Bayesian Generative Active Deep Learning

Toan Tran , Thanh-Toan Do , Ian Reid , Gustavo Carneiro

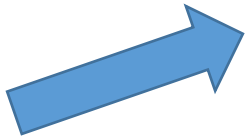
ICML (2019)

2019.12.11

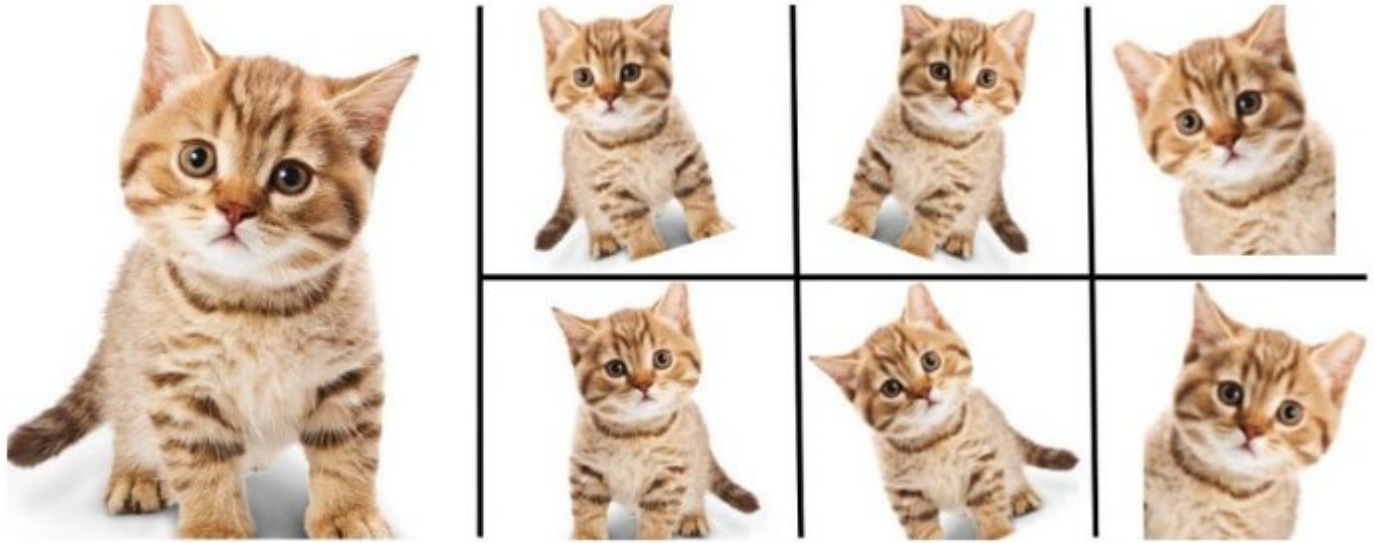
Content:

