Every Coin Matters: Reweighting Everything for Large-scale Noisy Image Classification

Shumin Han

Outline

- Task & Challenge
- Our Solution
- Conclusion

Task Description

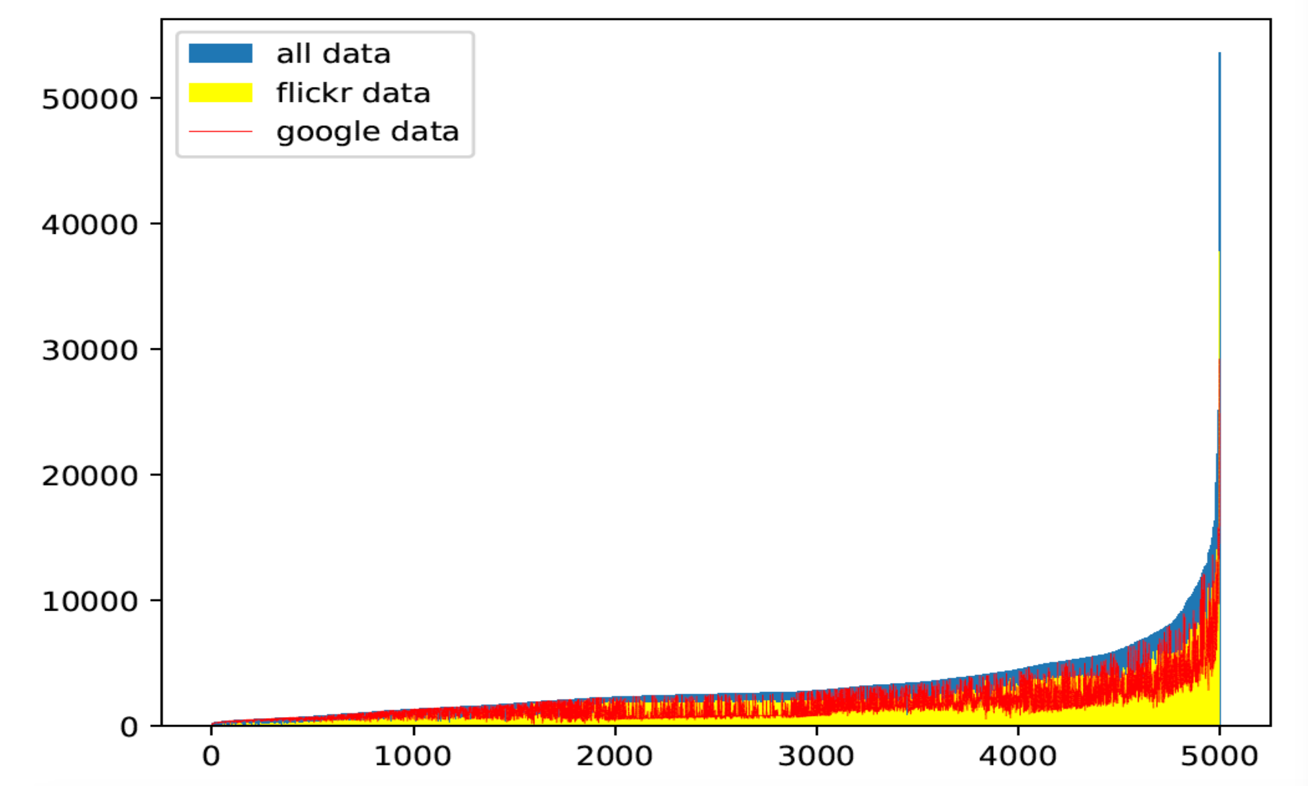
- Describe the task

5,000 classes

Name		number of images
Train dataset	Google	8,366,429
	Flickr	7,710,236
Validation dataset		294,099
Test dataset		294,099

Challenge Analysis

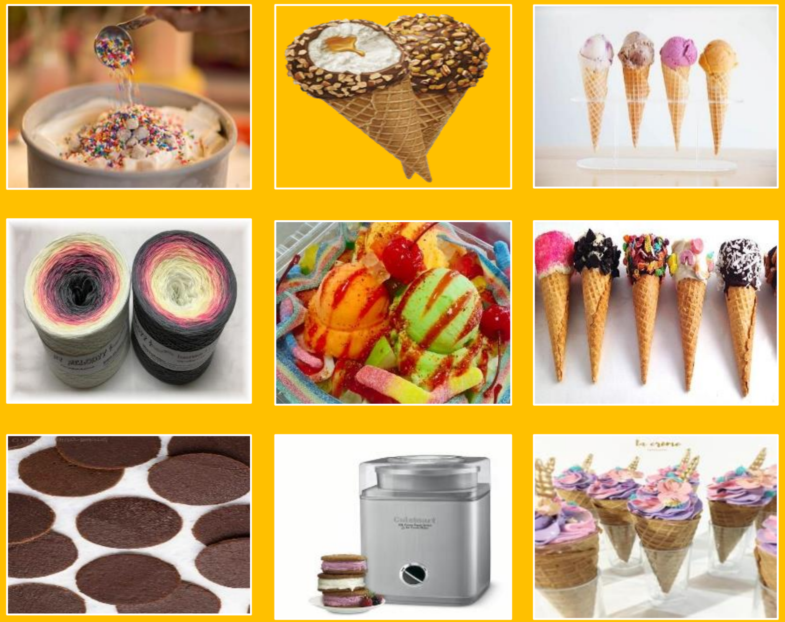
- Class imbalance: unbalance data in different classes.
  - Motivates: reweighting Class



Challenge Analysis

- Cluster Diversity: High inter-class similarity, low intra-class similarity.
  - Motivates: reweighting Cluster

