Variational Adversarial Active Learning

Samarth Sinha*
University of Toronto
samarth.sinha@mail.utoronto.ca

Sayna Ebrahimi*
UC Berkeley
sayna@eecs.berkeley.edu

Trevor Darrell
UC Berkeley
trevor@eecs.berkeley.edu

Abstract

Active learning aims to develop label-efficient algorithms by sampling the most representative queries to be labeled by an oracle. We describe a pool-based semi-supervised active learning algorithm that implicitly learns this sampling mechanism in an adversarial manner. Unlike conventional active learning algorithms, our approach is task agnostic, i.e., it does not depend on the performance of the task for which we are trying to acquire labeled data. Our method learns a latent space using a variational autoencoder (VAE) and an adversarial network trained to discriminate between unlabeled and labeled data. The mini-max game between the VAE and the adversarial network is played such that while the VAE tries to trick the adversarial network into predicting that all data points are from the labeled pool, the adversarial network learns how to discriminate between dissimilarities in the latent space. We extensively evaluate our method on various image classification and semantic segmentation benchmark datasets and establish a new state of the art on CIFAR10/100, Caltech-256, ImageNet, Cityscapes, and BDD100K. Our results demonstrate that our adversarial approach learns an effective low dimensional latent space in large-scale settings and provides for a computationally efficient sampling method.[1]

1. Introduction

The recent success of learning-based computer vision methods relies heavily on abundant annotated training examples, which may be prohibitively costly to label or impossible to obtain at large scale [10]. In order to mitigate this drawback, active learning [4] algorithms aim to incrementally select samples for annotation that result in high classification performance with low labeling cost. Active learning has been shown to require relatively fewer training instances when applied to computer vision tasks such as image classification [43, 31, 16, 1] and semantic segmentation [55, 30, 21].

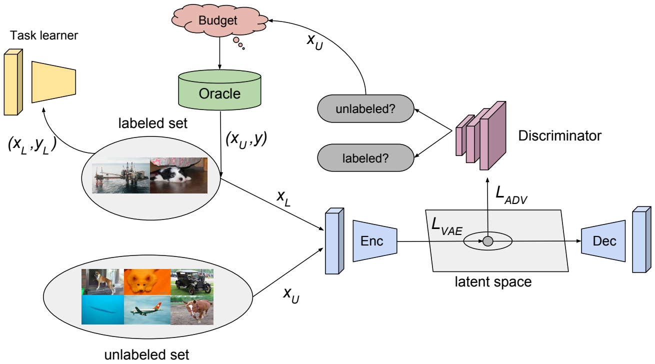

Figure 1. Our model (VAAL) learns the distribution of labeled data in a latent space using a VAE optimized using both reconstruction and adversarial losses. A binary adversarial classifier (discriminator) predicts unlabeled examples and sends them to an oracle for annotations. The VAE is trained to fool the adversarial network to believe that all the examples are from the labeled data while the discriminator is trained to differentiate labeled from unlabeled samples. Sample selection is entirely separate from the main-stream task for which we are labeling data inputs, making our method to be *task-agnostic*

This paper introduces a pool-based active learning strategy which learns a low dimensional latent space from labeled and unlabeled data using Variational Autoencoders (VAEs). VAEs have been well-studied and valued for both their generative properties as well as their ability to learn rich latent spaces. Our method, Variational Adversarial Active Learning (VAAL), selects instances for labeling from the unlabeled pool that are *sufficiently* different in the latent space learned by the VAE to maximize the performance of the representation learned on the newly labeled data. Sample selection in our method is performed by an adversarial network which classifies which pool the instances belong to (labeled or unlabeled) and does not depend on the task or tasks for which are trying to collect labels.

Our VAE learns a latent representation in which the sets of labeled and unlabeled data are mapped into a common

*Authors contributed equally, listed alphabetically.

[1] Our code and data are available at https://github.com/sinhasam/vaal.

embedding. We use an adversarial network in this space to correctly classify one from another. The VAE and the discriminator are framed as a two-player mini-max game, similar to GANs [20] such that the VAE is trained to learn a feature space to *trick* the adversarial network into predicting that all datapoints, from both the labeled and unlabeled sets, are from the labeled pool while the discriminator network learns how to discriminate between them. The strategy follows the intuition that once the active learner is trained, the probability associated with discriminator's predictions effectively estimates how representative each sample is from the pool that it has been deemed to be from. Therefore, instead of explicitly measuring uncertainty on the main task, we aim to choose points that would yield high uncertainty and thus are samples that are not well represented in the labeled set. We additionally consider oracles with different levels of labeling noise and demonstrate the robustness of our method to such noisy labels. In our experiments, we demonstrate superior performance on a variety of large scale image classification and segmentation datasets, and outperform current state of the art methods both in performance and computational cost.

2. Related Work

Active learning: Current approaches can be categorized as query-acquiring (pool-based) or query-synthesizing methods. Query-synthesizing approaches use generative models to generate *informative* samples [34, 36, 58] whereas pool-based algorithms use different *sampling strategies* to determine how to select the most *informative* samples. Since our work lies in the latter line of research, we will mainly focus on previous work in this direction.

Pool-based methods can be grouped into three major categories as follows: uncertainty-based methods [21, 53, 1], representation-based models [43], and their combination [55, 40]. Pool-based methods have been theoretically proven to be effective and achieve better performance than the random sampling of points [45, 14, 17]. Sampling strategies in pool-based algorithms have been built upon several methods, which are surveyed in [44], such as information-theoretic methods [32], ensembles methods [37, 14] and uncertainty heuristics such as distance to the decision boundary [50] and conditional entropy [31]. Uncertainty-based pool-based models are proposed in both Bayesian [16] and non-Bayesian frameworks. In the realm of Bayesian frameworks, probabilistic models such as Gaussian processes [25, 41] or Bayesian neural networks [9] are used to estimate uncertainty. Gal & Gharamani [16, 15], also showed the relationship between uncertainty and dropout to estimate uncertainty in prediction in neural networks and applied it for active learning in small image datasets using shallow [15] and deep [16] neural networks. In non-Bayesian classical active learning approaches, uncertainty heuristics such as distance from the decision boundary, highest entropy, and expected risk minimization have been widely investigated [3, 50, 54]. However, it was shown in [43] that such classical techniques do not scale well to deep neural networks and large image datasets. Instead, they proposed to use Core-sets, where they minimize the Euclidean distance between the sampled points and the points that were not sampled in the feature space of the trained model [43]. Using an ensemble of models to represent uncertainty was proposed by [30, 55], but [38] showed that using ensembles does not always yield high diversity in predictions which results in sampling redundant instances.

