Supplementary Material
Task-Aware Variational Adversarial Active Learning

Kwanyoung Kim[1,4], Dongwon Park[1], Kwang In Kim[2,3], Se Young Chun[1,2,5,†]
[1]Dept of EE, UNIST, [2]AIGS, UNIST, [3]Dept of CSE, UNIST, [4]Dept of Bio & Brain Eng, KAIST,
[5]Dept of ECE, INMC, Seoul National University, South Korea
cubeyoung@kaist.ac.kr, dong1@unist.ac.kr, kimki@unist.ac.kr, sychun@snu.ac.kr

In this supplemental document, we present:

• brief summaries of active data sampling strategy of our Task-Aware Variational Adversarial Active Learning (TA-VAAL), accompanied by the corresponding algorithm descriptions (Sec. 1);

• "independent" task learner performance results without attaching loss prediction module (LPM) to show that the attached LPM was not selected active sampling methods were the most important factor for performances (Sec. 2);

• performance comparisons of "learning loss" methods with the original loss and our modified ranking loss (called learning loss_v2 in the main paper) (Sec. 3);

• absolute average accuracies and mIOU of active learning results in the main paper. Additional results on SVHN and Fashion-MNIST datasets were also presented here. (Sec. 4);

• active learning results on CIFAR10 and CIFAR100 with a larger budget size that is consistent with the work of VAAL. (Sec. 5);

• active learning results on Caltech256 with a larger budget and using VGG16 network. (Sec. 6);

• details of hyperparameters for training in TA-VAAL (Sec. 7);

• example images that were selected at each stage of different active learning algorithms (Sec. 8);

1. Algorithm for active sampling strategy

Algorithm 1 details the sample selection strategy at each stage of the TA-VAAL training process. The goal of each sample selection strategy is collecting b number of samples from unlabeled data pool X_U to update the labeled pool. First, we predicted the ranking of loss from Ranker, denoted by $R(;\theta_R)$, for unlabeled data pool to obtain task-aware (model-uncertainty) information:

$$r_U \leftarrow R(x_U;\theta_R), \quad \forall x_U \in X_U;$$

This value will be normalized so that the absolute loss value will not be preserved while the relative ranking information will be preserved before embedded into the latent space of VAE. Second, we got the latent space values from the encoder of VAE, denoted by $q_\theta(\cdot)$, for unlabeled pool:

$$z_U \leftarrow q_\theta(z_U|x_U), \quad \forall x_U \in X_U;$$

Finally, we selected the data points $(x_1^*, .., x_b^*)$ by the following operation:

$$(x_1^*, ..., x_b^*) = \underset{(x_1,...,x_b)\subset X_U}{\operatorname{argmin}} D(R(x_U), q_\theta(z_U|x_U));$$

Note that the smaller the output of discriminator D, the more likely its latent space belongs to unlabeled pool. The main idea of our approach is that rather than relying only on latent space that represents the probability from unlabeled pool, our proposed method utilized task-aware information from Ranker that represents the score that task learner predicts with low confidence to select *influential* and *difficult* data points.

Algorithm 1: Active sampling in TA-VAAL

Input : budget size b and unlabeled data pool X_U
Output : acquisition data samples $(x_1^*, .., x_b^*)$

Predict ranking of loss for unlabeled data pool:

$$r_U \leftarrow R(x_U;\theta_R), \quad \forall x_U \in X_U;$$

Get latent space values for unlabeled data pool:

$$z_U \leftarrow q_\theta(z_U|x_U), \quad \forall x_U \in X_U;$$

Choose the data points $(x_1^*, .., x_b^*)$ by the following operation:

$$(x_1^*, ..., x_b^*) = \underset{(x_1,...,x_b)\subset X_U}{\operatorname{argmin}} D(R(x_U), q_\theta(z_U|x_U));$$

†Corresponding author.

