Adversarial Representation Active Learning

Ali Mottaghi
Stanford University
mottaghi@stanford.edu

Serena Yeung
Stanford University
syyeung@stanford.edu

Abstract

Active learning is the problem of choosing the most informative samples to reduce the burden of labeling in the machine learning model. By only labeling those informative samples, the model can achieve higher performance with fewer labels. In this work, we introduce a new active learning approach that exploits unlabeled data to even further reduce the number of labels required to achieve the desired performance. We focus on the problem of image classification and show that our model outperforms other approaches in the literature as it uses the available unlabeled images for both selecting the most informative samples as well as training the classifier. In our experiments, we show that our model achieves the state-of-the-art on a wide range of complexities (from MNIST to ImageNet) by efficiently using the already available unlabeled data.

1. Introduction

While deep learning has achieved great success in computer vision tasks ranging from image classification to detection and segmentation, its success is predicated on large amounts of labeled training examples. This is a significant challenge as deep learning practitioners increasingly apply deep learning models to solve new problems in diverse domains from medicine [11] to sustainability [5], placing undue burden on domain experts to label large amounts of training data. Active learning, where training examples are incrementally selected for labeling to yield high classification accuracy at low labeling budgets, has therefore emerged as an exciting paradigm with significant potential for democratizing the use of deep learning [4].

Active learning has been widely studied and most of the early works can be found in the classical survey of [9]. Current approaches can be categorized as query-acquiring (pool-based) and query-synthesizing algorithms. Pool-based methods use various acquisition functions for selecting the most informative examples [9] [8] [4] while query-synthesizing algorithms generate informative examples from scratch [12]. One of the highest developments in pool-based active learning is variational adversarial active learning (VAAL) [10], where the authors train a variational auto-encoder (VAE) and a discriminator to learn a latent representation on both labeled and unlabeled data. They then use the output of discriminator as a measure of uncertainty for selecting from unlabeled data. However, they only use labeled images to train the classifier.

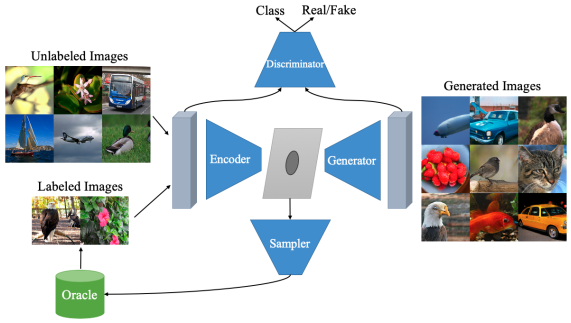

Figure 1: At each iteration, our model learns a latent representation of both labeled and unlabeled images, and simultaneously trains a classifier using labeled and class-conditional generated images. Then, it selects the most informative unlabeled images to be labeled by the oracle for the next iteration.

In this work, we introduce an active learning model based on the key observation that pool-based active learning approaches can more effectively use the unlabeled pool in the incremental training of the classifier itself. We propose a model that utilizes VAAL [10] but within a representation learning framework. Importantly, we perform the representation learning with a bidirectional GAN model which was first introduced by [2], and later refined by [3] to achieve the state-of-the-art representation learning on ImageNet. BiGAN allows modelling the underlying structure of the unlabeled data to infer additional class labels for generated images that can be used to train the classifier. We share the encoder and decoder of the VAAL as the encoder and generator of the conditional BiGAN, and co-train the acquisition function and conditional BiGAN jointly. This allows, for example, the classifier in the discriminator to improve the shared generator / decoder and hence the acquisition function and vice versa the classifier will be improved.

