Active Learning for DNNs

2019-06-13

Recent Papers

Cost-effective active learning for deep image classification TCSVT-2016

Active and continuous exploration with deep neural networks and Expected Model Output Changes

NIPS-2016

Suggestive Annotation A Deep Active Learning Framework for Biomedical Image Segmentation 2017

Deep bayesian active learning with image data ICML-2017 Yarin Gal，Zoubin Ghahramani

Active learning for convolutional neural networks： A core-set approach ICLR-2018

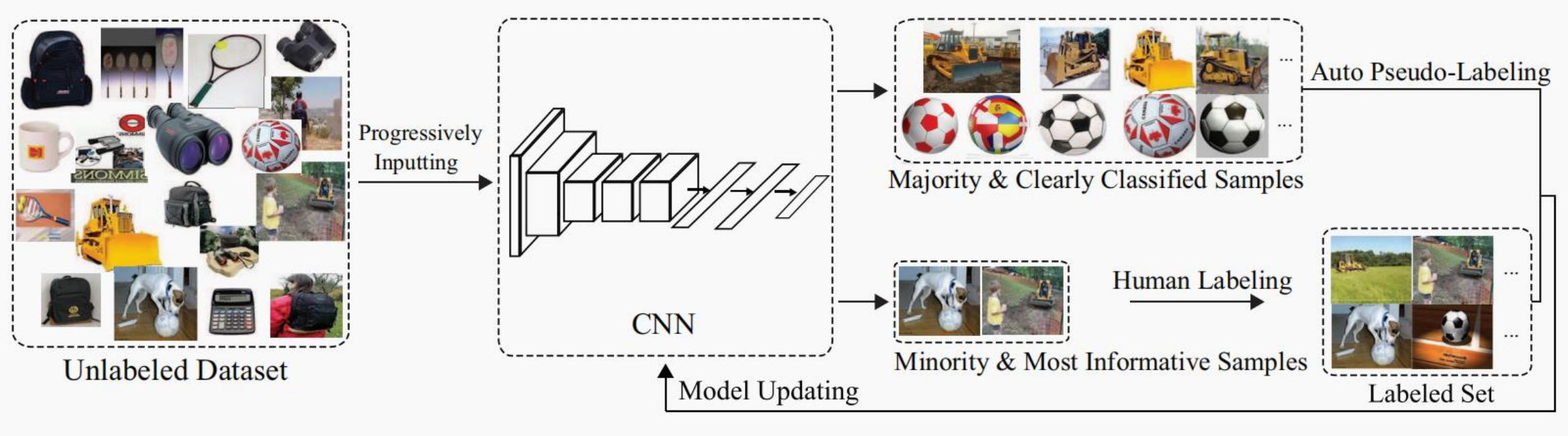
The power of ensembles for active learning in image classification CVPR-2018

Learning Loss for Active Learning CVPR-2019

Discriminative Active Learning ICLR2019被拒

Bayesian Generative Active Deep Learning ICML2019

Cost-effective active learning for deep image classification



Informative Sample Annotating:

Least confidence Margin sampling Entropy

Based the softmax output of the CNN !

High Confidence Sample Pseudo-labeling:

assign clearly predicted pseudo-labels to them

$$j^* = \arg\max_j p(y_i = j | x_i; \mathcal{W}),$$

$$y_i = \begin{cases} j^*, & en_i < \delta, \\ 0, & \text{otherwise.} \end{cases}$$

Expected Model Output Changes

- Selecting the instance that would impart the greatest change to the current model if we knew its label.

Using a stochastic gradient approximation with just a single sample to estimate model parameter updates:

$$\boldsymbol{\theta}' - \boldsymbol{\theta} \approx \gamma \nabla_{\boldsymbol{\theta}} \bar{\mathcal{L}}(\boldsymbol{\theta}; \mathcal{D} \cup (\mathbf{x}', y')) \approx \gamma \nabla_{\boldsymbol{\theta}} \bar{\mathcal{L}}(\boldsymbol{\theta}; (\mathbf{x}', y'))$$

Approximate the model output change by using a first-order approximation

$$\|f(\mathbf{x}; \boldsymbol{\theta}') - f(\mathbf{x}; \boldsymbol{\theta})\|_1 \approx \|\nabla_{\boldsymbol{\theta}} f(\mathbf{x}; \boldsymbol{\theta})^T (\boldsymbol{\theta}' - \boldsymbol{\theta})\|_1 \overset{(3)}{\approx} \gamma \|\nabla_{\boldsymbol{\theta}} f(\mathbf{x}; \boldsymbol{\theta})^T \nabla_{\boldsymbol{\theta}} \bar{\mathcal{L}}(\boldsymbol{\theta}; (\mathbf{x}', y'))\|_1$$

