Open-Set Long-Tailed Recognition and Domain Adaptation

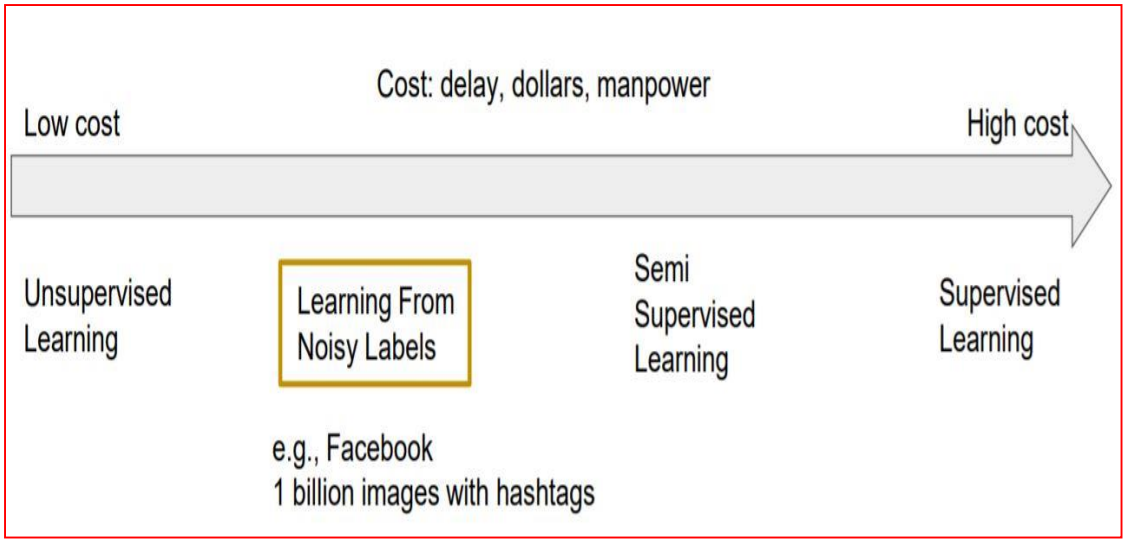
Jenq-Neng Hwang, Professor

Department of Electrical & Computer Engineering

University of Washington, Seattle WA

hwang@uw.edu

Costs of Labelled Data

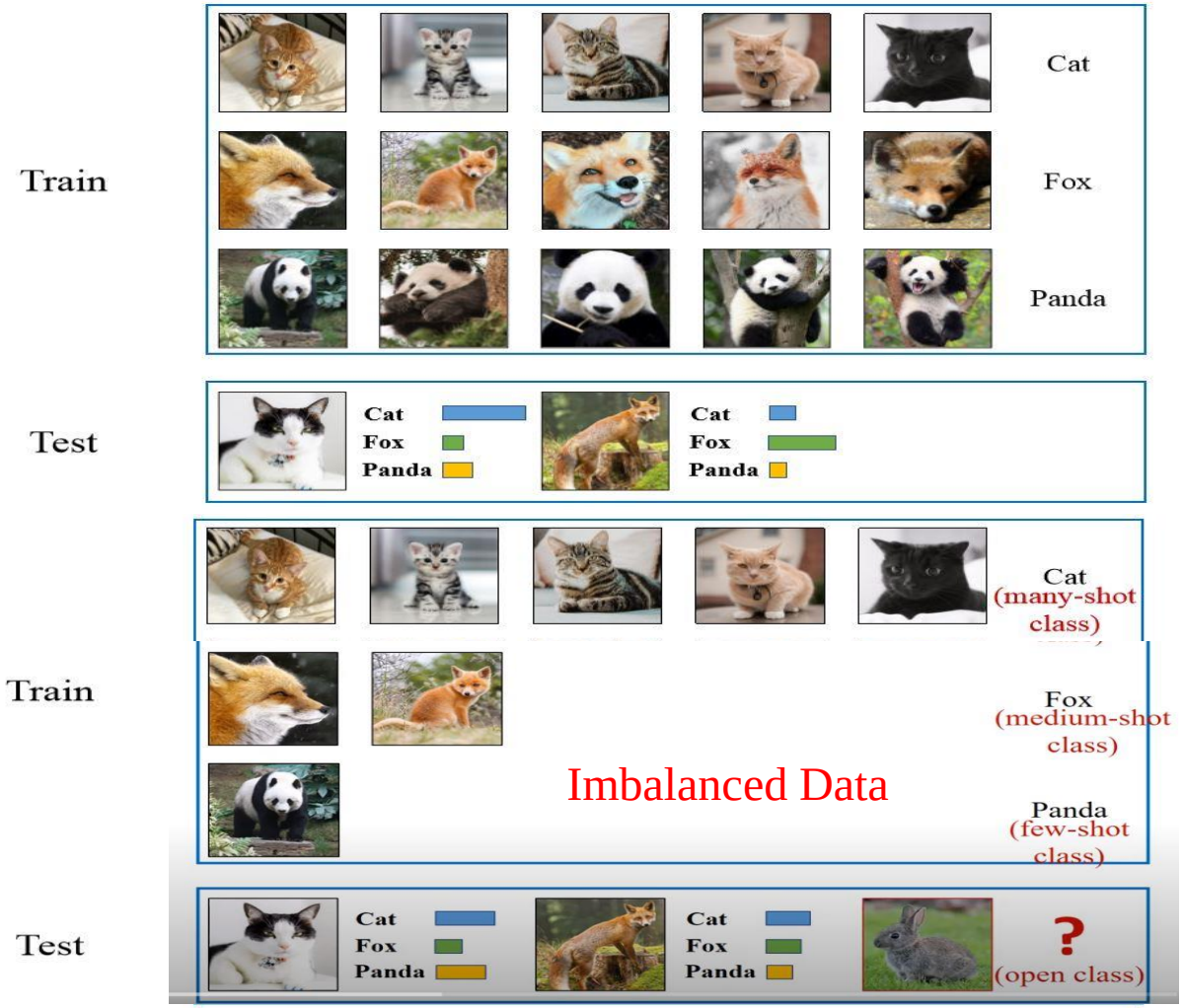


Good generalization ability to new examples — Novel (unseen) class

Good transfer ability from different concepts — Domain changes

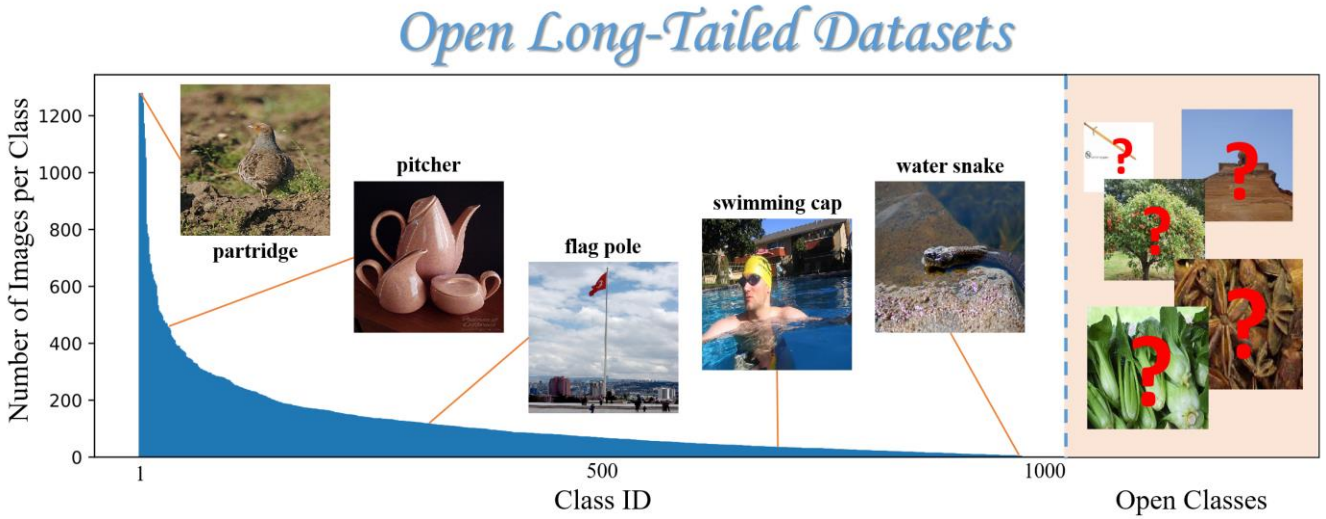
High classification accuracy with limited data — Imbalanced data distribution

Real World Object Recognition



Open-Set Long-Tailed Recognition (OLTR)

- Real world data often have a long-tailed and open-ended distribution
- A recognition system must:
  - generalize from a few known instances (few shot learning)
  - classify among majority and minority classes (LTR)
  - Detect novelty upon a never seen instance (OSR)



Visual Recognition in the Wild: Background

- Long-tailed Sources: training with imbalanced set
- Open Compound Targets: novel class detection
- Reasons
  - Collecting and annotating data is expensive.
  - It is impossible to cover all categories in thetraining dataset

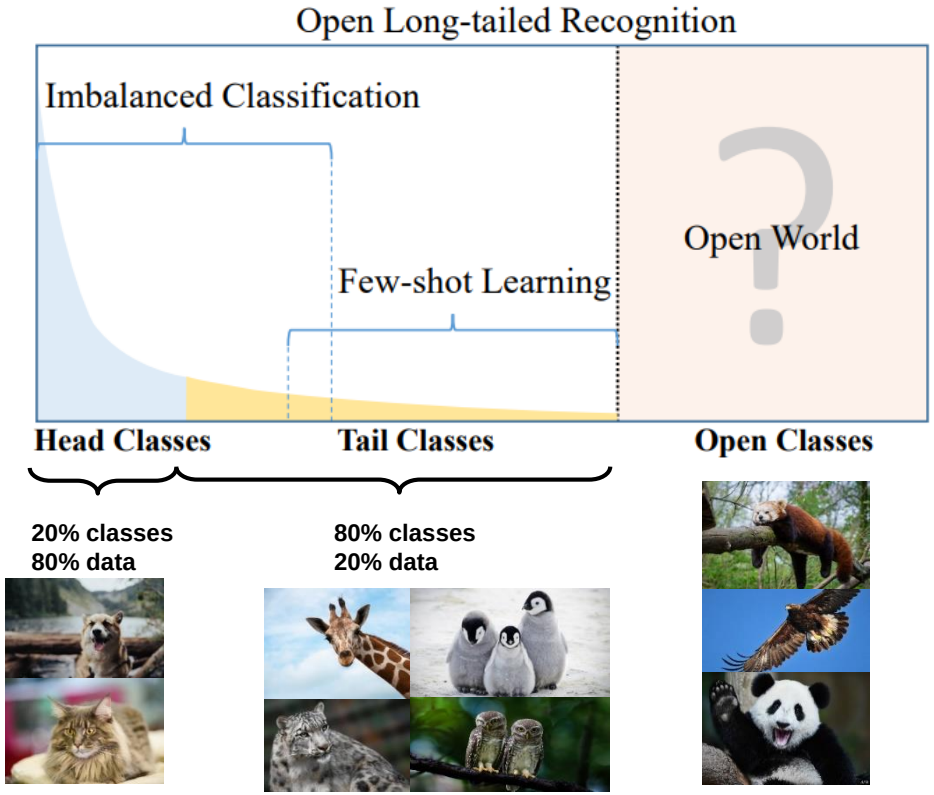
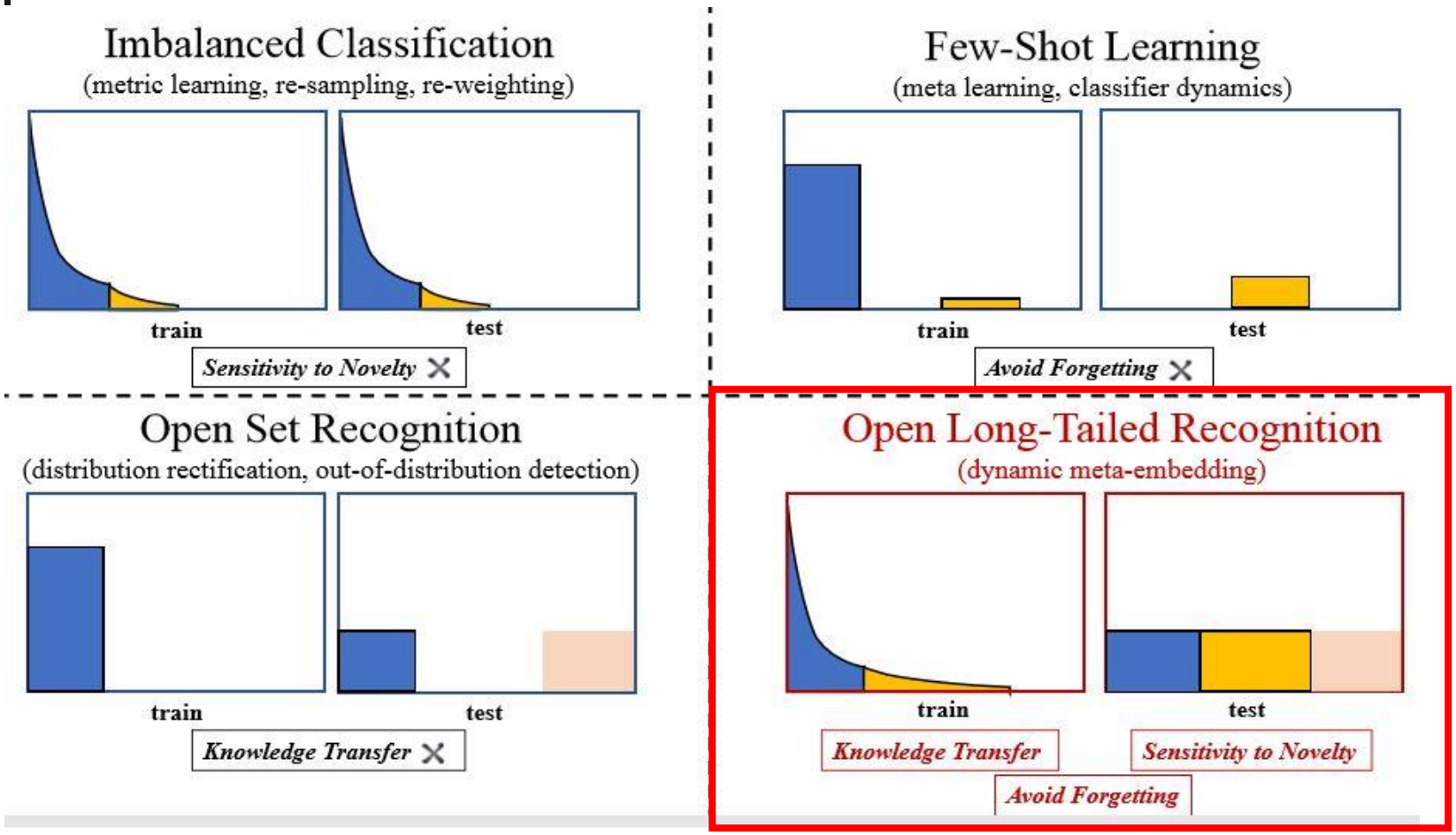


Figure credit: Liu, Ziwei, et al. "Large-scale long-tailed recognition in an open world." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019.

OLTR = LTR + OSR


Imbalanced Classification
(metric learning, re-sampling, re-weighting)
train
test
Sensitivity to Novelty ✕
Few-Shot Learning
(meta learning, classifier dynamics)
train
test
Avoid Forgetting ✕
Open Set Recognition
(distribution rectification, out-of-distribution detection)
train
test
Knowledge Transfer ✕
Open Long-Tailed Recognition
(dynamic meta-embedding)
train
test
Knowledge Transfer
Sensitivity to Novelty
Avoid Forgetting

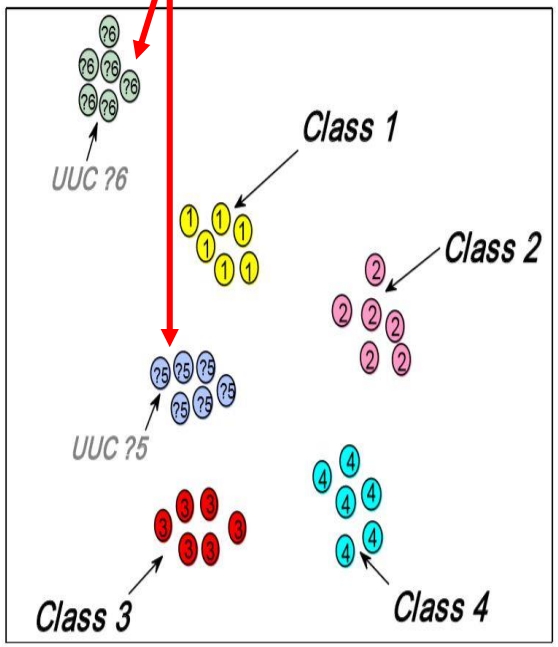
Few-Shot and Active Learning in OLTR

- Few Shot learning (for open-set novel class recognition)
  - recognize novel categories with only few labeled samples in each new class
- Active learning (can be for both open-set and long-tailed)
  - selected samples most worth labeling are those most different (informative) from current labeled pool to finetune the model
  - Out-of-distribution (OOD) samples in seen classes and unseen classes

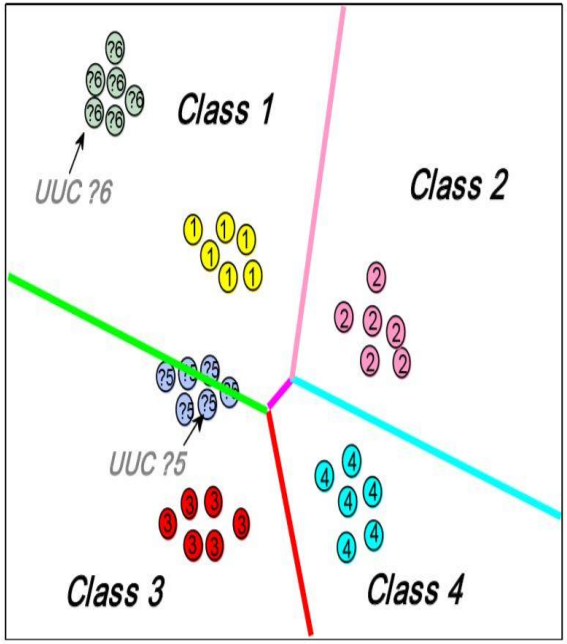
Task Setting	Imbalanced Train/Base Set	#Instances in Tail Class	Balanced Test Set	Open Class	Evaluation: Accuracy Over ?
Imbalanced Classification	√	20~50	X	X	all classes
Few-Shot Learning	X	1~20	√	X	novel classes
Open-Set Recognition	X	N/A	√	√	all classes
Open Long-Tailed Recognition	√	1~20	√	√	all classes

Open-Set Recognition

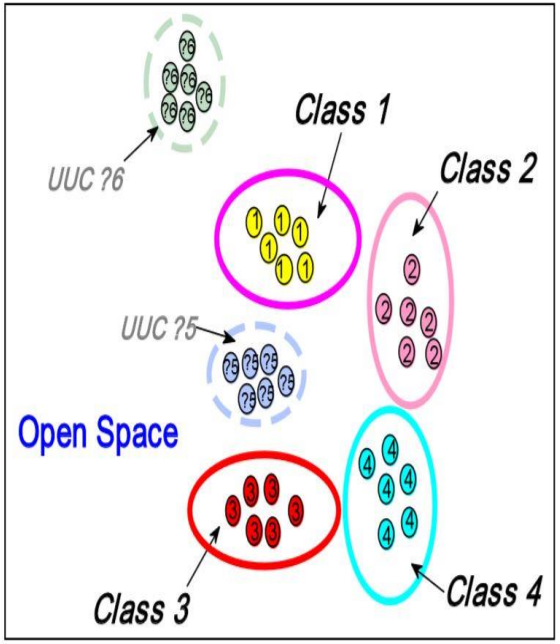
novel class (not in the training data)



(a) Distribution of the original data set.