Representation-based methods rely on selecting few examples by increasing *diversity* in a given batch [43, 8]. The Core-set technique was shown to be an effective representation learning method for large scale image classification tasks [43] and was theoretically proven to work best when the number of classes is small. However, as the number of classes grows, it deteriorates in performance. Moreover, for high-dimensional data, using distance-based representation methods, like Core-set, appears to be ineffective because in high-dimensions p-norms suffer from the curse of dimensionality which is referred to as the *distance concentration phenomenon* in the computational learning literature [12]. We overcome this limitation by utilizing VAEs which have been shown to be effective in unsupervised and semi-supervised representation learning of high dimensional data [28, 48].

Methods that aim to combine uncertainty and *representativeness* use a two-step process to select the points with high uncertainty as of the most representative points in a batch. A hybrid framework combining uncertainty using conditional entropy and representation learning using information density was proposed in [31] for classification tasks. A weakly supervised learning strategy was introduced in [53] that trains the model with pseudo labels obtained for instances with high *confidence* in predictions. However, for a fixed performance goal, they often need to sample more instances per batch compared to other methods. Furthermore, in [30] it was shown that having the representation step may not be necessary followed by suggesting an ensemble method that outperformed competitive approaches such as [55] which uses uncertainty together with Core-sets. While we show that our model outperforms both [30] and [55], we argue that VAAL achieves this by learning the representation and uncertainty together such that they act in favor of each other while being independent from the main-stream task, resulting in better active learning performance.

Variational autoencoders: Autoencoders have long been used to effectively learn a feature space and representation [2, 42]. A Variational AutoEncoder [28] is an example of a latent variable model that follows an encoder-decoder architecture of classical autoencoders which places a prior distribution on the feature space distribution and uses an Expected Lower Bound to optimize the learned posterior. Adversarial autoencoders are a family of autoencoders which minimize the adversarial loss in the latent space between a sample from the prior and the posterior distribution [35]. Prior work has investigated uncertainty modeling using a VAE for sequence generation in language applications [7],

Active learning for semantic segmentation: Segmentation labeling is one of the most expensive annotations to collect. Active learning in the literature has been broadly investigated for labeling medical images as it is one of the most prevailing applications of AL where only human experts with sophisticated knowledge are capable of providing labels and therefore, improving this process would reduce a lot of time and effort for them. Suggestive Annotation (SA) [55] uses uncertainty obtained from an ensemble of models trained on the labeled data and Core-sets for choosing representative data points in a two-step strategy. [30] also proposed an active learning algorithm for image segmentation using an ensemble of models, but they empirically showed their proposed information-theoretic heuristic for uncertainty is equal in performance to SA, without using Core-sets. [21] extended the work by [16] and proposed using Monte-Carlo dropout masks on the unlabeled images using a trained model and calculating the uncertainty on the predicted labels of the unlabeled images. Some active learning strategies developed for image classification can also be used for semantic segmentation. Core-sets and max-entropy strategies can both be used for active learning in semantic segmentation [43, 3].

Adversarial learning: Adversarial learning has been used for different problems such as generative models [20], representation learning [35, 39], domain adaptation [52, 24], deep learning robustness and security [33, 51] etc. The use of an adversarial network enables the model to train in a fully-differentiable by adjusting to solving the *minimax* optimization problem [20]. The adversarial network used in the feature space has been extensively researched in the representation learning and domain adaptation literature to efficiently learn a useful feature space for the task [35, 26, 49, 52, 24].

3. Adversarial Learning of Variational Autoencoders for Active Learning

Let (x_L, y_L) be a sample pair belonging to the pool of labeled data (X_L, Y_L). X_U denotes a much larger pool of samples (x_U) which are not yet labeled. The goal of the active learner is to train the most label-efficient model by iteratively querying a fixed sampling *budget*, b number of the most informative samples from the unlabeled pool $(x_U \sim X_U)$, using an acquisition function to be annotated by the oracle such that the expected loss is minimized.

3.1. Transductive representation learning.

We use a β-variational autoencoder for representation learning in which the encoder learns a low dimensional space for the underlying distribution using a Gaussian prior and the decoder reconstructs the input data. In order to capture the features that are missing in the representation learned on the labeled pool, we can benefit from using the unlabeled data and perform transductive learning. The objective function of the β-VAE is minimizing the variational lower bound on the marginal likelihood of a given sample formulated as

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

where q_ϕ and p_θ are the encooder and decoder parameterized by ϕ and θ, respectively. $p(z)$ is the prior chosen as a unit Gaussian, and β is the Lagrangian parameter for the optimization problem. The reparameterization trick is used for proper calculation of the gradients [28].

3.2. Adversarial representation learning

The representation learned by the VAE is a mixture of the latent features associated with both labeled and unlabeled data. An ideal active learning agent is assumed to have a perfect sampling strategy that is capable of sending the most *informative* unlabeled data to the oracle. Most of the sampling strategies rely on the model's uncertainty, i.e, the more uncertain the model is on the prediction, the more informative that specific unlabeled data must be. However, this introduces vulnerability to the outliers. In contrast we train an adversarial network for our sampling strategy to learn how to distinguish between the encoded features in the latent space. This adversarial network is analogous to discriminators in GANs where their role is to discriminate between fake and real images created by the generator. In VAAL, the adversarial network is trained to map the latent representation of $z_L \cup z_U$ to a binary label which is 1 if the sample belongs to X_L and is 0, otherwise. The key to our approach is that the VAE and the adversarial network are learned together in an adversarial fashion. While the VAE maps the labeled and unlabeled data into the same latent space with similar probability distribution $q_\phi(z_L|x_L)$ and $q_\phi(z_U|x_U)$, it fools the discriminator to classify all the inputs as labeled. On the other hand, the discriminator attempts to effectively estimate the probability that the data comes from the unlabeled data. We can formulate the objective function for the adversarial role of the VAE as a binary cross-entropy loss as below

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

The objective function to train the discriminator is also given as below

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

By combining Eq. (1) and Eq. (2) we obtain the full objective function for the VAE in VAAL as below

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

where λ_1 and λ_2 are hyperparameters that determine the effect of each component to learn an effective variational adversarial representation.

