Long-Tailed Classification by Keeping the Good and Removing the Bad Momentum Causal Effect

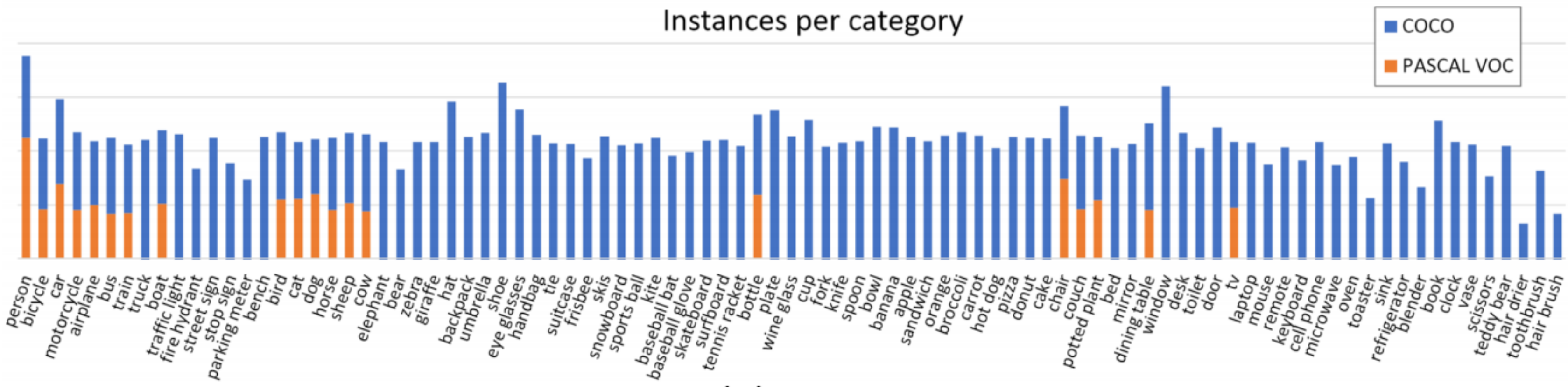
NeurlPS 2020

Catalogue

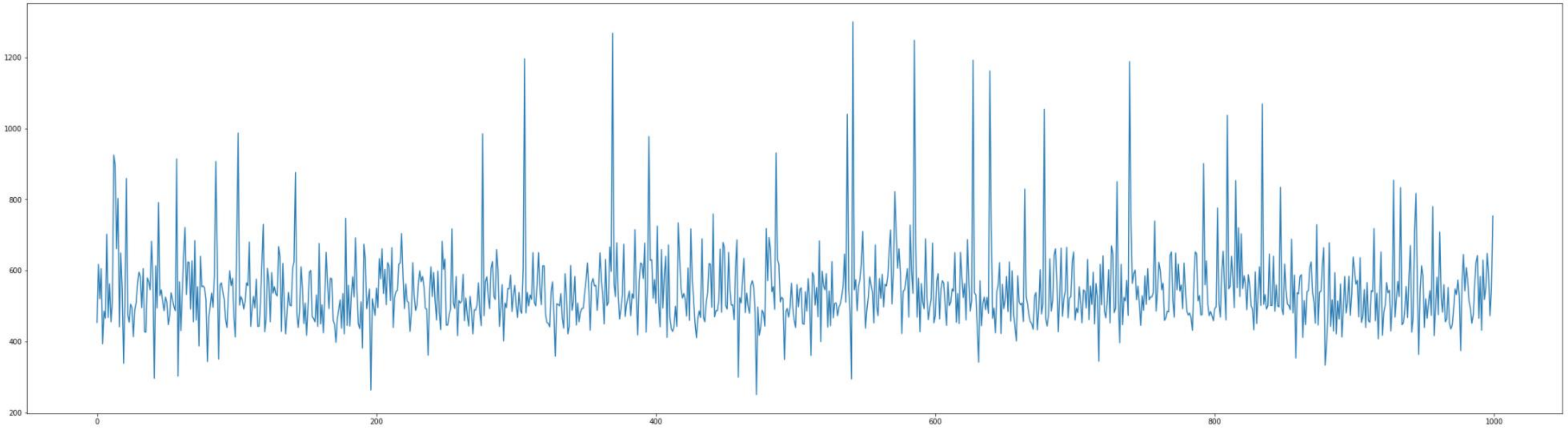
- Background
- Related work
- Introduction
- Method
- Conclusion
- My opinions & questions

Background

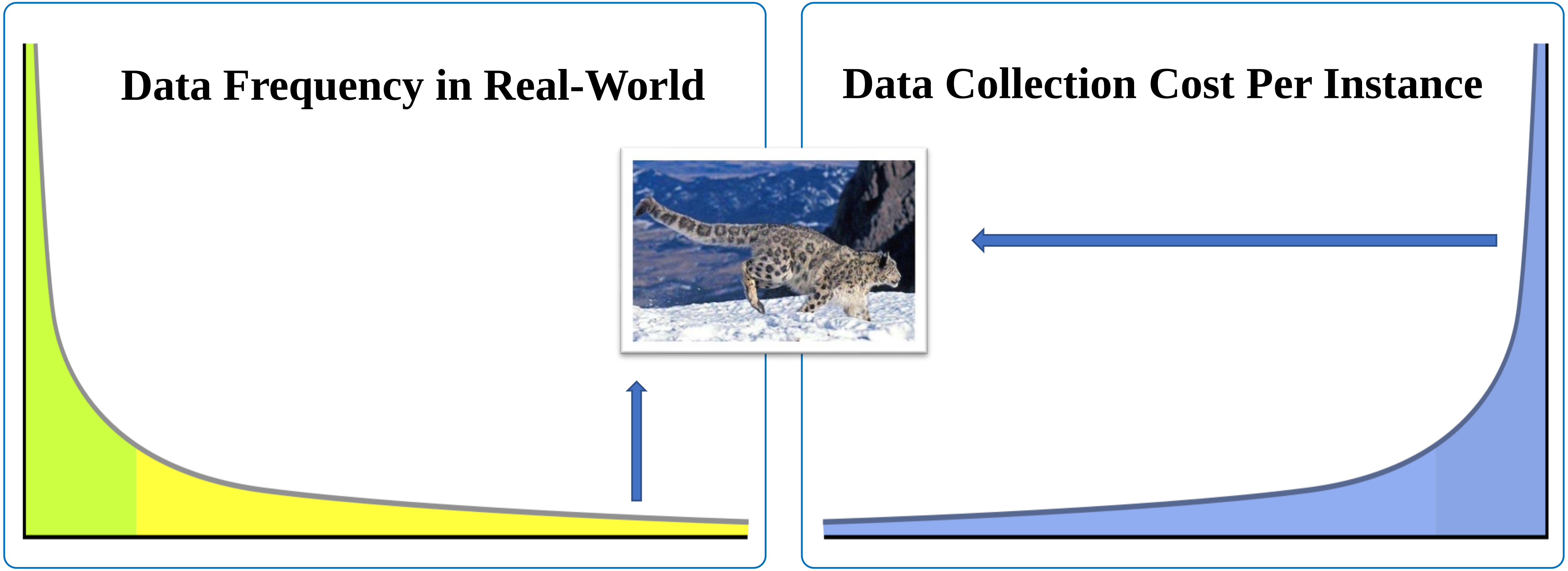
MS-COCO (Object Detection & Instance Segmentation)



