Unsupervised Deep Learning by

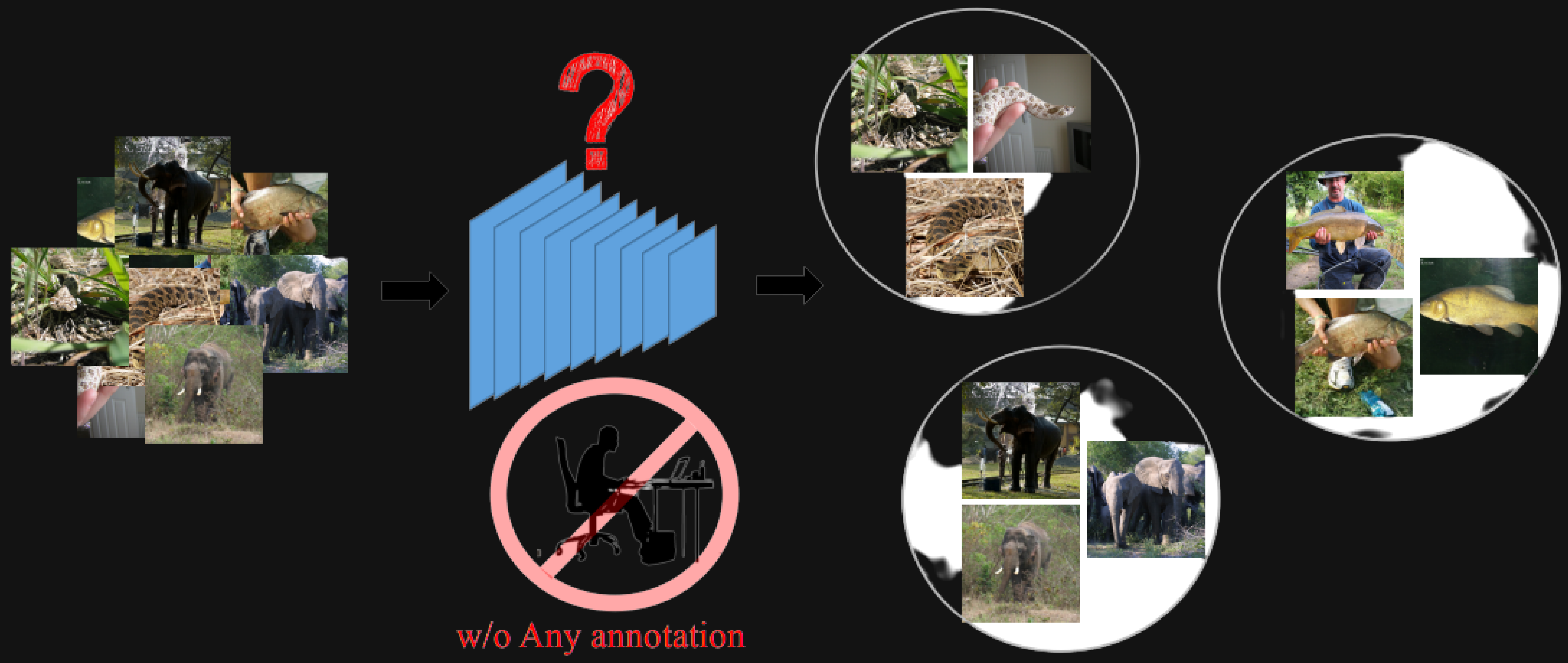
Neighbourhood Discovery

Jiabo Huang, Qi Dong, Shaogang Gong, Xiatian Zhu

Summary

- Introduction
- Related Work
- Anchored Neighborhood Discovery
- Experiments

Introduction



Related Work

Clustering Analysis

Jointly optimising clustering analysis and representation learning

Sample Specificity (Instance) Learning

Considering every single sample as an independent class

Self-supervised Learning

Exploiting information intrinsically available in data

Generative Models

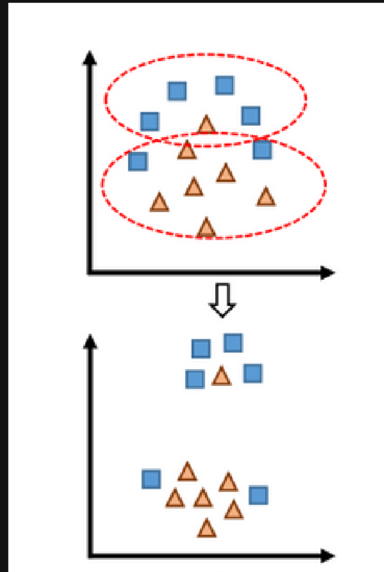
Learning the true data distribution of training set

Related Work

Clustering Analysis

Jointly optimising clustering analysis and representation learning

- Inconsistent inter-sample relations
- error-propagation

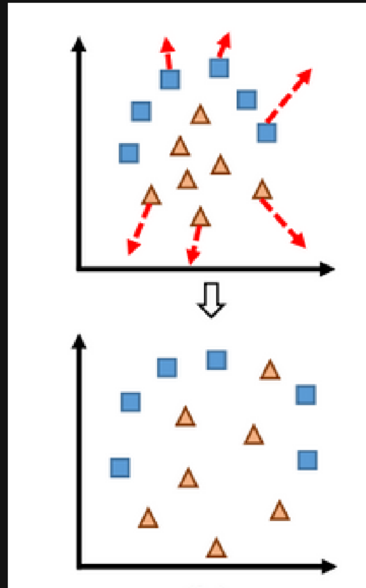


(a)

Sample Specificity (Instance) Learning

Considering every single sample as an independent class

- Leaving out the correlation between samples
- poor discriminative ability



(b)