The task module denoted as T in Fig. (1), learns the task for which the active learner is being trained. T is trained separately from the active learner as they do not depend on each other at any step. We report results below on image classification and semantic segmentation tasks, using VGG16 [47] and dilated residual network (DRN) architecture [56] with an unweighted cross-entropy cost function. Our full algorithm is shown in Alg. 1.

3.3. Sampling strategies and noisy-oracles

The labels provided by the oracles might vary in how *accurate* they are depending on the quality of available human resources. For instance, medical images annotated by expert humans are assumed to be more accurate than crowd-sourced data collected by non-expert humans and/or available information on the cloud. We consider two types of oracles: an ideal oracle which always provides correct labels for the active learner, and a noisy oracle which non-adversarially provides erroneous labels for some specific classes. This might occur due to similarities across some classes causing ambiguity for the labeler. In order to present this oracle realistically, we have applied a targeted noise on visually similar classes. The sampling strategy in VAAL is shown in Alg. (2). We use the probability associated with the discriminator's predictions as a score to collect b number of samples in every batch predicted as "unlabeled" with the lowest confidence to be sent to the oracle. Note that the closer the probability is to zero, the more likely it is that it comes from the unlabeled pool. The key idea to our approach is that instead of relying on the performance of the training alforithm on the main-stream task, which suffers from being inaccurate specially in the beginning, we select samples based on the likelihood of their *representativeness* with respect to other samples which discriminator thinks belong to the unlabeled pool.

4. Experiments

We begin our experiments with an initial labeled pool with 10% of the training set labeled. The budget size per batch is equal to 5% of the training dataset. The pool of unlabeled data contains the rest of the training set from which samples are selected to be annotated by the oracle. Once labeled, they will be added to the initial training set and training is repeated on the new training set. We assume the oracle is *ideal* unless stated otherwise. [2]

```
Algorithm 1 Variational Adversarial Active Learning
Input: Labeled pool $(X_L, Y_L)$, Unlabeled pool $(X_U)$, Initialized models for $\theta_T$, $\theta_{VAE}$, and $\theta_D$
Input: Hyperparameters: epochs, $\lambda_1$, $\lambda_2$, $\alpha_1$, $\alpha_2$, $\alpha_3$
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 $\theta_T, \theta_{VAE}, \theta_D$
```

```
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 $X_L, X_U$
```

Datasets. We have evaluated VAAL on two common vision tasks. For image classification we have used CIFAR10 [29] and CIFAR100 [29] both with 60K images of size 32×32, and Caltech-256 [22] which has 30607 images of size 224×224 including 256 object categories. For a better understanding of the scalability of VAAL we have also experimented with ImageNet [6] with more than 1.2M images of 1000 classes. For semantic segmentation, we evaluate our method on BDD100K [57] and Cityscapes [5] datasets both of which have 19 classes. BDD100K is a diverse driving video dataset with 10K images with full-frame instance segmentation annotations collected from distinct locations in the United State. Cityscapes is also another large scale driving video dataset containing 3475 frames with instance segmentation annotations recorded in street scenes from 50 different cities in Europe. The statistics of these datasets are summarized in Table 2 in the appendix.

Performance measurement. We evaluate the performance

[2] Data and code required to reproduce all plots are provided at `https://github.com/sinhasam/vaal/blob/master/plots/plots.ipynb`.