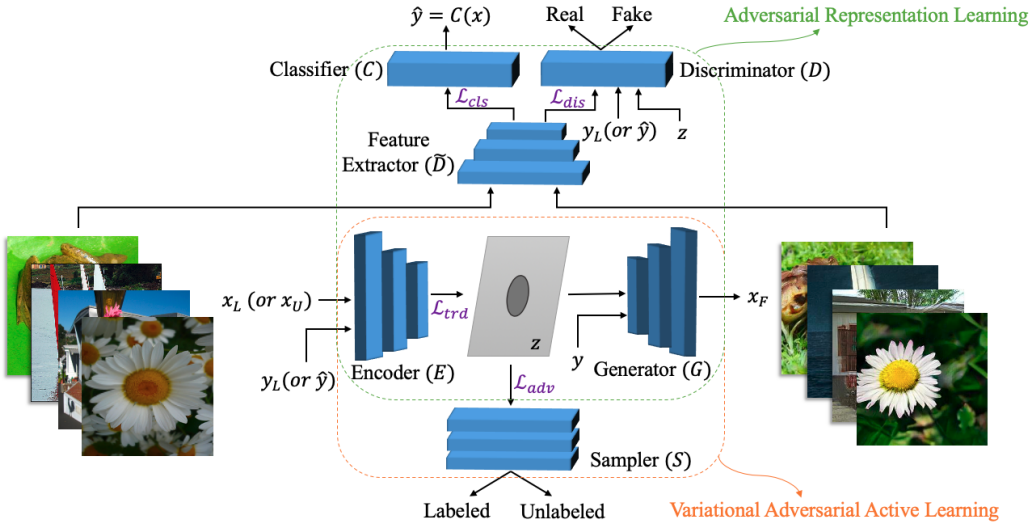

Figure 2: Our model consists of 1) Variational adversarial active learning [10] which attempts to learn a representation for labeled and unlabeled images in the latent space and select the most informative unlabeled images based on their latent code. 2) Adversarial representation learning [3] which couples the encoder-generator with a powerful discriminator and enhances the learned representation. By co-training all parts together we can actively learn a classifier with fewer labeled images.

2. Method

Our method tackles the standard active learning problem for image classification, where we have image space $\mathcal{X} \subseteq \mathbf{R}^{d_x}$, label space $\mathcal{Y} = \{1, 2, \ldots, K\}$, where K is the number of classes. Let (x_L, y_L) denote a sample (image, label) pair belonging to labeled data $\mathcal{D}_L = (X_L, Y_L)$, and x_U denote a sample image belonging to the pool of unlabeled data $\mathcal{D}_U = (X_U)$. Our goal is to train the most label-efficient classifier, i.e., with the highest classification accuracy for any given labeled dataset size $|\mathcal{D}_L|$. The active learner is allowed to iteratively select a fixed sampling budget b number of samples from the unlabeled pool ($x_U \sim X_U$), to be annotated by an oracle and added to the labeled dataset $\mathcal{D}_L$ for the next iteration, using a sample acquisition function $S(x_U)$.

2.1. Selecting informative samples with adversarial acquisition function

We use a similar approach as VAAL [10] acquisition function for selecting the most informative samples. This approach selects data examples for labeling that are not well-represented in the labeled training set by using a variational autoencoder with both reconstruction and adversarial losses. In contrast to VAAL which uses unconditional generative models, we use conditional ones since the class-conditional generated images from decoder can be further used to improve training of the classifier. Encoder and generator losses for the acquisition function become:

$$\begin{aligned}\mathcal{L}_E^{acq} + \mathcal{L}_G^{acq} = &\lambda_{trd}\mathbb{E}[p_{\theta_G}(x_L|z_L, y_L)] \\ &+ \lambda_{trd}\mathbb{E}[p_{\theta_G}(x_U|z_U, C(x_U))] \\ &- \lambda_{trd}\beta D_{KL}(q_{\theta_E}(z_L|x_L, y_L)||p(z))) \\ &- \lambda_{trd}\beta D_{KL}(q_{\theta_E}(z_U|x_U, C(x_U))||p(z))) \\ &- \lambda_{adv}\mathbb{E}[\log(S(q_{\theta_E}(z_L|x_L, y_L)))] \\ &- \lambda_{adv}\mathbb{E}[\log(S(q_{\theta_E}(z_U|x_U, C(x_U))))] \end{aligned} \quad (1)$$

where p_{θ_G} and q_{θ_E} are the encoder and generator parameterized by θ_G and θ_E, respectively. β is the Lagrangian parameter for the β-VAE reconstruction loss.

Also sampler (it is called discriminator in VAAL) loss can be written as:

$$\begin{aligned}\mathcal{L}_S^{acq} = &-\mathbb{E}[\log(S(q_{\theta_E}(z_L|x_L, y_L)))] \\ &-\mathbb{E}[\log(1 - S(q_{\theta_E}(z_U|x_U, C(x_U))))] \end{aligned} \quad (2)$$

2.2. Using unlabeled data in representation learning

Our framework for incorporating unlabeled data is based on the observation that the encoder-decoder in the acquisition function can also be used to additionally provide information about the unlabeled data and its underlying structure. In doing so, we add another discriminator to the encoder-decoder module to form the BiGAN. However, in contrast to BiGAN, we use a conditional discriminator which contains a classifier component that can naturally serve as the target classifier for active learning. Therefore, the discriminator is decomposed into a learned discriminator representation $\tilde{D}$ which is fed into a linear classifier $c_{r/f}$ for predicting real/fake, and linear classifier c_{cl} for predicting the class label. Using similar approach as [6], we also take into account the encoded latent variable z and the real class label or inferred class label in linear classifier $c_{r/f}$ for better predicting real/fake images. We denote $D(x, z, y) = c_{r/f}(\tilde{D}(x), z, y)$ for real/fake predictions and $C(x) = c_{cl}((\tilde{D}(x)))$ for class label predictions.