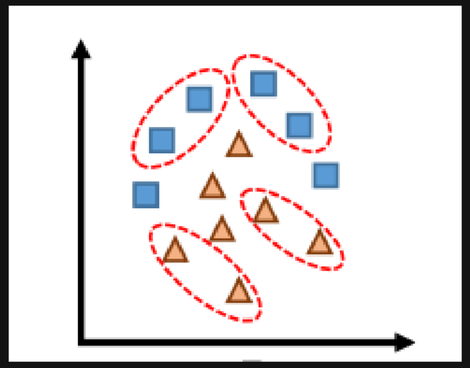
Clustering Analysis

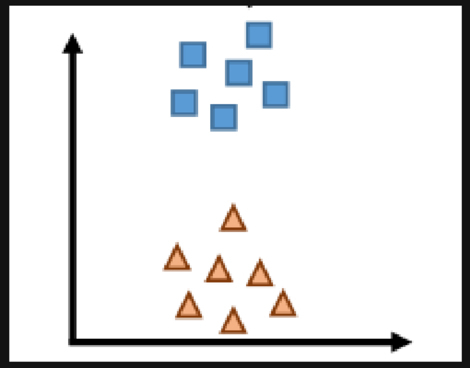
error-propagation

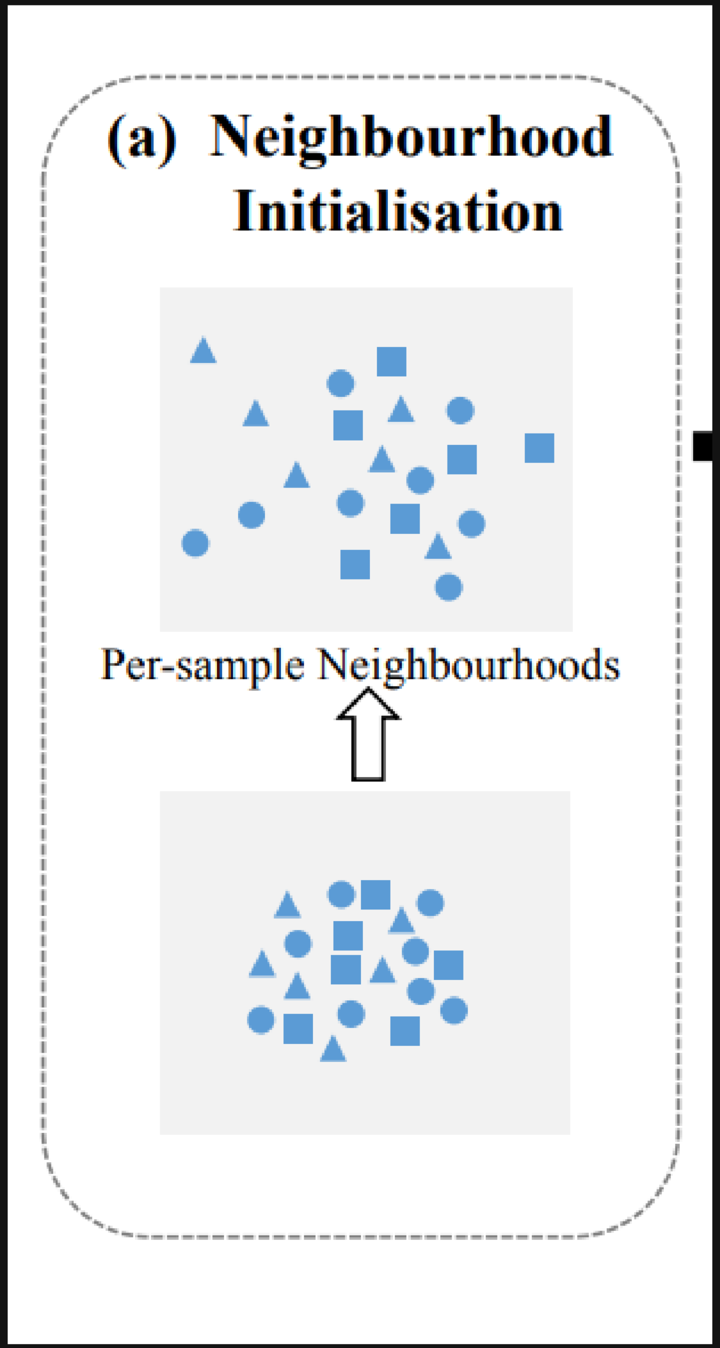
Sample Specificity

poor discriminative ability

Measuring the consistency of inter-sample relations and learn from those in high-confidence only







HOW TO IDENTIFY NEIGHBOURHOODS

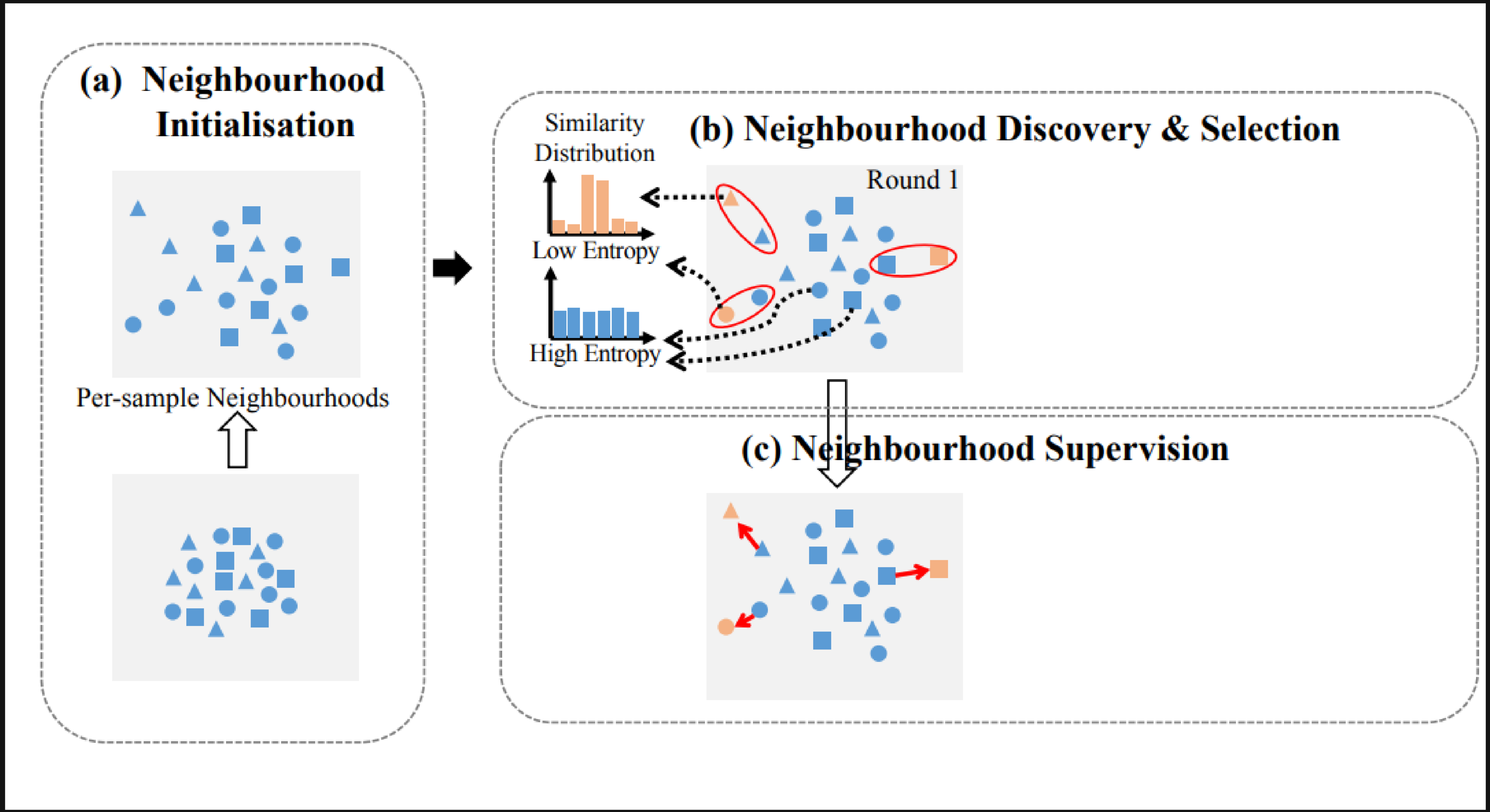
Nk(x) = {xi | s(xi , x) is top-k in X} U {x}