Suggestive Annotation

1. Uncertainty

Select top K uncertainty scores based on the softmax output

$S_c \in S_u, \quad |S_c| = K$

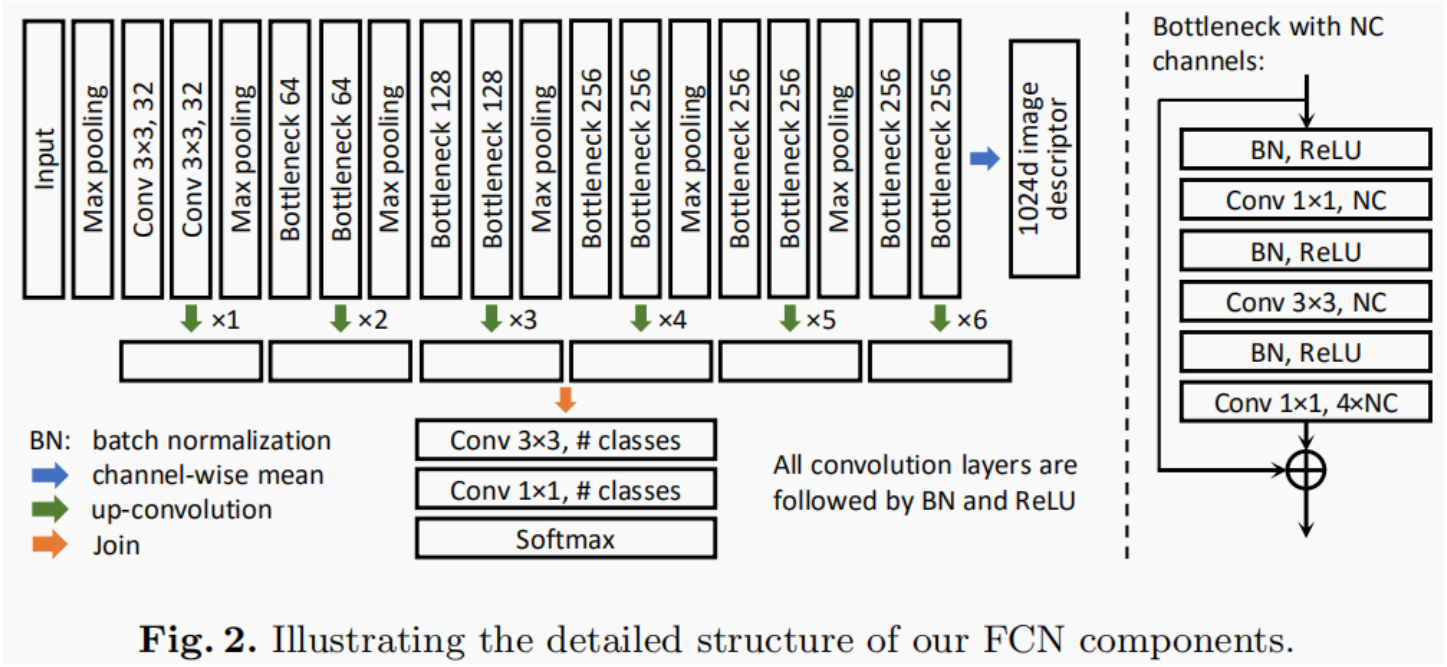


Fig. 2. Illustrating the detailed structure of our FCN components.

2. Representativeness

First define the representativeness of S_a for an image $I_x \in S_u$

as: $f(S_a, I_x) = \max_{I_i \in S_a} sim(I_i, I_x)$

Define the representativeness of S_a for S_u as: $F(S_a, S_u) = \sum_{I_j \in S_u} f(S_a, I_j)$

Finding $S_a \in S_c$ that $maximizes\ F(S_a, S_u)$

Deep Bayesian Active Learning

- Bayesian Convolutional Neural Networks

Gal, Yarin. *Uncertainty in Deep Learning*. PhD thesis, University of Cambridge, 2016.

Gal, Yarin and Ghahramani, Zoubin. Bayesian convolutional neural networks with Bernoulli approximate variational inference. *ICLR workshop track*, 2016a.

Gal, Yarin and Ghahramani, Zoubin. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. *ICML*, 2016b.

$$
\begin{aligned}
p(y = c|\mathbf{x}, \mathcal{D}_{\text{train}}) &= \int p(y = c|\mathbf{x}, \boldsymbol{\omega}) p(\boldsymbol{\omega}|\mathcal{D}_{\text{train}}) \mathrm{d}\boldsymbol{\omega} \\
&\approx \int p(y = c|\mathbf{x}, \boldsymbol{\omega}) q_\theta^*(\boldsymbol{\omega}) \mathrm{d}\boldsymbol{\omega} \\
&\approx \frac{1}{T} \sum_{t=1}^{T} p(y = c|\mathbf{x}, \widehat{\boldsymbol{\omega}}_t)
\end{aligned}
$$