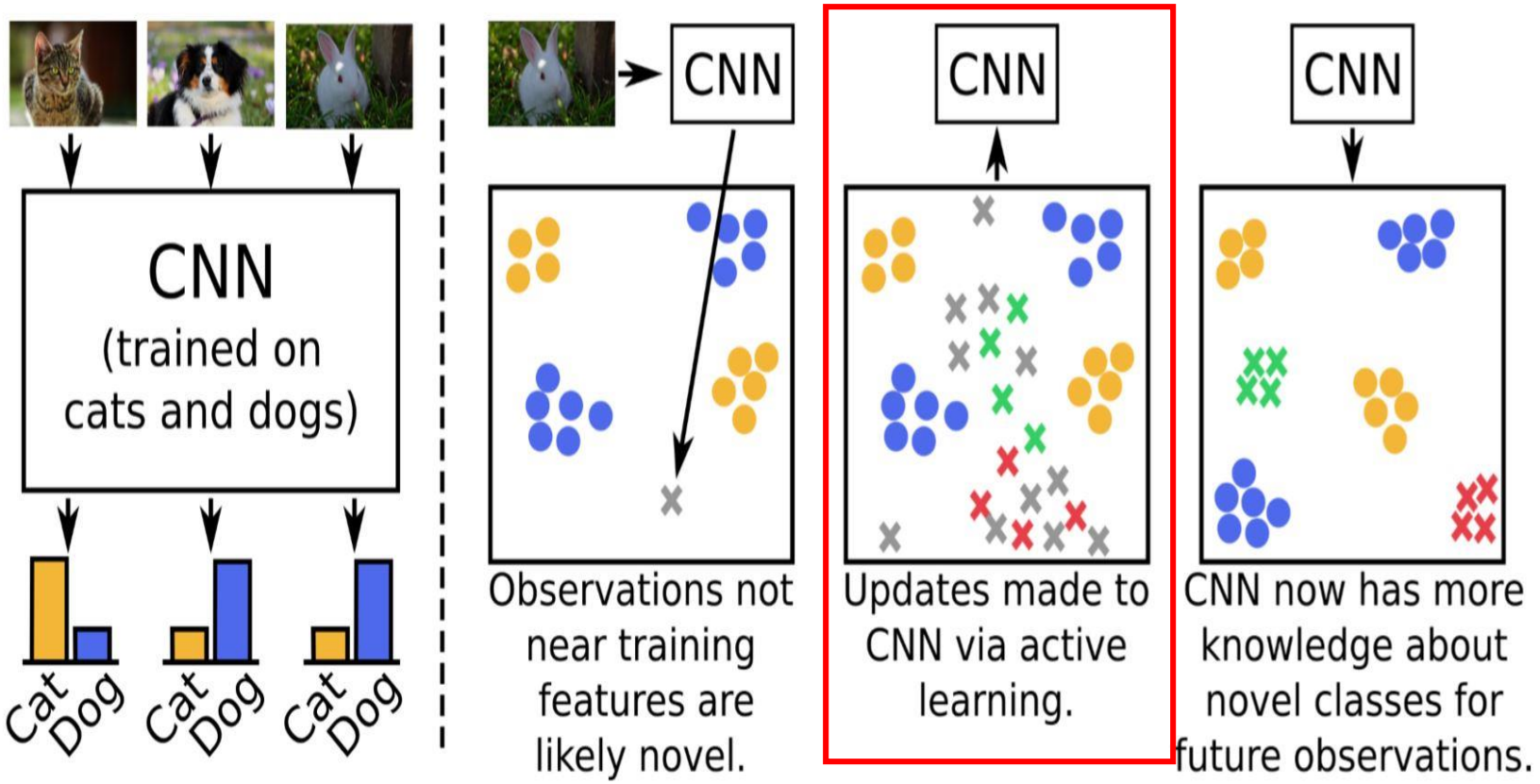


(b) Traditional recognition/classification problem.



(c) Open set recognition/classification problem.

Open Set Recognition via Active Learning



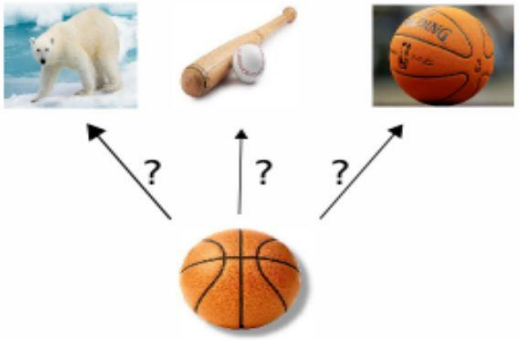
Few-Shot Metric Learning

- Three kinds of datasets:
  - Original training set (source data)
  - A support set (few-shot training set)
  - A query set (test set)
- If the support set contains *K* labeled samples for each of *C* categories, the target few-shot task is called as a *C*-way *K*-shot task.

One-Shot Learning: Same or Different

same	"cow" (speaker #1)	"cow" (speaker #2)	same
different	"cow" (speaker #1)	"cat" (speaker #2)	different
same	"can" (speaker #1)	"can" (speaker #2)	same
different	"can" (speaker #1)	"cab" (speaker #2)	different

Verification tasks (training)



"cot" (speaker #4) "cob" (speaker #4) "cog" (speaker #4)

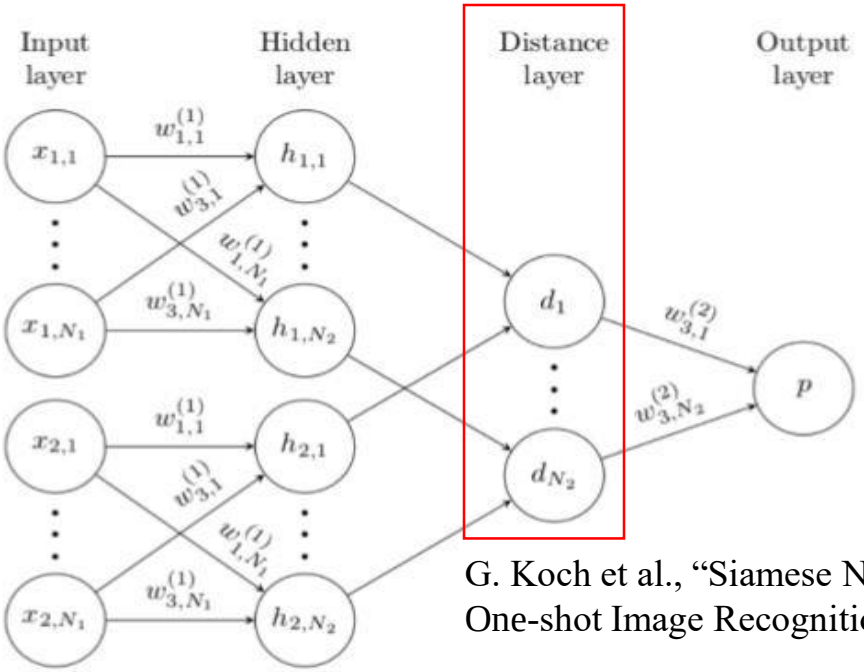
"cob" (speaker #3)

One-shot tasks (test)

A Siamese Network

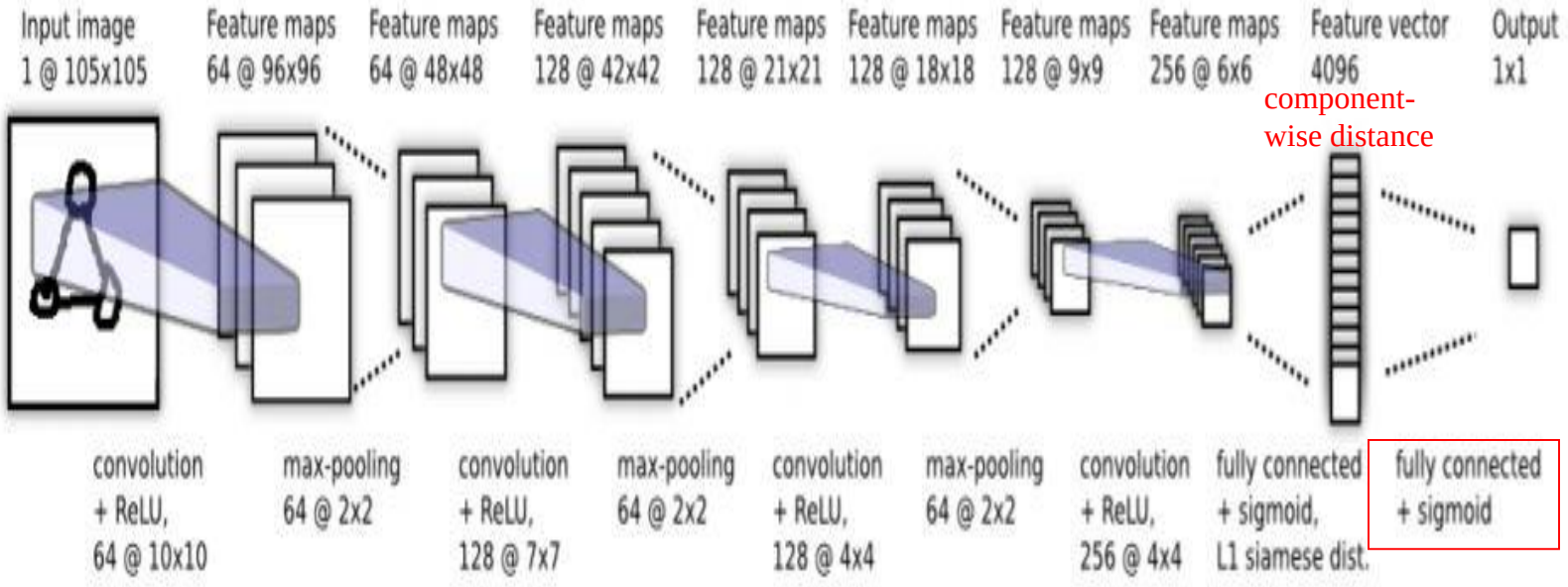
- Learn a Siamese network by "metric-learning loss" and "cross entropy classification loss" from the source dataset
- Reuse the network's features for the target one-shot learning

raw images



G. Koch et al., "Siamese Neural Networks for One-shot Image Recognition," ICML 2015)

Siamese Neural Network for One-Shot Learning



- Siamese twin is not depicted, but joins immediately after the 4096 unit fully-connected layer
- L1 component-wise distance between vectors is computed.

Siamese Network Learning

- Loss Function: a regularized cross-entropy, with a mini-batch of 128

$$\mathcal{L}(x_1^{(i)}, x_2^{(i)}) = \mathbf{y}(x_1^{(i)}, x_2^{(i)}) \log \mathbf{p}(x_1^{(i)}, x_2^{(i)}) + (1 - \mathbf{y}(x_1^{(i)}, x_2^{(i)})) \log (1 - \mathbf{p}(x_1^{(i)}, x_2^{(i)})) + \boxed{\lambda^T |\mathbf{w}|^2}$$

$$y(x_1^{(i)}, x_2^{(i)}) = 1$$

If x_1 and x_2 are from the same class in the *i-th* mini-batch, otherwise

$$y(x_1^{(i)}, x_2^{(i)}) = 0$$

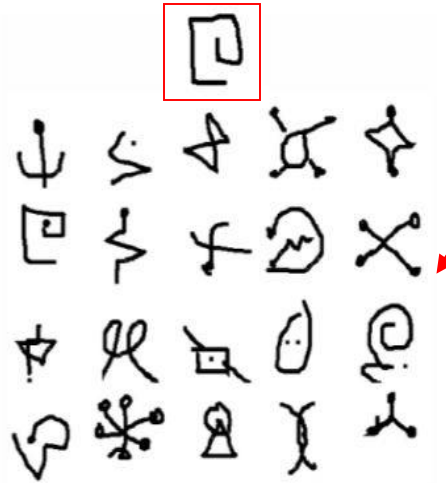
- Optimization: momentum based updating

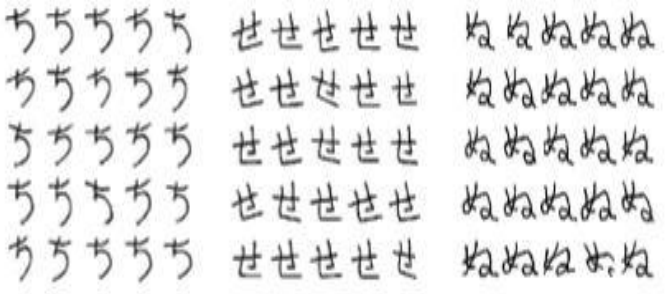
$$\mathbf{w}_{kj}^{(T)}(x_1^{(i)}, x_2^{(i)}) = \mathbf{w}_{kj}^{(T)} + \Delta \mathbf{w}_{kj}^{(T)}(x_1^{(i)}, x_2^{(i)}) + \boxed{2\lambda_j |\mathbf{w}_{kj}|}$$

$$\Delta \mathbf{w}_{kj}^{(T)}(x_1^{(i)}, x_2^{(i)}) = -\eta_j \nabla w_{kj}^{(T)} + \mu_j \Delta \mathbf{w}_{kj}^{(T-1)}$$

Performance

- Omniglot dataset (20-way one-shot learning)



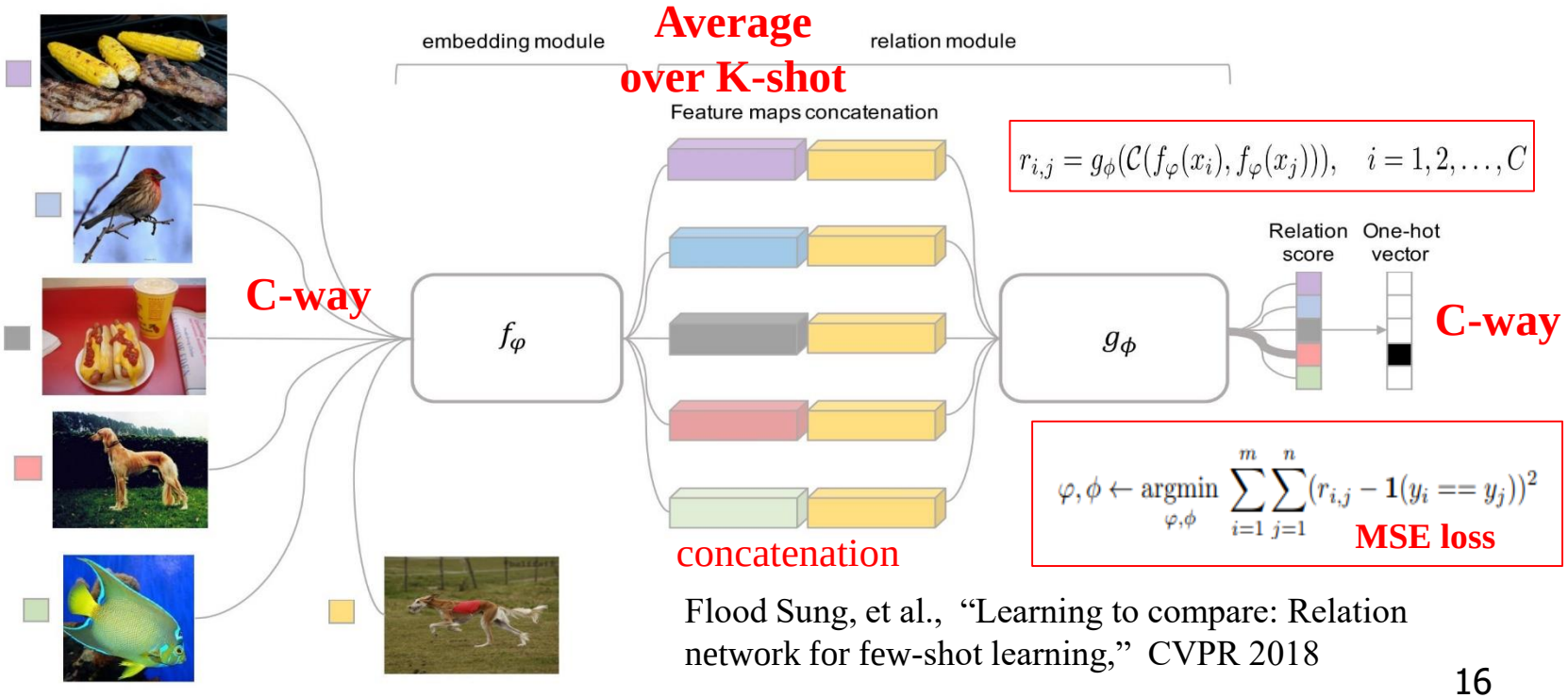


data augmentation

Method	Test
Humans	95.5
Hierarchical Bayesian Program Learning	95.2
Affine model	81.8
Hierarchical Deep	65.2
Deep Boltzmann Machine	62.0
Simple Stroke	35.2
1-Nearest Neighbor	21.7
Siamese Neural Net	58.3
Convolutional Siamese Net	92.0

A Relation Network (RN) for Few Shot Learning

- An embedding module f_ϕ and a relation module g_φ
- Compare the feature embeddings of query images with those from a few labeled images using a learned "metric" function



Flood Sung, et al., "Learning to compare: Relation network for few-shot learning," CVPR 2018

Few-Shot Learning via RN

- For *K*-shot where $K > 1$, element-wise average over the embedding module outputs (mean) of all samples from each training class to form this class' feature map.
- The pooled class-level feature map is combined with the query image feature map, the number of relation scores for one query is always C in both one-shot or few-shot setting (always choose one out of C).
- The meta-learning (ensemble representation) here is to train the auxiliary parameterization relation network that learns how to parameterize a given classification problem in terms of a few-shot sample set.