Class: 973
Query words: icecream, ice+cream

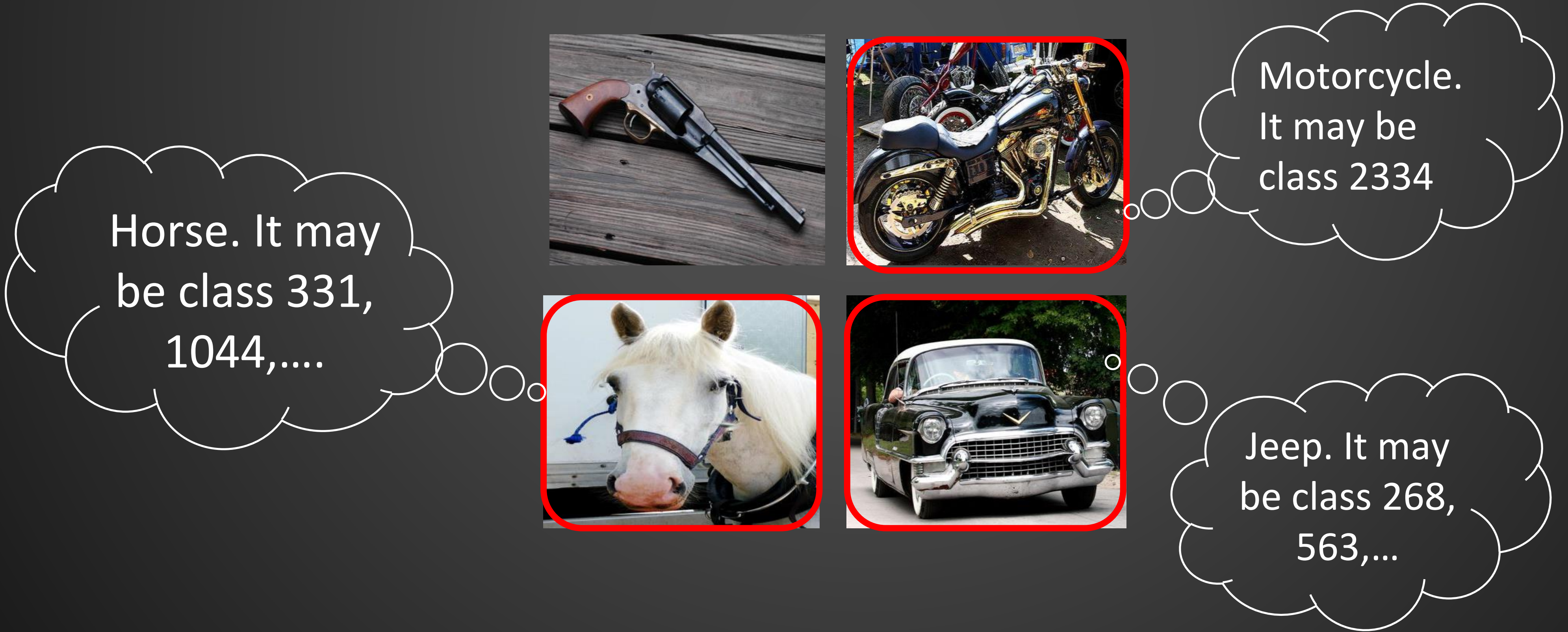


Class: 3440
Query words: ice-cream cone

Challenge Analysis

- Lable Ambiguity: different classes, similar labels
  - Motivates: reweighting/smoothing label

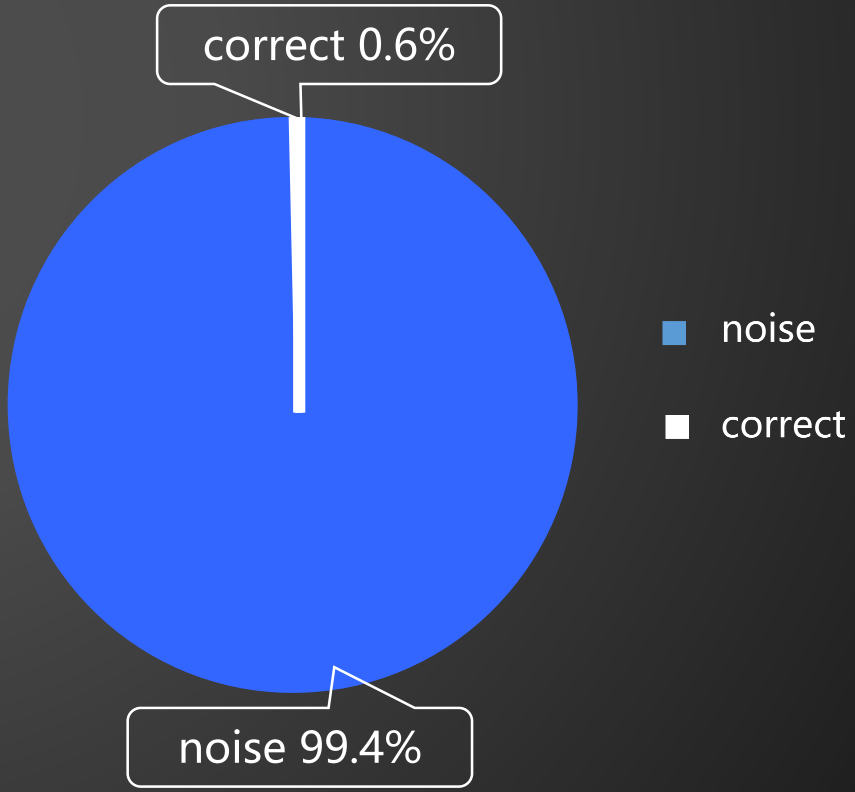
Class:218 Label : revolver pistol



Challenge Analysis

- Noisy Instance : Positive instances are overwhelmed by massive noisy instances.
  - Motivates: reweighting Instance selection probability

Class : 1048 Label : ben



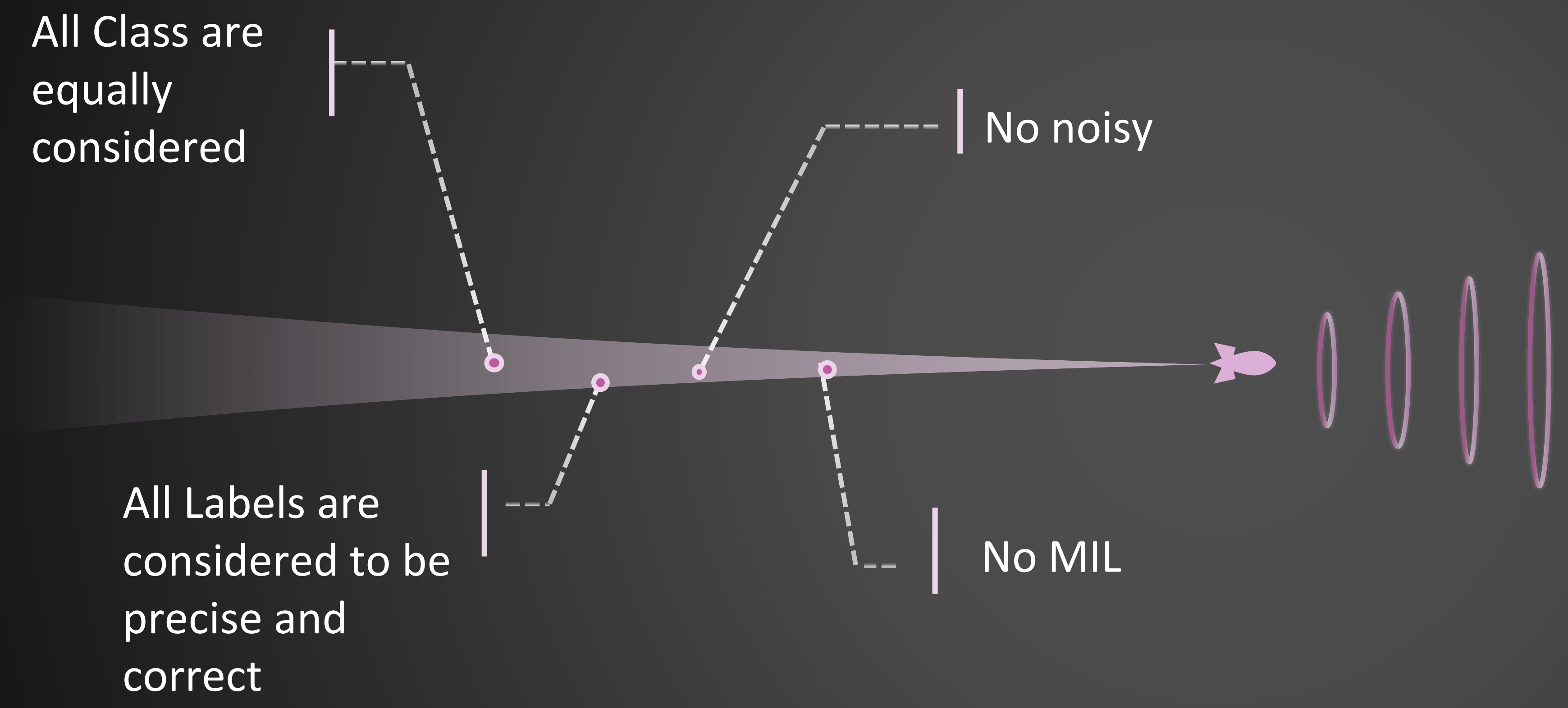
Challenge Analysis

- Instance Saliency: Not all positive instance contribute equally
  - Motivates: reweighting positive instances inside a bag (attention).

