Aprendizado Profundo (Deep Learning)

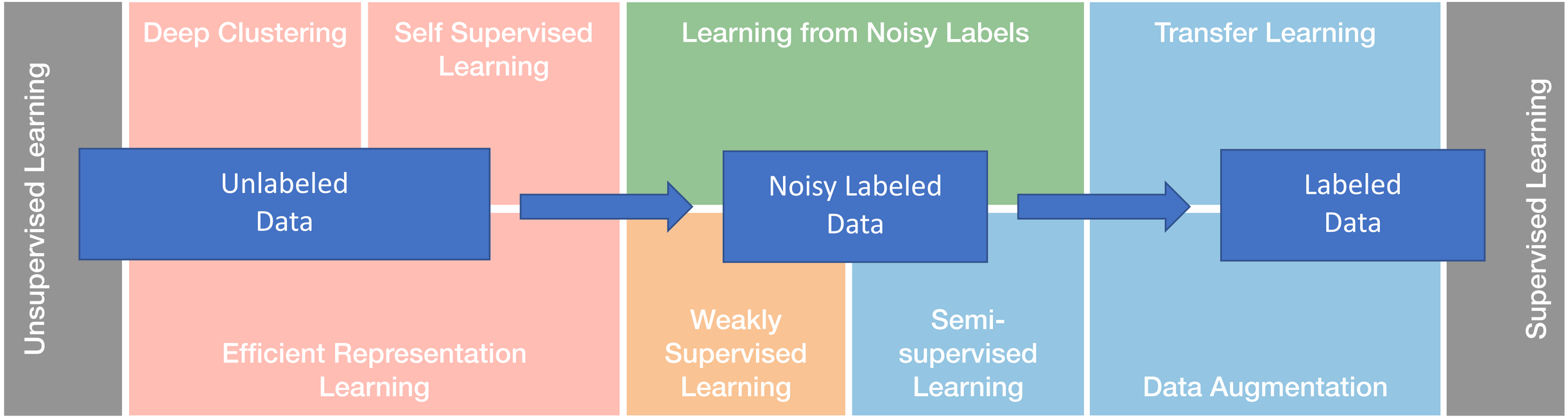
Learning from Unlabeled Data

Dario Oliveira (dario.oliveira@fgv.br)

Why learning from unlabeled data?

Moving towards supervised learning using poor training sets

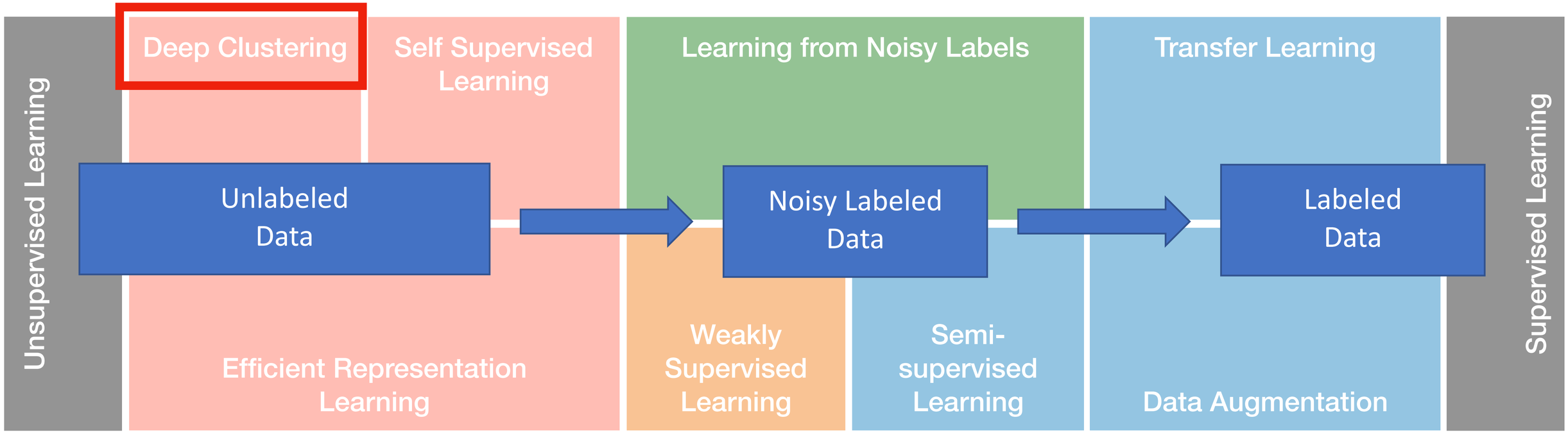
An evolving learning process mitigates massive manual labeling dependency!



Why learning from unlabeled data?

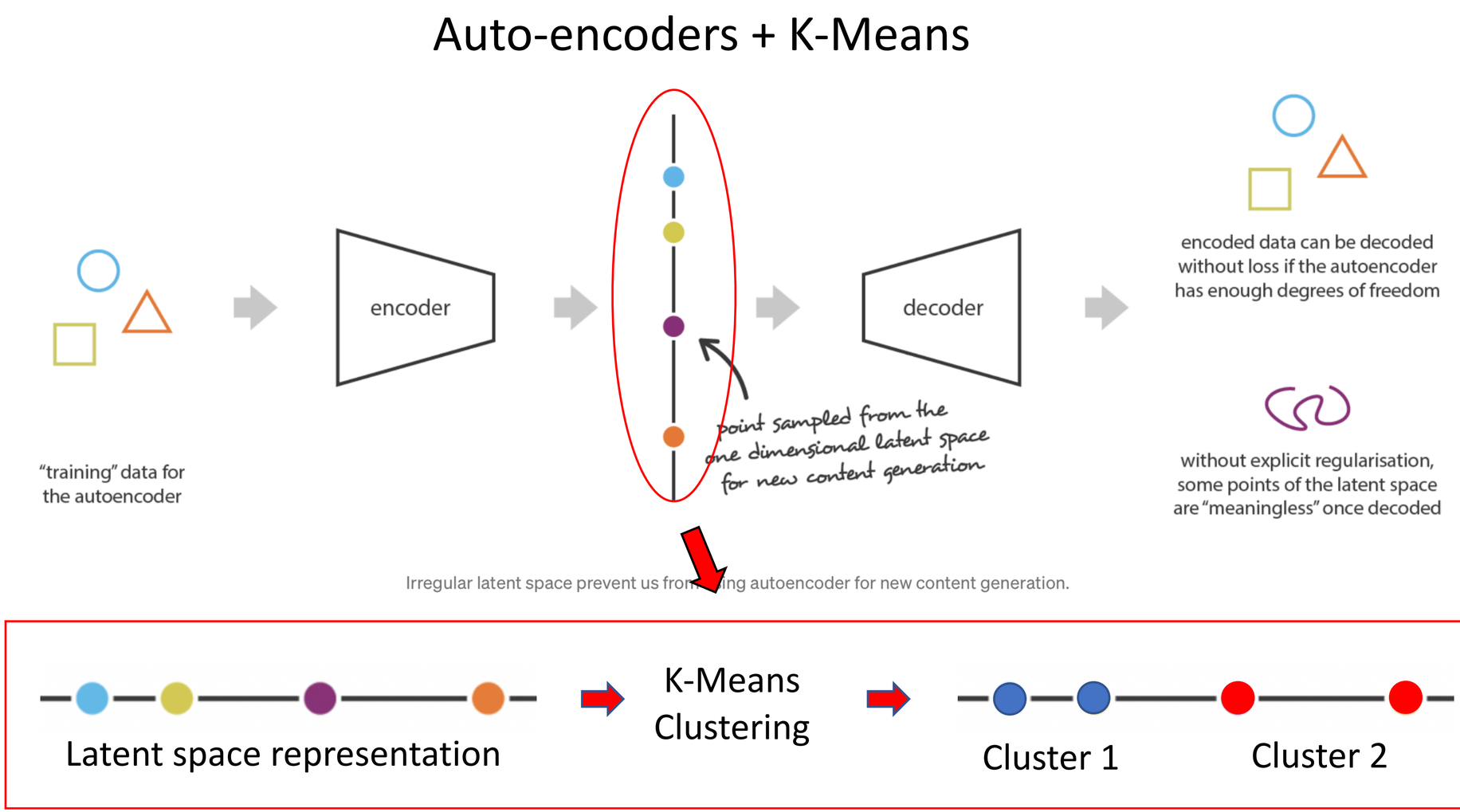
Moving towards supervised learning using poor training sets

An evolving learning process mitigates massive manual labeling dependency!



Deep Clustering

Clustering using efficient representation mapping



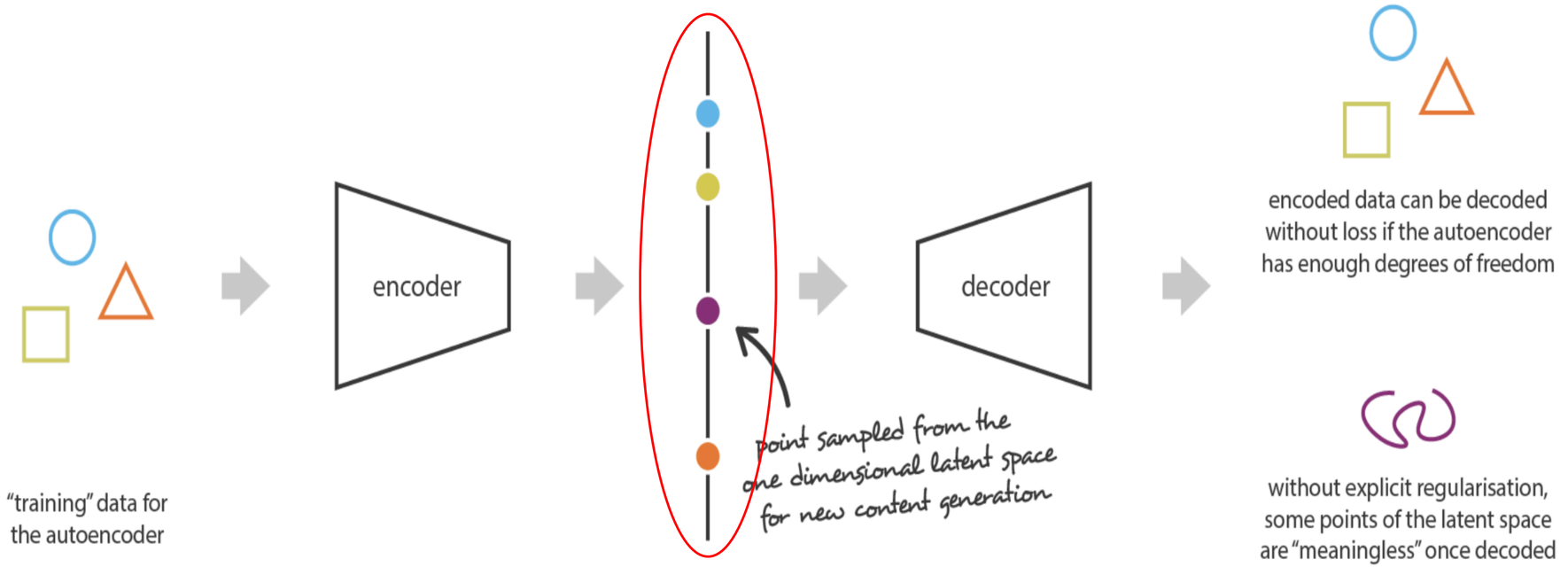
* Figure from https://towardsdatascience.com/understanding-variational-autoencoders-vaes-f70510919f73

Deep Clustering

Reinforcing the regularization of clusters' distribution: DEC

DEC has two phases:

1. parameter initialization with a deep autoencoder (Vincent et al., 2010) and
2. parameter optimization (i.e., clustering), where we iterate between computing an auxiliary target distribution and minimizing the Kullback–Leibler (KL) divergence to it.



Irregular latent space prevent us from using autoencoder for new content generation.

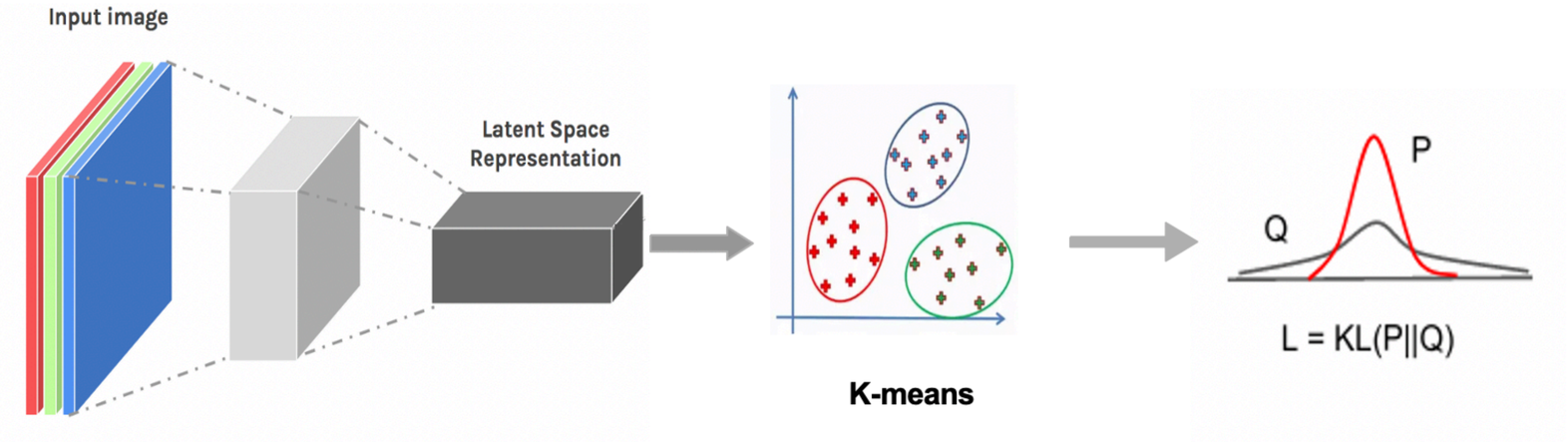
"Unsupervised Deep Embedding for Clustering Analysis", Xie et al, ICML 2016

Deep Clustering

Reinforcing the regularization of clusters' distribution: DEC

DEC has two phases:

1. parameter initialization with a deep autoencoder (Vincent et al., 2010) and
2. parameter optimization (i.e., clustering), where we iterate between computing an auxiliary target distribution and minimizing the Kullback–Leibler (KL) divergence to it.



soft assignment

$$q_{ij} = \frac{(1 + \|z_i - \mu_j\|^2/\alpha)^{-\frac{\alpha+1}{2}}}{\sum_{j'}(1 + \|z_i - \mu_{j'}\|^2/\alpha)^{-\frac{\alpha+1}{2}}}, \text{ we let } \alpha = 1 \text{ for all experiments.}$$

(softmax)

pdf matching

$$L = \mathrm{KL}(P\|Q) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}}.$$

$$p_{ij} = \frac{q_{ij}^2/f_j}{\sum_{j'} q_{ij'}^2/f_{j'}},$$

$f_j = \sum_i q_{ij}$ are soft cluster frequencies

Deep Clustering

Reinforcing the regularization of clusters' distribution: DEC

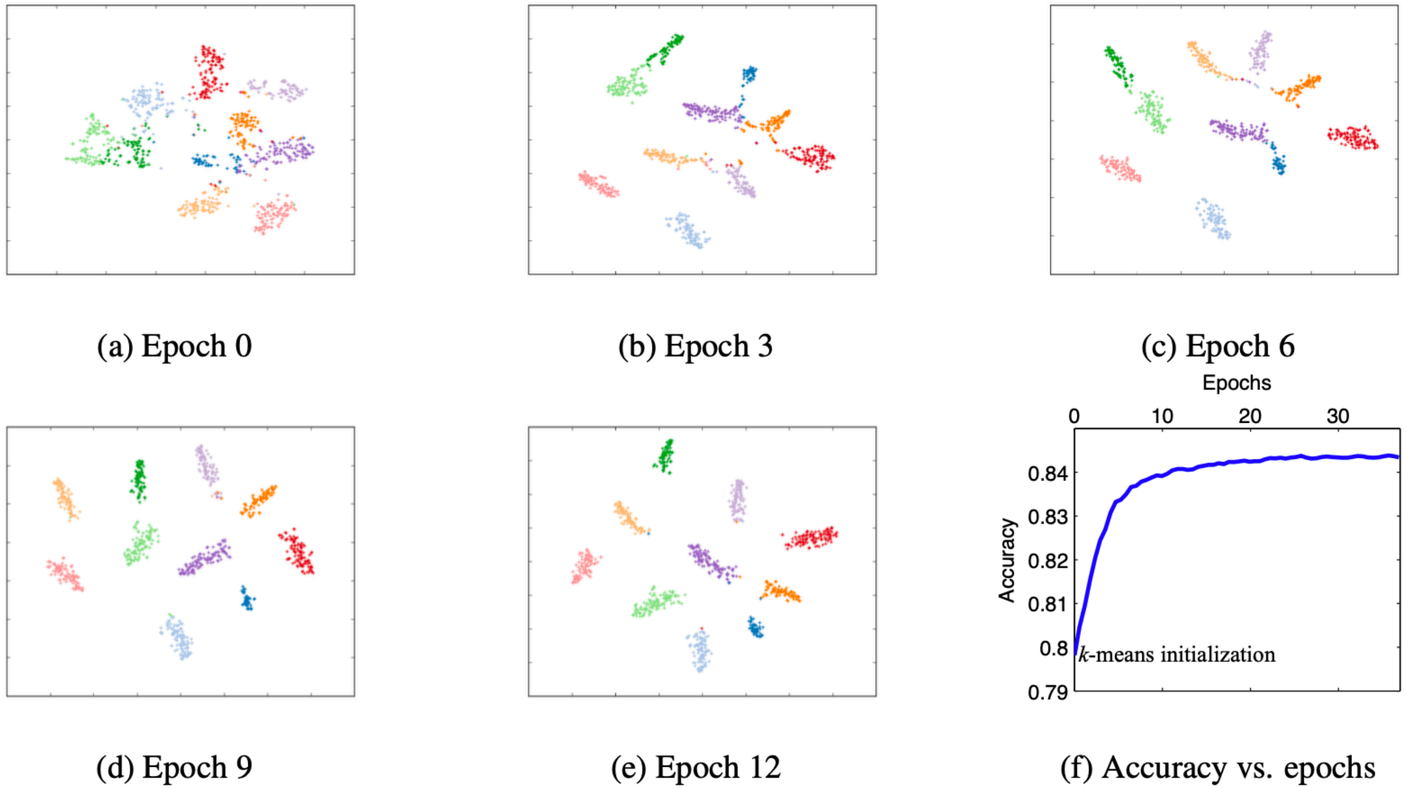


Figure 5. We visualize the latent representation as the KL divergence minimization phase proceeds on MNIST. Note the separation of clusters from epoch 0 to epoch 12. We also plot the accuracy of DEC at different epochs, showing that KL divergence minimization improves clustering accuracy. This figure is best viewed in color.

"Unsupervised Deep Embedding for Clustering Analysis", Xie et al, ICML 2016

Deep Clustering

Reinforcing the regularization of clusters' distribution: DEC

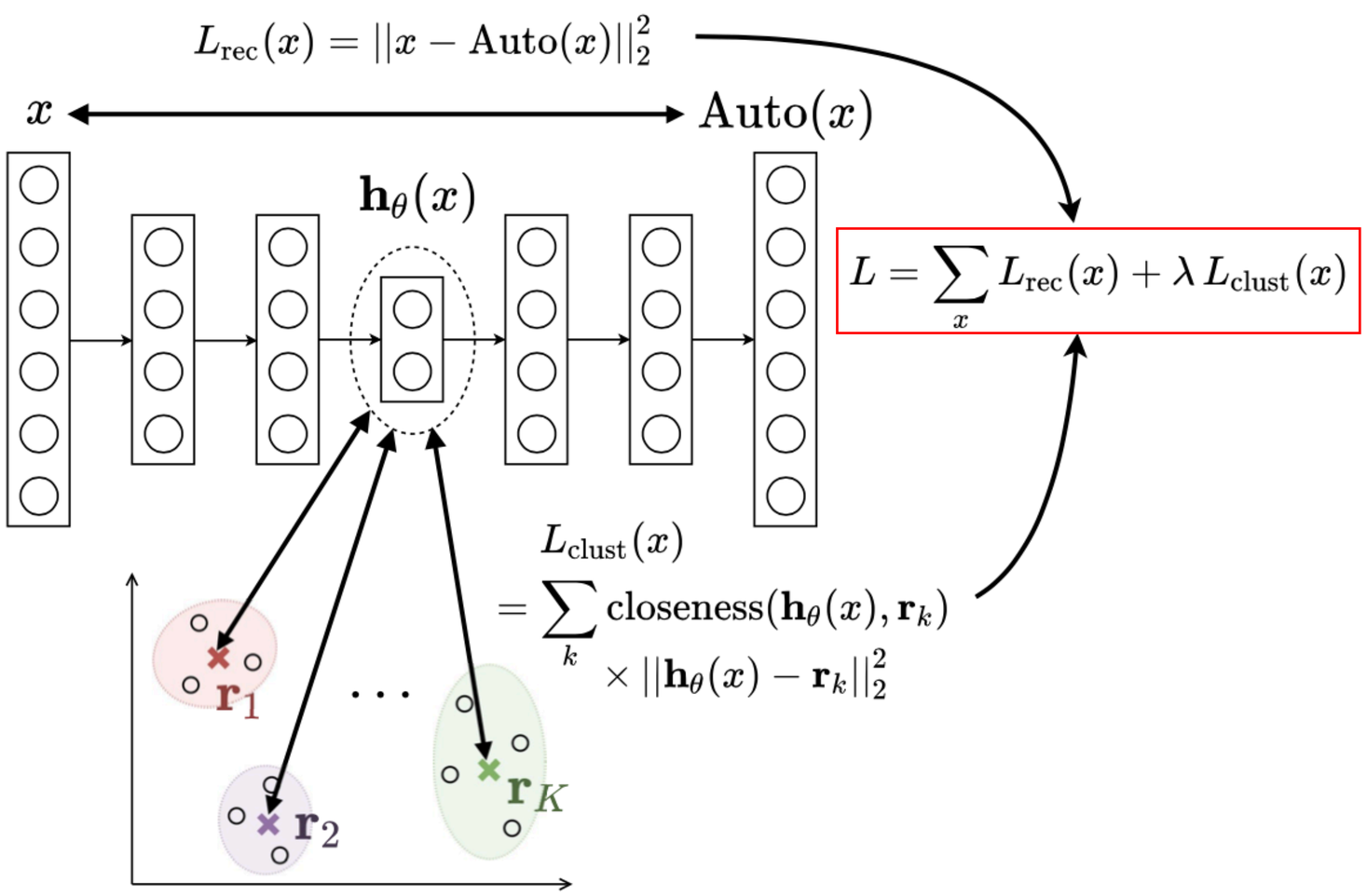
Table 3. Comparison of clustering accuracy (Eq. 10) on autoencoder (AE) feature.

Method	MNIST	STL-HOG	REUTERS-10k	REUTERS
AE+k-means	81.84%	33.92%	66.59%	71.97%
AE+LDMGI	83.98%	32.04%	42.92%	N/A
AE+SEC	81.56%	32.29%	61.86%	N/A
DEC (ours)	**84.30%**	**35.90%**	**72.17%**	**75.63%**

“Unsupervised Deep Embedding for Clustering Analysis”, Xie et al, ICML 2016

Deep Clustering

Implementing a clustering loss: deep k-means



x point in the input data space.

$h_\theta(x)$ representation of x in the embedding space.

θ autoencoder's parameters

r_k the representative of the cluster k, $1 \leq k \leq K$, (K: total number of clusters)

$L_{rec}(x)$ - measures the error between x and its reconstruction A(x;θ) provided by the autoencoder.

$L_{clust}(x)$ - clustering loss corresponding to the k-Means objective function in the embedding space.

λ define the trade-off between the reconstruction and the clustering error in the total loss function (L).

"Deep k-Means: Jointly clustering with k-Means and learning representations", Fard et al., Pattern Recognition Letters, 2020

Deep Clustering

Implementing a clustering loss: deep k-means

$$L = \sum_x L_{\text{rec}}(x) + \lambda L_{\text{clust}}(x)$$

$$\min_{\mathcal{R},\theta} \lim_{\alpha \to \alpha_0} \sum_{x \in \mathcal{X}} g(x, A(x;\theta)) + \lambda \sum_{k=1}^{K} f(\mathbf{h}_\theta(x), \mathbf{r}_k) G_{k,f}(\mathbf{h}_\theta(x), \alpha; \mathcal{R})$$

$\|x - Auto(x)\|_2^2$

input — decoder output

$\|h_\theta(x) - r_k\|_2^2$

encoder output — cluster centroid

$$G_{k,f}(\mathbf{h}_\theta(x), \alpha; \mathcal{R}) = \frac{e^{-\alpha f(\mathbf{h}_\theta(x), \mathbf{r}_k)}}{\sum_{k'=1}^{K} e^{-\alpha f(\mathbf{h}_\theta(x), \mathbf{r}_{k'})}}$$

(temperature-scaled softmax)

"Deep k-Means: Jointly clustering with k-Means and learning representations", Fard et al., Pattern Recognition Letters, 2020

Deep Clustering

Implementing a clustering loss: deep k-means

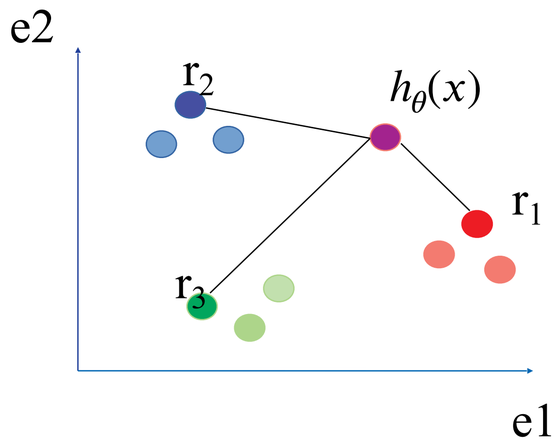
Increase α during training

When $\alpha \rightarrow 0$

Each data point in the embedding space is equally close. Corresponds to removing the influence of $G_{k,f}$

$$G_{k,f}(\mathbf{h}_\theta(x), \alpha; \mathcal{R}) = \frac{e^{-\alpha f(\mathbf{h}_\theta(x), \mathbf{r}_k)}}{\sum_{k'=1}^{K} e^{-\alpha f(\mathbf{h}_\theta(x), \mathbf{r}_{k'})}}$$

Exemplo:



For $\alpha = 0$

$f(h_\theta(x), r_1) = 11$

$f(h_\theta(x), r_2) = 13$

$f(h_\theta(x), r_3) = 14$

$G(h_\theta(x), \alpha, r_1) = 0.33$

$G(h_\theta(x), \alpha, r_2) = 033$

$G(h_\theta(x), \alpha, r_3) = 0\,.\,33$

$$\sum_{k=1}^{K} G(h_\theta(x), \alpha, r_k) = 1$$

"Deep k-Means: Jointly clustering with k-Means and learning representations", Fard et al., Pattern Recognition Letters, 2020

Deep Clustering

Implementing a clustering loss: deep k-means

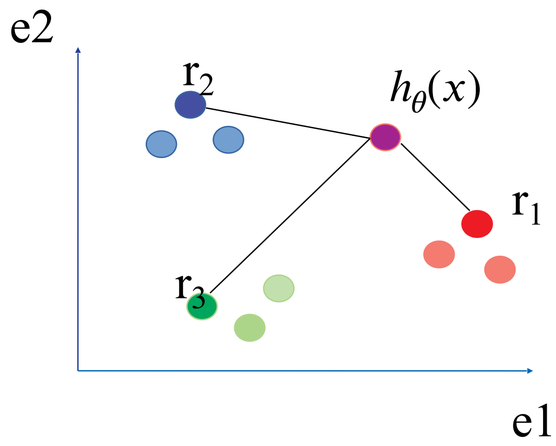
Increase α during training

When $\alpha \rightarrow 1$

Each data point in the embedding space is proportionally close do cluster centroids considering $G_{k,f}$.
Corresponds to a soft assignment.

$$G_{k,f}(\mathbf{h}_\theta(x), \alpha; \mathcal{R}) = \frac{e^{-\alpha f(\mathbf{h}_\theta(x), \mathbf{r}_k)}}{\sum_{k'=1}^{K} e^{-\alpha f(\mathbf{h}_\theta(x), \mathbf{r}_{k'})}}$$

Exemplo:



For α = 1

$f(h_\theta(x), r_1) = 11$ $\quad G(h_\theta(x), \alpha, r_1) = 0.85$

$f(h_\theta(x), r_2) = 13$ $\quad G(h_\theta(x), \alpha, r_2) = 0.11$

$f(h_\theta(x), r_3) = 14$ $\quad G(h_\theta(x), \alpha, r_3) = 0.04$

$$\sum_{k=1}^{K} G(h_\theta(x), \alpha, r_k) = 1$$

“Deep k-Means: Jointly clustering with k-Means and learning representations”, Fard et al., Pattern Recognition Letters, 2020

Deep Clustering

Implementing a clustering loss: deep k-means

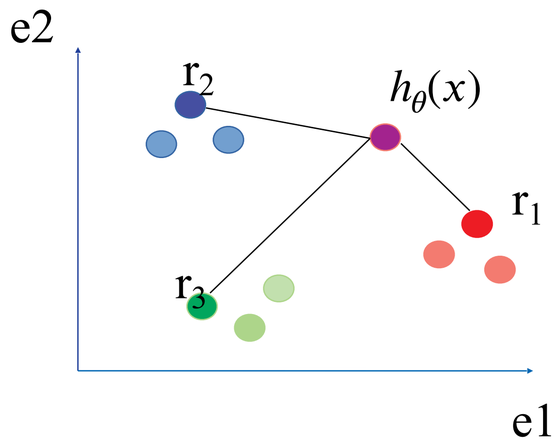
Increase α during training

When $\alpha \to \infty$

$G_{k;f}$ takes binary values in the limit where each point in the embedding space belongs to a single cluster
Corresponds to hard assigment.

$$G_{k,f}(\mathbf{h}_\theta(x), \alpha; \mathcal{R}) = \frac{e^{-\alpha f(\mathbf{h}_\theta(x), \mathbf{r}_k)}}{\sum_{k'=1}^{K} e^{-\alpha f(\mathbf{h}_\theta(x), \mathbf{r}_{k'})}}$$

Exemplo:



For α = 10

$$f(h_\theta(x), r_1) = 11$$
$$f(h_\theta(x), r_2) = 13$$
$$f(h_\theta(x), r_3) = 14$$

$$G(h_\theta(x), \alpha, r_1) \approx 1$$
$$G(h_\theta(x), \alpha, r_2) \approx 0$$
$$G(h_\theta(x), \alpha, r_3) \approx 0$$

$$\sum_{k=1}^{K} G(h_\theta(x), \alpha, r_k) = 1$$

"Deep k-Means: Jointly clustering with k-Means and learning representations", Fard et al., Pattern Recognition Letters, 2020

Deep Clustering

Implementing a clustering loss: deep k-means

Algorithm 1: Deep k-Means algorithm

Input: data $\mathcal{X}$, number of clusters K, balancing parameter λ, scheme for α, number of epochs T, number of minibatches N, learning rate η

Output: autoencoder parameters θ, cluster representatives $\mathcal{R}$

Initialize θ and $\mathbf{r}_k$, $1 \leq k \leq K$ (randomly or through pretraining)

for $\alpha = m_\alpha$ **to** M_α **do** `# inverse temperature`

  for $t = 1$ **to** T **do** `# epochs per` α

    for $n = 1$ **to** N **do** `# minibatches`

      Draw a minibatch $\tilde{\mathcal{X}} \subset \mathcal{X}$

      Update $(\theta, \mathcal{R})$ using SGD (Eq. 3)

    end

  end

end

"Deep k-Means: Jointly clustering with k-Means and learning representations", Fard et al., Pattern Recognition Letters, 2020

Deep Clustering

Implementing a clustering loss: deep k-means

Table 1
Clustering results of the *k*-Means-related methods. Performance is measured in terms of NMI and ACC (%); higher is better. Each cell contains the average and standard deviation computed over 10 runs. Bold (resp. underlined) values correspond to results with no significant difference ($p > 0.05$) to the best approach with (resp. without) pretraining for each dataset/metric pair.