ImageNet (Image Classification)



Background


Data Frequency in Real-World
Data Collection Cost Per Instance

Related work

Single-stage Rebalanced learning

- Re-sampling(under-sampling, over-sampling)
- Re-weighting(Focal loss, CB loss…)
- Transfer learning, Domain adaption, Synthetic samples
- Metric learning, Meta learning
- Ensemble
- semi-supervised learning, self-supervised pre-training

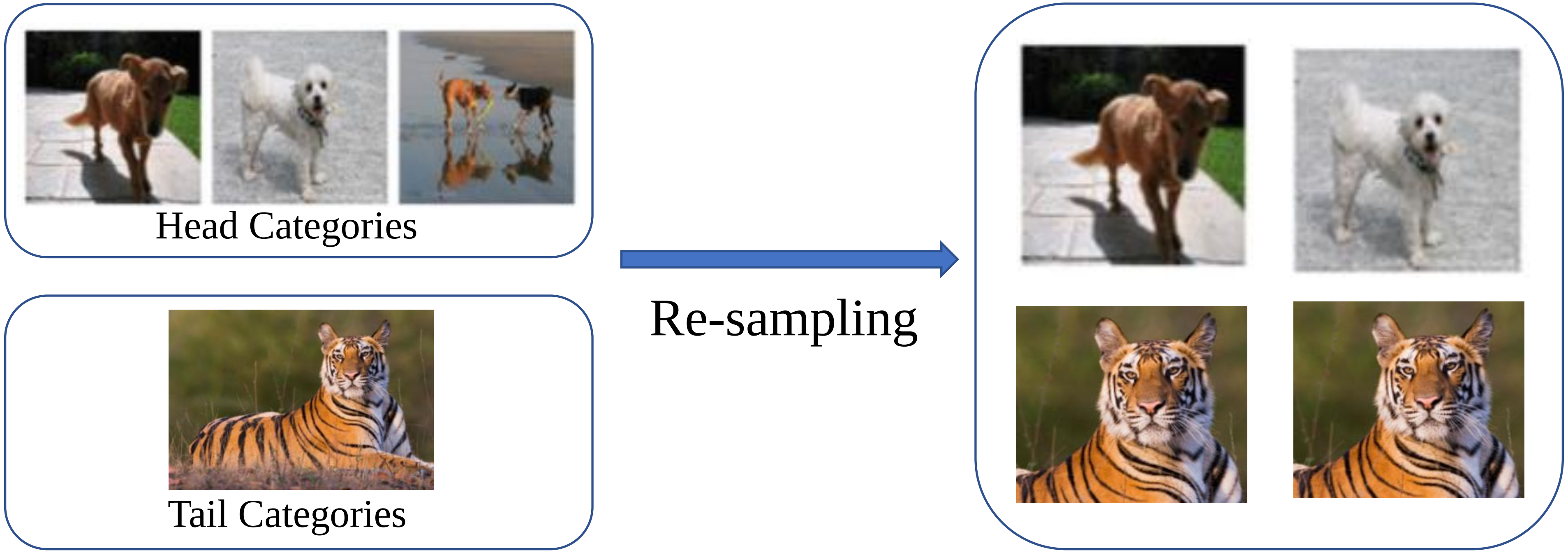
Two-stage Rebalanced learning

- BBN
- Decoupling representation & classifier(SOTA now)

Related work – Re-sampling & Re-weighting

Defects

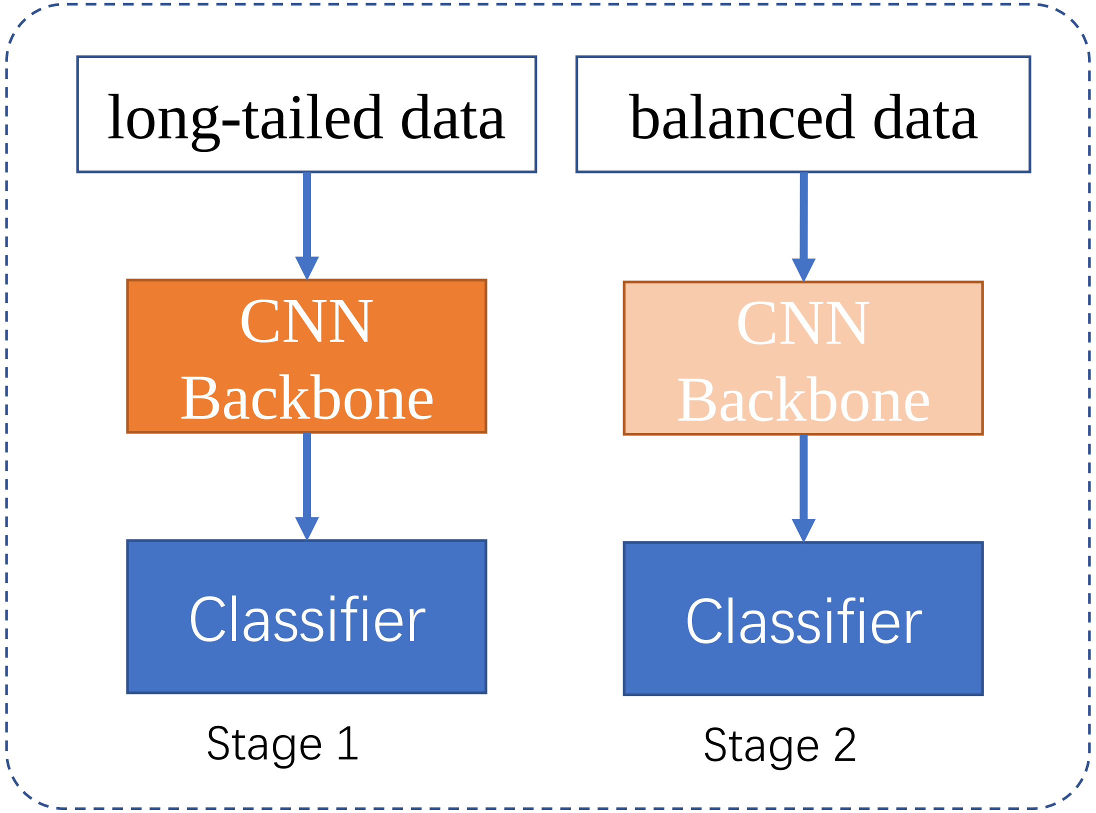
- The data distribution in training is not real, thus the learned backbone is bad.
- Inevitably cause the under-fitting/over-fitting problem to head/tail classes.
- Relying on the accessibility of data distribution also limits their application scope, e.g., not applicable in online and streaming data.