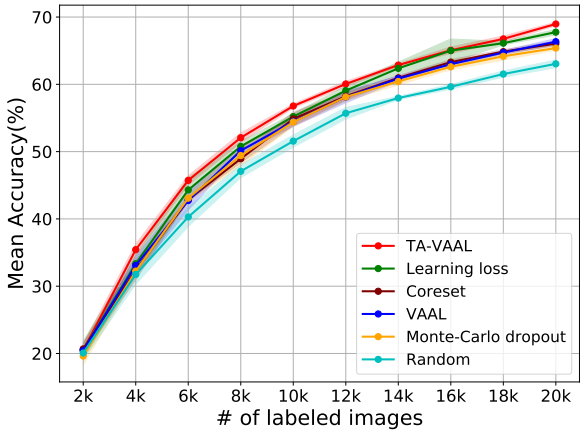


Figure 1: Active learning results with independent task learner on CIFAR100 dataset.

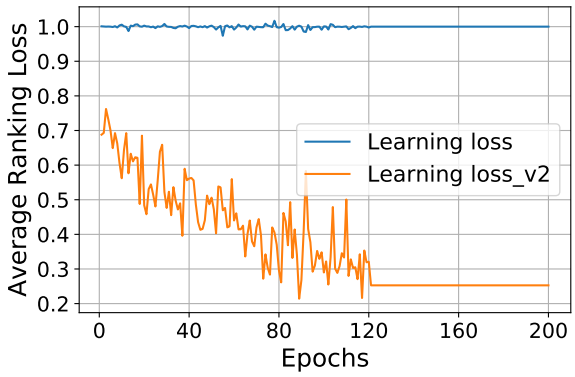


Figure 2: Losses for the loss prediction module over the number of epochs in learning loss and learning loss_$v2$.

2. Task learners without loss prediction modules

In [1], the task learner with loss prediction module was trained to show the performance on actively selected samples and was compared with other methods using the task learners without loss prediction modules. Since loss prediction modules are part of learning loss method and our proposed TA-VAAL method, it seems that these additional loss prediction modules may influence the overall performance as shown in Figures 3 and 4 of the main paper, often yielding slightly higher or lower performances. To measure the quality of the selected samples using active learning methods, we trained task learners of learning loss method and our proposed method again without attaching loss prediction module (or Ranker) on the selected samples up to 10k as shown in Figure 1.

Our proposed method was able to yield the best performance at all of stage on CIFAR100 as shown in Figure 5b of this material, but now our proposed method stil outperforms all other methods on CIFAR100 dataset, except for 16k stage, since the effect of loss prediction module for task learner training is minimized as illustrated in Figure 1. However, at the last stage (10k), our proposed method yielded slightly enhanced performance by 0.51% compared with the result of task learner with Ranker. Learning loss method also has slightly improved performance by 0.36%. Despite these changes, our proposed method still outperformed other state-of-the-art methods by achieving mean accuracy of 68.83 % in the last stage.

3. Loss comparisons for loss prediction module

Figure 2 shows the graph of average loss versus the number of epochs for learning loss methods with the original learning loss and our modified learning loss, called learning loss_v2. After the 120th epoch, we did not back propagate the gradient of Ranker so that the fluctuation of loss value is stopped as implemented in the original learning loss method. We observe that the loss for the loss prediction module of learning loss (blue line) is not minimized possibly due to fixed $\epsilon = 1$ to emphasize more on the exact loss predictions, less on the relative rankings of them. However, the ranking loss of our modified learning loss (orange line) can be minimized as iteration continues possibly due to the relaxed condition for predicting loss values so that training seems to be more stable than the original learning loss.

4. Absolute accuracies

In the main paper, we showed the improvements of accuracy from the random sampling baseline over the number of labeled dataset. In this section, we showed the absolute accuracy curves over the number of labeled dataset in Figures 5 and 6. To further validate the effectiveness of our proposed method, we performed additional experiments for image classification on SVHN [2] and Fashion-MNIST [3] datasets as illustrated in Figures 5c and 5d.

Dataset SVHN and Fashion-MNIST consist of 73,257 / 26,032 32×32 images, and 60,000 / 10,000 28×28 images for training / testing, respectively, all with 10 classes. We initially set labeled pool with randomly selected 1,000 images and the query size b was 1,000 at each stage. The other experiment setting is equal to the setting of the balanced image classification experiment in the main paper.