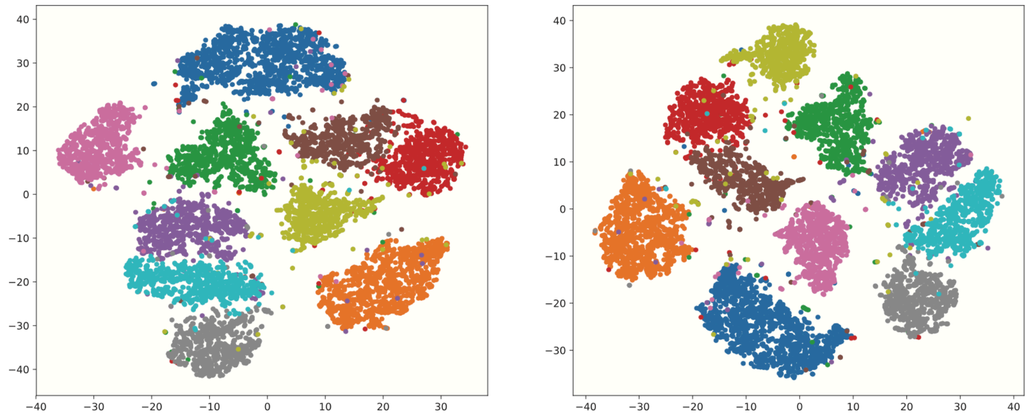
Model	MNIST		USPS		20NEWS		RCV1	
	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI
KM	53.5 ± 0.3	49.8 ± 0.5	67.3 ± 0.1	61.4 ± 0.1	23.2 ± 1.5	21.6 ± 1.8	50.8 ± 2.9	**31.3 ± 5.4**
AE-KM	80.8 ± 1.8	75.2 ± 1.1	72.9 ± 0.8	71.7 ± 1.2	**49.0 ± 2.9**	44.5 ± 1.5	**56.7 ± 3.6**	**31.5 ± 4.3**
Deep clustering approaches without pretraining								
DCN^{np}	34.8 ± 3.0	18.1 ± 1.0	36.4 ± 3.5	16.9 ± 1.3	17.9 ± 1.0	9.8 ± 0.5	41.3 ± 4.0	6.9 ± 1.8
DKM^a	82.3 ± 3.2	78.0 ± 1.9	**75.5 ± 6.8**	73.0 ± 2.3	44.8 ± 2.4	42.8 ± 1.1	**53.8 ± 5.5**	**28.0 ± 5.8**
Deep clustering approaches with pretraining								
DCN^p	81.1 ± 1.9	75.7 ± 1.1	73.0 ± 0.8	71.9 ± 1.2	**49.2 ± 2.9**	44.7 ± 1.5	**56.7 ± 3.6**	**31.6 ± 4.3**
DKM^p	**84.0 ± 2.2**	**79.6 ± 0.9**	**75.7 ± 1.3**	**77.6 ± 1.1**	**51.2 ± 2.8**	**46.7 ± 1.2**	**58.3 ± 3.8**	**33.1 ± 4.9**

DKM^p = DKM^a after fine tunning with alpha fixed = 10.000

"Deep k-Means: Jointly clustering with k-Means and learning representations", Fard et al., Pattern Recognition Letters, 2020

Deep Clustering

Implementing a clustering loss: deep k-means



(a) AE (b) DCN^{p}

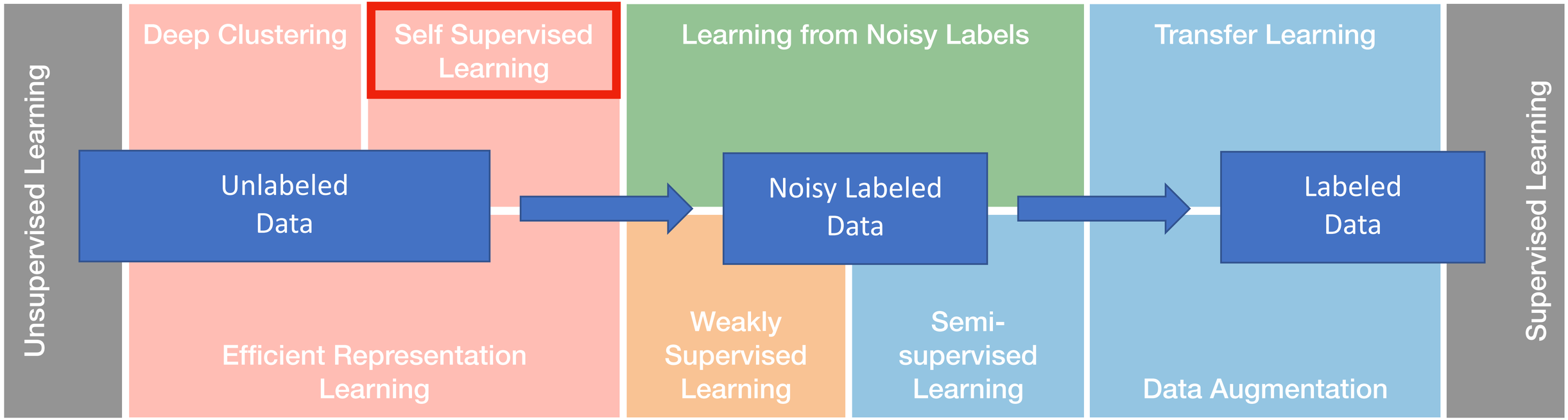
(c) DKM^{a} (d) DKM^{p}

"Deep k-Means: Jointly clustering with k-Means and learning representations", Fard et al., Pattern Recognition Letters, 2020

Why learning from unlabeled data?

Moving towards supervised learning using poor training sets

An evolving learning process mitigates massive manual labeling dependency!



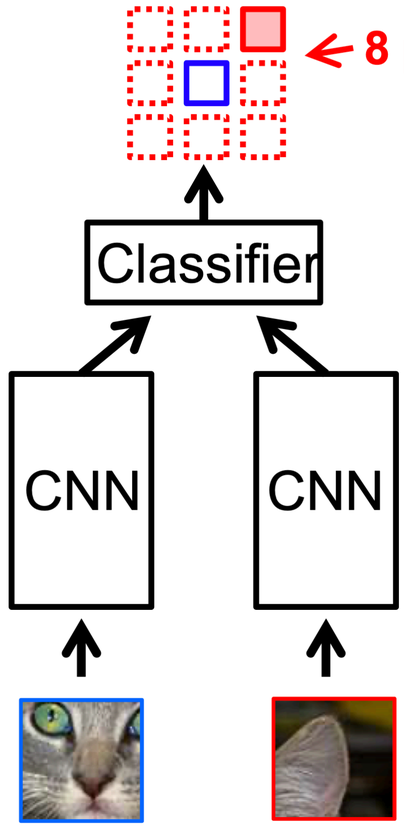
Self-Supervised Learning

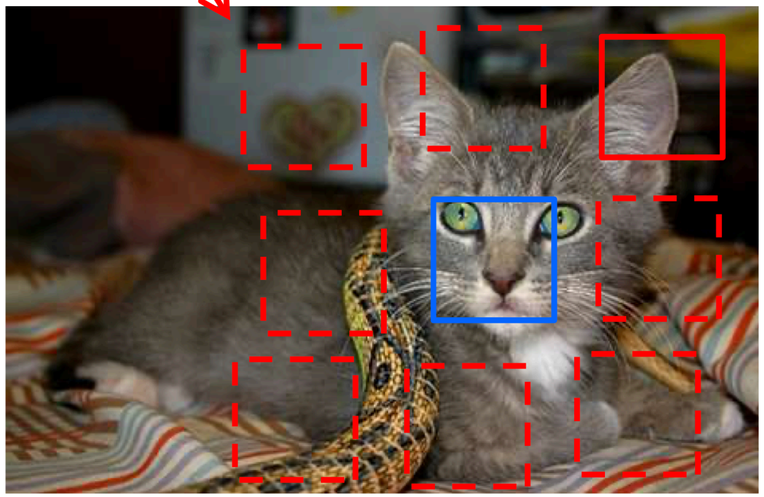
Training using supervision from the data itself

"A form of unsupervised learning where the data provides the supervision"

A. Zisserman

Train network to predict relative position of two regions in the same image





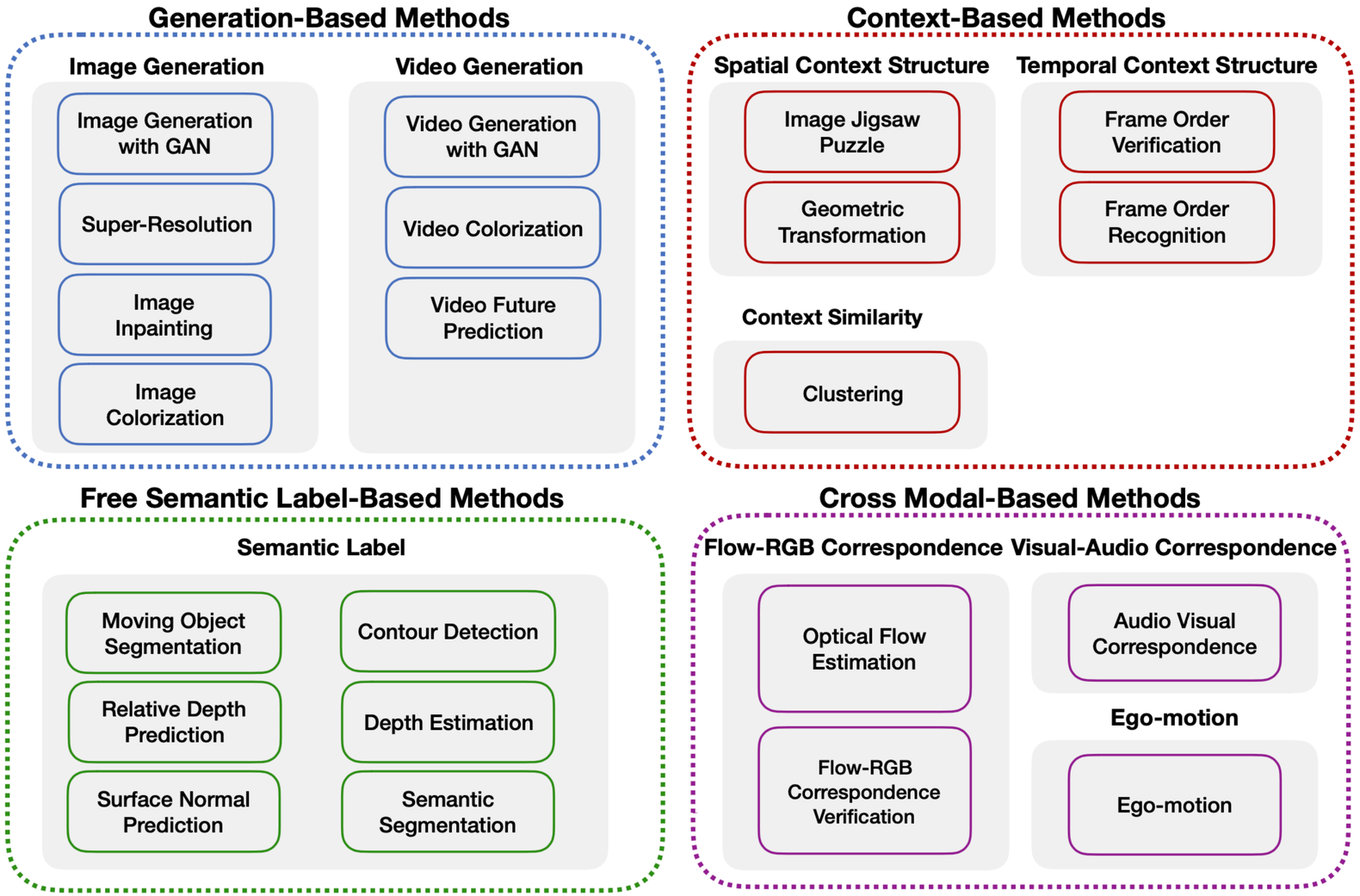
Randomly Sample Patch

Sample Second Patch

Unsupervised visual representation learning by context prediction,
Carl Doersch, Abhinav Gupta, Alexei A. Efros, ICCV 2015

Self-Supervised Learning

Training using supervision from the data itself

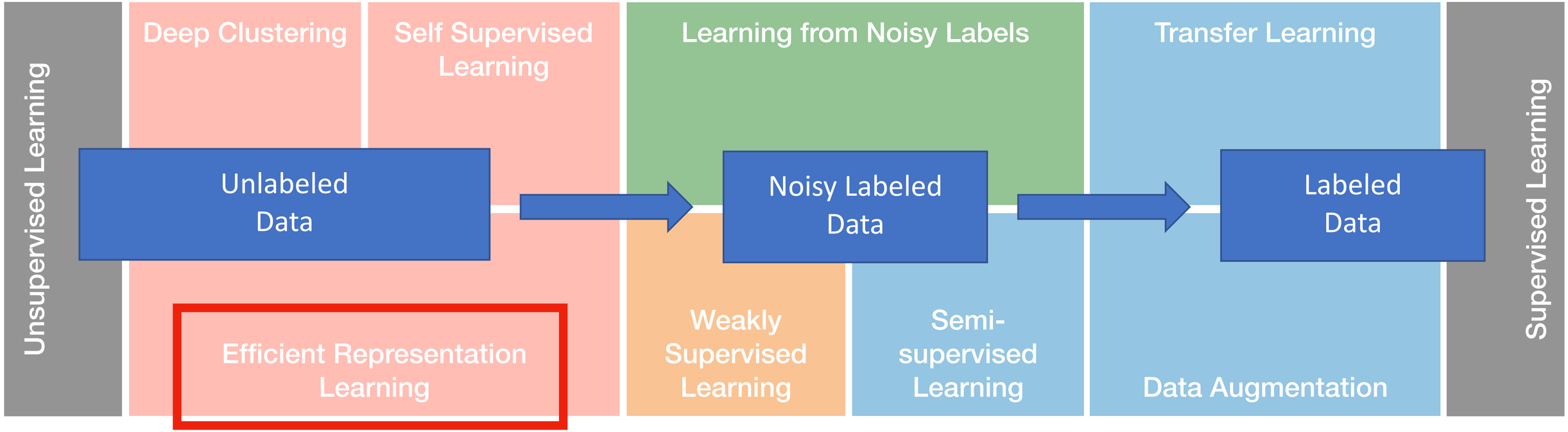


"Self-supervised Visual Feature Learning with Deep Neural Networks: A Survey", Jing and Tian, ArXiv 2019

Why learning from unlabeled data?

Moving towards supervised learning using poor training sets

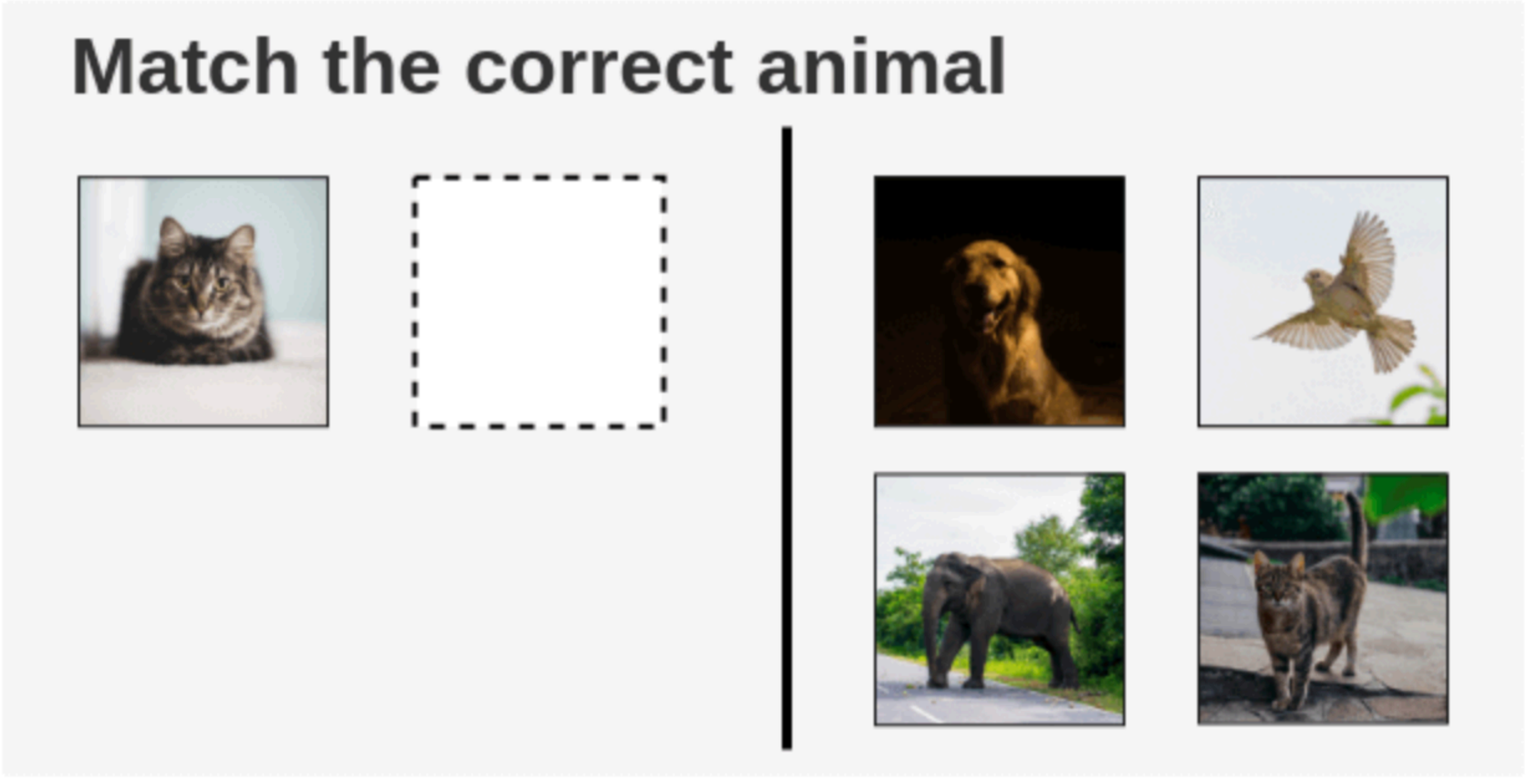
An evolving learning process mitigates massive manual labeling dependency!



Contrastive Learning

Learning efficient data representation without labeling

What is the inspiration behind?

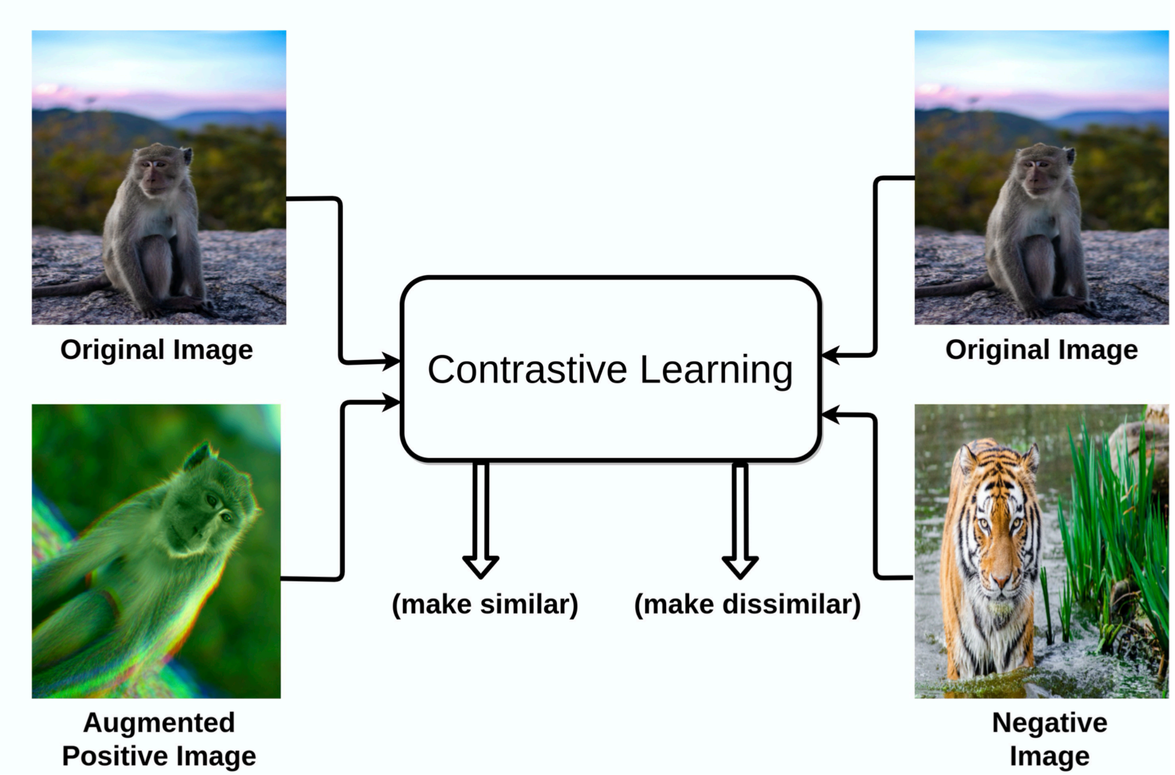


From https://amitness.com/2020/03/illustrated-simclr/

Contrastive Learning

Learning efficient data representation without labeling

Strong data augmentation leads to robust class organization!

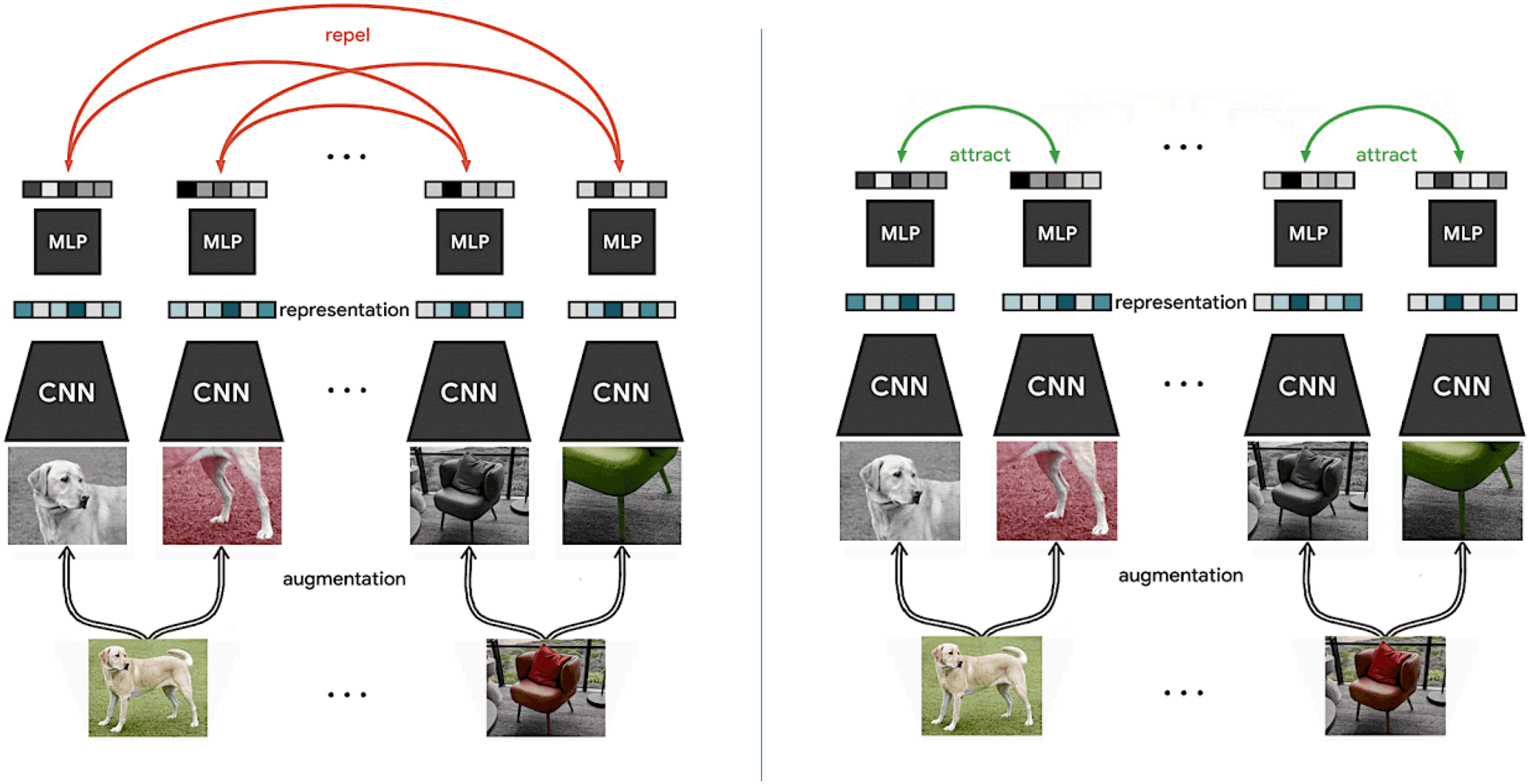


Tool for *efficient* visual representation learning → transfer learning to *downstream tasks*

A SURVEY ON CONTRASTIVE SELF-SUPERVISED LEARNING, Jaiswal et al., Arxiv 2020

Contrastive Learning

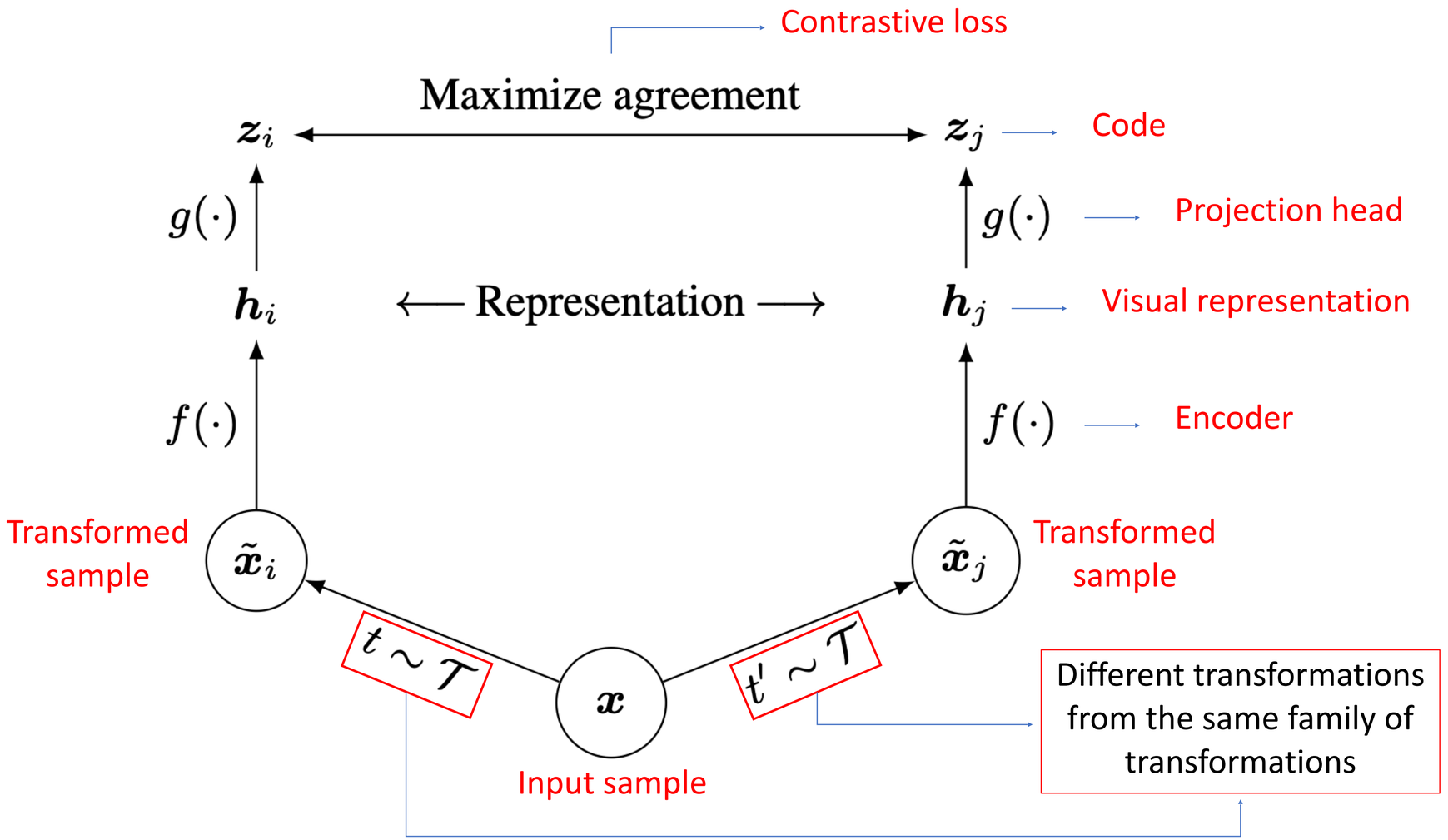
Learning efficient data representation: SimCLR



"A Simple Framework for Contrastive Learning of Visual Representations", Chen et al., ICML 2020

Contrastive Learning

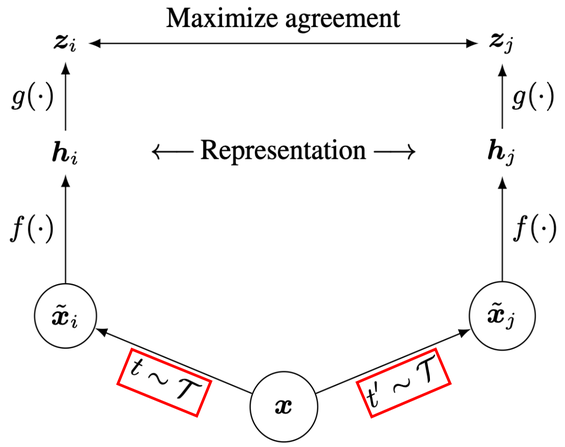
Learning efficient data representation: SimCLR



"A Simple Framework for Contrastive Learning of Visual Representations", Chen et al., ICML 2020

Contrastive Learning

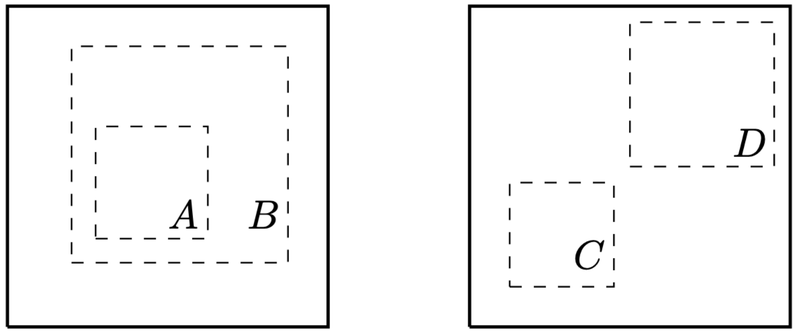
Learning efficient data representation: SimCLR



Stochastic *data augmentation* module

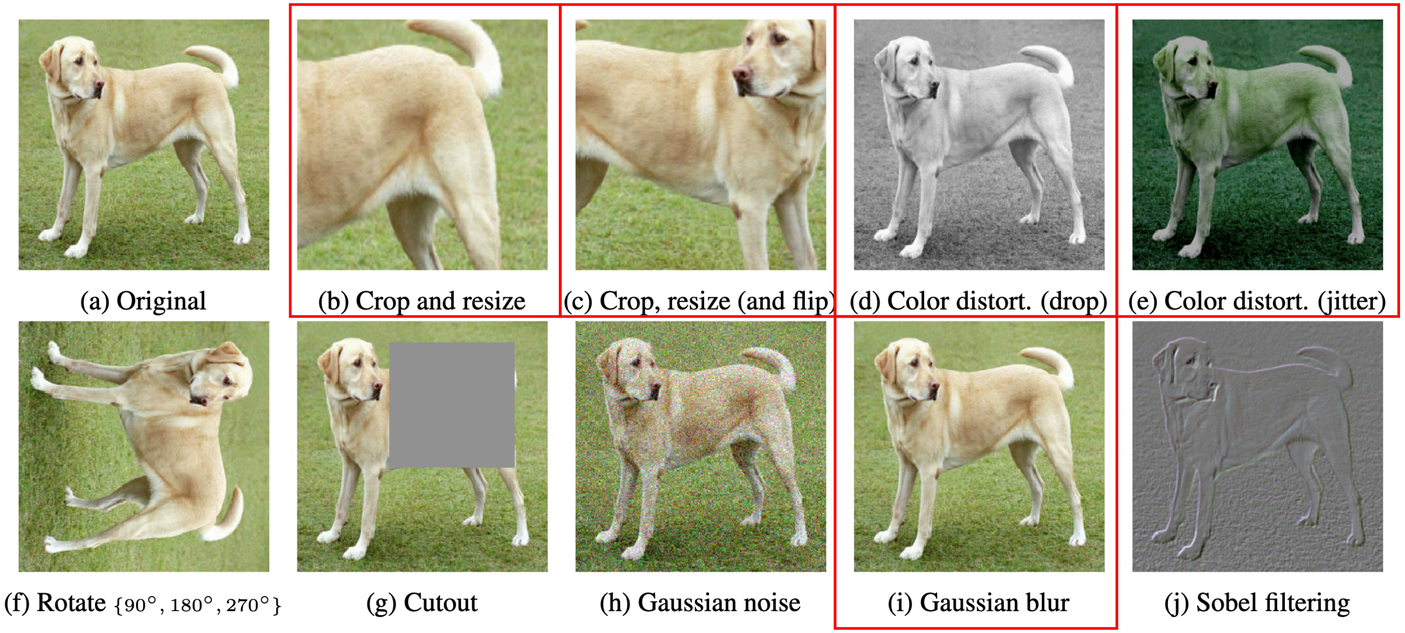
It is composed of three simple augmentations in sequence:

1. *random cropping* followed by resize back to the original size,
2. *random color distortions*, and
3. *random Gaussian blur*



(a) Global and local views. (b) Adjacent views.

Figure 3. Solid rectangles are images, dashed rectangles are random crops. By randomly cropping images, we sample contrastive prediction tasks that include global to local view ($B \rightarrow A$) or adjacent view ($D \rightarrow C$) prediction.

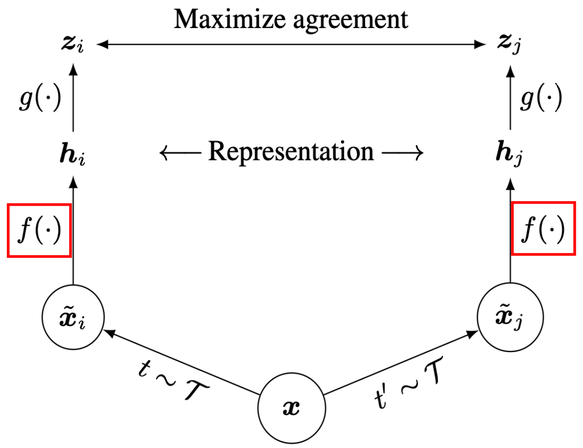


(a) Original (b) Crop and resize (c) Crop, resize (and flip) (d) Color distort. (drop) (e) Color distort. (jitter)

(f) Rotate $\{90°, 180°, 270°\}$ (g) Cutout (h) Gaussian noise (i) Gaussian blur (j) Sobel filtering

Contrastive Learning

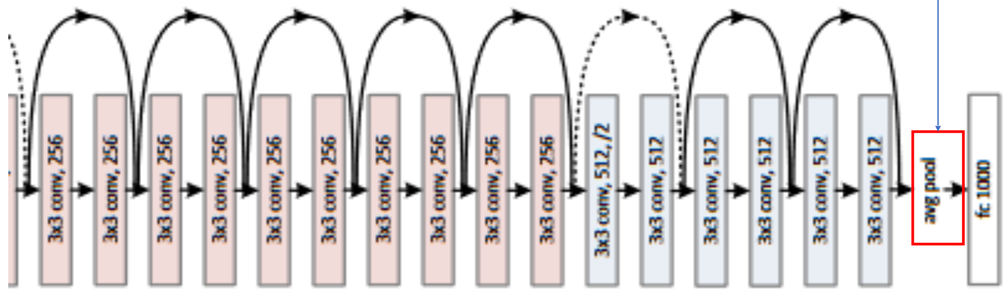
Learning efficient data representation: SimCLR



Encoder to visual representation module

The authors used ResNet-50 in 3 different hidden layer widths (width multipliers of 1×, 2×, and 4×)

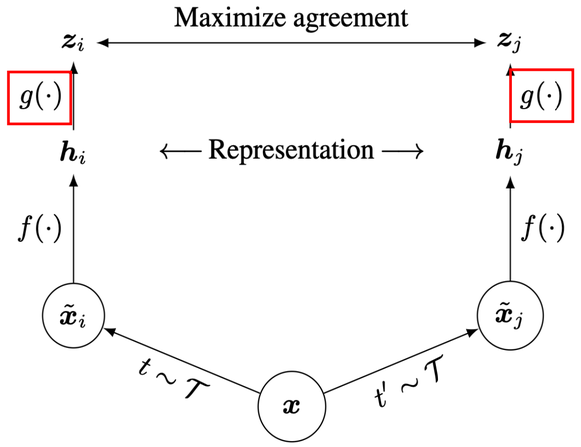
The visual representation is the output after the average pooling layer



Architecture	Label fraction 1%		10%		100%	
	Top 1	Top 5	Top 1	Top 5	Top 1	Top 5
ResNet-50	49.4	76.6	66.1	88.1	76.0	93.1
ResNet-50 (2×)	59.4	83.7	71.8	91.2	79.1	94.8
ResNet-50 (4×)	64.1	86.6	74.8	92.8	80.4	95.4

Contrastive Learning

Learning efficient data representation: SimCLR

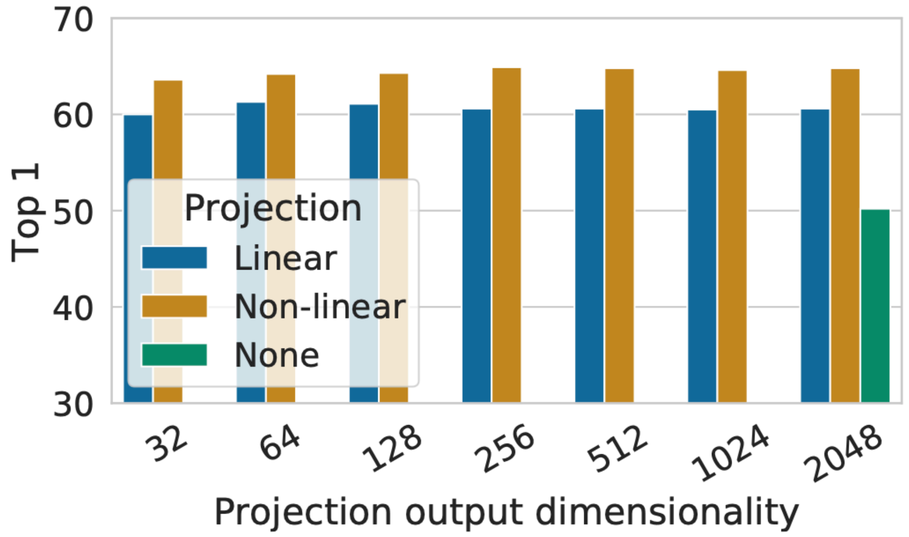


Projection Head Module

The authors use a MLP with one hidden layer to obtain $z_i = g(h_i) = W_2$

$\sigma(W_1 h_i)$ where σ is a ReLU non- linearity.