MSE based Episode Learning

In each training iteration, an episode is formed by randomly selecting C classes from the training set with K labelled samples from each of the C classes to act as the *sample* set $\mathcal{S} = \{(x_i, y_i)\}_{i=1}^{m}$ $(m = K \times C)$, as well as a fraction of the remainder of those C classes' samples to serve as the *query* set $\mathcal{Q} = \{(x_j, y_j)\}_{j=1}^{n}$.

- Mean square error (MSE) loss to train our model, regressing the relation score ri,j to the ground truth: matched pairs have similarity 1 and the mismatched pair have similarity 0

$$\varphi, \phi \leftarrow \underset{\varphi,\phi}{\operatorname{argmin}} \sum_{i=1}^{m}\sum_{j=1}^{n}(r_{i,j} - \mathbf{1}(y_i == y_j))^2$$

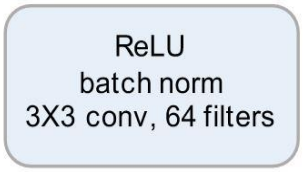
Performance

100 classes
MiniImageNet

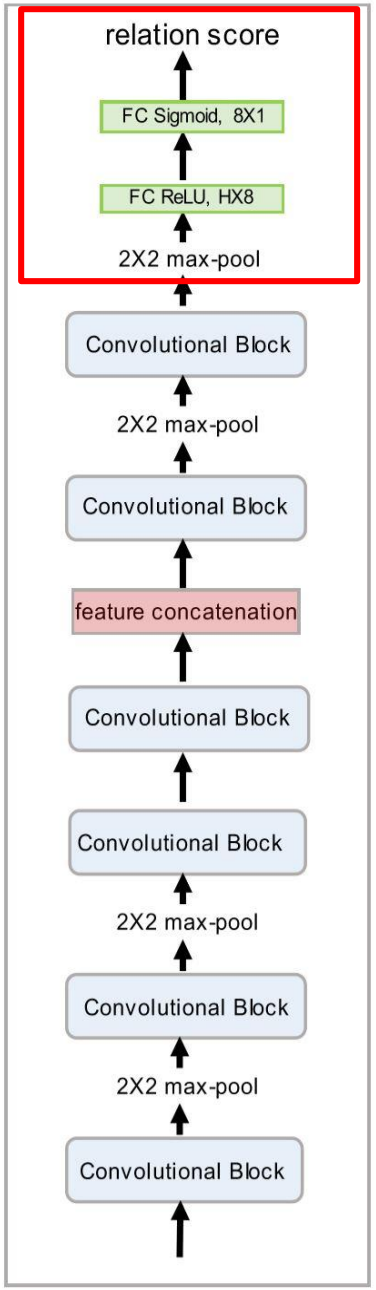
Model	FT	5-way Acc. 1-shot	5-way Acc. 5-shot
MATCHING NETS [39]	N	$43.56 \pm 0.84\%$	$55.31 \pm 0.73\%$
META NETS [27]	N	$49.21 \pm 0.96\%$	-
META-LEARN LSTM [29]	N	$43.44 \pm 0.77\%$	$60.60 \pm 0.71\%$
MAML [10]	Y	$48.70 \pm 1.84\%$	$63.11 \pm 0.92\%$
PROTOTYPICAL NETS [36]	N	$49.42 \pm 0.78\%$	$\mathbf{68.20 \pm 0.66\%}$
RELATION NET	N	$\mathbf{50.44 \pm 0.82\%}$	$65.32 \pm 0.70\%$

Table 2: Few-shot classification accuracies on *mini*Imagenet. All accuracy results are averaged over 600 test episodes and are reported with 95% confidence intervals, same as [36]. For each task, the best-performing method is highlighted, along with any others whose confidence intervals overlap. '-': not reported.

(a) Convolutional Block



(b) RN for few-shot learning



Long-Tailed Recognition (LTR)

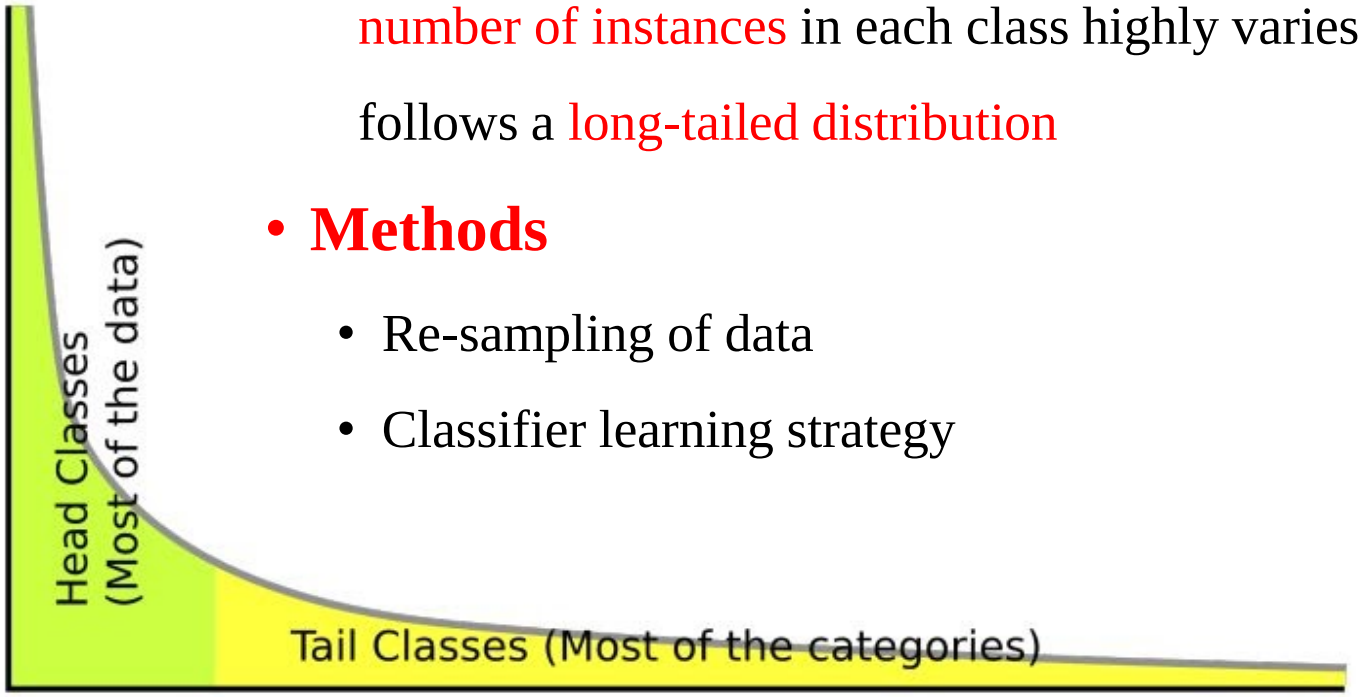
- What is an LTR task

  Recognition in a setting where the

  number of instances in each class highly varies

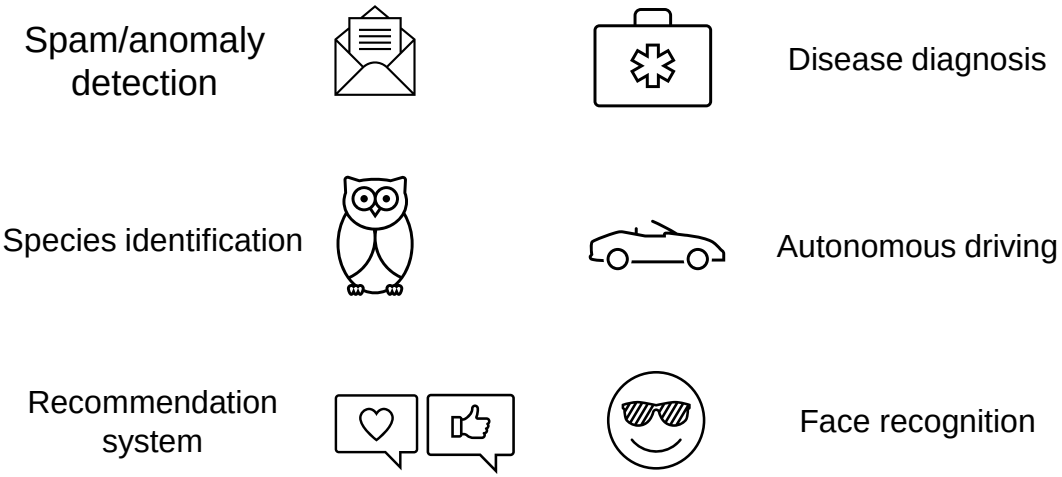
  follows a long-tailed distribution

- **Methods**
  - Re-sampling of data
  - Classifier learning strategy



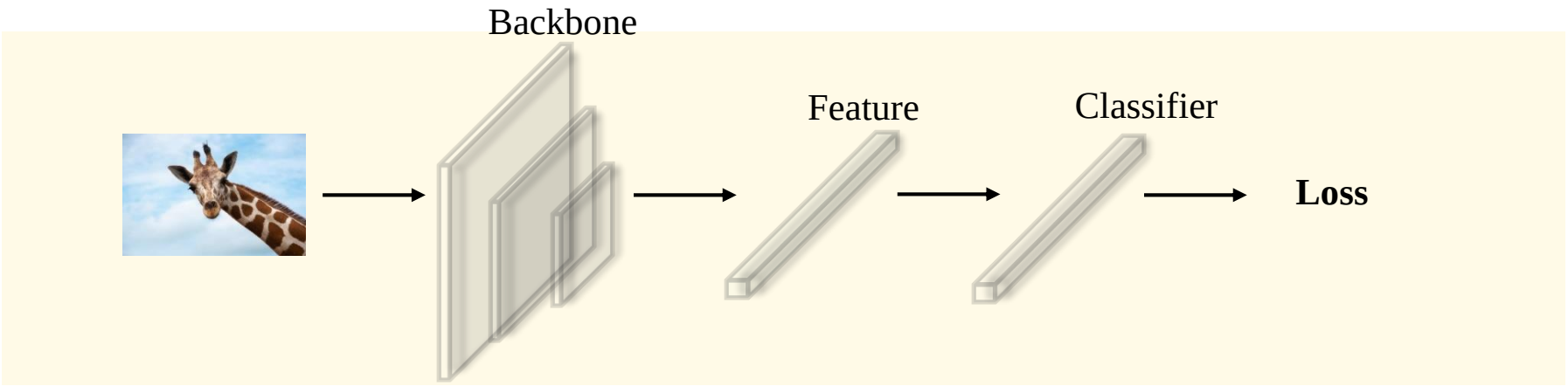
Motivations for LTR

- The long-tailed distribution phenomena inherently occur in most real-world scenarios.



- Not only in recognition, but also a bottleneck in detection & segmentation.

Existing LTR Methods



Re-sampling

- Class-balanced [CVPR16]
- Progressively-balanced [ICLR20]

Augmentation

- Input mixup [ICLR18]
- Rebalanced mixup [ECCVw18]
- Sample generation [AAAI21]

Representation Learning

- LDAM [CVPR19]
- Manifold mix-up [AAAI21]

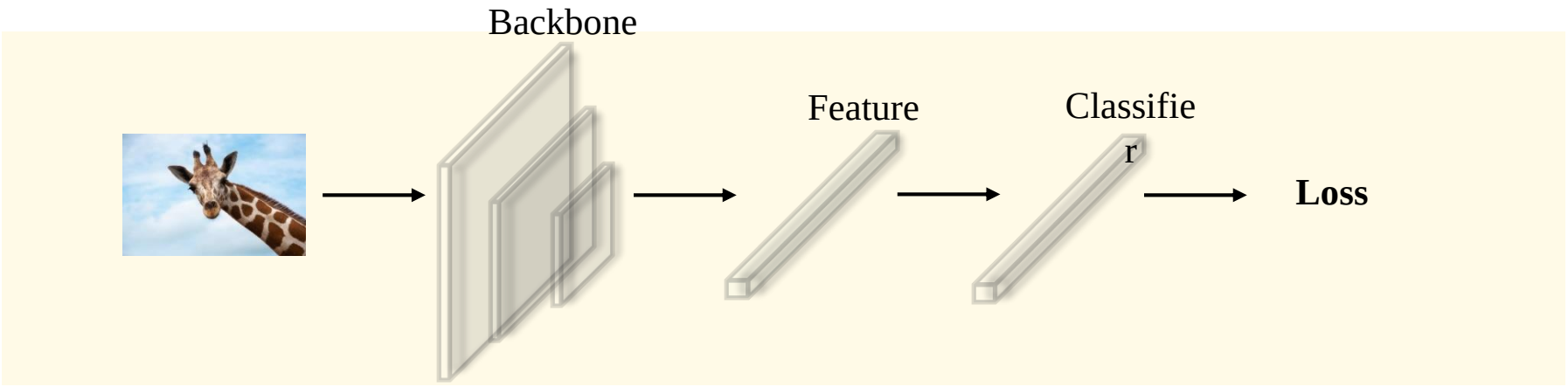
Classifier Adjustment

- τ-norm [ICLR20]
- Learnable weight scaling (LWS) [ICLR20]

Re-weighting

- Class-balanced CE [CVPR19]
- Class-balanced Focal [CVPR19]

Existing LTR Methods



One-stage

- **Re-balancing**
- **Augmentation**

Two-stage

- **Step-1:** pretrain with imbalanced set
- **Step-2:** finetune with re-balancing techniques

Multi-stage (Multi-Expert)

- **Step-1:** Two-stage training of each expert
- **Step-2:** Ensemble

Heads –
Tails ++

Heads +
Tails +

Re-Sampling for LTR

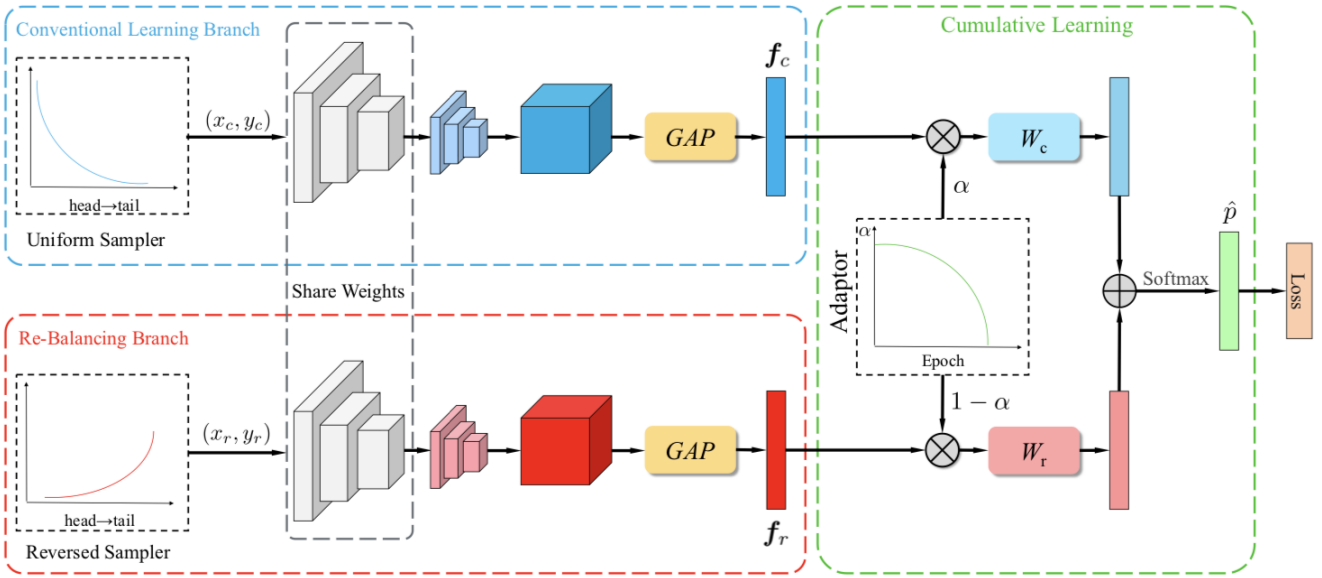
$$p_j = \frac{n_j^q}{\sum_{i=1}^{C} n_i^q}$$

$X = \{x_i, y_i\}, i \in \{1,2,\dots,n\}$ is the training set, C is the number of classes, n_j is the number of training sample for class j

- **Instance-balanced (IB) sampling**
  - $q = 1$
  - For imbalanced training datasets, instance-balanced sampling has been shown to be **sub-optimal,** leading to lower accuracy, especially for balanced test sets.
- **Class-balanced (CB) sampling**
  - $q = 0,\ p_j^{CB} = 1/C$
- **Square-root sampling** (Mikolov et al., 2013; Mahajan et al., 2018)
  - $q = 0.5$
- **Progressively-balanced sampling** (Cui et al., 2018; Cao et al., 2019)
  - $p_j^{PB}(t) = \left(1 - \frac{t}{T}\right) p_j^{IB} + \frac{t}{T} p_j^{CB}$
  - As epochs progress, sampling goes from instance-balanced to class-balanced sampling

Bilateral-Branch Network (BBN) for Long-Tailed Training

- Two parallel branches for re-sampling
- Cumulative learning for re-weighting



Re-sampling for each batch (128 images)

Backbone for feature extraction (with shared weights)

Re-weighting by adaptor:
1. First weights more on the conventional learning branch
2. Then fine-tune on the re-balancing branch automatically

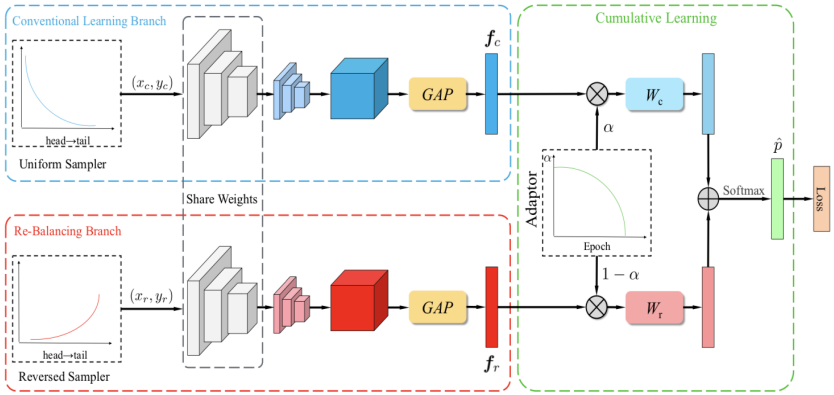
Mix-up

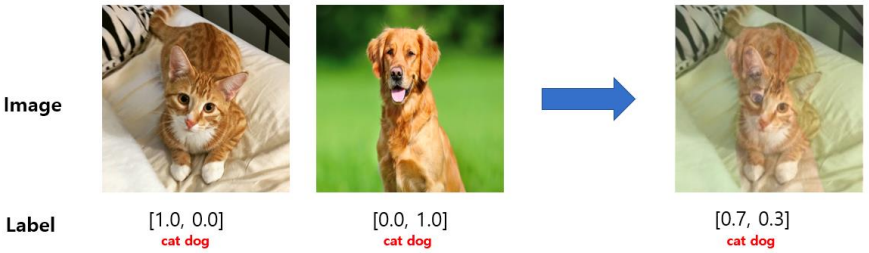
B. Zhou et al., "BBN: Bilateral-Branch Network with Cumulative Learning for Long-Tailed Visual Recognition," CVPR 2020

