Variational Adversarial Active Learning

1. 摘要翻译

主动学习旨在通过 Oracle 标记查询获取的最具代表性的样本从而开发出一种标记高效的算法。文章提出了一种 pool-based 的半监督主动学习算法，该算法以对抗性方式隐式学习了主动学习的采样策略。该方法使用变分自动编码器 (VAE) 和经过训练以区分未标记数据和标记数据的对抗网络来学习潜在空间。在 VAE 和对抗网络之间通过进行 min-max 游戏来训练：当 VAE 试图欺骗对抗网络以预测所有数据点都来自标记池时，对抗网络将学习如何区分差异在潜在的空间。论文在各种图像分类和语义分割基准数据集上评估了所提出的方法，并在 CIFAR10 / 100，Caltech-256，ImageNet，Cityscapes 和 BDD100K 上获得了较高的表现。文章的结果表明所提出的对抗方法在大规模环境中学习了有效的低维潜在空间，另外文章还提供了一种计算高效的采样方法。

2. 论文的贡献

近来基于学习的计算机视觉方法的成功在很大程度上取决于大量的带注释的训练样例，但是这些样例的标注成本往往过高或无法大规模获得。为了应对该缺陷，主动学习算法旨在增量式的选择用于标注的样本，从而以较低的标记成本实现较高的分类性能。

当前的主动学习方法主要分为 query-acquiring（pool-based）方法或者 query-synthesizing 方法等。query-synthesizing 方法使用生成模型来生成

信息样本，而 pool-based 的算法使用不同的采样策略来确定如何选择信息最多的样本。基于池的方法可以分为三大类：基于不确定性的方法，基于代表性的方法及两种方法的组合。

然而，

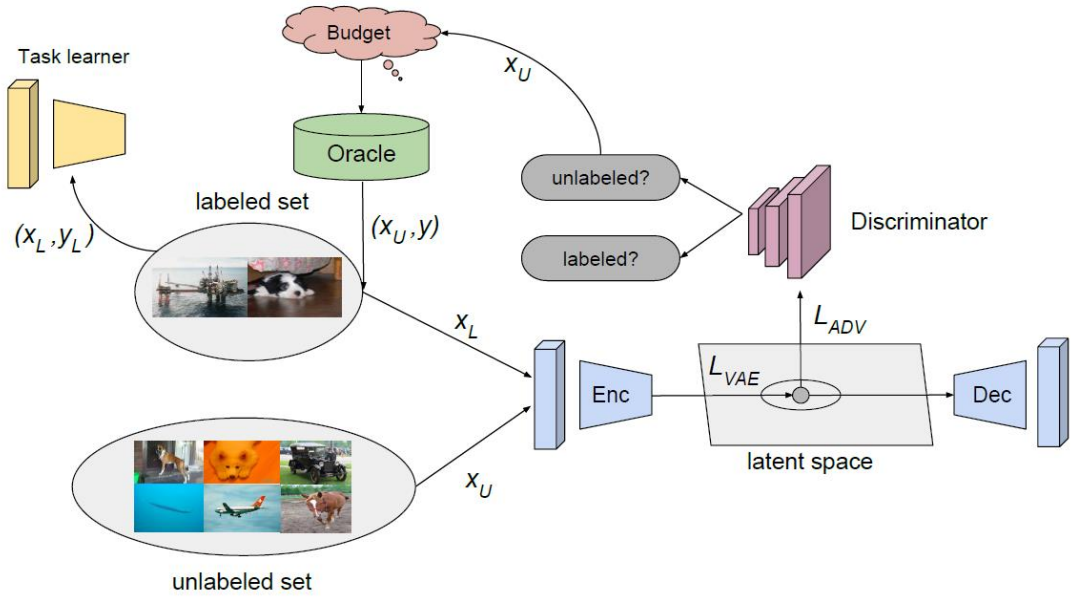
1. 对于基于不确定性的方法，许多经典的技术方法并不能很好的应用到深度学习中去。
2. 对于基于代表性的方法，虽然事实证明 Core-set 算法是一种可以用于大规模图像分类任务的有效表示学习方法，并且在理论上证明，当类别数较少时，这种方法最有效。但是随着样本类别数目增多，其表现会下降。此外，对于高维数据，使用像 Core-set 这样的基于距离的表示方法似乎是无效的，因为在高维中，p 范数会遭受维数的诅咒， 这称为 distance concentration 现象。
3. 对于两种思想混合的方法，为了达到固定的性能目标，与其他方法相比，他们通常需要每批次采样更多实例。