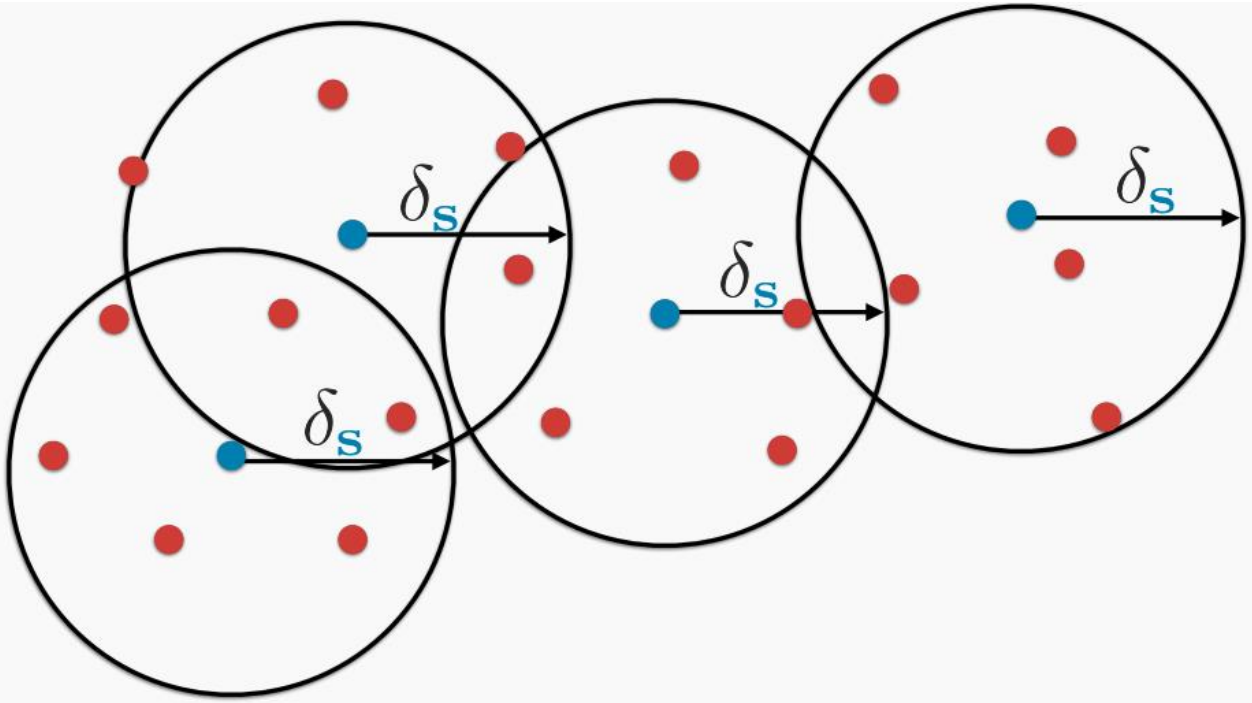
with $\widehat{\boldsymbol{\omega}}_t \sim q_\theta^*(\boldsymbol{\omega})$, where $q_\theta(\boldsymbol{\omega})$ is the Dropout distribution

- Acquisition Functions and their Approximations

$$
\begin{aligned}
&\mathbb{H}[y|\mathbf{x}, \mathcal{D}_{\text{train}}] := \\
&\quad - \sum_c p(y = c|\mathbf{x}, \mathcal{D}_{\text{train}}) \log p(y = c|\mathbf{x}, \mathcal{D}_{\text{train}}).
\end{aligned}
$$

A core-set approach

- Choosing set of points such that a model learned over the selected subset is competitive for the remaining data points.



Algorithm 1 k-Center-Greedy

Input: data $\mathbf{x}_i$, existing pool $\mathbf{s}^0$ and budget b
Initialize $\mathbf{s} = \mathbf{s}^0$
repeat
  $u = \arg\max_{i \in [n] \setminus \mathbf{s}} \min_{j \in \mathbf{s}} \Delta(\mathbf{x}_i, \mathbf{x}_j)$
  $\mathbf{s} = \mathbf{s} \cup \{u\}$
until $|\mathbf{s}| = b + |\mathbf{s}^0|$
return $\mathbf{s} \setminus \mathbf{s}^0$

Figure 1: **Visualization of the Theorem 1.** Consider the set of selected points $\mathbf{s}$ and the points in the remainder of the dataset $[n] \setminus \mathbf{s}$, our results shows that if $\mathbf{s}$ is the $\delta_{\mathbf{s}}$ cover of the dataset, $\left| \frac{1}{n} \sum_{i \in [n]} l(\mathbf{x}_i, y_i, A_{\mathbf{s}}) - \frac{1}{|\mathbf{s}|} \sum_{j \in \mathbf{s}} l(\mathbf{x}_j, y_j; A_{\mathbf{s}}) \right| \leq \mathcal{O}(\delta_{\mathbf{s}}) + \mathcal{O}\left(\sqrt{\frac{1}{n}}\right)$

Use the l_2 distance between activations of the final fully-connected layer as the distance.

A core-set approach

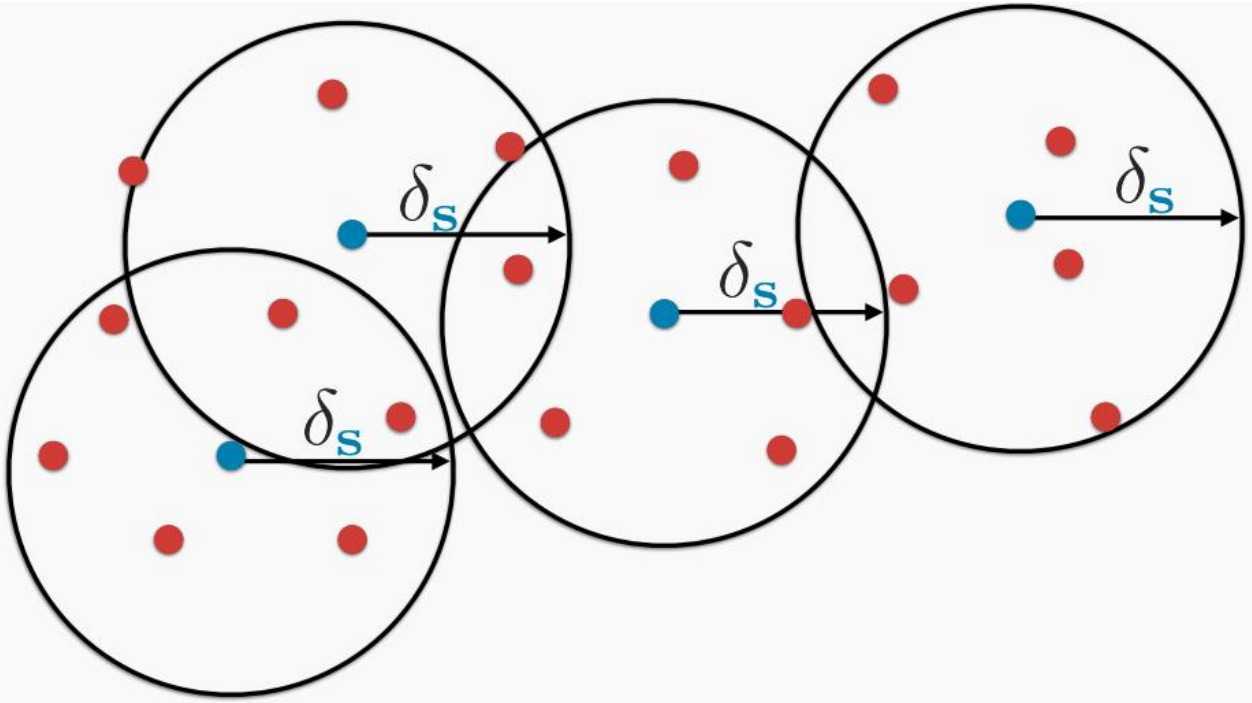
- Choosing set of points such that a model learned over the selected subset is competitive for the remaining data points.