Bilateral-Branch Network

$$\begin{aligned}\hat{x} &= \lambda x_i + (1-\lambda)x_j, \\ \hat{y} &= \lambda y_i + (1-\lambda)y_j,\end{aligned}$$

where $\lambda \in [0, 1]$ is a random number





$$\mathcal{L} = \alpha E(\hat{\boldsymbol{p}}, y_c) + (1-\alpha)E(\hat{\boldsymbol{p}}, y_r)$$

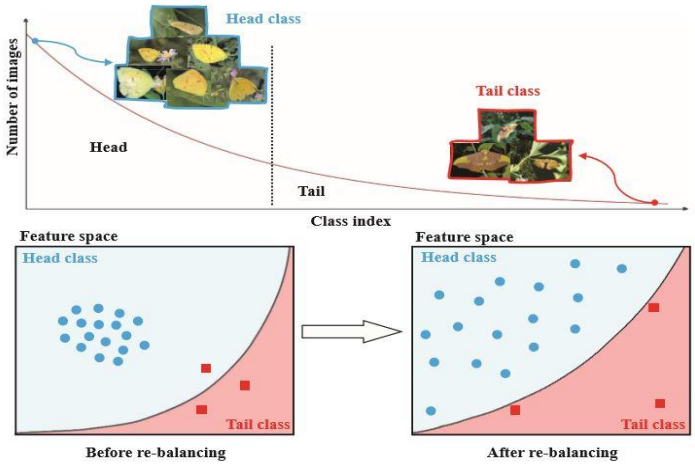
weighted cross-entropy loss

- **Feature Mix-up (ICLR'18)**: a data augmentation technique in feature space for robust representation learning

  $\tilde{x} = \lambda x_i + (1-\lambda)x_j$ where x_i, x_j are input feature vectors

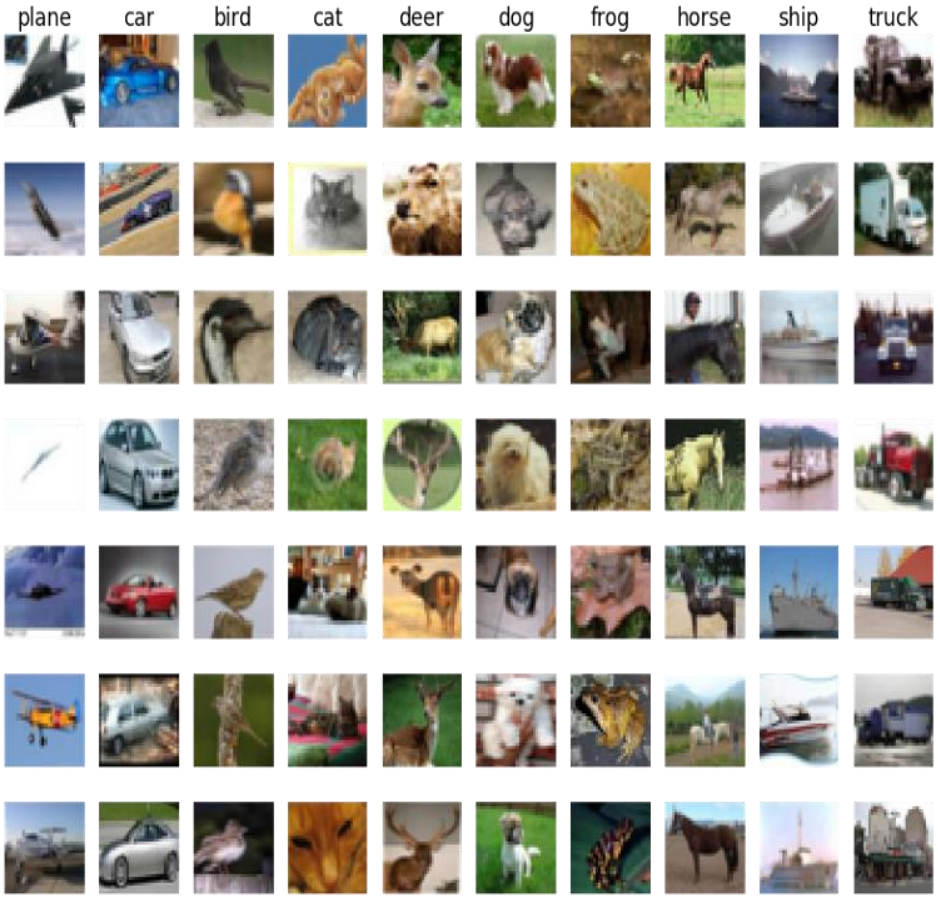
  $\tilde{y} = \lambda y_i + (1-\lambda)y_j$ where y_i, y_j are labels
- During inference, the test samples are fed into both branches and two features f_c' and f_r' are obtained. Because both branches are equally important, α is simply fiexed to 0.5 in the test phase

BBN Performance

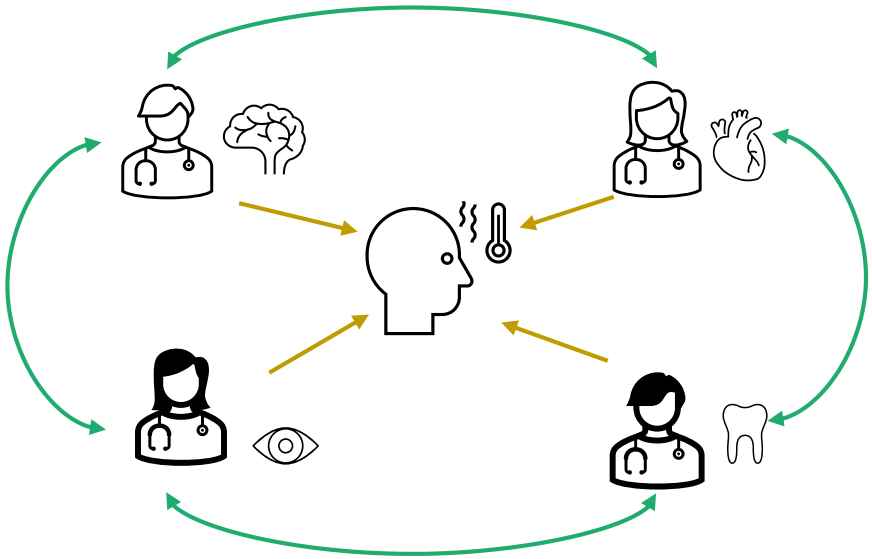


Dataset	Long-tailed CIFAR-10			Long-tailed CIFAR-100		
Imbalance ratio	100	50	10	100	50	10
CE	29.64	25.19	13.61	61.68	56.15	44.29
Focal [17]	29.62	23.28	13.34	61.59	55.68	44.22
Mixup [33]	26.94	22.18	12.90	60.46	55.01	41.98
Manifold Mixup [27]	27.04	22.05	12.97	61.75	56.91	43.45
Manifold Mixup (two samplers)	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**

Top-1 error rate



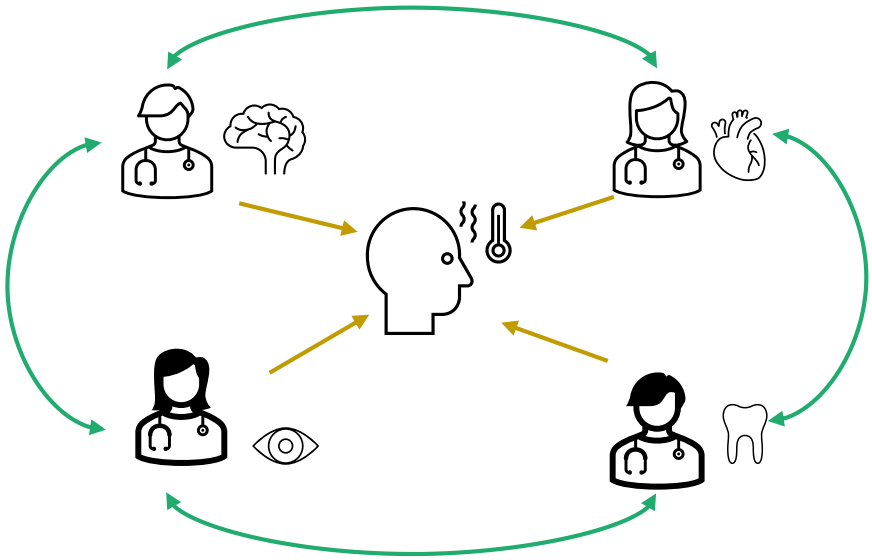
Multiple Complementary Experts Strategy

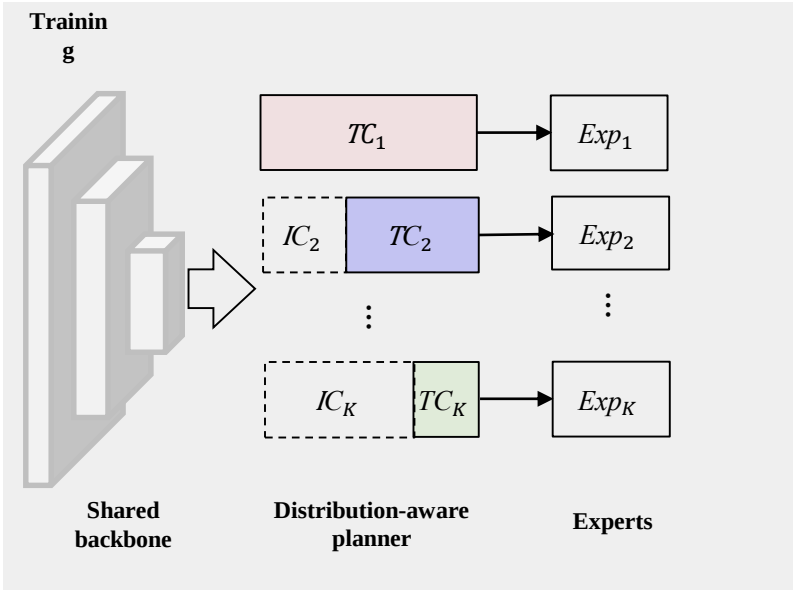


Complementary Experts

- **(Medical school)** They share elementary knowledge from the most diverse data source;
- **(Specialist)** They are professional at splits of data respectively, and aware of what they do not specialize in;
- **(Panel discussion)** Opinions from the experienced experts (who see more data) complement the judgment from junior experts (who see less) for optimal decision.

Ally Complementary Experts (ACE) for LTR



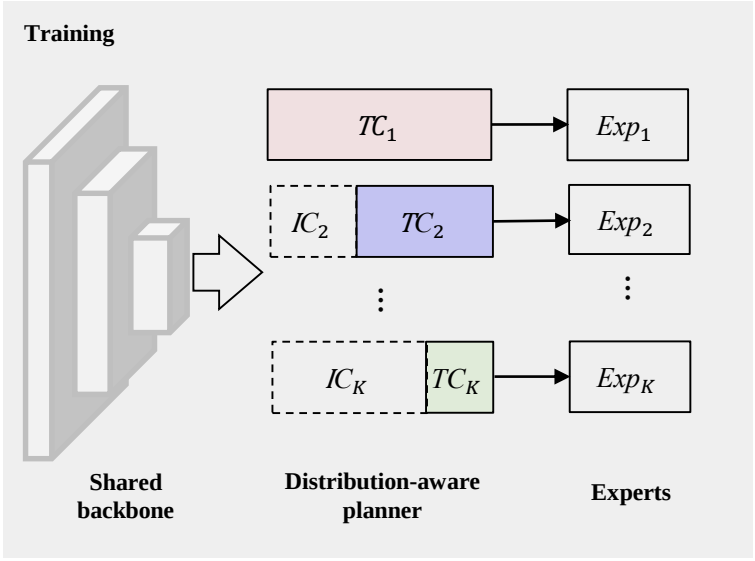


TC: Target categories **IC:** Interfering categories

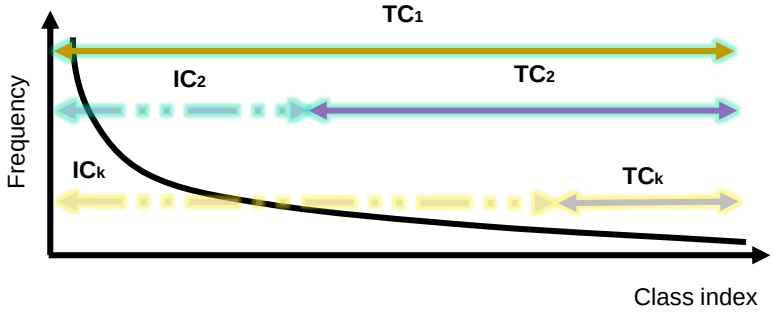
- Involve multiple specialists' insights
- Panel discussion to exclude interfering potentials

Jiarui Cai, et al., "ACE: Ally Complementary Experts for Solving Long-Tailed Recognition in One-Shot," ICCV 2021

Ally Complementary Experts (ACE)

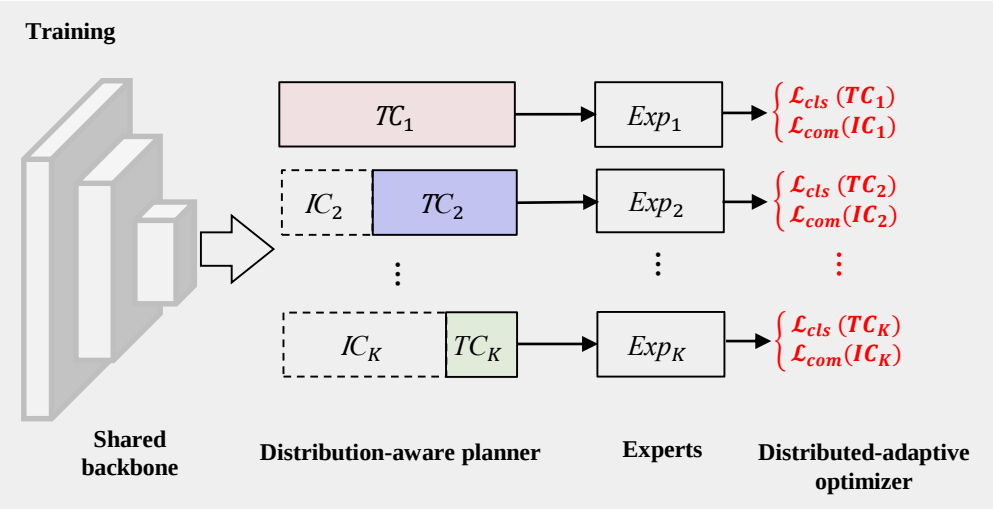


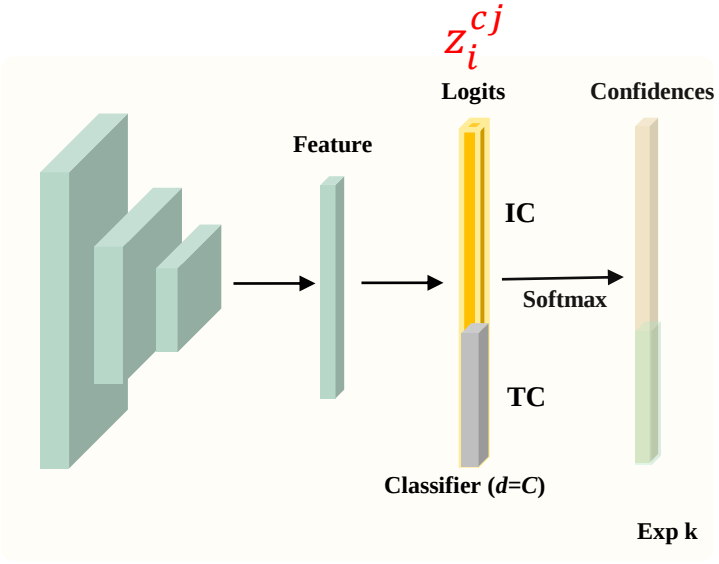
TC: Target categories **IC:** Interfering categories



- Target categories (TC)
  - Overlapping for each expert, especially on sample-few categories.
  - With different dominating classes, each expert has its own strength.
- Interfering categories (IC)
  - Learn not to bring ambiguity in the categories that have never been seen.

Ally Complementary Experts (ACE)



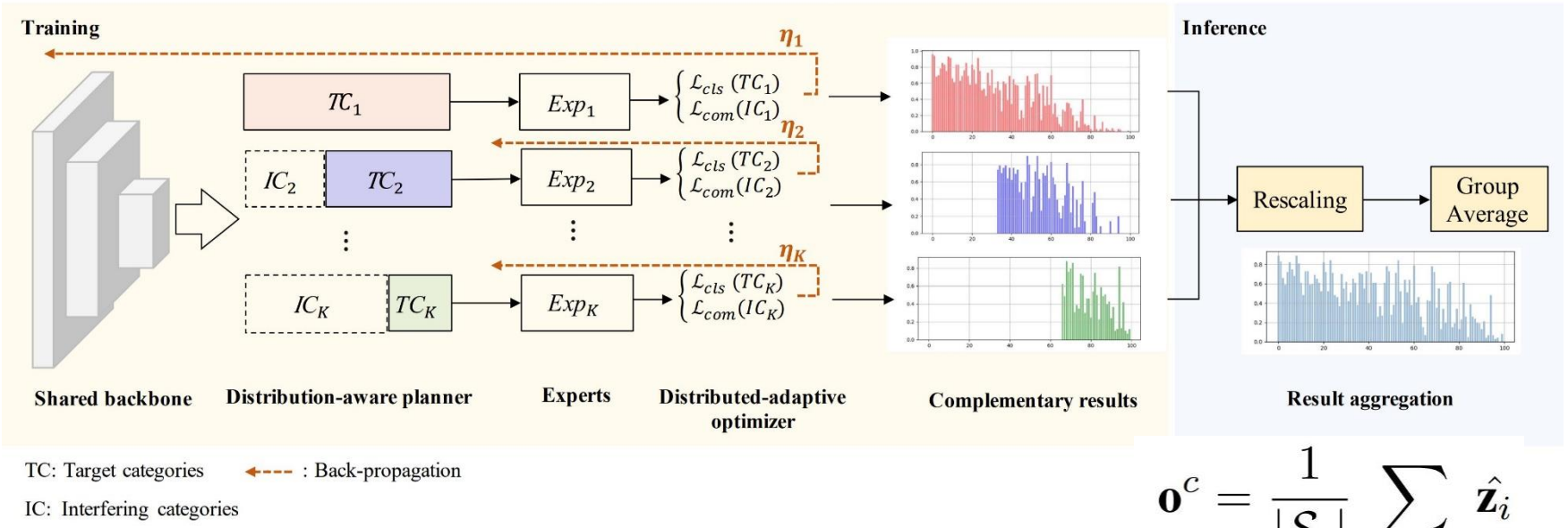


TC: Target categories IC: Interfering categories

Cross-entropy loss **Regularization**

$$L^i = L^i_{cls} + L^i_{com} = -\sum^{\{TC_i\}} ylog(\sigma(z_i)) + \sum_{c_j}^{\{IC_i\}} \left\| z_i^{cj} \right\|^2$$

Ally Complementary Experts (ACE)



SoftMax operation is applied on **o** to obtain the classification confidence.

$$\mathbf{o}^c = \frac{1}{|\mathcal{S}_c|} \sum_{\mathcal{E}_i \in \mathcal{S}^c} \hat{\mathbf{z}}_i$$

$$\hat{\mathbf{z}}_i = \frac{\|\mathbf{w}_1\|^2}{\|\mathbf{w}_i\|^2} \cdot \mathbf{z}_i$$

Learnable weight scaling (LWS)

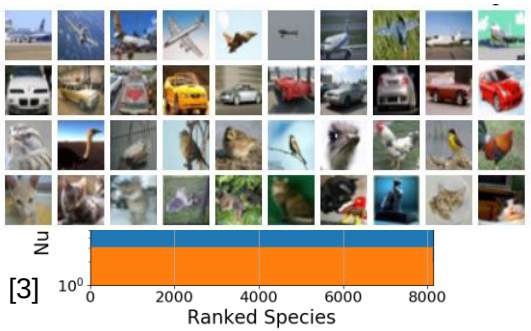
- **Output**
  - Average scaled logits among all experts for each class.
  - Softmax.