为了应对这些问题，本文提出一种 pool-based 的主动学习策略，通过使用变分自动编码器 (VAEs) 从标记和未标记的数据中学习低维潜在空间。该方法被称为 “变分对抗主动学习(VAAL)”，它从未标记的池子中选择要标记的实例，这些实例在 VAE 所学习的潜在空间中有足够的差异，以最大程度地提高在新标记的数据上学习的表示的性能。该方法中样本选择由对抗网络执行，该对抗网络对实例所属的池进行分类，判断其是属于标记样本还是未标记样本。其中，VAE 和鉴别器被构造为类似于 GAN 的 mini-max 游戏来进行训练。在文章的实验中，作者展示了该方法在各种大规模图像分类和分割数据集上的优越性能，并且

在性能和计算成本方面均优于当前的最新方法。

3. 提出的方法

文章提出的方法的整体框架如图所示，主要分为三个部分，VAE 部分，判别器部分和查询标记部分。



其训练流程大致如下：首先将标记样本和未标记样本放入 VAE 中进行训练来学习隐空间；之后将隐空间中的标记和未标记样本输入到判别器中进行分类训练；根据查询策略选择标记效益较高的样本交给 oracle 进行标记，并将标记后的样本从未标记集合中去除，添加到标记集合中，重新开始下一轮的训练。

接下来对具体细节进行说明：

- 基本说明：

  令 (x_L, y_L) 是属于标记数据池 (X_L, Y_L) 的样本对，X_U 表示尚未标记的更大样本池 (x_U)。主动学习的目标是通过迭代查询固定采样样本来训练标签效率最高的模型，即从未标记池 $(x_U \sim X_U)$ 中获取 b 个信息量最大的样本数，交由 oracle 进行标记后交给模型继续进行训练，从而使预

期损失最小化。

- VAE 部分：

使用 β-变分自动编码器进行表示学习，其中编码器使用高斯先验为基础学习低维空间，然后解码器重建输入数据。其损失函数如下：

$$\begin{aligned}\mathcal{L}_{\mathrm{VAE}}^{trd} = \quad & \mathbb{E}[\log p_\theta(x_L|z_L)] - \beta\, \mathrm{D}_{\mathrm{KL}}(q_\phi(z_L|x_L)||p(z)) \\ & +\mathbb{E}[\log p_\theta(x_U|z_U)] - \beta\, \mathrm{D}_{\mathrm{KL}}(q_\phi(z_U|x_U)||p(z)) \quad (1)\end{aligned}$$

p_θ ：encoder 的参数

$q_\emptyset$ ：decoder 的参数

$p(z)$ ： 服从高斯分布

β： 拉格朗日参数

- 判别器部分：

在 VAAL 中，训练对抗网络将（$z_L \cup z_U$）的潜在表示映射到二进制标签，如果样本属于X_L，则为 1，否则为 0。在该方法中 VAE 和对抗网络一同训练。当 VAE 将标记和未标记的数据映射到具有相似概率分布 $q_\emptyset(z_L|x_L)$ 和$q_\emptyset(z_U|x_U)$ 的相同潜在空间中时，它愚弄了鉴别器将所有输入分类为标记。另一方面，鉴别器试图有效地估计数据来自未标记数据的概率。VAE 和判别器的目标函数，如下所示：

$$\mathcal{L}_{\mathrm{VAE}}^{adv} = -\mathbb{E}[\log(D(q_\phi(z_L|x_L)))] - \mathbb{E}[\log(D(q_\phi(z_U|x_U)))] \quad (2)$$

$$\mathcal{L}_D = -\mathbb{E}[\log(D(q_\phi(z_L|x_L)))] - \mathbb{E}[\log(1 - D(q_\phi(z_U|x_U)))] \quad (3)$$

通过结合式（1）和等式（2）我们在 VAAL 中获得 VAE 的完整目标函数，如下所示：

$$\mathcal{L}_{\text{VAE}} = \lambda_1 \mathcal{L}_{\text{VAE}}^{trd} + \lambda_2 \mathcal{L}_{\text{VAE}}^{adv} \tag{4}$$

前面两个部分的伪代码如下：

Algorithm 1 Variational Adversarial Active Learning

Input: Labeled pool (X_L, Y_L), Unlabeled pool (X_U), Initialized models for θ_T, θ_{VAE}, and θ_D