Introduction

Labeled Data



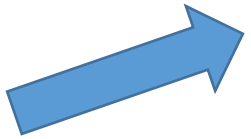
Data Augmentation



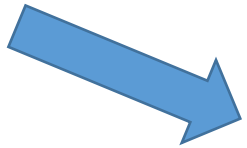
winter Yosemite → summer Yosemite

Introduction

Labeled Data

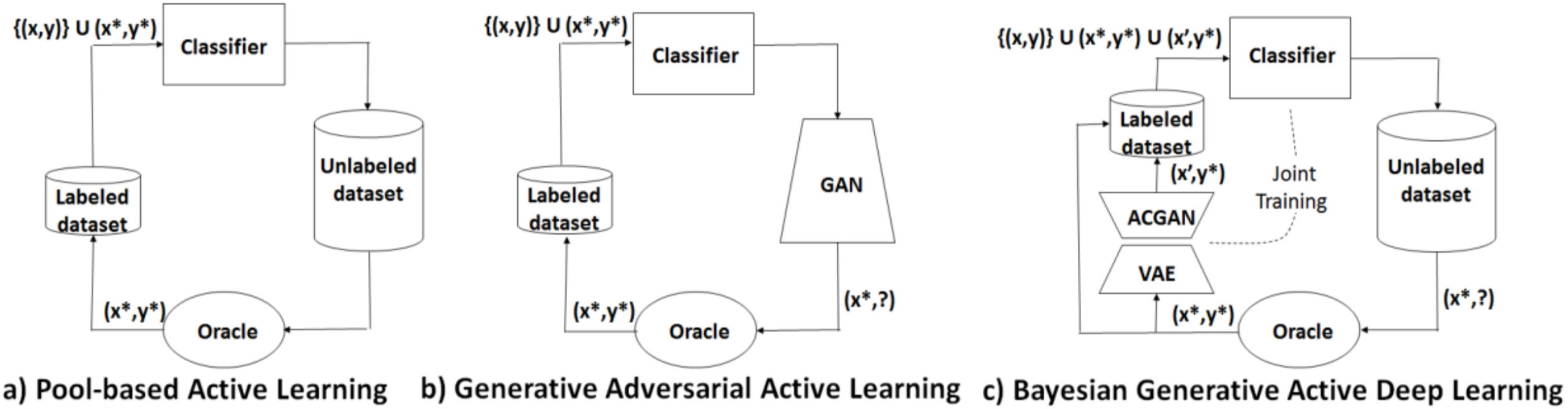


Data Augmentation



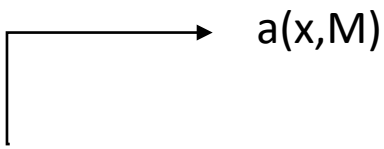
Active Learning

In this paper, we propose a Bayesian generative active deep learning approach that combines active learning with data augmentation.



2. Related Work

Bayesian Active Learning



Acquisition function, which is maximized in order to select the most informative samples.

Bayesian active learning by disagreement (BALD) scheme

Monte Carlo (MC) dropout method

Data Augmentation

1) Method

1.Data augmentation can be performed with “label-preserving” transformations → “poor’ s man” data augmentation (PMDA)

2.Bayesian data augmentation (BDA) trains a deep generative model (using the training set), which is then used to produce new artificial training samples. √ better

2) Drawback

informative ?

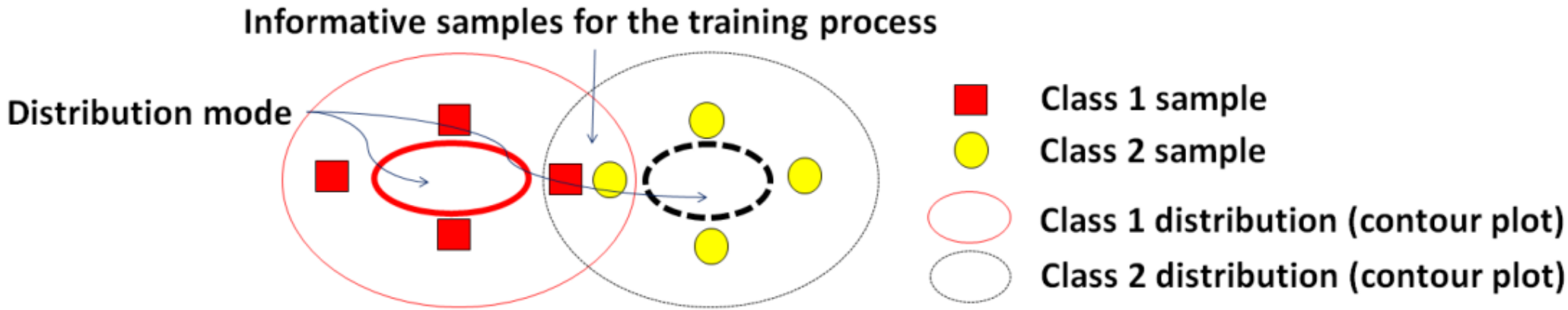


Figure 2. We target the generation of samples that belong to the generative distribution learned from the training set, and that are also informative for the training process. In particular, we aim to generate synthetic samples belonging to the intersection of different class distributions known as "disagreement region" (Settles, 2012). These generated instances are informative for the training process since the learning model is uncertain about them (Houlsby et al., 2011).