LTR Benchmark Datasets

Closed-set Long-tailed Dataset: CIFAR-LT[1], ImageNet-LT[2], iNaturalist [3]

Dataset	Source	Num of class	Imbalance Factor = N_{max}/N_{min}
CIFAR-LT	Artificially resample from CIFAR	10/100	10/50/100
ImageNet-LT	Artificially resample from ImageNet 2012	1000	256
iNaturalist 2018	Real-world dataset	8142	500

[1]



[3]

[2]

[3]

Ally Complementary Experts (ACE)

SOTA on CIFAR-LT-100 (IF=100) [1]

Type	Method	Multi experts	Strong aug	Accuracy			
				All	Many	Medium	Few
One-Stage	Baseline (ResNet-32)			38.3	65.2	37.1	9.1
	CB resampling [9]§			36.0 (-1.7)	59.0 (-6.2)	35.4 (-1.7)	10.9 (+1.8)
	Focal loss [14]			37.4 (-0.9)	64.3 (-0.9)	37.4 (+0.3)	7.1 (-2.0)
	CB Focal loss [3]§			38.7 (+0.4)	65.0 (-0.2)	37.6 (+0.5)	10.3 (+1.2)
	Progressive [10]			39.4 (+1.1)	63.3 (-1.9)	38.8 (+1.7)	13.1 (+4.0)
	ReMix [2]		✓	40.9 (+2.6)	69.6 (+4.4)	40.7 (+3.0)	8.8 (-0.3)
	Mixup [29]		✓	41.2 (+2.9)	70.7 (+5.5)	40.4 (+3.3)	8.8 (-0.3)
	BBN [33]	✓	✓	39.4 (+1.1)	47.2 (-18.0)	49.4 (+12.3)	19.8 (+10.7)
	ACE (3 experts)	✓		**48.4** (+10.1)	**65.8** (+0.6)	**51.9** (+14.8)	**25.0** (+15.9)
	ACE (4 experts)	✓		**49.6** (+11.2)	**66.3** (+1.1)	**52.8** (+15.7)	**27.2** (+18.1)
Multi-Stage	τ-norm [10]			43.2	65.7	43.6	17.3
	cRT [10]			43.3	64.0	44.8	18.1
	LDAM+DRW [1]			42.0	61.5	41.7	20.2
	LDAM+LFME [25]	✓		43.8	-	-	-
	LDAM+M2m [11]		✓	43.5	-	-	-
	CAM [31]	✓	✓	47.8	-	-	-
	RIDE [22] (2 experts)	✓		47.0	67.9	48.4	21.8
	RIDE [22] (3 experts)	✓		48.0	68.1	49.2	23.9
	RIDE [22] (4 experts)	✓		49.1	69.3	49.3	26.0

[1] Cui, Yin, et al. "Class-balanced loss based on effective number of samples." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019.

Expert Based Methods

Method	Data for Experts	Relationship of Experts	Number of Training Stages	Majority Gain	Minority Gain
LFME [26]	non-overlapping splits	independent	2	+	+
RIDE [23]	same full set	competing and complementary	3	++	+
ACE (Ours)	overlapping splits	supportive and complementary	1	+	++

Ally Complementary Experts (ACE)

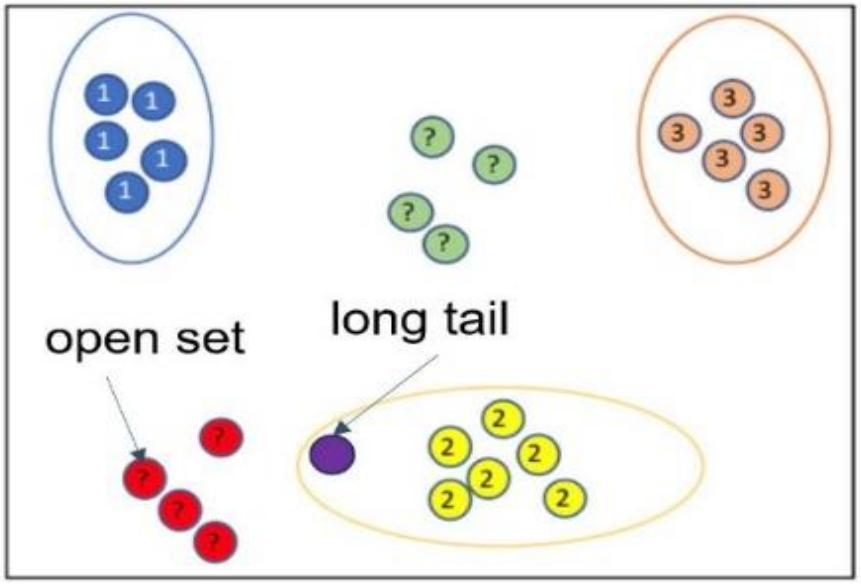
SOTA on ImageNet-LT[1], iNaturalist[2]

Method	ImageNet-LT			iNaturalist
	Res10	Res50	ResX50	Res50
Baseline	20.9	41.6	44.4	66.1
FSLwF [5]	28.4	-	-	-
Range Loss [30]	30.7	-	-	-
Lifted Loss [17]	30.8	-	-	-
Focal loss [14]	30.5	-	-	60.3
CB Focal loss [3]	-	-	-	61.1
BBN [33]	-	48.3	49.3	68.0
ACE (3 experts)	**48.2**	**54.7**	**56.6**	**72.9**
OLTR [15]	34.1	-	46.3	63.9
NCM [10]	35.5	44.3	47.3	-
LDAM+DRW [1]	36.0	-	-	68.0
cRT [10]	41.8	47.3	49.5	65.2
τ-norm [10]	40.6	46.7	49.4	65.6
LWS [10]	41.4	47.7	49.9	65.9
CAM [31]	43.1	-	-	70.9
LFME [25]	47.1	-	-	-
RIDE [22]†	-	54.4	55.9	71.4
RIDE [22]‡	-	54.9	56.4	72.2

[1] Liu, Ziwei, et al. "Large-scale long-tailed recognition in an open world." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019.
[2] Van Horn, Grant, et al. "The inaturalist species classification and detection dataset." Proceedings of the IEEE conference on computer vision and pattern recognition. 2018.

How About Open-Set and LTR Together?

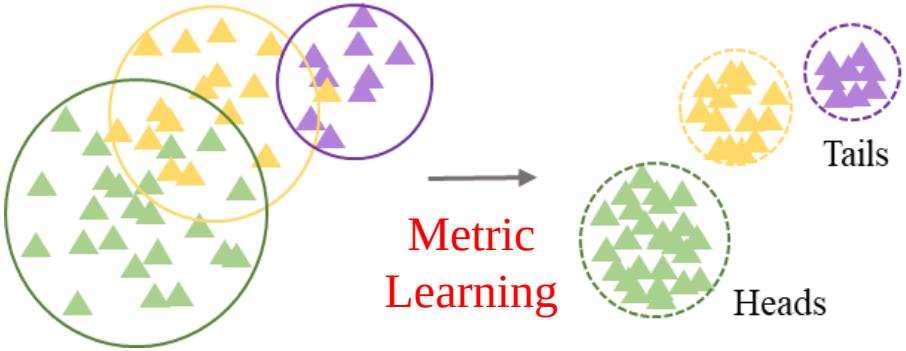
How do we know a datum is in an open-set (novel) class or is a long-tailed datum in the existing training classes?

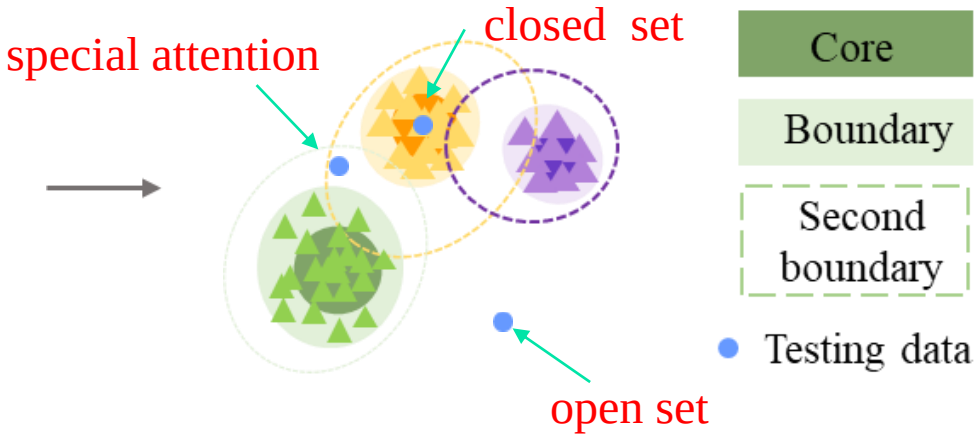


Challenges of Open-Set Recognition (OSR)

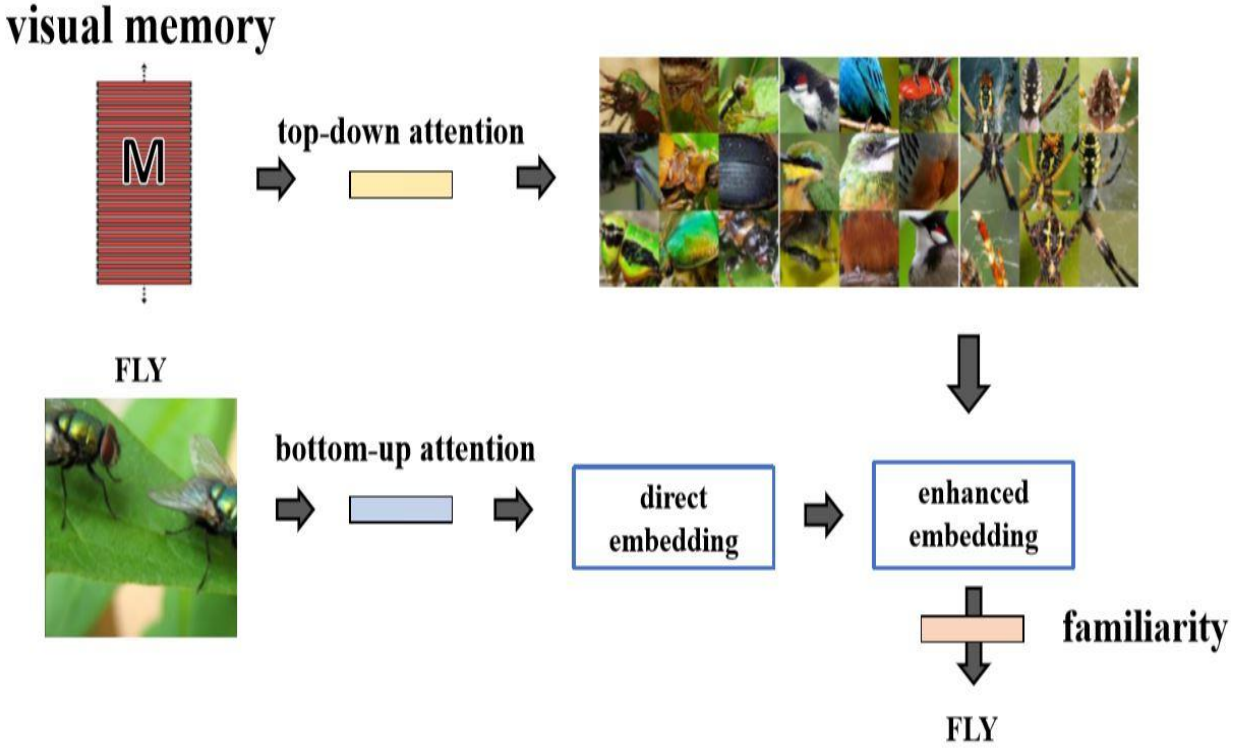
- Is It True? For an unknown input, all classes would have low probability and that thresholding on uncertainty would reject unknown classes.
- Recent deep learning vulnerability papers have shown "fooling" or "rubbish" images that are visually far from the desired class but produce high-probability/confidence scores.
- Meta-Recognition: ***Activation vector* (AV),** logit scores from the penultimate layer of deep networks (the fully connected layer before SoftMax), can be used
  - not an independent per-class score estimate
  - but provide a distribution of what classes are "related"

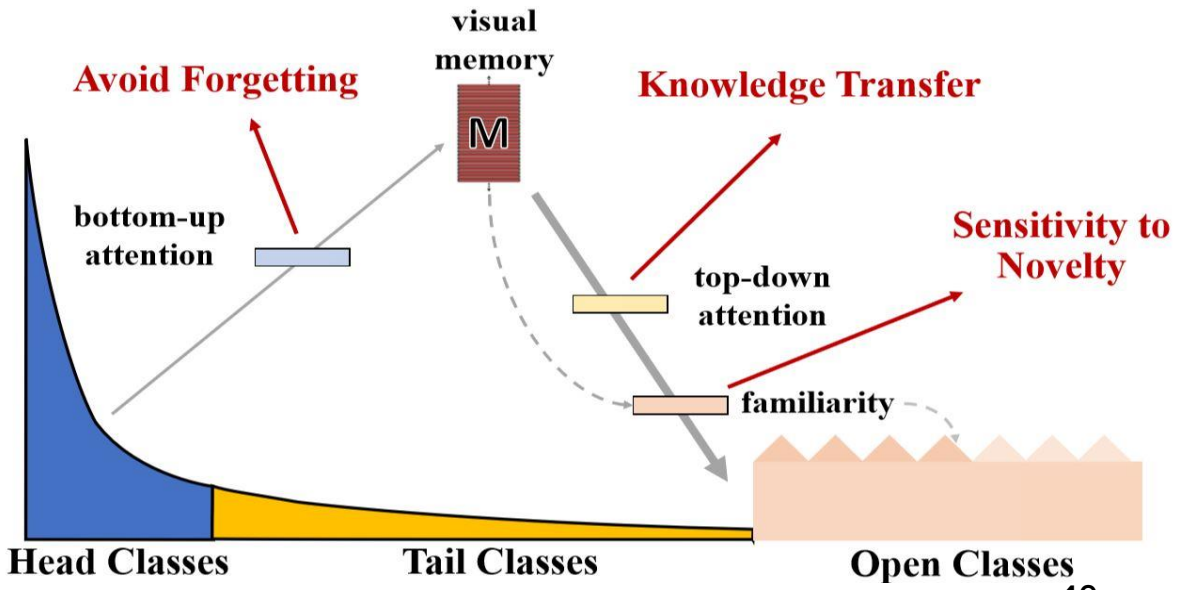
Open-Set Data Detection





Visual Memory for OLTR



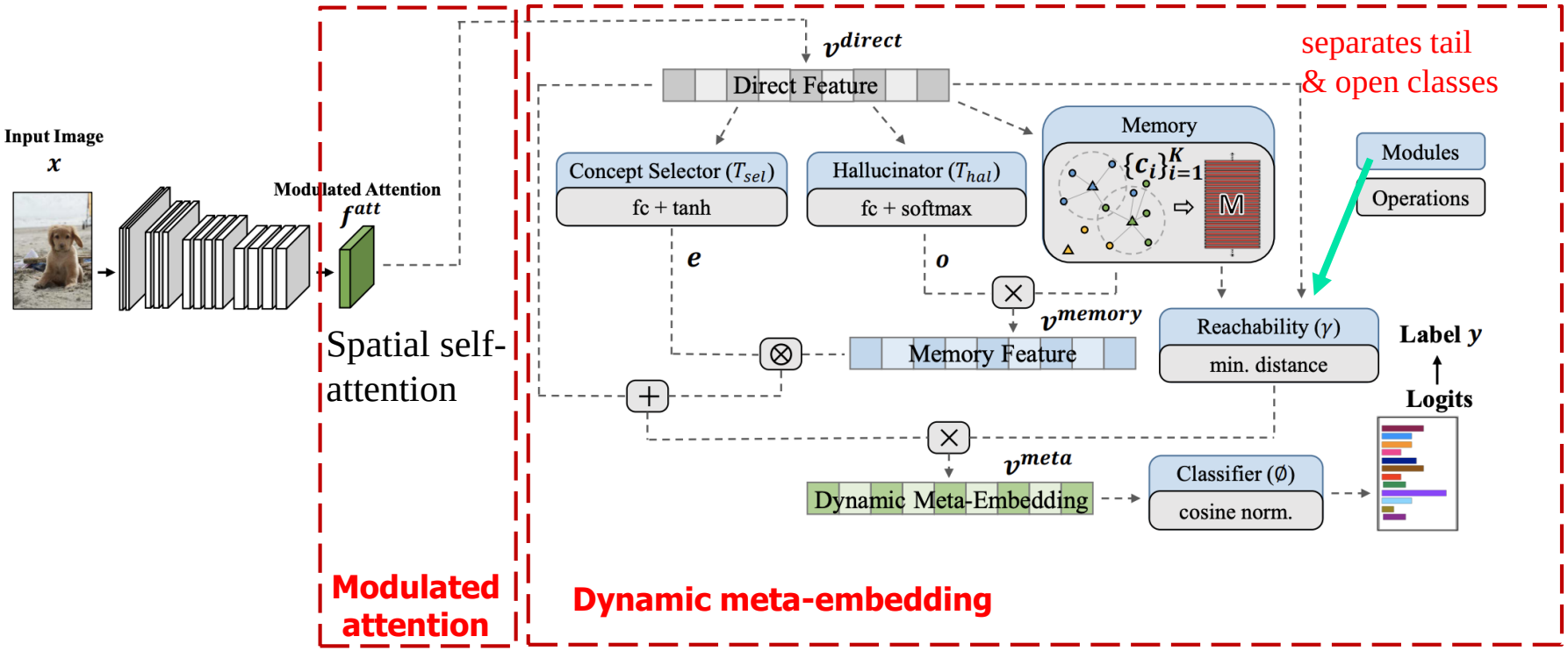


Liu, Ziwei, et al. "Large-scale long-tailed recognition in an open world." *IEEE CVPR* 2019.