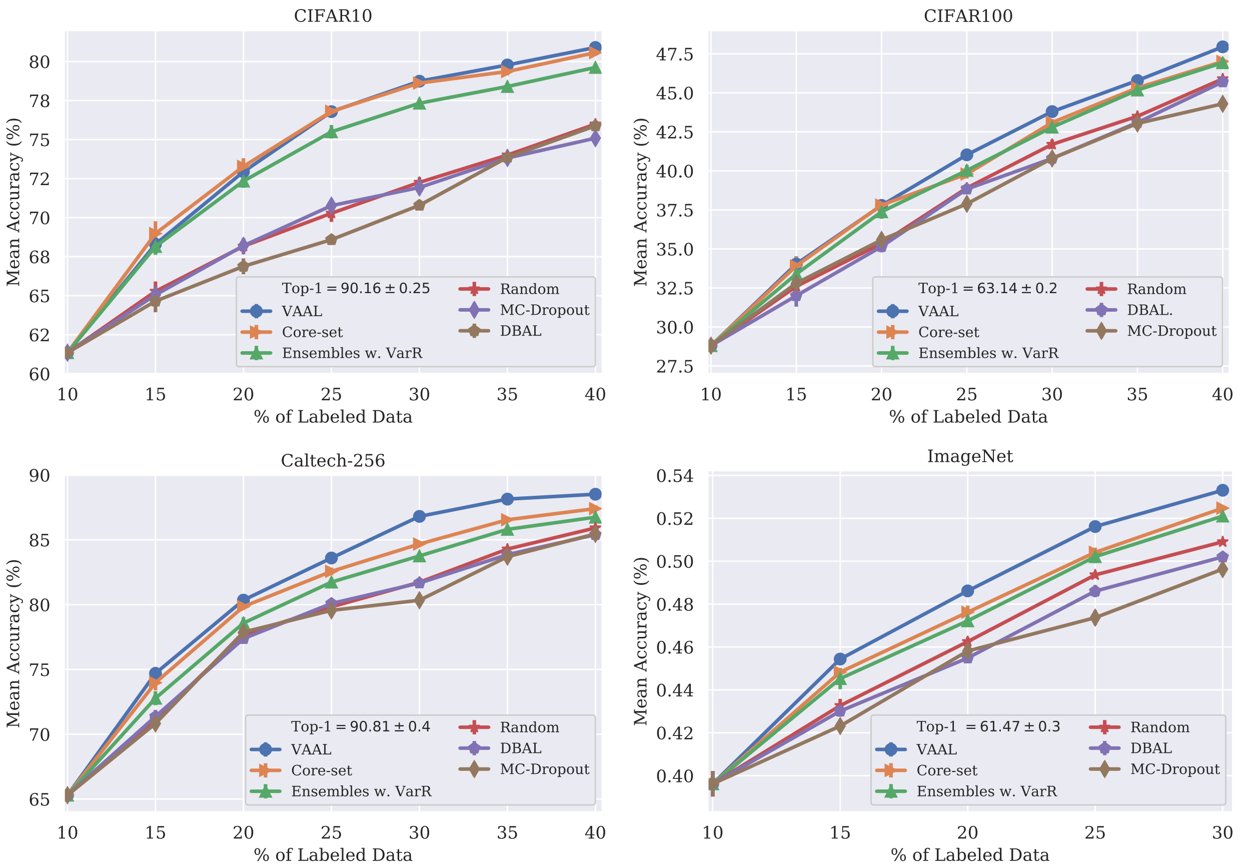

Figure 2. VAAL performance on classification tasks using CIFAR10, CIFAR100, Caltech-256, and ImageNet compared to Core-set [43], Ensembles w. VarR [1], MC-Dropout [15], DBAL [16], and Random Sampling. Best visible in color. Data and code required to reproduce are provided in our code repository

of VAAL in image classification and segmentation by measuring the accuracy and mean IoU, respectively achieved by T trained with 10%, 15%, 20%, 25%, 30%, 35%, 40% of the total training set as it becomes available with labels provided by the oracle. Results for all our experiments, except for ImageNet, are averaged over 5 runs. ImageNet results however, are obtained by averaging over 2 repetitions using 10%, 15%, 20%, 25%, 30% of the training data.

4.1. VAAL on image classification benchmarks

Baselines. We compare our results using VAAL for image classification against various approaches including Core-set [43], Monte-Carlo Dropout [15], and Ensembles using Variation Ratios (Ensembles w. VarR) [1, 13]. We also show the performance of deep Bayesian AL (DBAL) by following [16] and perform sampling using their proposed max-entropy scheme to measure uncertainty [46]. We also show the results using *random sampling* in which samples are uniformly sampled at random from the unlabeled pool. This method still serves as a competitive baseline in active learning. Moreover, we use the mean accuracy achieved on the entire dataset as an upper bound which does not adhere to the active learning scenario.

Implementation details. We used random horizontal flips for data augmentation. The architecture used in the task module for image classification is VGG16 [47] with Xavier initialization [19] and β-VAE has the same architecture as the Wasserstein autoencoder [49] with latent dimensionality given in Table 3 in the appendix. The discriminator is a 5-layer multilayer perceptron (MLP) and Adam [27] is used as the optimizer for all these three modules with an equal learning rate of 5×10^{-4} and batch size of 64. However, for ImageNet, learning rate varies across the modules such that the task learner has a learning rate of 1×10^{-1} while the VAE and the discriminator have a learning rate of 5×10^{-3}. Training continues for 100 epochs in ImageNet and for 100 epochs in all other datasets. The budget size for classification experiments is chosen to be 5% of the full training set, which is equivalent to 2500, 2500, 1530, and 64060 for CIFAR10, CIFAR100, Caltech-256, and ImageNet, respectively in VAAL and all other baselines. A complete list of hyperparameters used in our model are found through a grid search and are tabulated in Table 3 in the appendix.

VAAL performance CIFAR10/100 and Caltech-256. Figure 2 shows performance of VAAL compared to prior works. On CIFAR10, our method achieves mean accuracy of 80.9% by using 40% of the data whereas using the entire dataset yields accuracy of 90.16%, denoted as Top-1 accuracy in Fig. 2. Comparing the mean accuracy values for data ratios above 15% shows that VAAL evidently outperforms random sampling, DBAL, and MC-Dropout while beating Ensembles by a smaller margin and becoming on-par with Core-set. On CIFAR100, VAAL remains competitive with Ensembles w. VarR and Core-set, and outperforms all other baselines. The maximum achievable mean accuracy is 63.14% on CIFAR100 using 100% of the data while VAAL achieves 47.95% by only using 40% of it. Moreover, for data ratios above 20% of labeled data, VAAL consistently requires $\sim$ 2.5% less number of labels compared to Core-set or Ensembles w. VarR in order to achieve the same accuracy, which is equal to 1250 labels. On Caltech-256, which has real images of object categories, VAAL consistently outperforms all baselines by an average margin of 1.78% from random sampling and 1.01% from the most competitive baseline, Core-set. DBAL method performs nearly identical to random sampling while MC-Dropout yields lower accuracies than random sampling. By looking at the number of labels required to reach a fixed performance, for instance, 83.6%, VAAL needs 25% of data (7651 images) to be labeled whereas this number is approximately 9200 and 9500 for Core-set and Ensemble w. VarR, respectively. Random sampling, DBAL, and MC-Dropout all need more than 12200 images.

As can be seen in Fig. 2, VAAL outperforms Core-set with higher margins as the number of classes increases from 10 to 100 to 256. The theoretical analysis shown in [43] confirms that Core-set is more effective when fewer classes are present due to the negative impact of high dimensionality on p-norms in the Core-set method.