Implementation detail For training, data augmentation methods that were used for CIFAR10 dataset were also used on SVHN dataset. Only normalization was applied to Fashion MNIST. ResNet18 [4] was used for all task learners and stochastic gradient descent (SGD) optimizer was used with momentum 0.9 and weight decay 0.005. Learning rate was 0.1 for the first 160 epochs and then 0.01 for the rest

of 40 epochs. For the VAE, a modified Wasserstein auto-encoder [5] for taking ranking information was used and the discriminator was constructed as a 5-layer multi-layer perceptron (MLP). For both the VAE and the discriminator, the Adam optimizer [6] with learning rate 5 $\times 10^{-4}$ was used. Mini-batch size was 128 and the total epochs were 200 for all datasets.

Results for balanced benchmark datasets Six active learning methods were evaluated including random sampling, Monte-Carlo dropout [7], Core-set [8], Learning loss [1], VAAL [9] and our proposed TA-VAAL on two benchmark datasets, SVHN and Fashion MNIST. Figure 5 presents the number of labeled images versus the mean accuracy $\pm$ standard deviation from 5 trials.

In Figure 5c for SVHN, our proposed method outperformed other state-of-the-art methods in almost all stages except for one stage (5k). Note that our TA-VAAL yielded substantially higher mean accuracies than other methods at early stages such as 3k and 4k.

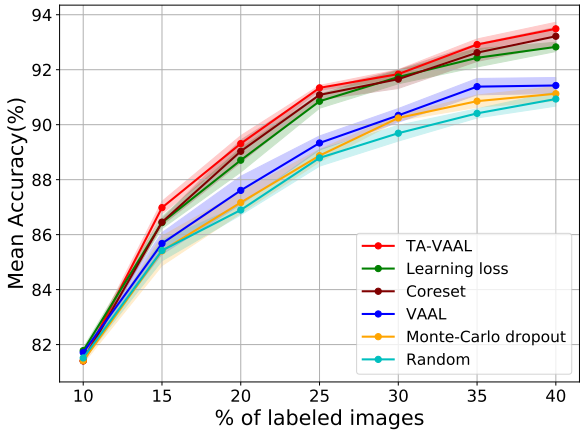
In Figure 5d for Fashion MNIST, our TA-VAAL outperformed all other compared methods over almost all stages except for the first stage and in particular yielded significantly improved performances at the 3k stage by more than 5% margin.

5. Another setting on initial size & budget

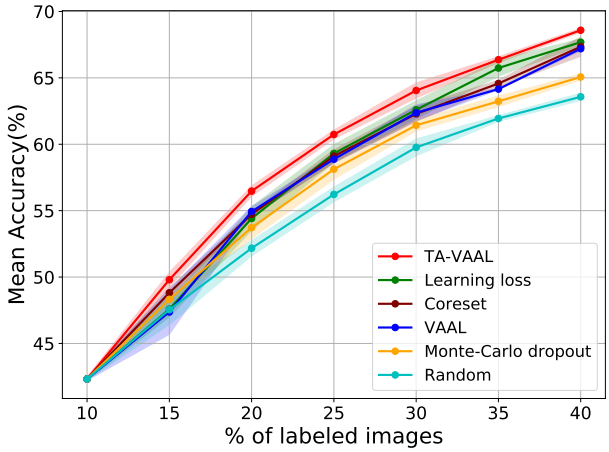
In the main paper, our experiments used smaller number of labels (1000 / 1000) or (2000 / 2000) that followed and extended the settings of the work of learning loss [1] than the reported VAAL experiments on CIFAR10, CIFAR100 (5000 / 2500) [9]. We argue that our setting with smaller initial size and budget would be more beneficial for active learning than larger initial size and budget. However, for those who are familiar with the work of VAAL [9], we briefly validate our implementations with the same active learning setting as VAAL for CIFAR10 and CIFAR100 (or 5000 / 2500 for initial / budget sizes) and the results are presented in Figure 3. Note that our proposed method still outperformed other state-of-the-art methods in all stages for the same setting in VAAL on both of dataset.

6. Another setting on a different network

In the main paper, we performed image classification tasks with a task learner using ResNet18 as also used in [1]. Here we performed another image classification task on Caltech256 with another task learner using VGG16 as also used in [9]. The results are illustrated in Figure 4, showing similar trends as other results in our paper. We employed SGD optimizer with hyperparameters as shown in Table 1. Note that there were a number of different setups for optimizers and hyperparameters among [9], its supplement material and its source codes.