OLTR Principles

- "Direct" metric-learned (bottom-up spatial attention) embedding features are useful for head-class recognition
- "Enhanced" centroid memory (top-down class weighted attention) can boost the tail-class recognition
- How much to trust enhanced memory or direct feature is based on selector weighting
- Reachability (familiarity or sensitivity to novelty) is used to control the degree of being recognized by the existing known classes, similar to few-shot classes (not in the memory) matching, a threshold of "$r>0.1$" is set for novel classes

An OLTR System: Metric + Dynamic Meta Learning



discriminate visual concepts between head and tail classes

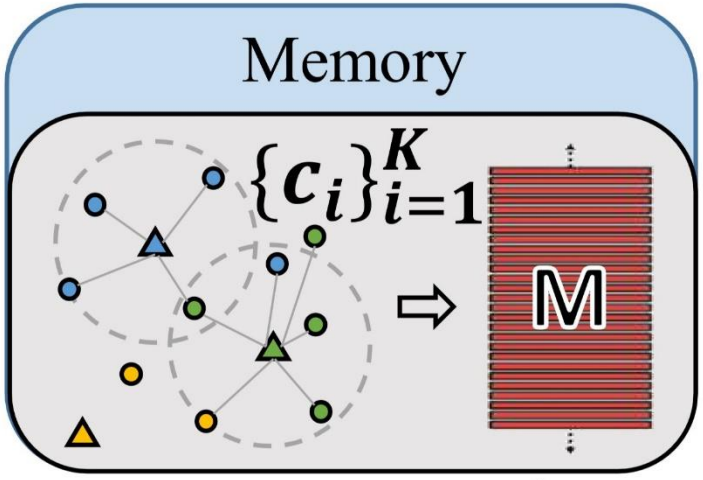
relates features between head and tail classes

the **min distance** between the direct feature and the centroids in memory

Liu, Ziwei, et al. "Large-scale long-tailed recognition in an open world." *IEEE CVPR* 2019.

Learning Visual Memory

- Visual Memory: $M = \{c_i\}_{i=1}^{K}$ are the discriminative centroids of all the training data, where K is the number of training classes.
  - Neighborhood Sampling: The metric-learned centroid c_i of each group is updated by the direct feature of this mini-batch.
  - Propagation: the direct feature v^{direct} and the centroids are alternatively updated (center loss metric learning)



Center Loss Metric Learning

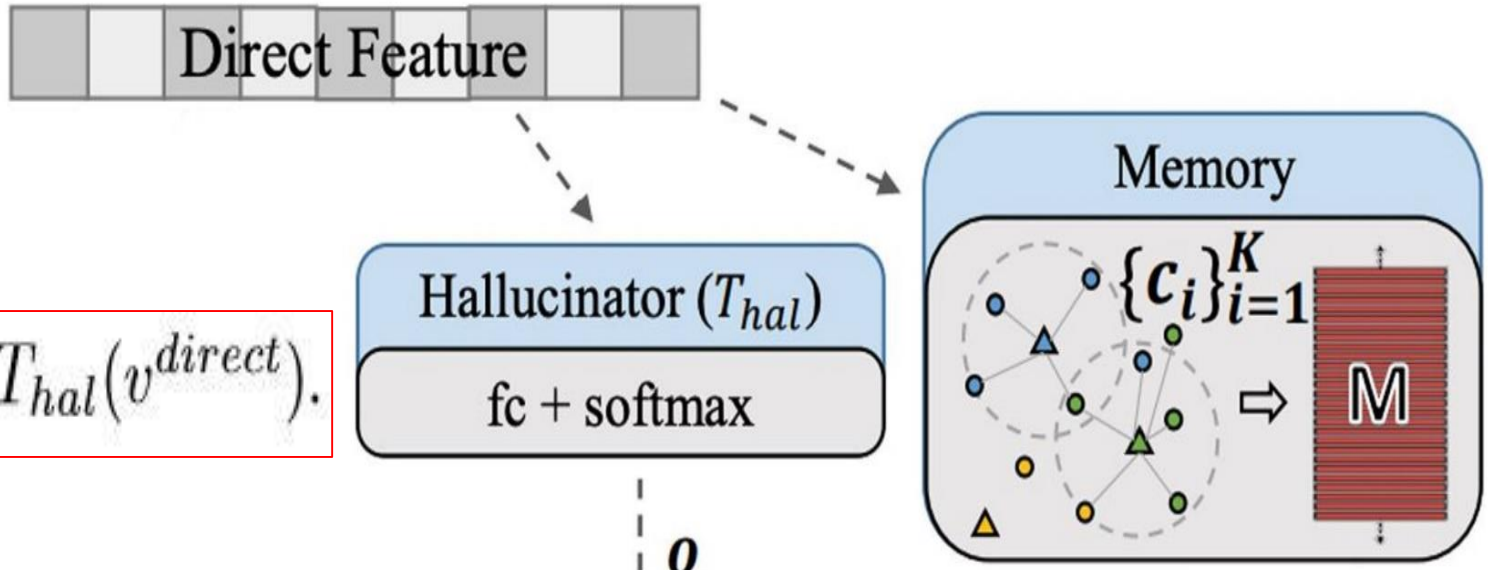
$$\mathcal{L} = \mathcal{L}_S + \lambda \mathcal{L}_C$$

$$= -\sum_{i=1}^{m} \log \frac{e^{W_{y_i}^T x_i + b_{y_i}}}{\sum_{j=1}^{n} e^{W_j^T x_i + b_j}} + \frac{\lambda}{2} \sum_{i=1}^{m} \|x_i - c_{y_i}\|_2^2$$

c_{y_i}: class center of sample x_i, same dimension as x_i

m: batch size

Knowledge Transfer to Tail

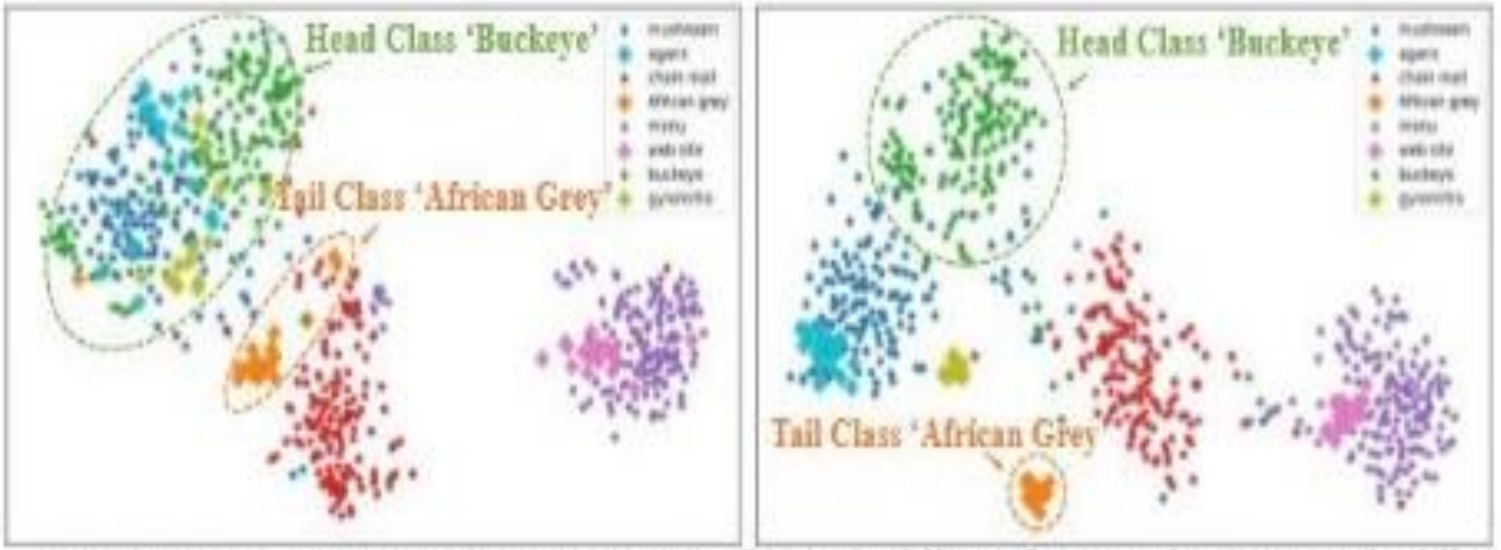


Meta Embedding

Different weightings for different classes

Dynamic Meta-Embedding

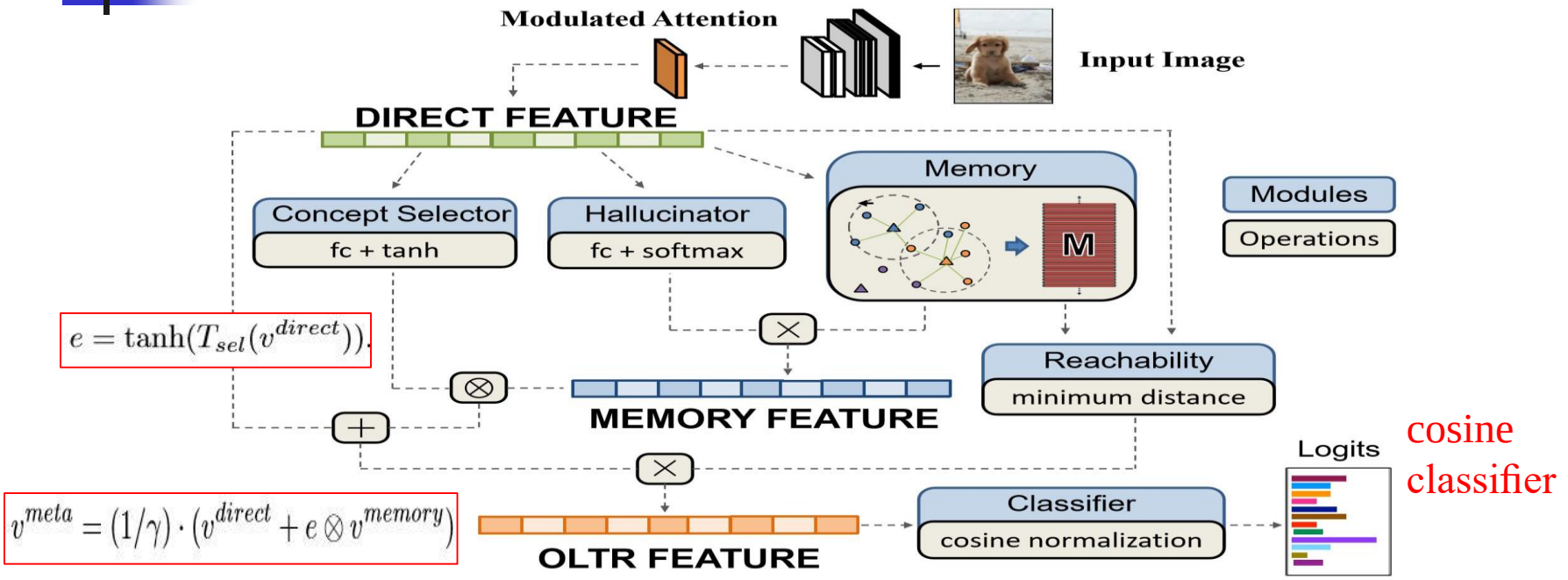
- Direct feature is often good enough for the data-rich head classes,
- Memory feature is more important for the data-poor tail classes.
- To adaptively select them in a soft manner, a lightweight network *Tsel(·)* is learned with a *tanh(·)* activation function:



(a) Embedding of Plain ResNet Model (b) Embedding of Dynamic Meta-Embedding

Reachability for Open-Set

$$\gamma := \text{reachability}(v^{direct}, M) = \min_i \|v^{direct} - c_i\|_2$$

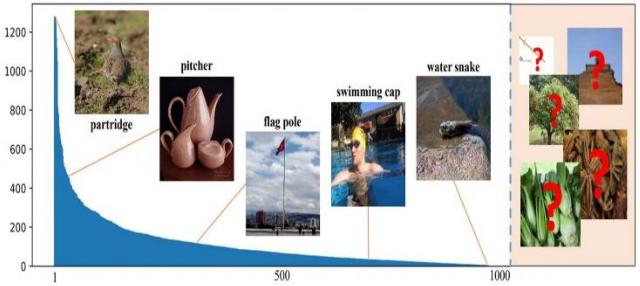


- When γ is small, the input likely belongs to a training class from which the centroids are derived,
- A large reachability, weight 1/γ (small and insignificant) is assigned to the resulting meta-embedding v^{meta}.

OLTR: Results

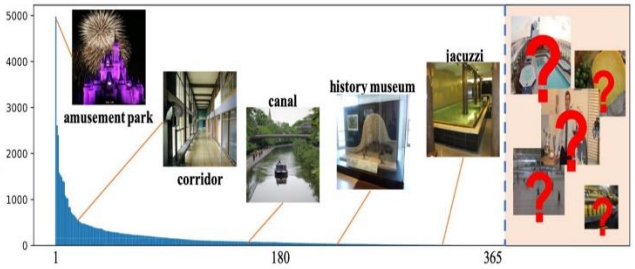
ImageNet-LT Benchmark

Absolute Performance Gain: ~20%



Places-LT Benchmark

Absolute Performance Gain: ~10%



$$F - measure = 2\frac{precision \cdot recall}{precision + recall}$$

Backbone Net ResNet-10 Methods	closed-set setting > 100 Many-shot	 $\leqslant 100$ & > 20 Medium-shot	 < 20 Few-shot	 Overall	open-set setting > 100 Many-shot	 $\leqslant 100$ & > 20 Medium-shot	 < 20 Few-shot	 F-measure
Plain Model [20]	40.9	10.7	0.4	20.9	40.1	10.4	0.4	0.295
Lifted Loss [37]	35.8	30.4	17.9	30.8	34.8	29.3	17.4	0.374
Focal Loss [29]	36.4	29.9	16	30.5	35.7	29.3	15.6	0.371
Range Loss [64]	35.8	30.3	17.6	30.7	34.7	29.4	17.2	0.373
+ OpenMax [3]	-	-	-	-	35.8	30.3	**17.6**	0.368
FSLwF [15]	40.9	22.1	15	28.4	40.8	21.7	14.5	0.347
Ours	**43.2**	**35.1**	**18.5**	**35.6**	**41.9**	**33.9**	17.4	**0.474**

(a) Top-1 classification accuracy on ImageNet-LT.

Backbone Net ResNet-152 Methods	closed-set setting > 100 Many-shot	 $\leqslant 100$ & > 20 Medium-shot	 < 20 Few-shot	 Overall	open-set setting > 100 Many-shot	 $\leqslant 100$ & > 20 Medium-shot	 < 20 Few-shot	 F-measure
Plain Model [20]	**45.9**	22.4	0.36	27.2	**45.9**	22.4	0.36	0.366
Lifted Loss [37]	41.1	35.4	24	35.2	41	35.2	23.8	0.459
Focal Loss [29]	41.1	34.8	22.4	34.6	41	34.8	22.3	0.453
Range Loss [64]	41.1	35.4	23.2	35.1	41	35.3	23.1	0.457
+ OpenMax [3]	-	-	-	-	41.1	35.4	23.2	0.458
FSLwF [15]	43.9	29.9	**29.5**	34.9	38.1	19.5	14.8	0.375
Ours	44.7	**37**	25.3	**35.9**	44.6	**36.8**	**25.2**	**0.464**

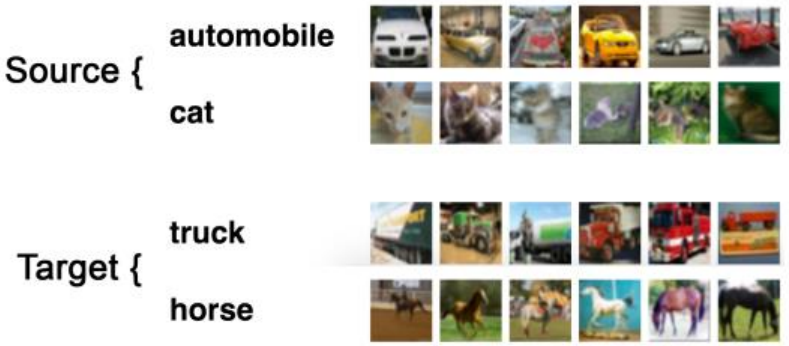
(b) Top-1 classification accuracy on Places-LT.