VAAL performance on ImageNet. ImageNet [6] is a challenging large scale dataset which we use to show scalability of our approach. Fig. 2 shows that we improve the state-of-the-art by 100% increase in the gap between the accuracy achieved by the previous state-of-the-art methods (Core-set and Ensemble) and random sampling. As can be seen in Fig. 2, this improvement can be also viewed in the number of samples required to achieve a specific accuracy. For instance, the accuracy of 48.61% is achieved by VAAL using 256K number of images whereas Core-set and Ensembles w. VarR should be provided with almost 32K more labeled images to obtain the same performance. Random sampling remains as a competitive baseline as both DBAL and MC-Dropout perform below that.

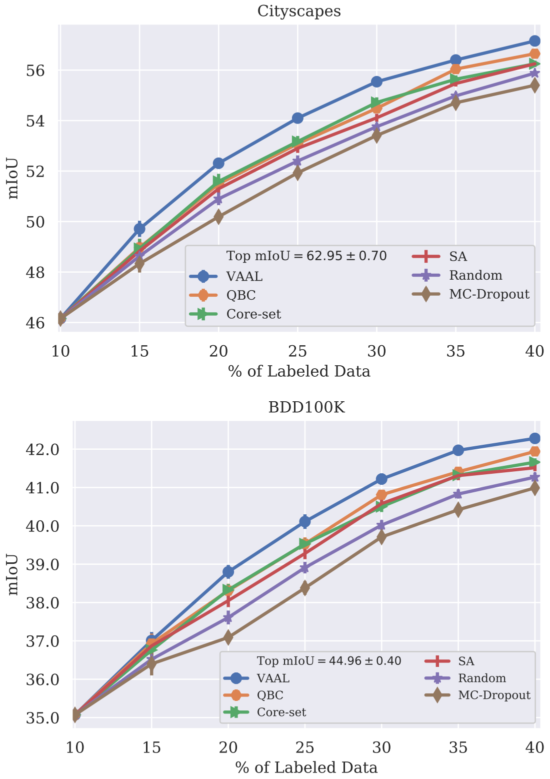

Figure 3. VAAL performance on segmentation tasks using Cityscapes and BDD100K compared to QBC [30], Core-set [43], MC-Dropout [15], and Random Sampling. Data and code required to reproduce are provided in our code repository

4.2. VAAL on image segmentation benchmarks

Baselines. We evaluate VAAL against state-of-the-art AL approaches for image segmentation including Core-set [43], MC-Dropout [21], Query-By-Committee (QBC) [30], and suggestive annotation (SA)[55]. SA is a hybrid ensemble method that uses bootstrapping for uncertainty estimation [11] and core-set for measuring *representativeness*.

Implementation details. Similar to the image classification setup, we used random horizontal flips for data augmentation. The β-VAE is a Wasserstein autoencoder [49], and the discriminator is also a 5-layer MLP. The architecture used in the task module for image segmentation is DRN [56] and Adam with a learning rate of 5×10^{-4} is chosen as the optimizer for all three modules. The batch size is set as 8 and training stops after 50 epochs in both datasets. The budget size used in VAAL and all baselines is set as 400 and 150 for BDD100K and Cityscapes, respectively. All hyperparameteres are shown in Table 3 in the appendix

VAAL performance on Cityscapes and BDD100K. Figure 3 demonstrates our results on the driving datasets compared with four other baselines as well as the reference random sampling. As we also observed in section 4.1

Core-set performs better with fewer number of classes in image classification tasks [43] . However, the large gap between VAAL and Core-set, despite only having 19 classes, suggests that Core-set and Ensemble-based methods (QBC in here) suffer from high dimensionality in the inputs (688 × 688 as opposed to thumbnail 32 × 32 images used in CIFAR10/100). QBC and Core-set, and SA (Core-set + QBC) perform nearly identical, while MC-Dropout remains less effective than random sampling. VAAL consistently demonstrate significantly better performance by achieving the highest mean IoU on both Cityscapes and BDD100K across different labeled data ratios. VAAL is able to achieve $\%$mIoU of 57.2 and 42.3 using only 40% labeled data while the maximum mIoU we obtained using 100% of these datasetes is 62.95 and 44.95 on Cityscapes and BDD100K, respectively. In terms of required labels by each method, on Cityscapes VAAL needs 743 annotations to reach 54.1% of mIoU whereas QBC, Core-set, SA, random sampling, MC-Dropout demand nearly 800, 890, 910, 960, and 1041 labels, respectively. Similarly on BDD100K in order to reach 41% of mIoU, other baselines need $5\% - 10\%$ more annotations than VAAL requires only 30%. Considering the difficulties in full frame instance segmentation, VAAL is able to effectively reduce the required time and effort for such dense annotations.

5. Analyzing VAAL in Detail

In this section, we take a deeper look into our model by first performing ablation and then evaluating the effect of possible biases and noise on its performance. Sensitivity of VAAL to budget size is also explored in 5.2.

5.1. Ablation study

Figure 4 presents our ablation study to inspect the contribution of the key modules in VAAL including the VAE, and the discriminator (D). We perform ablation on the segmentation task which is more challenging than classification and we use BDD100K as it is larger than Cityscapes. The variants of ablations we consider are: 1) eliminating VAE, 2) Frozen VAE with D, 3) eliminating D. In the first ablation, we explore the role of the VAE as the representation learner by having only a discriminator trained on the image space to discriminate between labeled and unlabeled pool. As shown in Fig. 4, this setting results in the discriminator to only memorize the data and yields the lowest performance. Also, it reveals the key role of the VAE in not only learning a rich latent space but also playing an effective mini-max game with the discriminator to avoid overfitting. In the second ablation scenario, we add a VAE to the previous setting to encode-decode a lower dimensional space for training D. However, here we avoid training the VAE and hence merely explore its role as an autoencoder. This setting performs better than having only the D trained in a high dimensional space, but yet performs similar or worse than random sampling suggesting that discriminator failed at learning *representativeness* of the samples in the unlabeled pool. In the last ablation, we explore the role of the discriminator by training only a VAE that uses 2-Wasserstein distance from the cluster-centroid of the labeled dataset as a heuristic to explicitly measure uncertainty. For a multivariate isotropic Gaussian distribution, the closed-form solution for the 2-Wasserstein distance between two probability distributions [18] can be written as

