Long-Tailed Recognition via Weight Balancing

Shaden Alshammari* Yu-Xiong Wang♮ Deva Ramanan♯,† Shu Kong♯
*MIT ♮UIUC †Argo AI ♯CMU
shaden@mit.edu yxw@illinois.edu {deva, shuk}@andrew.cmu.edu

CVPR 2022

The key to addressing Long-Tailed Recognition is to balance various aspects including:

- **Data distribution**;

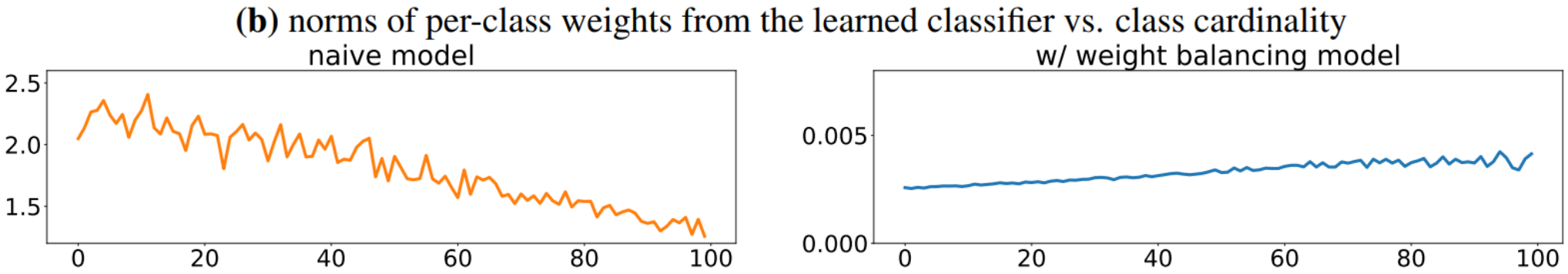
Balancing per-class data distributions during training by up-sampling rare classes or down-sampling common classes[1].

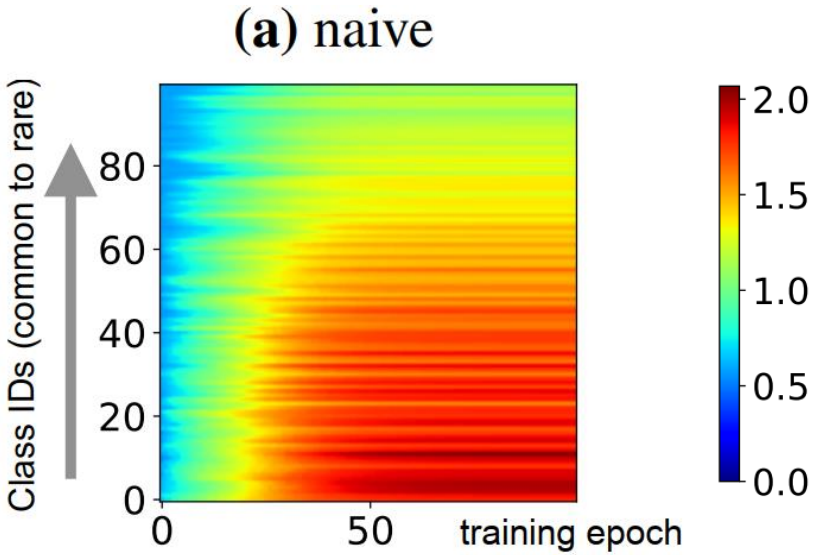
- **Training losses or gradient**;

Balancing the losses[2] or gradients[3] during training.

[1] Chawla N V, Bowyer K W, Hall L O, et al. SMOTE: synthetic minority over-sampling technique, In Journal of artificial intelligence research, 2002.
[2] Cao K, Wei C, Gaidon A, et al. Learning imbalanced datasets with label-distribution-aware margin loss, In NeurIPS 2019.
[3] Khan S H, Hayat M, et al. Cost-sensitive learning of deep feature representations from imbalanced data, In TNNLS 2017.

Motivation





A naively trained model on long-tailed class distributed data has large weights for head classes[4].

Weight balancing

[4] Kang B, Xie S, Rohrbach M, et al. Decoupling representation and classifier for long-tailed recognition, In ICLR 2019 .

Weight Balancing Techniques

Objective: $$\Theta^* = \arg\min_{\Theta} F(\Theta; \mathcal{D}) \equiv \sum_{i=1}^{N} \ell\big(f(\mathbf{x}_i; \Theta), y_i\big).$$

- L2-Normalization:

$$\Theta^* = \arg\min_{\Theta} F(\Theta; \mathcal{D}), \quad s.t. \quad \|\boldsymbol{\theta}_k\|_2^2 = 1, \ \forall\, k.$$

- Weight Decay:

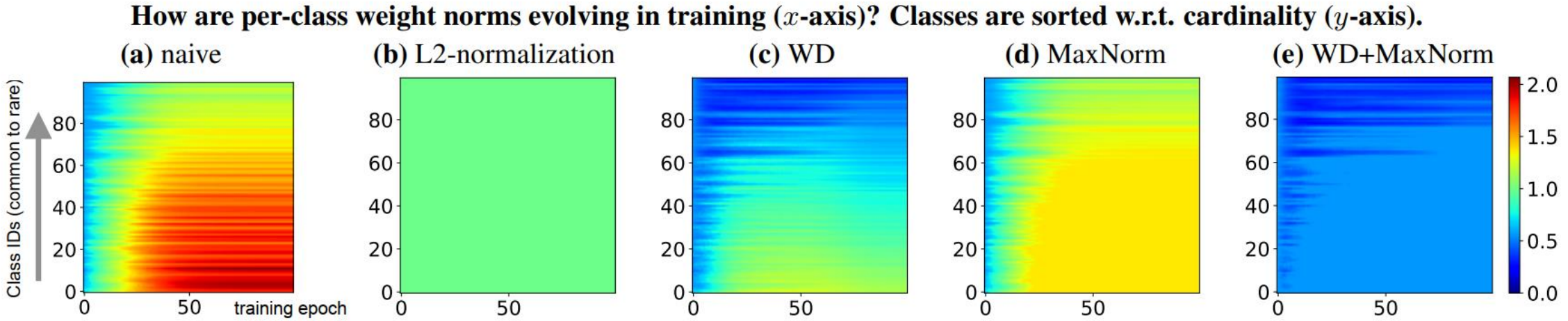
$$\Theta^* = \arg\min_{\Theta} F(\Theta; \mathcal{D}) + \lambda \sum_k \|\boldsymbol{\theta}_k\|_2^2,$$