Input: Hyperparameters: epochs, λ_1, λ_2, α_1, α_2, α_3

for $e = 1$ to epochs **do**
  sample $(x_L, y_L) \sim (X_L, Y_L)$
  sample $x_U \sim X_U$
  Compute $\mathcal{L}_{\text{VAE}}^{trd}$ by using Eq. 1
  Compute $\mathcal{L}_{\text{VAE}}^{adv}$ by using Eq. 2
  $\mathcal{L}_{\text{VAE}} \leftarrow \lambda_1 \mathcal{L}_{\text{VAE}}^{trd} + \lambda_2 \mathcal{L}_{\text{VAE}}^{adv}$
  Update VAE by descending stochastic gradients:
  $\theta'_{VAE} \leftarrow \theta_{VAE} - \alpha_1 \nabla \mathcal{L}_{\text{VAE}}$
  Compute $\mathcal{L}_{\text{D}}$ by using Eq. 3
  Update D by descending its stochastic gradient:
  $\theta'_D \leftarrow \theta_D - \alpha_2 \nabla \mathcal{L}_{\text{D}}$
  Train and update T:
  $\theta'_T \leftarrow \theta_T - \alpha_3 \nabla \mathcal{L}_{\text{T}}$
end for
return Trained $\theta_T, \theta_{VAE}, \theta_D$

- 查询策略部分：

查询策略是通过判别器选择出对预测为标记数据最不确定的一批样本。

其伪代码如下：

Algorithm 2 Sampling Strategy in VAAL

Input: b, X_L, X_U

Output: X_L, X_U

Select samples (X_s) with $\min_b\{\theta_D(z_U)\}$
$Y_o \leftarrow \mathcal{ORACLE}(X_s)$
$(X_L, Y_L) \leftarrow (X_L, Y_L) \cup (X_s, Y_o)$
$X_U \leftarrow X_U - X_s$
return X_L, X_U

4. 实验设置

实验中假设 oracle 都是理想的，没有噪声的。

- **数据集：**

  实验主要分为两个部分，在图像分类任务上的测试和在图像分割任务上的测试。图像分类数据集包括：CIFAR-10， CIFAR-100，Caltech-256，ImageNet。具体的以下实验设定如下表所示：

Dataset	#Classes	Train + Val	Test	Initially Labeled	Budget	Image Size
CIFAR10 [29]	10	$45000 + 5000$	10000	5000	2500	32×32
CIFAR100 [29]	100	$45000 + 5000$	10000	5000	2500	32×32
Caltech-256 [22]	256	$27607 + 3000$	2560	3060	1530	224×224
ImageNet [6]	1000	$1153047 + 128120$	50000	128120	64060	224×224
BDD100K [57]	19	$7000 + 1000$	2000	800	400	688×688
Cityscapes [5]	19	$2675 + 300$	500	300	150	688×688

- **评估：**

  通过测量标记集和达到总训练集的 10%，15%，20%，25%，30%，35%，40%时的训练准确性和平均 IoU 来评估 VAAL 在图像分类和分割中的性能表现。此外，除了 ImageNet 实验重复了两次外，其它实验都重复了 5 次，避免偶然性。

- **图像分类任务：**
  - 对比方法：Core-set，Monte-Carlo Dropout，Ensembles w. VarR，max-entropy deep Bayesian AL (DBAL)，random sampling。
  - 分类模型：VGG-16
  - VAE 部分：β-VAE
  - 判别器部分：5 层 MLP
- **图像分割任务：**

- 对比方法：Core-set，MC-Dropout，Query-By-Committee (QBC)，suggestive annotation (SA)，random sampling。
- 分类模型：DRN
- VAE 部分：β-VAE
- 判别器部分：5 层 MLP