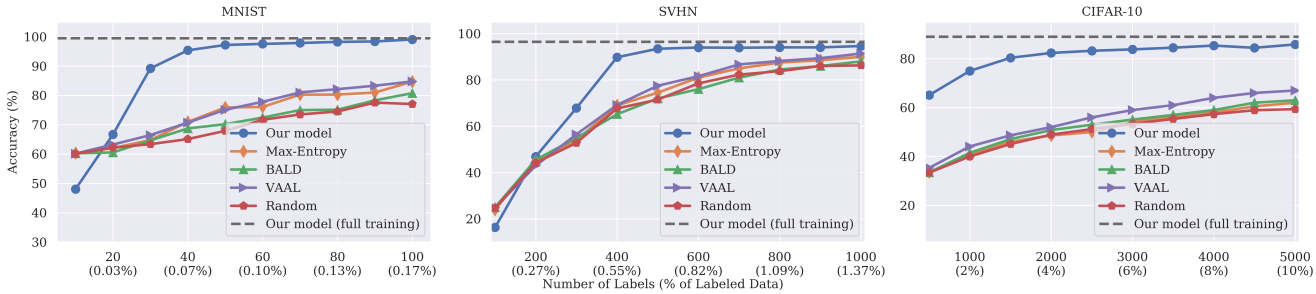

Figure 3: Our model performance compared to Max-Entropy [9], BALD [4], VAAL [10], Random baseline, and full training on MNIST, SVHN, and CIFAR-10 datasets

Discriminator: The discriminator structure follows that of standard conditional GANs, containing both a real/fake discriminator network and a classification network. The loss for the real/fake discriminator network is:

$$\begin{aligned}\mathcal{L}_D^{gan} = &- \mathbb{E}_{(x,y)\sim(X_L,Y_L)}[\log(D(x,E(x,y),y))] \\ &- \mathbb{E}_{x\sim X_U}[\log(D(x,E(x,C(x)),C(x)))] \\ &+ \mathbb{E}_{z\sim p(z),y\sim p(y)}[\log(D(G(z,y),z,y))] \end{aligned} \quad (3)$$

where the first and third term correspond to the discriminator loss for labeled and generated data. The second term corresponds to the discriminator loss for unlabeled data, where the labels for these examples are inferred through the classifier $C(x)$ in the discriminator.

The loss for the classifier in the discriminator is:

$$\begin{aligned}\mathcal{L}_D^{cls} = &- \mathbb{E}_{(x,y)\sim(X_L,Y_L)}[\log p(C(x)=y)] \\ &- \mathbb{E}_{z\sim P(z),y\sim P(y)}[\log p(C(G(z,y))=y)] \end{aligned} \quad (4)$$

Here the first and second term is the cross-entropy loss for real and generated images. Since the classification network and real/fake discriminator network have a shared trunk, the real/fake supervision additionally enables learning a stronger shared feature representation that can further improve classification performance.

Generator: The objective of the generator in the conditional GAN framework is to generate class-conditional images that can fool the discriminator into predicting them as real images. The generator loss is:

$$\mathcal{L}_G^{gan} = \mathbb{E}_{z\sim p(z),y\sim p(y)}[\log(1 - D(G(z,y),z,y))] \quad (5)$$

Encoder: In addition to the generator and discriminator, we add an encoder in our conditional GAN following BiGAN [2]. This has been shown to improve classification performance for more complex data [3], and is a natural choice since our acquisition function already has an encoder that can be shared. Our encoder loss following BiGAN is:

$$\begin{aligned}\mathcal{L}_E^{gan} = &- \mathbb{E}_{(x,y)\sim(X_L,Y_L)}[\log(D(x,E(x,y),y))] \\ &- \mathbb{E}_{x\sim X_U}[\log(D(x,E(x,C(x)),C(x)))] \end{aligned} \quad (6)$$