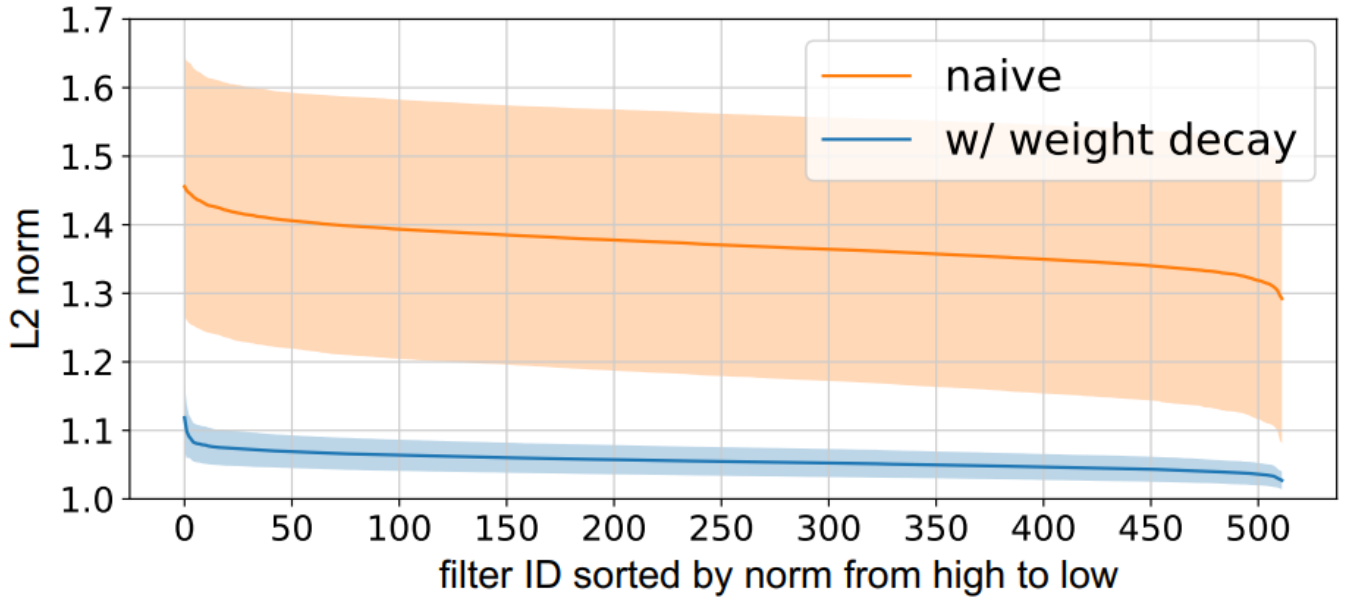
- MaxNorm:

$$\Theta^* = \arg\min_{\Theta} F(\Theta; \mathcal{D}), \quad s.t. \quad \|\boldsymbol{\theta}_k\|_2^2 \le \delta^2, \ \forall\, k,$$

$$\Theta^* = \arg\min_{\Theta} \max_{\gamma \ge 0} F(\Theta; \mathcal{D}) + \sum_k \gamma(\|\boldsymbol{\theta}_k\|_2^2 - \delta),$$

Weight Balancing Techniques





While individual filters in the hidden layers are not class-specific by design, recent work demonstrates that certain filters tend to fire on certain classes[5]!

[5] Bau D, Zhou B, Khosla A, et al. Network dissection: Quantifying interpretability of deep visual representations, In CVPR 2017.

Analysis

How and why the regularizers work for long-tailed recognition?

◆ **Weight Decay** and **MaxNorm**:

Weight decay encourages learning small weights, and **MaxNorm** encourages weights to grow within a norm ball but cap them when their norms exceed the radius. They have complementary advantages:

(1) Weight Decay on the small weights improves their generalization and reduces overfitting;

(2) MaxNorm prevents the large weights from dominating the training.

◆ Extreme cases:

(1) When $\delta \to \infty$ in **MaxNorm**, it down to the naïve training;

(2) A sufficiently small δ encourages all the weights to be close to the surface of the norm-ball.

◆ **Weight Decay** can easily balance all network weights:

(1) Don’t need to separate per-class filters;

(2) **MaxNorm** can also be applied to all layers, but it requires setting per-layer thresholds, which can be time-consuming.

Methods

First Stage:

Feature learning: train a network by using the **cross-entropy loss** and tuning **weight decay**;

Second Stage:

Classifier learning: train a classifier over the learned features using a **class-balanced loss**[6], **weight decay**, and **MaxNorm**.

[6] Cui Y, Jia M, Lin T Y, et al. Class-balanced loss based on effective number of samples, In CVPR 2019.