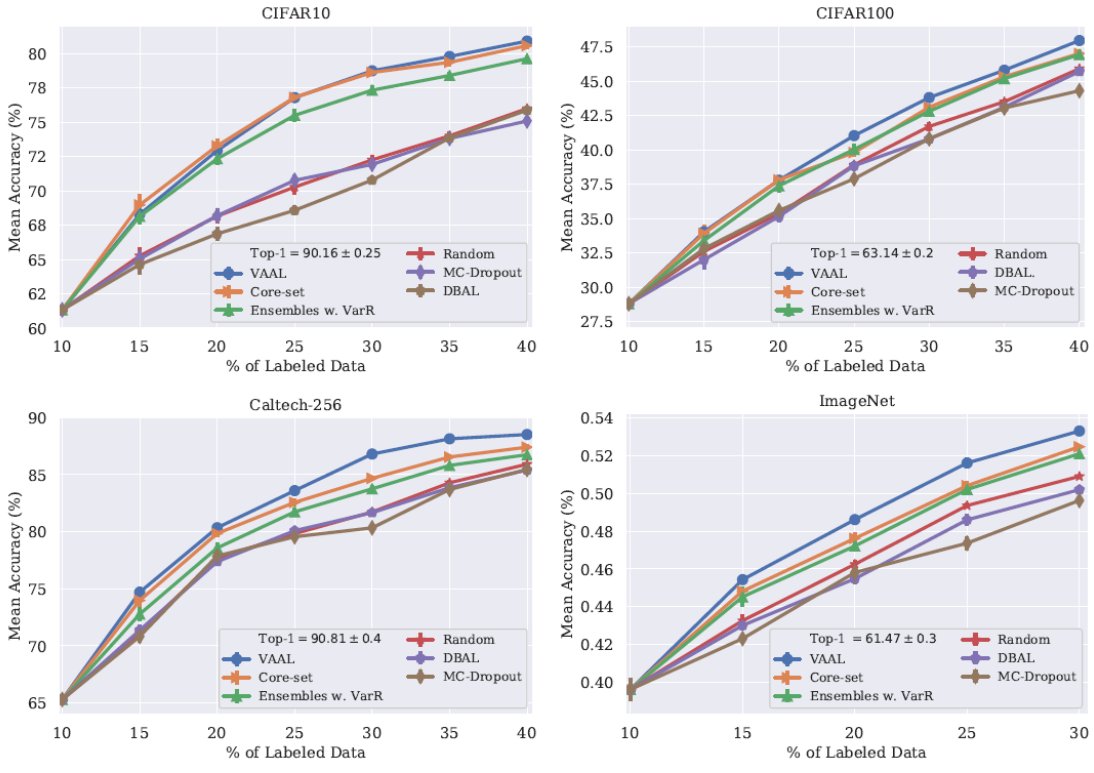
- **模型参数：**

Experiment	d	α_1	α_2	α_3	λ_1	λ_2	β	batch size	epochs
CIFAR10	32	5×10^{-4}	5×10^{-4}	5×10^{-4}	1	1	1	64	100
CIFAR100	32	5×10^{-4}	5×10^{-4}	5×10^{-4}	1	1	1	64	100
Caltech-256	64	5×10^{-4}	5×10^{-4}	5×10^{-4}	1	10	1	64	100
ImageNet	64	10^{-1}	10^{-3}	10^{-3}	1	10	1	64	100
BDD100K	128	10^{-3}	10^{-3}	10^{-3}	1	25	1	8	100
Cityscapes	128	10^{-3}	10^{-3}	10^{-3}	1	25	1	8	100