Generative Active Learning

Zhu & Bento (2017) introduced a generative adversarial active learning (GAAL) model to produce new synthetic samples that are informative for the current model.

Advantage

it can generate rich representative training data

Limitations

1.the acquisition function must be simple to compute and optimize

2.Model is not fine-tuned； not “co-evolve”.

Variational Autoencoder Generative Adversarial Networks

ACGAN to improve the quality of the GAN generated images

VAE tackle the low diversity problem (known as “mode collapse”)

Model

p(θ|D)：
an estimate of the posterior of the parameters θ of the model M given D.

the prediction p(y|x,D)
the distribution p(y|x,θ)

1.Bayesian Active Learning by Disagreement (BALD)

$$\begin{aligned}\mathbf{x}^* &= \arg\max_{\mathbf{x}\in\mathcal{D}_{\text{pool}}} a(\mathbf{x},\mathcal{M}) \\ &= \arg\max_{\mathbf{x}\in\mathcal{D}_{\text{pool}}} H[\mathbf{y}|\mathbf{x},\mathcal{D}] - \mathbb{E}_{\theta\sim p(\theta|\mathcal{D})}[H[\mathbf{y}|\mathbf{x},\theta]], \quad (1)\end{aligned}$$

the labeled data set is updated for the next training iteration: D ← D ∪ (x ∗ ,y ∗) .

2.MonteCarlo(MC)dropoutmethod

$$a(\mathbf{x},\mathcal{M}) \approx -\sum_c \left(\frac{1}{T}\sum_t \hat{p}_c^t\right)\log\left(\frac{1}{T}\sum_t \hat{p}_c^t\right) + \frac{1}{T}\sum_{c,t}\hat{p}_c^t \log \hat{p}_c^t, \quad (2)$$

where T is the number of dropout iterations, $\hat{\mathbf{p}}^t = [\hat{p}_1^t,\ldots,\hat{p}_C^t] = \text{softmax}(f(\mathbf{x};\theta^t))$, with f representing the network function parameterized by θ^t that is sampled from an estimate of the (commonly intractable) posterior $p(\theta|\mathcal{D})$ at the t-th iteration.

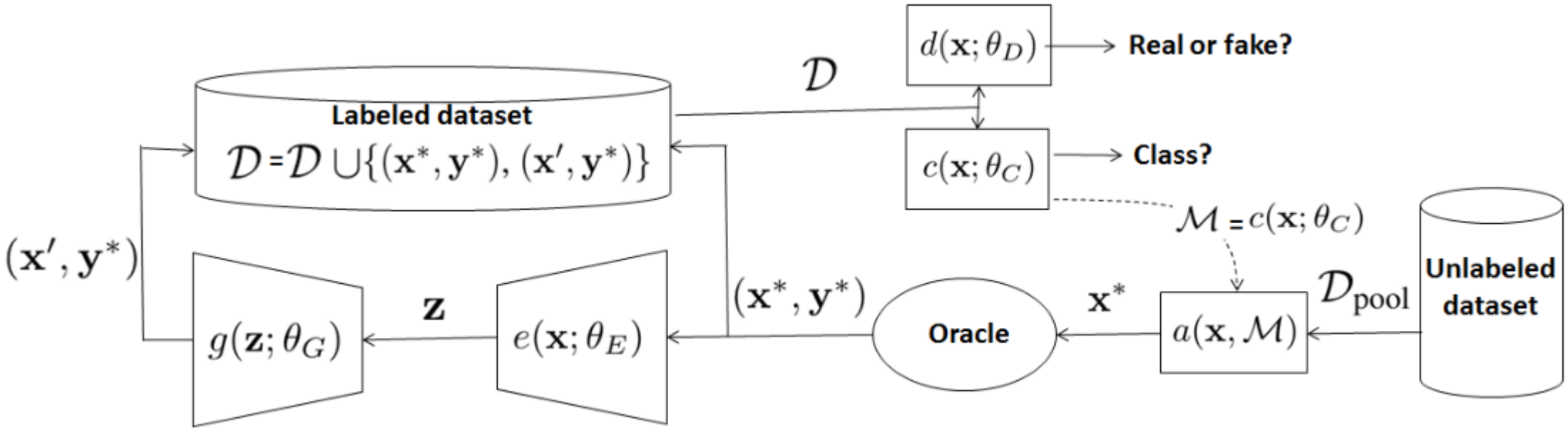


Figure 3. Network architecture of our proposed model.

