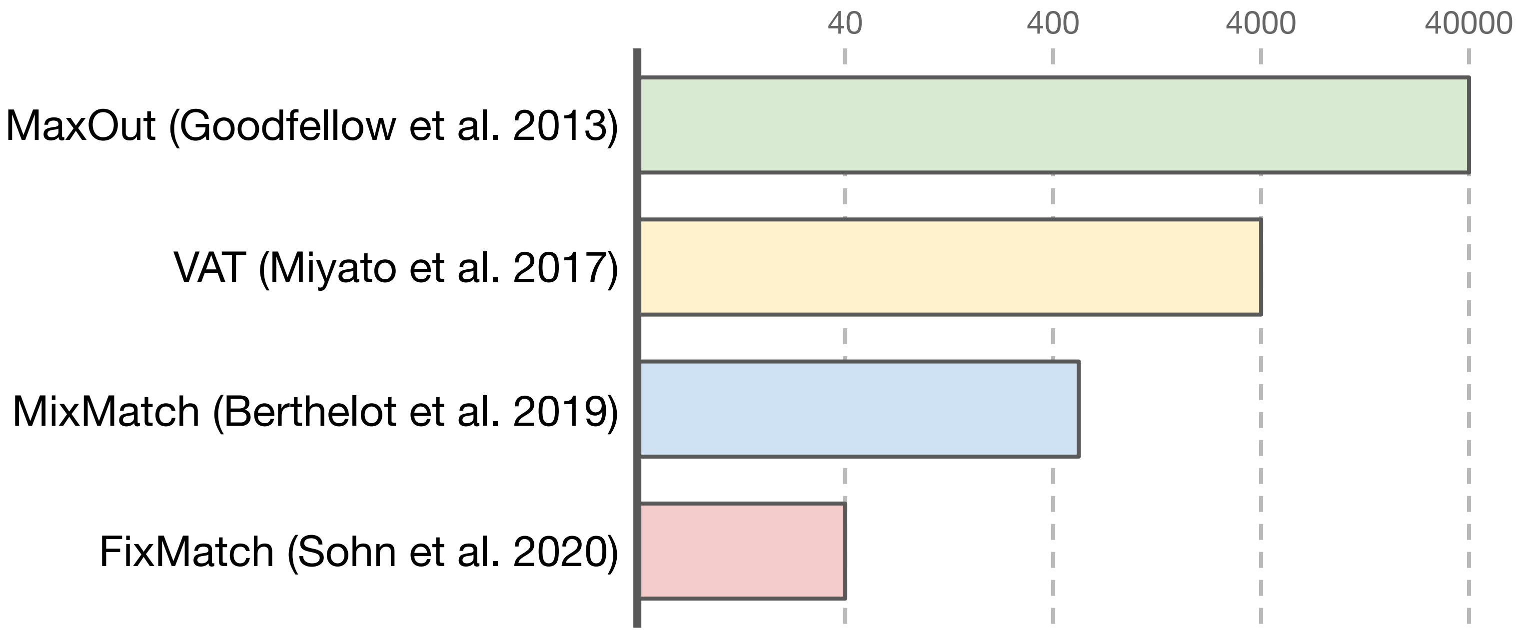
Explicit and Implicit Entropy Minimization in Proxy-Label-Based Semi-Supervised Learning

Colin Raffel

CVPR Workshop on Learning with Limited and Imperfect Data



Number of labels required to reach 90% accuracy on CIFAR-10

$$\mathbb{E}_{p(x,y)} - y \log p_\theta(y|x)$$

$$\mathbb{E}_{p(x)} -\hat{p}_\theta(y|x) \log p_\theta(y|x)$$

$$\mathbb{E}_{p(x)} -\underbrace{\hat{p}_\theta(y|x)}_{\text{“Proxy label”}} \log p_\theta(y|x)$$

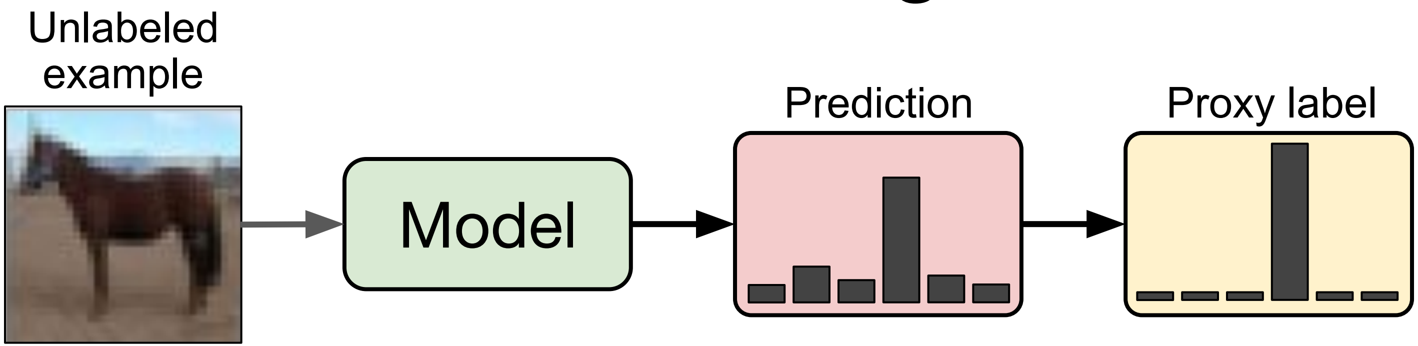
$$\mathbb{E}_{p(x)} -\hat{p}_\theta(y|x) \log p_\theta(y|x)$$

"Pseudo-label"