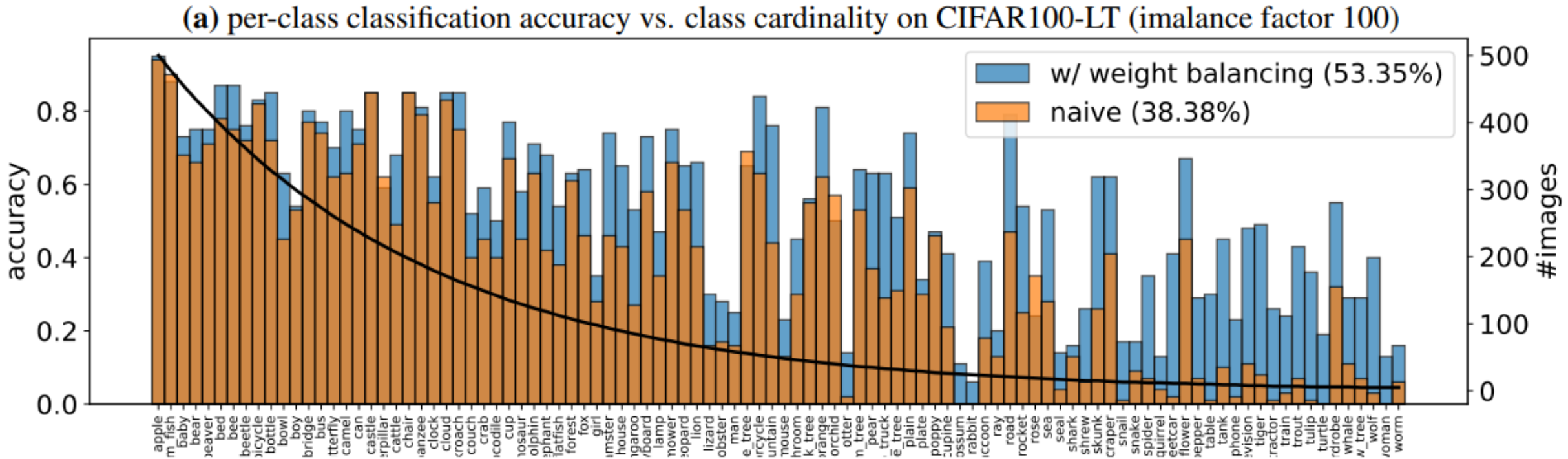
Experiment

CIFAR 100-LT

imbalance factor	100	50	10
CE [19]	38.32	43.85	55.71
CE+CB [19]	39.60	45.32	57.99
KD [33]	40.36	45.49	59.22
LDAM-DRW [12]	42.04	46.62	58.71
BBN [88]	42.56	47.02	59.12
LogitAjust [54]	42.01	47.03	57.74
LDAM+SSP [78]	43.43	47.11	58.91
Focal [47]	38.41	44.32	55.78
Focal+CB [19]	39.60	45.17	57.99
De-confound [71]	44.10	50.30	59.60
τ-norm [38]	47.73	52.53	63.80
SSD [46]	46.00	50.50	62.30
DiVE [32]	45.35	51.13	62.00
DRO-LT [65]	47.31	57.57	63.41
PaCo [17]	52.00	56.00	64.20
ACE (4-expert) [10]	49.60	51.90	—
RIDE (4-expert) [73]	49.10	—	—
Our methods (weight balancing)			
naive	38.38	43.99	57.31
WD	46.08	52.71	66.03
+ L2norm	49.60	56.33	67.16
+ τ-norm	51.31	57.65	67.79
+ WD	52.42	57.47	67.96
+ Max	50.24	56.06	67.10
+ WD & Max	**53.35**	**57.71**	**68.67**