(a) CIFAR10



(b) CIFAR100

Figure 3: Active learning results with a large budget size (5000 / 2500) on (a) CIFAR10 and (b) CIFAR100 dataset.

It seems that implementing the Learning loss method on VGG16 network is non-trivial. Our implementation for it is not optimized, but it still yielded slightly improved performance over random selections. Due to our sub-optimal Learning loss method with VGG16, our proposed TA-VAAL with it also yielded slightly improved performance over VAAL (except for the 4th stage). Thus, our proposed TA-VAAL method yielded expected performance by exploiting the Learning loss method [1] and the VAAL method [9] with another task learner (VGG16) on a relatively large dataset (Caltech256).

7. Detail on hyperparameters

Table 1 shows the hyperparameters for training our proposed method for different datasets. We set these hyperparameters based on VAAL settings and tuned these through a grid search.

Table 1: Hyperparameters used in TA-VAAL. d is the latent space dimension of VAE. ζ_1, ζ_2, and ζ_3 are learning rates of task learner T, VAE, and discriminator D, respectively. η is a scaling parameter for total loss of task learner in Eq. (2). λ are regularization parameters for transductive and adversarial losses of VAE. β is a Lagrangian parameter in Eq. (3). "Initial" represents that the size of labeled data pool at initial stage and "budget" indicates that the size of samples which to be selected at each stage. Zero padding was used. Large images were cropped considering the trade-off between training speed and GPU resources.

Dataset	d	ζ_1	ζ_2	ζ_3	η	λ	β	batch size	epochs	Inital / budget	image size
CIFAR10	64	$1\ 10^{-1}$	$5\ 10^{-4}$	$5\ 10^{-4}$	1	1	1	128	200	1000 / 1000	32 32 (padded)
CIFAR100	64	$1\ 10^{-1}$	$5\ 10^{-4}$	$5\ 10^{-4}$	1	1	1	128	200	2000 / 2000	32 32 (padded)
Caltech101	128	$1\ 10^{-2}$	$1\ 10^{-4}$	$1\ 10^{-4}$	0.2	15	1	16	200	1000 / 500	224 224 (cropped)
Caltech256	128	$5\ 10^{-4}$	$5\ 10^{-4}$	$5\ 10^{-4}$	0.1	15	1	64	100	3060 / 1530	224 224 (cropped)
Cityscapes	128	$1\ 10^{-3}$	$1\ 10^{-4}$	$1\ 10^{-4}$	0.1	25	1	4	100	200 / 200	412 412 (cropped)

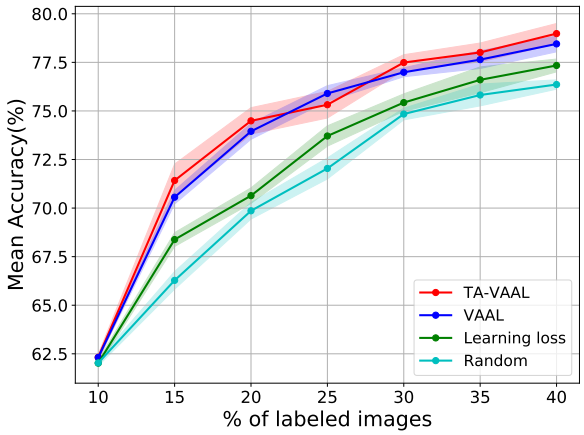


Figure 4: Active learning results with different task learner (VGG16) on Caletch256 dataset.

8. Example sampled images

Figure 7 shows some selected images at each stage of different active learning methods, corresponding to the result of Figure 3(a) in the main paper.

References

[1] Donggeun Yoo and In So Kweon. Learning loss for active learning. In *CVPR*, pages 93–102, 2019. 2, 3