Anchor Neighbourhoods (AN)

NEIGHBOURHOOD INITIALISATION

Each individual sample represents a distinct AN. The instance loss to commence the model learning:

$$\mathcal{L}_{\text{init}} = -\sum_{i=1}^{n_{\text{bs}}} \log(p_{i,i}), \;\; p_{i,j} = \frac{\exp(\boldsymbol{x}_i^\top \boldsymbol{x}_j/\tau)}{\sum_{k=1}^{N} \exp(\boldsymbol{x}_i^\top \boldsymbol{x}_k/\tau)}$$



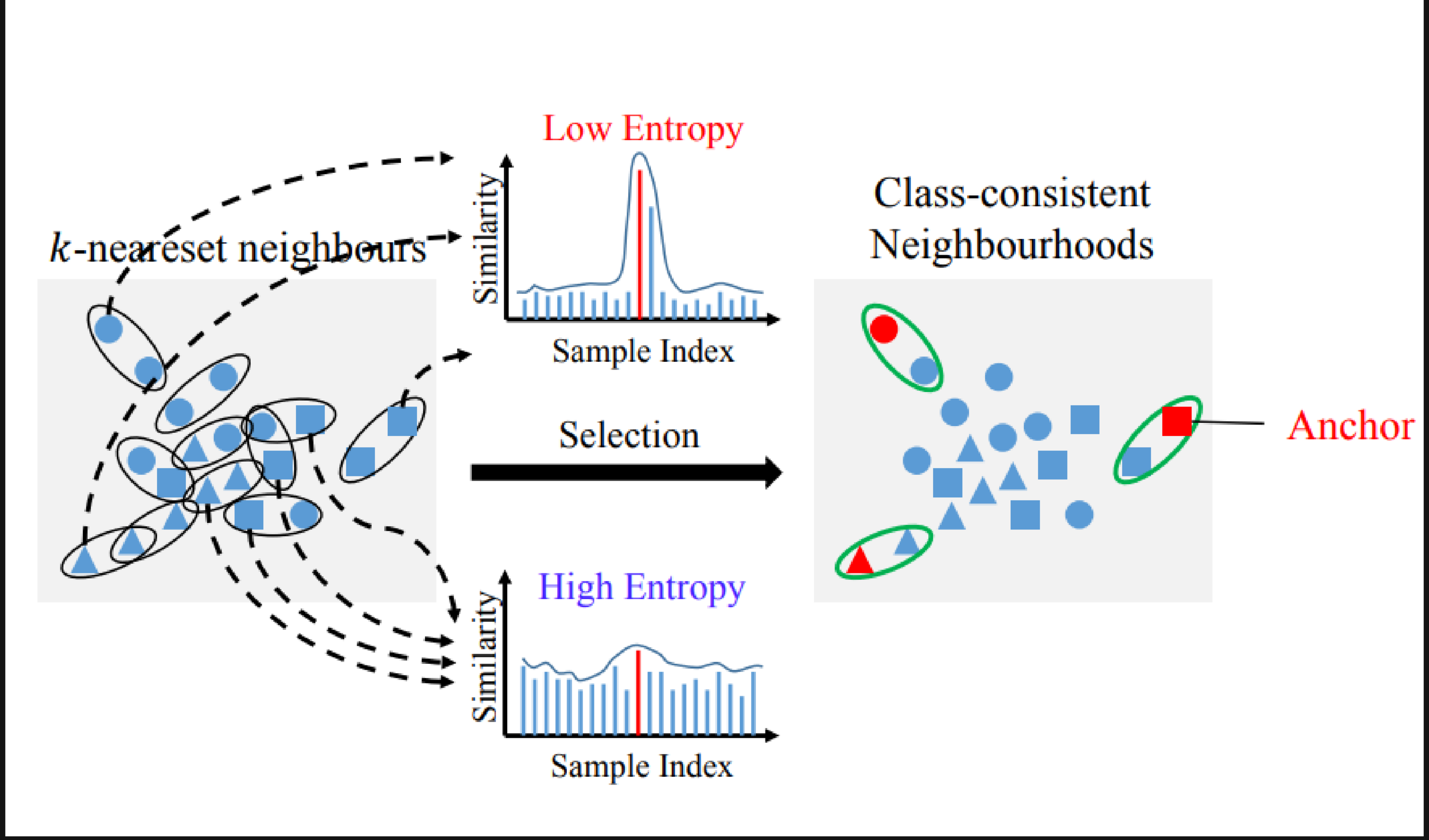
ENTROPY

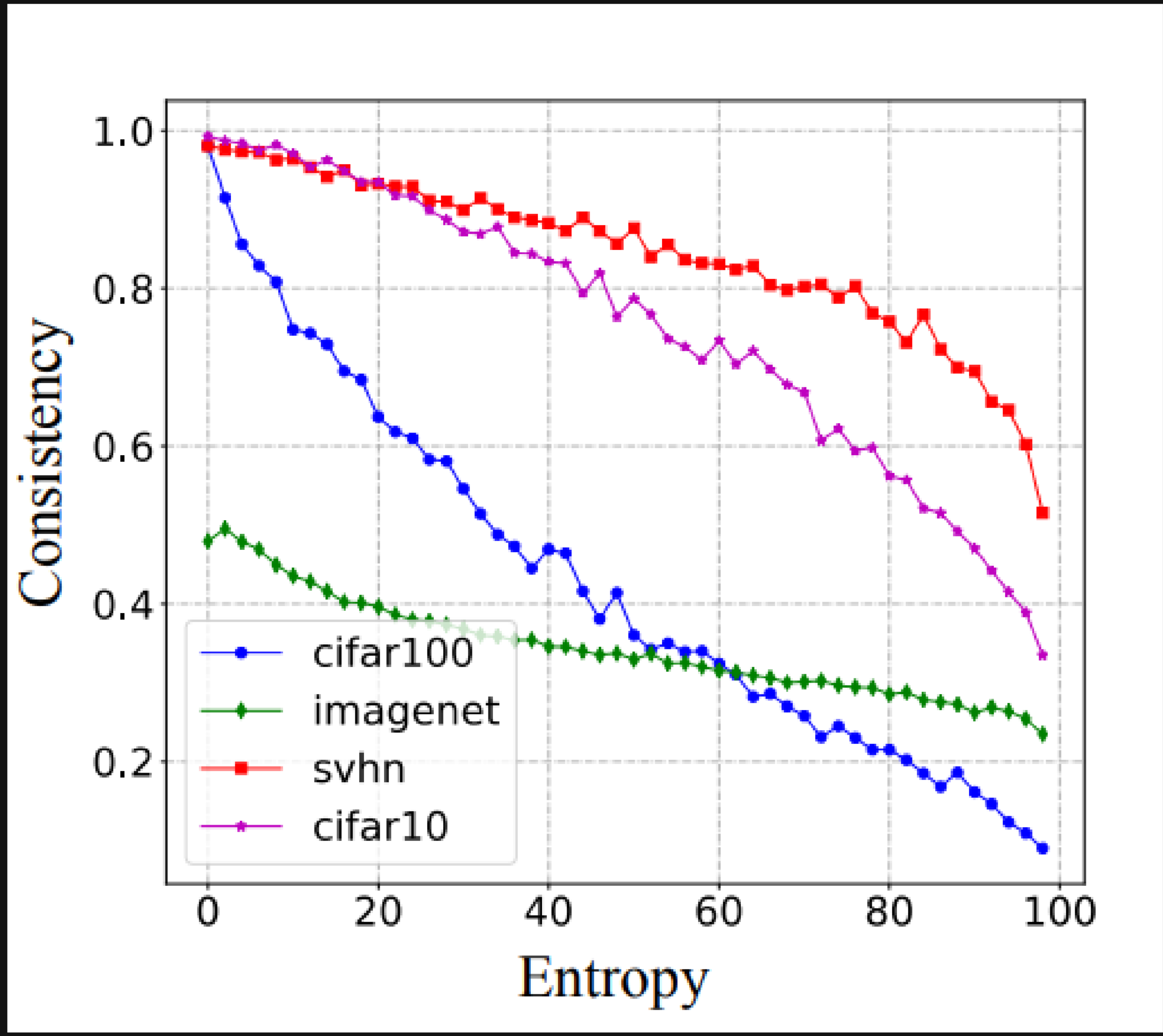
$$H(\boldsymbol{x}_i) = -\sum_{j=1}^{N} p_{i,j} \log(p_{i,j}).$$

$$\mathcal{L}_{\mathrm{AN}} = -\sum_{i=1}^{n_{\mathrm{bs}}} \log\Big(\sum_{j \in \mathcal{N}_k(\boldsymbol{x}_i)} p_{i,j}\Big)$$