The VAE-GAN loss function：

$$\mathcal{L} = \mathcal{L}_{\text{VAE}} + \mathcal{L}_{\text{ACGAN}},$$

$$\begin{aligned}\mathcal{L}_{\text{VAE}} &= \mathcal{L}_{\text{rec}} + \mathcal{L}_{\text{prior}} \\ &= \mathcal{L}_{\text{rec}}(\mathbf{x}, g(e(\mathbf{x};\theta_E);\theta_G) + D_{\text{KL}}(q(\mathbf{z}|\mathbf{x})\|p(\mathbf{z})), \qquad (8)\end{aligned}$$

$$\mathcal{L}_{\text{ACGAN}} = \log(d(\mathbf{x};\theta_D)) + \log(1 - d(g(\mathbf{z};\theta_G);\theta_D))$$
$$+ \log(1 - d(g(\mathbf{u};\theta_G);\theta_D)) + \log(\text{softmax}(c(\mathbf{x};\theta_C)))$$
$$+ \log(\text{softmax}(c(g(\mathbf{z};\theta_G);\theta_C)))$$
$$+ \log(\text{softmax}(c(g(\mathbf{u};\theta_G);\theta_C))), \qquad (9)$$

$u \sim N(0,I)$

Algorithm 1 Bayesian Generative Active Learning

Initialize network parameters $\theta_E, \theta_G, \theta_C, \theta_D$, and pre-train the classifier $c(\mathbf{x}; \theta_C)$ with $\mathcal{D}$

repeat

Pick the most informative $\mathbf{x}^*$ from $\mathcal{D}_{\text{pool}}$ with $\mathbf{x}^* = \arg\max_{\mathbf{x} \in \mathcal{D}_{\text{pool}}} a(\mathbf{x}, \mathcal{M})$ in (1) and (2), where $\mathcal{M}$ is represented by the classifier $c(\mathbf{x}; \theta_C)$;

Request the oracle to label the selected sample, which forms $(\mathbf{x}^*, \mathbf{y}^*)$

$\mathbf{z} \leftarrow e(\mathbf{x}^*; \theta_E)$

$\mathcal{L}_{\text{prior}} \leftarrow D_{\text{KL}}(q(\mathbf{z}|\mathbf{x}^*) \| p(\mathbf{z}))$

$\mathbf{x}' = g(e(\mathbf{x}^*); \theta_G)$

$\mathcal{L}_{\text{rec}} \leftarrow \mathcal{L}_{\text{rec}}(\mathbf{x}^*, \mathbf{x}')$

Sample $\mathbf{u} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$

$\mathcal{L}_{\text{ACGAN}} \leftarrow \log(d(\mathbf{x}^*)) + \log(1 - d(\mathbf{x}')) + \log(1 - d(g(\mathbf{u}))) + \log(\text{softmax}(c(\mathbf{x}^*))) + \log(\text{softmax}(c(\mathbf{x}'))) + \log(\text{softmax}(c(g(\mathbf{u}))))$

$\theta_E \leftarrow \theta_E - \nabla_{\theta_E}(\mathcal{L}_{\text{rec}} + \mathcal{L}_{\text{prior}})$

$\theta_G \leftarrow \theta_G - \nabla_{\theta_G}(\gamma \mathcal{L}_{\text{rec}} - \mathcal{L}_{\text{ACGAN}})$ (parameter $\gamma = 0.75$ (Larsen et al., 2016) in our experiments)