2.3. Co-training of full model

To perform active learning, the acquisition function and the conditional BiGAN presented above are jointly co-trained, where the losses for the full model are:

$$\mathcal{L}_E + \mathcal{L}_G = \mathcal{L}_E^{acq} + \mathcal{L}_G^{acq} + \lambda_{gan}\mathcal{L}_E^{gan} + \lambda_{gan}\mathcal{L}_G^{gan} \quad (7)$$

$$\mathcal{L}_D = \lambda_{gan}\mathcal{L}_D^{gan} + \lambda_{cls}\mathcal{L}_D^{cls} \quad (8)$$

$$\mathcal{L}_S = \mathcal{L}_S^{acq} \quad (9)$$

After every selection of sampler and labeling of new samples, all components of the model are updated using the new labeled dataset D_L and unlabeled pool D_U.

3. Experiments

We assess performance by measuring the accuracy of the classifier trained during the active learning procedure versus the number of labeled images used in the training on a wide range of complexities. We compare our results against the following baselines: 1) **Uncertainty-based methods**: In these approaches, unlabeled images will be labeled based on the uncertainty in the classifier's prediction. We compare against the Max-Entropy [9] method which is the most prevalent method in this category. 2) **Bayesian methods**: In Bayesian frameworks, probabilistic models such as Gaussian processes and Bayesian neural networks are used to estimate the expected improvement by each query. We report the performance of Bayesian Active Learning by Disagreement (BALD) [4] which uses dropout as an approximation to Bayesian inference. 3) **Representation-based methods**: In these methods, samples are selected by increasing the diversity in each batch. We compare our model with VAAL [10] since they show that Core-set approach [8] (previous SOTA in this category) which minimizes the Euclidean distance between the labeled and unlabeled images cannot generalize well to high dimensional and complex data. 4) **Random**: We show results using random sampling, in which the classifier is trained on only labeled data that is uniformly sampled. 5) **Full training of our model**: This serves as an upper bound for the performance of the model.

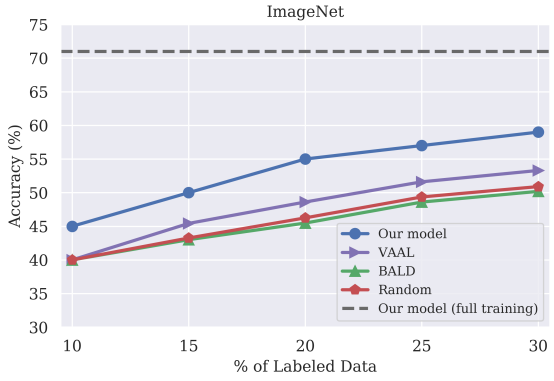

Figure 4: Our model performance compared to Max-Entropy [9], BALD [4], VAAL [10], and Random baseline, and full training on ImageNet dataset.

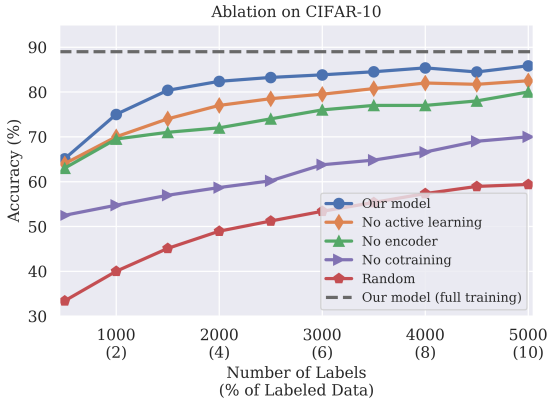

Figure 5: Ablation studies for analysing the effect of each component in our model on CIFAR-10 dataset.

3.1. Performance on a range of complexities

We started with MNIST, SVHN, and CIFAR-10 as classic benchmarks for active learning. We begin our experiments with an initial labeled pool with only 10, 100, and 500 labels for the MNIST, SVHN, and CIFAR-10 datasets respectively, and we add the same number of labels for 10 iterations. Fig. 3 shows the performance of our model compared to the baselines. Our model significantly outperforms all the baselines, as it uses the unlabeled images as well as generated images in the training procedure. The performance gap is more clear especially for very few number of labels.