Related work – Two-stage Rebalancing

Defects

- They fail to explain the whys and wherefores of their solutions.
- This kind of approaches are less effective or efficient.



Introduction - Experiments on ImageNet-LT

Methods	Many-shot	Medium-shot	Few-shot	Overall
Focal Loss† [24]	64.3	37.1	8.2	43.7
OLTR† [8]	51.0	40.8	20.8	41.9
Decouple-OLTR† [8, 10]	59.9	45.8	27.6	48.7
Decouple-Joint [10]	65.9	37.5	7.7	44.4
Decouple-NCM [10]	56.6	45.3	28.1	47.3
Decouple-cRT [10]	61.8	46.2	27.4	49.6
Decouple-τ-norm [10]	59.1	46.9	30.7	49.4
Decouple-LWS [10]	60.2	47.2	30.3	49.9
Baseline	66.1	38.4	8.9	45.0
Cosine† [38, 39]	67.3	41.3	14.0	47.6
Capsule† [8, 42]	67.1	40.0	11.2	46.5
(Ours) De-confound	**67.9**	42.7	14.7	48.6
(Ours) Cosine-TDE	61.8	47.1	30.4	50.5
(Ours) Capsule-TDE	62.3	46.9	30.6	50.6
(Ours) De-confound-TDE	62.7	**48.8**	**31.6**	**51.8**

Introduction - Motivation

- We, human beings, also live in a long-tailed world.
- The problem must reside in the learning framework of computer.

- Find that the **SGD momentum** is essentially a confounder in long-tailed classification.
- Then, establish a **causal inference framework**, which unravels the whys of previous methods.

Introduction - Motivation

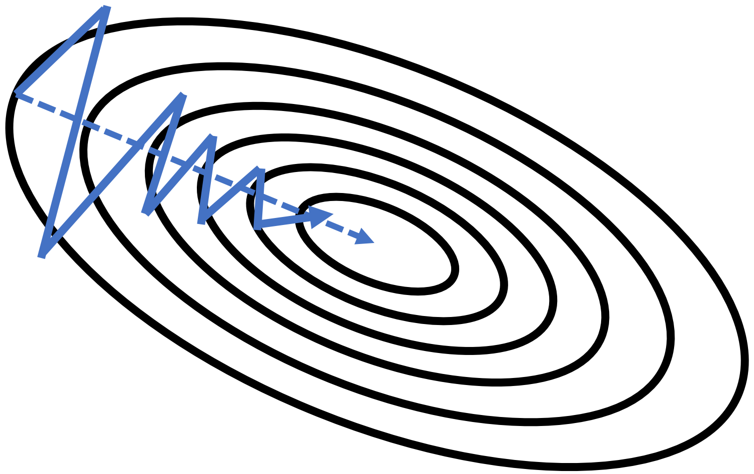
The PyTorch implementation of SGD with momentum

$$v_t = \underbrace{\mu \cdot v_{t-1}}_{momentum} + g_t, \qquad \theta_t = \theta_{t-1} - lr \cdot v_t$$

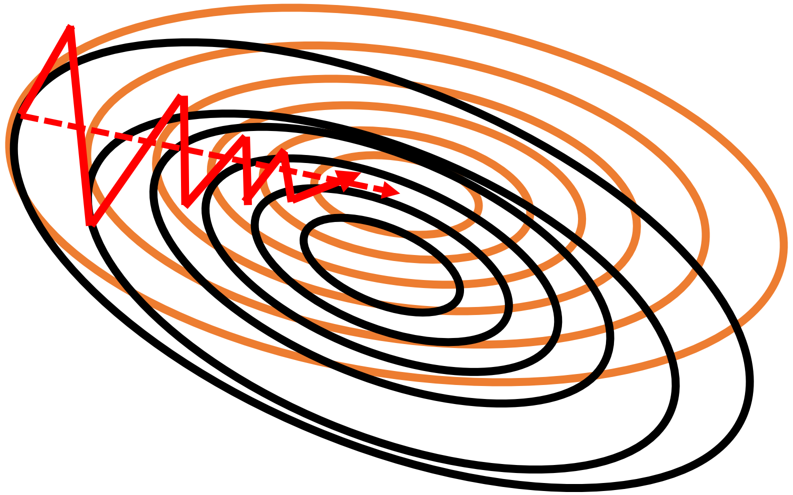
- The momentum is a moving average of the gradient over all past samples.
- Thus, it will encode the data distribution, that creates a shortcut towards the head.

Introduction – Motivation- Accumulative momentum effect

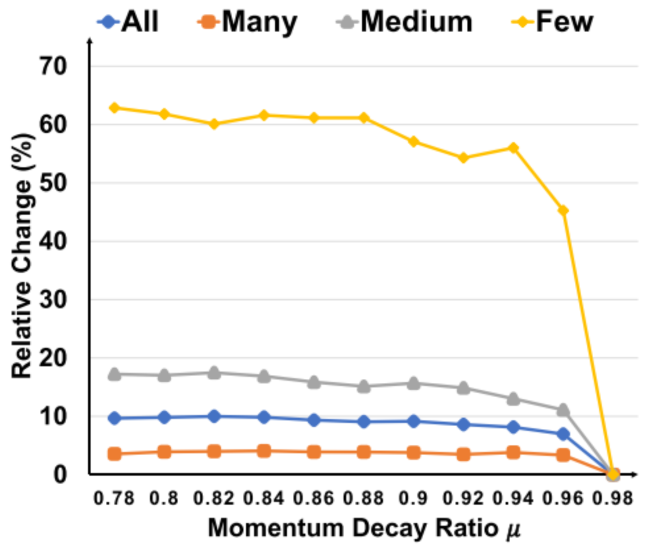
Accumulative Momentum Effect



SGD Momentum in
Balanced Dataset



SGD Momentum in
Long-Tailed Dataset



- Global Optima for All Categories
- Local Optima for Head Categories
- Momentum Direction in Balanced Data
- Momentum Direction in Long-Tailed Data

Introduction – Motivation- Accumulative momentum effect

Why not remove the momentum
when training the long-tailed dataset?