$\theta_D \leftarrow \theta_D - \nabla_{\theta_D} \mathcal{L}_{\text{ACGAN}}$

$\theta_C \leftarrow \theta_C - \nabla_{\theta_C} \mathcal{L}_{\text{ACGAN}}$

until convergence

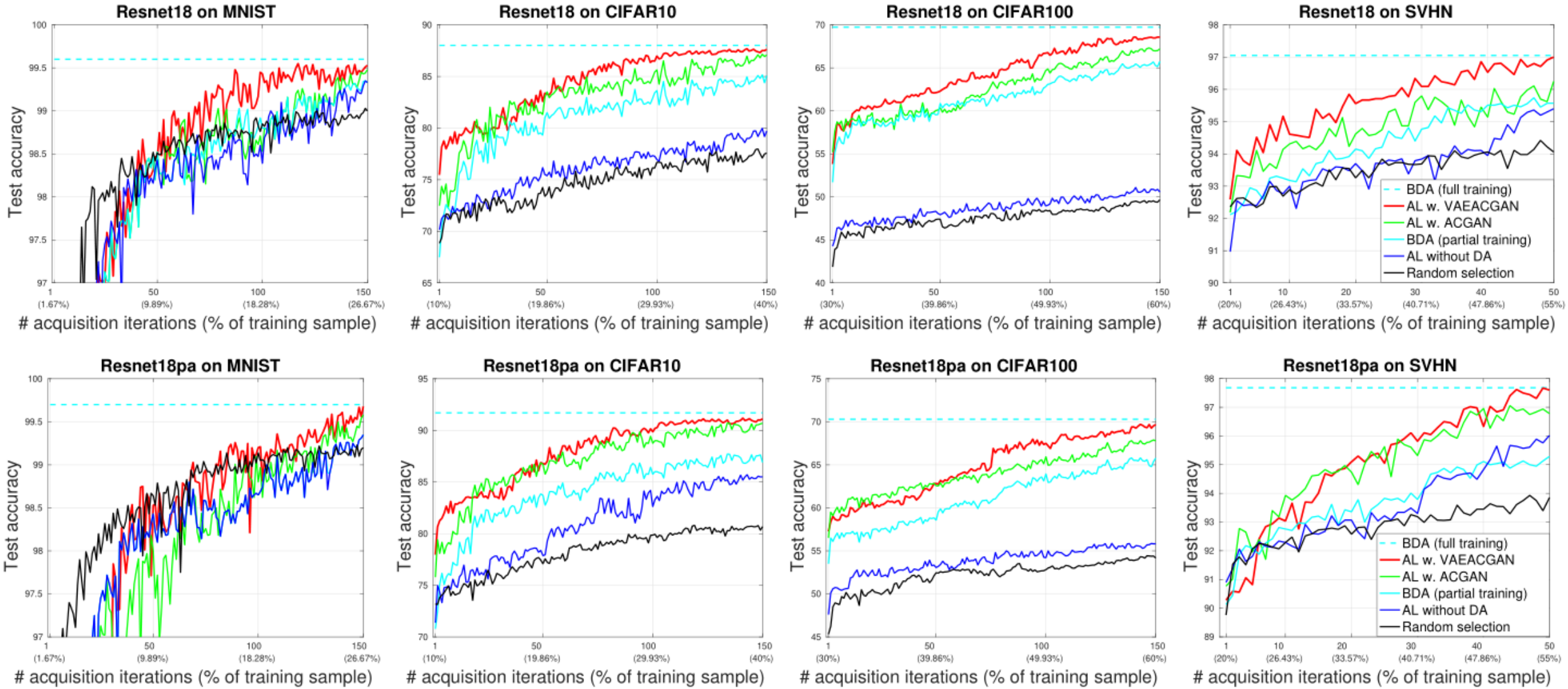
Experiments

DataSet: MNIST,CIFAR-10,CIFAR-100, and SVHN.