In order to show the scalability of our approach we also implemented our method on the ImageNet dataset. Due to our limitation in accessing large computational power, we trained a relatively smaller (compared to BigGAN [1]) generator-encoder module as well as a sampler using a similar approach as [10], in order to select the most informative examples in each iteration by mapping both labeled and unlabeled data into the latent space. We then added the high quality fake images generated by a pretrained S^3GAN model (with 256×256 pixels resolution) [6] on 10% of labels to augment our labeled data, and finally train the classifier network ($VGG16$) using the labeled and generated images. Therefore, in our experiments here, we train the encoder and sampler modules separately from the generator and discriminator. Fig. 4 also shows the performance of our model on the ImageNet dataset. As we are not co-training all the parts together, the performance gap from the model trained on all of the labels is larger, however it still significantly outperforms all the baselines.

3.2. Ablation study

For showing the effectiveness of each part of our model, we consider the following variants of ablation and compare the performances on CIFAR-10 dataset: 1) **No active learning**: we remove the sampler and adversarial loss for the encoder and use random sampling at each iteration of training the model. 2) **No encoder**: we have only the generator and discriminator modules, and we use $BALD$ as our sampling strategy. 3) **No co-training**: similar to our experiment on ImageNet, we use VAAL as our acquisition function. However, we add generated images from a pretrained S^3GAN [6] model to train the classifier. This shows the effect of co-training all parts of the model. 4) **Random**: samples are uniformly sampled from the unlabeled pool and the classifier is trained on the labeled data. As shown in Fig. 5, each module contributes to the final performance of the model. This experiment shows the important role of exploiting unlabeled data and co-training all parts of the model in our active learning approach.

References

[1] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *ICLR*, 2019. 4

[2] Jeff Donahue, Philipp Krähenbühl, and Trevor Darrell. Adversarial feature learning. *ICLR*, 2017. 1, 3

[3] Jeff Donahue and Karen Simonyan. Large scale adversarial representation learning. In *NeuRIPS*, 2019. 1, 2, 3

[4] Yarin Gal, Riashat Islam, and Zoubin Ghahramani. Deep bayesian active learning with image data. In *ICML*, 2017. 1, 3, 4

[5] Neal Jean, Marshall Burke, Michael Xie, W Matthew Davis, David B Lobell, and Stefano Ermon. Combining satellite imagery and machine learning to predict poverty. *Science*, 2016. 1

[6] Mario Lucic, Michael Tschannen, Marvin Ritter, Xiaohua Zhai, Olivier Bachem, and Sylvain Gelly. High-fidelity image generation with fewer labels. *ICML*, 2019. 2, 4, 5

[7] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *NeuRIPS*, 2016. 5

[8] Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. In *ICLR*, 2018. 1, 3

[9] Burr Settles. Active learning literature survey. Technical report, University of Wisconsin-Madison Department of Computer Sciences, 2009. 1, 3, 4

[10] Samarth Sinha, Sayna Ebrahimi, and Trevor Darrell. Variational adversarial active learning. In *ICCV*, 2019. 1, 2, 3, 4

[11] Serena Yeung, Francesca Rinaldo, Jeffrey Jopling, Bingbin Liu, Rishab Mehra, N Lance Downing, Michelle Guo, Gabriel M Bianconi, Alexandre Alahi, Julia Lee, et al. A computer vision system for deep learning-based detection of patient mobilization activities in the icu. *NPJ digital medicine*, 2019. 1

[12] Jia-Jie Zhu and José Bento. Generative adversarial active learning. *arXiv preprint arXiv:1702.07956*, 2017. 1

Appendix

In this section, we review the summary of the details in our experiments including the datasets and the hyper-parameters.

Datasets: The datasets used in this works along with the budget size associated with them are showed in Table 1.

The MNIST dataset contains 28×28 images of handwritten digits of 10 classes (with $60k$ training samples and $10k$ test samples). The SVHN and CIFAR-10 datasets have images of size 32×32 of 10 classes (with 73257 and $50k$ training samples, and 26032 and $10k$ test samples, respectively).

ImageNet is a large dataset with more than $1.2M$ images of 1000 classes. The validation set for this dataset contains $50k$ images. We augment our dataset by horizontally flipping the images. Then we resize all the images into 224×224 pixels and normalize them before feeding them into the model.

Architecture and Hyper-parameters: The hyper-parameters that are tuned for our model are also summarized in Table 2. The number of epochs in the table shows the number of epochs for each training iteration (after each acquisition). We found starting from the trained model in the previous iteration to be useful, as it makes the whole training procedure faster by starting from good initialization at each iteration.