$$S = \frac{r}{R} * 100\%$$

For the r-th round (among a total of R rounds), we select the top-S* of ANs according to their corresponding entropy for model learning by the proposed neighbourhood supervision loss LAN*.





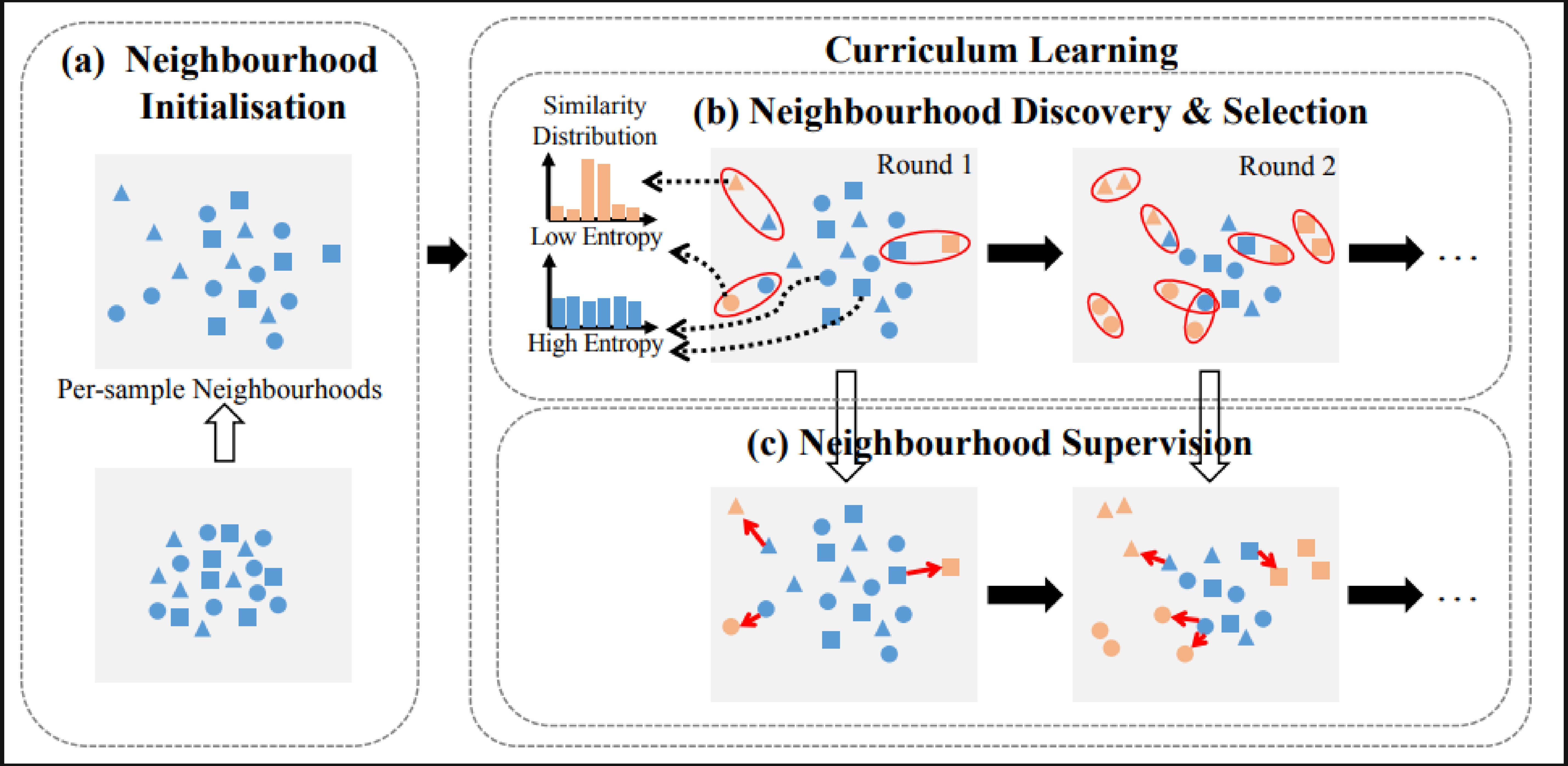
ENTROPY

$$H(\boldsymbol{x}_i) = -\sum_{j=1}^{N} p_{i,j} \log(p_{i,j}).$$

$$\mathcal{L}_{\text{AN}} = -\sum_{i=1}^{n_{\text{bs}}} \log\Big(\sum_{j \in \mathcal{N}_k(\boldsymbol{x}_i)} p_{i,j}\Big)$$

$$S = \frac{r}{R} * 100\%$$

For the r-th round (among a total of R rounds), we select the top-S* of ANs according to their corresponding entropy for model learning by the proposed neighbourhood supervision loss LAN*.