Transfer Learning of Different Domains or Tasks

- In practice, we may not have enough data/supervision in $\boldsymbol{X}$ to generalize well in the learning
- Transfer learning: to learn from data in other domains or even other tasks

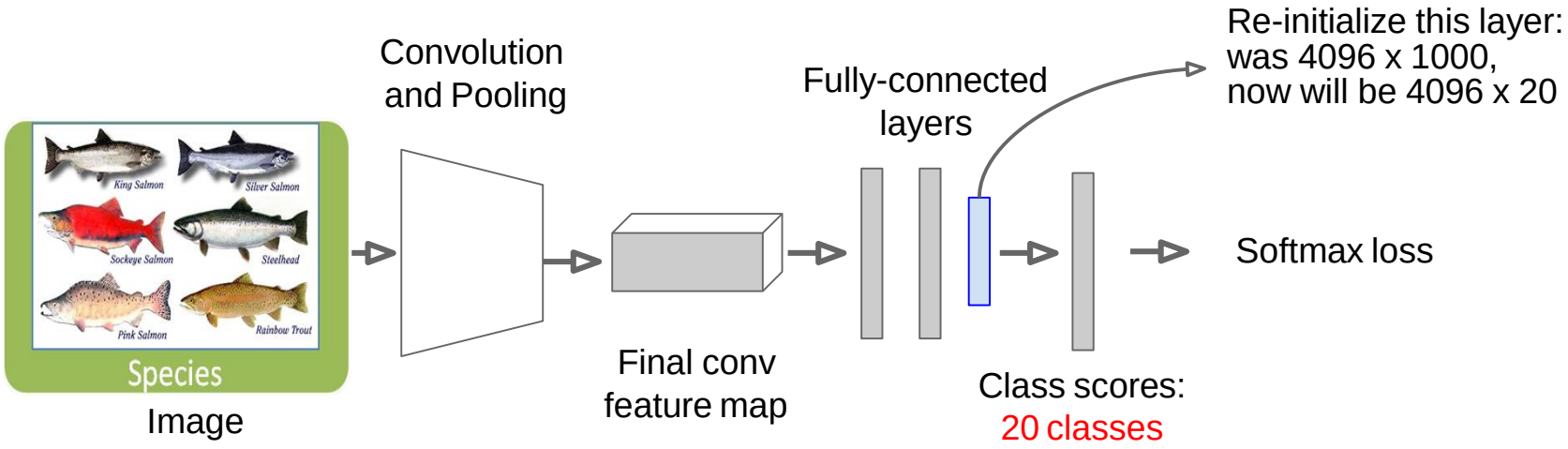
Different Domains (the same tasks)

Different Tasks



New Task Transfer Learning

- Transfer a pre-trained model (e.g., AlexNet, VGG) for a new task
- For example, instead of 1000-class ImageNet classification task, which has been a successful feature extraction architecture, we want transfer to a 20-class task
  - Throw away final fully-connected layer, reinitialize from scratch
  - Keep training model using the labelled data from the new task

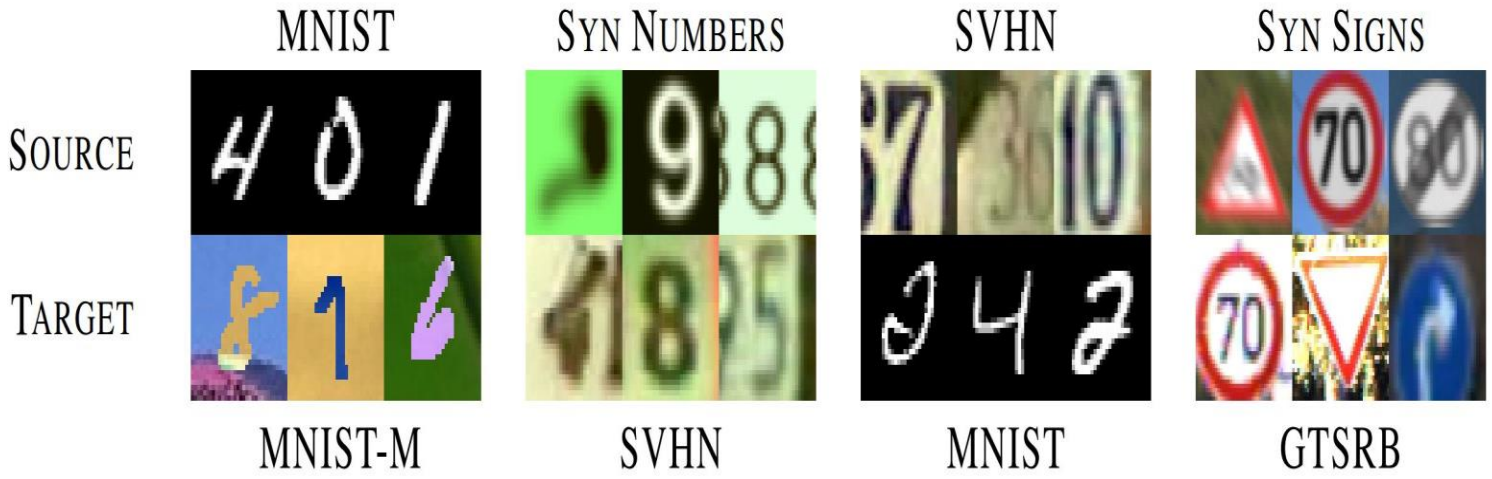


Domain Adaptation

- **Label shift**, changing of data distribution between source and target domain, can be solved by distribution agnostic LTR approaches
- **Domain shift**, consequences of changing conditions, i.e., background, location, pose changes, but the domain mismatch might be more severe when, e.g., the source and target domains contain images of different types, such as photos, NIR images, paintings or sketches.
- Transfer knowledge from the labeled dataset (**Source**) to the unlabeled dataset (**Target**) of the same task

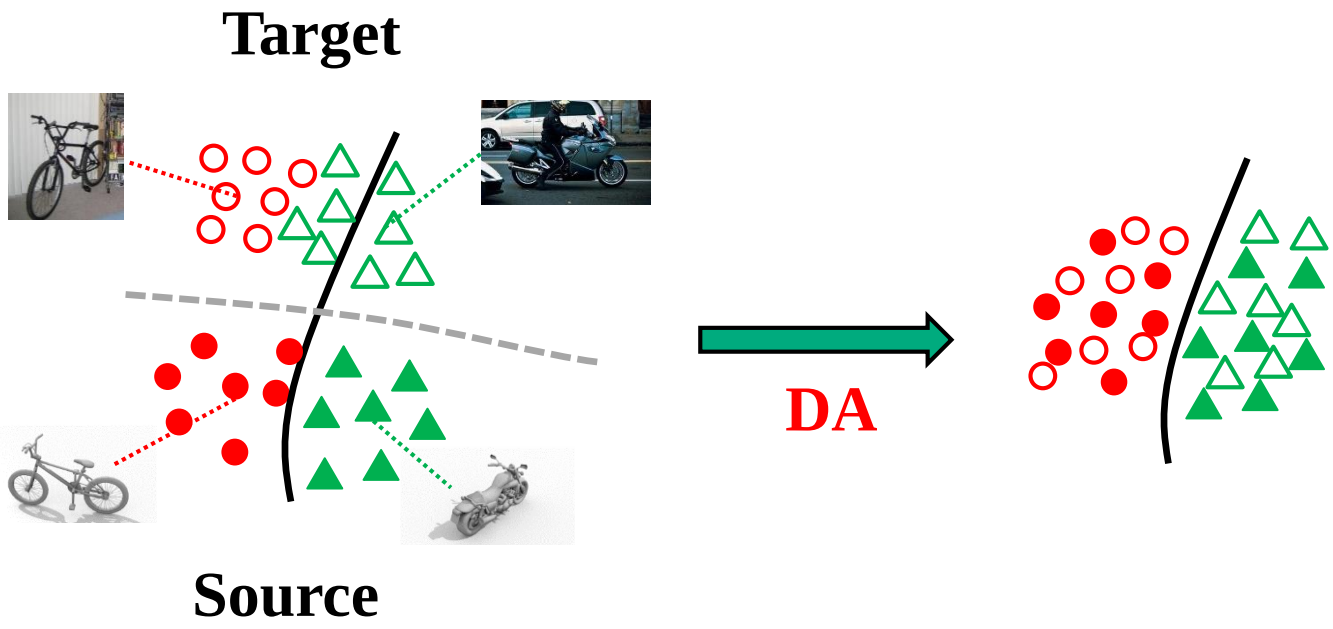
	Source domain	Target domain
Unsupervised DA		Unlabeled data
Semi-supervised DA	Labeled data are available	Unlabeled data+ Small amount of labeled data
Supervised DA		Labeled data are available

Domain MisMatch

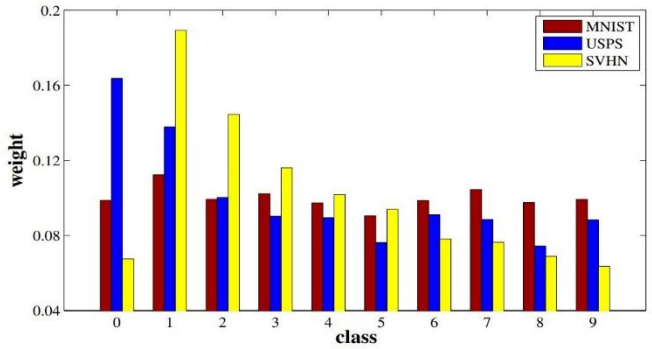


Domain Adaptation (DA)

- Objective: align target domain with source domain



Need for Domain Adaptation



MNIST

Source	Target
MNIST	
98.2%	

USPS

Source	Target
USPS	
98.4%	

Domain Shift

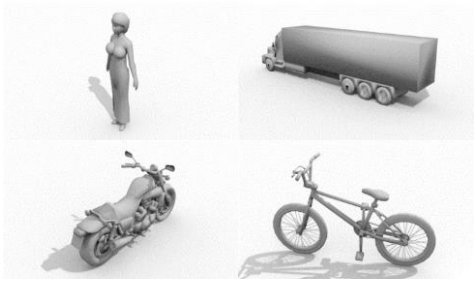
Source	Target
MNIST	USPS
75.2%	

Source	Target
USPS	MNIST
67.1%	

J. J. Hull. A database for handwritten text recognition research. **PAMI**, 16(5), **1994**.

Unsupervised Domain Adaptation (UDA)

Source domain: (synthetic images(w labels)



Target domain : real images (no labels)

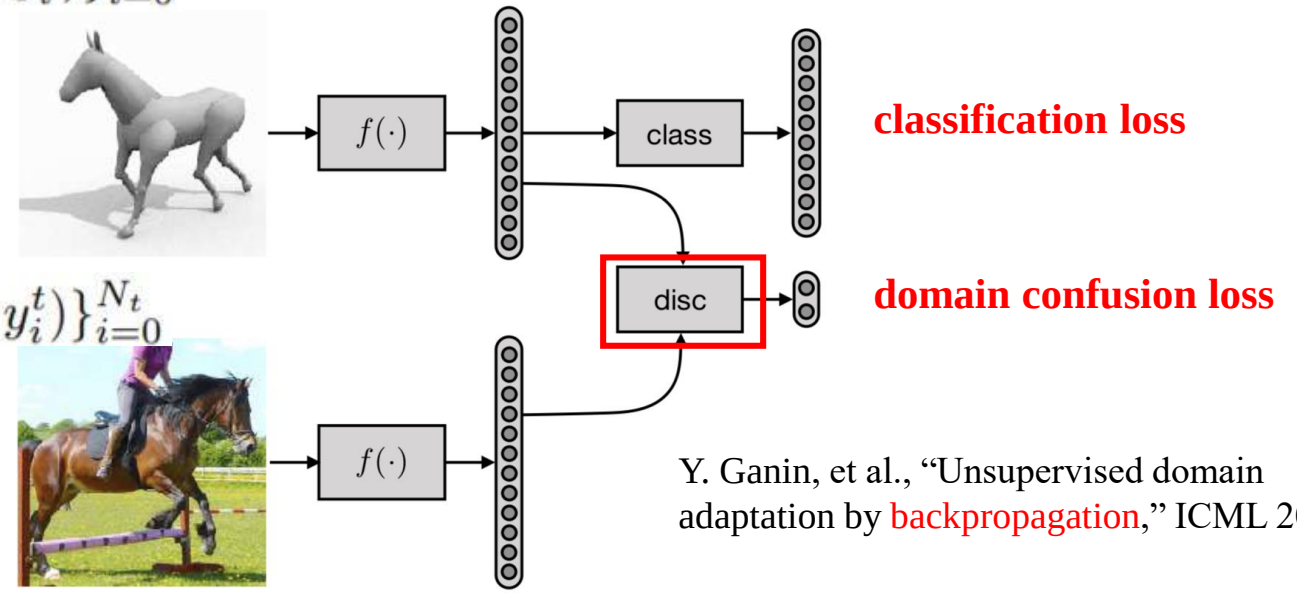
(Domain Adaptation)

Adversarial-based Domain Adaptation

- Extended from DA by BP, which is similar to GAN
- A network tries to extract features that remain the same across the domains.

$$\mathbf{X}_s = \{(x_i^s, y_i^s)\}_{i=0}^{N_s}$$

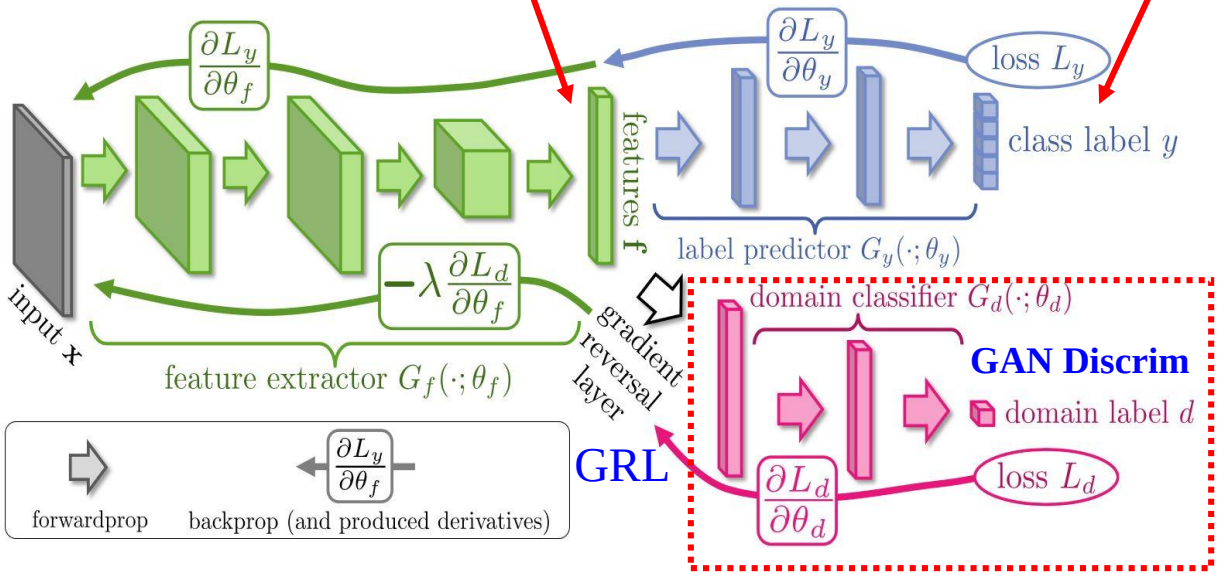
$$\mathbf{X}_t = \{(x_i^t, y_i^t)\}_{i=0}^{N_t}$$



Y. Ganin, et al., "Unsupervised domain adaptation by backpropagation," ICML 2015.

UDA by BackPropagation

- A deep feature extractor (green) and a deep label predictor (blue).
- Adding a domain classifier (red, like a discriminator) to ensure that the feature appearance over the two domains are made similar (as indistinguishable as possible for the domain classifier), thus resulting in the domain-invariant features.



$$\theta_f \leftarrow \theta_f - \mu\left(\frac{\partial L_y^i}{\partial \theta_f} - \lambda\frac{\partial L_d^i}{\partial \theta_f}\right)$$

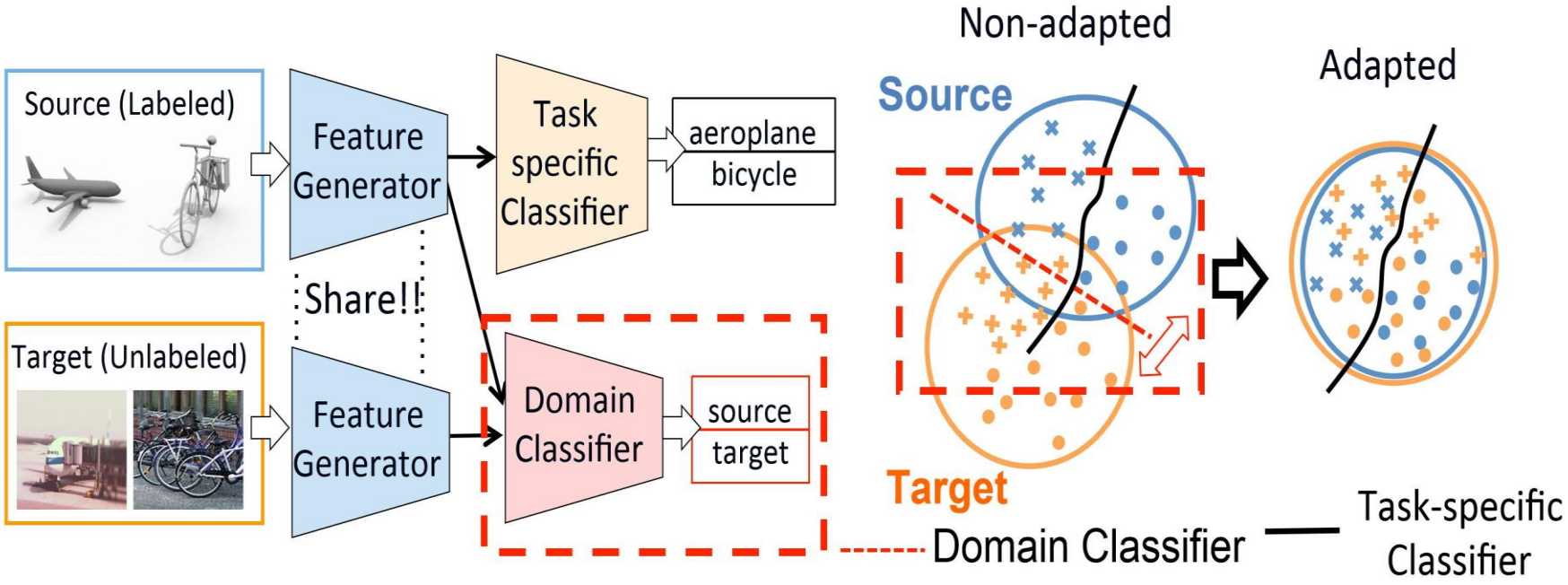
$$\theta_y \leftarrow \theta_y - \mu\frac{\partial L_y^i}{\partial \theta_y}$$

$$\theta_d \leftarrow \theta_d - \mu\frac{\partial L_d^i}{\partial \theta_d}$$

Y. Ganin, et al., "Unsupervised domain adaptation by backpropagation," ICML 2015.

UDA by BackPropagation

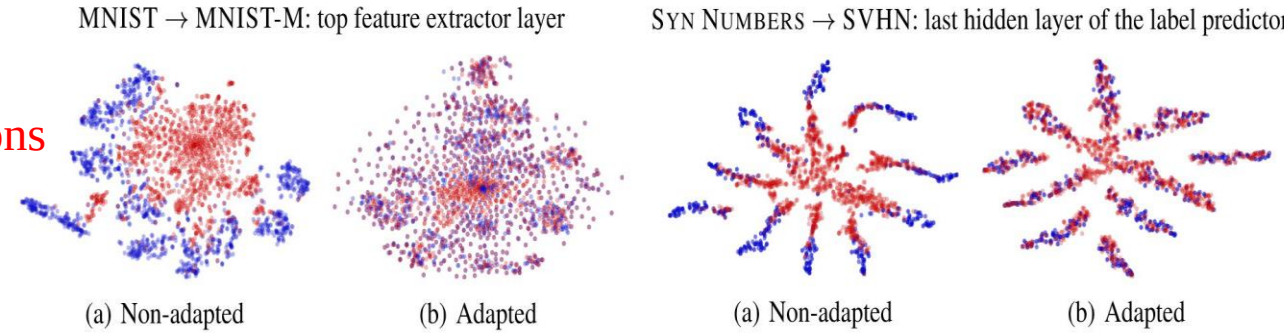
- Domain Classifier: Discriminate the domain of features
- Feature Generator: Deceive the domain classifier