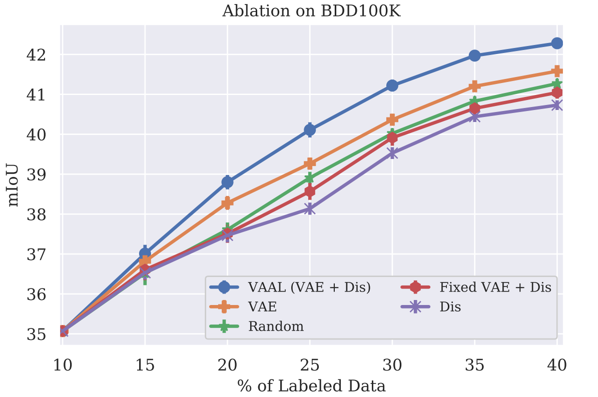

Figure 4. Ablation results on analyzing the effect of the VAE and the discriminator denoted as Dis here. Data and code required to reproduce are provided in our code repository

$$\mathrm{W_{ij}} = \left[||\mu_i - \mu_j||_2^2 + ||\Sigma_i^{\frac{1}{2}} - \Sigma_j^{\frac{1}{2}}||_{\mathcal{F}}^2 \right]^{\frac{1}{2}} \tag{5}$$

where $||.||_{\mathcal{F}}$ represents the Frobenius norm and μ_i, Σ_i denote the μ, Σ predicted by the encoder and μ_j, Σ_j are the mean and variance for the normal distribution over the labeled data from which the latent variable z is generated. In this setting, we see an improvement over random sampling which shows the effect of explicitly measuring the uncertainty in the learned latent space. However, VAAL appears to outperform all these scenarios by implicitly learning the uncertainty over the adversarial game between the discriminator and the VAE.

5.2. VAAL's Robustness

Effect of biased initial labels in VAAL. We investigate here how bias in the initial labeled pool affect VAAL's performance as well as other baselines on CIFAR100 dataset. Intuitively, bias can affect the training such that it causes the initially labeled samples to be not representative of the underlying data distribution by being inadequate to cover most of the regions in the latent space. We model a possible form of bias in the labeled pool by not providing labels for m chosen classes at random and we compare it to the case where samples are randomly selected from all classes.

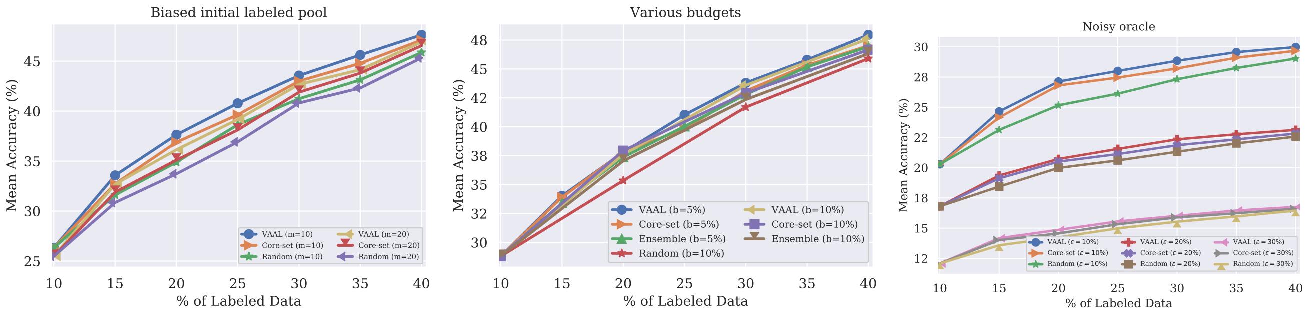

Figure 5. Analyzing robustness of VAAL to noisy labels, budget size, and biased initial labeled pool using CIFAR100. Best viewed in color. Data and code required to reproduce are provided in our code repository

We exclude the data for $m = 10$ and $m = 20$ classes at random in the initial labeled pool to explore how it affects the performance of the model. Figure 5 shows for $m = 10$ and $m = 20$, VAAL is superior to Core-set and random sampling in selecting informative samples from the classes that were underrepresented in the initial labeled set. We also observe that VAAL with $m = 20$ missing classes performs nearly identical to Core-Set and significantly better than random sampling where each has half number of missing classes.

Effect of budget size on performance. Figure 5 illustrates the effect of the budget size on our model compared to the most competitive baselines on CIFAR100. We repeated our experiments in section 4.1 for a lower budget size of $b = 5\%$. We observed that VAAL outperforms Core-Set and Ensemble w. VarR, as well as random sampling, on both budget sizes of $b = 5\%$ and $b = 10\%$. Core-set comes at the second best method followed by Ensemble in Fig 5. We note that $b = 5\%$ for all methods, including VAAL, has a slightly better performance compared to when $b = 10\%$ which is expected to happen because a larger sampled batch results in adding redundant samples instead of more informative ones.

Noisy vs. ideal oracle in VAAL. In this analysis we investigate the performance of VAAL in the presence of noisy data caused by an inaccurate oracle. We assume the erroneous labels are due to the ambiguity between some classes and are not adversarial attacks. We model the noise as targeted noise on specific classes that are *meaningful* to be mislabeled by a human labeler. We used CIFAR100 for this analysis because of its hierarchical structure in which 100 classes in CIFAR100 are grouped into 20 super-classes. Each image comes with a *fine* label (the class to which it belongs) and a *coarse* label (the super-class to which it belongs). We randomly change the ground truth labels for 10%, 20% and 30% of the training set to have an incorrect label within the same super-class. Figure 5 shows how a noisy oracle effects the performance of VAAL, Core-set, and random sampling. Because both Core-set and VAAL do not depend on the task learner, we see that the relative performance is comparable to the ideal oracle presented in Section 4.1. Intuitively, as the percentage of noisy labels increases, all of the active learning strategies converge to random sampling.

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**

Table 1. Time taken to sample, for one sampling iteration, from the unlabeled pool on CIFAR10 dataset. For a fair comparison we use the same PyTorch data-loader across VAAL and baselines.