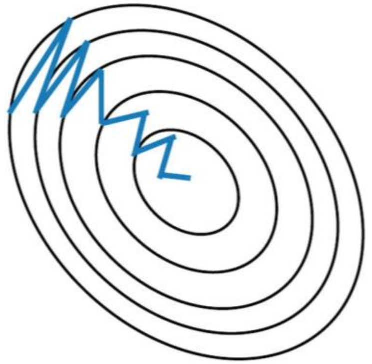
Remove Momentum:

- Unstable Gradient
- Local Optima
- SGD Still Accumulates

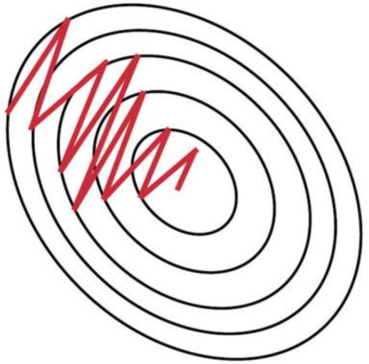
Keep Momentum in Training

+

Remove Bad Causal Effect

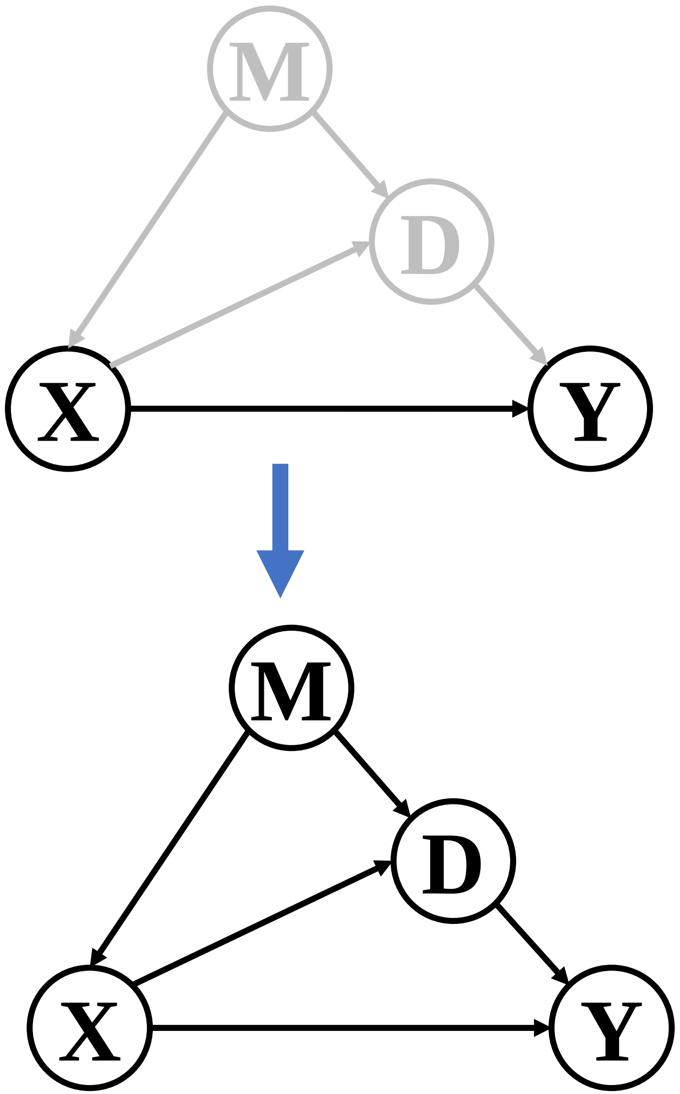


Stochastic Gradient Descent **with** Momentum



Stochastic Gradient Descent **withhout** Momentum

Introduction – Motivation- Causal Graph of momentum

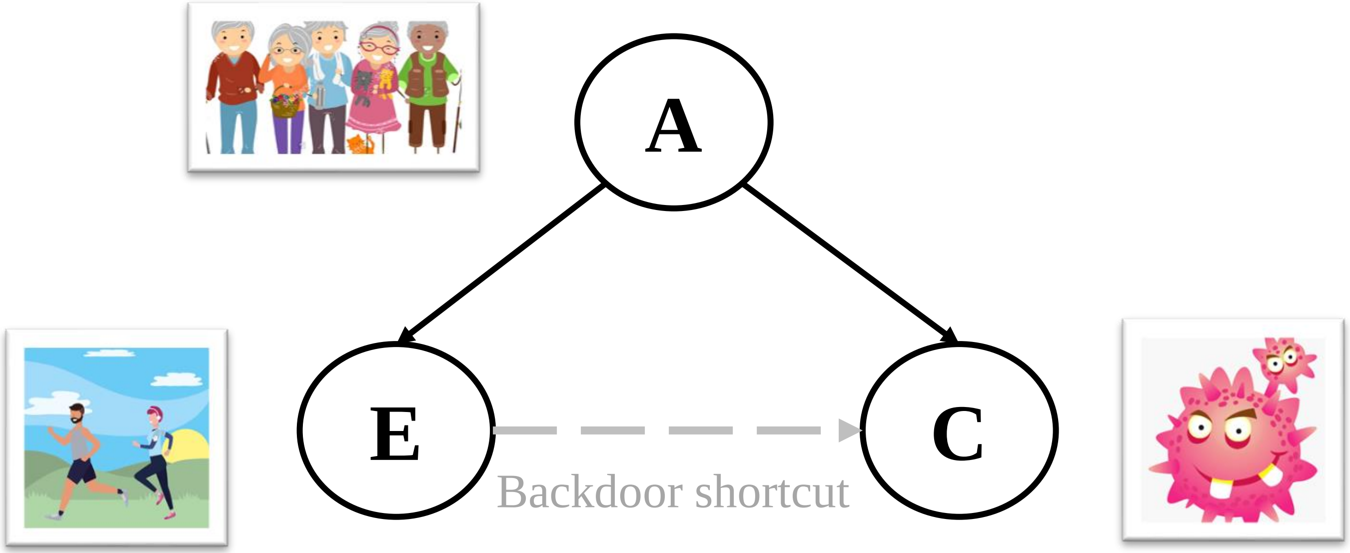


X : Feature
Y : Prediction
M: Momentum
D : Projection on Head

Two Undesired Causal Effects of Momentum:

- Backdoor shortcut
- Indirect Mediator Effect

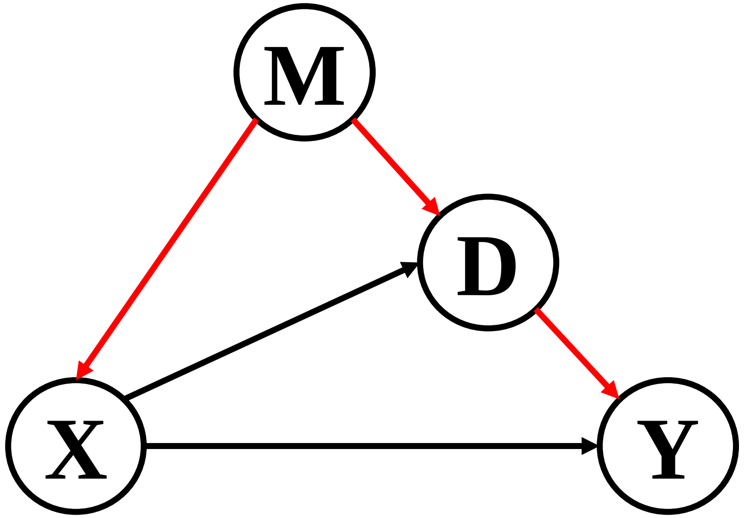
Introduction – Backdoor shortcut



Backdoor shortcut:

- $A \uparrow \;\Rightarrow\; E \uparrow$
- $A \uparrow \;\Rightarrow\; C \uparrow$
- $E \uparrow \;\Rightarrow ?\; C \uparrow$