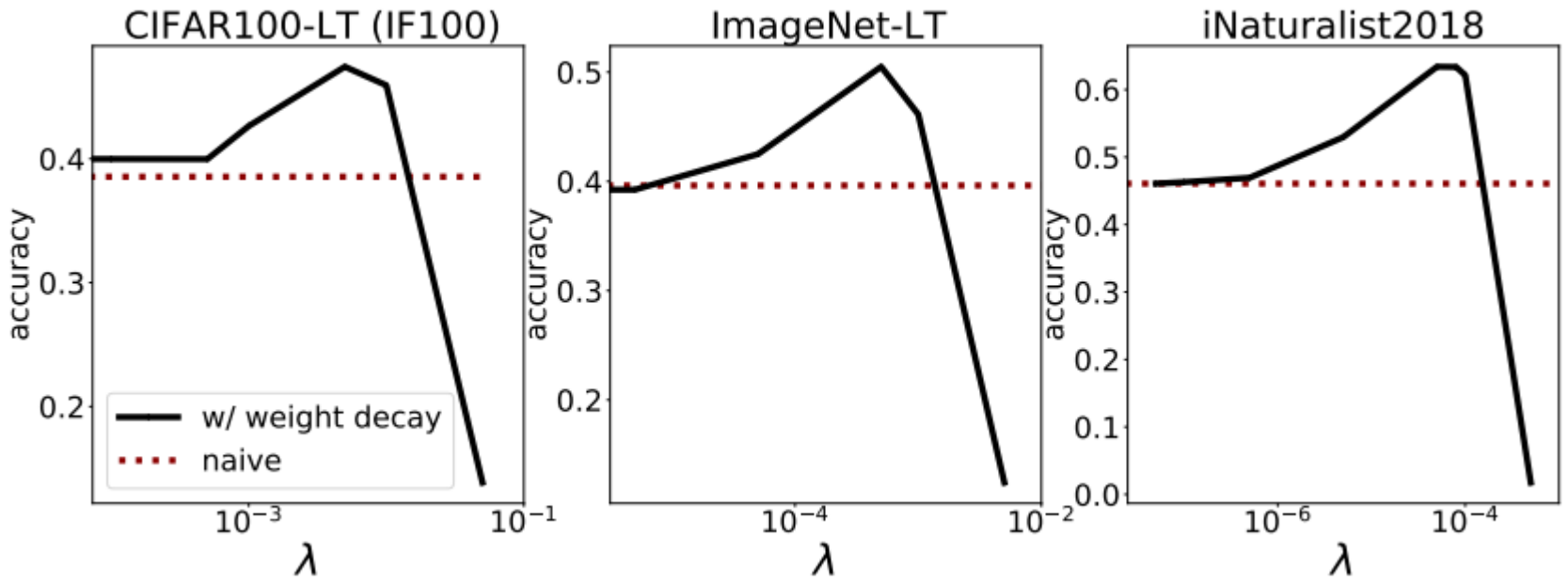
	ImageNet-LT				iNaturalist			
	Many	Med.	Few	All	Many	Med.	Few	All
CE [38]	65.9	37.5	7.7	44.4	72.2	63.0	57.2	61.7
CE+CB [19]	39.6	32.7	16.8	33.2	53.4	54.8	53.2	54.0
KD [33]	58.8	26.6	3.4	35.8	72.6	63.8	57.4	62.2
Focal [19]	36.4	29.9	16.0	30.5	—	—	—	61.1
OLTR [49]	43.2	35.1	18.5	35.6	59.0	64.1	64.9	63.9
LFME [76]	47.1	35.0	17.5	37.2	—	—	—	—
BBN [88]	—	—	—	—	49.4	70.8	65.3	66.3
cRT [38]	61.8	46.2	27.3	49.6	69.0	66.0	63.2	65.2
τ-norm [38]	59.1	46.9	30.7	49.4	65.6	65.3	65.5	65.6
De-confound [71]	62.7	48.8	31.6	51.8	—	—	—	—
DiVE [32]	64.1	50.4	31.5	53.1	70.6	70.0	67.6	69.1
DRO-LT [65]	64.0	49.8	33.1	53.5	—	—	—	69.7
DisAlign [83]	61.3	**52.2**	31.4	52.9	69.0	**71.1**	**70.2**	**70.6**
Our methods (weight balancing)								
naive	55.3	31.4	12.5	38.0	54.7	46.0	43.9	46.1
WD	**68.5**	42.4	14.2	48.6	**74.5**	66.5	61.5	65.4
+ L2norm	61.2	48.9	**42.6**	52.8	11.2	47.4	66.9	51.3
+ τ-norm	64.0	49.0	36.3	53.1	71.3	69.8	68.9	69.6
+ WD	62.0	49.7	41.0	53.3	71.0	70.3	69.4	70.0
+ Max	62.2	50.1	37.5	53.0	71.4	68.9	69.1	69.2
+ WD & Max	62.5	50.4	41.5	**53.9**	71.2	70.4	69.7	70.2
SOTA with "bells and whistles": ensembles, data augmentation, and self-supervised pretraining								
RIDE [73]	67.9	52.3	36.0	56.1	66.5	72.1	71.5	71.3
ACE [10]	—	—	—	56.6	—	—	—	72.9
SSD [46]	66.8	53.1	35.4	56.0	—	—	—	71.5
PaCo [17]	63.2	51.6	39.2	54.4	69.5	72.3	73.1	72.3

Experiment

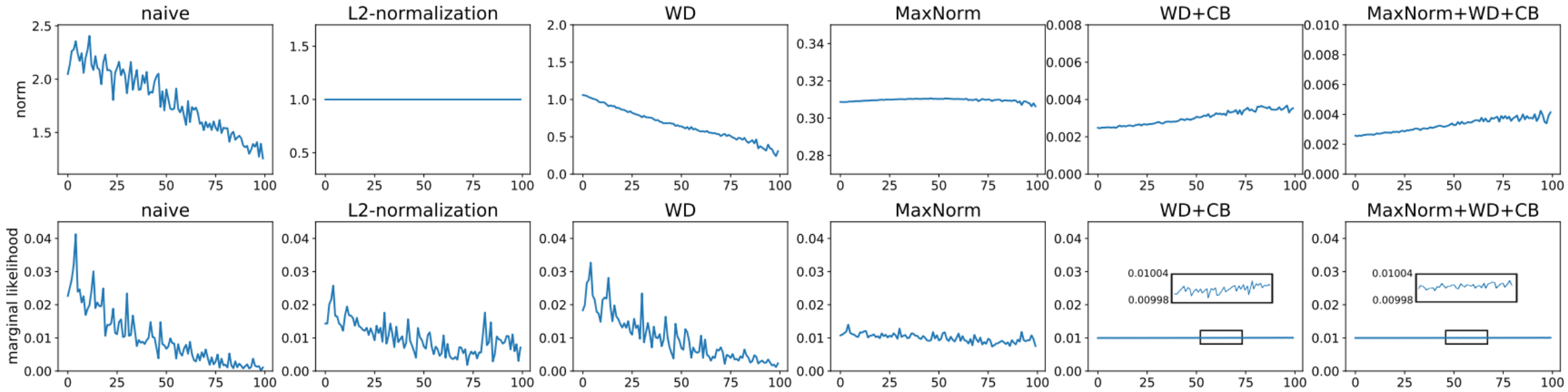

(a) per-class classification accuracy vs. class cardinality on CIFAR100-LT (imalance factor 100)
w/ weight balancing (53.35%)
naive (38.38%)
accuracy
#images
0.0
0.2
0.4
0.6
0.8
0
100
200
300
400
500

Ablation Study

Simply use CE loss and tune weight decay λ to regularize all network weight.



Ablation Study



L2-normalization that “perfectly” balances classifier weights does not produce balanced marginal likelihood!

Ablation Study

CIFAR100-LT (IF=100)

Model	Many	Medium	Few	All
on the last layer (classifier)				
WD=0 (w/ CE)	64.05	35.80	11.43	38.38
+ τ-norm	59.54	38.23	25.93	42.00
WD tuned (w/ CE)	76.94	44.28	12.17	46.08
+ τ-norm	73.11	47.69	30.10	51.31
+ L2norm	76.09	47.74	20.87	49.60
+ CE & L2norm	76.37	48.11	21.00	49.87
+ CE & WD	76.97	45.94	14.00	47.22
+ CB	**77.00**	45.89	13.60	47.09
+ CB & L2norm	76.43	48.20	21.60	50.10
+ CB & WD	72.77	49.74	31.80	52.42
+ CB & Max	76.49	49.23	20.67	50.20
+ CB & WD & Max	72.60	**51.86**	32.63	53.35
on the last two layers				
+ CB & WD & Max	71.37	51.17	**35.53**	**53.55**

Exploring Weight Balancing on Long-Tailed Recognition Problem

Naoya Hasegawa & Issei Sato
The University of Tokyo
{hasegawa-naoya410, sato}@g.ecc.u-tokyo.ac.jp

ICLR 2024

Motivation

The reason why Weight Balancing leads to a significant improvement in LTR performance remains unclear.