[2] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, pages 1–9, 2011. 2

[3] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017. 2

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 2

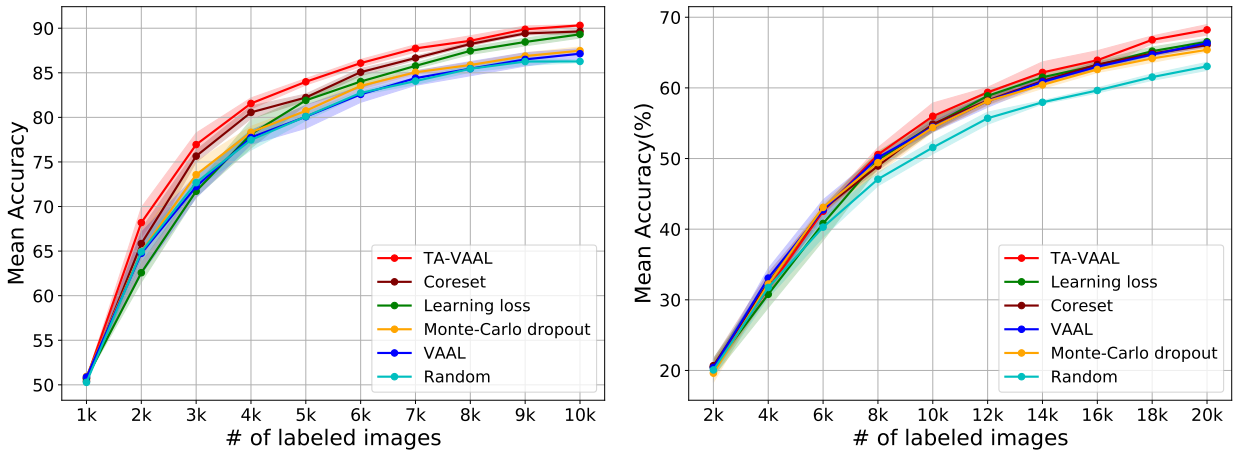
[5] Ilya Tolstikhin, Olivier Bousquet, Sylvain Gelly, and Bernhard Schoelkopf. Wasserstein auto-encoders. In *ICLR*, 2018. 3

[6] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 3

[7] Yarin Gal, Riashat Islam, and Zoubin Ghahramani. Deep bayesian active learning with image data. In *ICML*, pages 1183–1192, 2017. 3

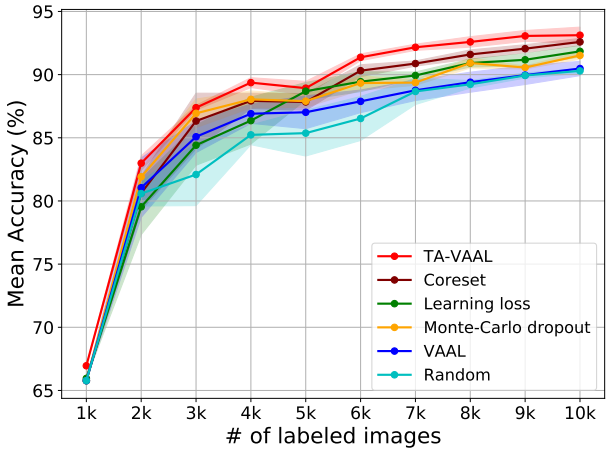
[8] Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. In *ICLR*, 2018. 3

[9] Samarth Sinha, Sayna Ebrahimi, and Trevor Darrell. Variational adversarial active learning. In *ICCV*, pages 5972–5981, 2019. 3