"the hidden layer before the projection head is a better representation than the layer after"
"the importance of using the representation before the nonlinear projection is due to loss of information induced by the contrastive loss"

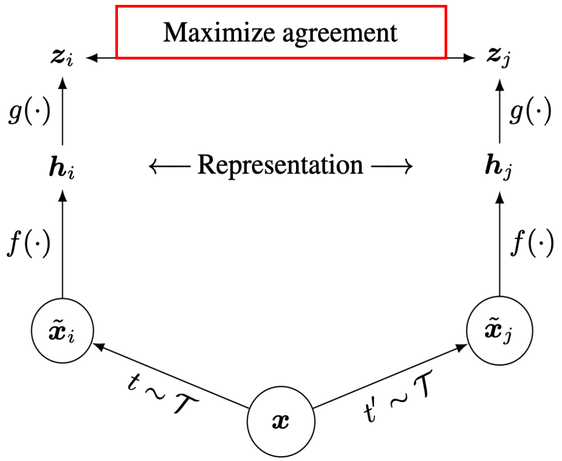


What to predict?	Random guess	Representation h	Representation $g(h)$
Color vs grayscale	80	99.3	97.4
Rotation	25	67.6	25.6
Orig. vs corrupted	50	99.5	59.6
Orig. vs Sobel filtered	50	96.6	56.3

Table 3. Accuracy of training additional MLPs on different representations to predict the transformation applied.

Contrastive Learning

Learning efficient data representation: SimCLR



Contrastive Loss

The authors used the "normalized temperature-scaled cross-entropy"

Given a positive pair, we treat the other 2(N − 1) augmented examples within a minibatch as negative examples.

The loss function for a positive pair of examples (i, j) is:

temperature

τ	Entropy	Contrastive acc.	Top 1
0.05	1.0	90.5	59.7
0.1	4.5	87.8	64.4
0.5	8.2	68.2	60.7
1	8.3	59.1	58.0

$$\ell_{i,j} = -\log \frac{\exp(\text{sim}(\boldsymbol{z}_i, \boldsymbol{z}_j)/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(\text{sim}(\boldsymbol{z}_i, \boldsymbol{z}_k)/\tau)}$$ (softmax)

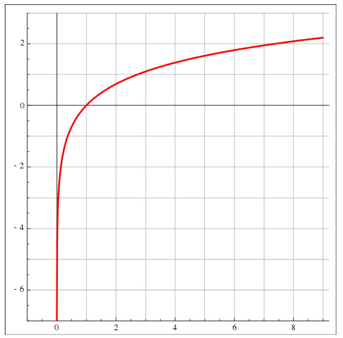
$$\text{sim}(\boldsymbol{u}, \boldsymbol{v}) = \boldsymbol{u}^\top \boldsymbol{v} / \|\boldsymbol{u}\| \|\boldsymbol{v}\|$$

The final loss is computed across all positive pairs, both (i,j) and (j,i), in a mini-batch.

$$L = \frac{1}{2N} \sum_{k=1}^{N} [l(2k-1, 2k) + l(2k, 2k-1)]$$

Contrastive Learning

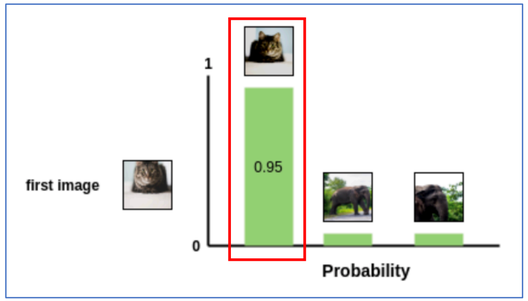
Learning efficient data representation: SimCLR



Log function

Better Understanding the Contrastive Loss

$$l(i,j) = -log\frac{exp(s_{i,j})}{\sum_{k=}^{2N}\mathbb{1}l_{[k!=i]}exp(s_{i,k})}$$



$$l(\,,\,) = -log\left(\frac{e^{similarity(\,,\,)}}{e^{similarity(\,,\,)} + e^{similarity(\,,\,)} + e^{similarity(\,,\,)}}\right)$$

Interchanged

$$l(\,,\,) = -log\left(\frac{e^{similarity(\,,\,)}}{e^{similarity(\,,\,)} + e^{similarity(\,,\,)} + e^{similarity(\,,\,)}}\right)$$

From https://amitness.com/2020/03/illustrated-simclr/

Contrastive Learning

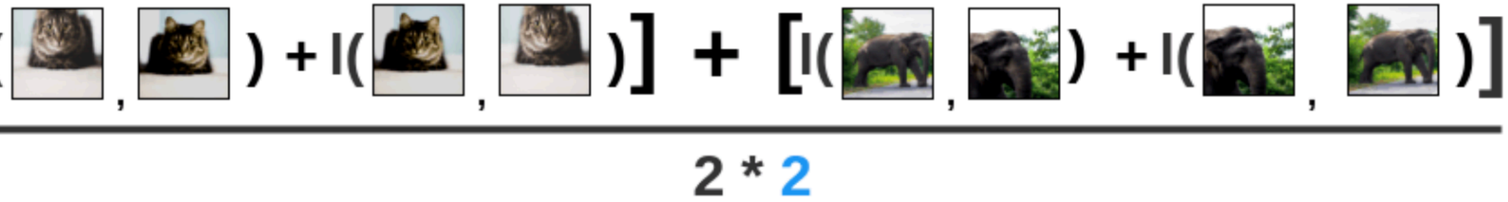
Learning efficient data representation: SimCLR

Better Understanding the Contrastive Loss

Minimize L: $$L = \frac{1}{2N}\sum_{k=1}^{N}[l(2k-1,2k) + l(2k,2k-1)]$$

Pair 1 Loss (k=1) Pair 2 Loss (k=2)

$$L = \frac{[l(\,,\,) + l(\,,\,)] + [l(\,,\,) + l(\,,\,)]}{2 * 2}$$



From https://amitness.com/2020/03/illustrated-simclr/

Contrastive Learning

Learning efficient data representation: SimCLR

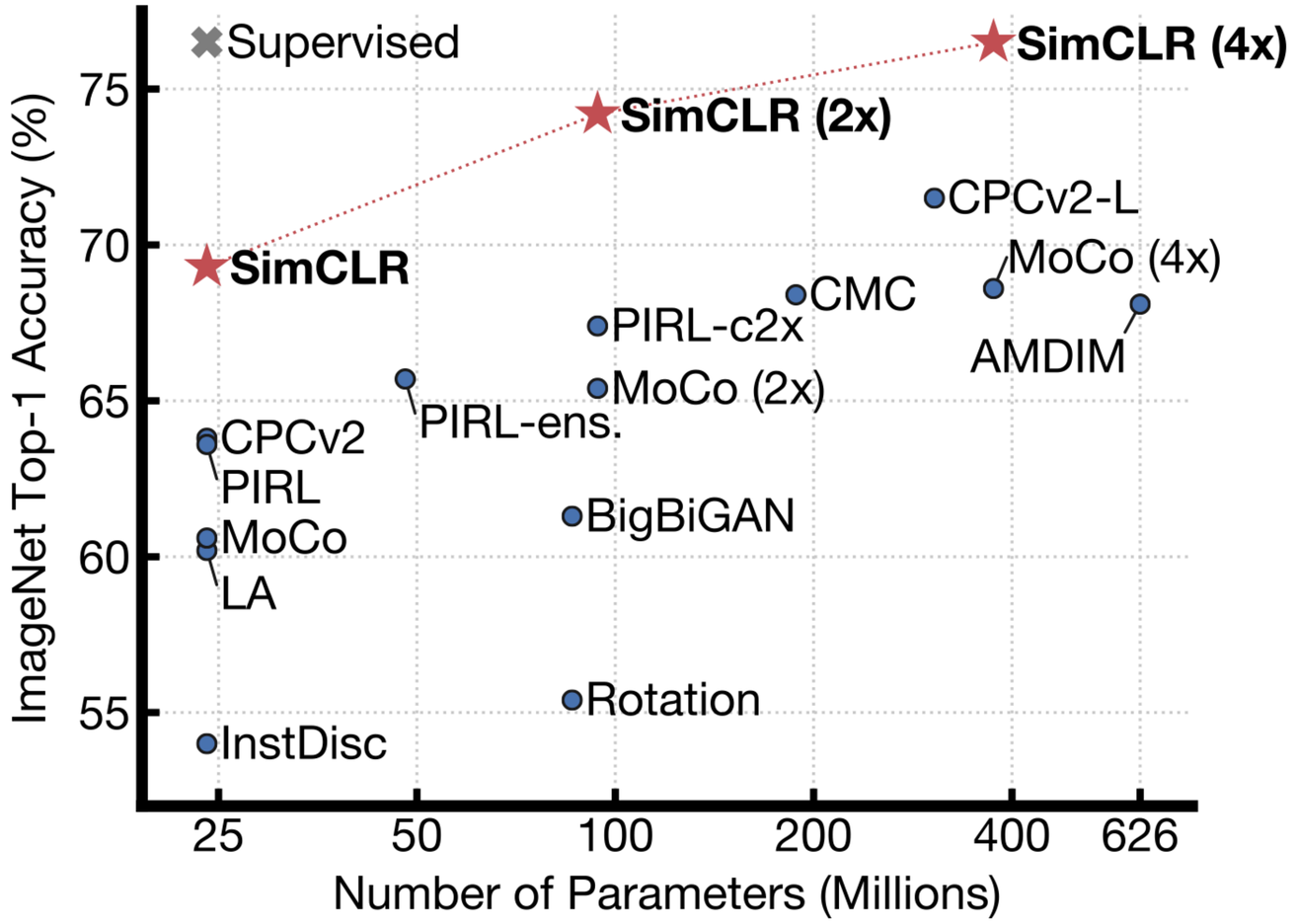
Algorithm 1 SimCLR's main learning algorithm.

input: batch size N, constant τ, structure of f, g, $\mathcal{T}$.
for sampled minibatch $\{\boldsymbol{x}_k\}_{k=1}^{N}$ **do**
  for all $k \in \{1, \dots, N\}$ **do**
    draw two augmentation functions $t \sim \mathcal{T}$, $t' \sim \mathcal{T}$
    # the first augmentation
    $\tilde{\boldsymbol{x}}_{2k-1} = t(\boldsymbol{x}_k)$
    $\boldsymbol{h}_{2k-1} = f(\tilde{\boldsymbol{x}}_{2k-1})$ # representation
    $\boldsymbol{z}_{2k-1} = g(\boldsymbol{h}_{2k-1})$ # projection
    # the second augmentation
    $\tilde{\boldsymbol{x}}_{2k} = t'(\boldsymbol{x}_k)$
    $\boldsymbol{h}_{2k} = f(\tilde{\boldsymbol{x}}_{2k})$ # representation
    $\boldsymbol{z}_{2k} = g(\boldsymbol{h}_{2k})$ # projection
  end for
  for all $i \in \{1, \dots, 2N\}$ and $j \in \{1, \dots, 2N\}$ **do**
    $s_{i,j} = \boldsymbol{z}_i^\top \boldsymbol{z}_j / (\|\boldsymbol{z}_i\| \|\boldsymbol{z}_j\|)$ # pairwise similarity
  end for
  define $\ell(i, j)$ **as** $\ell(i,j) = -\log \frac{\exp(s_{i,j}/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(s_{i,k}/\tau)}$
  $\mathcal{L} = \frac{1}{2N} \sum_{k=1}^{N} [\ell(2k-1, 2k) + \ell(2k, 2k-1)]$
  update networks f and g to minimize $\mathcal{L}$
end for
return encoder network $f(\cdot)$, and throw away $g(\cdot)$

To keep it simple, we do not train the model with a memory bank (Wu et al., 2018; He et al., 2019). Instead, we vary the training batch size N from 256 to 8192. A batch size of 8192 gives us 16382 negative examples per positive pair from both augmentation views. Training with large batch size may be unstable when using standard SGD/Momentum with linear learning rate scaling (Goyal et al., 2017). To stabilize the training, we use the LARS optimizer (You et al., 2017) for all batch sizes. We train our model with Cloud TPUs, using 32 to 128 cores depending on the batch size.[2]

Contrastive Learning

Learning efficient data representation: SimCLR



Method	Architecture	Label fraction 1%	Label fraction 10%
		Top 5	
Supervised baseline	ResNet-50	48.4	80.4
Methods using other label-propagation:			
Pseudo-label	ResNet-50	51.6	82.4
VAT+Entropy Min.	ResNet-50	47.0	83.4
UDA (w. RandAug)	ResNet-50	-	88.5
FixMatch (w. RandAug)	ResNet-50	-	89.1
S4L (Rot+VAT+En. M.)	ResNet-50 (4×)	-	91.2
Methods using representation learning only:			
InstDisc	ResNet-50	39.2	77.4
BigBiGAN	RevNet-50 (4×)	55.2	78.8
PIRL	ResNet-50	57.2	83.8
CPC v2	ResNet-161(∗)	77.9	91.2
SimCLR (ours)	ResNet-50	75.5	87.8
SimCLR (ours)	ResNet-50 (2×)	83.0	91.2
SimCLR (ours)	ResNet-50 (4×)	**85.8**	**92.6**

Transfer learning for classification

"A Simple Framework for Contrastive Learning of Visual Representations", Chen et al., ICML 2020

AAAI 2022

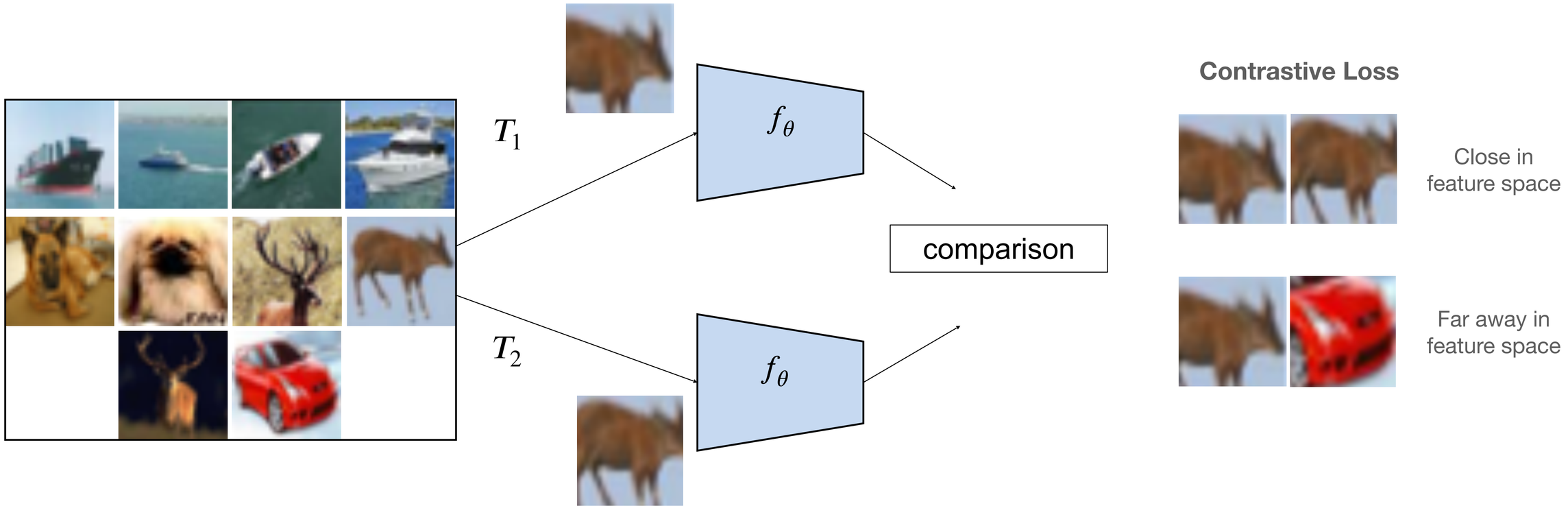
Rosa et Oliveira. *Learning from Label Proportions with Prototypical Contrastive Clustering*

Data Representation Learning + Learning from Label Proportions

Individual Crop Classification from Global Crop Area Estimate

How to find efficient crop data representation without labels?

- Recently, contrastive learning methods have closed the gap between supervised and unsupervised representation learning in many computer vision applications.
- Contrastive learning is a self-supervised representation learning approach that finds optimal data representations by enforcing consistency between different augmented views of the same input image via a contrastive loss function in the feature space.



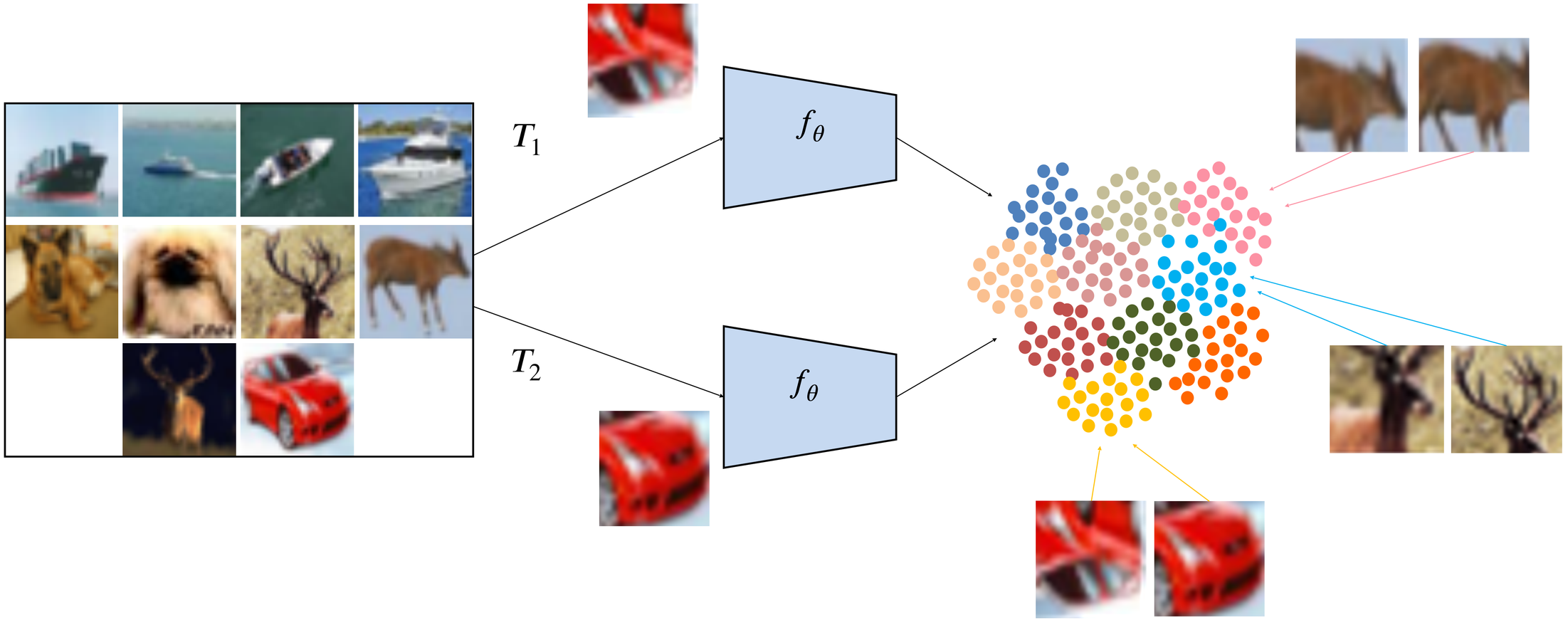
"Learning from Label Proportions with Prototypical Contrastive Clustering". Rosa et Oliveira, AAAI 2022.

Data Representation Learning + Learning from Label Proportions

Individual Crop Classification from Global Crop Area Estimate

How to support clustered data representation learning without labels?

- Recent works use optimal transport theory to compute pseudo labels for samples by assigning them to the most similar from a set of prototypes under an equipartition constraint.



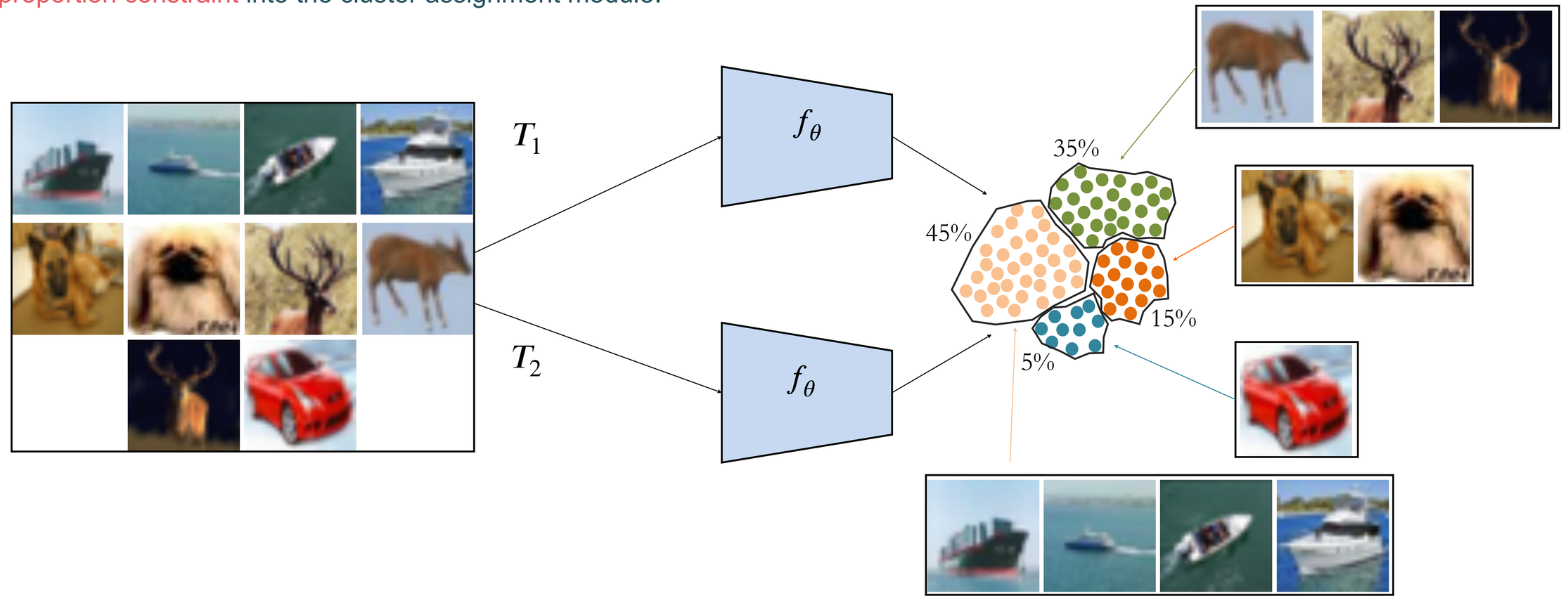
"Learning from Label Proportions with Prototypical Contrastive Clustering". Rosa et Oliveira, AAAI 2022.

Data Representation Learning + Learning from Label Proportions

Individual Crop Classification from Global Crop Area Estimate

How to support clustered data representation learning without labels?

- Our work combines prototypical contrastive learning and bag-level cluster proportions to implement efficient LLP classification.
- We propose a modified optimal transport algorithm that explicitly relaxes the equipartition constraint and embeds an exact cluster proportion constraint into the cluster assignment module.



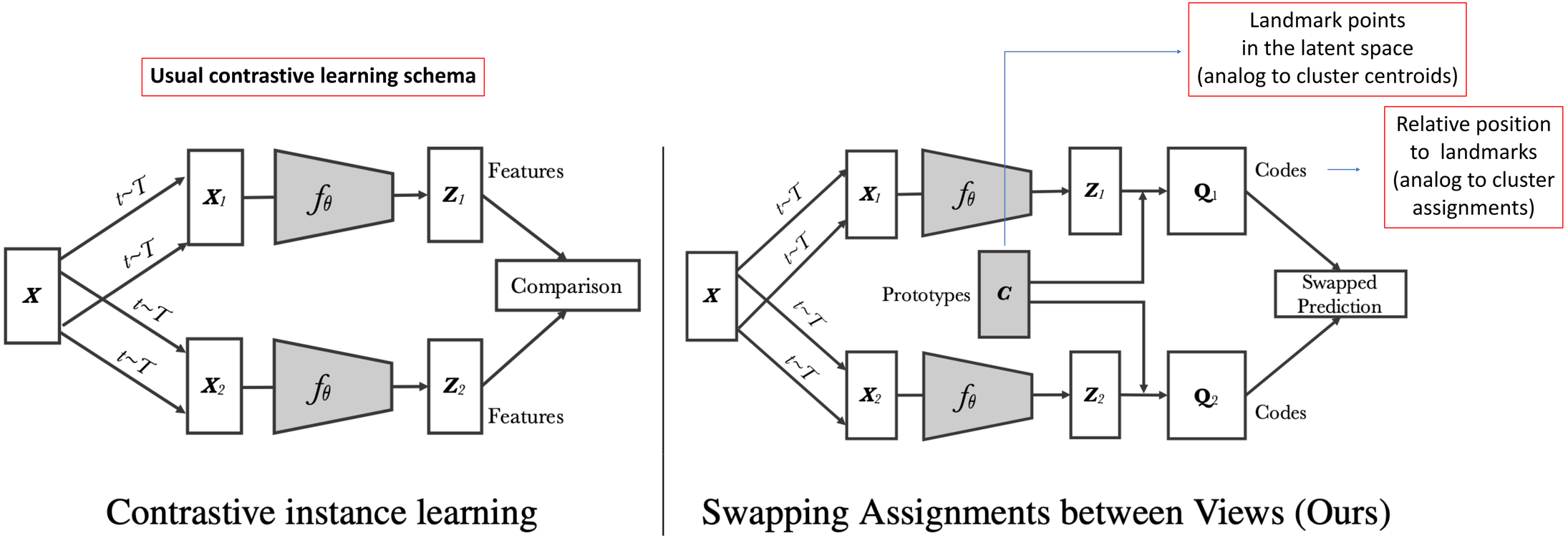
"Learning from Label Proportions with Prototypical Contrastive Clustering". Rosa et Oliveira, AAAI 2022.

Data Representation Learning + Learning from Label Proportions

Individual Crop Classification from Global Crop Area Estimate

SwAV: Unsupervised Learning of Visual Features by Contrasting Cluster Assignments

Learning by contrasting cluster assignments of multiple *image views* instead of image features

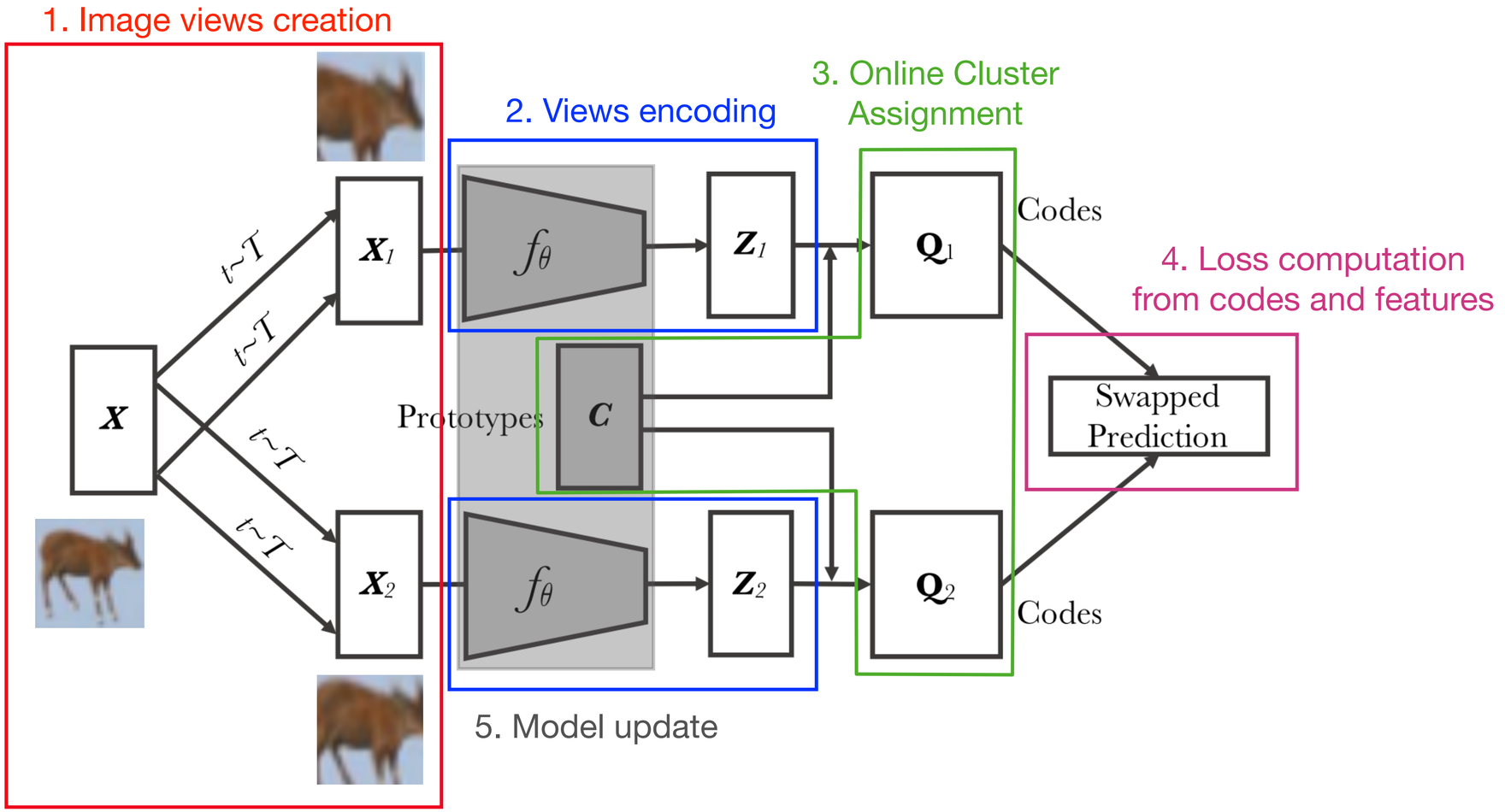


"Unsupervised Learning of Visual Features by Contrasting Cluster Assignments", Caron et al., NeurIPS 2020

Data Representation Learning + Learning from Label Proportions

Individual Crop Classification from Global Crop Area Estimate

SwaV overall optimization procedure consists of 5 steps



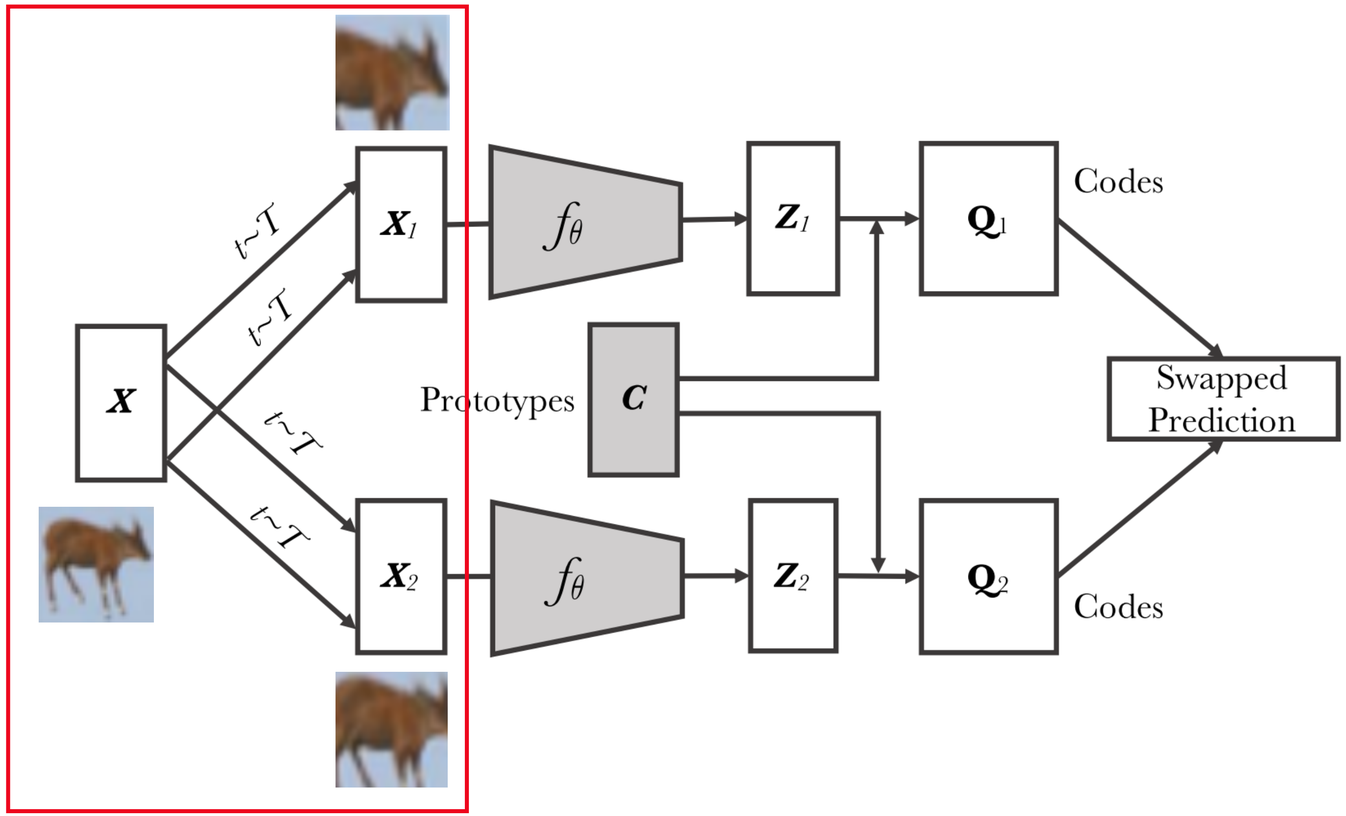
"Learning from Label Proportions with Prototypical Contrastive Clustering". Rosa et Oliveira, AAAI 2022.