$$E_{\mathbf{x},y\sim p_{\mathcal{Z}}}[l(\mathbf{x},y;A_{\mathbf{s}})] \leq \underbrace{\left|E_{\mathbf{x},y\sim p_{\mathcal{Z}}}[l(\mathbf{x},y;A_{\mathbf{s}})] - \frac{1}{n}\sum_{i\in[n]} l(\mathbf{x}_i,y_i;A_{\mathbf{s}})\right|}_{\text{Generalization Error}} + \underbrace{\frac{1}{|\mathbf{s}|}\sum_{j\in\mathbf{s}} l(\mathbf{x}_j,y_j;A_{\mathbf{s}})}_{\text{Training Error}}$$

$$+ \underbrace{\left|\frac{1}{n}\sum_{i\in[n]} l(\mathbf{x}_i,y_i;A_{\mathbf{s}}) - \frac{1}{|\mathbf{s}|}\sum_{j\in\mathbf{s}} l(\mathbf{x}_j,y_j;A_{\mathbf{s}}),\right|}_{\text{Core-Set Loss}}$$

$$\min_{\mathbf{s}^1:|\mathbf{s}^1|\leq b} \left|\frac{1}{n}\sum_{i\in[n]} l(\mathbf{x}_i,y_i;A_{\mathbf{s}^0\cup\mathbf{s}^1}) - \frac{1}{|\mathbf{s}^0+\mathbf{s}^1|}\sum_{j\in\mathbf{s}^0\cup\mathbf{s}^1} l(\mathbf{x}_j,y_j;A_{\mathbf{s}^0\cup\mathbf{s}^1})\right|$$

A core-set approach



Algorithm 1 k-Center-Greedy

Input: data $\mathbf{x}_i$, existing pool $\mathbf{s}^0$ and a budget b
Initialize $\mathbf{s} = \mathbf{s}^0$
repeat
  $u = \arg\max_{i \in [n] \setminus \mathbf{s}} \min_{j \in \mathbf{s}} \Delta(\mathbf{x}_i, \mathbf{x}_j)$
  $\mathbf{s} = \mathbf{s} \cup \{u\}$
until $|\mathbf{s}| = b + |\mathbf{s}^0|$
return $\mathbf{s} \setminus \mathbf{s}^0$

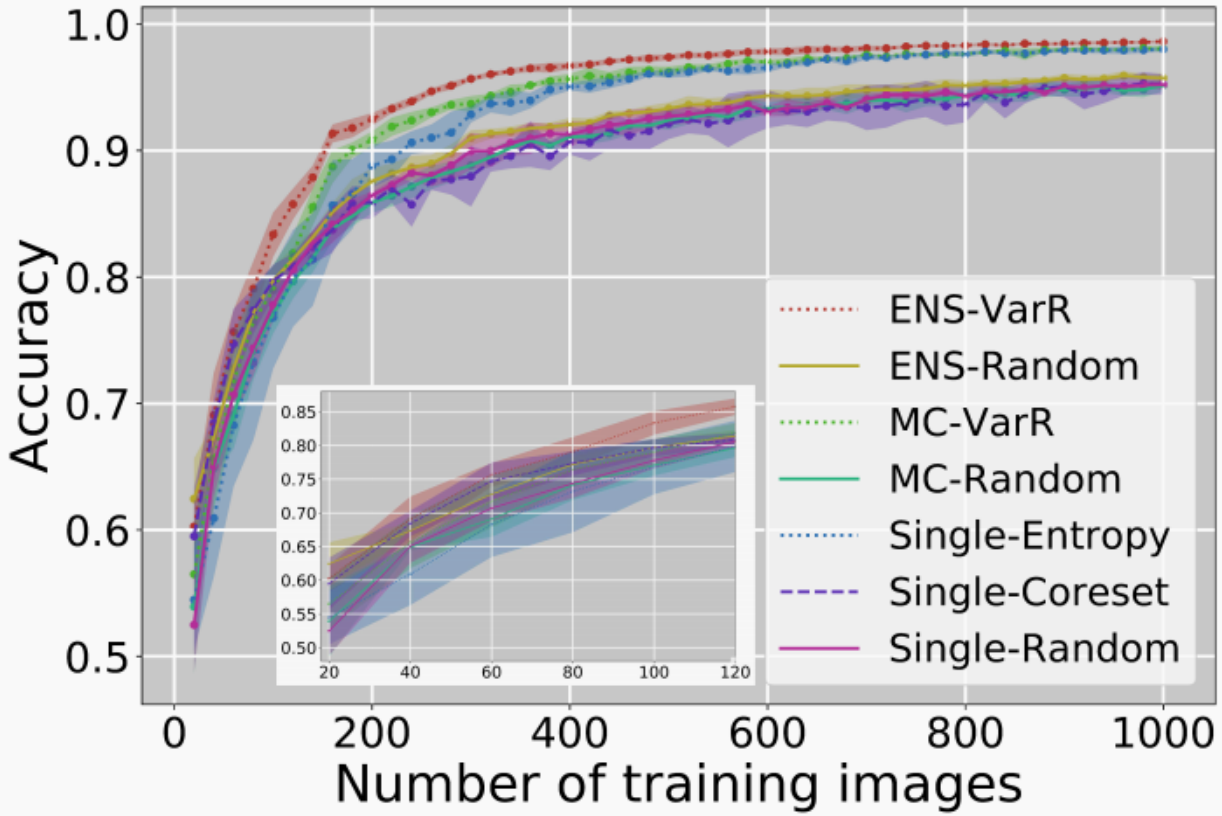
Figure 1: **Visualization of the Theorem 1.** Consider the set of selected points $\mathbf{s}$ and the points in the remainder of the dataset $[n] \setminus \mathbf{s}$, our results shows that if $\mathbf{s}$ is the $\delta_{\mathbf{s}}$ cover of the dataset, $\left| \frac{1}{n} \sum_{i \in [n]} l(\mathbf{x}_i, y_i, A_{\mathbf{s}}) - \frac{1}{|\mathbf{s}|} \sum_{j \in \mathbf{s}} l(\mathbf{x}_j, y_j; A_{\mathbf{s}}) \right| \leq \mathcal{O}(\delta_{\mathbf{s}}) + \mathcal{O}\left(\sqrt{\frac{1}{n}}\right)$