5. 实验结果

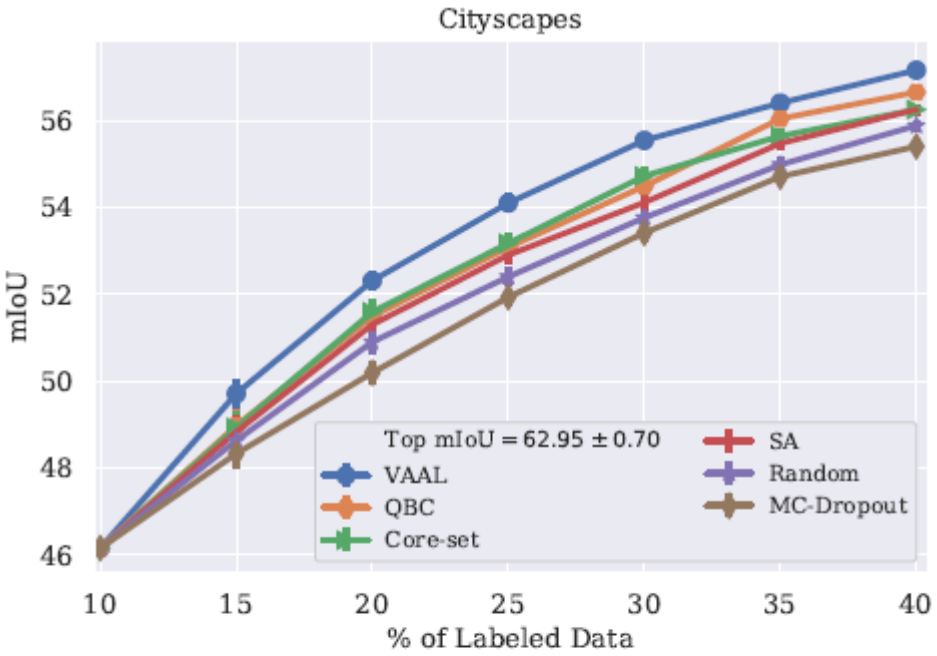
- **图像分类任务：**

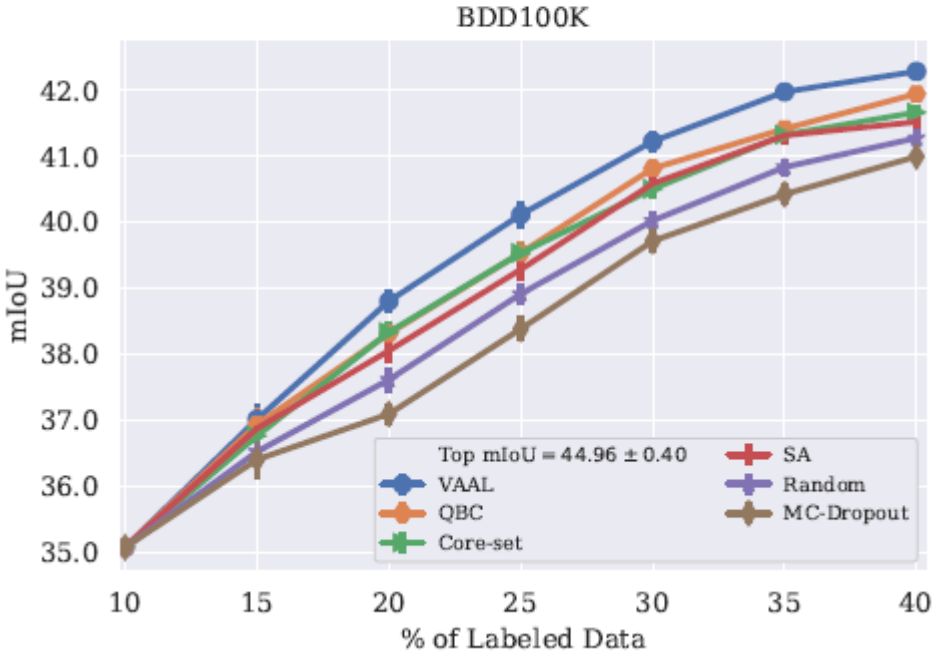
实验结果如下如所示，可以看出在各实验中 VAAL 方法都取得最优成绩。



- **图像分割任务：**

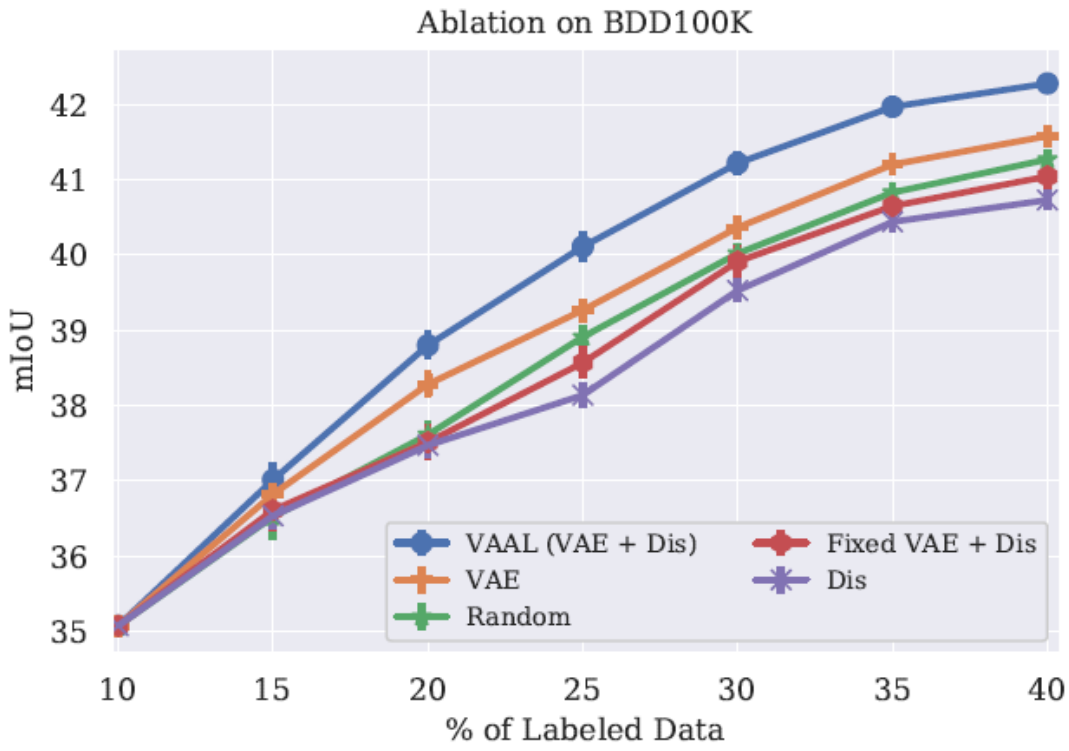
  实验结果如下如所示，可以看出在各实验中 VAAL 方法都取得最优成绩。