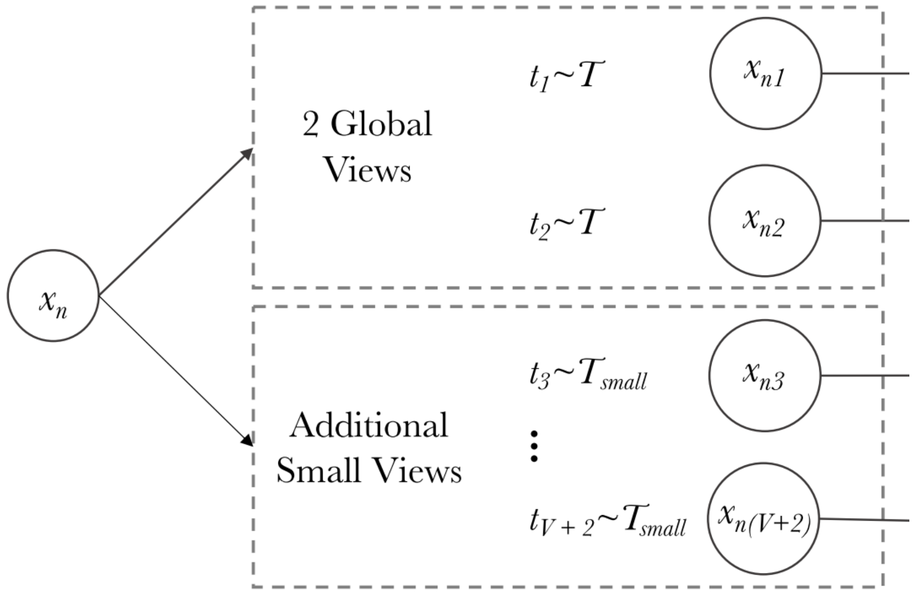
Data Representation Learning + Learning from Label Proportions

Individual Crop Classification from Global Crop Area Estimate

(1) Self-supervised strategy: strong data augmentation

We draw functions from a set of transformations and create *image views* (modified versions of given input images)





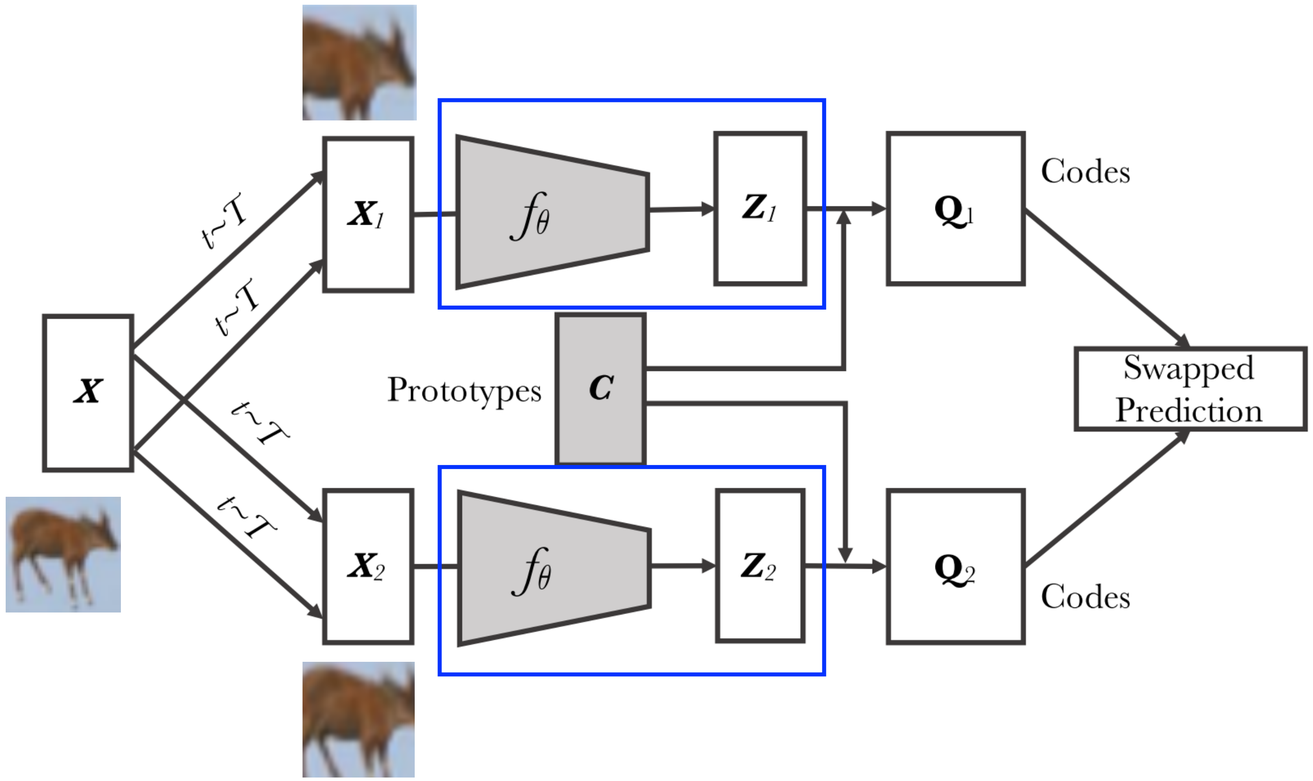
"Learning from Label Proportions with Prototypical Contrastive Clustering". Rosa et Oliveira, AAAI 2022.

Data Representation Learning + Learning from Label Proportions

Individual Crop Classification from Global Crop Area Estimate

(2) Feature extraction and projection to the unit sphere with L2-norm

The encoder computes features for the *image views* generated before



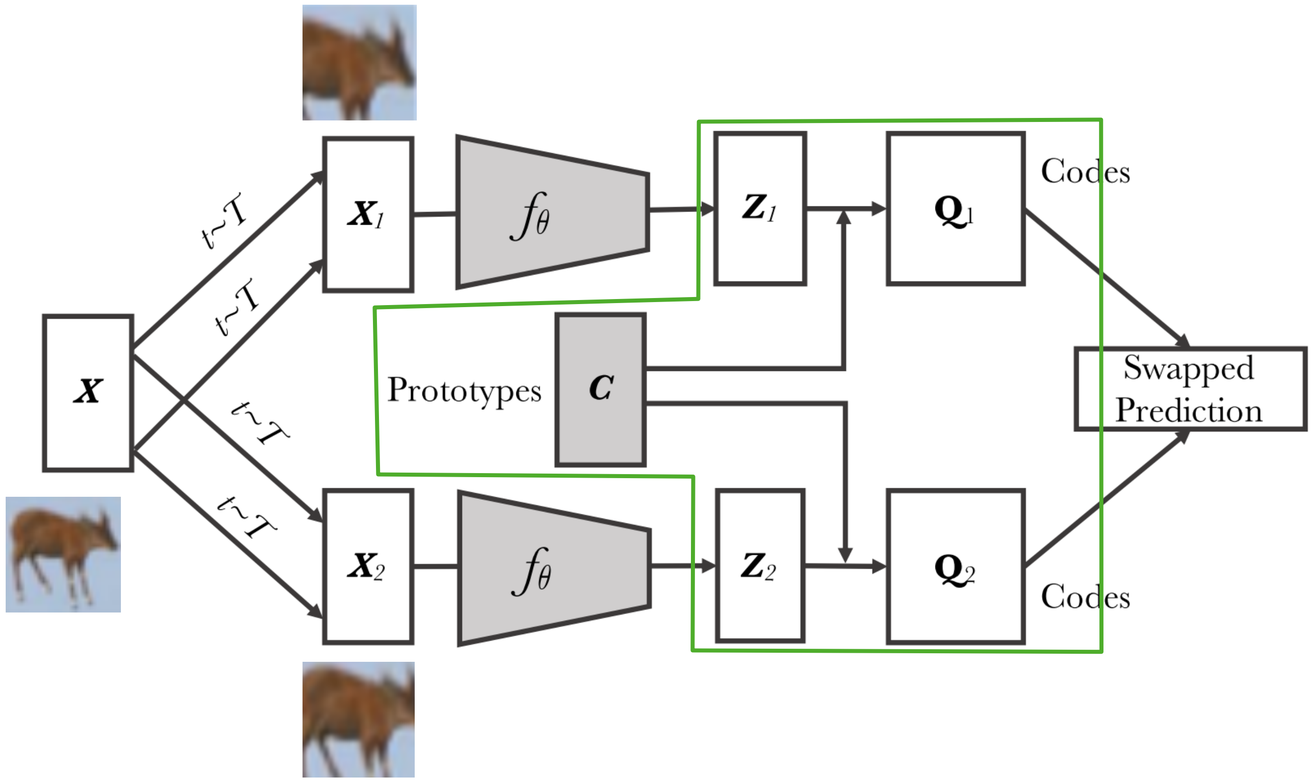
"Learning from Label Proportions with Prototypical Contrastive Clustering". Rosa et Oliveira, AAAI 2022.

Data Representation Learning + Learning from Label Proportions

Individual Crop Classification from Global Crop Area Estimate

(3) Prototype assignment using optimal transport

OT assigns features (samples) to prototypes (cluster centroids) with **equipartition** and **entropic** constraints



$\mathbf{Q}$ assigns B samples [$\mathbf{z}1$, . . . , $\mathbf{z}B$] to K (prototype) clusters [$\mathbf{c}1$, . . . , $\mathbf{c}K$]

$$\mathbf{Q} = [\mathbf{q}1, ..., \mathbf{q}B] \in R^{K \times B}$$

codes

SwAV solves the online cluster assignment using the entropic regularized OT where samples are equally divided among clusters.

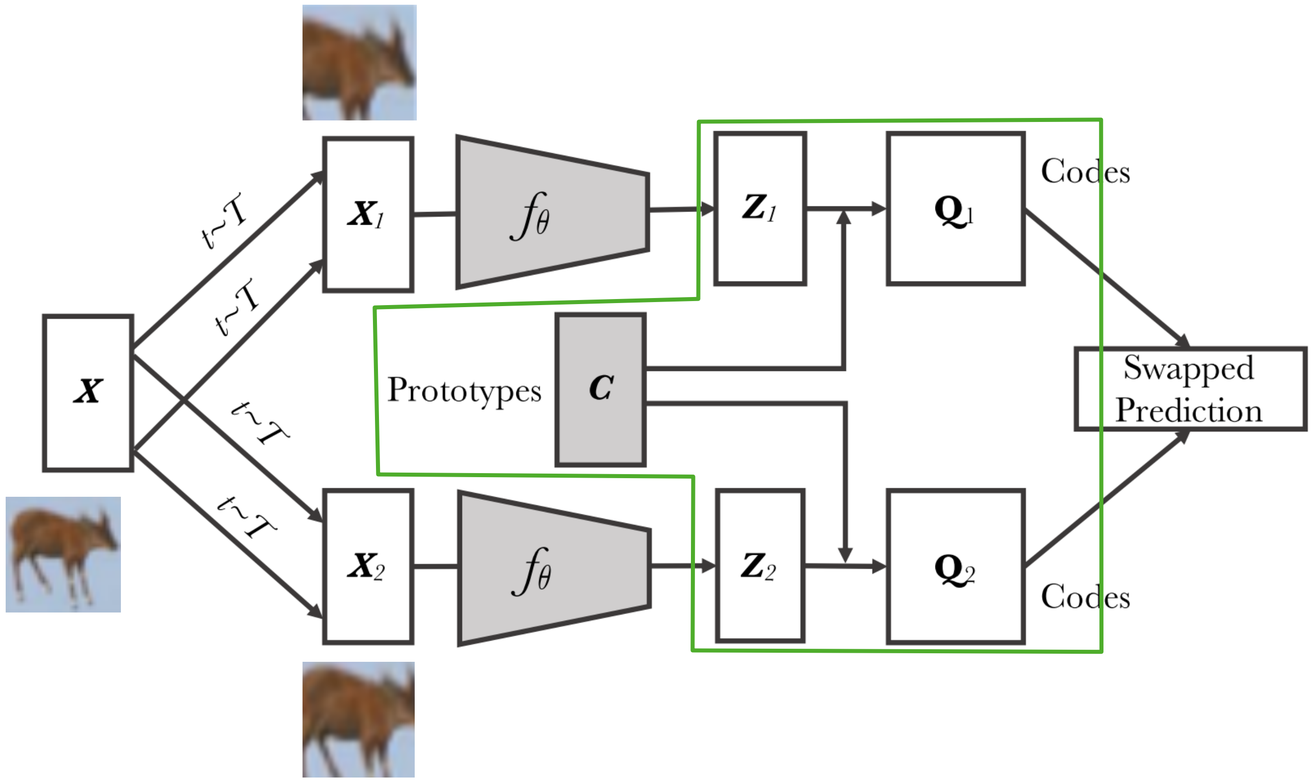
"Unsupervised Learning of Visual Features by Contrasting Cluster Assignments", Caron et al., NeurIPS 2020

Data Representation Learning + Learning from Label Proportions

Individual Crop Classification from Global Crop Area Estimate

(3) Prototype assignment using optimal transport

OT assigns features (samples) to prototypes (cluster centroids) with **equipartition** and **entropic** constraints



$\mathbf{Q}$ assigns B samples [$\mathbf{z}1$, . . . , $\mathbf{z}B$] to K (prototype) clusters [$\mathbf{c}1$, . . . , $\mathbf{c}K$]

$$\mathbf{Q} = [\mathbf{q}1, ..., \mathbf{q}B] \in R^{K \times B}$$

$$\mathbf{Q}^* = \max_{\mathbf{Q} \in \mathcal{Q}} \operatorname{Tr}\left(\mathbf{Q}^\top \mathbf{C}^\top \mathbf{Z}\right) + \varepsilon H(\mathbf{Q})$$

$$\mathcal{Q} = \left\{ \mathbf{Q} \in \mathbb{R}_+^{K \times B} \mid \mathbf{Q}\mathbf{1}_B = \frac{1}{K}\mathbf{1}_K,\ \mathbf{Q}^\top \mathbf{1}_K = \frac{1}{B}\mathbf{1}_B \right\}$$

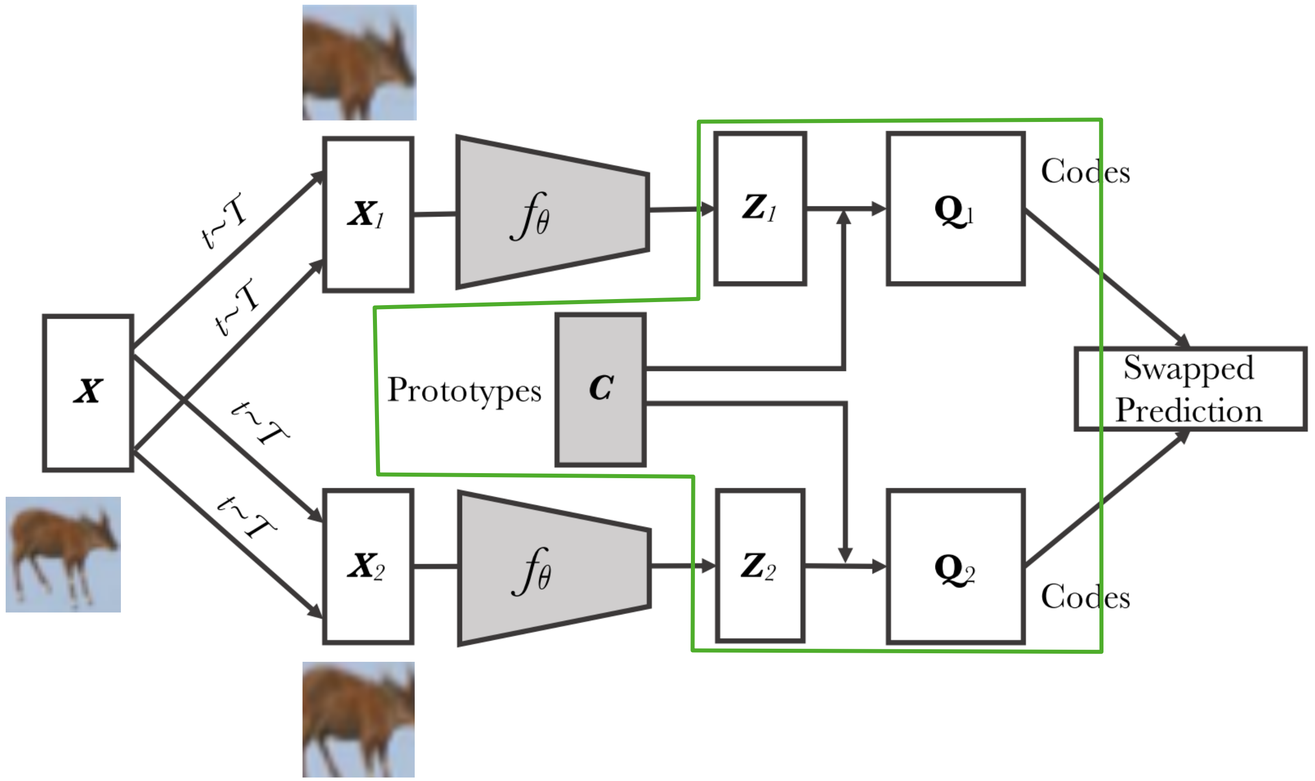
"Unsupervised Learning of Visual Features by Contrasting Cluster Assignments", Caron et al., NeurIPS 2020

Data Representation Learning + Learning from Label Proportions

Individual Crop Classification from Global Crop Area Estimate

(3) Prototype assignment using optimal transport

OT assigns features (samples) to prototypes (cluster centroids) with **equipartition** and **entropic** constraints



$\mathbf{Q}$ assigns B samples [$\mathbf{z}1$, . . . , $\mathbf{z}B$] to K (prototype) clusters [$\mathbf{c}1$, . . . , $\mathbf{c}K$]

$$\mathbf{Q} = [\mathbf{q}1, ..., \mathbf{q}B] \in R^{K \times B}$$

feature vectors that we want to map to the prototypes

$$\mathbf{Q}^* = \max_{\mathbf{Q} \in \mathcal{Q}} \operatorname{Tr}\left(\mathbf{Q}^\top \mathbf{C}^\top \mathbf{Z}\right) + \varepsilon H(\mathbf{Q})$$

$$\mathcal{Q} = \left\{ \mathbf{Q} \in \mathbb{R}_+^{K \times B} \mid \mathbf{Q}\mathbf{1}_B = \frac{1}{K}\mathbf{1}_K,\ \mathbf{Q}^\top \mathbf{1}_K = \frac{1}{B}\mathbf{1}_B \right\}$$

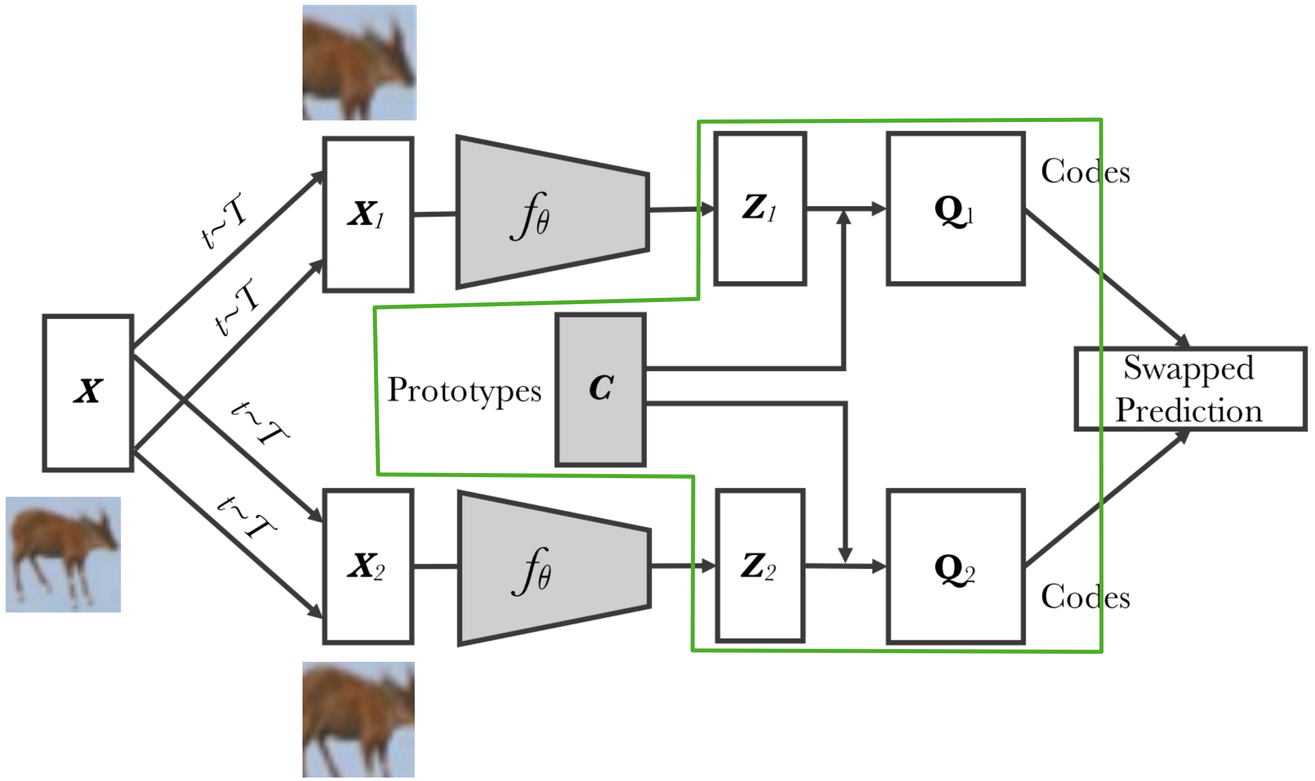
"Learning from Label Proportions with Prototypical Contrastive Clustering". Rosa et Oliveira, AAAI 2022.

Data Representation Learning + Learning from Label Proportions

Individual Crop Classification from Global Crop Area Estimate

(3) Prototype assignment using optimal transport

OT assigns features (samples) to prototypes (cluster centroids) with **equipartition** and **entropic** constraints



$\mathbf{Q}$ assigns B samples [$\mathbf{z}1$, . . . , $\mathbf{z}B$] to K (prototype) clusters [$\mathbf{c}1$, . . . , $\mathbf{c}K$]

$$\mathbf{Q} = [\mathbf{q}1, ..., \mathbf{q}B] \in R^{K \times B}$$

prototypes vectors

$$\mathbf{Q}^* = \max_{\mathbf{Q} \in \mathcal{Q}} \operatorname{Tr}\left(\mathbf{Q}^\top \mathbf{C}^\top \mathbf{Z}\right) + \varepsilon H(\mathbf{Q})$$

$$\mathcal{Q} = \left\{ \mathbf{Q} \in \mathbb{R}_+^{K \times B} \mid \mathbf{Q}\mathbf{1}_B = \frac{1}{K}\mathbf{1}_K,\ \mathbf{Q}^\top \mathbf{1}_K = \frac{1}{B}\mathbf{1}_B \right\}$$

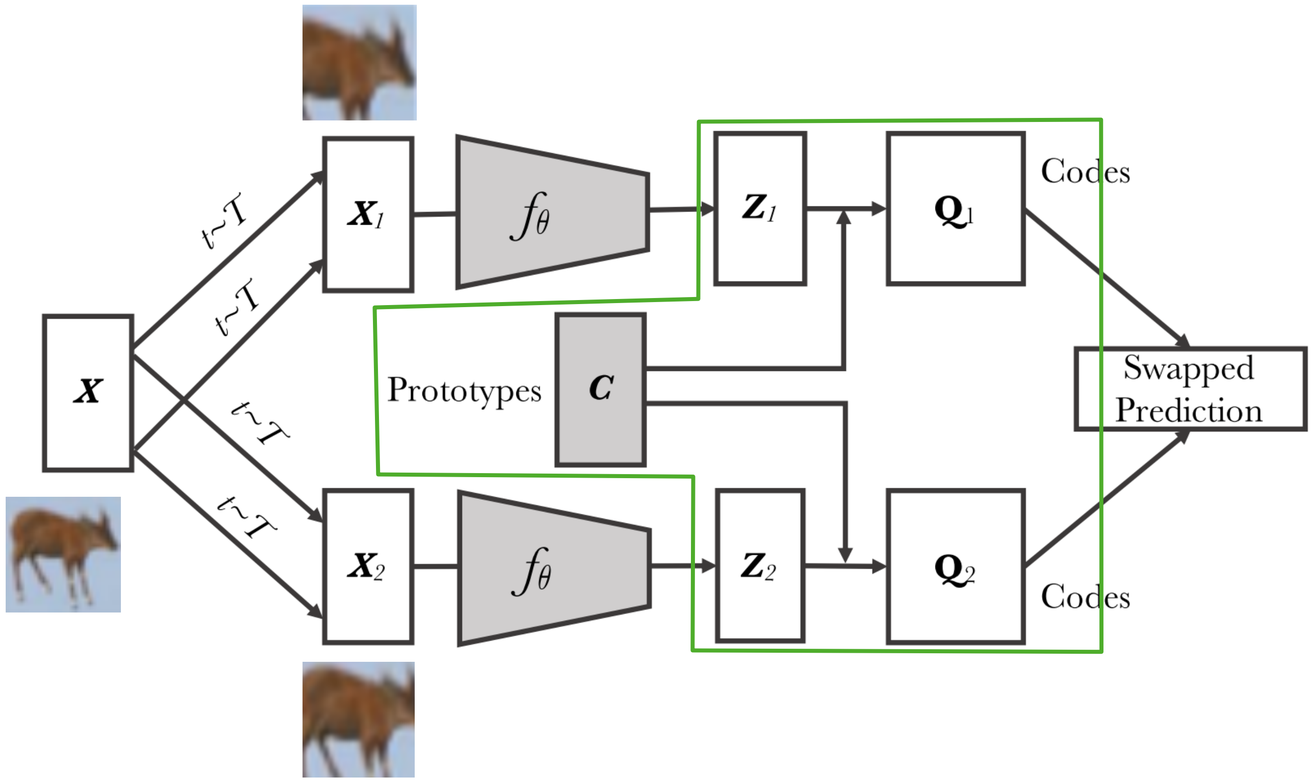
"Learning from Label Proportions with Prototypical Contrastive Clustering". Rosa et Oliveira, AAAI 2022.

Data Representation Learning + Learning from Label Proportions

Individual Crop Classification from Global Crop Area Estimate

(3) Prototype assignment using optimal transport

OT assigns features (samples) to prototypes (cluster centroids) with **equipartition** and **entropic** constraints



$\mathbf{Q}$ assigns B samples [$\mathbf{z}1$, . . . , $\mathbf{z}B$] to K (prototype) clusters [$\mathbf{c}1$, . . . , $\mathbf{c}K$]

$$\mathbf{Q} = [\mathbf{q}1, ..., \mathbf{q}B] \in R^{K \times B}$$

K x B matrix with cosine similarities between all given samples and each prototype

$$\mathbf{Q}^* = \max_{\mathbf{Q} \in \mathcal{Q}} \operatorname{Tr}\left(\mathbf{Q}^\top \mathbf{C}^\top \mathbf{Z}\right) + \varepsilon H(\mathbf{Q})$$

$$\mathcal{Q} = \left\{ \mathbf{Q} \in \mathbb{R}_+^{K \times B} \mid \mathbf{Q}\mathbf{1}_B = \frac{1}{K}\mathbf{1}_K,\ \mathbf{Q}^\top \mathbf{1}_K = \frac{1}{B}\mathbf{1}_B \right\}$$

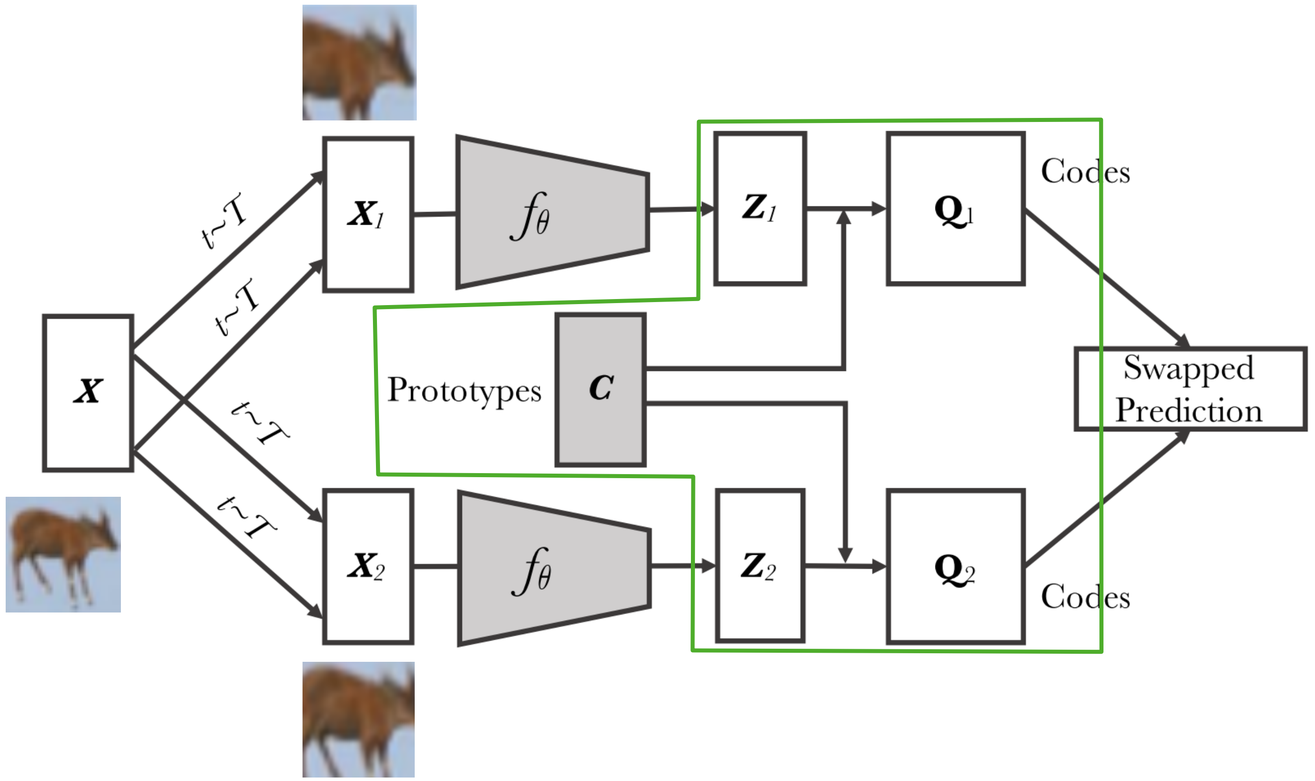
"Learning from Label Proportions with Prototypical Contrastive Clustering". Rosa et Oliveira, AAAI 2022.

Data Representation Learning + Learning from Label Proportions

Individual Crop Classification from Global Crop Area Estimate

(3) Prototype assignment using optimal transport

OT assigns features (samples) to prototypes (cluster centroids) with **equipartition** and **entropic** constraints



$\mathbf{Q}$ assigns B samples [$\mathbf{z}1$, . . . , $\mathbf{z}B$] to K (prototype) clusters [$\mathbf{c}1$, . . . , $\mathbf{c}K$]

$$\mathbf{Q} = [\mathbf{q}1, ..., \mathbf{q}B] \in R^{K \times B}$$

codes that perform the transportation, i.e, weights the similarities, restricted to the equipartition constraint

$$\mathbf{Q}^* = \max_{\mathbf{Q} \in \mathcal{Q}} \mathrm{Tr}\left(\mathbf{Q}^\top \mathbf{C}^\top \mathbf{Z}\right) + \varepsilon H(\mathbf{Q})$$

$$\mathcal{Q} = \left\{ \mathbf{Q} \in \mathbb{R}_+^{K \times B} \mid \mathbf{Q}\mathbf{1}_B = \frac{1}{K}\mathbf{1}_K,\ \mathbf{Q}^\top \mathbf{1}_K = \frac{1}{B}\mathbf{1}_B \right\}$$

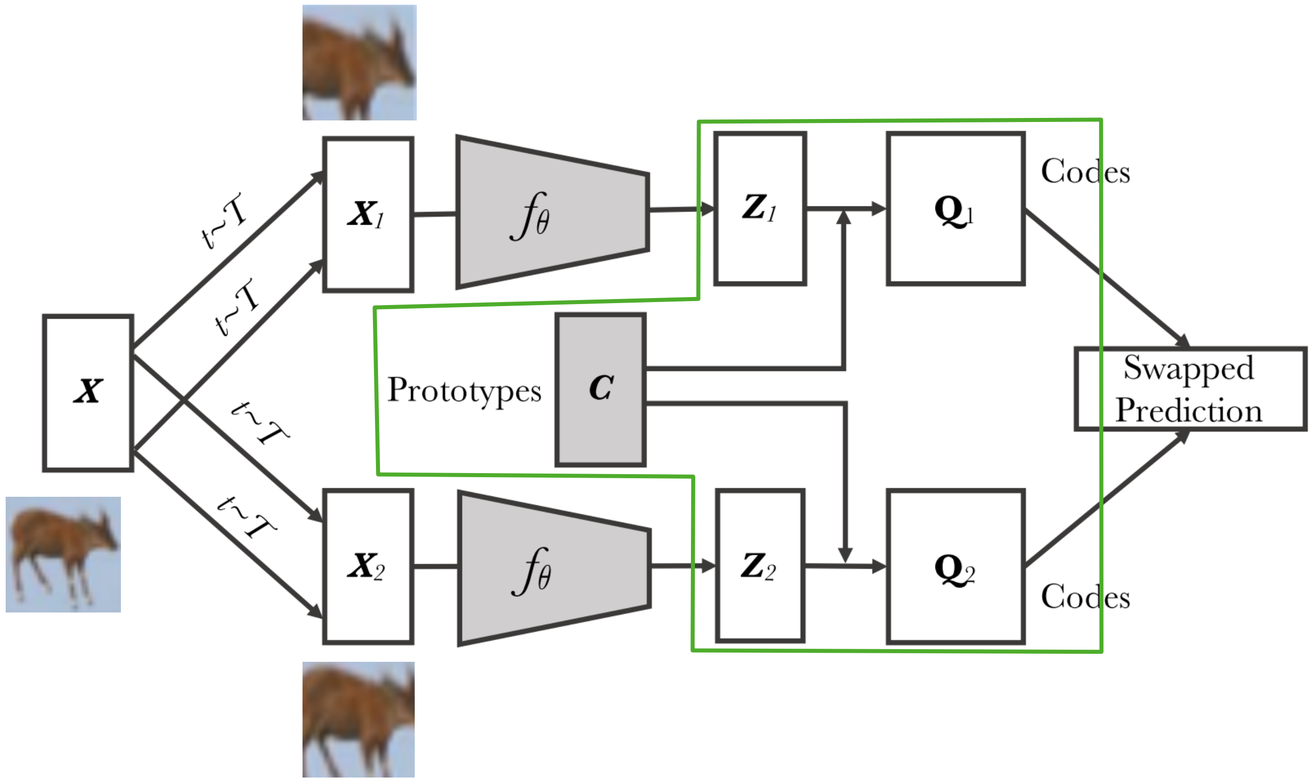
"Learning from Label Proportions with Prototypical Contrastive Clustering". Rosa et Oliveira, AAAI 2022.