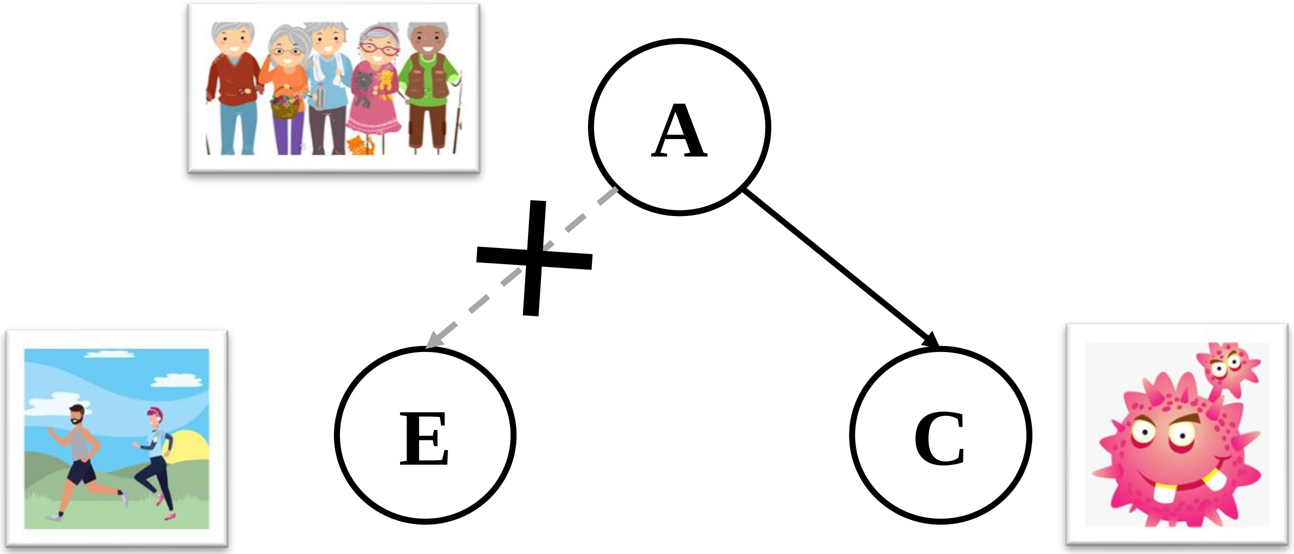
A: age E: exercise C: cancer

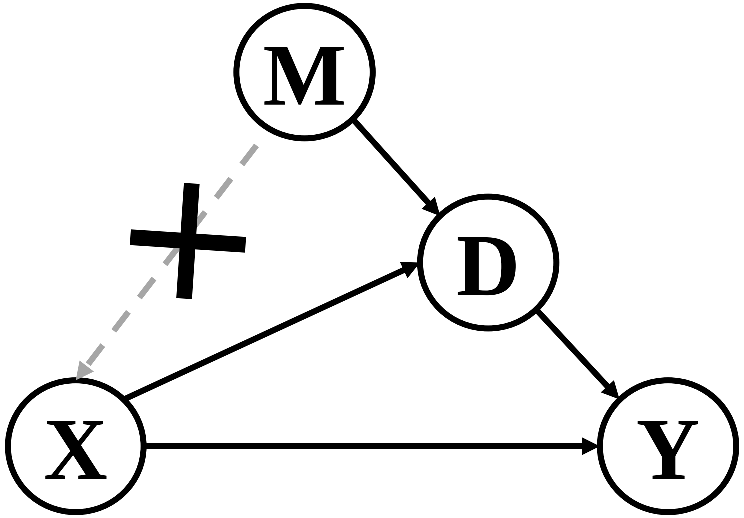


How to avoid?

- Backdoor adjustment

Introduction – Backdoor adjustment



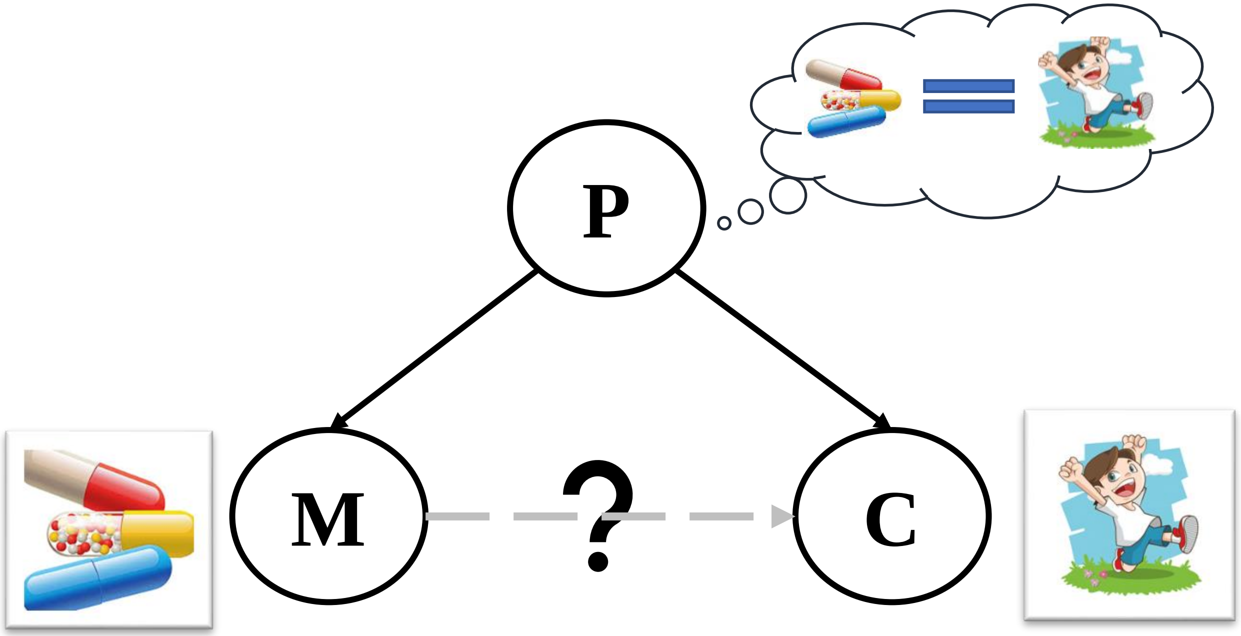


Backdoor adjustment:

$$P(C|do(E)) = \sum_{a} P(C|E, A = a)P(A = a)$$

do(E) : intervention on E

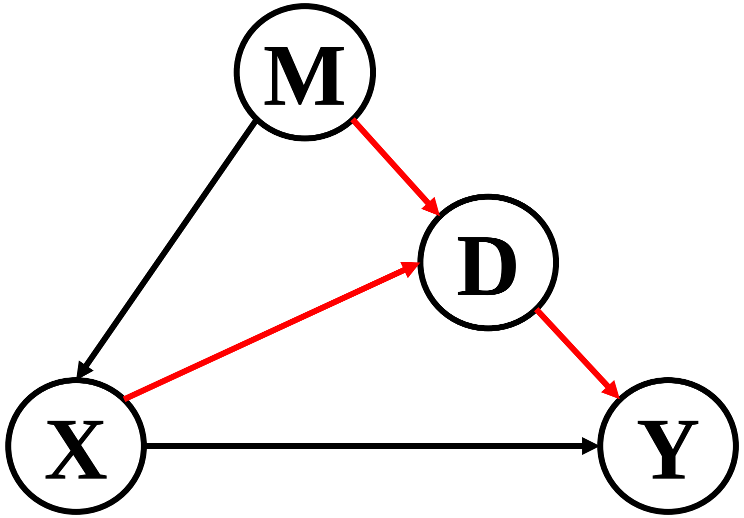
Introduction – Indirect Mediator Effect



Indirect Mediator Effect :