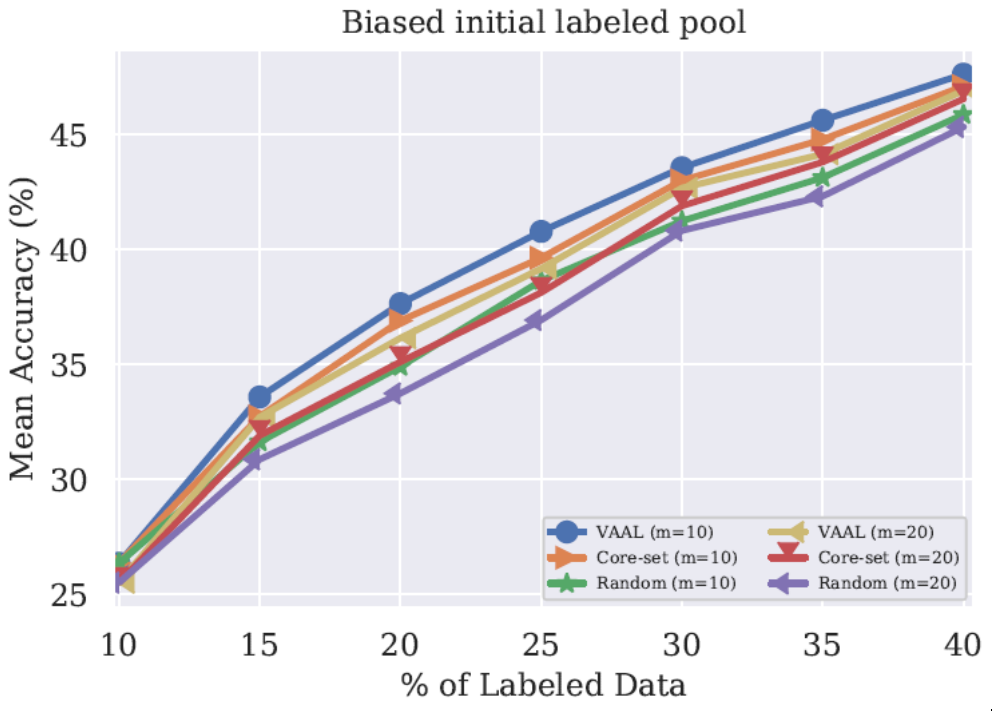
- **消融研究：**

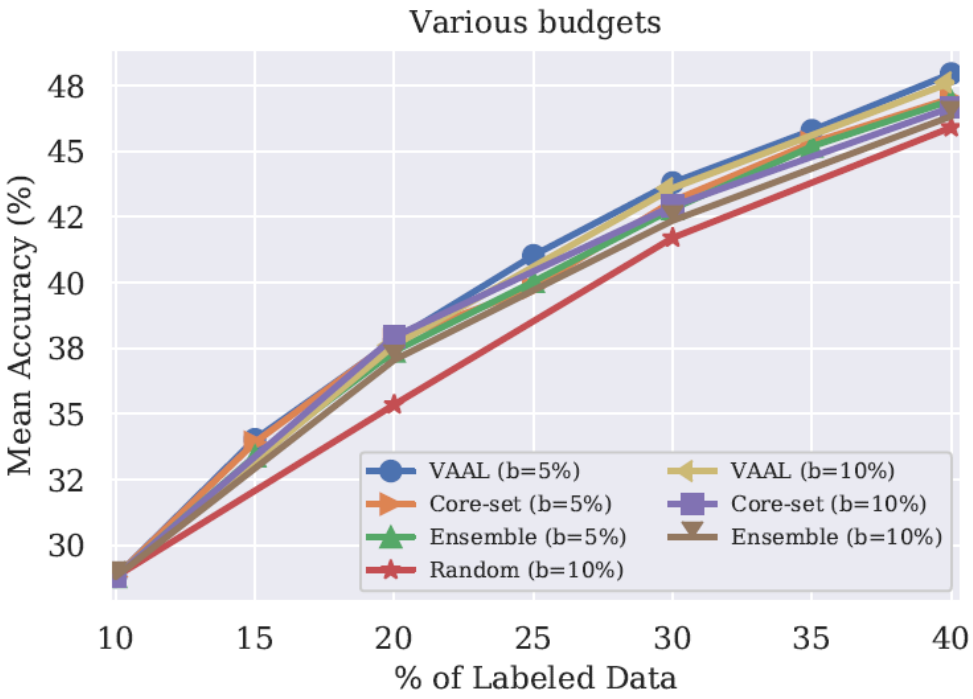
  为了检查 VAAL 中 VAE 和鉴别器的各自发挥的作用大小，为此进行消融研究，结果如下：

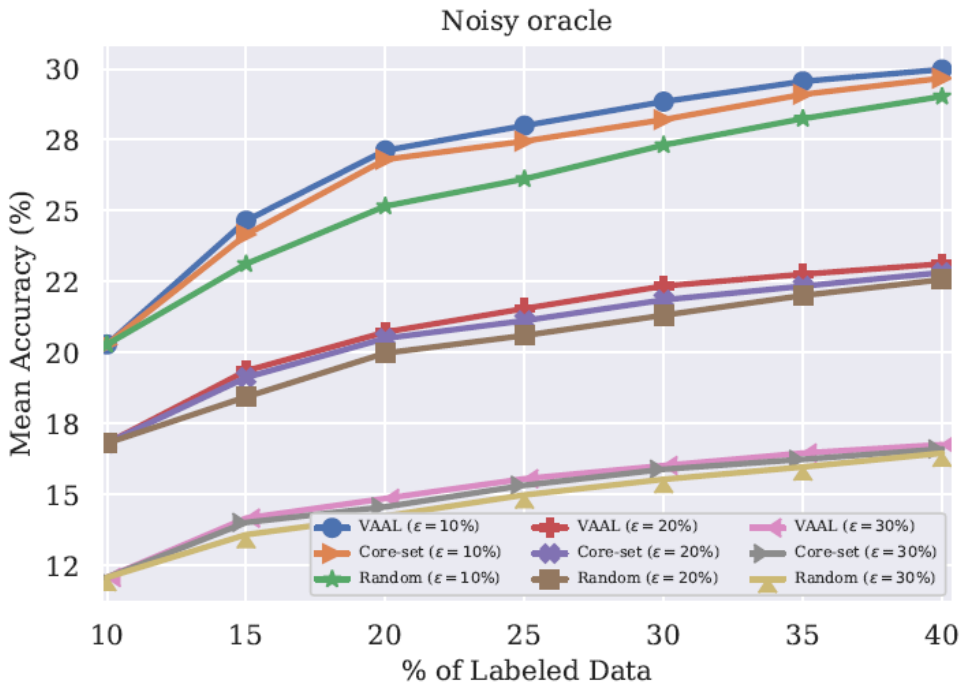


- **模型鲁棒性研究：**

  为了探究模型的鲁棒性，通过改变初始集合，改变 budget size 和引入有噪音的 oracle 来进行实验对比，结果如下。可以看出 VAAL 模型在各自情况下都仍然表现良好，取得优异的效果，具有良好的鲁棒性。







- **时间对比：**

  在应用主动学习到实际中的时候，样本的查询速度应该越快越好，将 VAAL 方法与其它方法在查询速度上面进行对比，对比结果如下，可以发现 VAAL 方法的查询速度也十分优异。

Method	Time (sec)
MC-Dropout [15]	81.05
Core-set [43]	75.33
Ensembles w. VarR [1]	20.48
DBAL. [16]	10.95
VAAL (ours)	**10.59**