$$\mathbb{E}_{p(x)} - \hat{p}_\theta(y|x) \log p_\theta(y|x)$$

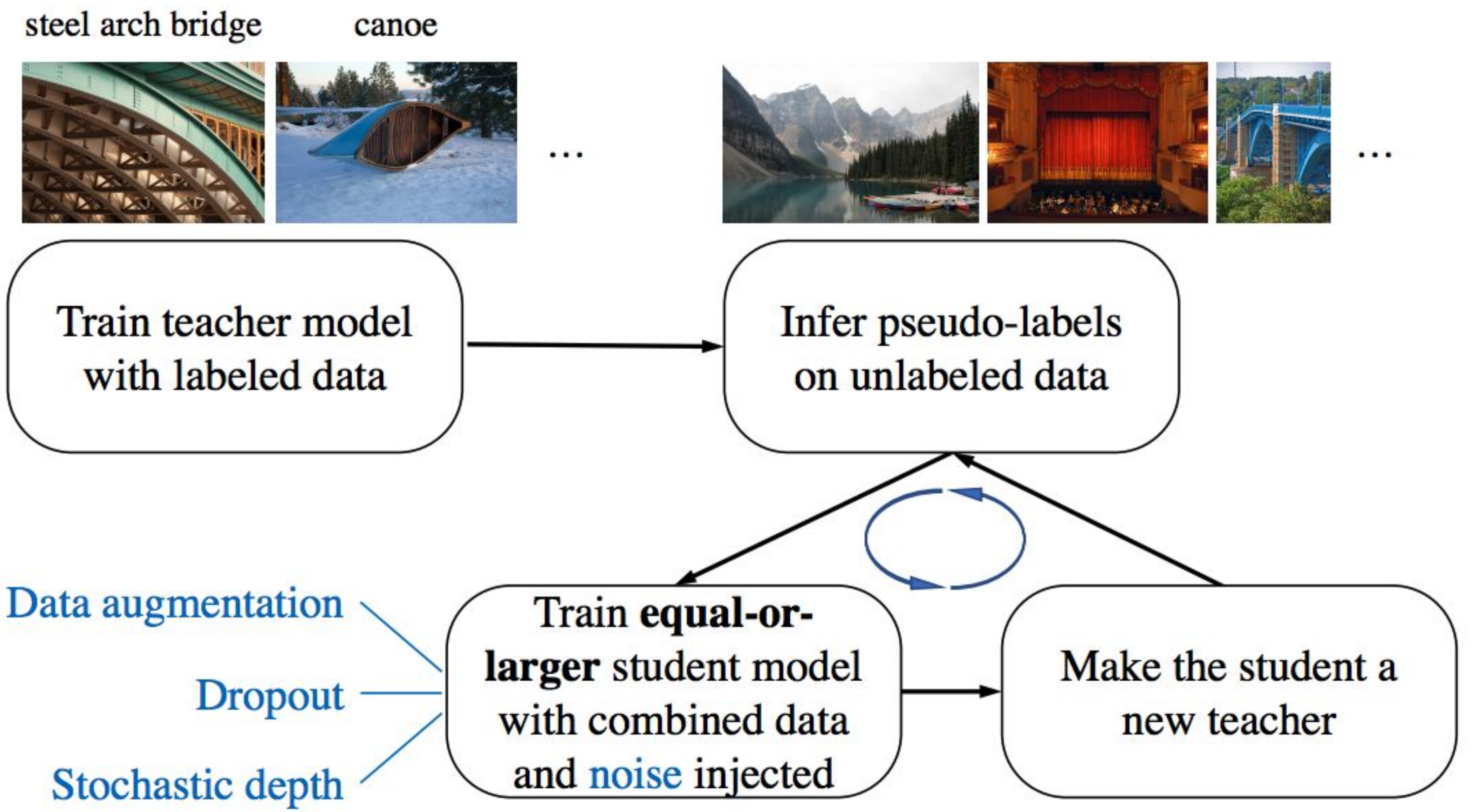
“Label guess”

Self-training



$$\mathbb{E}_{p(x)} -\hat{p}_\theta(y|x) \log p_\theta(y|x)$$

$$\hat{p}_\theta(y|x) = \arg\max_y [p_\theta(y|x)]$$



Self-training with Noisy Student improves ImageNet classification, Xie et al. 2019

Probability of Error of Some Adaptive Pattern-Recognition Machines

H. J. SCUDDER, III, MEMBER, IEEE

We will make the untaught machine from the same basic configuration as the taught machine and use the output of the machine $\hat{\theta}_n$ to iterate the estimate, instead of teaching the machine with the correct observation θ_n each time.

$$\mathbb{E}_{p(x)} -\hat{p}_\theta(y|x) \log p_\theta(y|x)$$

$$\hat{p}_\theta(y|x) = \arg\max_y [p_\theta(y|x)]$$

$$\mathbb{E}_{p(x)} -\hat{p}_\theta(y|x) \log p_\theta(y|x)$$

$$\hat{p}_\theta(y|x) = \cancel{\arg\max_y}[p_\theta(y|x)]$$

$$\mathbb{E}_{p(x)} -\hat{p}_\theta(y|x) \log p_\theta(y|x)$$

$$\hat{p}_\theta(y|x) = \cancel{\arg\max_y}[p_\theta(y|x)]$$

$$\hat{p}_\theta(y|x) = p_\theta(y|x)$$

$$\mathbb{E}_{p(x)} -\hat{p}_\theta(y|x) \log p_\theta(y|x)$$

$$\hat{p}_\theta(y|x) = \cancel{\arg\max_y}[p_\theta(y|x)]$$

$$\hat{p}_\theta(y|x) = p_\theta(y|x)$$

$$\mathbb{E}_{p(x)} -p_\theta(y|x) \log p_\theta(y|x)$$

$$\mathbb{E}_{p(x)} -\hat{p}_\theta(y|x) \log p_\theta(y|x)$$

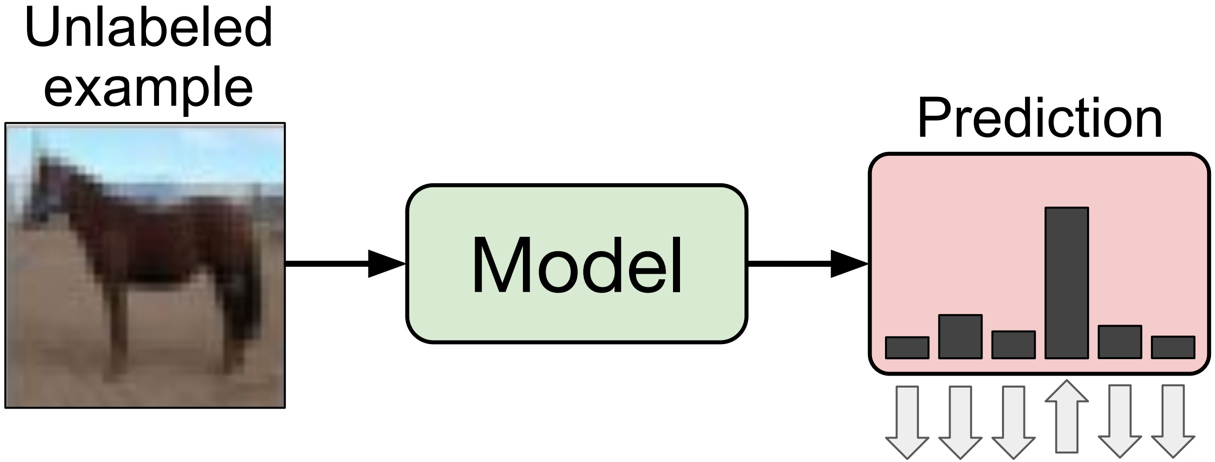
$$\hat{p}_\theta(y|x) = \cancel{\arg\max_y}[p_\theta(y|x)]$$