Choice of the network architecture in T. In order to assure VAAL is insensitive to the VGG16 architecture used in our classification experiments, we also used ResNet18 [23] in VAAL and the most competitive baseline (Core-set). Figure 6 in the appendix shows the choice of the architecture does not affect the performance gap between VAAL and Core-set.

5.3. Sampling time analysis

The sampling strategy of an active learner has to select samples in a time-efficient manner. In other words, it should be as close as possible to random sampling, considering the fact that random sampling is still an effective baseline. Table 1 shows our comparison for VAAL and all our baselines on CIFAR10 using a single NVIDIA TITAN Xp. Table 1 shows the time needed to sample a fixed budget of images from the unlabeled pool for all the methods. MC-Dropout performs multiple forward passes to measure the uncertainty from 10 dropout masks which explains why it appears to be very slow in sample selection. Core-set and Ensembles w. VarR, are the most competitive baselines to VAAL in terms of their achieved mean accuracy. However, in sampling time, VAAL takes 10.59 seconds while Core-set requires 75.33 sec and Ensembles w. VarR needs 20.48 sec. DBAL [16] is on-par in sampling time with VAAL, however, DBAL is outperformed in accuracy by all other methods including random sampling which can sample in only a few milliseconds. The significant difference between Core-set and VAAL is due to the fact that Core-set needs to solve an optimization problem for sample selection as opposed to VAAL which only needs to perform inference on the discriminator and rank its output probabilities. The Ensembles w. VarR method uses 5 models to measure the uncertainty resulting in better computational efficiency but it does not yet perform as fast as VAAL.

6. Conclusion

In this paper we proposed a new batch mode task-agnostic active learning algorithm, VAAL, that learns a latent representation on both labeled and unlabeled data in an adversarial game between a VAE and a discriminator, and implicitly learns the uncertainty for the samples deemed to be from the unlabeled pool. We demonstrate state-of-the-art results, both in terms of accuracy and sampling time, on small and large-scale image classification (CIFAR10, CIFAR100, Caltech-256, ImageNet) and segmentation datasets (Cityscapes, BDD100K). We further showed that VAAL is robust to noisy labels and biased initial labeled data, and it performs consistently well, given different oracle budgets.

References

[1] William H Beluch, Tim Genewein, Andreas Nürnberger, and Jan M Köhler. The power of ensembles for active learning in image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9368–9377, 2018.

[2] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.

[3] Klaus Brinker. Incorporating diversity in active learning with support vector machines. In *Proceedings of the 20th international conference on machine learning (ICML-03)*, pages 59–66, 2003.

[4] David A Cohn, Zoubin Ghahramani, and Michael I Jordan. Active learning with statistical models. *Journal of artificial intelligence research*, 4:129–145, 1996.

[5] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016.

[6] Jia Deng, Wei Dong, Richard Socher, Lie-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

[7] Yue Deng, KaWai Chen, Yilin Shen, and Hongxia Jin. Adversarial active learning for sequences labeling and generation. In *IJCAI*, pages 4012–4018, 2018.

[8] Suyog Dutt Jain and Kristen Grauman. Active image segmentation propagation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2864–2873, 2016.

[9] Sayna Ebrahimi, Mohamed Elhoseiny, Trevor Darrell, and Marcus Rohrbach. Uncertainty-guided continual learning with bayesian neural networks. *arXiv preprint arXiv:1906.02425*, 2019.

[10] Sayna Ebrahimi, Anna Rohrbach, and Trevor Darrell. Gradient-free policy architecture search and adaptation. *arXiv preprint arXiv:1710.05958*, 2017.

[11] Bradley Efron and Robert J Tibshirani. *An introduction to the bootstrap*. CRC press, 1994.

[12] Damien François. High-dimensional data analysis. In *From Optimal Metric to Feature Selection*, pages 54–55. VDM Verlag Saarbrucken, Germany, 2008.

[13] Linton C Freeman. *Elementary applied statistics: for students in behavioral science*. John Wiley & Sons, 1965.

[14] Yoav Freund, H Sebastian Seung, Eli Shamir, and Naftali Tishby. Selective sampling using the query by committee algorithm. *Machine learning*, 28(2-3):133–168, 1997.

[15] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059, 2016.

[16] Yarin Gal, Riashat Islam, and Zoubin Ghahramani. Deep bayesian active learning with image data. *arXiv preprint arXiv:1703.02910*, 2017.

[17] Ran Gilad-Bachrach, Amir Navot, and Naftali Tishby. Query by committee made real. In *Advances in neural information processing systems*, pages 443–450, 2006.

[18] Clark R Givens, Rae Michael Shortt, et al. A class of wasserstein metrics for probability distributions. *The Michigan Mathematical Journal*, 31(2):231–240, 1984.

[19] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256, 2010.

[20] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[21] Marc Gorriz, Axel Carlier, Emmanuel Faure, and Xavier Giro-i Nieto. Cost-effective active learning for melanoma segmentation. *arXiv preprint arXiv:1711.09168*, 2017.

[22] Gregory Griffin, Alex Holub, and Pietro Perona. Caltech-256 object category dataset. 2007.

[23] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[24] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei A Efros, and Trevor Darrell. Cycada: Cycle-consistent adversarial domain adaptation. *arXiv preprint arXiv:1711.03213*, 2017.

[25] Ashish Kapoor, Kristen Grauman, Raquel Urtasun, and Trevor Darrell. Active learning with gaussian processes for object categorization. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–8. IEEE, 2007.

[26] Hyunjik Kim and Andriy Mnih. Disentangling by factorising. *arXiv preprint arXiv:1802.05983*, 2018.

[27] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

[28] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[29] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

[30] Weicheng Kuo, Christian Häne, Esther Yuh, Pratik Mukherjee, and Jitendra Malik. Cost-sensitive active learning for intracranial hemorrhage detection. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 715–723. Springer, 2018.

[31] Xin Li and Yuhong Guo. Adaptive active learning for image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 859–866, 2013.