One of the critical design choices is the distance metric $\Delta(\cdot,\cdot)$. Use the l_2 distance between activations of the final fully-connected layer as the distance.

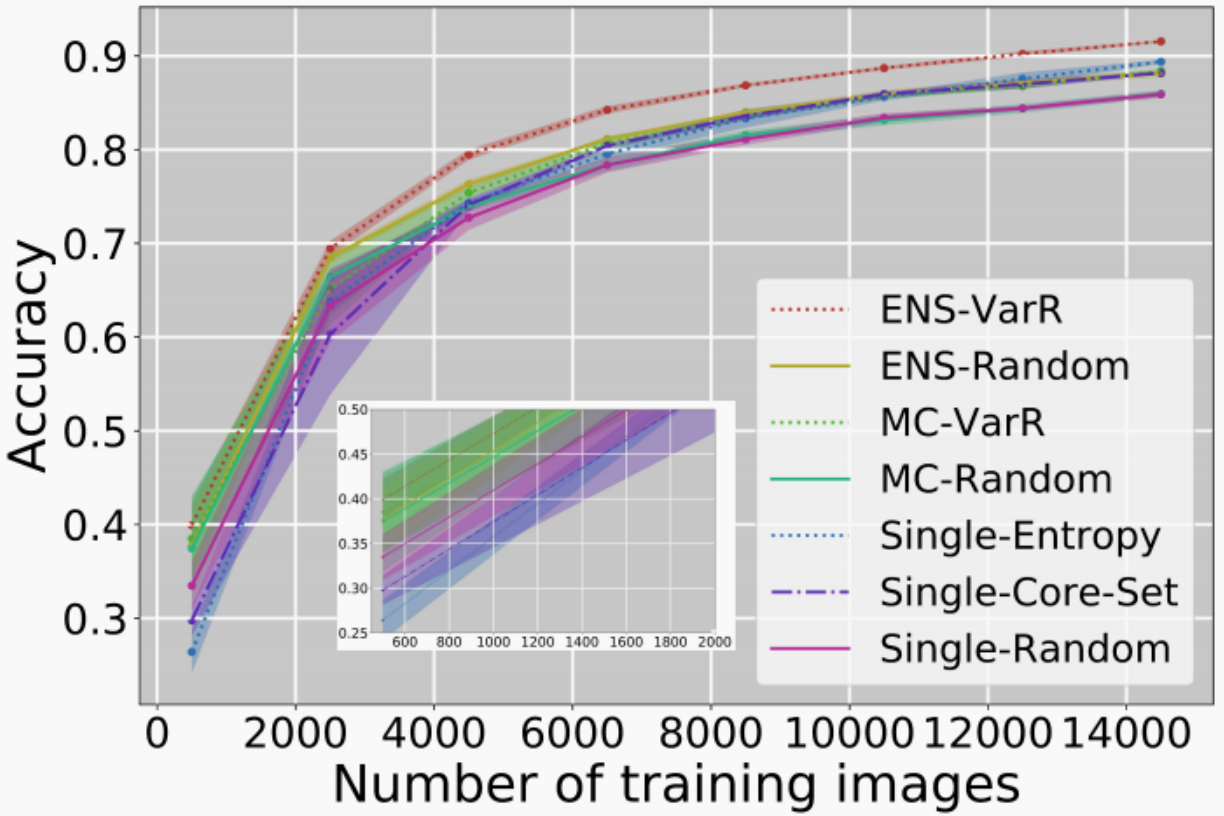
Ensembles for Active Learning

- The power of ensembles for active learning in image classification.