- $M \quad \Rightarrow \quad P$
- $P \quad \Rightarrow \quad C$
- $M \quad \Rightarrow ? \quad C$

M: medicine **P: placebo** **C: cure**



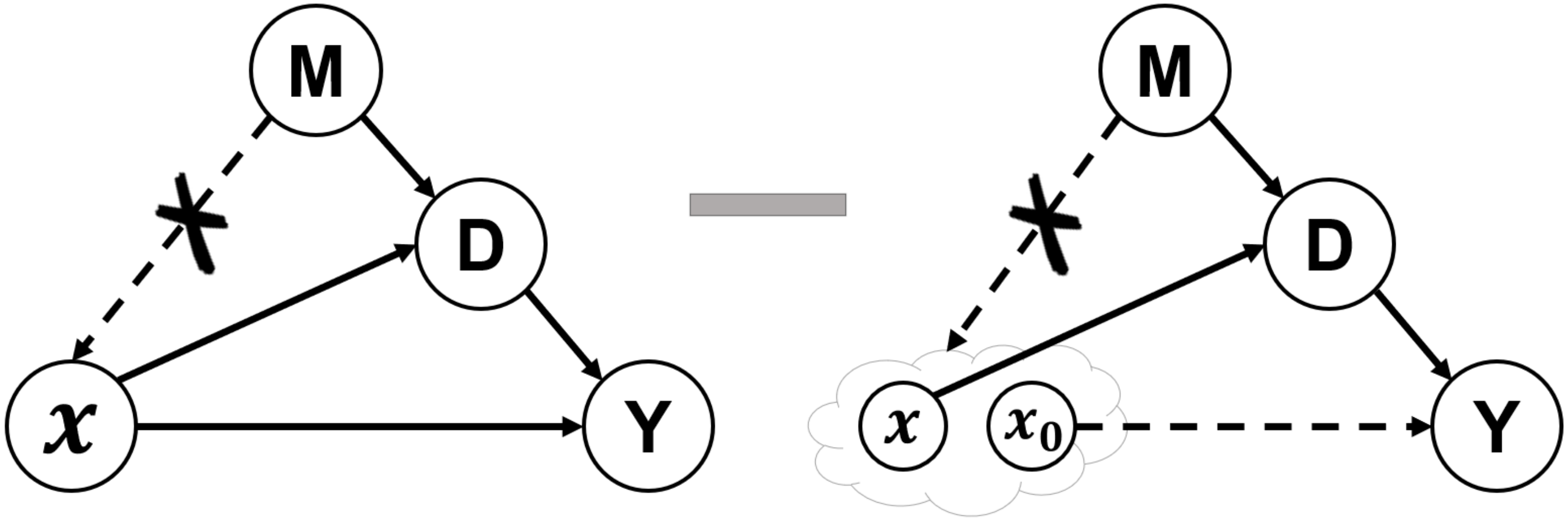
How to avoid?

- Setting control group:
- $C(M = m_0, P = p)$

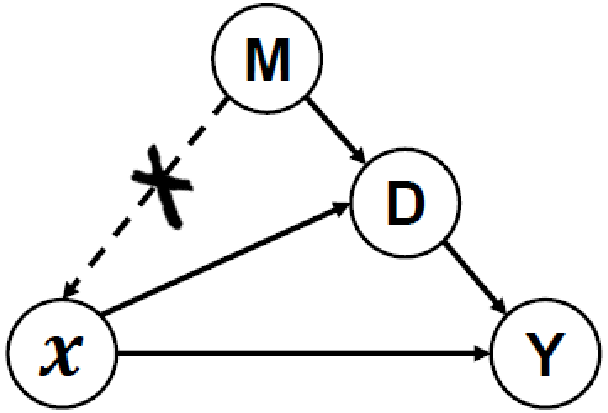
Method - De-confound TDE Classifier

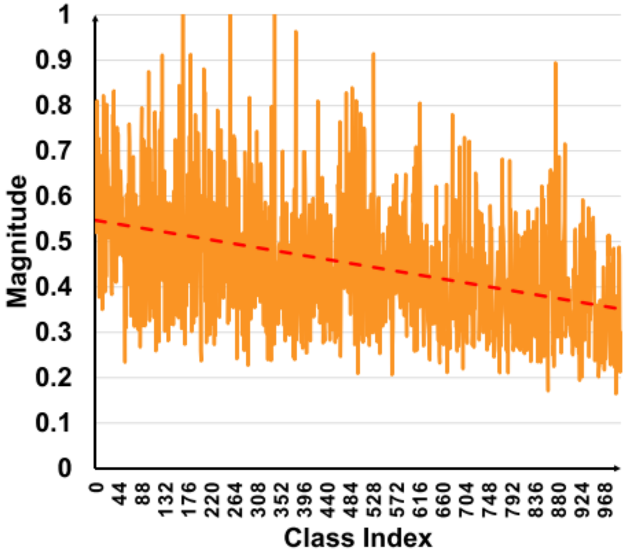
$$argmax_{i \in C}\, TDE(Y_i) = [Y_d = i | do(X = x)] - [Y_d = i | do(X = x_0)]$$