[32] David JC MacKay. Information-based objective functions for active data selection. *Neural computation*, 4(4):590–604, 1992.

[33] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

[34] Dwarikanath Mahapatra, Behzad Bozorgtabar, Jean-Philippe Thiran, and Mauricio Reyes. Efficient active learning for image classification and segmentation using a sample selection and conditional generative adversarial network. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 580–588. Springer, 2018.

[35] Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, Ian Goodfellow, and Brendan Frey. Adversarial autoencoders. *arXiv preprint arXiv:1511.05644*, 2015.

[36] Christoph Mayer and Radu Timofte. Adversarial sampling for active learning. *arXiv preprint arXiv:1808.06671*, 2018.

[37] Andrew Kachites McCallumzy and Kamal Nigamy. Employing em and pool-based active learning for text classification. In *Proc. International Conference on Machine Learning (ICML)*, pages 359–367. Citeseer, 1998.

[38] Prem Melville and Raymond J Mooney. Diverse ensembles for active learning. In *Proceedings of the twenty-first international conference on Machine learning*, page 74. ACM, 2004.

[39] Lars Mescheder, Sebastian Nowozin, and Andreas Geiger. Adversarial variational bayes: Unifying variational autoencoders and generative adversarial networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2391–2400. JMLR. org, 2017.

[40] Hieu T Nguyen and Arnold Smeulders. Active learning using pre-clustering. In *Proceedings of the twenty-first international conference on Machine learning*, page 79. ACM, 2004.

[41] Nicholas Roy and Andrew McCallum. Toward optimal active learning through monte carlo estimation of error reduction. *ICML, Williamstown*, pages 441–448, 2001.

[42] Edgar Schonfeld, Sayna Ebrahimi, Samarth Sinha, Trevor Darrell, and Zeynep Akata. Generalized zero-and few-shot learning via aligned variational autoencoders. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8247–8255, 2019.

[43] Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. In *International Conference on Learning Representations*, 2018.

[44] Burr Settles. Active learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 6(1):1–114, 2012.

[45] Burr Settles. Active learning literature survey. 2010. *Computer Sciences Technical Report*, 1648, 2014.

[46] Claude Elwood Shannon. A mathematical theory of communication. *Bell system technical journal*, 27(3):379–423, 1948.

[47] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[48] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. In *Advances in Neural Information Processing Systems*, pages 3483–3491, 2015.

[49] Ilya Tolstikhin, Olivier Bousquet, Sylvain Gelly, and Bernhard Schoelkopf. Wasserstein auto-encoders. *arXiv preprint arXiv:1711.01558*, 2017.

[50] Simon Tong and Daphne Koller. Support vector machine active learning with applications to text classification. *Journal of machine learning research*, 2(Nov):45–66, 2001.

[51] Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. Ensemble adversarial training: Attacks and defenses. *arXiv preprint arXiv:1705.07204*, 2017.

[52] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7167–7176, 2017.

[53] Keze Wang, Dongyu Zhang, Ya Li, Ruimao Zhang, and Liang Lin. Cost-effective active learning for deep image classification. *IEEE Transactions on Circuits and Systems for Video Technology*, 27(12):2591–2600, 2017.

[54] Zheng Wang and Jieping Ye. Querying discriminative and representative samples for batch mode active learning. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 9(3):17, 2015.

[55] Lin Yang, Yizhe Zhang, Jianxu Chen, Siyuan Zhang, and Danny Z Chen. Suggestive annotation: A deep active learning framework for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 399–407. Springer, 2017.

[56] Fisher Yu, Vladlen Koltun, and Thomas A Funkhouser. Dilated residual networks. In *CVPR*, volume 2, page 3, 2017.

[57] Fisher Yu, Wenqi Xian, Yingying Chen, Fangchen Liu, Mike Liao, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving video database with scalable annotation tooling. *arXiv preprint arXiv:1805.04687*, 2018.

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

SUPPLEMENTARY MATERIAL

A. Datasets

Table 2 shows a summary of the datasets utilized in our work along with their size and number of classes and budget size.

				Initially		
Dataset	#Classes	Train + Val	Test	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

Table 2. A summary of the datasets used in our experiments. CIFAR10, CIFAR100, Caltech-256 and ImageNet are datasets used for image classification, while BDD100K and Cityscapes are large scale segmentation datasets. The budget for each dataset is the number of images that can be sampled at each training iteration.

B. Hyperparameter Selection

Table 3 shows the hyperparameters found for our models through a grid search.

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

Table 3. Hyperparameters used in our experiments for VAAL. d is the latent space dimension of VAE. α_1, α_2, and α_3 are learning rates for VAE, discriminator (D), and task module (T), respectively. λ_1 and λ_2 are the regularization parameters for transductive and adversarial terms used in Eq. (4). β is the Lagrangian parameter in Eq. (1).

Figure 6 shows the performance of our method is robust to the choice of the architecture by having consistently better performance over Core-set [43] on CIFAR100.

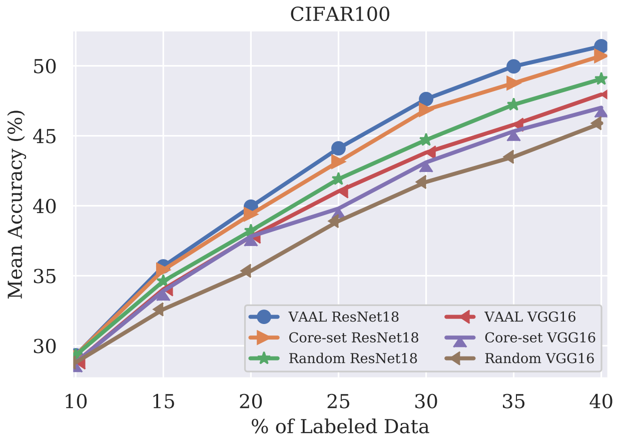

Figure 6. Performance of VAAL using ResNet18 and VGG16 on CIFAR100. Best visible in color. Data and code required to reproduce are provided in our code repository