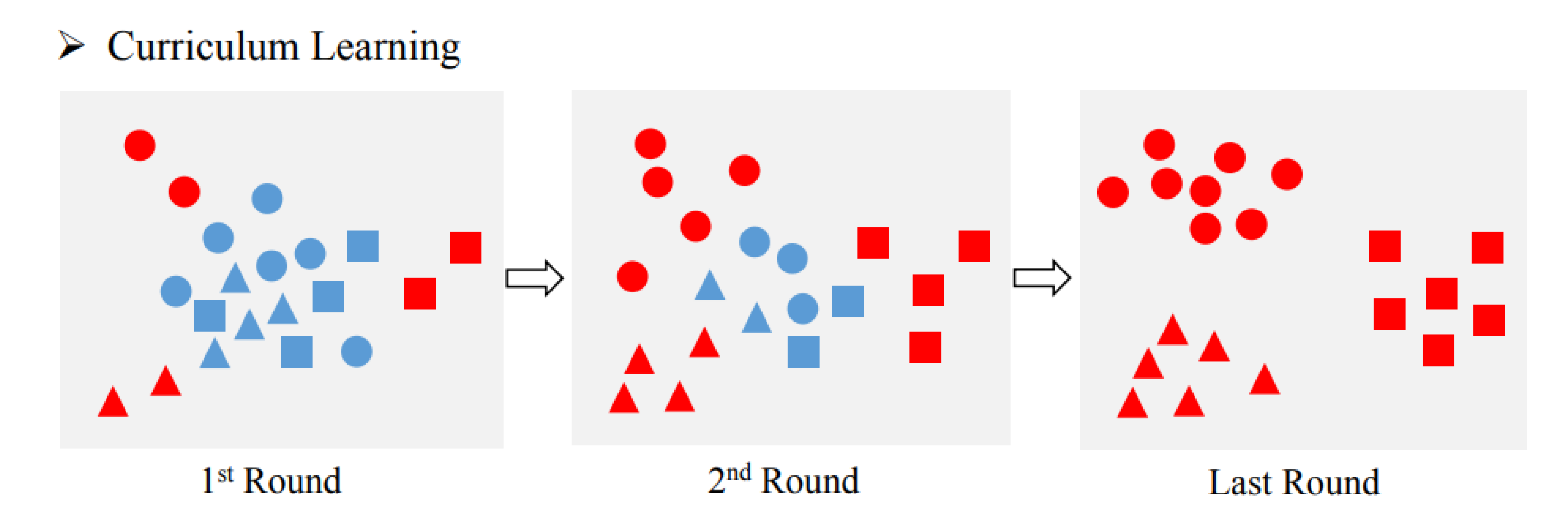
ENTROPY

$$H(\boldsymbol{x}_i) = -\sum_{j=1}^{N} p_{i,j} \log(p_{i,j}).$$

$$\mathcal{L}_{\mathrm{AN}} = -\sum_{i=1}^{n_{\mathrm{bs}}} \log\Big(\sum_{j \in \mathcal{N}_k(\boldsymbol{x}_i)} p_{i,j}\Big)$$

$$S = \frac{r}{R} * 100\%$$

For the r-th round (among a total of R rounds), we select the top-S* of ANs according to their corresponding entropy for model learning by the proposed neighbourhood supervision loss LAN*.