The proposed classifier = De-confounded Training + TDE Inference in test



Method - De-confound Training





Mean magnitude of x for each class i

- The backdoor adjustment:

$$P(Y=i|do(X=x))$$

$$=\sum_m P(Y=i|X=x,M=m)P(M=m)$$

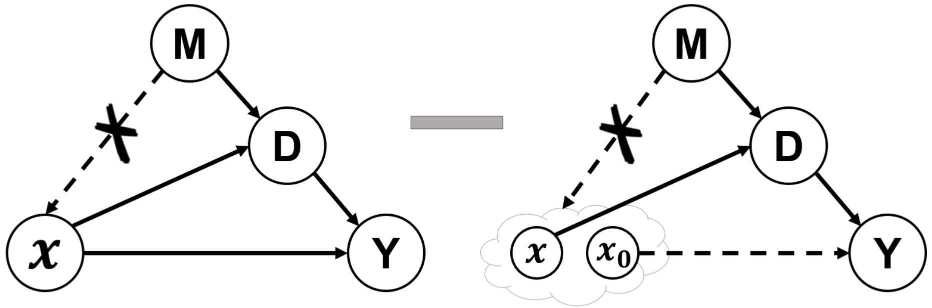
Approximation

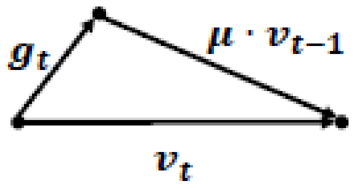
$$\approx \frac{1}{K}\sum_{k=1}^{K}\tilde{P}(Y=i,X=x^k,D=d^k)$$

$$\tilde{P} \propto \tau\frac{f(x^k,d^k;w_i^k)}{g(x^k,d^k;w_i^k)}$$

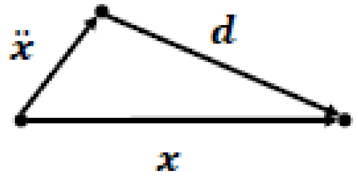
$$=\frac{\tau}{K}\sum_{k=1}^{K}\frac{\left(w_i^k\right)^T\cdot x^k}{\|x^k\|\cdot\|w_i^k\|+\boxed{\gamma\|x^k\|}}$$

Method - TDE





(a) Decompose the gradient velocity



(b) Decompose the biased feature vector

- **D : Head projection d for each x**

(Caused by the biased parameters of backbone)

$$d = \|d\| \cdot \hat{d} = \cos(x, \hat{d}) \cdot \|x\| \cdot \hat{d}$$

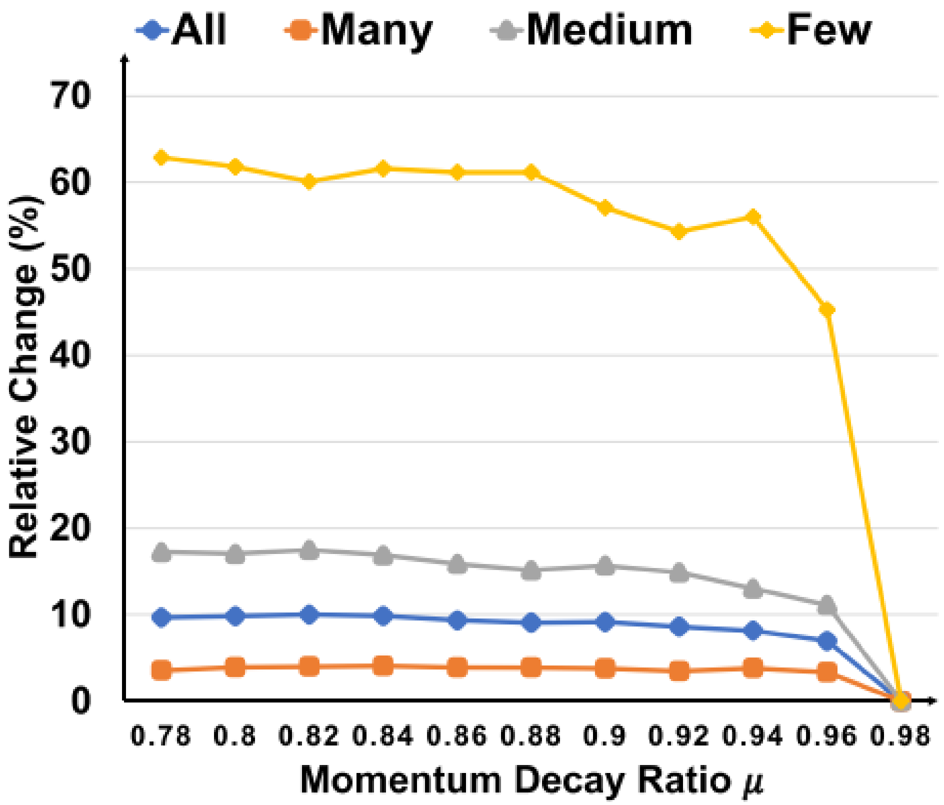
- **Assumption 1:**

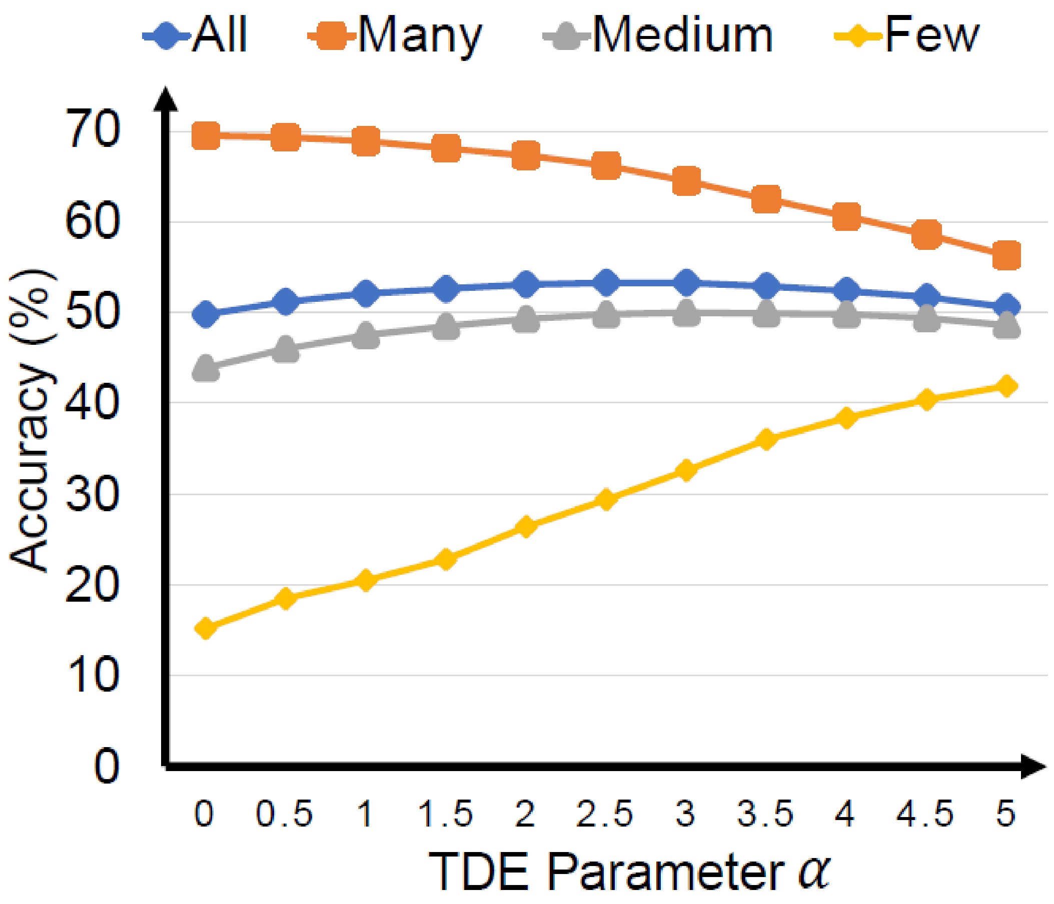
The head direction $\hat{d}$ is the unit vector of the exponential moving average of features the same as momentum (T is the number of the total training iterations).