Class : 0   Label : kit fox, vulpes macrotis

Do they make the **same** contribution**?**

What if…



- Baseline performance

model	Top1 accuracy	Top5 accuracy
Resnext101	47.2%	71.7%

Outline

- Task & Challenge
- Our Solution
- Conclusion

Motivation

unbalance data in different classes: reweighting class

High inter-class similarity, low intra-class similarity: reweight cluster

Label Ambiguity: reweighting label

Noisy Instance: reweighting Instance selection probability

Not all positive instance contribute equally: reweighting positive instances inside a bag

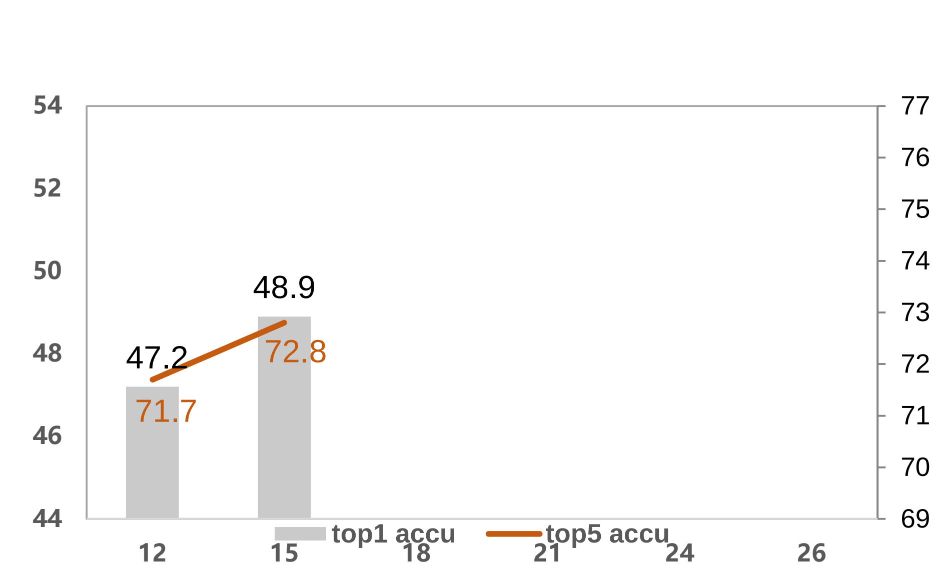
Reweighting Class

- Assigning loss weight of each class

$$ratio[i] = \frac{const}{count[train[i]]}$$

$$weight[i] = (1-\partial) + \partial * \frac{ration[i]}{\sum_{j=1}^{total\_class\_num} ration[j]}, i \in [1, total\_class\_num]$$

- Performance



Reweighting Cluster

- Unsupervised learning: assigning weight by cluster density

- Step1: calculate top5 confusion classes

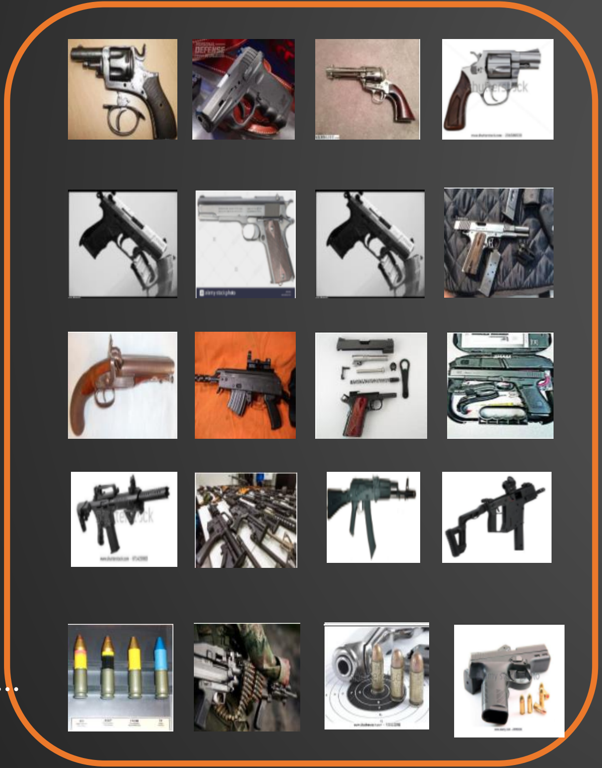
218：revolver pistol,…

1410：automatic pistol,…

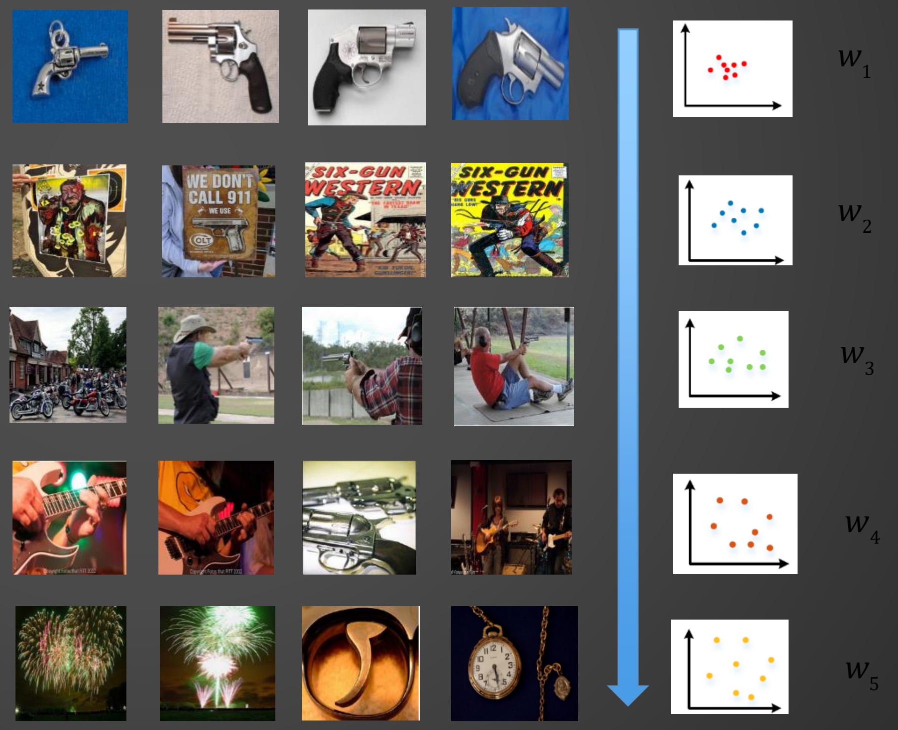
4648：gat, …

539：assault+rifle,…

3528：ammunition arms,…



- Step2: merge all images together and cluster by k-means.