Conclusion:

1st stage: WD and CE increase the Fisher's discriminant ratio (FDR) of features.
– Degrade the inter-class cosine similarities;
– Decrease the scaling parameters of batch normalization (BN) ,This has a positive effect on feature training;
– Facilitate improvement of FDR as features pass through layers.

1st stage: WD increases the norms of features for tail classes.

2nd stage: WD and CB perform implicit logit adjustment (LA) by making the norm of classifier's weights higher for tail classes. This stage does not work well for datasets with a small class number.

FDR is the ratio of the inter-class variance to the inner-class variance and indicates the ease of linear separation of the features.

The FDR of training features is $Tr(S_W^{-1}S_B)$

$$S_B = \sum_{k=1}^{C} N_k(\mu_k - \mu)(\mu_k - \mu)^\top$$

$$S_W = \sum_{k=1}^{C} \sum_{x_i \in D_k} (x_i - \mu_k)(x_i - \mu_k)^\top$$

ETF(Equiangular Tight Frame) classifier. Neural Collapse(NC) indicates that the matrix of classifier weights in deep learning models converges to an ETF when trained with CE[7].

The idea of ETF classifiers is to train only the feature extractor by fixing the linear layer to an ETF from the initial step.

ETF classifiers' weights $W \in \mathbb{R}^{d\times C}$ satisfy $W = \sqrt{E_W \frac{C}{C-1}} U(I_C - \frac{1}{C} 1_C 1_C^\intercal)$, where $U \in R^{d\times C}$ is a matrix such that $U^\intercal U$ is an identity matrix, $I_C \in R^{C\times C}$ is an identity matrix, and 1_C is a C-dimensional vector, all elements in which are 1.

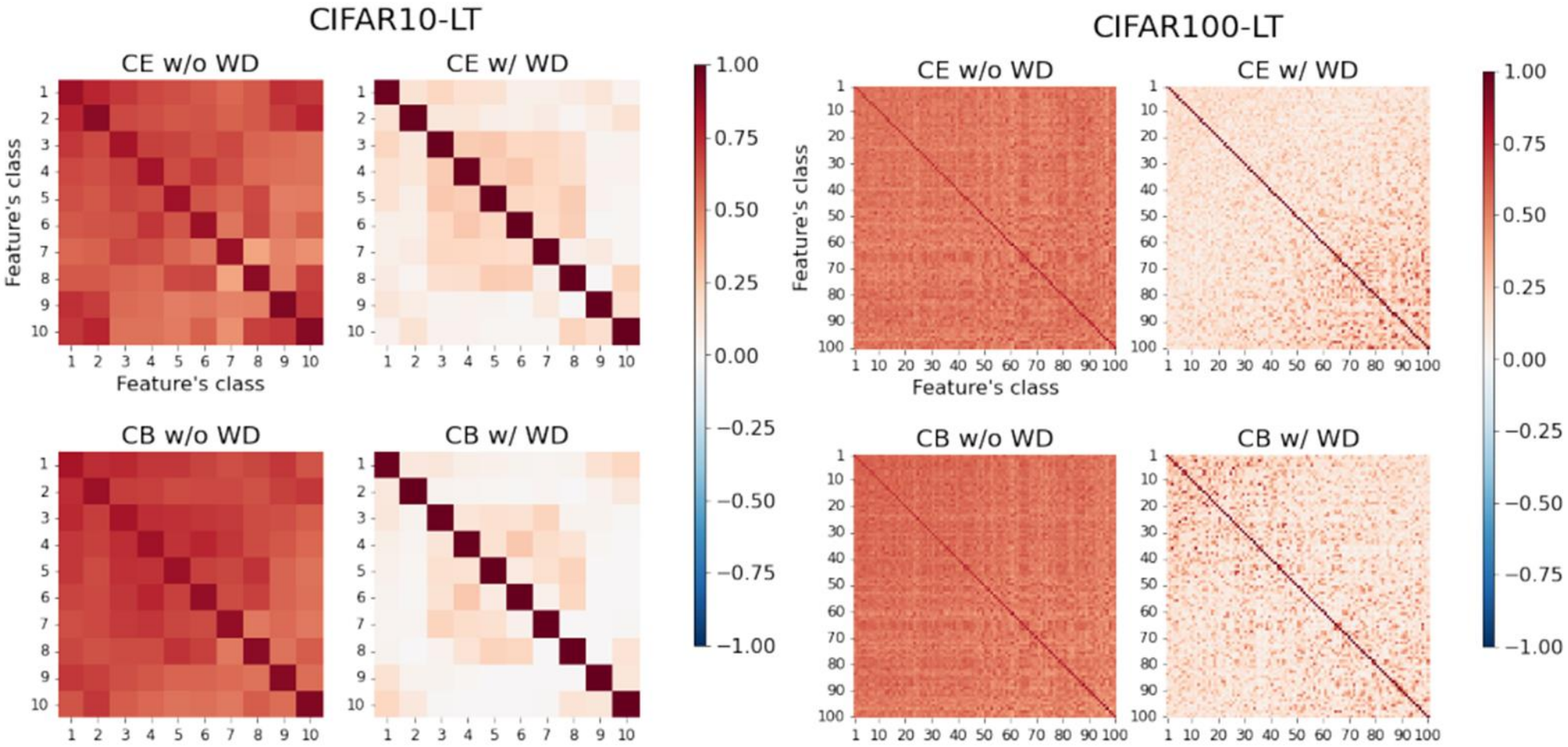
[7] Papyan V, Han X Y, Donoho D L. Prevalence of neural collapse during the terminal phase of deep learning training[J]. In Proceedings of the National Academy of Sciences, 2020.