$$p(y = c|x, D_{\text{train}}) = \frac{1}{T}\sum_{t=1}^{T} p(y = c|x, w_t)$$



(a) MNIST on S-CNN



(b) CIFAR-10 on DenseNet

Ensembles for Active Learning

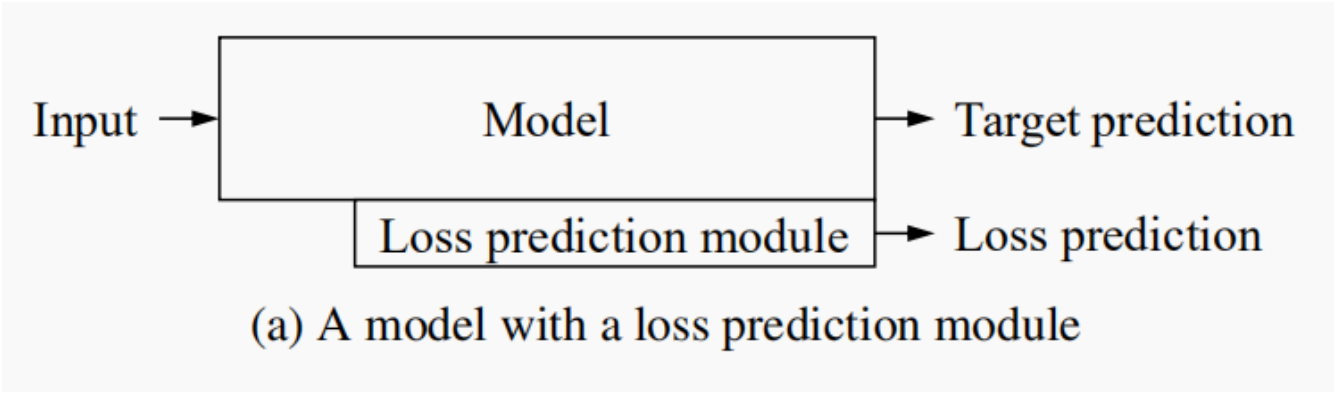
- Single-Core-Set perform worse than random
    1. A different network (原文为VGG16，该篇为K-CNN)，then the different feature representations provided by the various network architectures
    2. Perhaps the prevalence of outliers hindering the greedy core-set approach

- Comparing ensemble-based against Monte-Carlo Dropout performance
    1. **Number of networks** in the ensemble or forward passes in MC dropout
    2. **Model capacity** by reducing the number of neurons for the ensemble networks
    3. **Fixed initialization** to have the same initialization for all ensemble members
    4. **Fixed order within a mini-batch** to have the same order of images within a mini-batch for all ensemble members

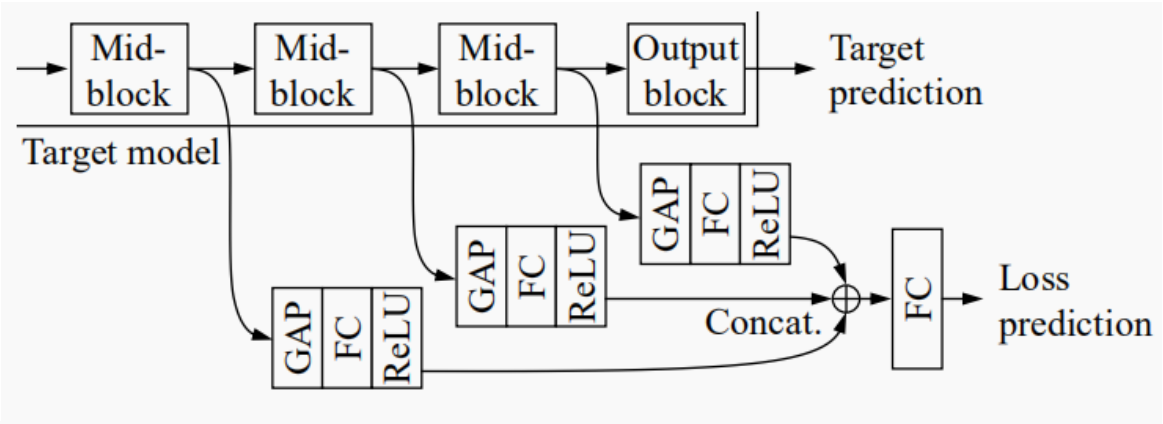
Fixed initialization, **Model capacity** has a significant effect on the results.

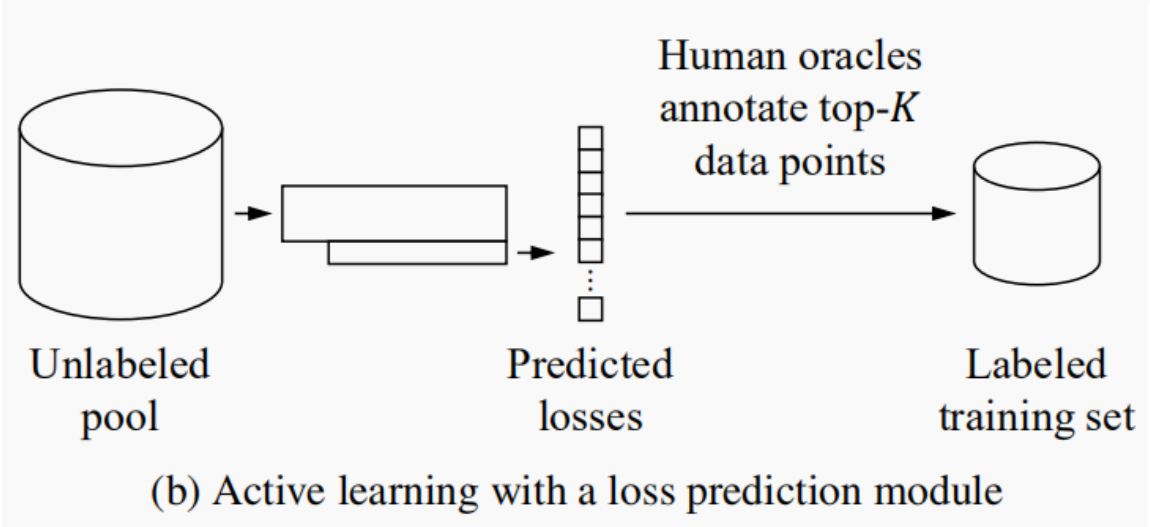
Learning Loss

- The selected data points would be more informative to the current model if it has high loss.