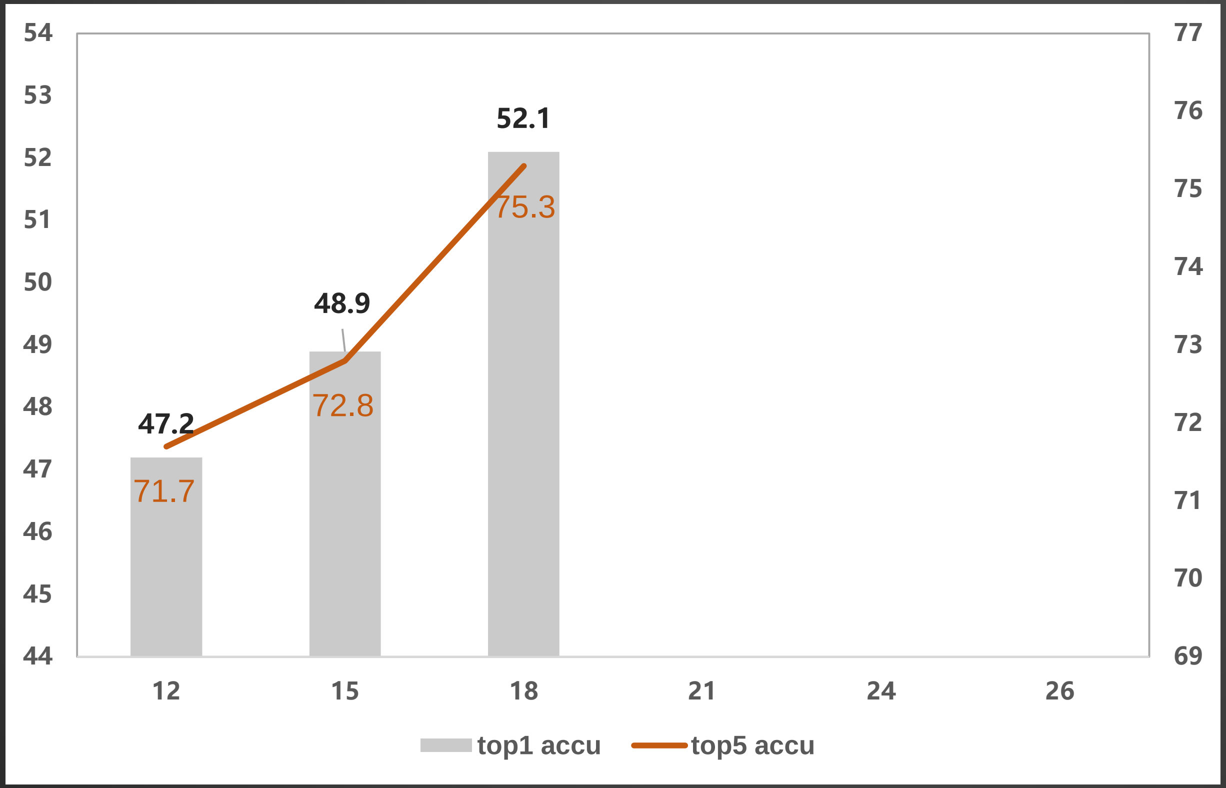
- Step3: assign each cluster a sampling weight according to density

w1>w2>w3>w4>w5

Reweighting Cluster

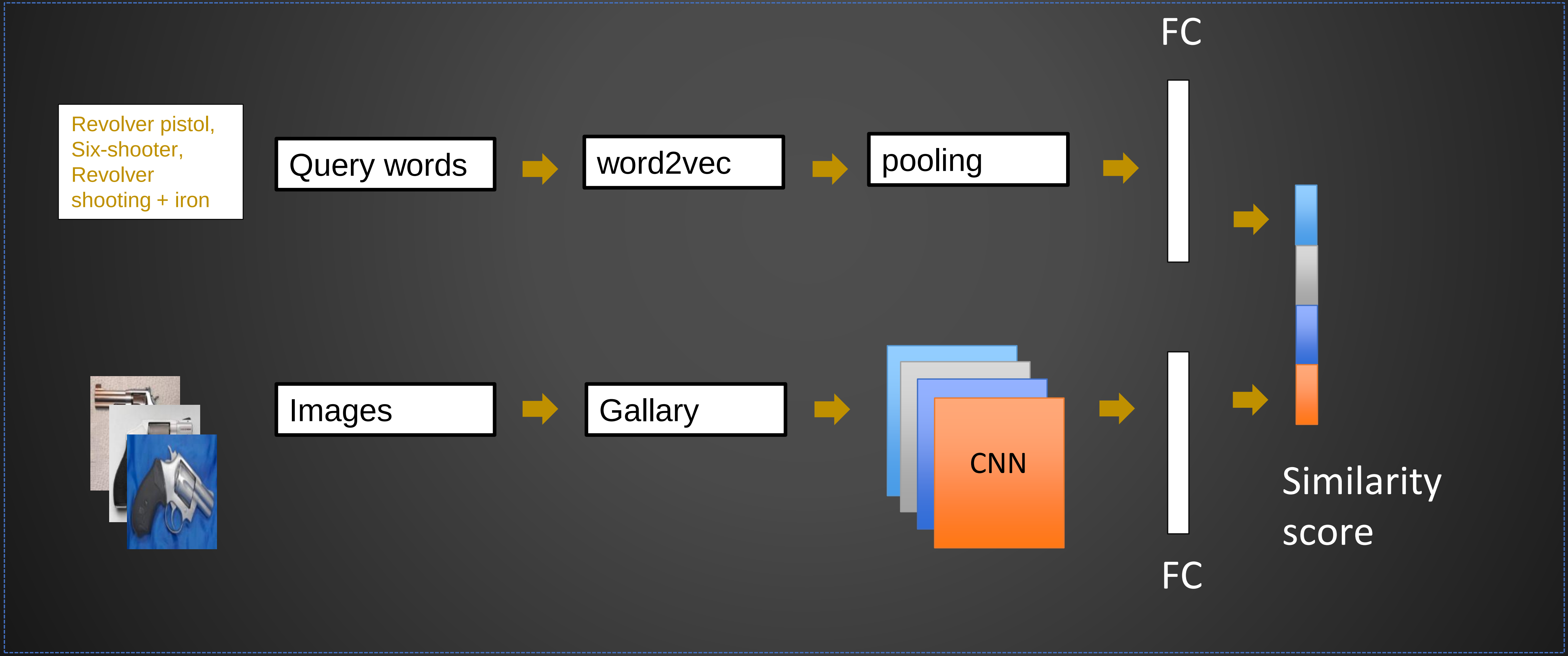
- Unsupervised learning: assigning weight by cluster density

- Performance



Reweighting Instance

- Text-image correlated model structure



Reweighting Instance

- How to train Text-image correlated model

Train a text-image correlated model
Training loss:

$$Loss = \max(\cos(u_p, v_p) - \cos(u_p, v_n) + m), 0)$$
$$u_p: imagefeature$$
$$v_p, v_n: textfeature$$

Step2

Step1

Pick Gallary image collection of high quality google images from each class, rank < 30
word2vec: fastText 300
CNN: Pretrained Resnext101

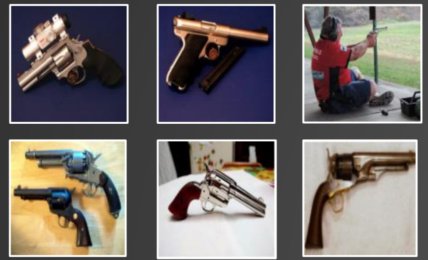
Step3

Score (reweight) the whole training dataset

Reweighting Instance

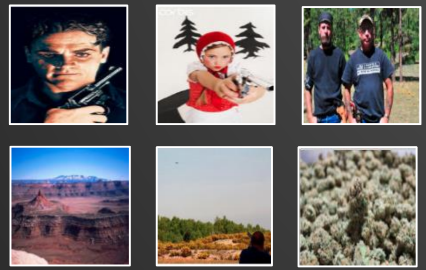
- Score results

Score in [0.9,1.0]