Data Representation Learning + Learning from Label Proportions

Individual Crop Classification from Global Crop Area Estimate

(3) Prototype assignment using optimal transport

OT assigns features (samples) to prototypes (cluster centroids) with **equipartition** and **entropic** constraints



$\mathbf{Q}$ assigns B samples [$\mathbf{z}1$, . . . , $\mathbf{z}B$] to K (prototype) clusters [$\mathbf{c}1$, . . . , $\mathbf{c}K$]

$$\mathbf{Q} = [\mathbf{q}1, ..., \mathbf{q}B] \in R^{K \times B}$$

$$\mathbf{Q}^* = \max_{\mathbf{Q} \in \mathcal{Q}} \operatorname{Tr}\left(\mathbf{Q}^\top \mathbf{C}^\top \mathbf{Z}\right) + \varepsilon H(\mathbf{Q})$$

$$\mathcal{Q} = \left\{ \mathbf{Q} \in \mathbb{R}_+^{K \times B} \mid \mathbf{Q}\mathbf{1}_B = \frac{1}{K}\mathbf{1}_K,\ \mathbf{Q}^\top \mathbf{1}_K = \frac{1}{B}\mathbf{1}_B \right\}$$

equipartition constraint

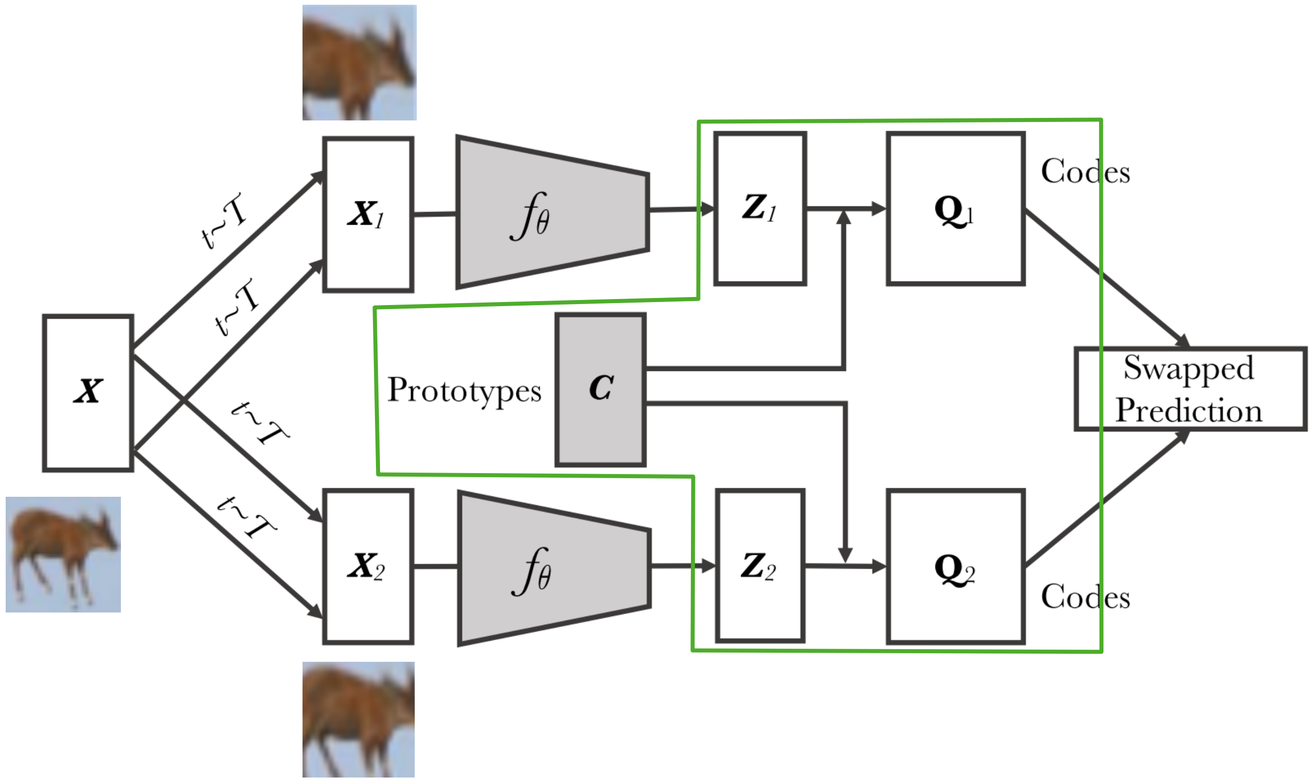
"Unsupervised Learning of Visual Features by Contrasting Cluster Assignments", Caron et al., NeurIPS 2020

Data Representation Learning + Learning from Label Proportions

Individual Crop Classification from Global Crop Area Estimate

(3) Prototype assignment using optimal transport

OT assigns features (samples) to prototypes (cluster centroids) with **equipartition** and **entropic** constraints



$\mathbf{Q}$ assigns B samples [$\mathbf{z}1$, . . . , $\mathbf{z}B$] to K (prototype) clusters [$\mathbf{c}1$, . . . , $\mathbf{c}K$]

$$\mathbf{Q} = [\mathbf{q}1, ..., \mathbf{q}B] \in R^{K \times B}$$

Max Tr would assign large weights to larger similarity scores

$$\mathbf{Q}^* = \max_{\mathbf{Q} \in \mathcal{Q}} \mathrm{Tr}\left(\mathbf{Q}^\top \mathbf{C}^\top \mathbf{Z}\right) + \varepsilon H(\mathbf{Q})$$

$$\mathcal{Q} = \left\{ \mathbf{Q} \in \mathbb{R}_+^{K \times B} \mid \mathbf{Q}\mathbf{1}_B = \frac{1}{K}\mathbf{1}_K,\ \mathbf{Q}^\top \mathbf{1}_K = \frac{1}{B}\mathbf{1}_B \right\}$$

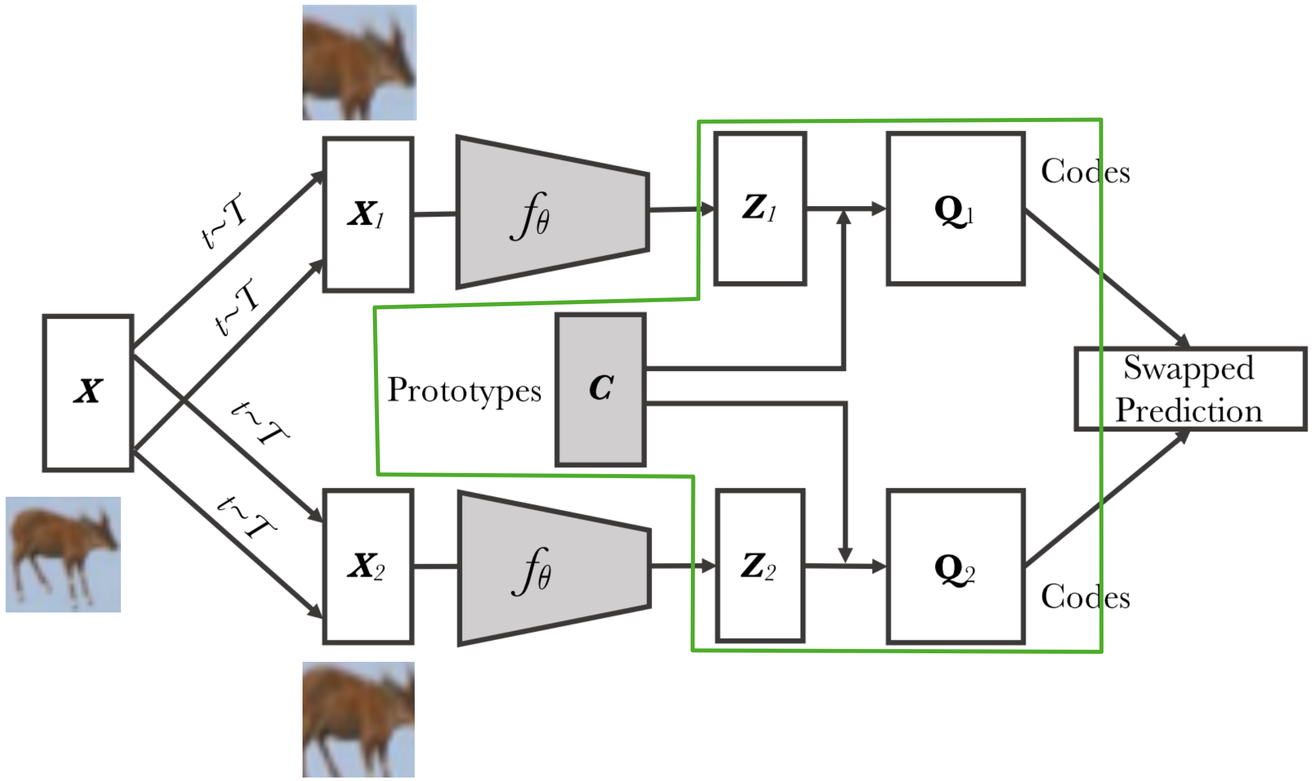
"Learning from Label Proportions with Prototypical Contrastive Clustering". Rosa et Oliveira, AAAI 2022.

Data Representation Learning + Learning from Label Proportions

Individual Crop Classification from Global Crop Area Estimate

(3) Prototype assignment using optimal transport

OT assigns features (samples) to prototypes (cluster centroids) with **equipartition** and **entropic** constraints



$\mathbf{Q}$ assigns B samples [$\mathbf{z}1$, . . . , $\mathbf{z}B$] to K (prototype) clusters [$\mathbf{c}1$, . . . , $\mathbf{c}K$]

$$\mathbf{Q} = [\mathbf{q}1, ..., \mathbf{q}B] \in R^{K \times B}$$

entropic constraint

$$\mathbf{Q}^* = \max_{\mathbf{Q} \in \mathcal{Q}} \operatorname{Tr}\left(\mathbf{Q}^\top \mathbf{C}^\top \mathbf{Z}\right) + \varepsilon H(\mathbf{Q})$$

$$\mathcal{Q} = \left\{ \mathbf{Q} \in \mathbb{R}_+^{K \times B} \mid \mathbf{Q}\mathbf{1}_B = \frac{1}{K}\mathbf{1}_K,\ \mathbf{Q}^\top \mathbf{1}_K = \frac{1}{B}\mathbf{1}_B \right\}$$

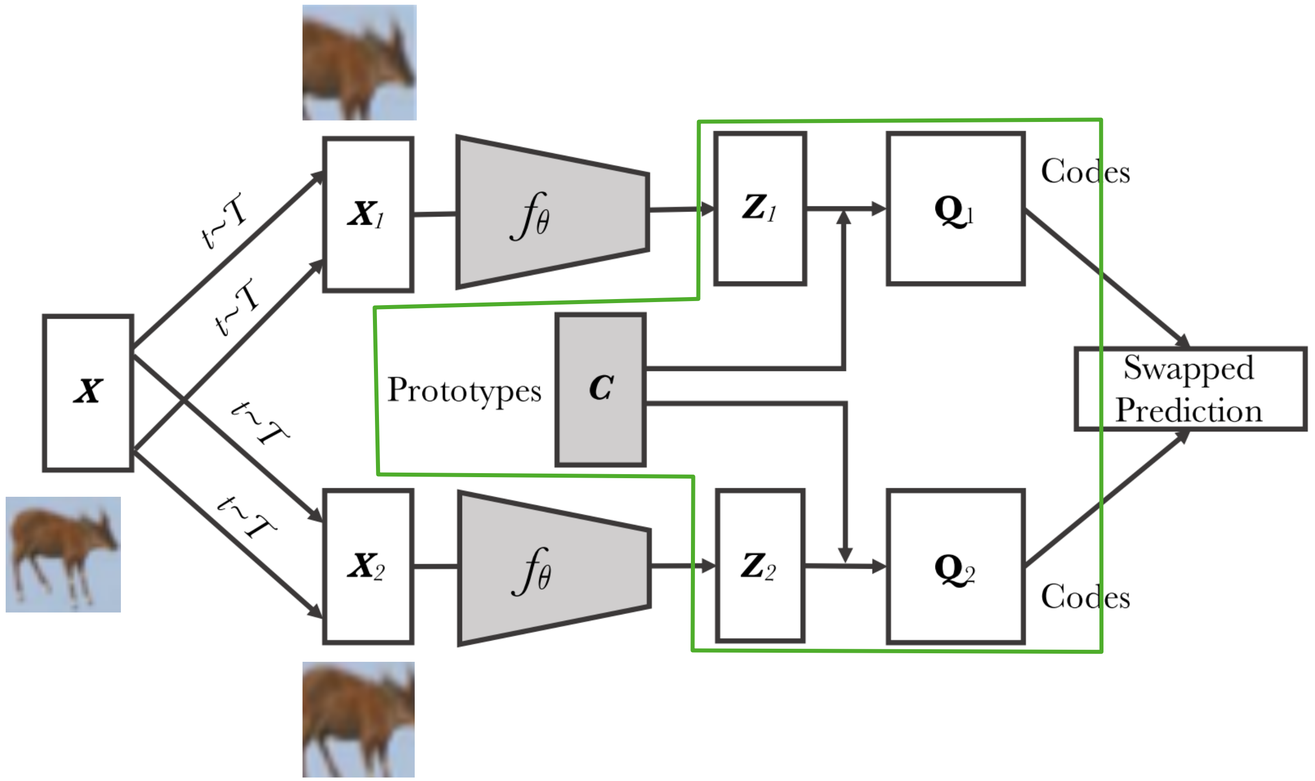
"Unsupervised Learning of Visual Features by Contrasting Cluster Assignments", Caron et al., NeurIPS 2020

Data Representation Learning + Learning from Label Proportions

Individual Crop Classification from Global Crop Area Estimate

(3) Prototype assignment using optimal transport

OT assigns features (samples) to prototypes (cluster centroids) with **equipartition** and **entropic** constraints



$\mathbf{Q}$ assigns B samples [$\mathbf{z}1$, . . . , $\mathbf{z}B$] to K (prototype) clusters [$\mathbf{c}1$, . . . , $\mathbf{c}K$]

$$\mathbf{Q} = [\mathbf{q}1, ..., \mathbf{q}B] \in R^{K \times B}$$

Max with all Q entries equal to 1/(BK)

$$\mathbf{Q}^* = \max_{\mathbf{Q} \in \mathcal{Q}} \operatorname{Tr}\left(\mathbf{Q}^\top \mathbf{C}^\top \mathbf{Z}\right) + \varepsilon H(\mathbf{Q})$$

$$\mathcal{Q} = \left\{ \mathbf{Q} \in \mathbb{R}_+^{K \times B} \mid \mathbf{Q}\mathbf{1}_B = \frac{1}{K}\mathbf{1}_K,\ \mathbf{Q}^\top \mathbf{1}_K = \frac{1}{B}\mathbf{1}_B \right\}$$

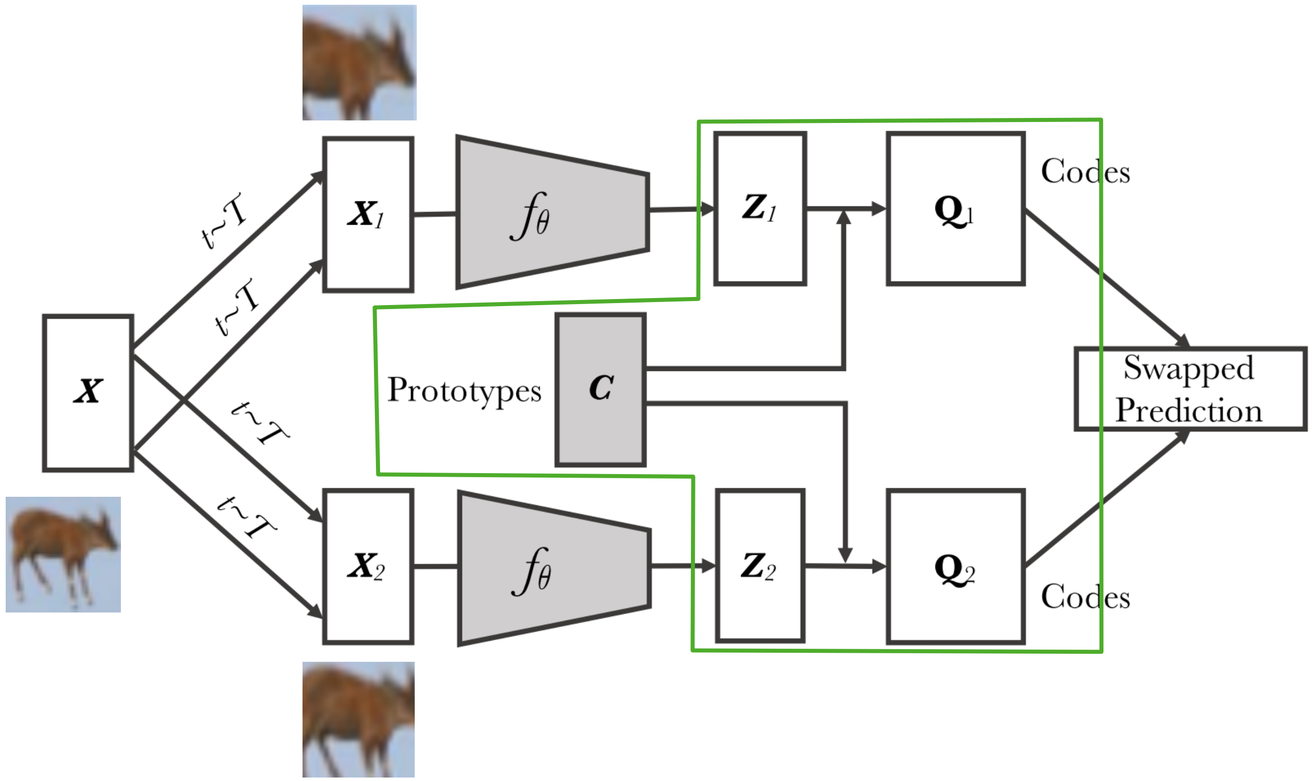
"Learning from Label Proportions with Prototypical Contrastive Clustering". Rosa et Oliveira, AAAI 2022.

Data Representation Learning + Learning from Label Proportions

Individual Crop Classification from Global Crop Area Estimate

(3) Prototype assignment using optimal transport

OT assigns features (samples) to prototypes (cluster centroids) with **equipartition** and **entropic** constraints



$\mathbf{Q}$ assigns B samples [$\mathbf{z}1$, . . . , $\mathbf{z}B$] to K (prototype) clusters [$\mathbf{c}1$, . . . , $\mathbf{c}K$]

$$\mathbf{Q} = [\mathbf{q}1, ..., \mathbf{q}B] \in R^{K \times B}$$

Parameter for smoothing the solution

$$\mathbf{Q}^* = \max_{\mathbf{Q} \in \mathcal{Q}} \operatorname{Tr}\left(\mathbf{Q}^\top \mathbf{C}^\top \mathbf{Z}\right) + \varepsilon H(\mathbf{Q})$$

$$\mathcal{Q} = \left\{ \mathbf{Q} \in \mathbb{R}_+^{K \times B} \mid \mathbf{Q}\mathbf{1}_B = \frac{1}{K}\mathbf{1}_K,\ \mathbf{Q}^\top \mathbf{1}_K = \frac{1}{B}\mathbf{1}_B \right\}$$

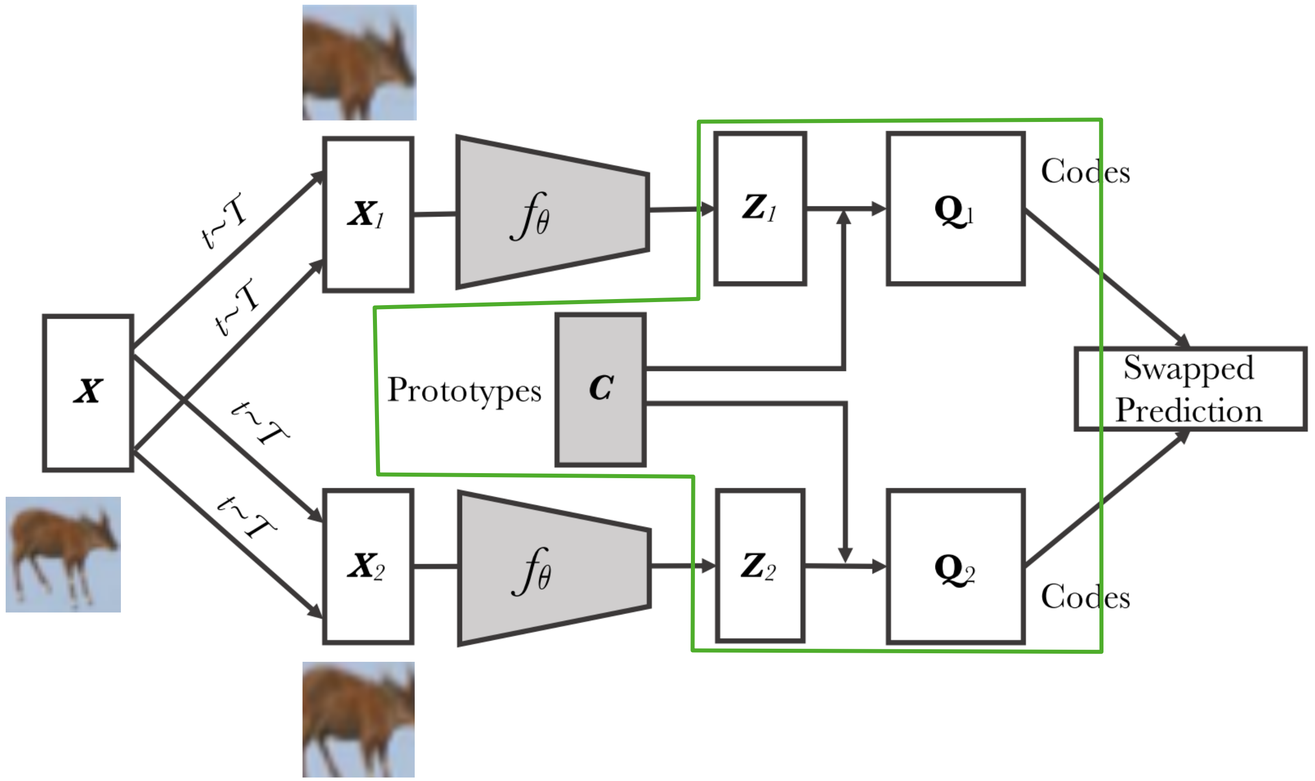
"Learning from Label Proportions with Prototypical Contrastive Clustering". Rosa et Oliveira, AAAI 2022.

Data Representation Learning + Learning from Label Proportions

Individual Crop Classification from Global Crop Area Estimate

(3) Prototype assignment using optimal transport with LLP constraint

OT assigns features (samples) to prototypes (cluster centroids) with **LLP** and **entropic** constraints



Approximate Solution: Sinkhorn-Knopp algorithm (Cuturi, Neurips 2013)

Lemma 2 (Cuturi, Neurips 2013)

Optimal Transport formulation

$$\mathbf{Q}^* = \max_{\mathbf{Q} \in \mathcal{Q}} \operatorname{Tr}\left(\mathbf{Q}^\top \mathbf{C}^\top \mathbf{Z}\right) + \varepsilon H(\mathbf{Q})$$

$$\mathbf{Q}^* = \operatorname{Diag}(\mathbf{u}) \exp\left(\frac{\mathbf{C}^T \mathbf{Z}}{\varepsilon}\right) \operatorname{Diag}(\mathbf{v})$$

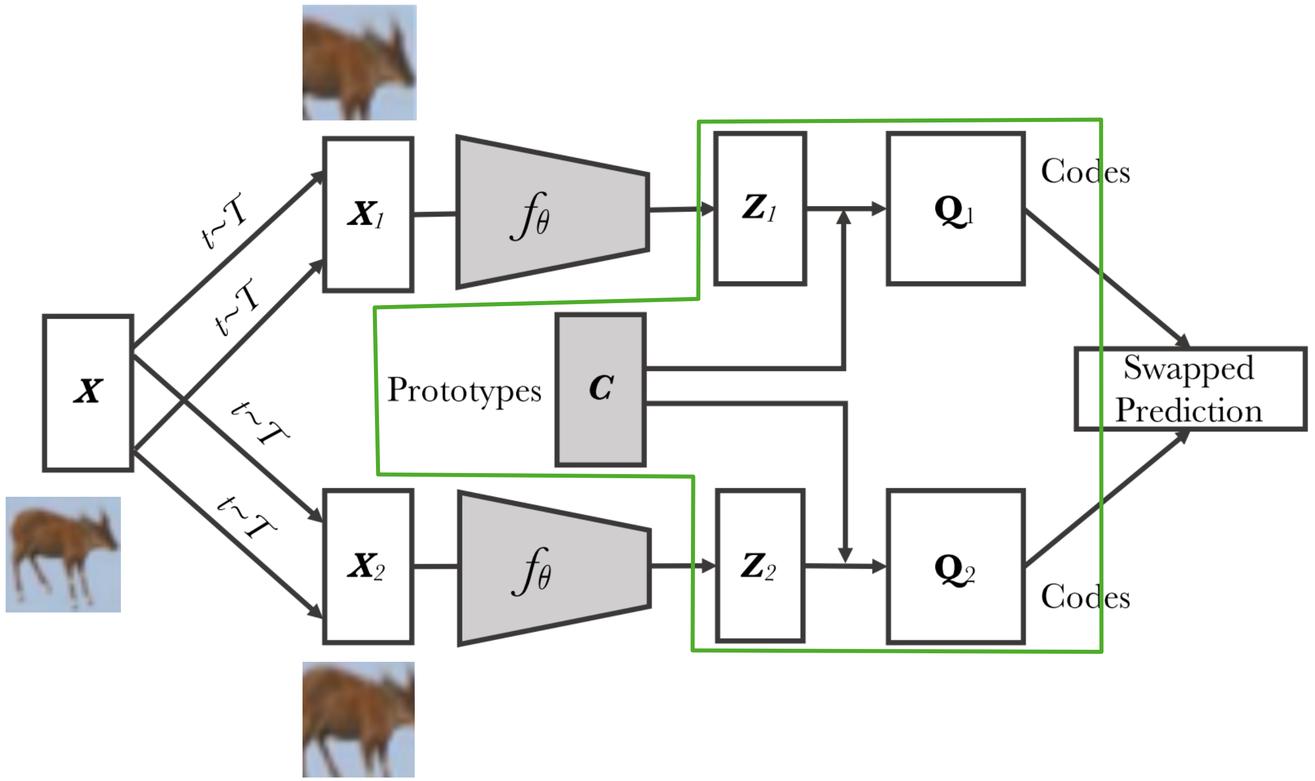
"Learning from Label Proportions with Prototypical Contrastive Clustering". Rosa et Oliveira, AAAI 2022.

Data Representation Learning + Learning from Label Proportions

Individual Crop Classification from Global Crop Area Estimate

(3) Prototype assignment using optimal transport with LLP constraint

OT assigns features (samples) to prototypes (cluster centroids) with **LLP** and **entropic** constraints



Approximate Solution: Sinkhorn-Knopp algorithm (Cuturi, Neurips 2013)

$$\mathbf{Q}^* = \underbrace{\mathrm{Diag}(\mathbf{u})}_{1} \exp\left(\frac{\mathbf{C}^T\mathbf{Z}}{\varepsilon}\right) \underbrace{\mathrm{Diag}(\mathbf{v})}_{2}$$

$$\mathbf{Q}\mathbf{1}_B = \frac{1}{K}\mathbf{1}_K$$

normalization

$$\mathbf{Q}^\top\mathbf{1}_K = \frac{1}{B}\mathbf{1}_B$$

normalization

Iteratively optimizes $\mathbf{Q} \in \mathcal{Q}$ so as to maximize the similarity between sample features and their respective assigned clusters while keeping its entropy relatively high

Experimentally three iterations are enough for convergence

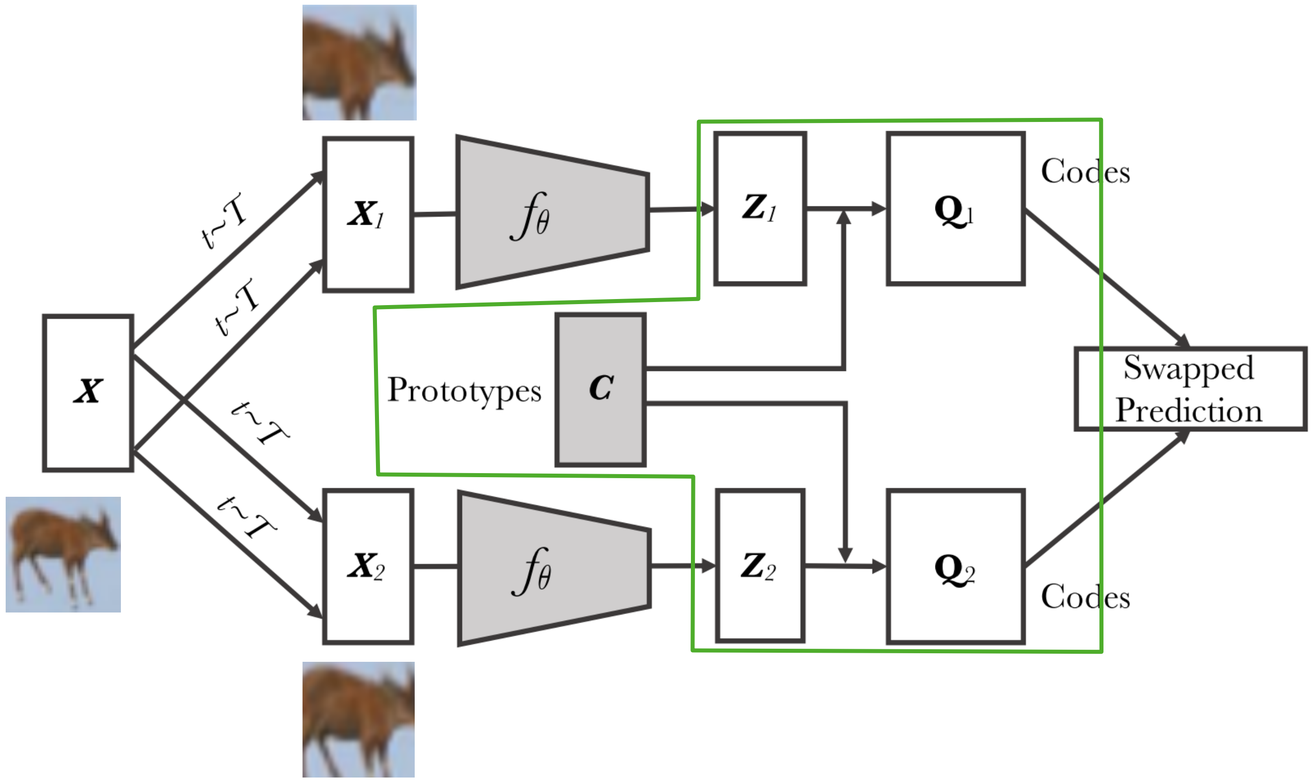
"Learning from Label Proportions with Prototypical Contrastive Clustering". Rosa et Oliveira, AAAI 2022.