Performance of UDA by BP

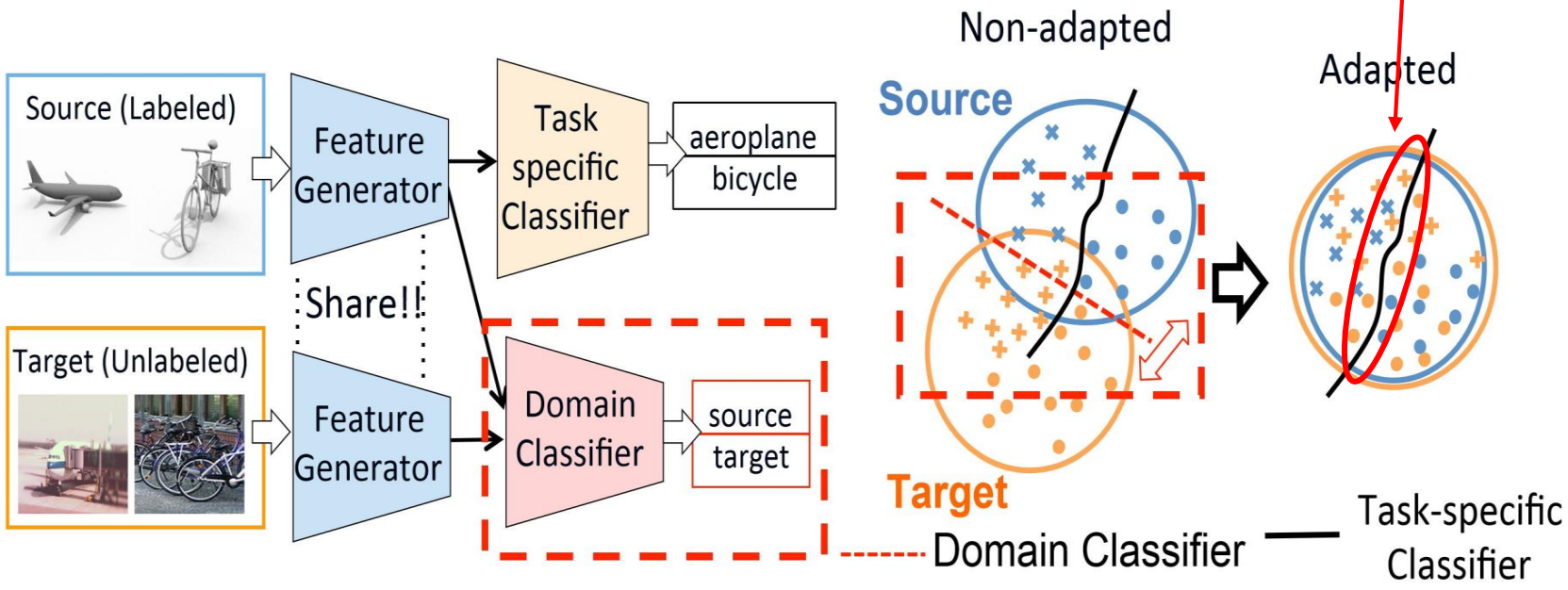
METHOD	SOURCE	MNIST	SYN NUMBERS	SVHN	SYN SIGNS
	TARGET	MNIST-M	SVHN	MNIST	GTSRB
SOURCE ONLY		.5749	.8665	.5919	.7400
SA (FERNANDO ET AL., 2013)		.6078 (7.9%)	.8672 (1.3%)	.6157 (5.9%)	.7635 (9.1%)
PROPOSED APPROACH		**.8149** (57.9%)	**.9048** (66.1%)	**.7107** (29.3%)	**.8866** (56.7%)
TRAIN ON TARGET		.9891	.9244	.9951	.9987

Feature Distributions



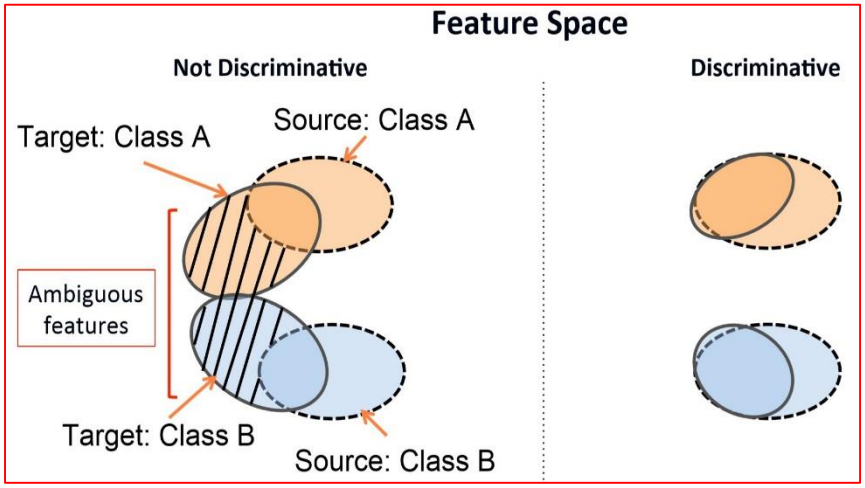
An Issue of UDA by Backpropagation

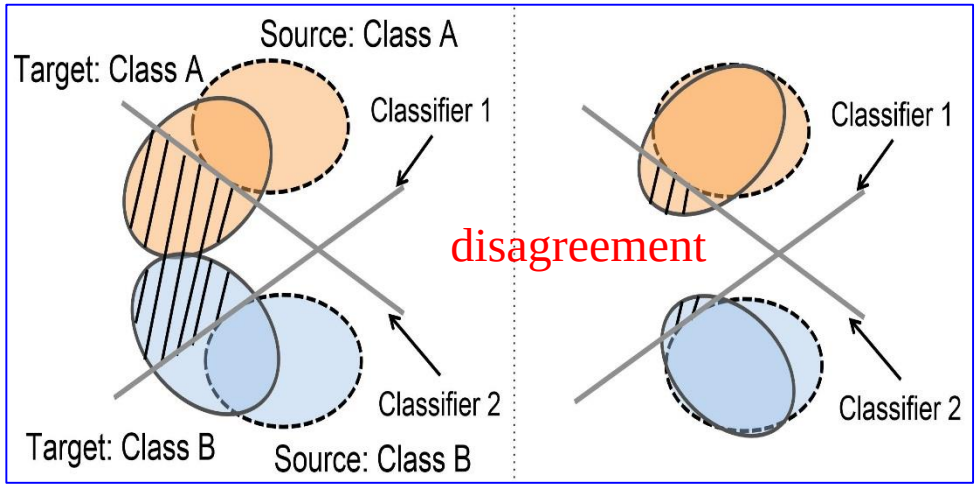
- Target features can be near a task-specific classifier's boundary, easily misclassified after alignment



Ambiguous Features and Classification Discrepancy

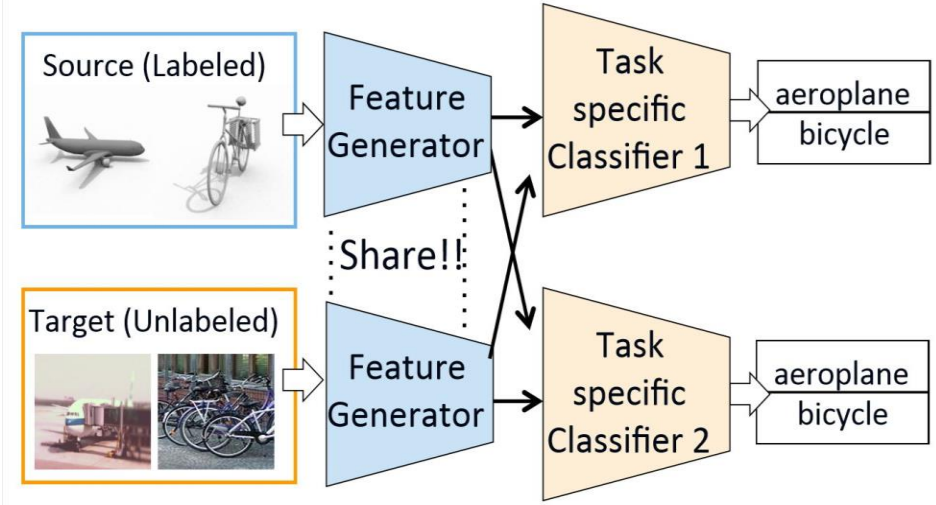
- Target samples far from source ones (ambiguous features) are likely to be misclassified after alignment
- Two distinct classifiers that classify source features correctly
  - Agree: target features are near source (discriminative)
  - Disagree: target features are ambiguous (non-discriminative)

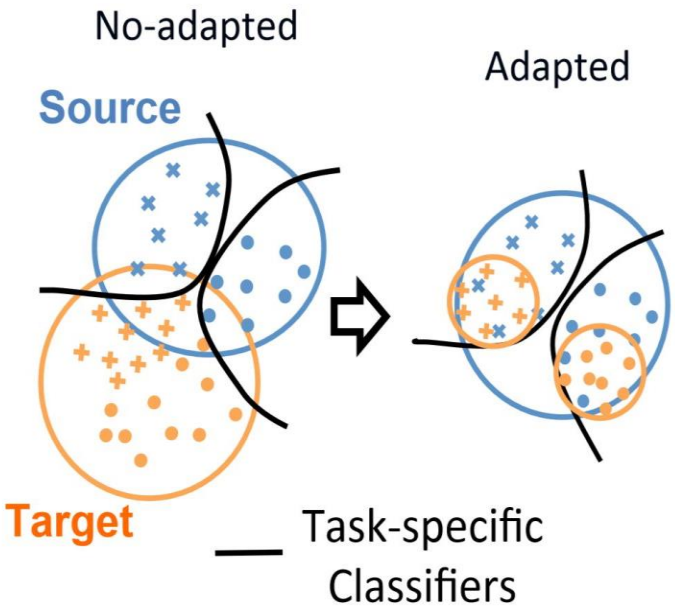




Task Specific Classifier

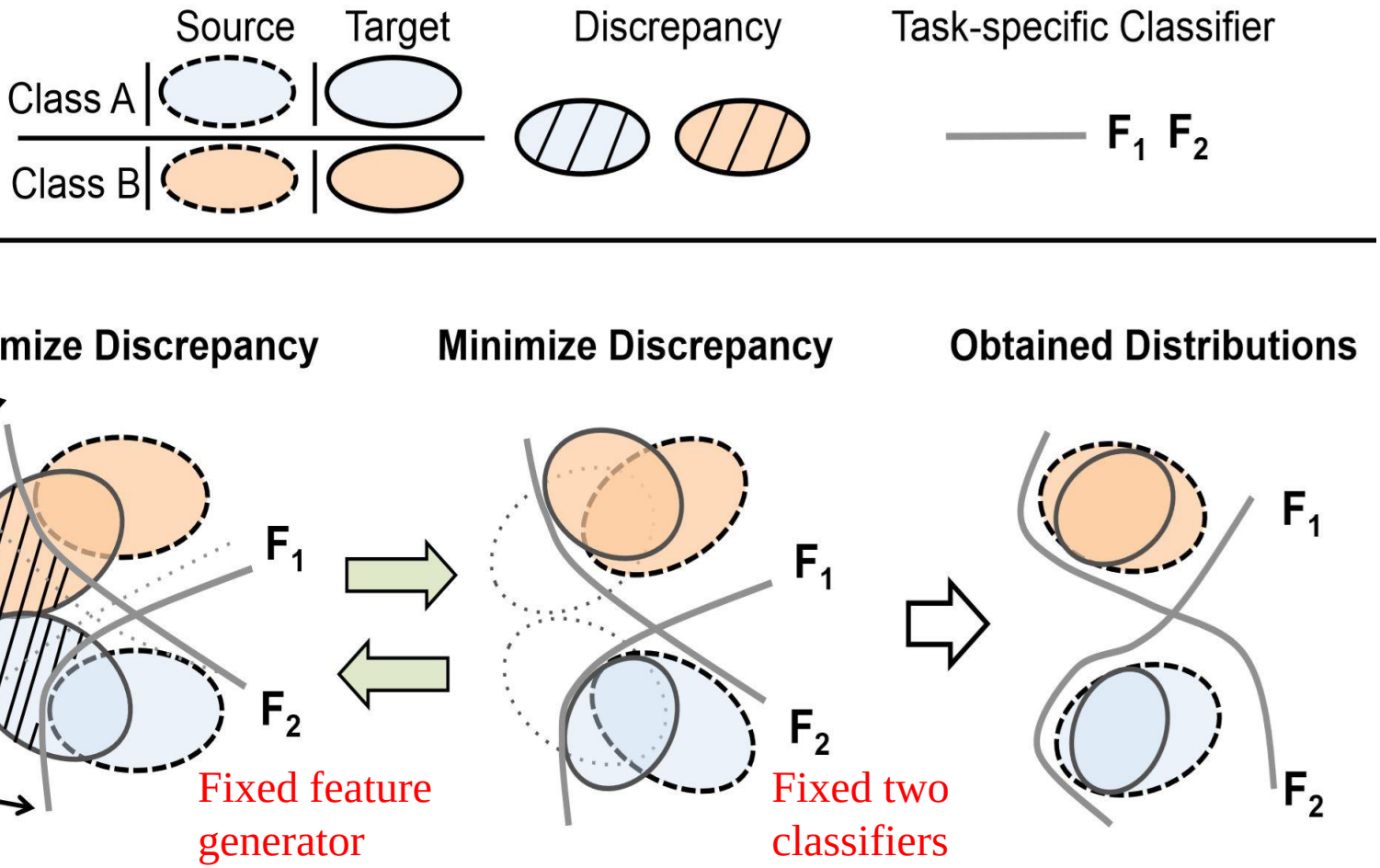
- Task-specific classifier based distribution alignment
  - Relationship between decision boundary and target features
  - Discriminative features



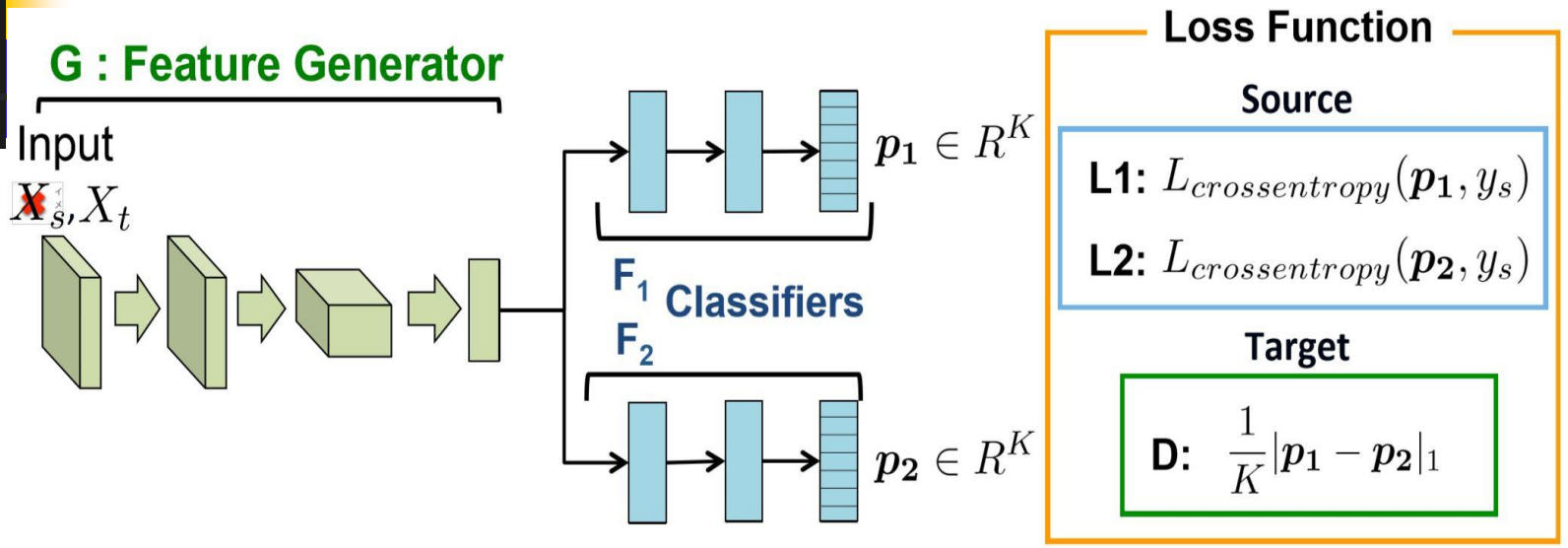


Kuniaki Saito, et al., "Maximum Classifier Discrepancy for Unsupervised Domain Adaptation," CVPR 2018

Maximum Classifier Discrepancy (MCD)

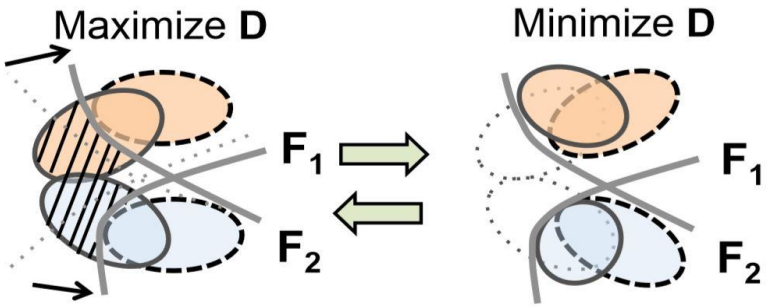


MCD Training Procedure



Training per one mini-batch (sample images from both domains)

1, Fix G and update F1, F2 to decrease **L1+L2-D (maximize the discrepancy)**
2, Update G,F1,F2 to decrease **L1+L2 (minimize error on source)**
3, Fix F1,F2 and update G to decrease **D (minimize the discrepancy)**



UDA for Image Classification

METHOD	SVHN to MNIST	SYNSIG to GTSRB	MNIST to USPS	MNIST* to USPS*	USPS to MNIST
Source Only	67.1	85.1	76.7	79.4	63.4
MMD [Long et al., ICML 2015]	71.1	91.1	-	81.1	-
DANN [Ganin et al., ICML 2015]	71.1	88.7	77.1±1.8	85.1	73.0±0.2
DSN [Bousmalis et al., NIPS 2016]	82.7	93.1	91.3	-	-
ADDA [Tzeng et al., CVPR 2017]	76.0±1.8	-	89.4±0.2	-	90.1±0.8
Ours	**96.2**±0.4	**94.4**±0.3	**94.2**±0.7	**96.5**±0.3	**94.1**±0.3