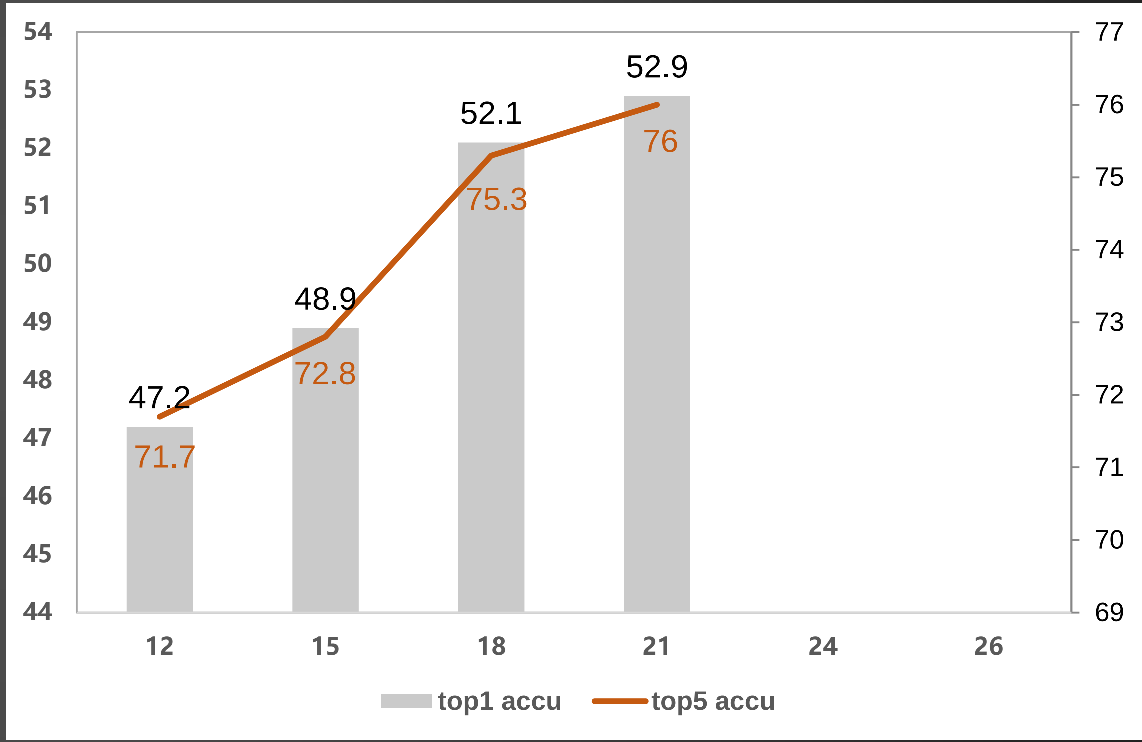


Score in [0.7,0.8]

Score in [0.0,0.6]