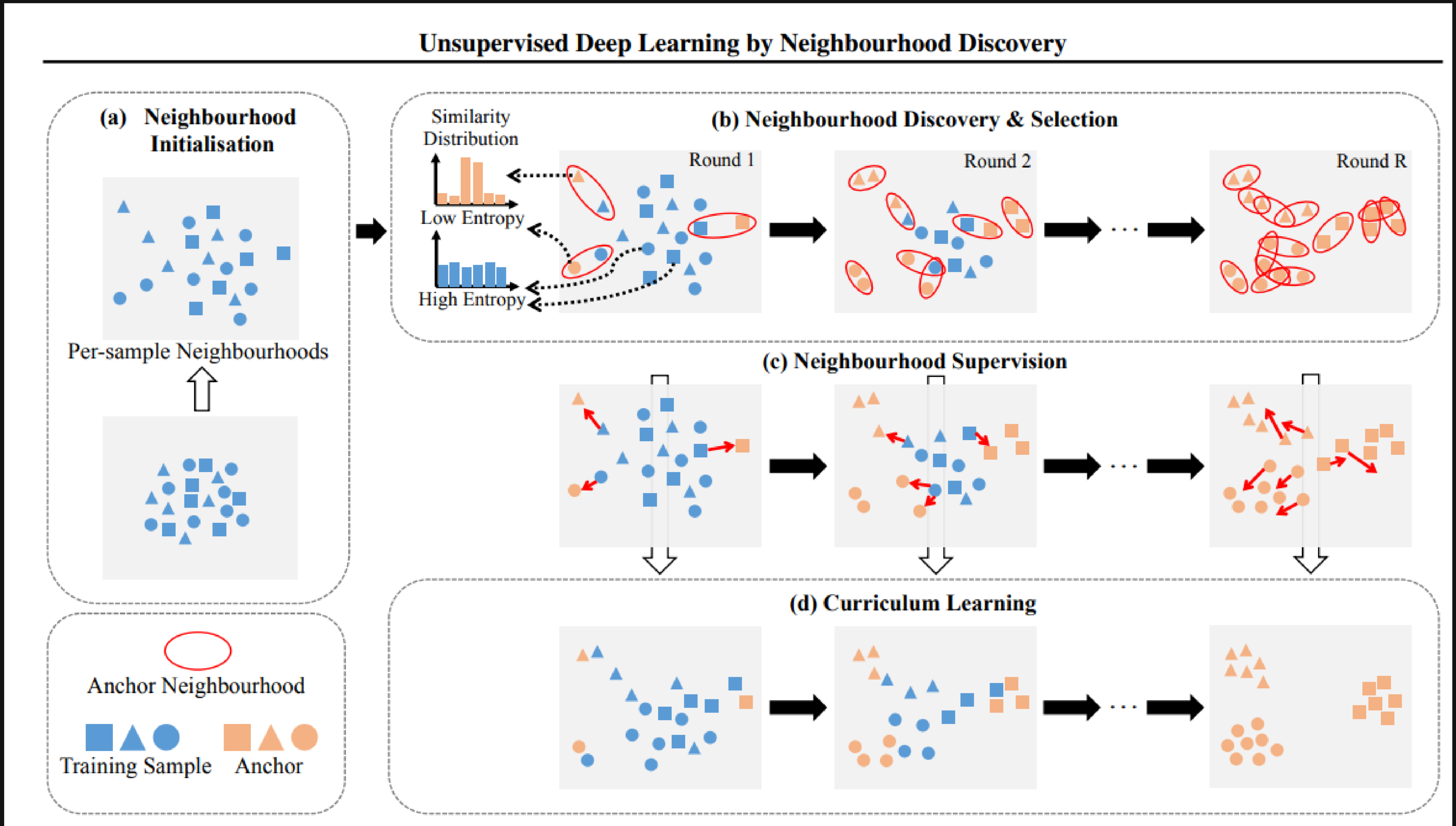
ENTROPY

$$H(\boldsymbol{x}_i) = -\sum_{j=1}^{N} p_{i,j}\log(p_{i,j}).$$

$$\mathcal{L}_{\mathrm{AN}} = -\sum_{i=1}^{n_{\mathrm{bs}}}\log\Big(\sum_{j\in\mathcal{N}_k(\boldsymbol{x}_i)} p_{i,j}\Big)$$

$$S = \frac{r}{R} * 100\%$$

For the r-th round (among a total of R rounds), we select the top-S* of ANs according to their corresponding entropy for model learning by the proposed neighbourhood supervision loss LAN*.



Objective Loss Function. With the progressive neighbourhood discovery as above, we obtain the model objective loss function for the r-th round as:

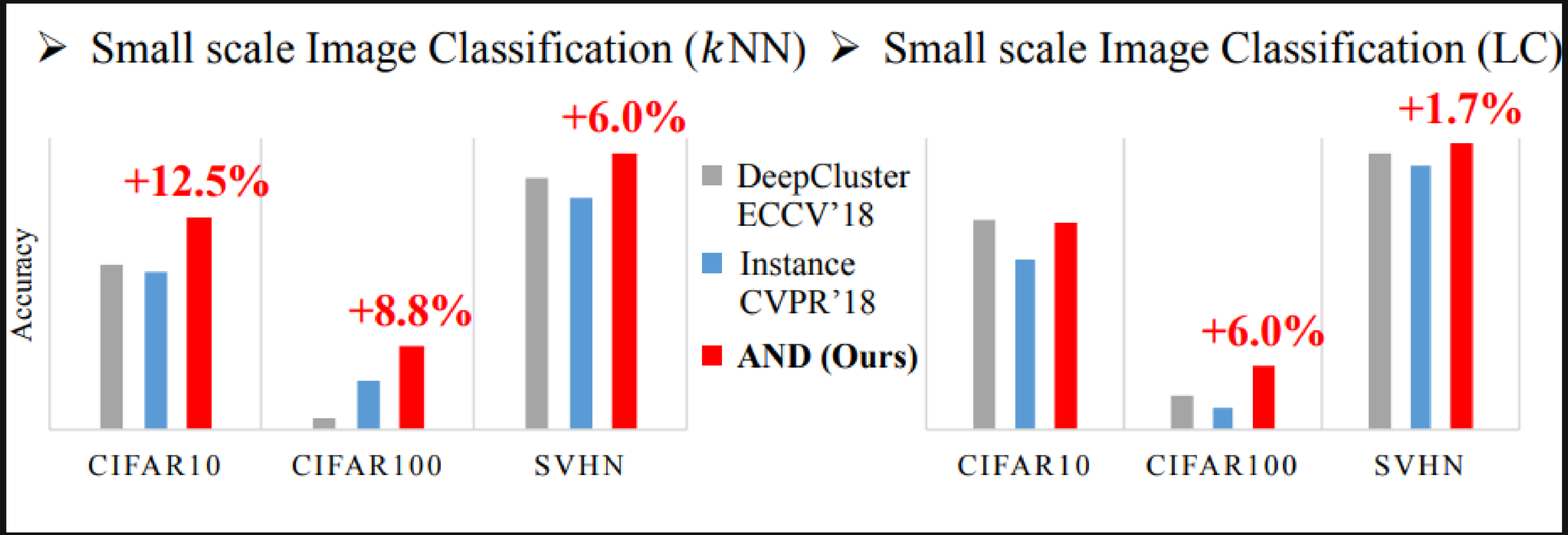
$$\mathcal{L}^r = -\sum_{i \in B^r_{\text{inst}}} \log(p_{i,i}) - \sum_{i \in B^r_{\text{AN}}} \log\Big(\sum_{j \in \mathcal{N}_k(\boldsymbol{x}_i)} p_{i,j}\Big) \quad (7)$$

where B^r_{inst} and B^r_{AN} denote the set of instances and the set of ANs in a mini-batch at the r-th round, respectively.

The proposed loss function (Eq (7)) is differentiable therefore enabling the stochastic gradient descent algorithm for model training.