$$\hat{p}_\theta(y|x) = p_\theta(y|x)$$

$$\mathbb{E}_{p(x)} -p_\theta(y|x) \log p_\theta(y|x)$$

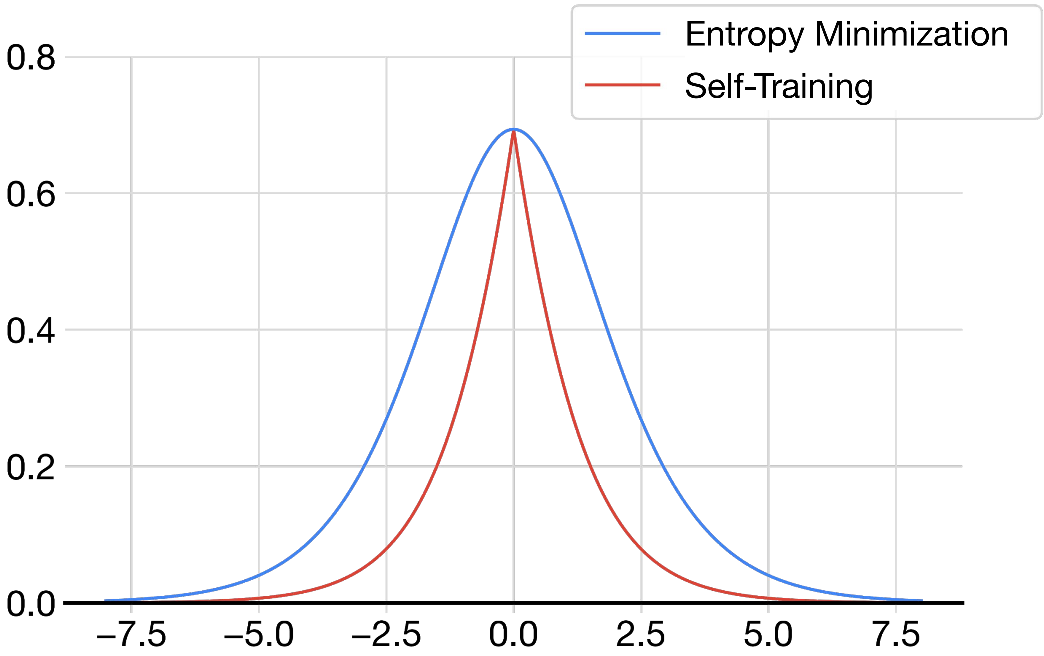
Entropy!

Entropy Minimization



$$\mathbb{E}_{p(x)} -\hat{p}_\theta(y|x) \log p_\theta(y|x)$$

$$\hat{p}_\theta(y|x) = p_\theta(y|x)$$


Entropy Minimization
Self-Training
0.8
0.6
0.4
0.2
0.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5

$$\begin{aligned}
\mathcal{I}(c\,;\mathbf{x}) &= \iint dc\,d\mathbf{x}\,p(c,\mathbf{x})\log\frac{p(c,\mathbf{x})}{p(c)p(\mathbf{x})} && (4)\\
&= \int d\mathbf{x}\,p(\mathbf{x})\int dc\,p(c|\mathbf{x})\log\frac{p(c|\mathbf{x})}{p(c)} && (5)\\
&= \int d\mathbf{x}\,p(\mathbf{x})\int dc\,p(c|\mathbf{x})\log\frac{p(c|\mathbf{x})}{\int d\mathbf{x}\,p(\mathbf{x})p(c|\mathbf{x})} && (6)
\end{aligned}$$

The elements of this expression are separately recognizable:

$\int d\mathbf{x}\,p(\mathbf{x})(\cdot)$ is equivalent to an average over a training set $\frac{1}{N_{ts}}\sum_{ts}(\cdot)$;

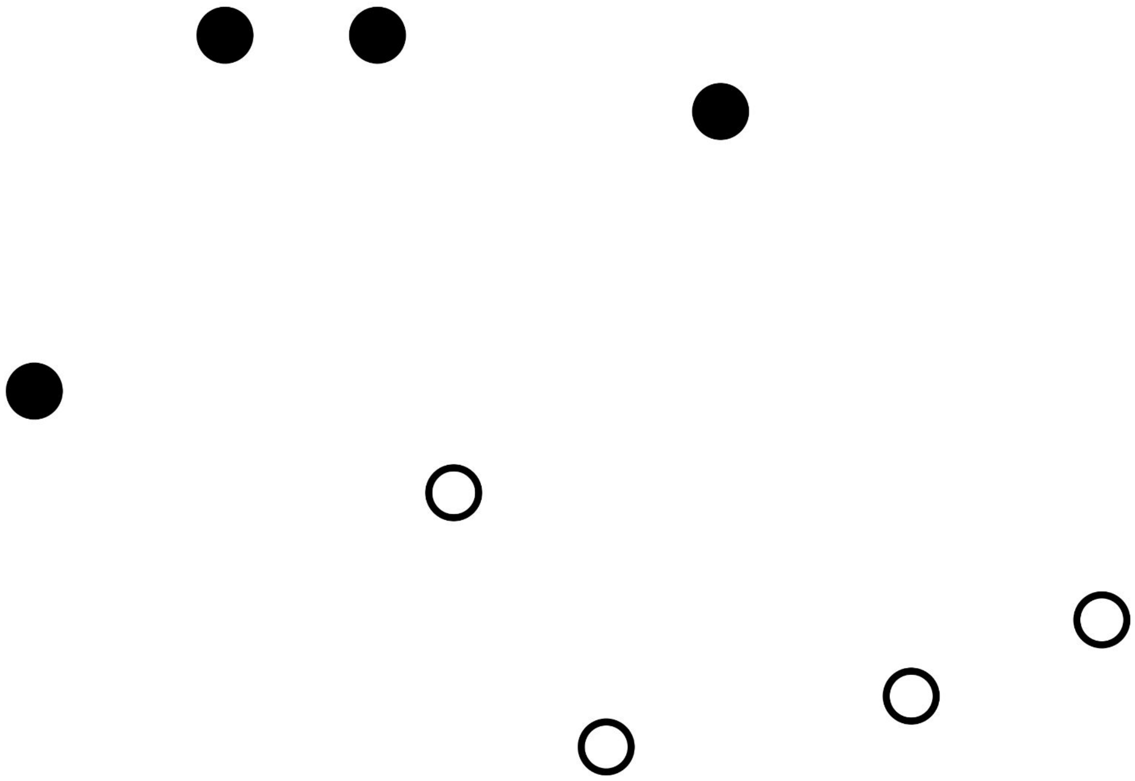
$p(c|\mathbf{x})$ is simply the network output y_c;