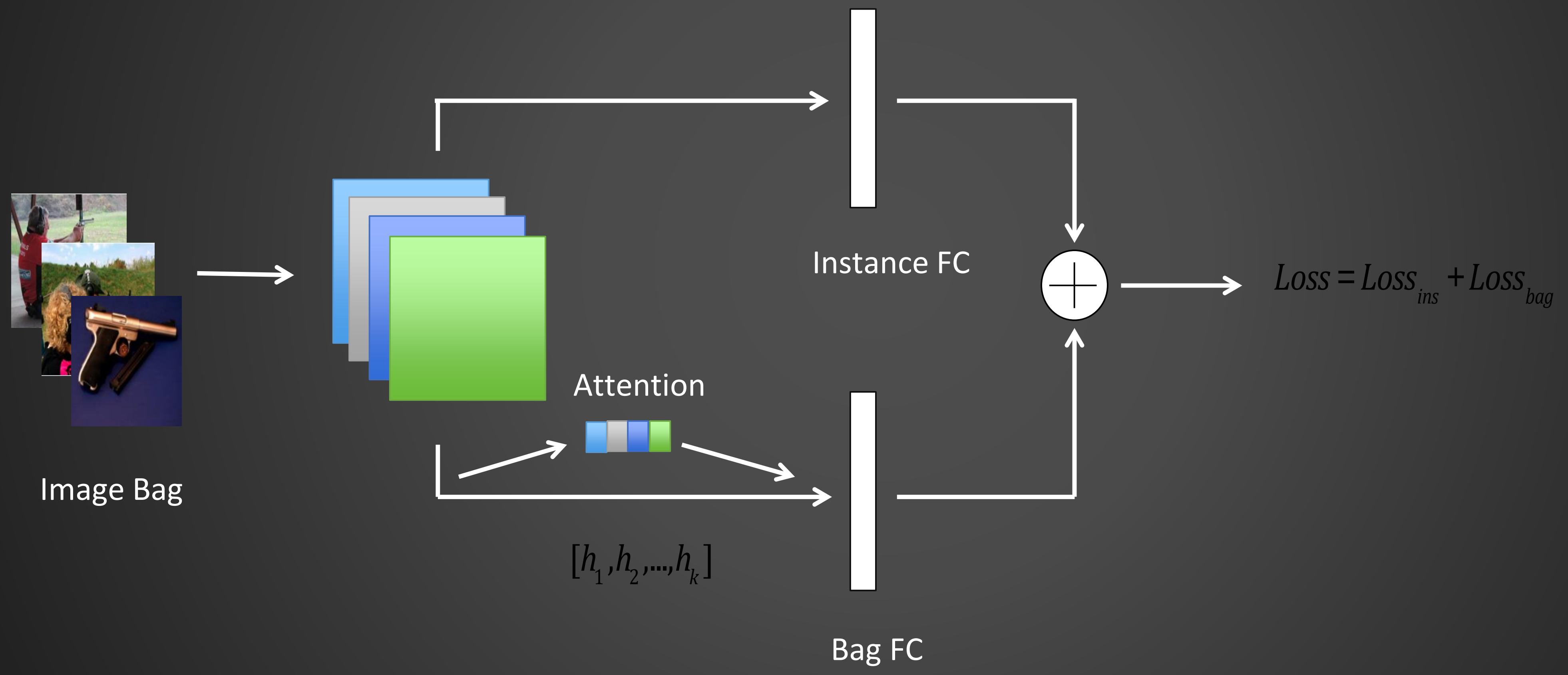


- Performance



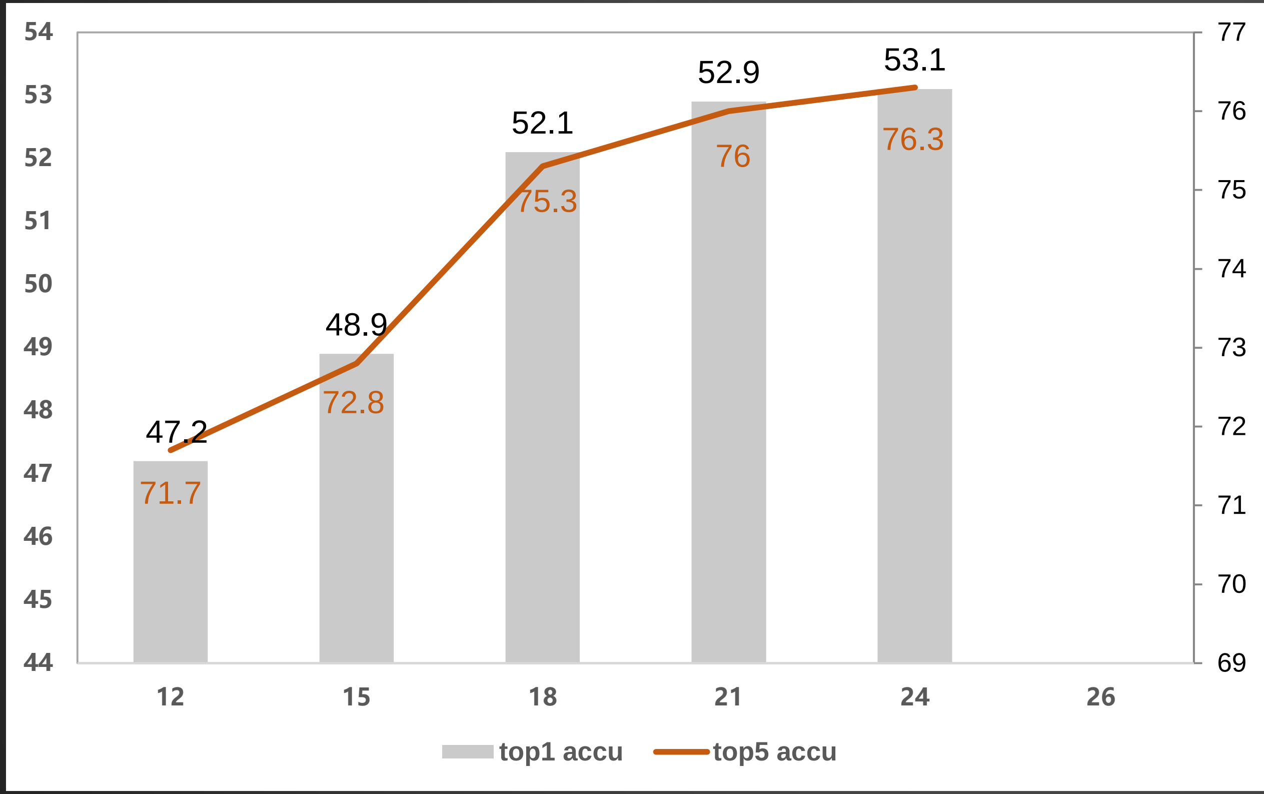
Reweighting Bag-Specific Instance Saliency