(a) A model with a loss prediction module





(b) Active learning with a loss prediction module

Discriminative Active Learning

- If the examples from the labeled set are indistinguishable from the unlabeled pool, then we have successfully represented the distribution with our labeled set.
- Trains a classifier to discriminate between the examples in the labeled and unlabeled data at each iteration

$$\psi: X \to \hat{X}$$

$$\hat{P}(y|\psi(x)) \, , y \in \{L, U\}$$

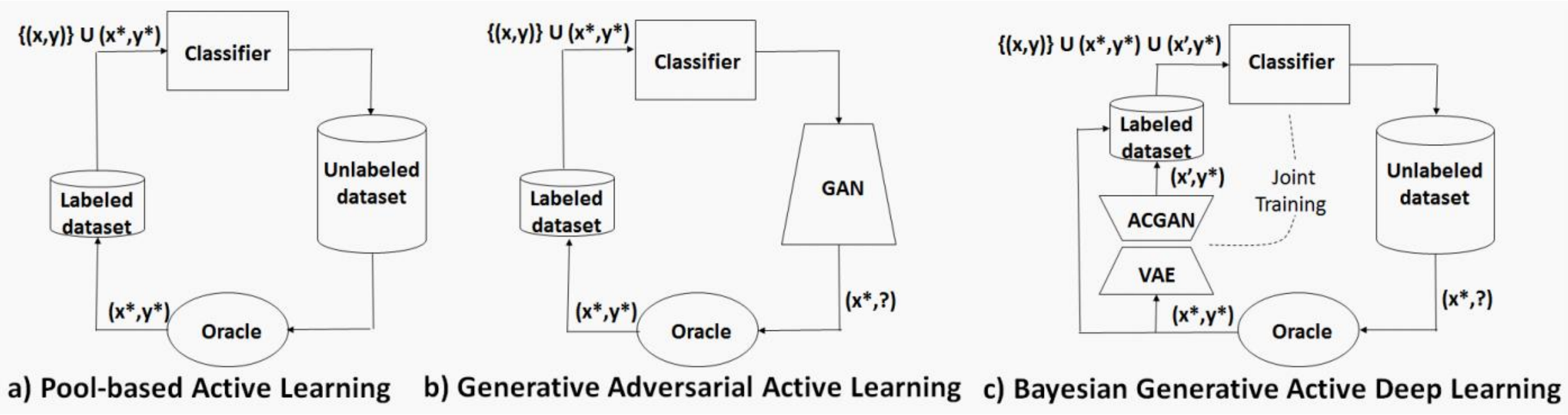
$$\underset{x \in U}{\operatorname{argmx}} \hat{P}(y = u|\psi(x))$$

Algorithm 1 Discriminative Active Learning

procedure DAL QUERY($\mathcal{U}, \mathcal{L}, K, n$) ▷ K is the total budget, n is the amount of mini-queries
  for i=1...n **do**
    $P \leftarrow TRAIN\_BINARY\_CLASSIFIER(\mathcal{U}, \mathcal{L})$
    for j=1...$\frac{K}{n}$ **do**
      $\hat{x} \leftarrow \underset{x \in \mathcal{U}}{\operatorname{argmax}} \, P(y = u|\Psi(x))$
      $\mathcal{L} \leftarrow \mathcal{L} \cup \hat{x}$
      $\mathcal{U} \leftarrow \mathcal{U} \backslash \hat{x}$
    end for
  end for
  return $\mathcal{U}, \mathcal{L}$
end procedure

Bayesian Generative Active Deep Learning

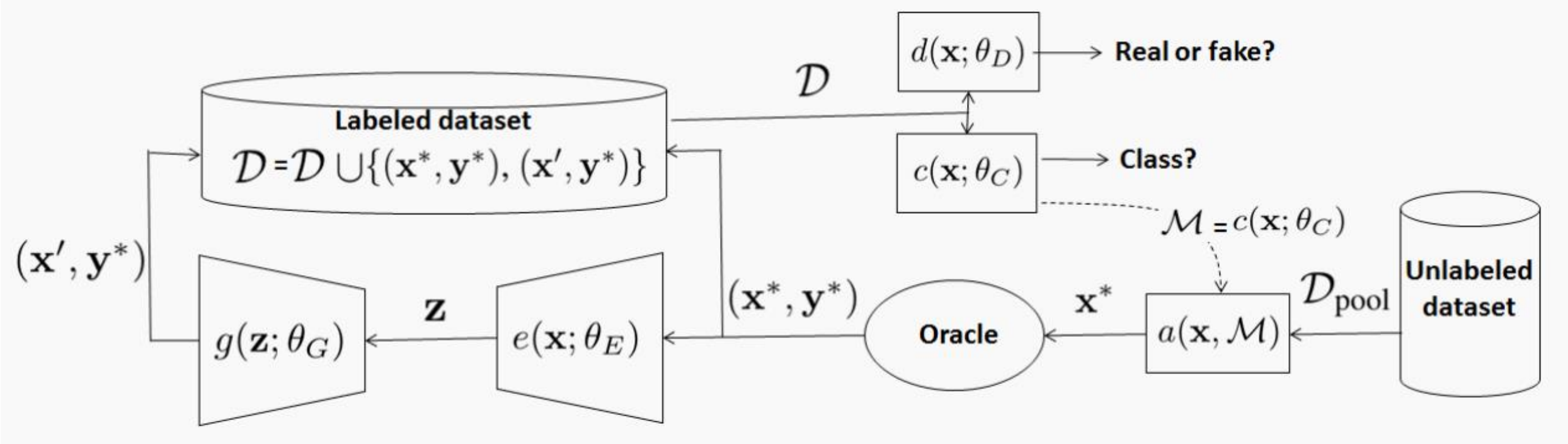
- Targets the augmentation of the labeled data set with informative generated samples