$$\hat{\boldsymbol{d}} = \frac{\overline{\boldsymbol{x}}_{\boldsymbol{T}}}{\|\overline{\boldsymbol{x}}_{\boldsymbol{T}}\|},\ where\ \overline{\boldsymbol{x}}_{\boldsymbol{t}} = \boldsymbol{\mu} \cdot \overline{\boldsymbol{x}}_{\boldsymbol{t-1}} + \boldsymbol{x}_{\boldsymbol{t}}$$

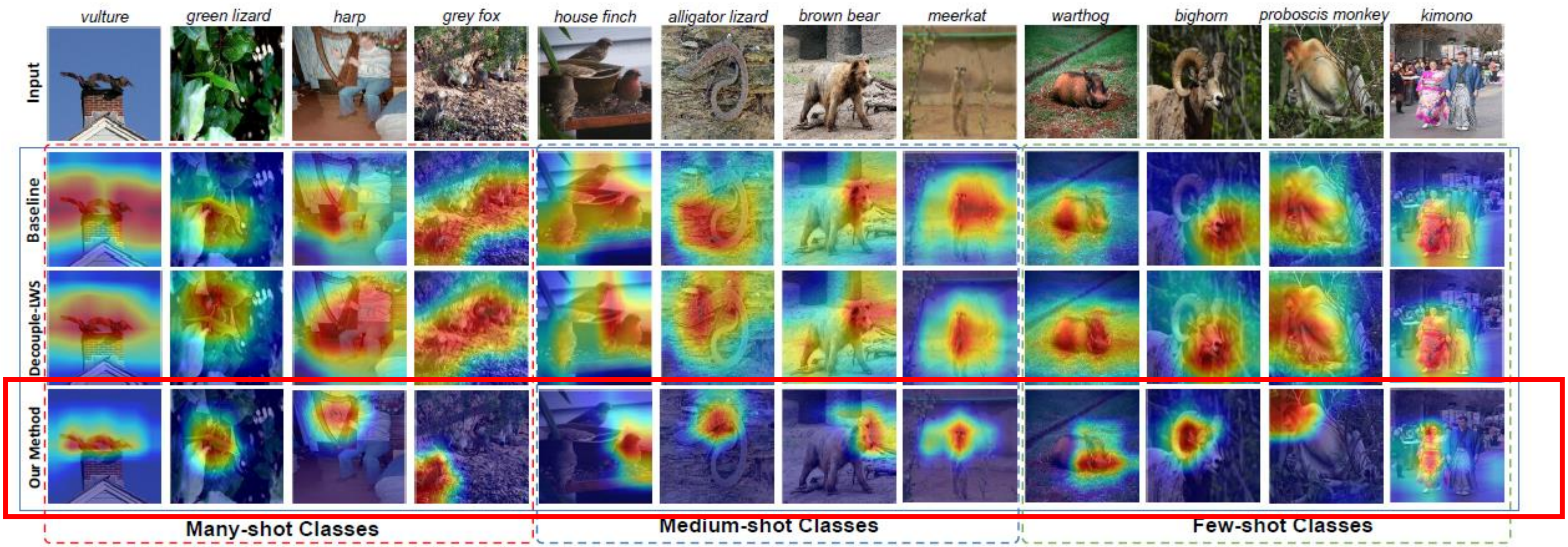
$$TDE(Y_i) = \frac{\tau}{K} \sum_{k=1}^{K} \left(\frac{(w_i^k)^T x^k}{(\|w_i^k\| + \gamma)\|x^k\|} - \alpha \cdot \frac{\cos(x^k, \hat{d}^k) \cdot (w_i^k)^T \hat{d}^k}{(\|w_i^k\| + \gamma)} \right)$$

Experiments – Impact of momentum and TDE





Experiments – Grad-cam Visualization on ImageNet-LT



Smaller areas of focus

Conclusion & My opinion & questions about the paper

Conclusion:

The proposed de-confound TDE **simple**, **adaptive**, and **agnostic** to the prior statistics of the class distribution:

- It doesn't introduce any additional stages or modules.
- It can be applied to a variety of tasks, including but not limited to image classification, object detection, instance segmentation.
- It doesn't rely on the accessibility of data distribution.

Opinion:

It's really good that the paper firstly proposed a theory of the long-tailed problem based on cause and effect analysis. However, its theory is too obscure. Based on the results of the theory and experiments of this paper, I think the paper essentially changes the classifier through two facets. On the one hand, it adopts normalization which alleviates the bias about the classifier's modulus, besides, the idea of multi-head is also fantastic. On the other hand, it alleviates the bias about the classifier's directions.

My questions about the paper

Question:

- What if we decompose the confounder d in another way i.e. not in orthogonal way?
- Is there anything which is also the confounder in the learning process like batchnorm?
- Besides, it's confusing that my code results in CIFAR100-LT with imbalanced ratio 100 is worse than the results showed in the paper about 1~2%.

Code results in CIFAR100-LT with imbalanced ratio 100

```
 Phase: val

 Evaluation_accuracy_micro_top1: 0.427
 Averaged F-measure: 0.395
 Many_shot_accuracy_top1: 0.629 Median_shot_accuracy_top1: 0.422 Low_shot_accuracy_top1: 0.196

===> Saving checkpoint
./logs/CIFAR100_LT/models/resnet32_e200_warmup_causal_norm_ratio100
=====> Current Learning Rate of model classifier : 2e-05
=====> Current Learning Rate of model feat_model : 2e-05
Epoch: [200/200] Step:     0  Minibatch_loss_performance: 0.091 Minibatch_accuracy_micro: 0.982
Epoch: [200/200] Step:    10  Minibatch_loss_performance: 0.102 Minibatch_accuracy_micro: 0.982
Epoch: [200/200] Step:    20  Minibatch_loss_performance: 0.102 Minibatch_accuracy_micro: 0.988

 Training acc Top1: 0.986
 Many_top1: 0.988 Median_top1: 0.977 Low_top1: 0.923

Phase: val
100%|


 Phase: val

 Evaluation_accuracy_micro_top1: 0.427
 Averaged F-measure: 0.395
 Many_shot_accuracy_top1: 0.628 Median_shot_accuracy_top1: 0.422 Low_shot_accuracy_top1: 0.197

===> Saving checkpoint

Training Complete.
Best validation accuracy is 0.429 at epoch 195
Phase: test
100%|


 Phase: test

 Evaluation_accuracy_micro_top1: 0.428
 Averaged F-measure: 0.397
 Many_shot_accuracy_top1: 0.629 Median_shot_accuracy_top1: 0.424 Low_shot_accuracy_top1: 0.199

62.9      42.4     19.9     42.8
Done
=========================  ALL COMPLETED  =========================
```

Thanks for listening.