- Bag-instance learning structure



Reweighting Bag-Specific Instance Saliency

- Performance



Reweighting Label

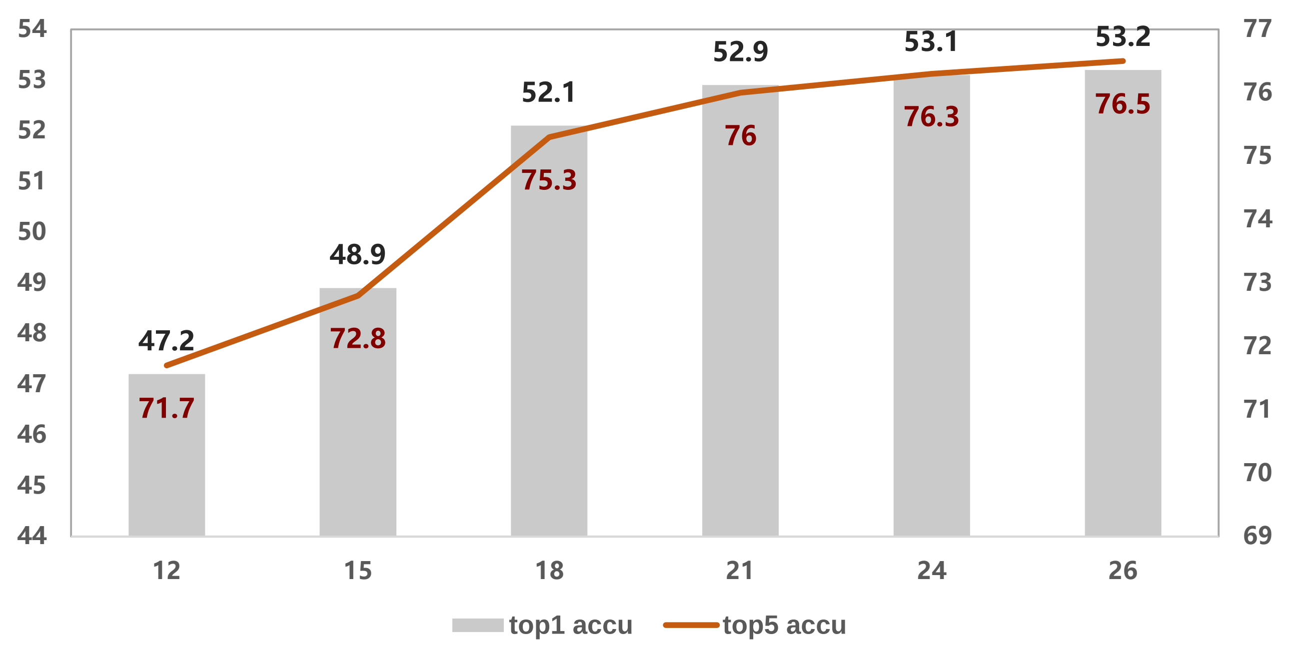
- bootstrapping

$$Loss(q,t) = \sum_{k-1}^{total\_cluass\_num} [Bt_k + (1-B)z_k]\log(q_k)$$

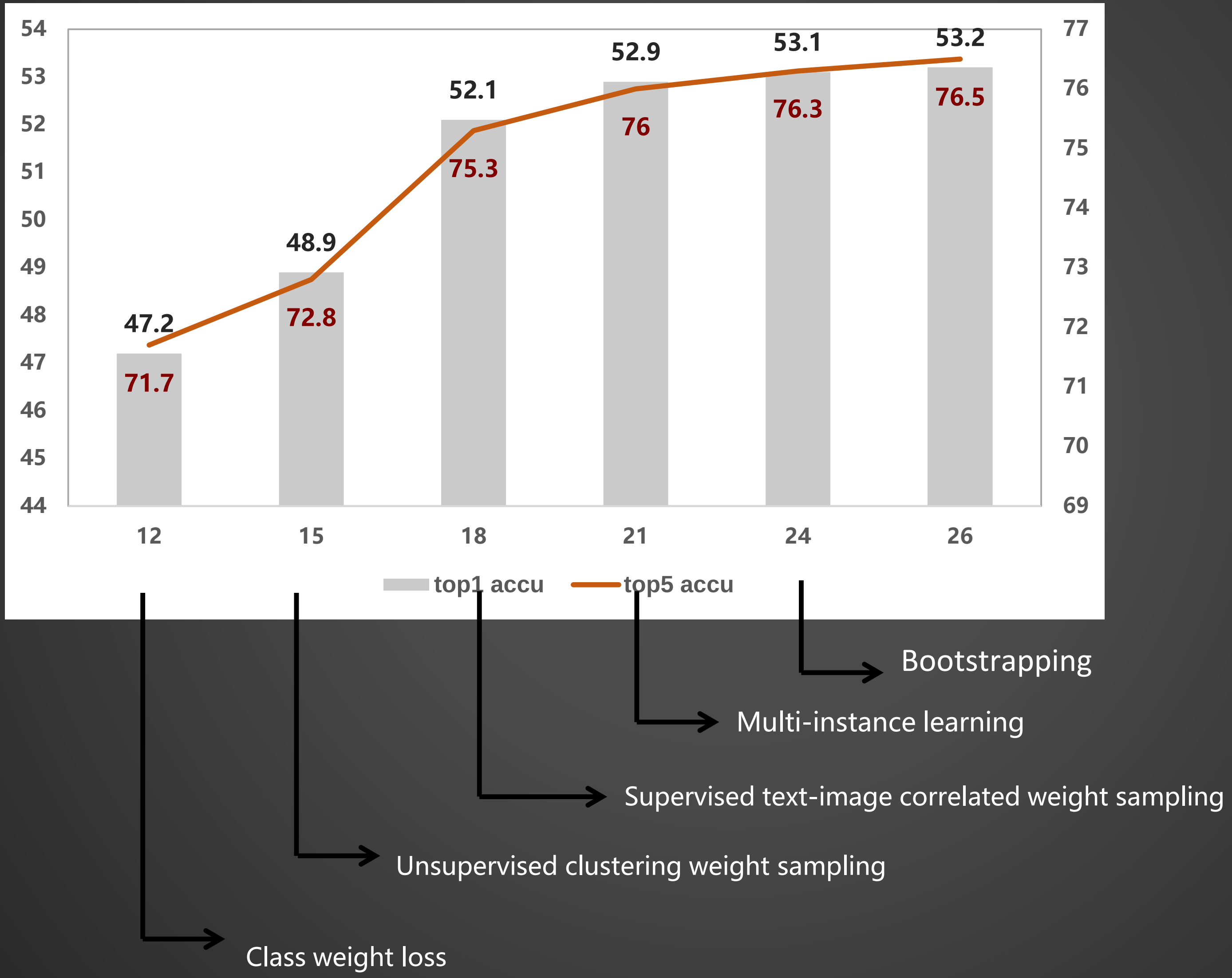
$$z_k = 1(k = \text{argmax}q_i, i = 1,\ldots\ldots, total\_class\_num)$$