- Data augmentation can waste computational resources because it indiscriminately generates samples that are not guaranteed to be informative
- Active learning selects a small subset of informative samples (from a large un-annotated set)that may be insufficient for the training process.

Bayesian Generative Active Deep Learning

- Combine BALD and BDA for generating new labeled samples that are informative for the training process

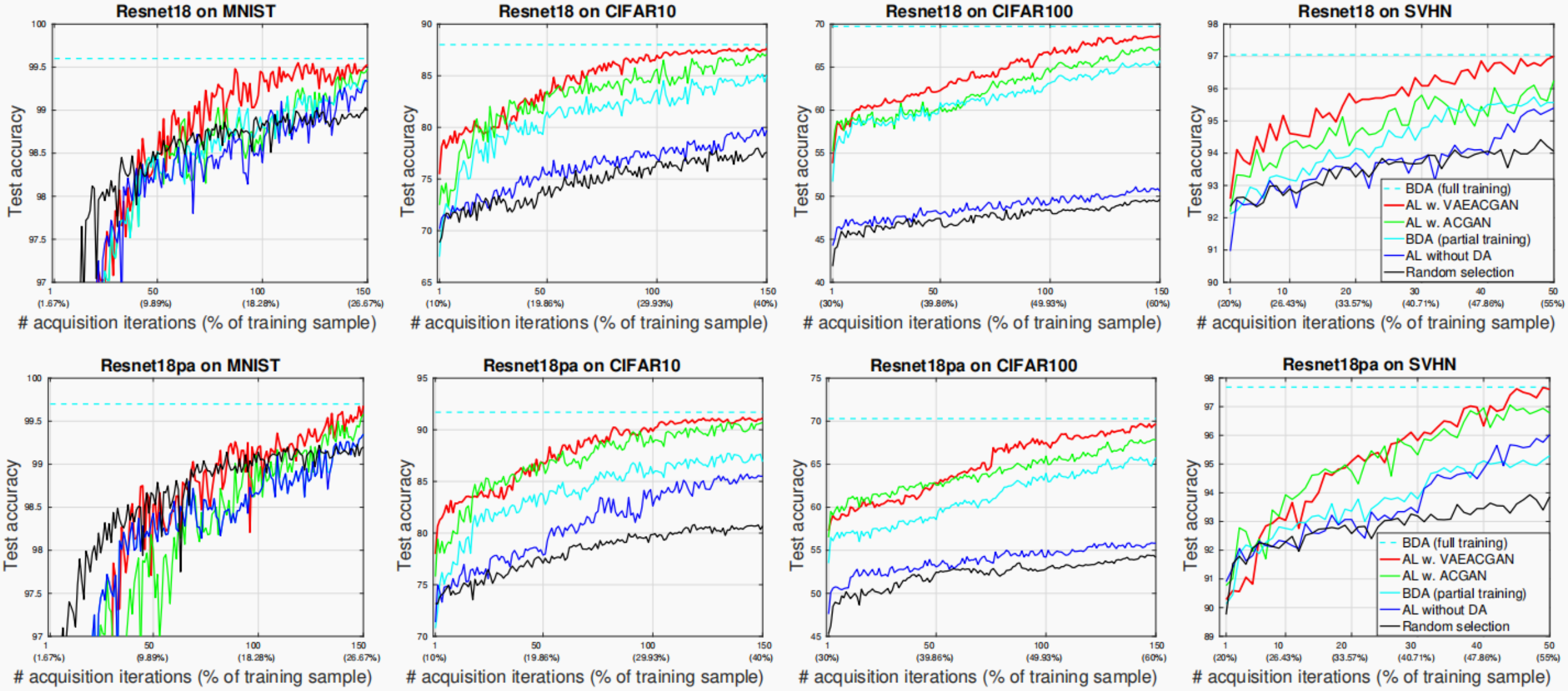


$$\mathbf{x}^* = \arg\max_{\mathbf{x}\in\mathcal{D}_{\text{pool}}} a(\mathbf{x},\mathcal{M})$$
$$= \arg\max_{\mathbf{x}\in\mathcal{D}_{\text{pool}}} H[\mathbf{y}|\mathbf{x},\mathcal{D}] - \mathbb{E}_{\theta\sim p(\theta|\mathcal{D})}[H[\mathbf{y}|\mathbf{x},\theta]],$$

$$\mathbf{x}' = g(e(\mathbf{x}^*)). \qquad \|\mathbf{x}'-\mathbf{x}^*\| < \varepsilon,$$

$$(\mathbf{x}^*,\mathbf{y}^*) \text{ and } (\mathbf{x}',\mathbf{y}^*)$$

Bayesian Generative Active Deep Learning



(a) MNIST (b) CIFAR-10 (c) CIFAR-100 (d) SVHN