Data Representation Learning + Learning from Label Proportions

Individual Crop Classification from Global Crop Area Estimate

(3) Prototype assignment using optimal transport

OT assigns features (samples) to prototypes (cluster centroids) with **equipartition** and **entropic** constraints



$\mathbf{Q}$ assigns B samples [$\mathbf{z}1$, . . . , $\mathbf{z}B$] to K (prototype) clusters [$\mathbf{c}1$, . . . , $\mathbf{c}K$]

$$\mathbf{Q} = [\mathbf{q}1, ..., \mathbf{q}B] \in R^{K \times B}$$

$$\mathbf{Q}^* = \max_{\mathbf{Q} \in \mathcal{Q}} \operatorname{Tr}\left(\mathbf{Q}^\top \mathbf{C}^\top \mathbf{Z}\right) + \varepsilon H(\mathbf{Q})$$

$$\mathcal{Q} = \left\{ \mathbf{Q} \in \mathbb{R}_+^{K \times B} \mid \mathbf{Q}\mathbf{1}_B = \frac{1}{K}\mathbf{1}_K,\ \mathbf{Q}^\top \mathbf{1}_K = \frac{1}{B}\mathbf{1}_B \right\}$$

equipartition constraint

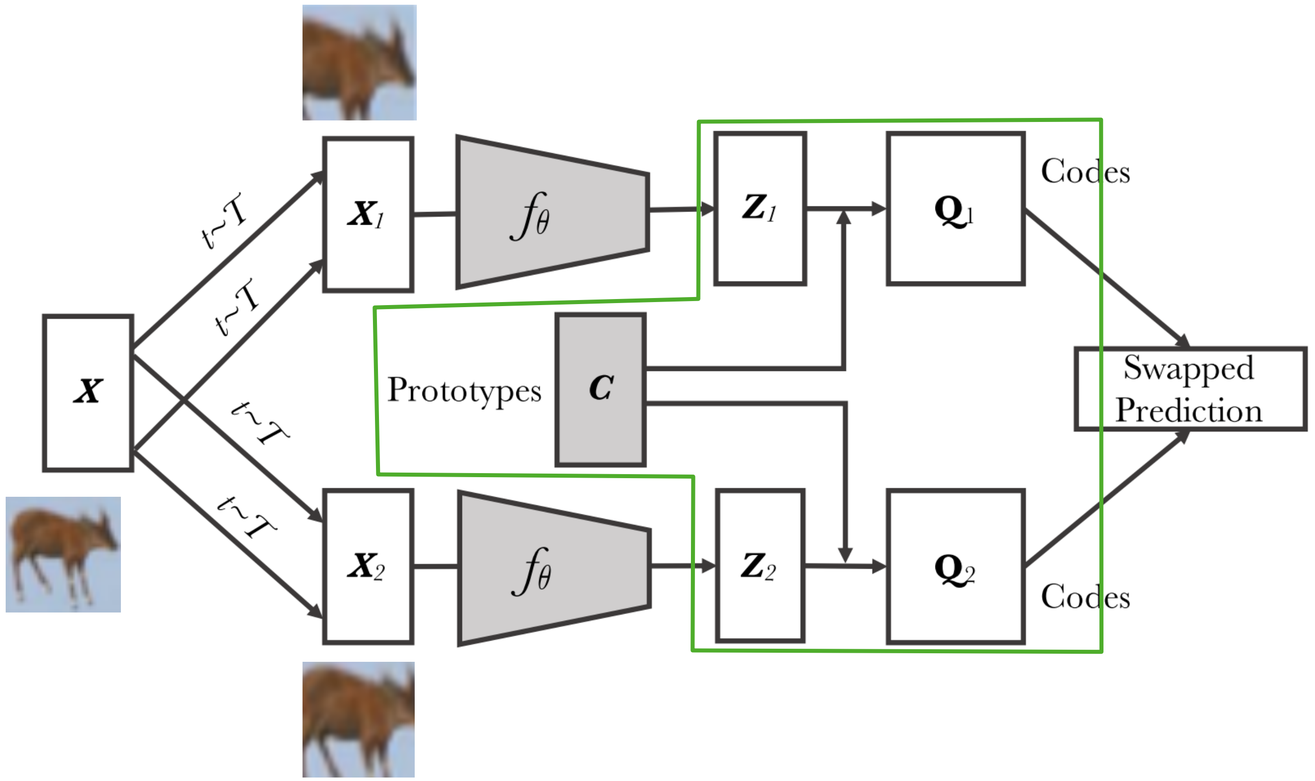
"Unsupervised Learning of Visual Features by Contrasting Cluster Assignments", Caron et al., NeurIPS 2020

Data Representation Learning + Learning from Label Proportions

Individual Crop Classification from Global Crop Area Estimate

(3) Prototype assignment using optimal transport with LLP constraint

OT assigns features (samples) to prototypes (cluster centroids) with **LLP** and **entropic** constraints



$\mathbf{Q}$ assigns B samples [$\mathbf{z}1$, . . . , $\mathbf{z}B$] to K (prototype) clusters [$\mathbf{c}1$, . . . , $\mathbf{c}K$]

$$\mathbf{Q} = [\mathbf{q}1, ..., \mathbf{q}B] \in R^{K \times B}$$

$$\mathbf{Q}^* = \max_{\mathbf{Q} \in \mathcal{Q}} \mathrm{Tr}\left(\mathbf{Q}^\top \mathbf{C}^\top \mathbf{Z}\right) + \varepsilon H(\mathbf{Q})$$

$$\mathcal{Q} = \left\{ \mathbf{Q} \in \mathbb{R}_+^{K \times B} \mid \mathbf{Q}\mathbf{1}_B = \boxed{\mathbf{W}} \, , \; \mathbf{Q}^\top \mathbf{1}_K = \frac{1}{B}\mathbf{1}_B \right\}$$

We solve the online cluster assignment using the entropic regularized OT where samples in the bag are partitioned according to bag-level cluster/label proportions $\mathbf{W}$

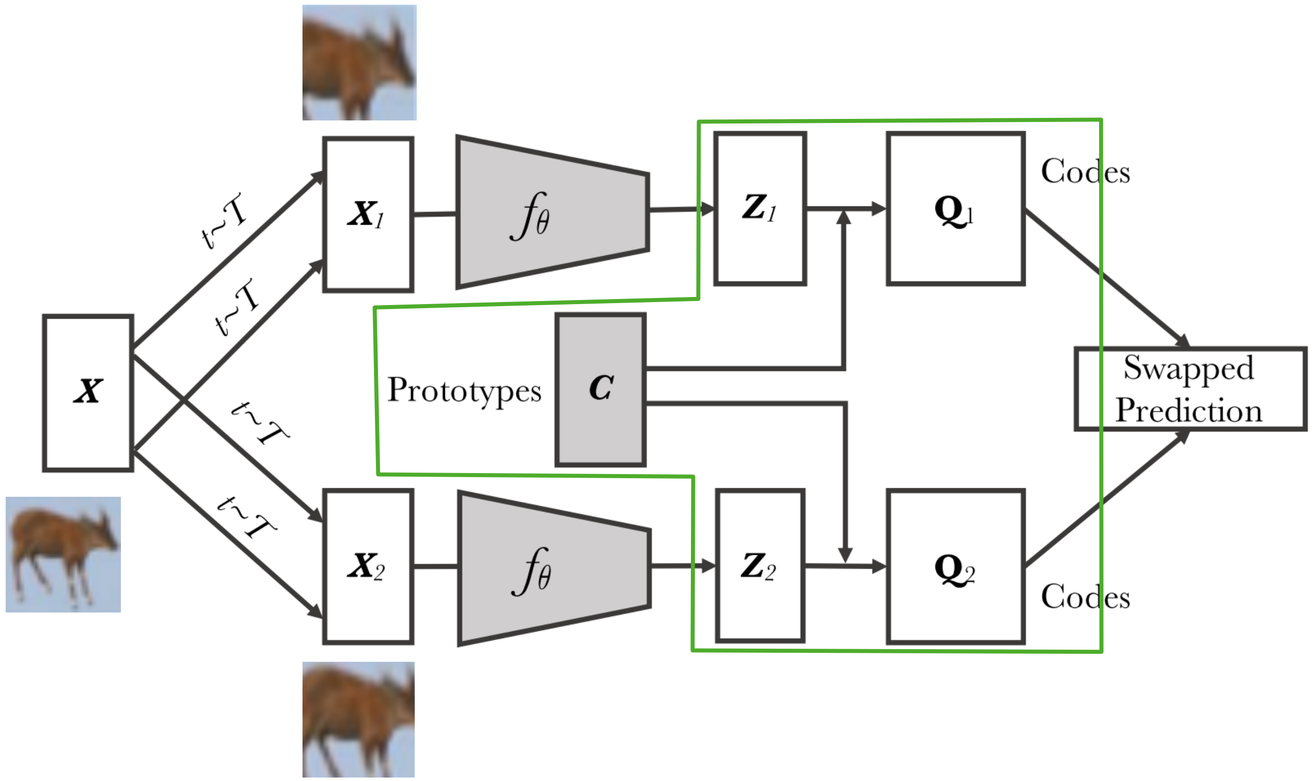
"Learning from Label Proportions with Prototypical Contrastive Clustering". Rosa et Oliveira, AAAI 2022.

Data Representation Learning + Learning from Label Proportions

Individual Crop Classification from Global Crop Area Estimate

(3) Prototype assignment using optimal transport with LLP constraint

OT assigns features (samples) to prototypes (cluster centroids) with **LLP** and **entropic** constraints



Approximate Solution: Sinkhorn-Knopp algorithm
(Cuturi, Neurips 2013)

$$\mathbf{Q}^* = \mathrm{Diag}(\mathbf{u}) \exp\left(\frac{\mathbf{C}^T\mathbf{Z}}{\varepsilon}\right) \mathrm{Diag}(\mathbf{v})$$

1 2

$\mathbf{Q}\mathbf{1}_B = \mathbf{W}$ normalization

$\mathbf{Q}^\top \mathbf{1}_K = \frac{1}{B}\mathbf{1}_B$ normalization

Iteratively optimizes $\mathbf{Q} \in \mathcal{Q}$ so as to maximize the similarity between sample features and their respective assigned clusters while keeping its entropy relatively high

Experimentally three iterations are enough for convergence

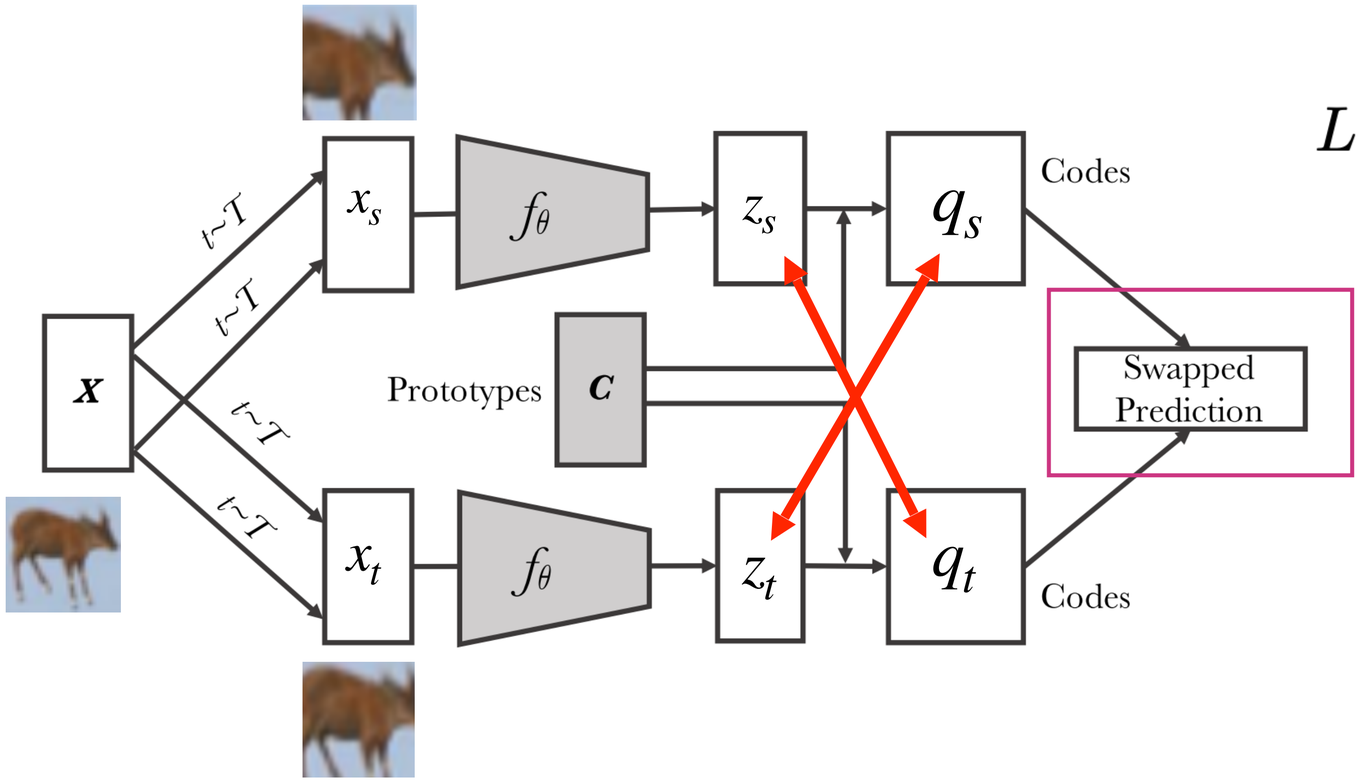
"Learning from Label Proportions with Prototypical Contrastive Clustering". Rosa et Oliveira, AAAI 2022.

Data Representation Learning + Learning from Label Proportions

Individual Crop Classification from Global Crop Area Estimate

(4) compute the swapped contrastive loss and update both **C** and **θ**

The contrastive loss predicts the assignment of one feature from the code of the other one.



swapped loss

$$L(\mathbf{z}_t, \mathbf{z}_s) = \ell(\mathbf{z}_t, \mathbf{q}_s) + \ell(\mathbf{z}_s, \mathbf{q}_t)$$

Intuition: If $\mathbf{z}_t$ and $\mathbf{z}_s$ capture similar information, we can predict the code $\mathbf{q}_s$ (soft class) from the other feature $\mathbf{z}_t$. In other words, **if the two views share the same semantics, their targets (codes) will be similar**. This is the whole "swapping" idea.

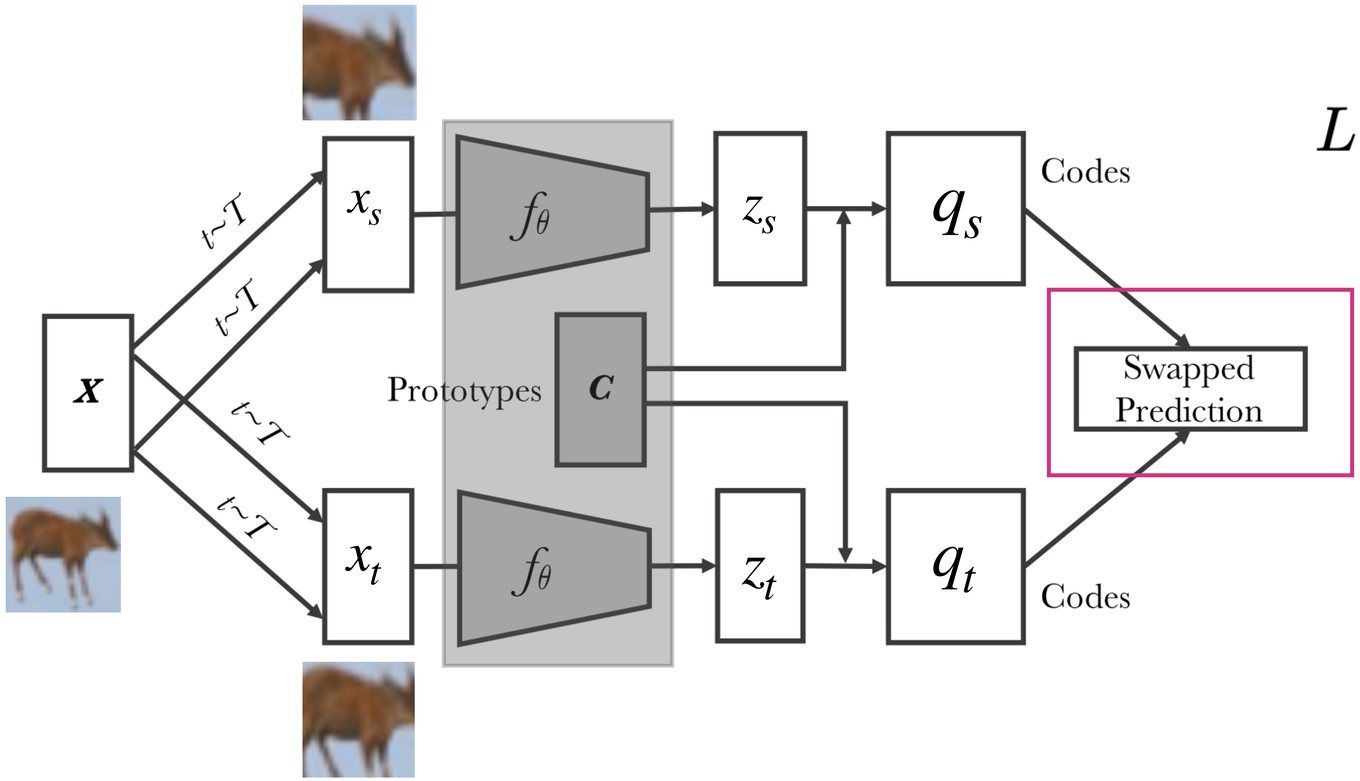
"Learning from Label Proportions with Prototypical Contrastive Clustering". Rosa et Oliveira, AAAI 2022.

Data Representation Learning + Learning from Label Proportions

Individual Crop Classification from Global Crop Area Estimate

(4) compute the swapped contrastive loss and update both **C** and **θ**

The contrastive loss predicts the assignment of one feature from the code of the other one.



$$L(\mathbf{z}_t, \mathbf{z}_s) = \ell(\mathbf{z}_t, \mathbf{q}_s) + \ell(\mathbf{z}_s, \mathbf{q}_t)$$

cross-entropy

$$\ell(\mathbf{z}_t, \mathbf{q}_s) = -\sum_k \mathbf{q}_s^{(k)} \log \mathbf{p}_t^{(k)}$$

"target" codes computed from OT

softmax between features z_t and each prototype

$$\mathbf{p}_t^{(k)} = \frac{\exp\left(\frac{1}{\tau}\mathbf{z}_t^\top \mathbf{c}_k\right)}{\sum_{k'} \exp\left(\frac{1}{\tau}\mathbf{z}_t^\top \mathbf{c}_{k'}\right)}$$

"Learning from Label Proportions with Prototypical Contrastive Clustering". Rosa et Oliveira, AAAI 2022.

Data Representation Learning + Learning from Label Proportions

Individual Crop Classification from Global Crop Area Estimate

Algorithm 1: LLP-Co training loop using two views

Input: bags samples and proportions $\mathcal{D} = \{(\mathcal{B}_i, \mathbf{w}_i)\}_{i=1}^{N}$
Input: $\varepsilon > 0$, epochs
Initialize: encoder network f_θ and prototypes $\mathbf{V}$ with random weights

for i = 1 to epochs **do**
  for each $\mathcal{B}_i$ in $\mathcal{D}$ **do**
    Generate two random views $\mathbf{X}_i^{t,s}$
    Obtain the feature vectors $\mathbf{Z}_i^{t,s}$
    Compute the prototype scores $\mathbf{V}^\top \mathbf{Z}_i^{t,s}$
    Compute the codes $\mathbf{Q}_i^{t,s}$ through Sinkhorn constrained to $\mathbf{w}_i$
    Convert prototype scores to probabilities $\mathbf{P}_i^{t,s}$
    Compute loss using the swap prediction problem: $\text{loss} = -0.5 * mean(\mathbf{Q}_i^t * log(\mathbf{P}_i^s) + \mathbf{Q}_i^s * log(\mathbf{P}_i^t)$
    Update θ and $\mathbf{V}$ with a gradient step
    Normalize prototypes $\mathbf{V}$
  end for
end for

"Learning from Label Proportions with Prototypical Contrastive Clustering". Rosa et Oliveira, AAAI 2022.

Data Representation Learning + Learning from Label Proportions

Individual Crop Classification from Global Crop Area Estimate

- **Datasets**: Two standard image classification benchmarks, CIFAR-10 and CIFAR-100.

- **Bag-level label proportions generation:** Random selecting samples from the training set, such as each sample within the bag is unique. We defined four experiments with different bag sizes = [16, 32, 64, 128].

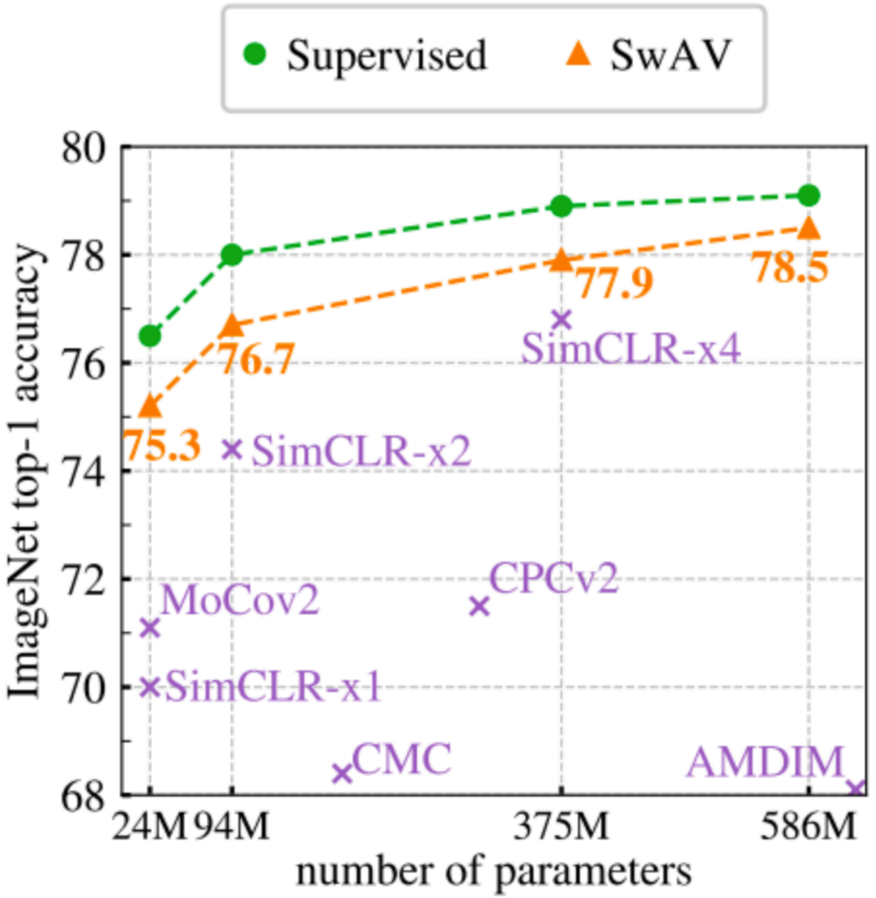
- **Architecture and training:** ResNet18 as backbone architecture followed by a projection head that projects the output feature to a 1024-D space. All the experimented models were trained using SGD. The input images size is 32 × 32, and we used the same augmentations strategy that SwAV to obtain four different image views.

- **Evaluation metrics:**
  - Acc_H that takes the cluster assignment as the prediction and computes the optimal matching between clusters and labels using the Hungarian algorithm
  - Acc_A, which takes the exact cluster assignment as the predicted label, i.e., if a test sample was assigned to the prototype v_1 the corresponding label will be 1.

"Learning from Label Proportions with Prototypical Contrastive Clustering". Rosa et Oliveira, AAAI 2022.

Data Representation Learning + Learning from Label Proportions

Individual Crop Classification from Global Crop Area Estimate



	Method	1% labels Top-1	1% labels Top-5	10% labels Top-1	10% labels Top-5
	Supervised	25.4	48.4	56.4	80.4
Methods using label-propagation	UDA [58]	-	-	68.8*	88.5*
	FixMatch [49]	-	-	**71.5***	89.1*
Methods using self-supervision only	PIRL [42]	30.7	57.2	60.4	83.8
	PCL [35]	-	75.6	-	86.2
	SimCLR [10]	48.3	75.5	65.6	87.8
	SwAV	**53.9**	**78.5**	70.2	**89.9**

Transfer learning for classification

"Unsupervised Learning of Visual Features by Contrasting Cluster Assignments", Caron et al., NeurIPS 2020

Data Representation Learning + Learning from Label Proportions

Individual Crop Classification from Global Crop Area Estimate

Table 1: Test accuracy rates (%) on CIFAR-10 and CIFAR-100 datasets with different bag sizes.

Dataset	LLP Methods	Bag Size				SwAV (kNN)	ConCURL	Supervised
		16	32	64	128			
CIFAR-10	DLLP-KL	86.0	72.0	56.0	41.0	80.0	84.6	93.6
	DLLP-ROT	78.0	65.0	53.0	40.0			
	LLP-GAN	86.3	83.8	79.0	72.6			
	LLP-GAN-PLOT	89.3	88.2	84.1	79.1			
	LLP-Co (Acc_H) (**ours**)	**90.0**	**89.8**	**90.9**	**86.2**			
	LLP-Co (Acc_A) (**ours**)	**90.0**	**89.8**	**90.9**	72.1			
CIFAR-100	DLLP-KL	58.0	38.0	24.0	14.0	45.8	47.9	78.3
	DLLP-ROT	51.0	37.0	24.0	14.0			
	LLP-GAN	49.1	43.6	35.6	15.0			
	LLP-GAN-PLOT	**65.4**	61.7	55.7	43.4			
	LLP-Co (Acc_H) (**ours**)	59.5	**65.9**	**66.5**	**64.7**			
	LLP-Co (Acc_A) (**ours**)	59.4	65.7	**66.5**	62.0			

"Learning from Label Proportions with Prototypical Contrastive Clustering". Rosa et Oliveira, AAAI 2022.

Data Representation Learning + Learning from Label Proportions

Individual Crop Classification from Global Crop Area Estimate

Table 1: Test accuracy rates (%) on CIFAR-10 and CIFAR-100 datasets with different bag sizes.

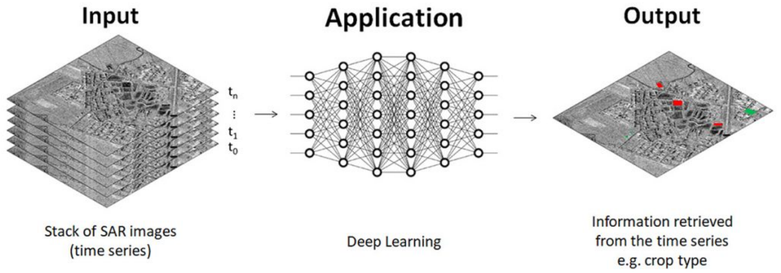
Dataset	LLP Methods	Bag Size				SwAV (kNN)	ConCURL	Supervised
		16	32	64	128			
CIFAR-10	DLLP-KL	86.0	72.0	56.0	41.0	80.0	84.6	93.6
	DLLP-ROT	78.0	65.0	53.0	40.0			
	LLP-GAN	86.3	83.8	79.0	72.6			
	LLP-GAN-PLOT	89.3	88.2	84.1	79.1			
	LLP-Co (Acc_H) (**ours**)	**90.0**	**89.8**	**90.9**	**86.2**			
	LLP-Co (Acc_A) (**ours**)	**90.0**	**89.8**	**90.9**	72.1			
CIFAR-100	DLLP-KL	58.0	38.0	24.0	14.0	45.8	47.9	78.3
	DLLP-ROT	51.0	37.0	24.0	14.0			
	LLP-GAN	49.1	43.6	35.6	15.0			
	LLP-GAN-PLOT	**65.4**	61.7	55.7	43.4			
	LLP-Co (Acc_H) (**ours**)	59.5	**65.9**	**66.5**	**64.7**			
	LLP-Co (Acc_A) (**ours**)	59.4	65.7	**66.5**	62.0			

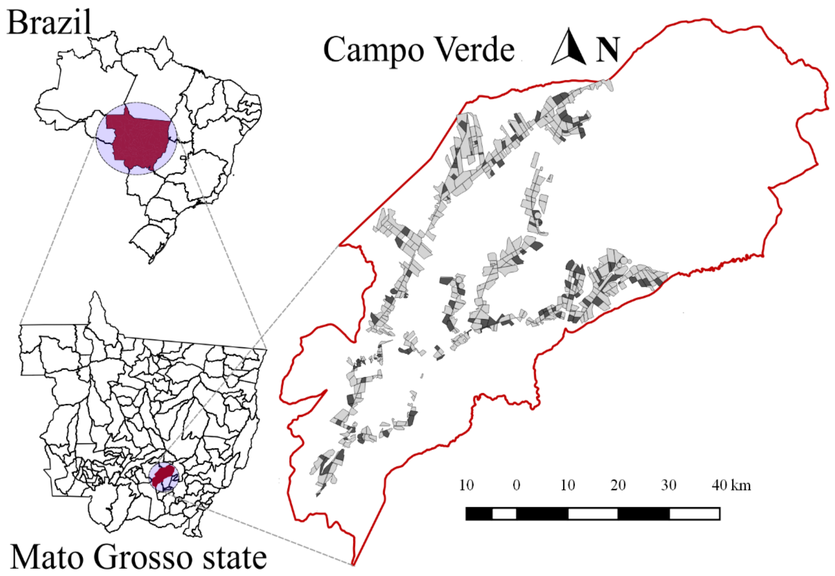
"Learning from Label Proportions with Prototypical Contrastive Clustering". Rosa et Oliveira, AAAI 2022.

Data Representation Learning + Learning from Label Proportions

Individual Crop Classification from Global Crop Area Estimate

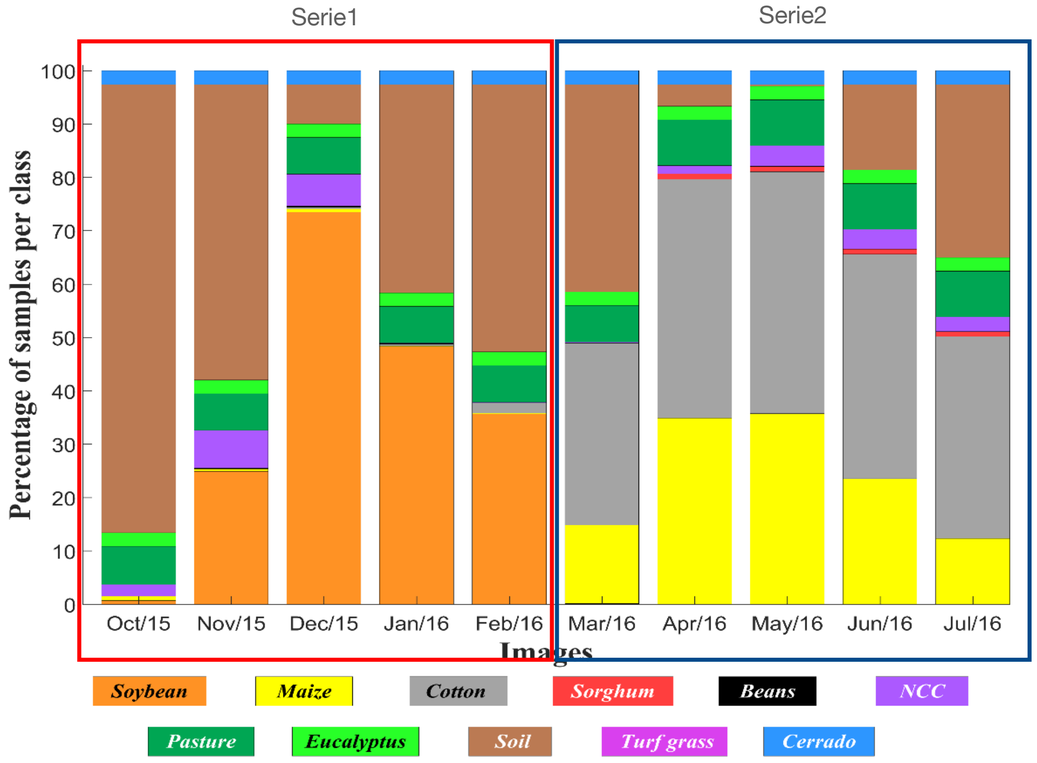
Spatiotemporal data and Agriculture





Campo Verde:

- Mato Grosso, Brazil
-14 Sentinel-1 VV/VH polarization
- From October 2015 to July 2016
- 11 classes



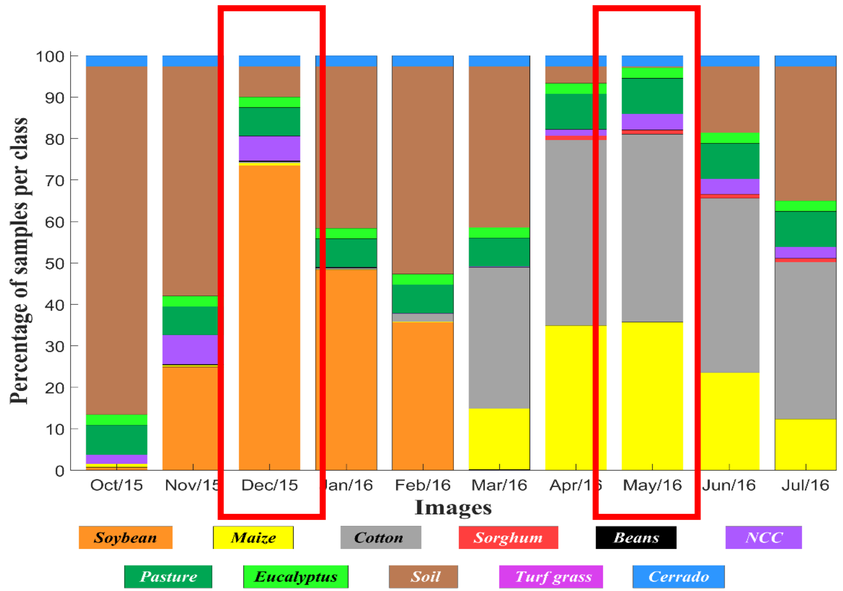
[5] Ieda Del'Arco Sanches, Raul Queiroz Feitosa, Pedro Marco Achanccaray Diaz, Marinalva Dias Soares, Alfredo Jose Barreto Luiz, Bruno Schultz, Luis Eduardo Pinheiro Maurano, October 22, 2017, "Campo Verde Database", IEEE Dataport, doi: https://dx.doi.org/10.21227/H2804B.

Data Representation Learning + Learning from Label Proportions

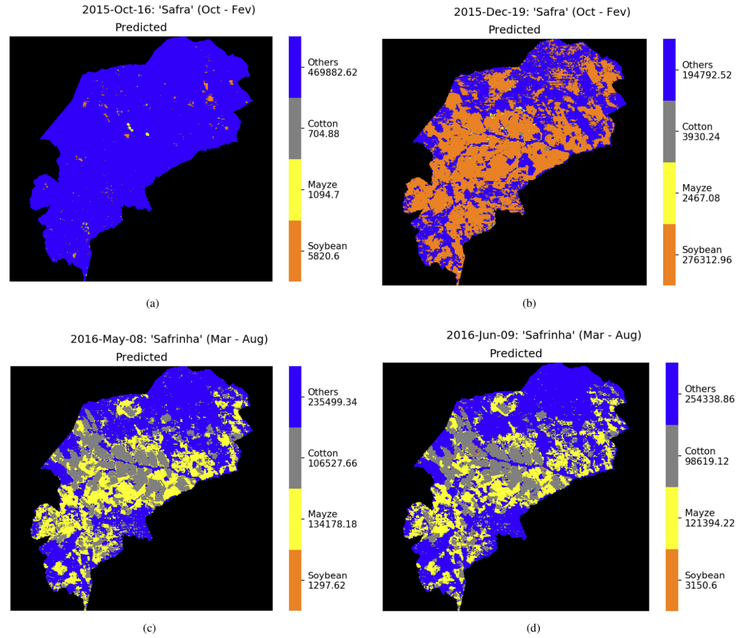
Individual Crop Classification from Global Crop Area Estimate

How LLP applies to Agriculture?