For the MNIST dataset, we used a 3-layer MLP for both the encoder and generator, and a 5-layer MLP with added Gaussian noise (with standard deviation of 0.5) between the layers for the discriminator. Our sampler module for this dataset is also a 5-layer MLP. Adam with learning rate of 1×10^{-3} is chosen as the optimizer for the encoder and generator modules, and with learning rate of 3×10^{-3} for the discriminator and sampler modules. For the SVHN and CIFAR-10 datasets, we use a CNN with 3 hidden layers for both the encoder and generator, and 3 convolutional blocks, each with 3 layers and dropout (with rate of 0.5) between the blocks, for the discriminator. Our sampler module for this dataset is the same 5-layer MLP. Adam with learning rate of 3×10^{-4} is chosen as the optimizer for the encoder and generator modules, and with learning rate of 6×10^{-4} for the discriminator and sampler modules. We also use $\lambda_{gan} = \lambda_{cls} = 1, \lambda_{trd} = \lambda_{adv} = 0.001$, and $\beta = 1$ as our hyper parameters. For these datasets we use a latent space with $d_z = 100$ dimensions. We also use the feature matching technique which is proposed in [7] for matching the features in the discriminator for generated and unlabeled images. In this way, generated images will be a better representative of the unlabeled parts of the dataset.

For ImageNet we use the same architecture and hyper-parameters as S3GAN in [6], which is the sate-of-the-art in image generation on the ImageNet dataset. The rest of hyper-parametrs are same as the previous experiments except the latent space that has $d_z = 64$ dimensions. (We use 64 dimensional latent space to be able to compare directly with VAAL baseline which uses the same dimensionality.)

Full Training Algorithm: The loss functions are defined in section 2.3 for encoder and generator (that are shared between acquisition function and the BiGAN), sample and discriminator (that are not shared). The full training algorithm for our model is presented in Alg. 1.

Algorithm 1 Adversarial Representation Active Learning

Input: Labeled pool $\mathcal{D}_L$, Unlabeled pool $\mathcal{D}_U$, Labeling budget, Initialized models for θ_G, θ_E, θ_D, and θ_S

repeat
  Pick samples X_s from $\mathcal{D}_U$ with $\min_b\{S(E(X_s))\}$
  Label the selected samples by oracle, which forms (X_s, Y_o)
  Update labeled and unlabeled datasets:
  $\mathcal{D}_L \leftarrow \mathcal{D}_L \cup (X_s, Y_o)$
  $\mathcal{D}_U \leftarrow \mathcal{D}_U - X_s$
  for $e = 1$ to epochs **do**
    Sample $(x_L, y_L) \sim \mathcal{D}_L$
    Sample $x_U \sim \mathcal{D}_U$
    Compute $\mathcal{L}_E + \mathcal{L}_G$ using Eq. 7
    Update E and G by descending stochastic gradient
    $\theta'_E \leftarrow \theta_E - \alpha_E \nabla(\mathcal{L}_E + \mathcal{L}_G)$
    $\theta'_G \leftarrow \theta_G - \alpha_E \nabla(\mathcal{L}_E + \mathcal{L}_G)$
    Compute $\mathcal{L}_D$ using Eq. 8
    Update D by descending stochastic gradient
    $\theta'_D \leftarrow \theta_D - \alpha_D \nabla \mathcal{L}_D$
    Compute $\mathcal{L}_S$ using Eq. 9
    Update S by descending stochastic gradient
    $\theta'_S \leftarrow \theta_S - \alpha_S \nabla \mathcal{L}_S$
  end for
until Total labeling budget is finished
return Trained θ_E, θ_G, θ_D, and θ_S

Dataset	Image Size	Number of Classes	Training Size	Test Size	Initially Labeled	Budget
MNIST	28×28	10	60000	10000	10	10
SVHN	32×32	10	73257	26032	100	100
CIFAR-10	32×32	10	50000	10000	500	500
ImageNet	224×224	1000	1281167	50000	128120	64060

Table 1: The datasets used in our experiments

Dataset	α_D	α_E	α_S	λ_{gan}	λ_{cls}	λ_{trd}	λ_{adv}	β	d_z	batch size	epochs
MNIST	3×10^{-3}	1×10^{-3}	3×10^{-3}	1	1	0.001	0.001	1	100	100	100
SVHN	1×10^{-3}	1×10^{-3}	1×10^{-3}	1	1	0.001	0.001	1	100	64	500
CIFAR-10	6×10^{-4}	3×10^{-4}	6×10^{-4}	1	1	0.001	0.001	1	100	100	1200
ImageNet	1×10^{-3}	1×10^{-1}	1×10^{-3}	1	1	1	10	1	64	64	100

Table 2: The hyper-parameters used in our experiments