Where z_k *is predicted label,* t_k *is ground-truth label, B=0.8.*

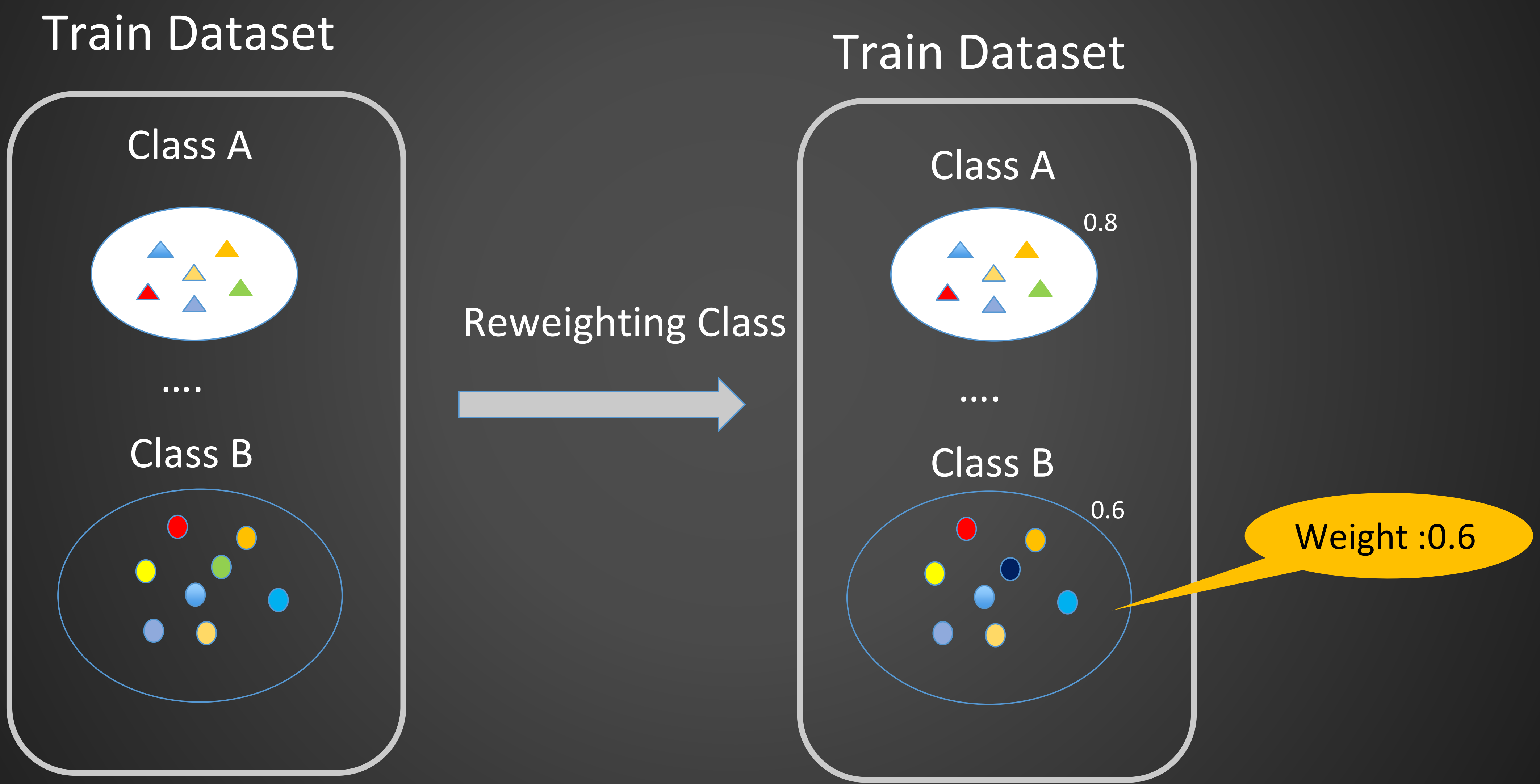
- Performance



Solution Summary

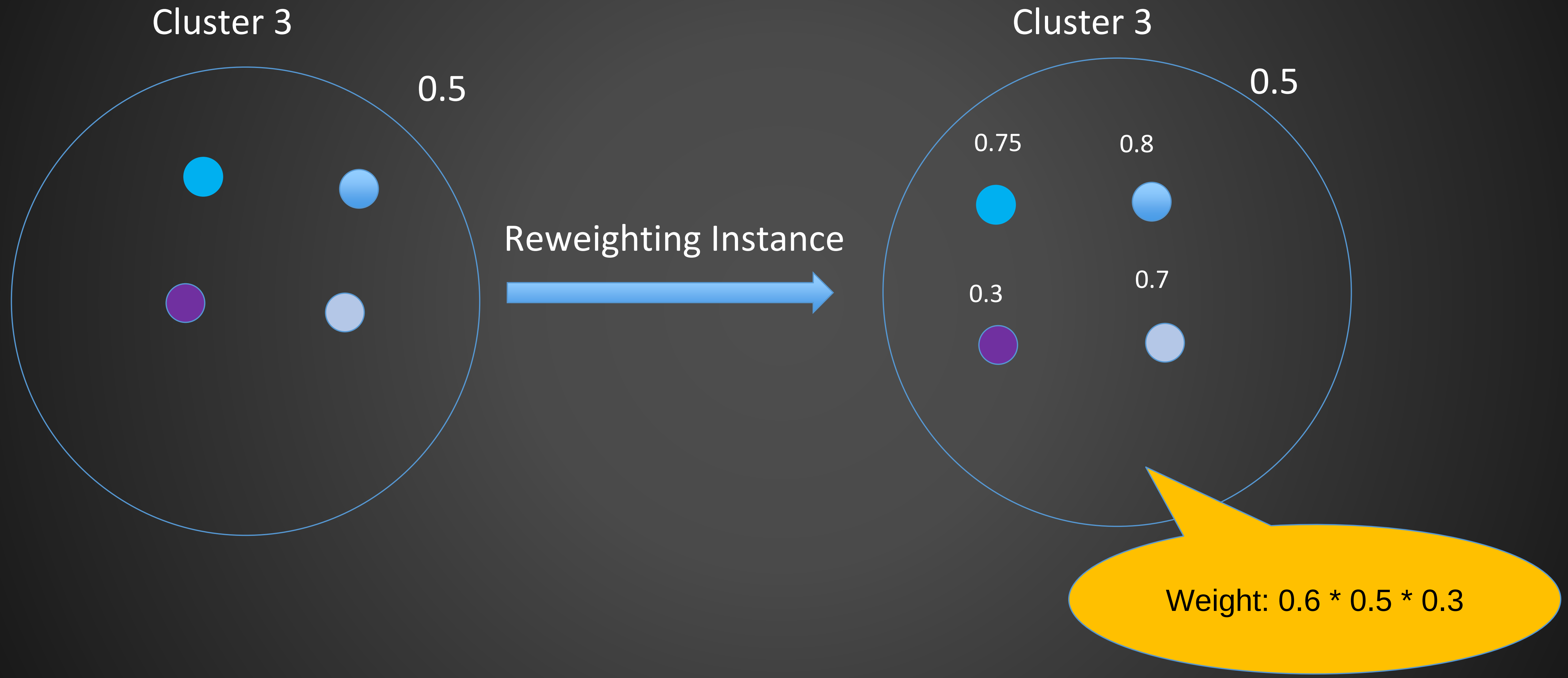


Solution Summary

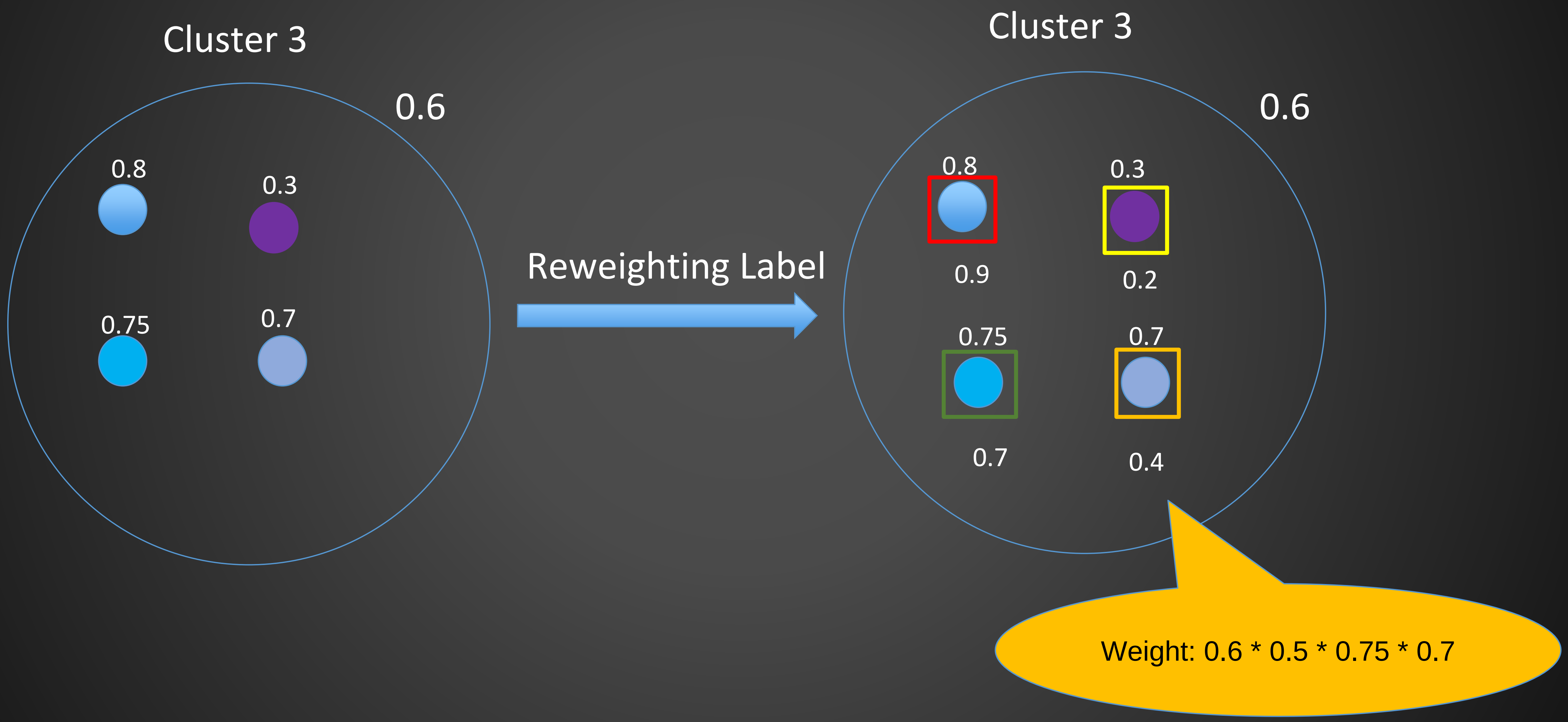


Solution Summary

Solution Summary



Solution Summary



Training and testing tricks

Tricks	Results
Remove noise out of top 15	+0.5% top5
Training different models (eg. Resnet, DPN, etc) for ensembling	+1.3% top5
Multi-crop testing	+1.0% top5
Multi-scale testing	+0.5% top5

Outline

- Task & Challenge
- Our Solution
- Conclusion

The result

Challenge Results

Rank	Team name	Top-5 Accuracy (%)
1	Vibranium	79.25
2	Overfit	75.30
3	ACRV_ANU	69.56

Future Work

Data cleaning

1

Optimize training models

2

Iterative cleaning of data and optimization models

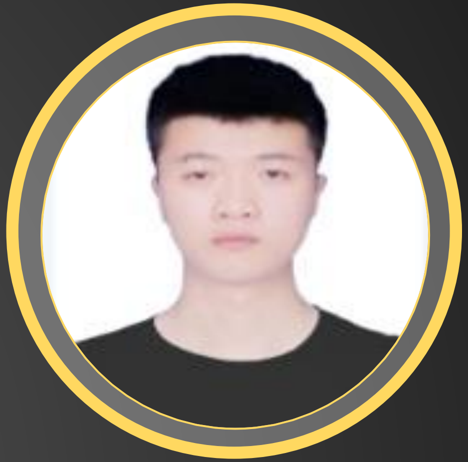
3

Our Team

Jianfeng Zhu

Lele Cheng

Ying Su



Pengcheng Yuan

Lin Ye

Shumin Han

Jia Li

Contact : hanshumin@baidu.com