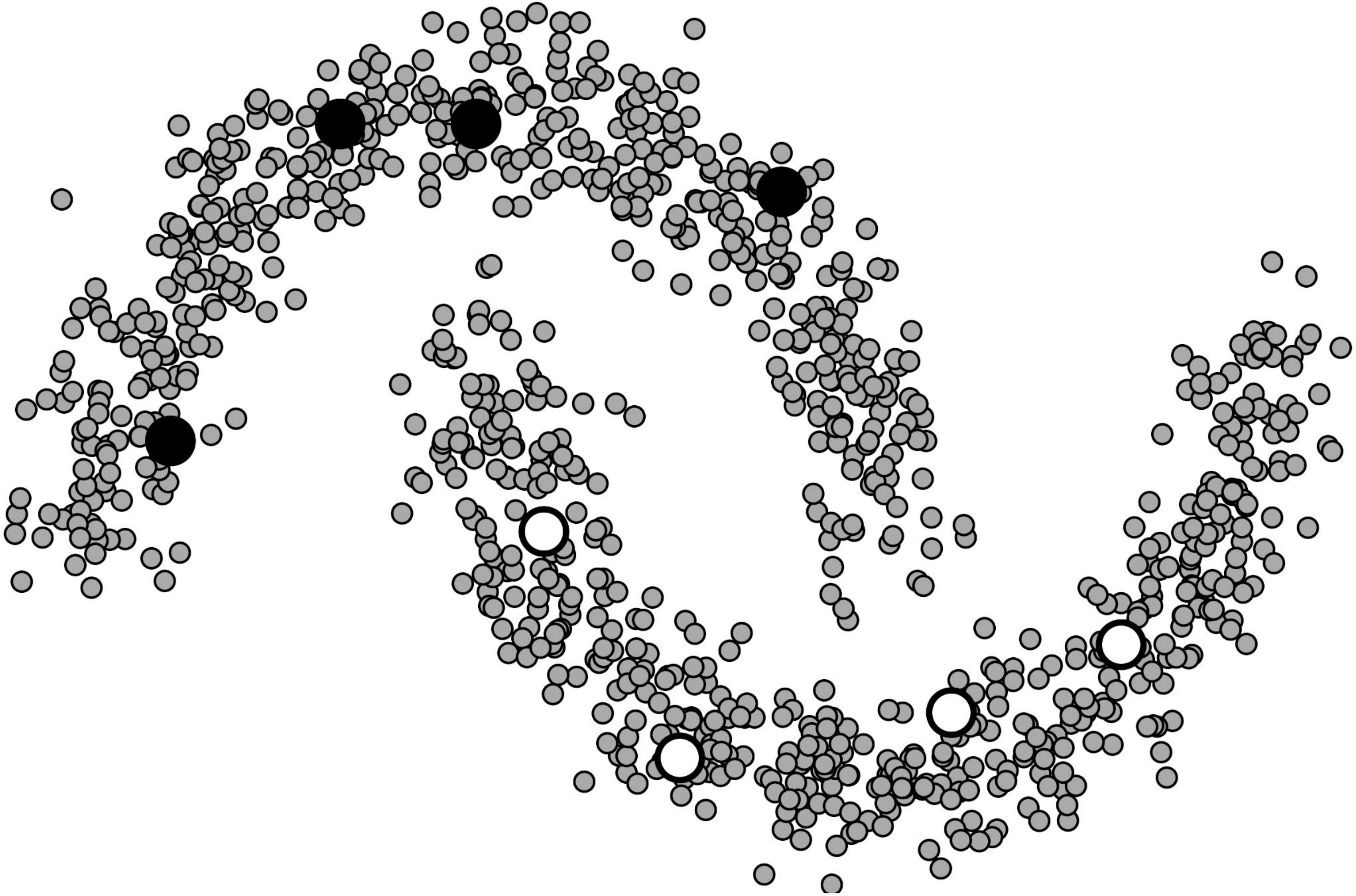
$\int dc\,(\cdot)$ is a sum over the class labels and corresponding network outputs.

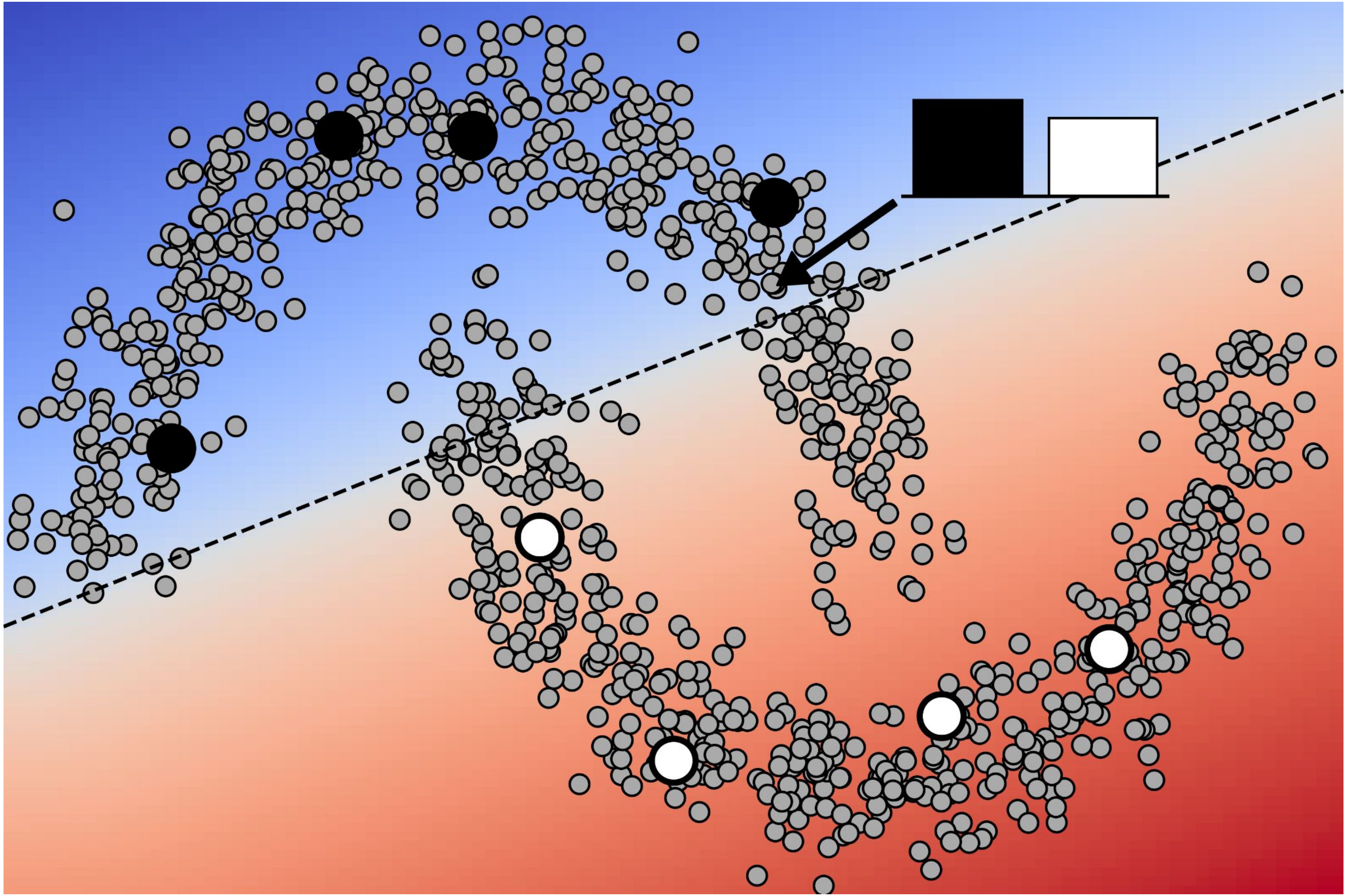
Hence:

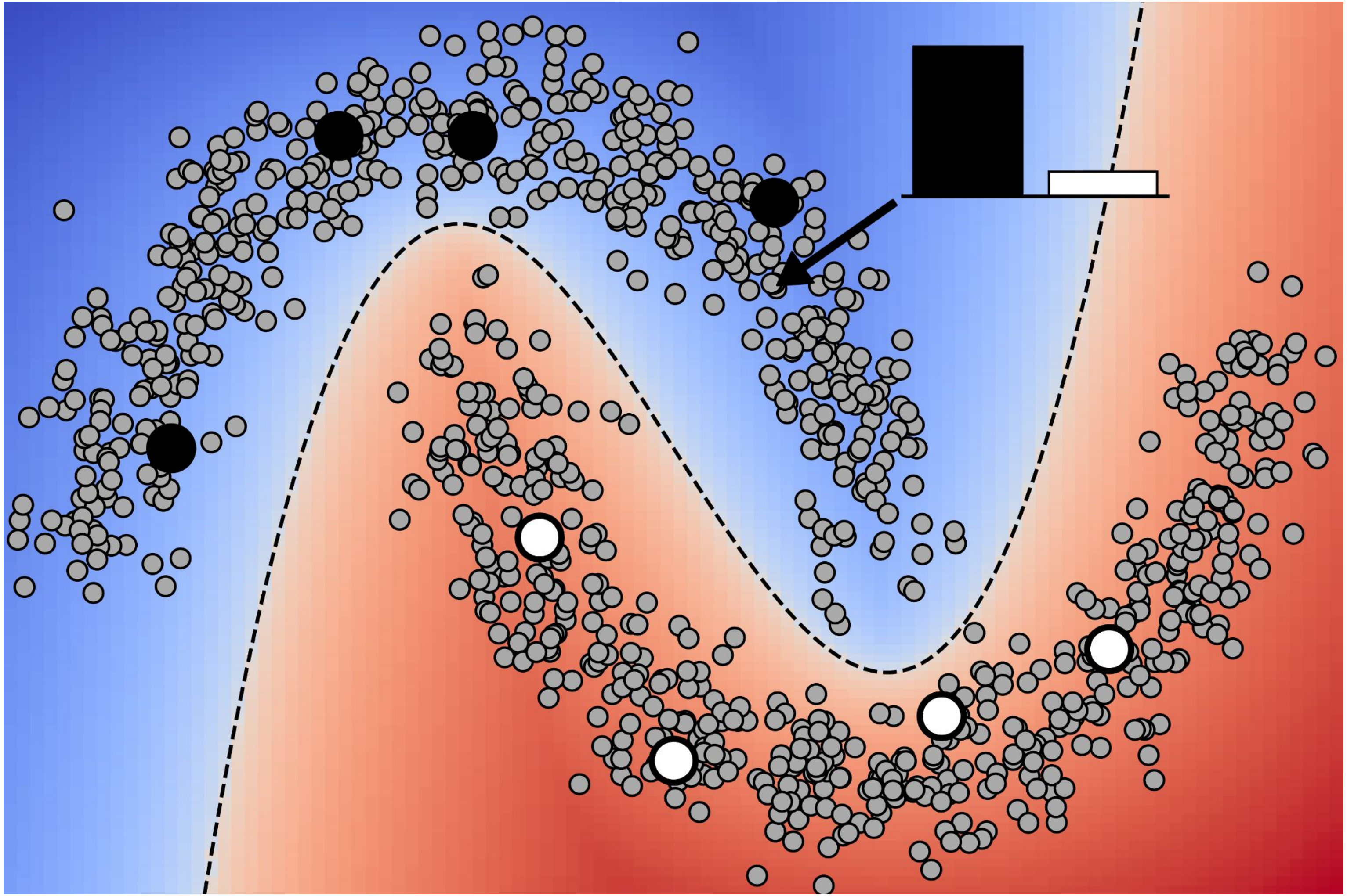
$$\begin{aligned}
\mathcal{I}(c\,;\mathbf{x}) &= \frac{1}{N_{ts}}\sum_{ts}\sum_{i=1}^{N_c} y_i\log\frac{y_i}{\overline{y}_i} && (7)\\
&= -\sum_{i=1}^{N_c}\overline{y}_i\log\overline{y}_i + \frac{1}{N_{ts}}\sum_{ts}\sum_{i=1}^{N_c} y_i\log y_i && (8)\\
&= \mathcal{H}(\overline{\mathbf{y}}) - \overline{\mathcal{H}(\mathbf{y})} && (9)
\end{aligned}$$