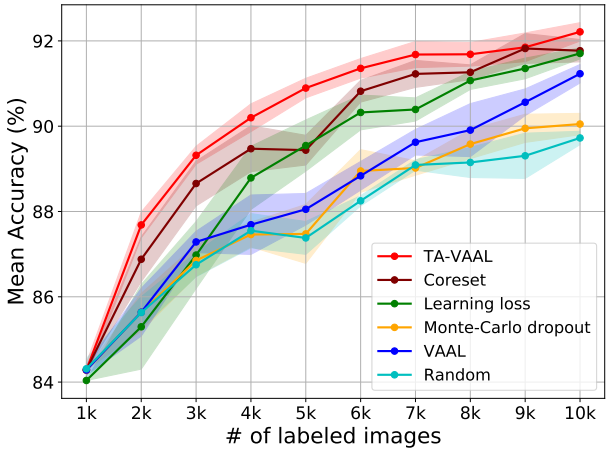


(a) CIFAR10

(b) CIFAR100

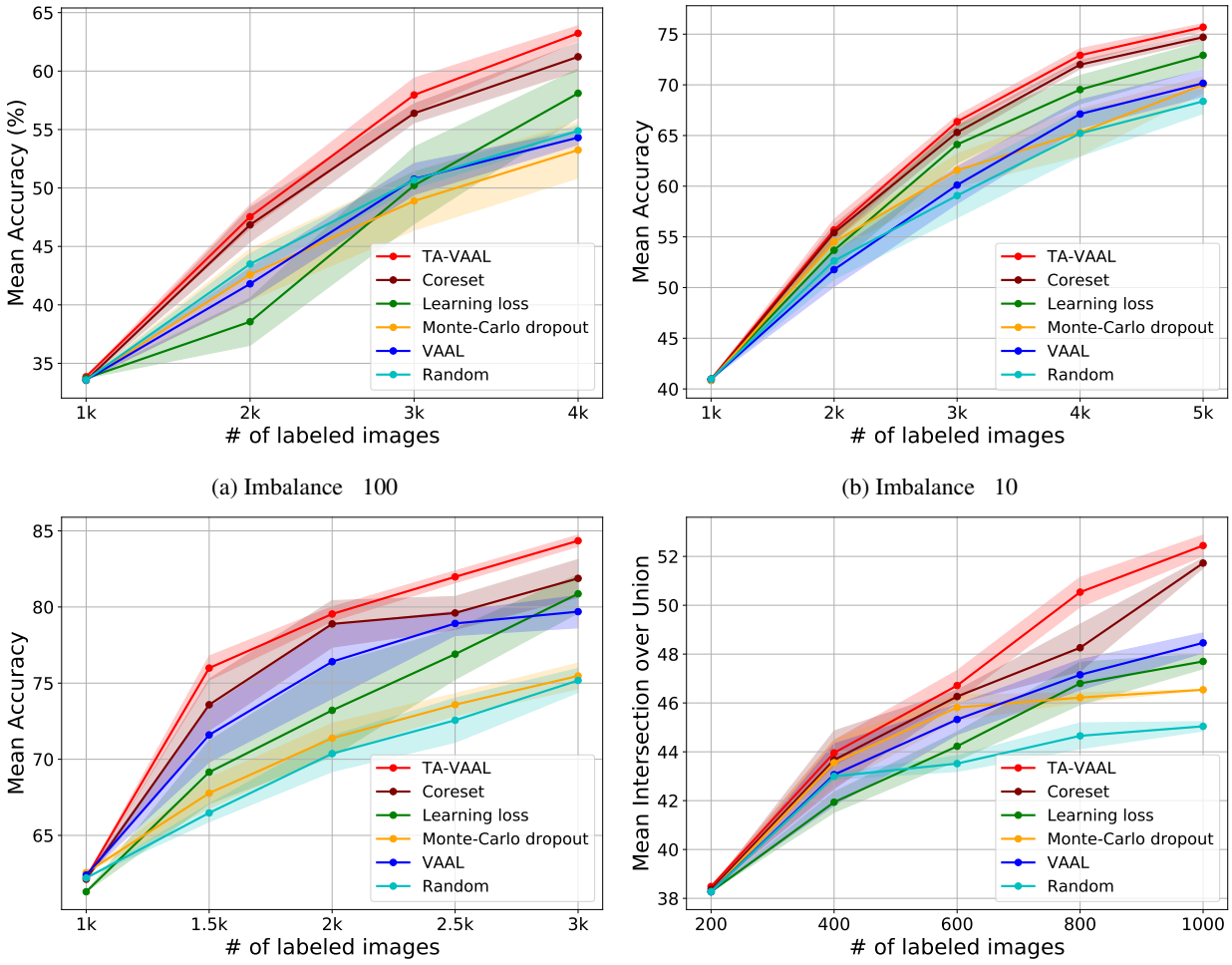


(c) SVHN



(d) Fashion-MNIST

Figure 5: Mean accuracy curves with standard deviation (shaded) of active learning methods over the number of labeled samples on (a) CIFAR10 and (b) CIFAR100.



(a) Imbalance 100

(b) Imbalance 10

(c) Caltech101

(d) Cityscape

Figure 6: Mean accuracy or Mean intersection over union (IOU) curves with standard deviation (shaded) of active learning methods over the number of labeled samples for image classifications on (a) modified CIFAR10 with imbalanced $\times 100$, (b) modified CIFAR10 with imbalanced $\times 10$, (c) Caltech101 datasets and for semantic segmentation on (d) Cityscape dataset.