Methods: 贝叶斯生成式主动深层学习模型 (AL w. VAEACGAN)
使用 BDA 的主动学习模型 (AL w. ACGAN)
未使用数据增加处理的 BALD (AL without DA)
未使用主动学习方法的 BDA (BDA)
随机生成样本 (random selection)

An upper bound ： full training set (BDA (full training)) and 10× data augmentation

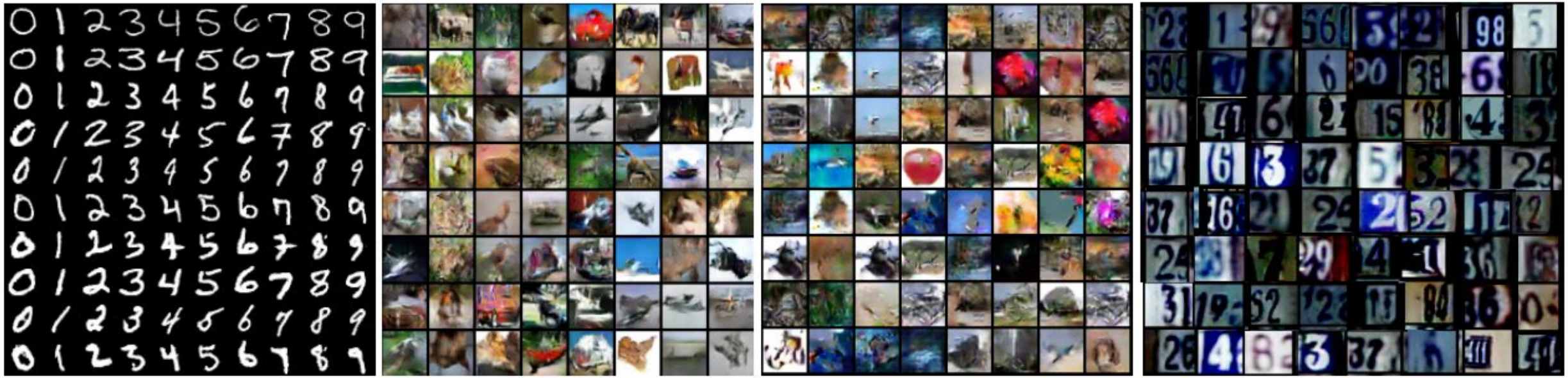
Classifier： ResNet18、ResNet18pa



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

Table 1. Mean ± standard deviation of the classification accuracy on MNIST, CIFAR-10, and CIFAR-100 after 150 iterations over 3 runs

	AL W. VAEACGAN	AL W. ACGAN	AL W. PMDA	AL WITHOUT DA	BDA (PARTIAL TRAINING)	RANDOM SELECTION
			MNIST			
RESNET18	$\mathbf{99.53 \pm 0.05}$	99.45 ± 0.02	99.37 ± 0.15	99.33 ± 0.10	99.33 ± 0.04	99.00 ± 0.13
RESNET18PA	$\mathbf{99.68 \pm 0.08}$	99.57 ± 0.07	99.49 ± 0.09	99.35 ± 0.11	99.35 ± 0.07	99.20 ± 0.12
			CIFAR-10			
RESNET18	$\mathbf{87.63 \pm 0.11}$	86.80 ± 0.45	82.17 ± 0.35	79.72 ± 0.19	85.08 ± 0.31	77.29 ± 0.23
RESNET18PA	$\mathbf{91.13 \pm 0.10}$	90.70 ± 0.24	87.70 ± 0.39	85.51 ± 0.21	86.90 ± 0.27	80.69 ± 0.19
			CIFAR-100			
RESNET18	$\mathbf{68.05 \pm 0.17}$	66.50 ± 0.63	55.24 ± 0.57	50.57 ± 0.20	65.76 ± 0.40	49.67 ± 0.52
RESNET18PA	$\mathbf{69.69 \pm 0.13}$	67.79 ± 0.76	59.67 ± 0.60	55.82 ± 0.31	65.79 ± 0.51	54.77 ± 0.29



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

Figure 7. Images generated by our proposed *AL w. VAEACGAN* approach for each data set.