Comparisons to the State-of-the-Art Methods

Small scale evaluation

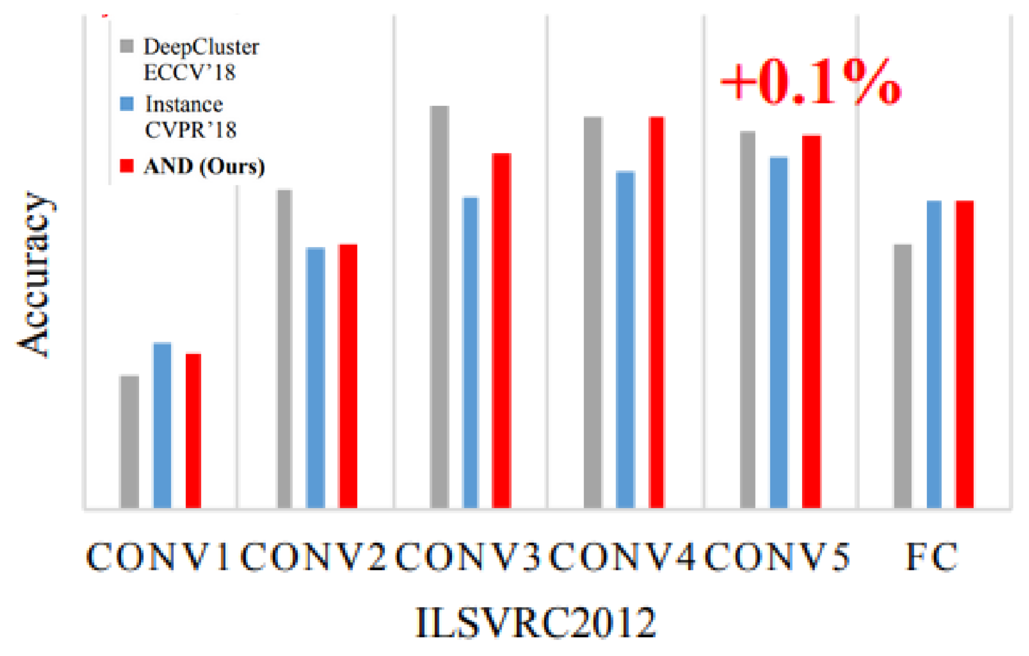


Dataset	CIFAR10	CIFAR100	SVHN
Classifier/Feature	Weighted kNN / FC		
Split-Brain*	11.7	1.3	19.7
Counting*	41.7	15.9	43.4
DeepCluster	62.3	22.7	84.9
Instance	60.3	32.7	79.8
AND	**74.8**	**41.5**	**90.9**
Supervised	91.9	69.7	96.5
Classifier/Feature	Linear Classifier / conv5		
Split-Brain*	67.1	39.0	77.3
Counting*	50.9	18.2	63.4
DeepCluster	**77.9**	41.9	92.0
Instance	70.1	39.4	89.3
AND	77.6	**47.9**	**93.7**
Supervised	91.8	71.0	96.1

Comparisons to the State-of-the-Art Methods

Large scale evaluation

➤ Large scale Image Classification

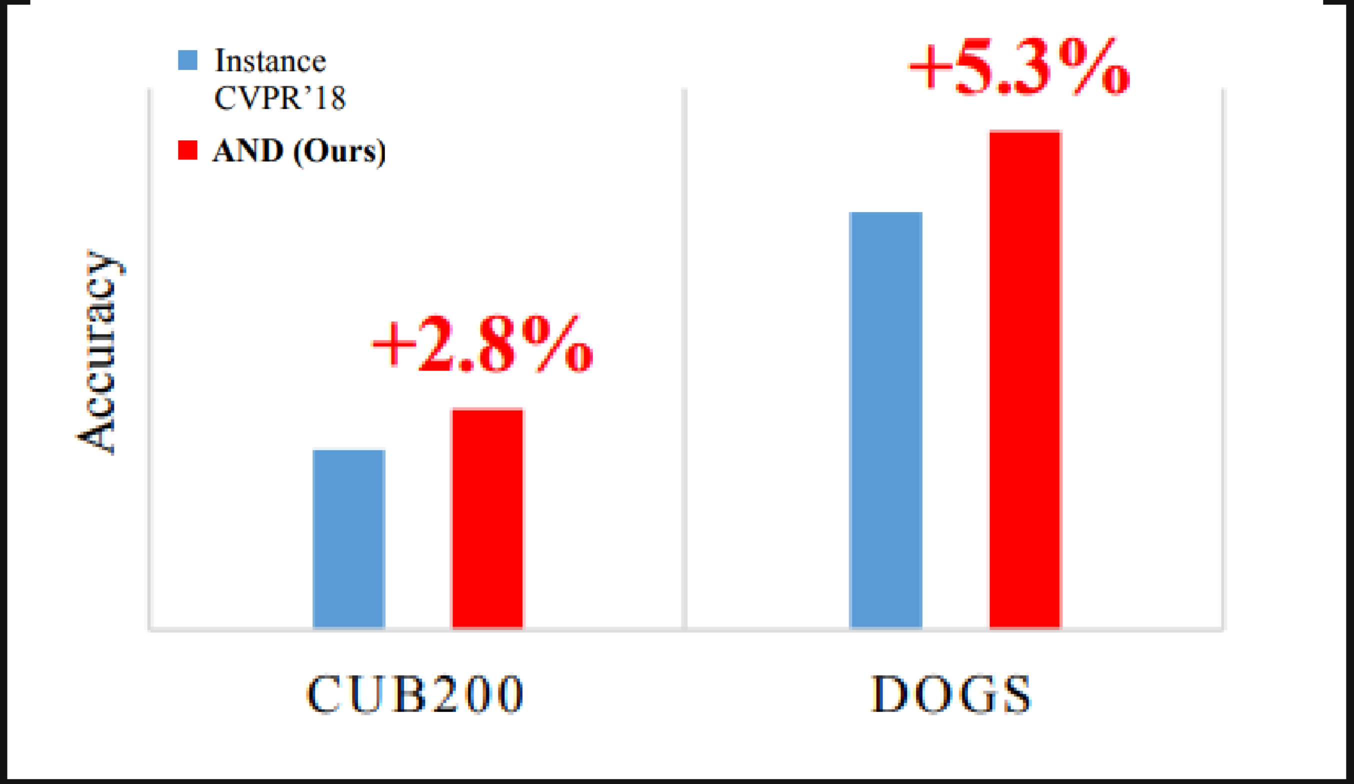


Classifier	Linear Classifier						*k*NN
Feature	conv1	conv2	conv3	conv4	conv5	FC	FC
Random	11.6	17.1	16.9	16.3	14.1	12.0	3.5
Supervised	19.3	36.3	44.2	48.3	50.5	-	-
Context	17.5	23.0	24.5	23.2	20.6	30.4	-
BiGAN	17.7	24.5	31.0	29.9	28.0	32.2	-
Colour	13.1	24.8	31.0	32.6	31.8	35.2	-
Jigsaw	**19.2**	30.1	34.7	33.9	28.3	**38.1**	-
NAT	-	-	-	-	-	36.0	-
Counting	18.0	30.6	34.3	32.5	25.7	-	-
Split-Brain	17.7	29.3	35.4	35.2	32.8	-	11.8
DeepCluster	13.4	**32.3**	**41.0**	39.6	**38.2**	-	26.8
Instance	16.8	26.5	31.8	34.1	35.6	-	**31.3**
AND	15.6	27.0	35.9	**39.7**	37.9	36.7	**31.3**