Weight Decay and Cross-Entropy degrade inter-class cosine similarities.

Table 1: FDRs for each dataset of models trained with each method. A higher FDR indicates that features are more easily linearly separable. For all datasets, the method with WD or CE produces a higher FDR and using both results in the highest FDR.

	CIFAR10-LT		CIFAR100-LT		mini-ImageNet-LT	
Method	Train	Test	Train	Test	Train	Test
CE w/o WD	8.17×10^1	2.17×10^1	1.28×10^2	4.16×10^1	7.56×10^1	4.28×10^1
CB w/o WD	4.43×10^1	1.50×10^1	8.17×10^1	2.42×10^1	4.68×10^1	2.93×10^1
CE w/ WD	$\mathbf{2.60 \times 10^3}$	$\mathbf{3.89 \times 10^1}$	$\mathbf{2.87 \times 10^4}$	$\mathbf{1.07 \times 10^2}$	$\mathbf{6.58 \times 10^2}$	$\mathbf{1.01 \times 10^2}$
CB w/ WD	3.39×10^2	3.04×10^1	2.12×10^4	6.74×10^1	4.84×10^2	6.84×10^1
WD w/o BN	2.19×10^2	3.42×10^1	5.77×10^2	7.95×10^1	1.61×10^2	6.73×10^1
WD fixed BN	$\mathbf{1.63 \times 10^3}$	$\mathbf{4.16 \times 10^1}$	$\mathbf{2.04 \times 10^4}$	$\mathbf{1.05 \times 10^2}$	$\mathbf{3.94 \times 10^2}$	$\mathbf{1.04 \times 10^2}$

Weight Decay and Cross-Entropy degrade inter-class cosine similarities.



Weight Decay and Cross-Entropy degrade inter-class cosine similarities.

The cone effect is a phenomenon in which features from a DNN tend to have high cosine similarities to each other even if they belong to different classes[8].

Theorem 1. *For all* $(\mathbf{x}_i, y_i), (\mathbf{x}_j, y_j) \in \mathcal{D}$ *s.t.* $y_i \neq y_j$, *if* $\mathbf{W}$ *is an ETF and there exists* ϵ *and* L *s.t.* $\left\| \frac{\partial \ell_{\mathrm{CE}}}{\partial \boldsymbol{g}(\mathbf{x}_i)} \right\|_2, \left\| \frac{\partial \ell_{\mathrm{CE}}}{\partial \boldsymbol{g}(\mathbf{x}_j)} \right\|_2 \leq \epsilon < \frac{1}{C}$ *and* $\|\boldsymbol{g}(\mathbf{x}_i)\|_2, \|\boldsymbol{g}(\mathbf{x}_j)\|_2 \leq L \leq 2\sqrt{2}\log(C-1)$, *the following holds:*

$$\cos(\boldsymbol{g}(\mathbf{x}_i), \boldsymbol{g}(\mathbf{x}_j)) \leq 2\delta\sqrt{1-\delta^2}, \tag{2}$$

where $\delta \equiv \frac{1}{L}\frac{C-1}{C}\log\left(\frac{(C-1)(1-\epsilon)}{\epsilon}\right) \in \left(\frac{1}{\sqrt{2}}, 1\right]$ *and* $\cos(\cdot,\cdot)$ *means cosine similarity of the two vectors.*

The cone effect is suppressed when the following two conditions hold:

1. the weight matrix of the linear layer is an ETF;
2. the norms of the features are sufficiently small.

[8] Liang V W, Zhang Y, Kwon Y, et al. Mind the gap: Understanding the modality gap in multi-modal contrastive representation learning, In NeurIPS 2022.

Weight Decay and Cross-Entropy Decrease scaling parameters of BN.

The effect of weight decay on the model is split into that on the **convolution layers** and that on the **BN layers**.

	CIFAR10-LT		CIFAR100-LT		mini-ImageNet-LT	
Method	Train	Test	Train	Test	Train	Test
CE w/o WD	8.17×10^1	2.17×10^1	1.28×10^2	4.16×10^1	7.56×10^1	4.28×10^1
CB w/o WD	4.43×10^1	1.50×10^1	8.17×10^1	2.42×10^1	4.68×10^1	2.93×10^1
CE w/ WD	$\mathbf{2.60 \times 10^3}$	$\mathbf{3.89 \times 10^1}$	$\mathbf{2.87 \times 10^4}$	$\mathbf{1.07 \times 10^2}$	$\mathbf{6.58 \times 10^2}$	$\mathbf{1.01 \times 10^2}$
CB w/ WD	3.39×10^2	3.04×10^1	2.12×10^4	6.74×10^1	4.84×10^2	6.84×10^1
WD w/o BN	2.19×10^2	3.42×10^1	5.77×10^2	7.95×10^1	1.61×10^2	6.73×10^1
WD fixed BN	$\mathbf{1.63 \times 10^3}$	$\mathbf{4.16 \times 10^1}$	$\mathbf{2.04 \times 10^4}$	$\mathbf{1.05 \times 10^2}$	$\mathbf{3.94 \times 10^2}$	$\mathbf{1.04 \times 10^2}$

Enabling WD for the convolution layers is essential for improving accuracy!

Weight Decay and Cross-Entropy Decrease scaling parameters of BN.

The effect of weight decay on the model is split into that on the **convolution layers** and that on the **BN layers**.

