdaBoost$_{10}$ | AUCPRC | 0.034±0.005 | - - - | - - - | 0.108±0.011 | 0.945±0.005 | **0.999**±0.001 |
| | | F1 | 0.050±0.005 | - - - | - - - | 0.259±0.058 | 0.965±0.005 | **0.991**±0.003 |
| | | GM | 0.164±0.011 | - - - | - - - | 0.329±0.015 | 0.967±0.008 | **0.988**±0.004 |
| | | MCC | 0.175±0.016 | - - - | - - - | 0.214±0.004 | 0.905±0.056 | **0.986**±0.004 |
| Record Linkage | GBDT$_{10}$ | AUCPRC | 0.988±0.011 | - - - | - - - | 0.999±0.000 | **1.000**±0.000 | **1.000**±0.000 |
| | | F1 | 0.995±0.000 | - - - | - - - | 0.996±0.000 | **0.998**±0.000 | **0.998**±0.000 |
| | | GM | 0.994±0.002 | - - - | - - - | 0.996±0.000 | **0.998**±0.000 | **0.998**±0.000 |
| | | MCC | 0.780±0.000 | - - - | - - - | 0.884±0.000 | 0.940±0.000 | **0.998**±0.000 |
| Payment Simulation | GBDT$_{10}$ | AUCPRC | 0.278±0.030 | - - - | - - - | 0.676±0.058 | 0.776±0.004 | **0.944**±0.001 |
| | | F1 | 0.446±0.030 | - - - | - - - | 0.709±0.021 | 0.851±0.003 | **0.885**±0.001 |
| | | GM | 0.530±0.020 | - - - | - - - | 0.735±0.011 | 0.851±0.001 | **0.885**±0.001 |
| | | MCC | 0.290±0.023 | - - - | - - - | 0.722±0.015 | 0.856±0.002 | **0.876**±0.001 |

# THANKS