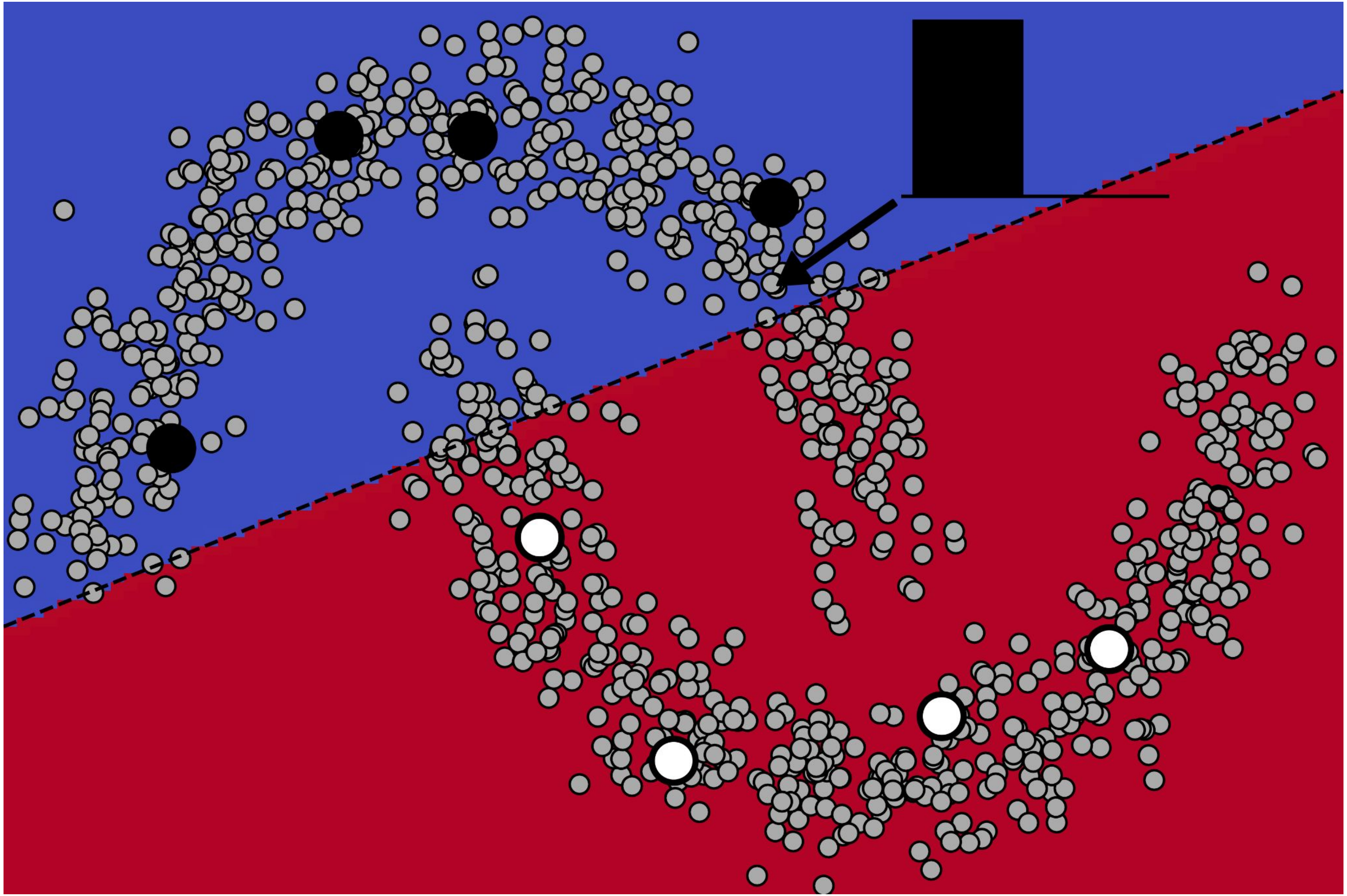
Unsupervised Classifiers, Mutual Information and Phantom Targets, Bridle et al. 1991

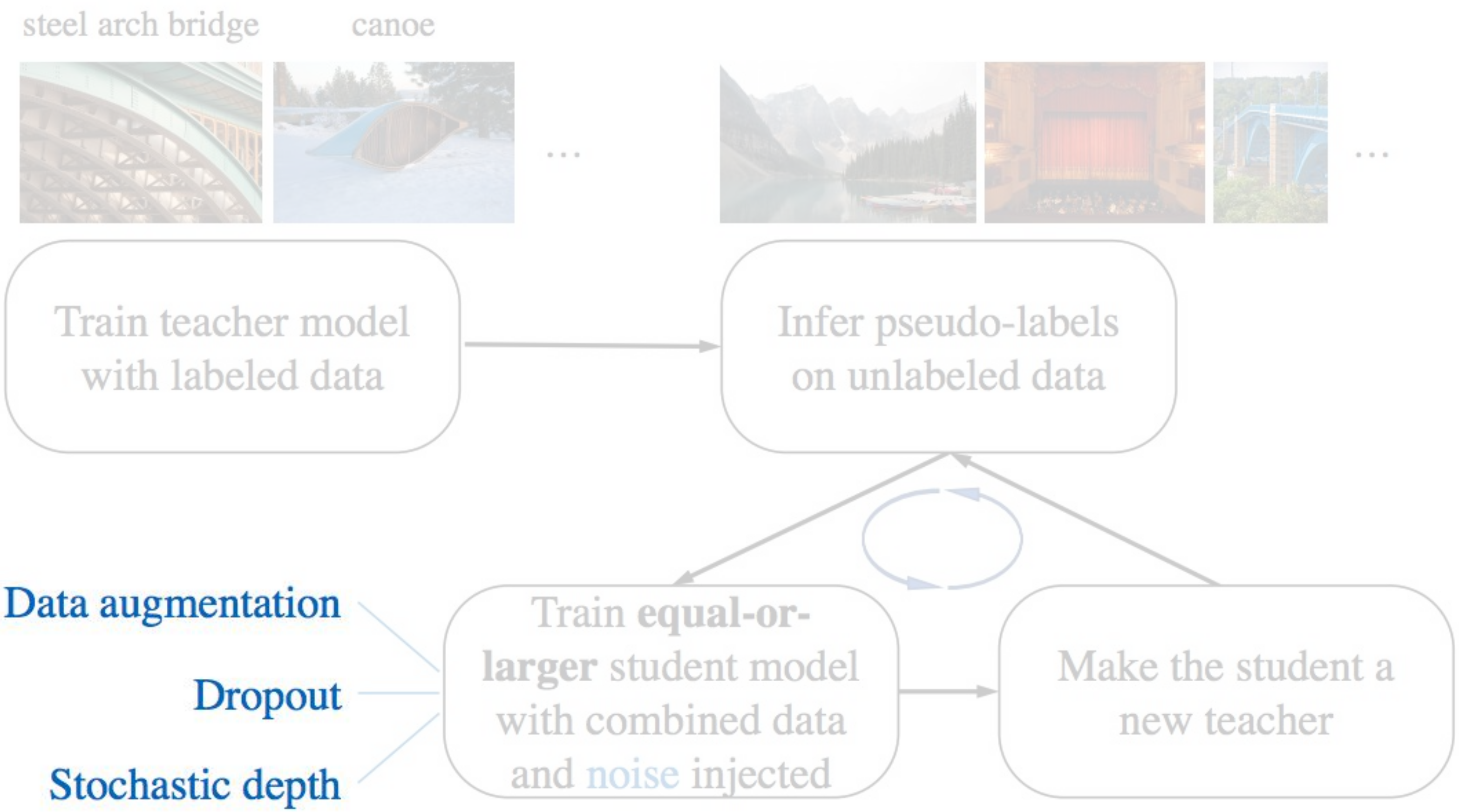









[redacted]