Previous studies raises the possibility of using crop proportions to validate the generalization of trained networks using data from agriculture ministry census (in Brazil)*



Roughly known proportions from governmental census data*



Labelled Map for the corresponding geographic region

* Macedo, M; Mattos A and Oliveira, DAB. “Generalization of Convolutional LSTM Models for Crop Area Estimation.” *IEEE JSTARS, 2020.*

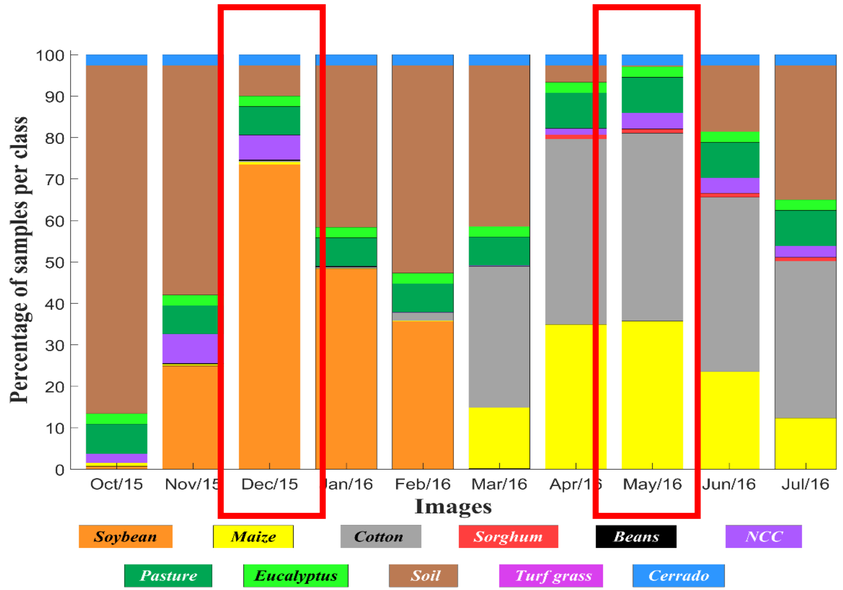
Data Representation Learning + Learning from Label Proportions

Individual Crop Classification from Global Crop Area Estimate

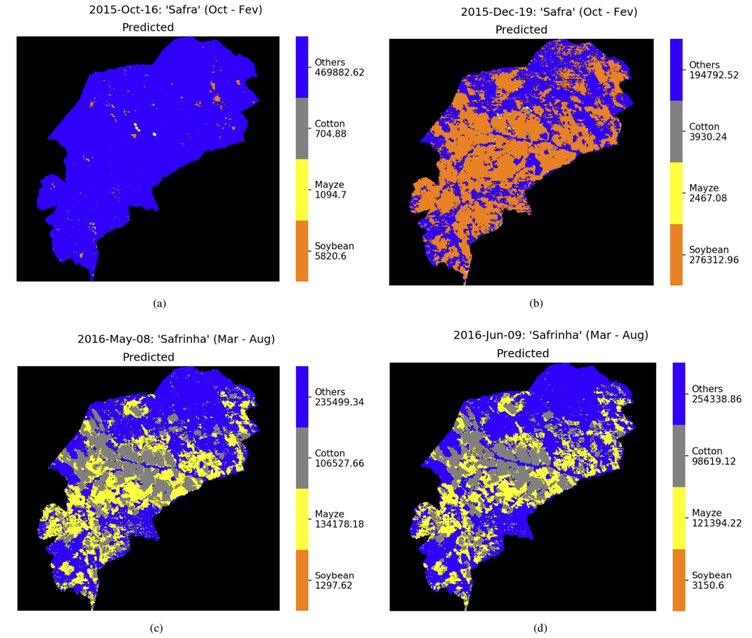
How LLP applies to Agriculture?

Previous studies raises the possibility of using crop proportions to validate the generalization of trained networks using data from agriculture ministry census (in Brazil)*

Goal: Evaluate the feasibility of using crop proportions data from agriculture census to automatically classify crops in regions with known crop proportion estimate.



Roughly known proportions from governmental census data*



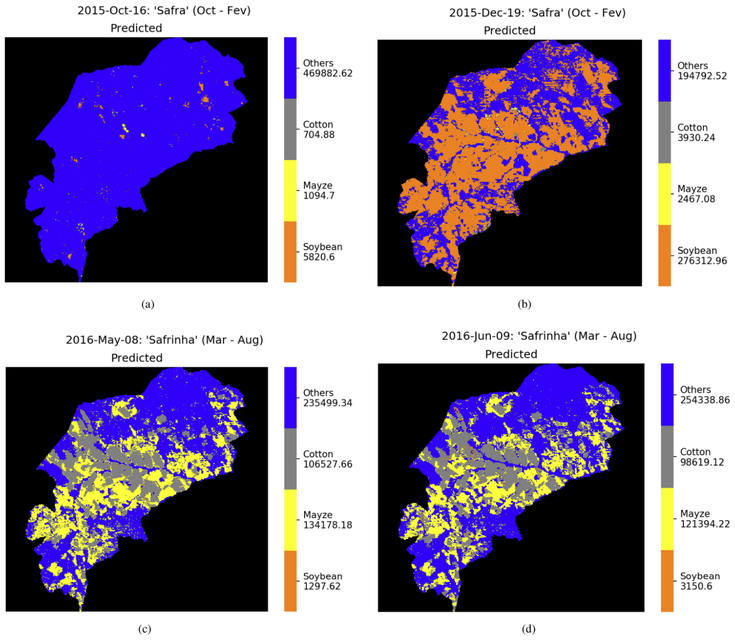
Labelled Map for the corresponding geographic region

* Macedo, M; Mattos A and Oliveira, DAB. “Generalization of Convolutional LSTM Models for Crop Area Estimation.” *IEEE JSTARS, 2020.*

Data Representation Learning + Learning from Label Proportions

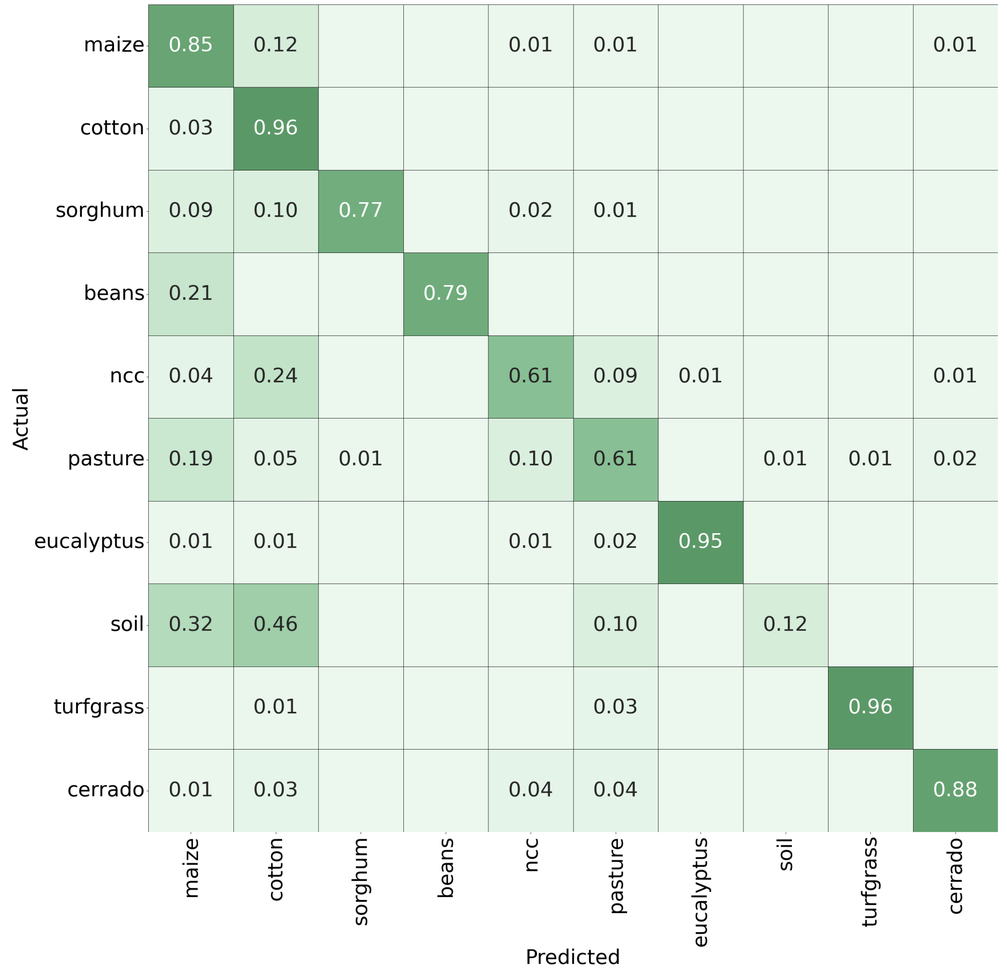
Individual Crop Classification from Global Crop Area Estimate

Goal: Evaluate the feasibility of using crop proportions data from agriculture census to automatically classify crops in regions with known crop proportion estimate.



Labelled Map for the corresponding geographic region

Preliminary results (CM) using crop prototypical clustering from label proportions

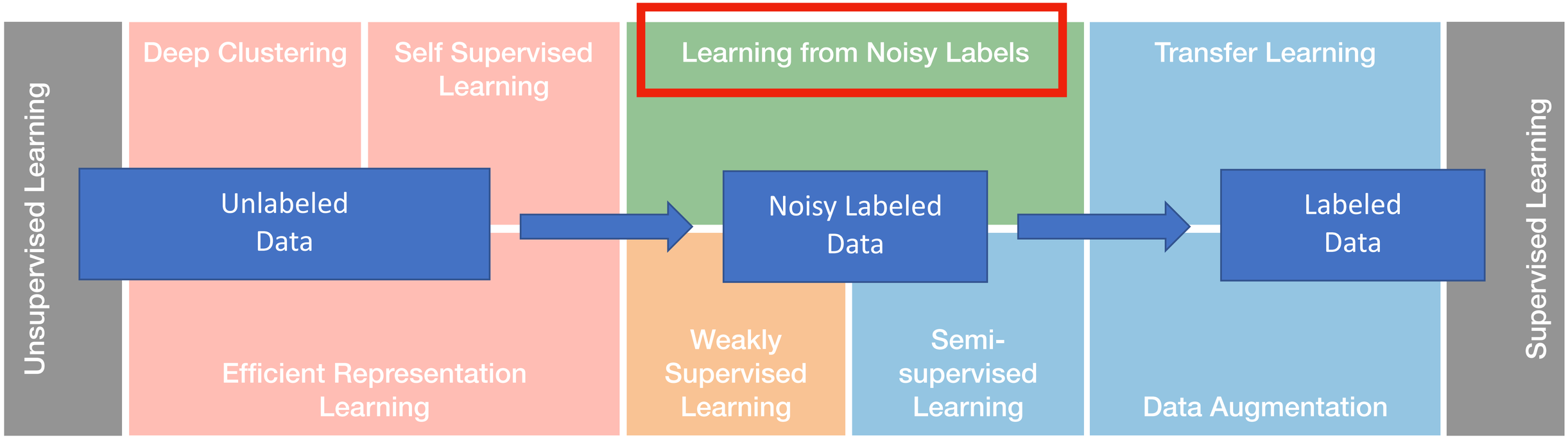


"*Crop type mapping from label proportions using Optical and SAR data*". Rosa et Oliveira, IEEE TGARS 2022 (to be submitted).

Why learning from unlabeled data?

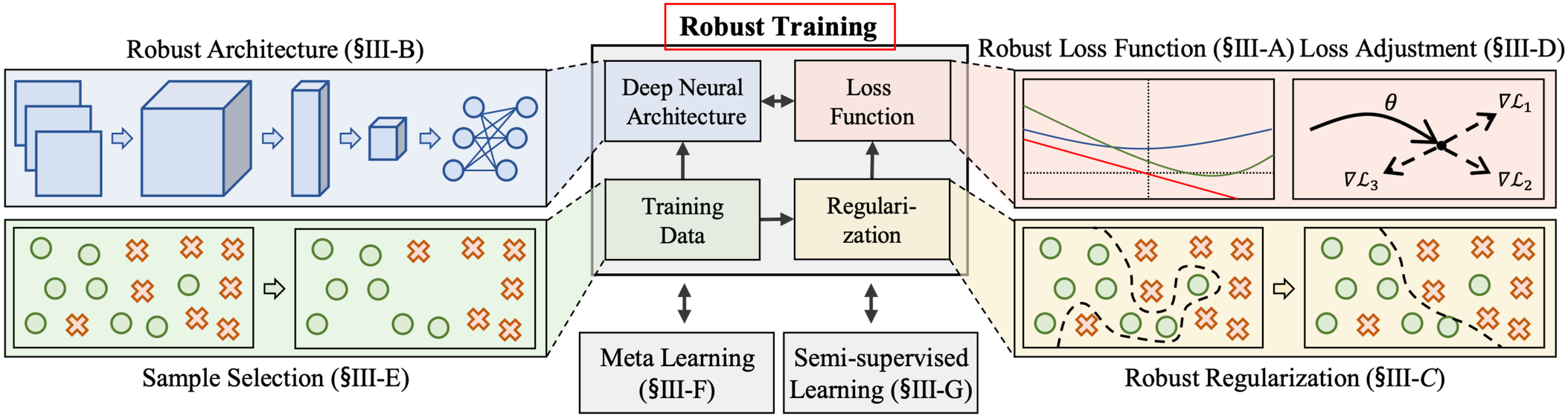
Moving towards supervised learning using poor training sets

An evolving learning process mitigates massive manual labeling dependency!

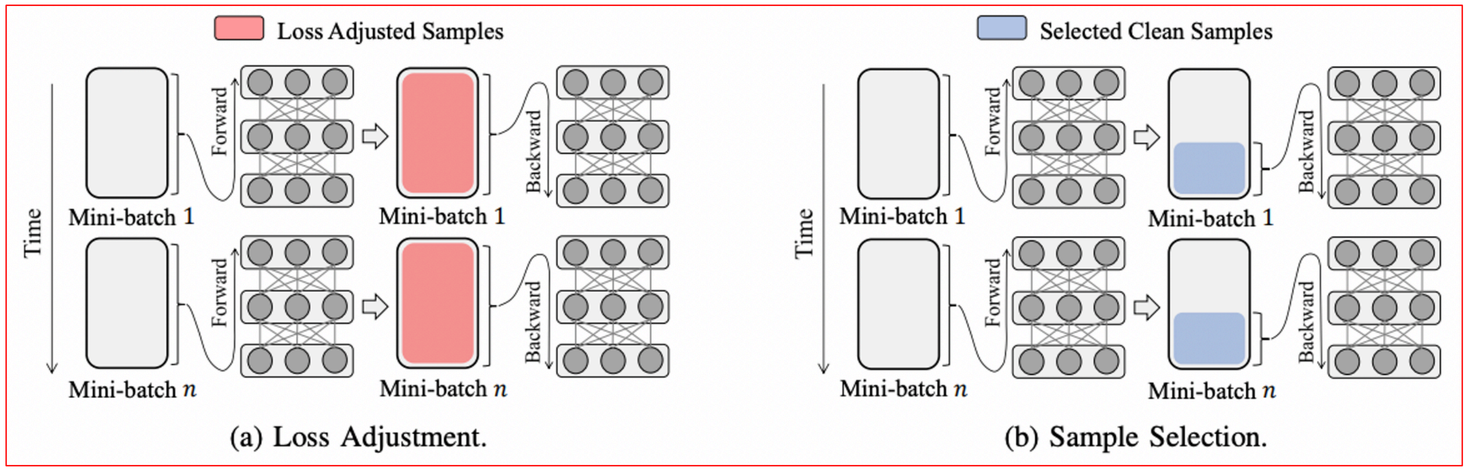


Learning from Noisy Labels

What to do when your labels are poorly annotated



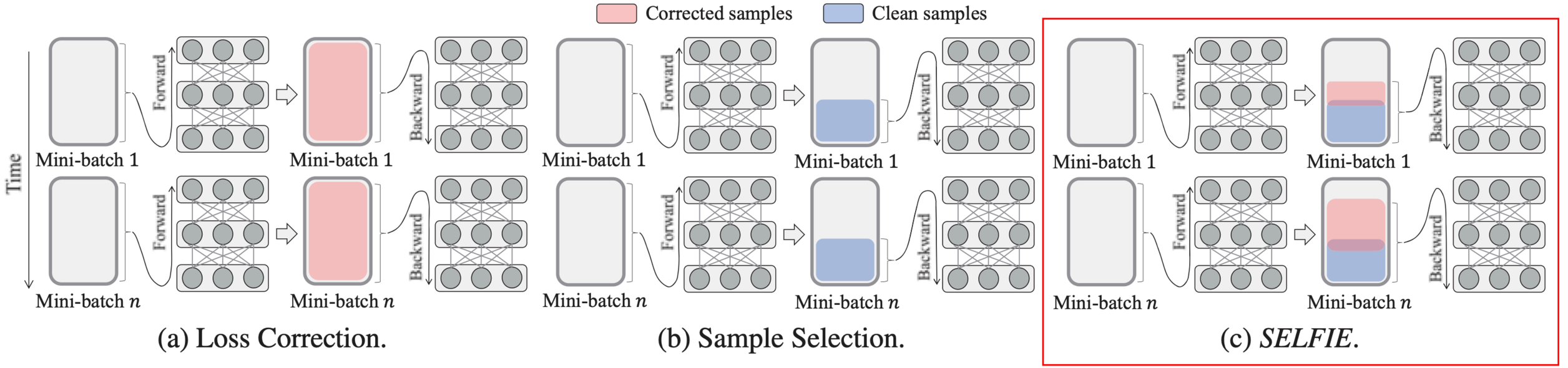
Category	Deep Learning Method
Robust Loss Function	*Robust MAE* [49], *Generalized Cross Entropy* [54], *Symmetric Cross Entropy* [55], *Curriculum Loss* [56]
Robust Architecture	*Webly Learning* [57], *Noise Model* [58], *Dropout Noise Model* [59], *S–model* [60], *C–model* [60], *NLNN* [61], *Probabilistic Noise Model* [15], *Masking* [62], *Contrastive-Additive Noise Network* [63]
Robust Regularization	*Adversarial Training* [64], *Label Smoothing* [65], *Mixup* [66], *Bilevel Learning* [67], *Annotator Confusion* [68], *Pre-training* [69]
Loss Adjustment	*Backward Correction* [70], *Forward Correction* [70], *Gold Loss Correction* [71], *Importance Reweighting* [72], *Active Bias* [73], *Bootstrapping* [74], *Dynamic Bootstrapping* [75], *D2L* [76], *SELFIE* [18]
Sample Selection	*Decouple* [51], *MentorNet* [77], *Co-teaching* [78], *Co-teaching+* [79], *Iterative Detection* [80], *ITLM* [81], *INCV* [82], *SELFIE* [18], *SELF* [83], *Curriculum Loss* [56]
Meta Learning	*Meta-Regressor* [52], *Knowledge Distillation* [84], *L2LWS* [85], *CWS* [86], *Automatic Reweighting* [87], *MLNT* [88], *Meta-Weight-Net* [89], *Data Coefficients* [101]
Semi-supervised Learning	*Label Aggregation* [53], *Two-Stage Framework* [90], *SELF* [83], *DivideMix* [91]



"Learning from Noisy Labels with Deep Neural Networks: A Survey", Song et al., ArXiv 2020

Learning from Noisy Labels

SELFIE: Reclassifying inconsistent labels



Usual model update

learning rate

loss function

$$\theta_{t+1} = \theta_t - \alpha \nabla \Big(\frac{1}{|\mathcal{M}|} \sum_{x \in \mathcal{M}} \mathcal{L}(x, \tilde{y}; \theta_t) \Big)$$

mini-batch

Selfie model update

$$\theta_{t+1} = \theta_t - \alpha \nabla \Big(\frac{1}{|\mathcal{R} \cup \mathcal{C}|} \Big(\sum_{x \in \mathcal{R}} \mathcal{L}(x, y^{refurb}; \theta_t) + \sum_{x \in \mathcal{C} \cap \mathcal{R}^{-1}} \mathcal{L}(x, \tilde{y}; \theta_t) \Big) \Big)$$

refurbishable samples

$\tilde{y} \rightarrow y^{refurb}$

clean samples

"SELFIE: Refurbishing Unclean Samples for Robust Deep Learning ", Song et al., ICML 19

Learning from Noisy Labels

SELFIE: Reclassifying inconsistent labels

1. How to select $\mathcal{C}$?

2. How to select $\mathcal{R}$?

3. How to correct $\mathcal{R}$ ($\tilde{y} \rightarrow y^{refurb}$) ?

Lowest loss-based separation method

$\mathcal{C}$ = $(1 - \tau) \times 100\%$ samples with the lowest losses

error rate

Usually empirical or computed using the aggregated error between different folds in cross-validation

"SELFIE: Refurbishing Unclean Samples for Robust Deep Learning ", Song et al., ICML 19

Learning from Noisy Labels

SELFIE: Reclassifying inconsistent labels

1. How to select $\mathcal{C}$?

2. How to select $\mathcal{R}$?

3. How to correct $\mathcal{R}$ ($\tilde{y} \rightarrow y^{refurb}$) ?

q predictions

Definition 3.1. A sample x is *refurbishable* if the predictive uncertainty in Eq. (3) $F(x;q) \leq \epsilon$ $(0 \leq \epsilon \leq 1)$.

$$F(x;q) = (1/\delta)\ \text{entropy}\big(P(y|x;q)\big)$$

Low entropy = assertive classification

$$\delta = -\sum_{i=1}^{k}(1/k)\log(1/k) = -\log(1/k)$$

normalization term so *F*∈ *[0,1]*

$$\text{entropy}(P(y|x;q)) = -\sum_{j=1}^{k} P(j|x;q)\log P(j|x;q)$$

proportion of predictions of a given class in the history of *q* predictions

$$P(y|x;q) = \frac{\sum_{\hat{y}\in H_x(q)}[\hat{y}=y]}{|H_x(q)|}$$

history of *q* predictions for x

Learning from Noisy Labels

SELFIE: Reclassifying inconsistent labels

1. How to select $\mathcal{C}$?

2. How to select $\mathcal{R}$?

3. How to correct $\mathcal{R}$ ($\tilde{y} \rightarrow y^{refurb}$) ?

Definition 3.2. A *refurbished* label y^{refurb} of the refurbishable sample x is the most frequently predicted label for previous q times, as in Eq. (7), where the sample x satisfies the condition $F(x;q) \leq \epsilon$.

$$y^{refurb} = \text{argmax}_{1 \leq j \leq k} P(j|x;q)$$

$$P(y|x;q) = \frac{\sum_{\hat{y} \in H_x(q)} [\hat{y} = y]}{|H_x(q)|}$$

history of q predictions for x

Learning from Noisy Labels

SELFIE: Reclassifying inconsistent labels

Algorithm 1 *SELFIE* Algorithm

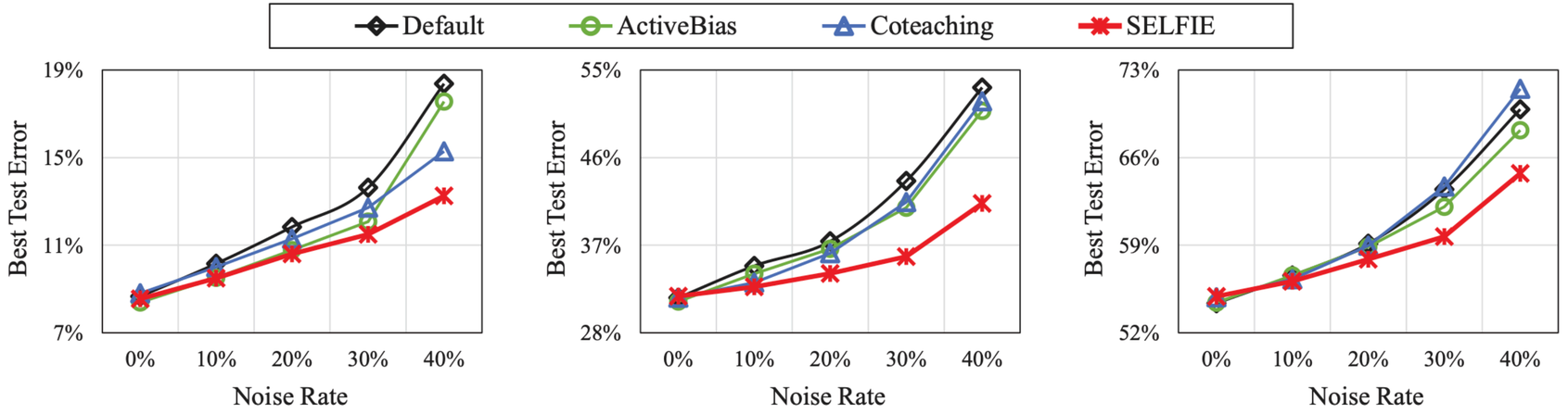
INPUT: $\mathcal{D}$: data, *epochs*, γ: warm-up, τ: noise rate, ϵ: uncertainty threshold, q: history length, $restart$: # restarts
OUTPUT: θ_t: model parameter, $\mathbb{R}$: refurbished samples

$\mathbb{R} \leftarrow \emptyset$; /* $\mathbb{R}$ is entire refurbished samples in $\mathcal{D}$ */
for $r = 0$ **to** $restart$ **do**
  $t \leftarrow 1$;
  $\theta_t \leftarrow$ Initialize the model parameter;
  for $i = 1$ **to** $epochs$ **do**
    for $j = 1$ **to** $|\mathcal{D}|/|\mathcal{M}|$ **do**
      Draw a mini-batch $\mathcal{M}$ from $\mathcal{D}$;
      if $i \leq \gamma$ **then** /* Warm-up periods */
        /* Update by Eq. (1) */
        $\theta_{t+1} = \theta_t - \alpha\nabla\big(\frac{1}{|\mathcal{M}|}\sum_{x\in\mathcal{M}}\mathcal{L}(x,\tilde{y};\theta_t)\big)$;
      else
        /* Clean sample selection */
        $\mathcal{C} \leftarrow (1-\tau)\times 100\%$ of low-loss samples in $\mathcal{M}$;
        /* Selective loss correction in Sec. 3.2 */
        $\mathcal{R} \leftarrow \emptyset$; /* $\mathcal{R}$ is refurbished samples in $\mathcal{M}$ */;
        for each $x \in \mathcal{M}$ **do**
          if $F(x,q) \leq \epsilon$ **or** $x \in \mathbb{R}$ **do** /* By Eq. (3) */
            $\mathcal{R} \leftarrow \mathcal{R} \cup (x, y^{refurb})$; /* Refurbish */
        $\mathbb{R} \leftarrow \mathbb{R} \cup \mathcal{R}$; /* Aggregation */
        /* Update by Eq. (2) */
        $\theta_{t+1} = \theta_t - \alpha\nabla\Big(\frac{1}{|\mathcal{R}\cup\mathcal{C}|}\big(\sum_{x\in\mathcal{R}}\mathcal{L}(x,y^{refurb};\theta_t) + \sum_{x\in\mathcal{C}\cap\mathcal{R}^{-1}}\mathcal{L}(x,\tilde{y};\theta_t)\big)\Big)$;
      $t \leftarrow t+1$;
return $\theta_t, \mathbb{R}$;

"SELFIE: Refurbishing Unclean Samples for Robust Deep Learning ", Song et al., ICML 19

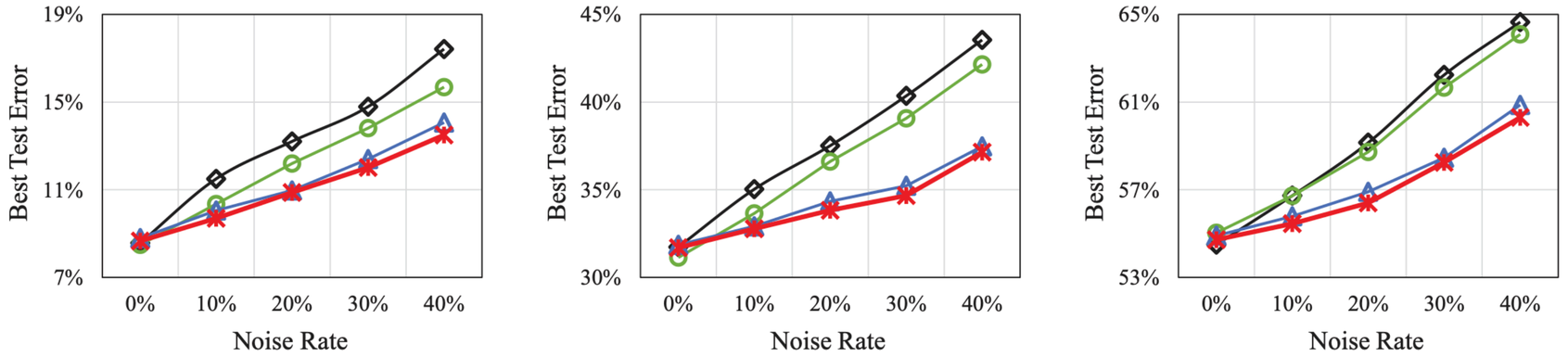
Learning from Noisy Labels

SELFIE: Reclassifying inconsistent labels



(a) CIFAR-10. (b) CIFAR-100. (c) Tiny-ImageNet.

Figure 4. The best test error of the four training methods using DenseNet on three data sets with varying **pair noise** rates.



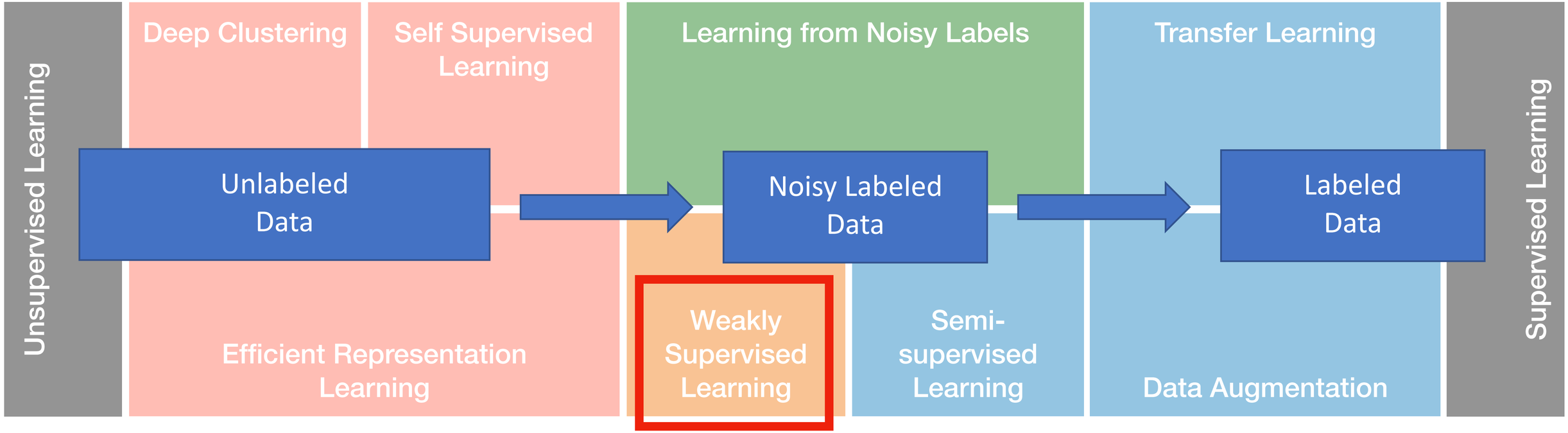
(a) CIFAR-10. (b) CIFAR-100. (c) Tiny-ImageNet.

"SELFIE: Refurbishing Unclean Samples for Robust Deep Learning ", Song et al., ICML 19

Why learning from unlabeled data?

Moving towards supervised learning using poor training sets

An evolving learning process mitigates massive manual labeling dependency!