Comparisons to the State-of-the-Art Methods

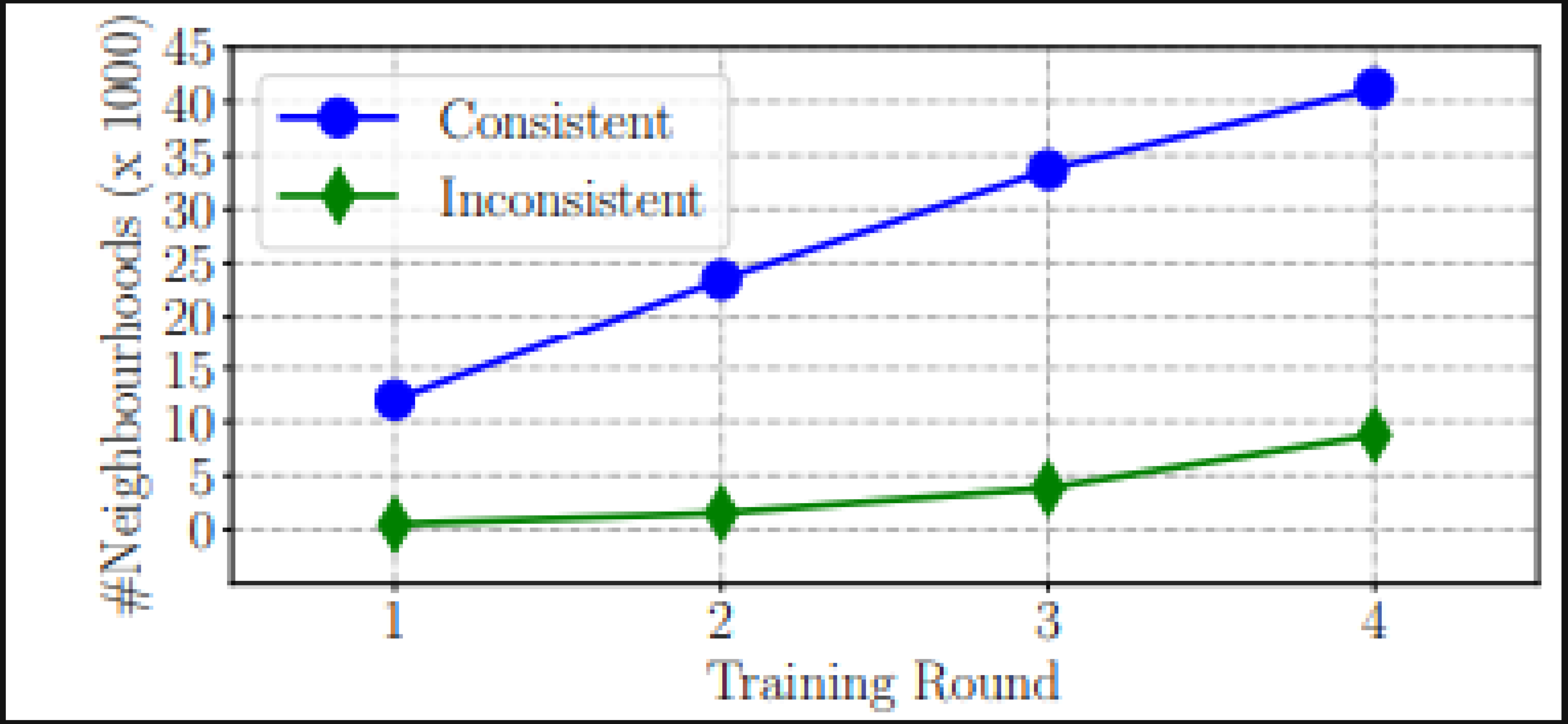
Fine-grained evaluation

Dataset	CUB200	Dogs
Instance	11.6	27.0
AND	**14.4**	**32.3**



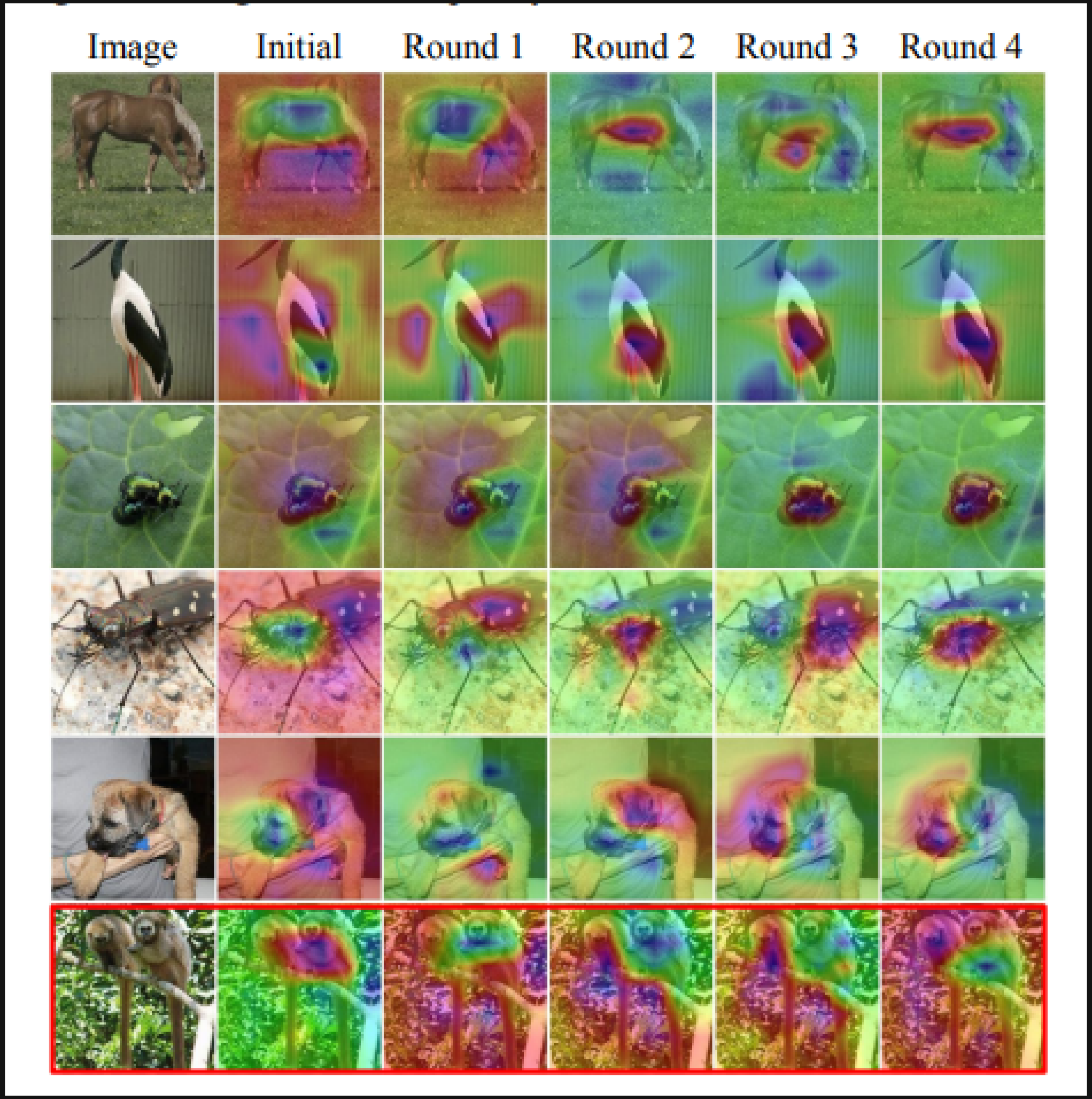
Experiments

Neighbourhood quality



Experiments

Learning attention dynamics



References

Paper:
https://arxiv.org/pdf/1904.11567.pdf

Video:
https://youtu.be/NvaxhaftN_o?list=LL

Github:
https://github.com/Raymond-sci/AND