Table 2: Means and standard deviations of all scaling parmeters in BN layers of models trained with

	CIFAR10-LT		CIFAR100-LT		mini-ImageNet-LT	
Method	Train	Test	Train	Test	Train	Test
CE w/o WD	8.17×10^1	2.17×10^1	1.28×10^2	4.16×10^1	7.56×10^1	4.28×10^1
CB w/o WD	4.43×10^1	1.50×10^1	8.17×10^1	2.42×10^1	4.68×10^1	2.93×10^1
CE w/ WD	$\mathbf{2.60 \times 10^3}$	$\mathbf{3.89 \times 10^1}$	$\mathbf{2.87 \times 10^4}$	$\mathbf{1.07 \times 10^2}$	$\mathbf{6.58 \times 10^2}$	$\mathbf{1.01 \times 10^2}$
CB w/ WD	3.39×10^2	3.04×10^1	2.12×10^4	6.74×10^1	4.84×10^2	6.84×10^1
WD w/o BN	2.19×10^2	3.42×10^1	5.77×10^2	7.95×10^1	1.61×10^2	6.73×10^1
WD fixed BN	$\mathbf{1.63 \times 10^3}$	$\mathbf{4.16 \times 10^1}$	$\mathbf{2.04 \times 10^4}$	$\mathbf{1.05 \times 10^2}$	$\mathbf{3.94 \times 10^2}$	$\mathbf{1.04 \times 10^2}$

CE w/o WD	-0.0236 ± 0.0255	-0.0444 ± 0.0348	-0.0904 ± 0.0698
CE w/ WD	0.00510 ± 0.0244	-0.00667 ± 0.0309	-0.0186 ± 0.0448

The improvement in FDR caused by applying WD to BN layers is mainly because smaller scaling parameters have a positive effect on the learning dynamics and improve FDR!

Weight Decay and Cross-Entropy facilitate improvement of FDR as features pass through layers

Table 4: FDRs of features from each model trained with each method and features passed throug randomly initialized linear layer and ReLU after model. C10, C100, and mIm are the abbreviation for CIFAR10, CIFAR100, and mini-ImageNet respectively. Red text indicates higher values tha before, blue text indicates lower values. The FDR of features obtained from models trained with WI improves by passing the features through a random initialized layer and ReLU.

		CE		WD w/o BN		WD	
Model	Dataset	before	after	before	after	before	after
ResNet34	C100	7.51×10^1	7.11×10^1	2.29×10^2	2.34×10^2	3.98×10^2	4.10×10^2
	C100-LT	4.16×10^1	4.02×10^1	7.95×10^1	7.89×10^1	1.07×10^2	1.13×10^2
	C10	4.77×10^1	5.56×10^1	7.96×10^1	1.02×10^2	1.81×10^2	2.35×10^2
	C10-LT	2.17×10^1	2.36×10^1	3.42×10^1	3.94×10^1	3.89×10^1	4.30×10^1
	mIm	7.34×10^1	6.66×10^1	1.81×10^2	1.86×10^2	3.63×10^2	3.93×10^2
	mIm-LT	4.28×10^1	3.97×10^1	6.73×10^1	6.30×10^1	1.01×10^2	1.08×10^2
MLP3	MNIST	1.58×10^2	1.54×10^2	2.48×10^2	3.53×10^2	2.36×10^2	7.96×10^2
MLP4		1.98×10^2	1.96×10^2	3.60×10^2	4.97×10^2	4.12×10^2	1.43×10^3
MLP5		2.44×10^2	2.37×10^2	4.91×10^2	6.76×10^2	6.10×10^2	2.47×10^3
ResBlock1		1.39×10^2	1.36×10^2	2.20×10^2	2.69×10^2	2.44×10^2	5.81×10^2
ResBlock2		1.66×10^2	1.59×10^2	2.97×10^2	3.65×10^2	4.44×10^2	9.55×10^2

Weight Decay increases the norms of features for tail classes

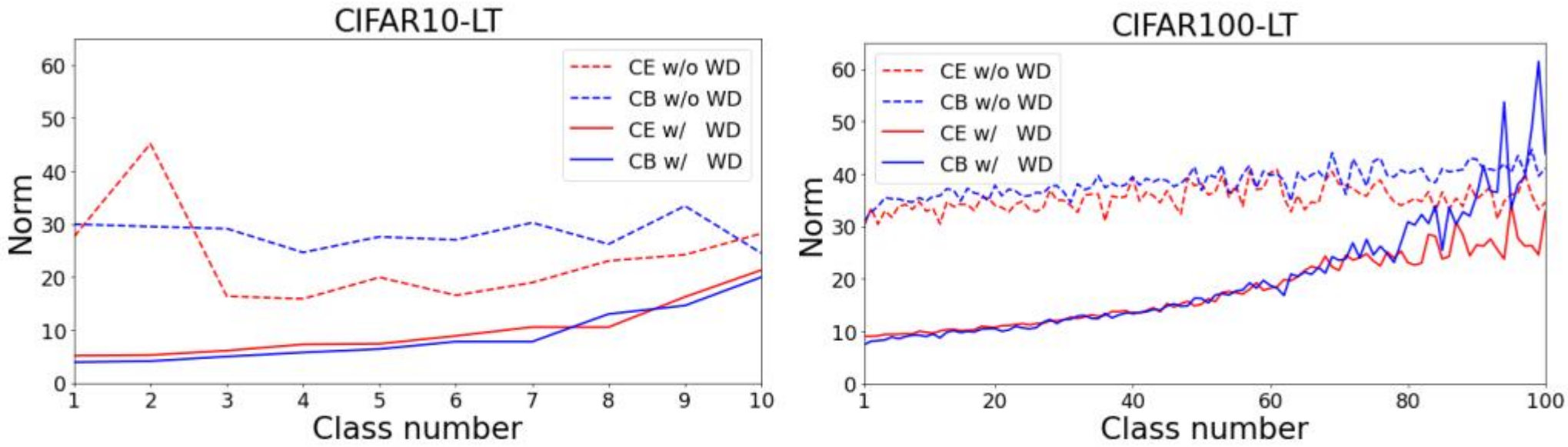


Figure 2: Norm of mean per-class training features produced from models trained with each method. Features learned with methods with WD all demonstrate that the norms of the *Many* classes' features tend to be smaller than those of the *Few* classes.