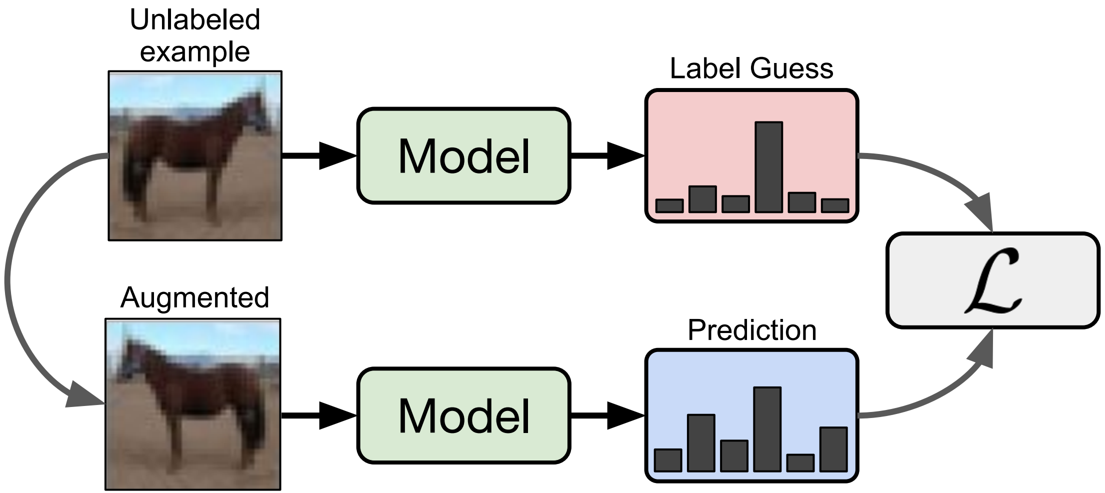
Self-training with Noisy Student improves ImageNet classification, Xie et al. 2019

Pseudo-Labeling and Confirmation Bias in Deep Semi-Supervised Learning

Eric Arazo, Diego Ortego, Paul Albert, Noel E. O'Connor, Kevin McGuinness
Insight Centre for Data Analytics, Dublin City University (DCU)
{eric.arazo, diego.ortego}@insight-centre.org

Experiments show that this naive pseudo-labeling is limited by confirmation bias as prediction errors are fit by the network. To deal with this issue, we propose to use mixup augmentation [25] as an effective regularization that helps calibrate deep neural networks [26] and, therefore, alleviates confirmation bias.