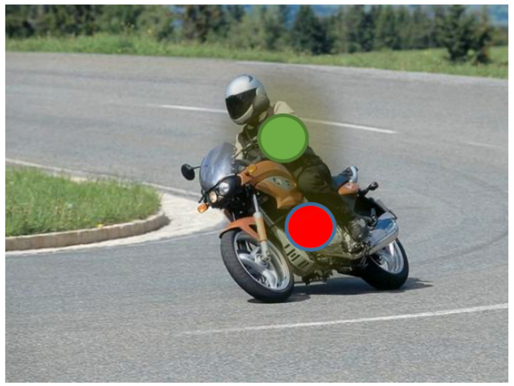


Weakly-Supervised Learning

Training with labels cheaper than what we need in test

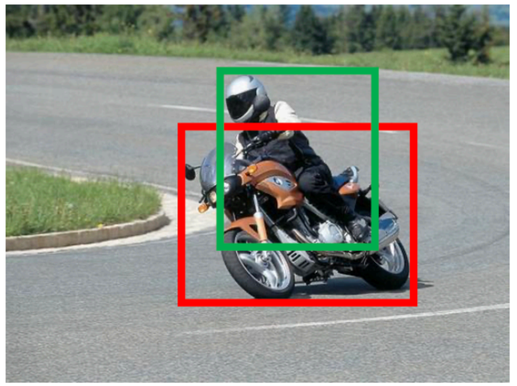
{motorbike,person}

1 sec
per class



{motorbike (point), person (point)}

2.4 sec
per class



{motorbike (b-box), person (b-box)}

10 sec
per class

{motorbike (pixel labels), person (pixel labels)}

78 sec
per class

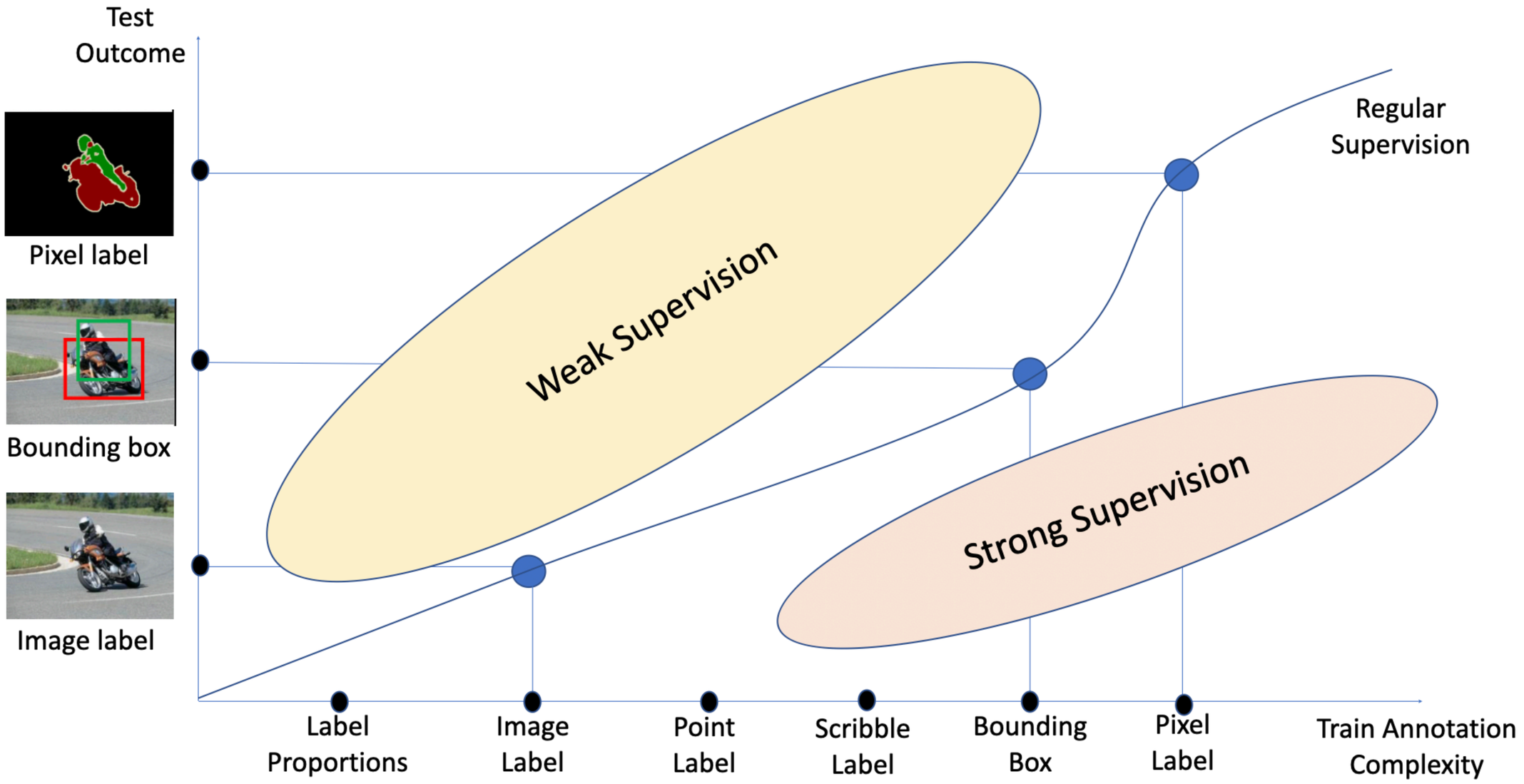
Berman et al., What's the Point: Semantic Segmentation with Point Supervision, ECCV 16

<u>Weak Supervision</u>: Lower degree (or cheaper) annotation at train time than the required output at test time

* From EECV 2020 Weakly Supervised Learning in CV Tutorial

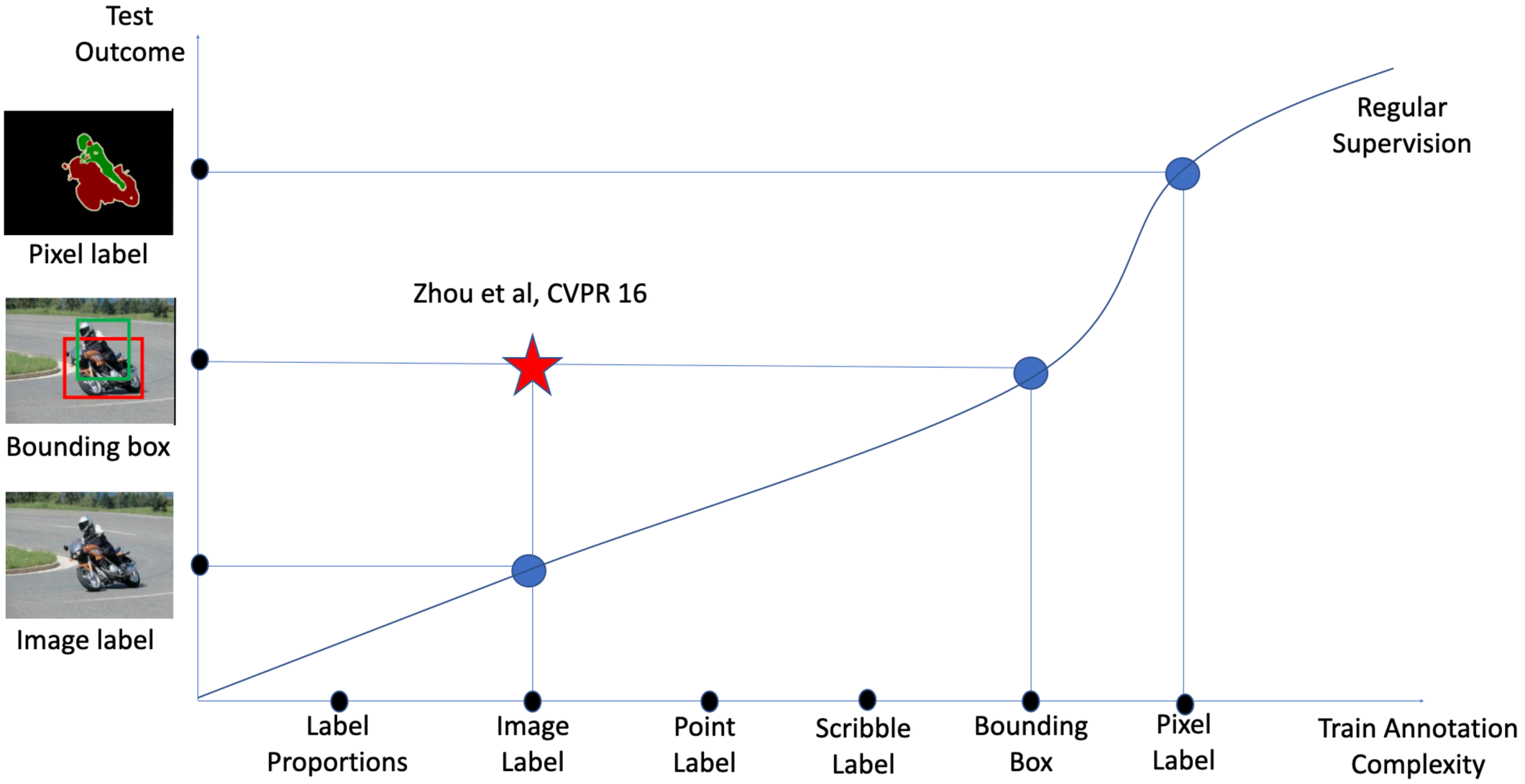
Weakly-Supervised Learning

Training with labels cheaper than what we need in test



Weakly-Supervised Learning

Training with labels cheaper than what we need in test



Weakly-Supervised Learning

Training with classification labels for detection/segmentation

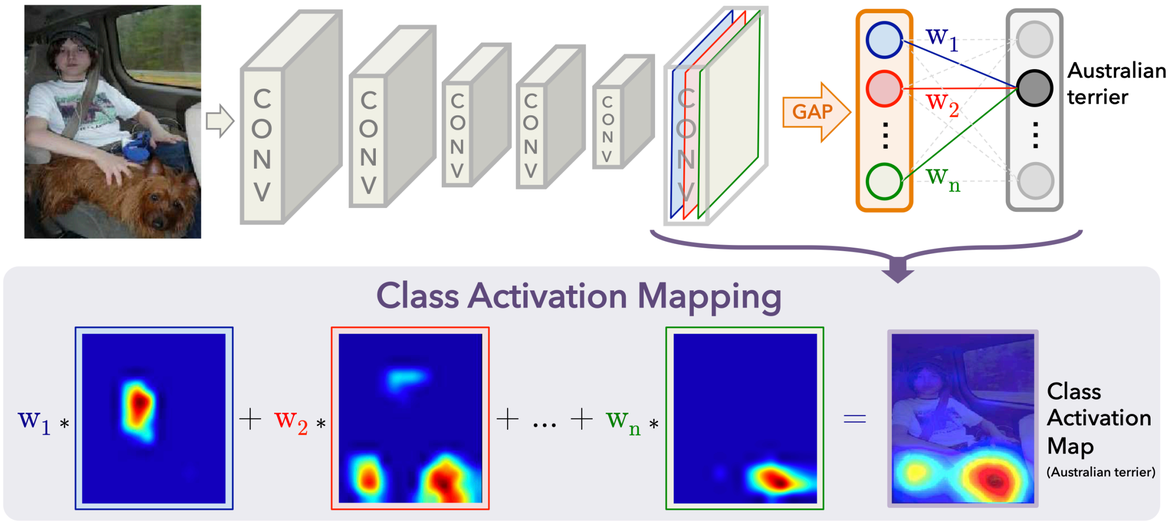


Figure 2. Class Activation Mapping: the predicted class score is mapped back to the previous convolutional layer to generate the class activation maps (CAMs). The CAM highlights the class-specific discriminative regions.

For a given image, let $f_k(x, y)$ represent the activation of unit k in the last convolutional layer at spatial location (x, y). Then, for unit k, the result of performing global average pooling, F^k is $\sum_{x,y} f_k(x, y)$. Thus, for a given class c, the input to the softmax, S_c, is $\sum_k w_k^c F_k$ where w_k^c is the weight corresponding to class c for unit k. Essentially, w_k^c indicates the *importance* of F_k for class c. Finally the output of the softmax for class c, P_c is given by $\frac{\exp(S_c)}{\sum_c \exp(S_c)}$.

By plugging $F_k = \sum_{x,y} f_k(x, y)$ into the class score, S_c, we obtain

$$S_c = \sum_k w_k^c \sum_{x,y} f_k(x, y) = \sum_{x,y} \sum_k w_k^c f_k(x, y). \quad (1)$$

We define M_c as the class activation map for class c, where each spatial element is given by

$$M_c(x, y) = \sum_k w_k^c f_k(x, y). \quad (2)$$

First train the classification network

↓

Then fix the convolutional layers and train a new MLP layer to learn the weights (or the contributions) of each activation map to the classification

↓

Sum up the contributions and get a sketch of the active zones for detection/segmentation

Learning Deep Features for Discriminative Localization, Zhou et al., CVPR 2016

Weakly-Supervised Learning

Training with classification labels for detection/segmentation

Action recognition CAM

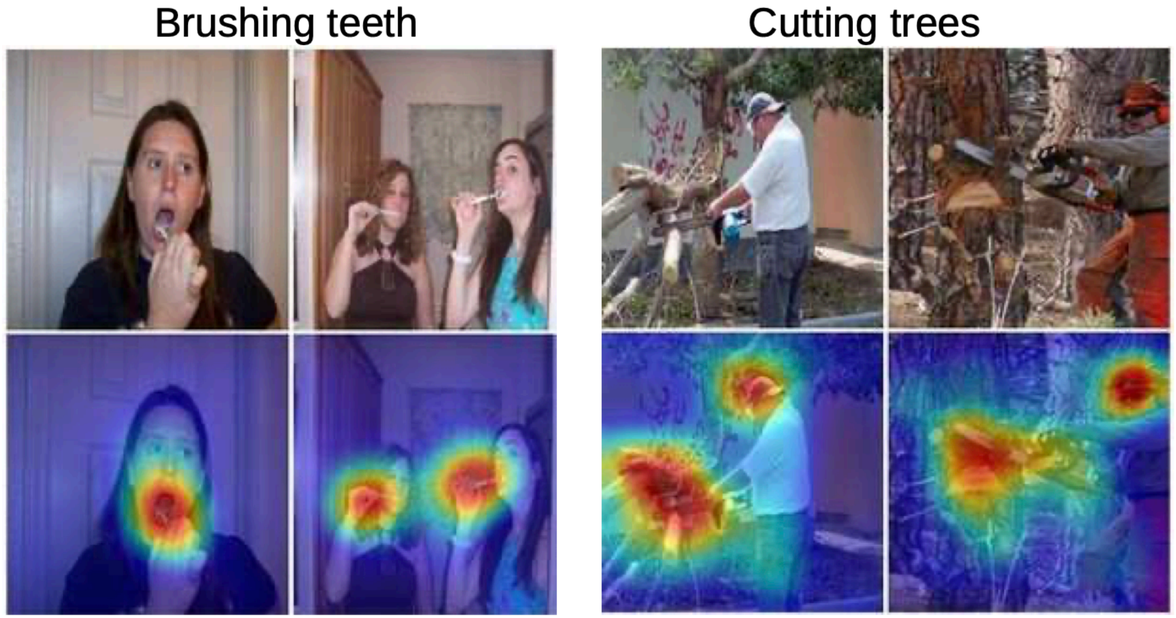
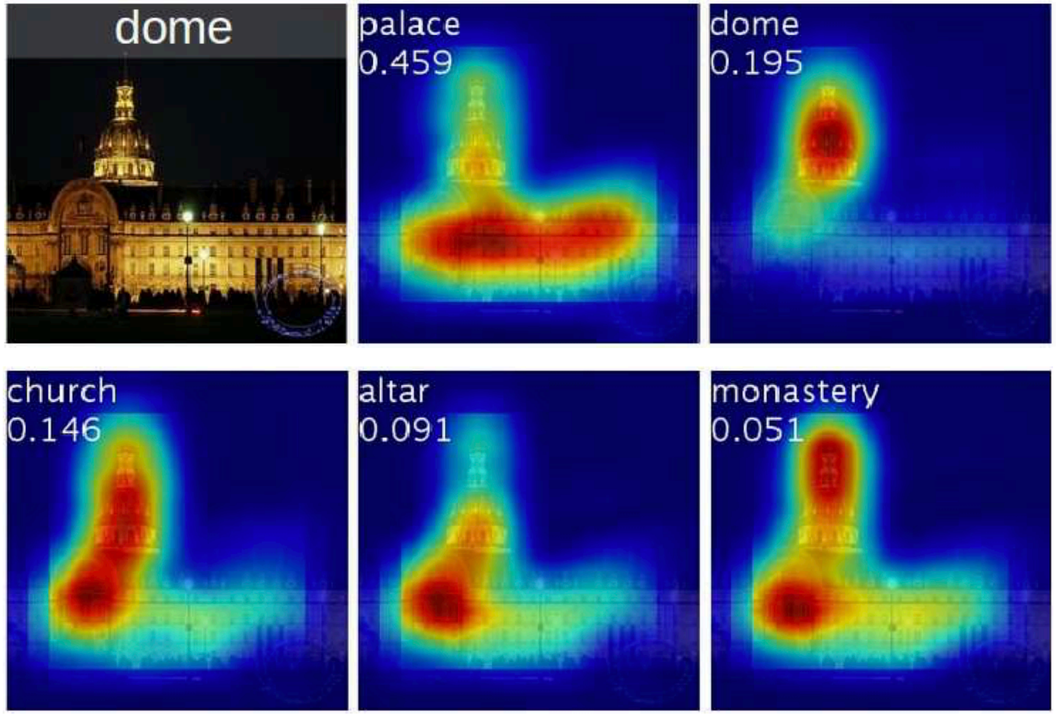


Image Classification CAM



Learning Deep Features for Discriminative Localization, Zhou et al., CVPR 2016

Weakly-Supervised Learning

Training with classification labels for detection/segmentation

Localization: In order to perform localization, we need to generate a bounding box and its associated object category. To generate a bounding box from the CAMs, we use a simple thresholding technique to segment the heatmap. We first segment the regions of which the value is above 20% of the max value of the CAM. Then we take the bounding box that covers the largest connected component in the segmentation map. We do this for each of the top-5 predicted classes for the top-5 localization evaluation metric.

Table 2. Localization error on the ILSVRC validation set. *Backprop* refers to using [23] for localization instead of CAM.

Method	top-1 val.error	top-5 val. error
GoogLeNet-GAP	**56.40**	**43.00**
VGGnet-GAP	57.20	45.14
GoogLeNet	60.09	49.34
AlexNet*-GAP	63.75	49.53
AlexNet-GAP	67.19	52.16
NIN	65.47	54.19
Backprop on GoogLeNet	61.31	50.55
Backprop on VGGnet	61.12	51.46
Backprop on AlexNet	65.17	52.64
GoogLeNet-GMP	57.78	45.26

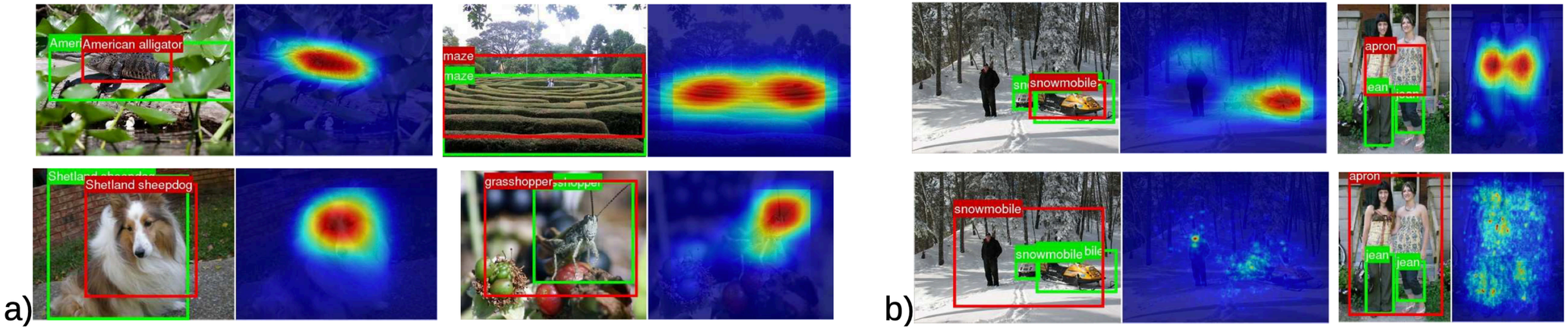
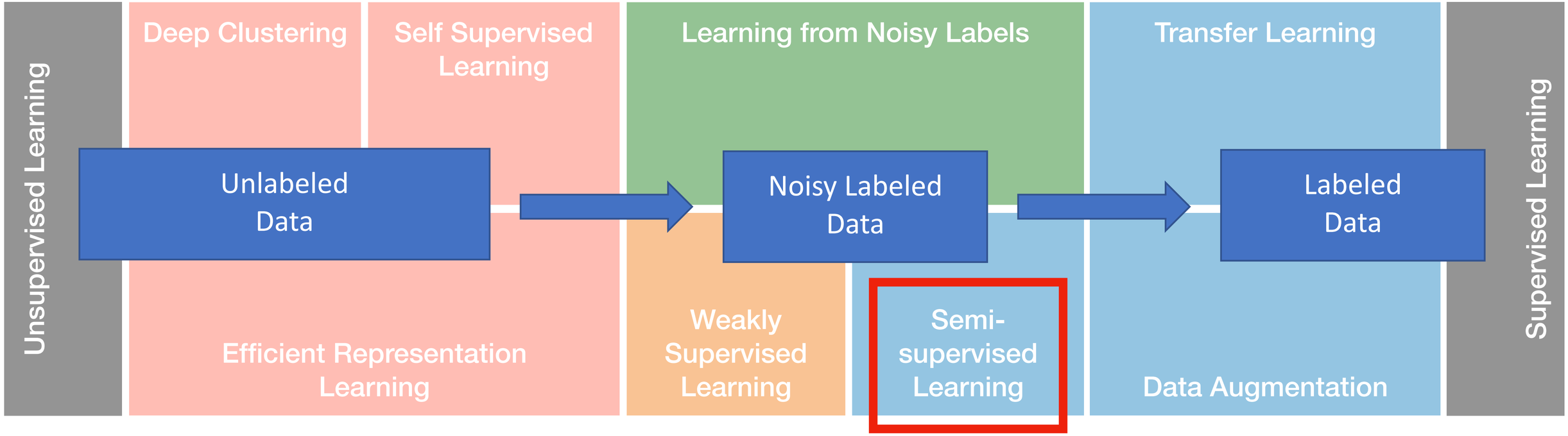


Figure 6. a) Examples of localization from GoogleNet-GAP. b) Comparison of the localization from GooleNet-GAP (upper two) and the backpropagation using AlexNet (lower two). The ground-truth boxes are in green and the predicted bounding boxes from the class activation map are in red.

Why learning from unlabeled data?

Moving towards supervised learning using poor training sets

An evolving learning process mitigates massive manual labeling dependency!



Semi-Supervised Learning

Training with both labeled and unlabeled data

(a) Supervised learning

(b) Weakly supervised learning

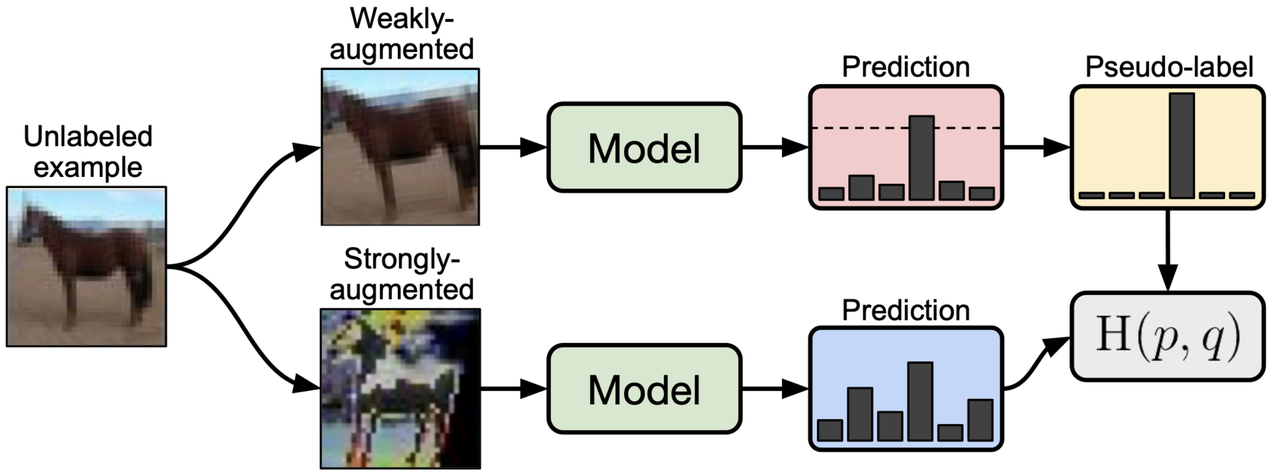
(c) Weakly semi-supervised learning

(d) Semi-supervised learning

* From EECV 2020 Weakly Supervised Learning in CV Tutorial

Semi-Supervised Learning

Training with both labeled and unlabeled data



"FixMatch algorithm is a simple combination of two common approaches to SSL: consistency regularization and pseudo-labeling."

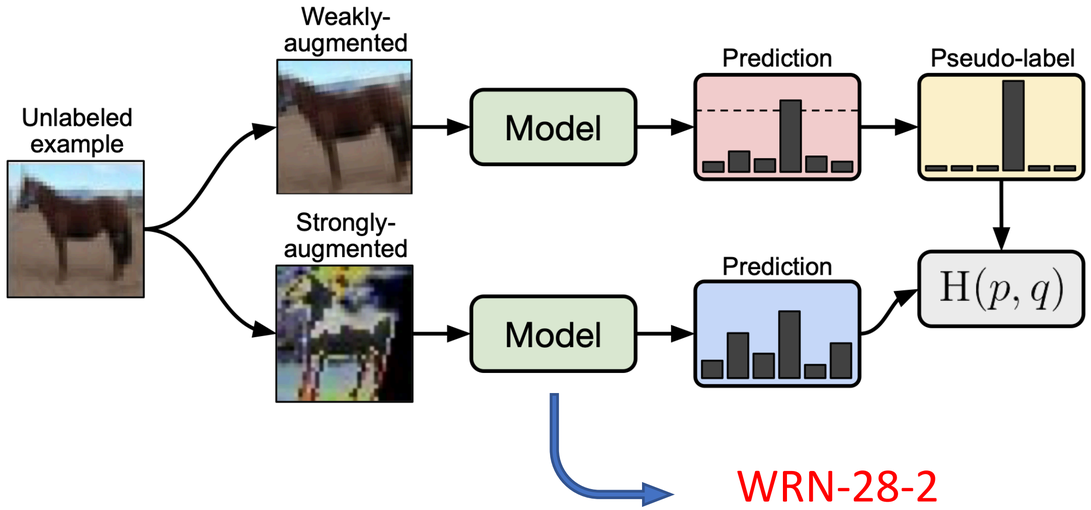
Consistency regularization utilizes unlabeled data by relying on the assumption that the model should output similar predictions when fed perturbed versions of the same image.

Pseudo-labeling leverages the idea that we should use the model itself to obtain artificial labels for unlabeled data.

FixMatch: Simplifying Semi-Supervised Learning with Consistency and Confidence, Sohn et al., 2020, arXiv

Semi-Supervised Learning

Training with both labeled and unlabeled data



WRN-28-2
Wide-ResNet
depth=28
widening=2

group name	output size	block type = $B(3,3)$
conv1	32×32	[3×3, 16]
conv2	32×32	$\begin{bmatrix} 3\times3, 16\times k \\ 3\times3, 16\times k \end{bmatrix} \times N$
conv3	16×16	$\begin{bmatrix} 3\times3, 32\times k \\ 3\times3, 32\times k \end{bmatrix} \times N$
conv4	8×8	$\begin{bmatrix} 3\times3, 64\times k \\ 3\times3, 64\times k \end{bmatrix} \times N$
avg-pool	1×1	$[8 \times 8]$

Table 1: Structure of wide residual networks. Network width is determined by factor k. Original architecture [13] is equivalent to $k = 1$. Groups of convolutions are shown in brackets where N is a number of blocks in group, downsampling performed by the first layers in groups `conv3` and `conv4`. Final classification layer is omitted for clearance. In the particular example shown, the network uses a ResNet block of type $B(3,3)$.

FixMatch: Simplifying Semi-Supervised Learning with Consistency and Confidence, Sohn et al., 2020, arXiv

Semi-Supervised Learning

Training with both labeled and unlabeled data

Algorithm 1 FixMatch algorithm.

Input: Labeled batch $\mathcal{X} = \{(x_b, p_b) : b \in (1, \dots, B)\}$, unlabeled batch $\mathcal{U} = \{u_b : b \in (1, \dots, \mu B)\}$, confidence threshold τ, unlabeled data ratio μ, unlabeled loss weight λ_u.

$\ell_s = \frac{1}{B} \sum_{b=1}^{B} \mathrm{H}(p_b, \alpha(x_b))$ *// Cross-entropy loss for labeled data*

for $b = 1$ **to** μB **do**

  $\tilde{u}_b = \mathcal{A}(u_b)$ *// Apply strong data augmentation to* u_b

  $q_b = p_{\mathrm{m}}(y \mid \alpha(u_b); \theta)$ *// Compute prediction after applying weak data augmentation of* u_b

end for

$\ell_u = \frac{1}{\mu B} \sum_{b=1}^{\mu B} \mathbb{1}\{\max(q_b) > \tau\} \mathrm{H}(\arg\max(q_b), \tilde{u}_b)$ *// Cross-entropy loss with pseudo-label and confidence for unlabeled data*

return $\ell_s + \lambda_u \ell_u$

Input data:

1. labeled and unlabeled samples
2. a know ratio of unlabeled samples with respect to labeled samples
3. a loss weight that weights the importance of unsupervised loss in the total loss

FixMatch: Simplifying Semi-Supervised Learning with Consistency and Confidence, Sohn et al., 2020, arXiv

Semi-Supervised Learning

Training with both labeled and unlabeled data

Algorithm 1 FixMatch algorithm.

Input: Labeled batch $\mathcal{X} = \big\{(x_b, p_b) : b \in (1, \dots, B)\big\}$, unlabeled batch $\mathcal{U} = \big\{u_b : b \in (1, \dots, \mu B)\big\}$, confidence threshold τ, unlabeled data ratio μ, unlabeled loss weight λ_u.

$\ell_s = \frac{1}{B} \sum_{b=1}^{B} \mathrm{H}(p_b, \alpha(x_b))$ *// Cross-entropy loss for labeled data*

for $b = 1$ **to** μB **do**

  $\tilde{u}_b = \mathcal{A}(u_b)$ *// Apply strong data augmentation to* u_b

  $q_b = p_{\mathrm{m}}(y \mid \alpha(u_b); \theta)$ *// Compute prediction after applying weak data augmentation of* u_b

end for

$\ell_u = \frac{1}{\mu B} \sum_{b=1}^{\mu B} \mathbb{1}\{\max(q_b) > \tau\} \, \mathrm{H}(\arg\max(q_b), \tilde{u}_b)$ *// Cross-entropy loss with pseudo-label and confidence for unlabeled data*

return $\ell_s + \lambda_u \ell_u$

Computes the supervised cross-entropy loss using labeled data

$$\ell_s = \frac{1}{B} \sum_{b=1}^{B} \mathrm{H}(p_b, p_{\mathrm{m}}(y \mid \alpha(x_b)))$$

weak augmentation

FixMatch: Simplifying Semi-Supervised Learning with Consistency and Confidence, Sohn et al., 2020, arXiv

Semi-Supervised Learning

Training with both labeled and unlabeled data

Algorithm 1 FixMatch algorithm.

Input: Labeled batch $\mathcal{X} = \big\{(x_b, p_b) : b \in (1, \dots, B)\big\}$, unlabeled batch $\mathcal{U} = \big\{u_b : b \in (1, \dots, \mu B)\big\}$, confidence threshold τ, unlabeled data ratio μ, unlabeled loss weight λ_u.

$\ell_s = \frac{1}{B} \sum_{b=1}^{B} \mathrm{H}(p_b, \alpha(x_b))$ *// Cross-entropy loss for labeled data*

for $b = 1$ **to** μB **do**

  $\tilde{u}_b = \mathcal{A}(u_b)$ *// Apply strong data augmentation to u_b*

  $q_b = p_{\mathrm{m}}(y \mid \alpha(u_b); \theta)$ *// Compute prediction after applying weak data augmentation of u_b*

end for

$\ell_u = \frac{1}{\mu B} \sum_{b=1}^{\mu B} \mathbb{1}\{\max(q_b) > \tau\} \, \mathrm{H}(\arg\max(q_b), \tilde{u}_b)$ *// Cross-entropy loss with pseudo-label and confidence for unlabeled data*

return $\ell_s + \lambda_u \ell_u$

For all unlabeled samples:

1. Creates and stores a synthetic sample with strong data augmentation
2. Computes the soft pseudo-labeling for unsupervised samples after weak data augmentation

FixMatch: Simplifying Semi-Supervised Learning with Consistency and Confidence, Sohn et al., 2020, arXiv

Semi-Supervised Learning

Training with both labeled and unlabeled data

Algorithm 1 FixMatch algorithm.

Input: Labeled batch $\mathcal{X} = \big\{(x_b, p_b) : b \in (1, \dots, B)\big\}$, unlabeled batch $\mathcal{U} = \big\{u_b : b \in (1, \dots, \mu B)\big\}$, confidence threshold τ, unlabeled data ratio μ, unlabeled loss weight λ_u.

$\ell_s = \frac{1}{B} \sum_{b=1}^{B} \mathrm{H}(p_b, \alpha(x_b))$ *// Cross-entropy loss for labeled data*

for $b = 1$ **to** μB **do**

  $\tilde{u}_b = \mathcal{A}(u_b)$ *// Apply strong data augmentation to* u_b

  $q_b = p_{\mathrm{m}}(y \mid \alpha(u_b); \theta)$ *// Compute prediction after applying weak data augmentation of* u_b

end for

$\ell_u = \frac{1}{\mu B} \sum_{b=1}^{\mu B} \mathbb{1}\{\max(q_b) > \tau\} \mathrm{H}(\arg\max(q_b), \tilde{u}_b)$ *// Cross-entropy loss with pseudo-label and confidence for unlabeled data*

return $\ell_s + \lambda_u \ell_u$

Computes the unsupervised loss using unlabeled data

$$\ell_u = \frac{1}{\mu B} \sum_{b=1}^{\mu B} \mathbb{1}(\max(q_b) \geq \tau) \, \mathrm{H}(q_b, p_{\mathrm{m}}(y \mid \mathcal{A}(u_b)))$$

consistency regularization

hard pseudo-label

activation threshold

FixMatch: Simplifying Semi-Supervised Learning with Consistency and Confidence, Sohn et al., 2020, arXiv

Semi-Supervised Learning

Training with both labeled and unlabeled data

Algorithm 1 FixMatch algorithm.

Input: Labeled batch $\mathcal{X} = \big\{(x_b, p_b) : b \in (1, \dots, B)\big\}$, unlabeled batch $\mathcal{U} = \big\{u_b : b \in (1, \dots, \mu B)\big\}$, confidence threshold τ, unlabeled data ratio μ, unlabeled loss weight λ_u.

$\ell_s = \frac{1}{B} \sum_{b=1}^{B} \mathrm{H}(p_b, \alpha(x_b))$ *// Cross-entropy loss for labeled data*

for $b = 1$ **to** μB **do**

  $\tilde{u}_b = \mathcal{A}(u_b)$ *// Apply strong data augmentation to* u_b

  $q_b = p_{\mathrm{m}}(y \mid \alpha(u_b); \theta)$ *// Compute prediction after applying weak data augmentation of* u_b

end for

$\ell_u = \frac{1}{\mu B} \sum_{b=1}^{\mu B} \mathbb{1}\{\max(q_b) > \tau\} \mathrm{H}(\arg\max(q_b), \tilde{u}_b)$ *// Cross-entropy loss with pseudo-label and confidence for unlabeled data*

return $\ell_s + \lambda_u \ell_u$

Computes the final loss as the weighted sum of supervised and unsupervised losses

FixMatch: Simplifying Semi-Supervised Learning with Consistency and Confidence, Sohn et al., 2020, arXiv

Semi-Supervised Learning

Training with both labeled and unlabeled data

Table 2: Error rates for CIFAR-10, CIFAR-100 and SVHN on 5 different folds. FixMatch (RA) uses RandAugment [10] and FixMatch (CTA) uses CTAugment [2] for strong-augmentation. All baseline models (Π-Model [36], Pseudo-Labeling [22], Mean Teacher [43], MixMatch [3], UDA [45], and ReMixMatch [2]) are tested using the same codebase.

	CIFAR-10			CIFAR-100			SVHN		
Method	40 labels	250 labels	4000 labels	400 labels	2500 labels	10000 labels	40 labels	250 labels	1000 labels
Π-Model	-	54.26±3.97	14.01±0.38	-	57.25±0.48	37.88±0.11	-	18.96±1.92	7.54±0.36
Pseudo-Labeling	-	49.78±0.43	16.09±0.28	-	57.38±0.46	36.21±0.19	-	20.21±1.09	9.94±0.61
Mean Teacher	-	32.32±2.30	9.19±0.19	-	53.91±0.57	35.83±0.24	-	3.57±0.11	3.42±0.07
MixMatch	47.54±11.50	11.05±0.86	6.42±0.10	67.61±1.32	39.94±0.37	28.31±0.33	42.55±14.53	3.98±0.23	3.50±0.28
UDA	29.05±5.93	8.82±1.08	4.88±0.18	59.28±0.88	33.13±0.22	24.50±0.25	52.63±20.51	5.69±2.76	**2.46**±0.24
ReMixMatch	**19.10**±9.64	**5.44**±0.05	4.72±0.13	**44.28**±2.06	**27.43**±0.31	**23.03**±0.56	**3.34**±0.20	**2.92**±0.48	2.65±0.08
FixMatch (RA)	**13.81**±3.37	**5.07**±0.65	**4.26**±0.05	48.85±1.75	28.29±0.11	**22.60**±0.12	**3.96**±2.17	**2.48**±0.38	**2.28**±0.11
FixMatch (CTA)	**11.39**±3.35	**5.07**±0.33	**4.31**±0.15	49.95±3.01	28.64±0.24	23.18±0.11	7.65±7.65	**2.64**±0.64	**2.36**±0.19

FixMatch: Simplifying Semi-Supervised Learning with Consistency and Confidence, Sohn et al., 2020, arXiv

Next Lecture

Tuesday

Lab
Unsupervised Learning

See you next class!