The method without WD shows almost no relationship between the number of samples and the norm of the features for each class.

Weight Decay perform implicit logit adjustment[9]

Theorem 2. *Assume* $\boldsymbol{\mu} = \mathbf{0}$. *For any* $k \in \mathcal{Y}$, *if there exists* $\mathbf{w}_k^*$ *s.t.* $\left.\frac{\partial F_{\mathrm{WB}}}{\partial \mathbf{w}_k}\right|_{\mathbf{w}_k=\mathbf{w}_k^*} = \mathbf{0}$ *and* $\|\mathbf{w}_k^*\|_2 = O\left(\frac{1}{\lambda\rho C}\right)$, *we have*

$$\forall k \in \mathcal{Y}, \left\|\mathbf{w}_k^* - \frac{\overline{N}}{\lambda N}\boldsymbol{\mu}_k\right\|_2 = O\left(\frac{1}{\lambda^2\rho^2 C^2}\right), \tag{3}$$

If the number of classes or the imbalance factor is sufficiently large, and if NC has occurred in the first stage, there exists linear layer weights that are constant multiples of the corresponding features and sufficiently close to the stationary point in the optimization.

[9] Kim B, Kim J. Adjusting decision boundary for class imbalanced learning, In IEEE Access, 2020.

Is Two-Stage Learning really necessary?

The linear layer is fixed and the feature extractor is trained with CE, WD, and FR. Then, the norm of the linear layer is adjusted with multiplicative LA.

Method	LA	FDR		Accuracy (%)			
		Train	Test	*Many*	*Medium*	*Few*	Average
CE	N/A	$1.28\times10^{2}_{\pm0.9\times10^{1}}$	$4.16\times10^{1}_{\pm1.0\times10^{0}}$	$64.6_{\pm0.9}$	$36.9_{\pm0.8}$	$11.9_{\pm0.7}$	$38.5_{\pm0.6}$
CB	N/A	$8.17\times10^{1}_{\pm8.0\times10^{0}}$	$2.42\times10^{1}_{\pm0.6\times10^{0}}$	$47.6_{\pm1.0}$	$23.2_{\pm0.6}$	$5.61_{\pm0.4}$	$26.1_{\pm0.4}$
WD	N/A	$2.87\times10^{4}_{\pm6.0\times10^{3}}$	$1.07\times10^{2}_{\pm0.2\times10^{1}}$	$75.9_{\pm0.5}$	$45.3_{\pm0.6}$	$13.9_{\pm0.7}$	$46.0_{\pm0.4}$
	Add	$2.87\times10^{4}_{\pm6.0\times10^{3}}$	$1.07\times10^{2}_{\pm0.2\times10^{1}}$	$70.7_{\pm0.9}$	$45.7_{\pm0.9}$	$\mathbf{30.9_{\pm0.9}}$	$49.6_{\pm0.4}$
	Mult	$2.87\times10^{4}_{\pm6.0\times10^{3}}$	$1.07\times10^{2}_{\pm0.2\times10^{1}}$	$72.6_{\pm0.7}$	$48.5_{\pm0.8}$	$29.5_{\pm0.9}$	$50.8_{\pm0.4}$
WB	N/A	$2.94\times10^{4}_{\pm6.0\times10^{3}}$	$1.07\times10^{2}_{\pm0.2\times10^{1}}$	$73.8_{\pm0.7}$	$50.2_{\pm0.6}$	$25.6_{\pm1.0}$	$50.6_{\pm0.2}$
WD&ETF	N/A	$3.33\times10^{4}_{\pm2.0\times10^{3}}$	$1.13\times10^{2}_{\pm0.1\times10^{1}}$	$76.3_{\pm0.3}$	$46.0_{\pm0.4}$	$15.5_{\pm0.6}$	$46.9_{\pm0.2}$
	Add	$3.33\times10^{4}_{\pm2.0\times10^{3}}$	$1.13\times10^{2}_{\pm0.1\times10^{1}}$	$73.8_{\pm0.4}$	$48.9_{\pm0.3}$	$25.8_{\pm0.4}$	$50.2_{\pm0.2}$
	Mult	$3.33\times10^{4}_{\pm2.0\times10^{3}}$	$1.13\times10^{2}_{\pm0.1\times10^{1}}$	$70.4_{\pm0.7}$	$51.4_{\pm0.3}$	$\mathbf{31.7_{\pm0.6}}$	$51.7_{\pm0.3}$
WD&FR &ETF	N/A	$\mathbf{8.81\times10^{4}_{\pm2.0\times10^{3}}}$	$\mathbf{1.22\times10^{2}_{\pm0.1\times10^{1}}}$	$\mathbf{77.9_{\pm0.3}}$	$46.8_{\pm1.0}$	$15.3_{\pm0.3}$	$47.6_{\pm0.5}$
	Add	$\mathbf{8.85\times10^{4}_{\pm2.0\times10^{3}}}$	$\mathbf{1.22\times10^{2}_{\pm0.1\times10^{1}}}$	$75.1_{\pm0.3}$	$49.3_{\pm1.0}$	$26.2_{\pm0.8}$	$50.9_{\pm0.3}$
	Mult	$\mathbf{8.85\times10^{4}_{\pm2.0\times10^{3}}}$	$\mathbf{1.22\times10^{2}_{\pm0.1\times10^{1}}}$	$74.2_{\pm0.3}$	$\mathbf{52.9_{\pm0.9}}$	$29.9_{\pm0.8}$	$\mathbf{53.0_{\pm0.3}}$

CIFAR100-LT

Thanks