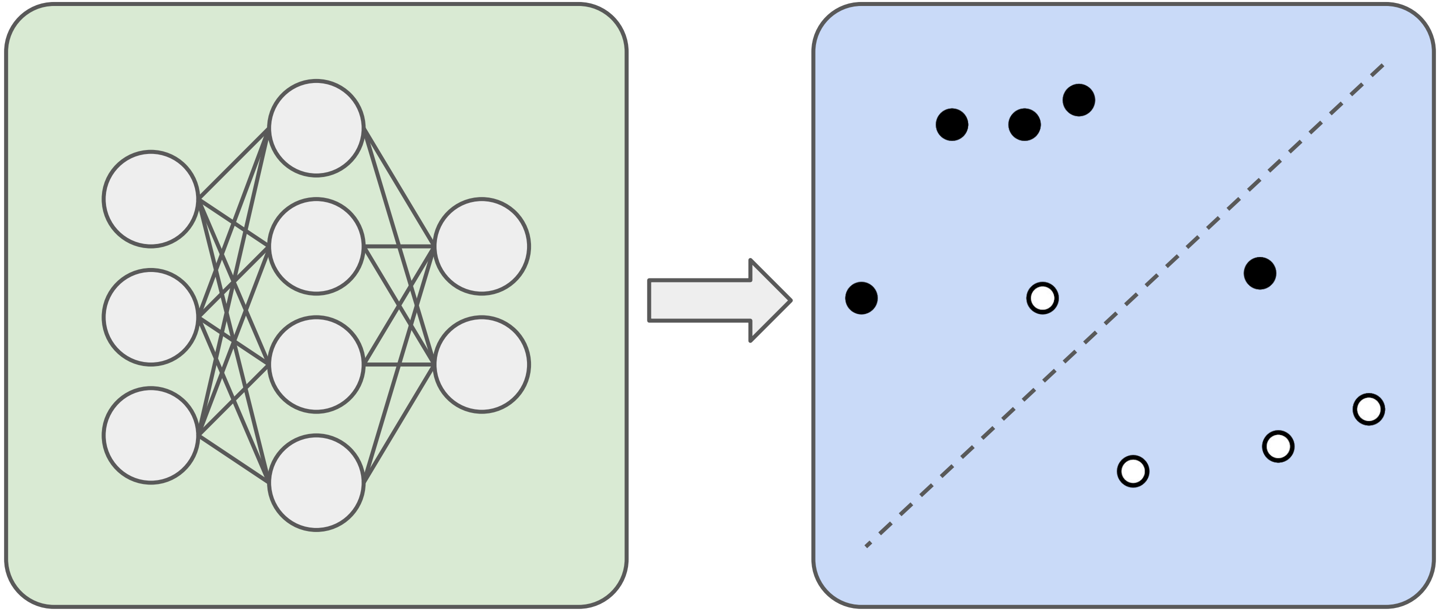
Consistency Regularization

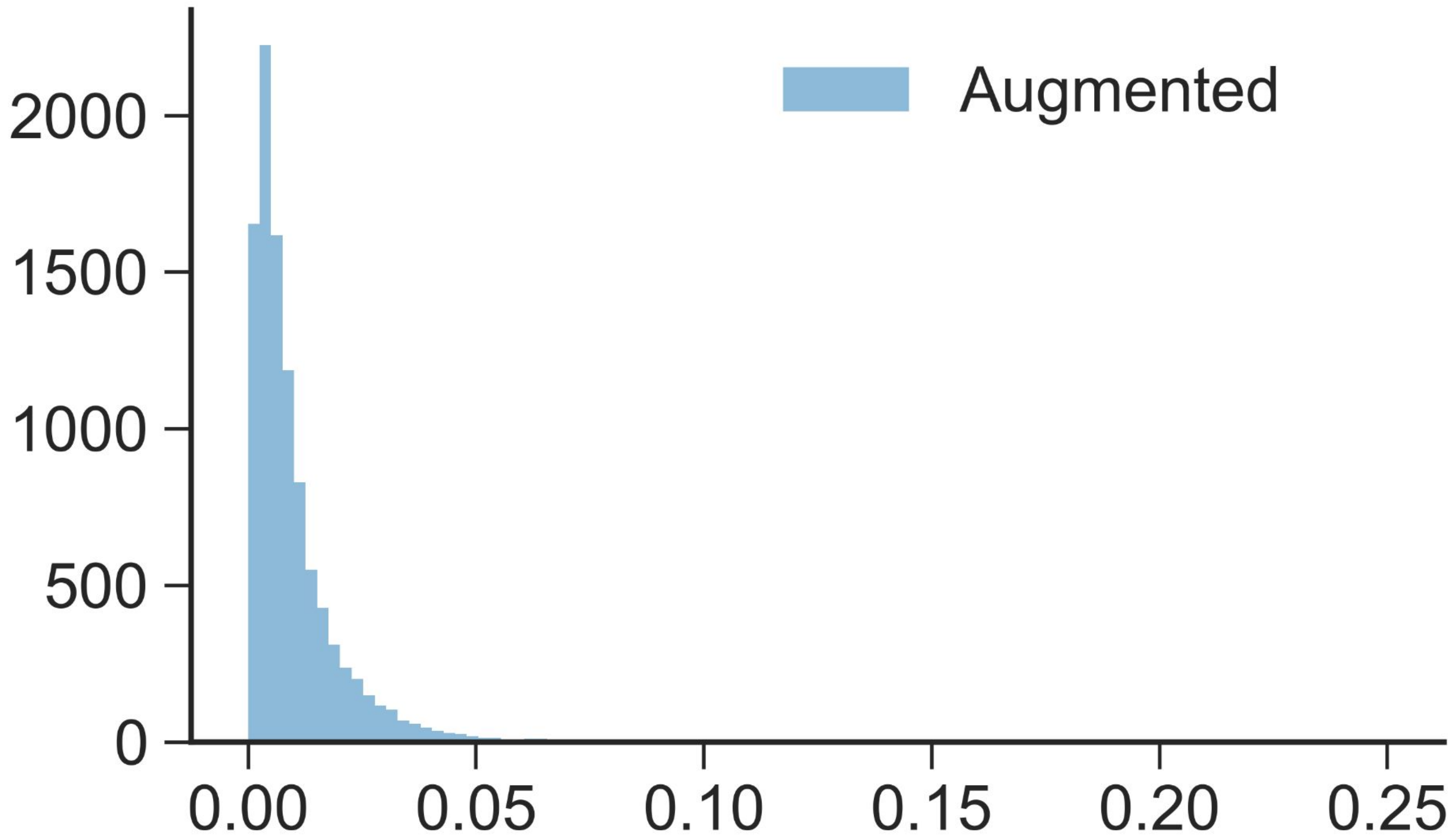


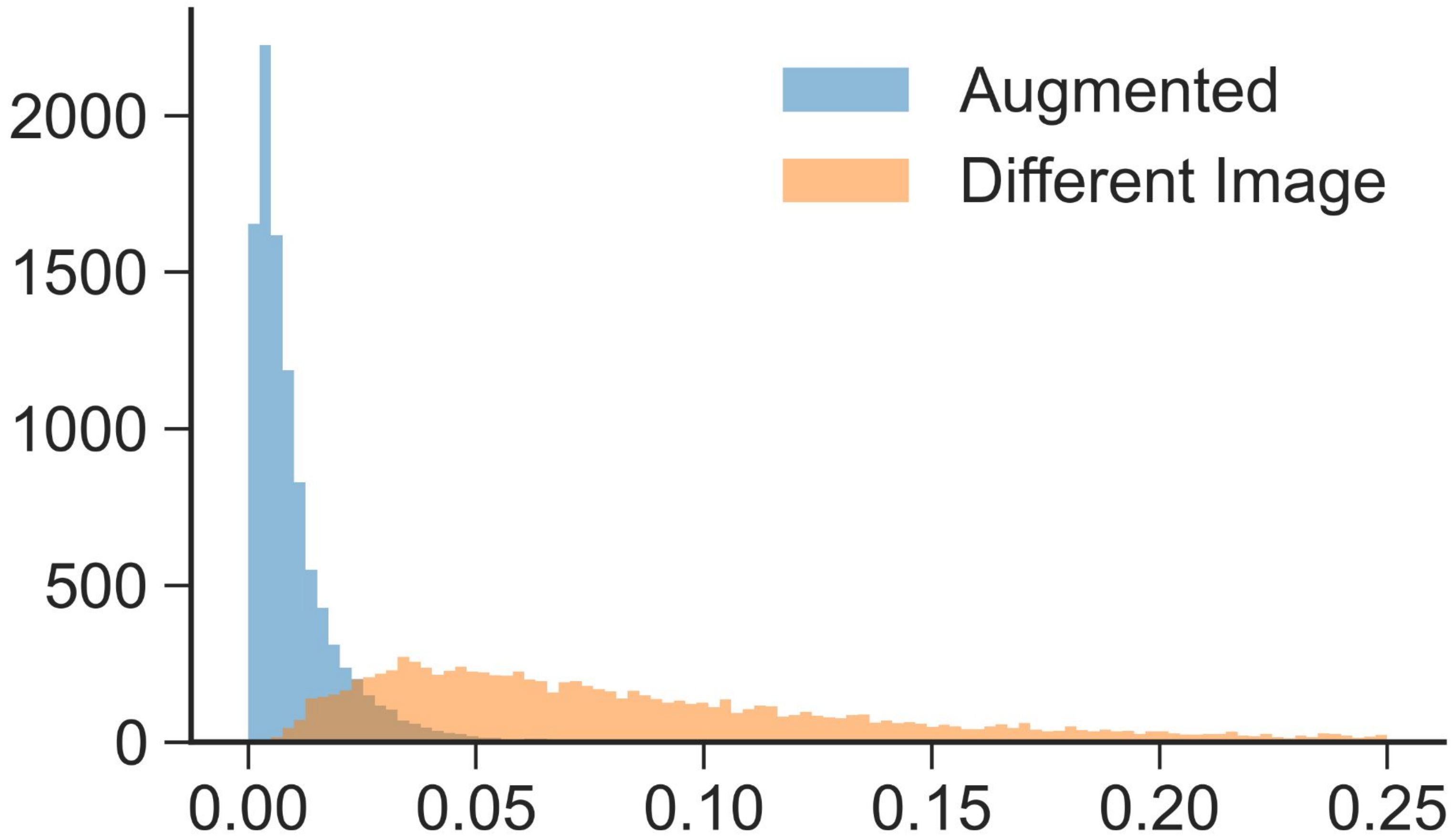
$$\mathbb{E}_{p(x)} - \hat{p}_\theta(y|x) \log p_\theta(y|x)$$

$$\hat{p}_\theta(y|x) = \hat{p}_\theta(y|x')$$

Learned feature extraction → *Classifier*




Augmented
2000
1500
1000
500
0
0.00
0.05
0.10
0.15
0.20
0.25



Augmented
Different Image
2000
1500
1000
500
0
0.00
0.05
0.10
0.15
0.20
0.25

craffel@gmail.com