TiDAL: Learning Training Dynamics for Active Learning

Anonymous authors
Paper under double-blind review

Abstract

Active learning (AL) aims to select the most useful data samples from an unlabeled data pool and annotate them to expand the labeled dataset under a limited budget. Especially, uncertainty-based methods choose the most uncertain samples, which are known to be effective in improving model performance. However, AL literature often overlooks *training dynamics* (TD), defined as the ever-changing model behavior during optimization via stochastic gradient descent, even though other areas of literature have empirically shown that TD provides important clues for measuring the sample uncertainty. In this paper, we propose a novel AL method, Training Dynamics for Active Learning (TiDAL), which leverages the TD to quantify uncertainties of unlabeled data. Since tracking the TD of all the large-scale unlabeled data is impractical, TiDAL utilizes an additional prediction module that learns the TD of labeled data. To further justify the design of TiDAL, we provide theoretical and empirical evidence to argue the usefulness of leveraging TD for AL. Experimental results show that our TiDAL achieves better or comparable performance on both balanced and imbalanced benchmark datasets compared to state-of-the-art AL methods, which estimate data uncertainty using only static information after model training.

1 Introduction

"There is a tide in the affairs of men. Which taken at the flood, leads on to fortune." — Shakespeare

Active learning (AL) (Atlas et al., 1990; Lewis & Gale, 1994) aims to solve the real-world problem of selecting the most useful data samples from large-scale unlabeled data pools and annotating them to expand labeled data under a limited budget. Since the current deep neural networks are often data-hungry, AL has increasingly gained attention in recent years. Existing AL methods can be divided into two mainstream categories: diversity-based and uncertainty-based methods. Diversity-based methods (Sener & Savarese, 2018; Gissin & Shalev-Shwartz, 2019) focus on constructing a subset that follows the target data distribution. Uncertainty-based methods (Gal et al., 2017; Beluch et al., 2018; Yoo & Kweon, 2019) choose the most uncertain samples, which are known to be effective in improving model performance. Hence, the most critical question for the latter becomes, *"How can we quantify the data uncertainty?"*

In this paper, we leverage ***training dynamics*** (TD) to quantify data uncertainty. TD is defined as the ever-changing model behavior on each data sample during optimization via stochastic gradient descent. Recent studies (Chang et al., 2017; Samuli & Timo, 2017; Toneva et al., 2018; Swayamdipta et al., 2020) have provided empirical evidence that TD provides important clues for measuring the contribution of each data sample to model performance improvement. Inspired by these studies, we hypothesize that the data uncertainty of unlabeled data can be estimated with TD. However, most uncertainty-based methods quantify data uncertainty based on static information (*e.g.*, loss (Yoo & Kweon, 2019) or predicted probability (Sinha et al., 2019)) from a fully-trained model "*snapshot*," neglecting the valuable information generated during training.

Despite its huge potential, TD is not yet actively explored in the domain of AL due to the following two critical challenges: (1) AL assumes a massive unlabeled data pool, thus tracking their TD is infeasible. Previous studies track TD only for the training data every epoch as it can be recorded easily during model optimization. On the other hand, AL targets a large number of unlabeled data,

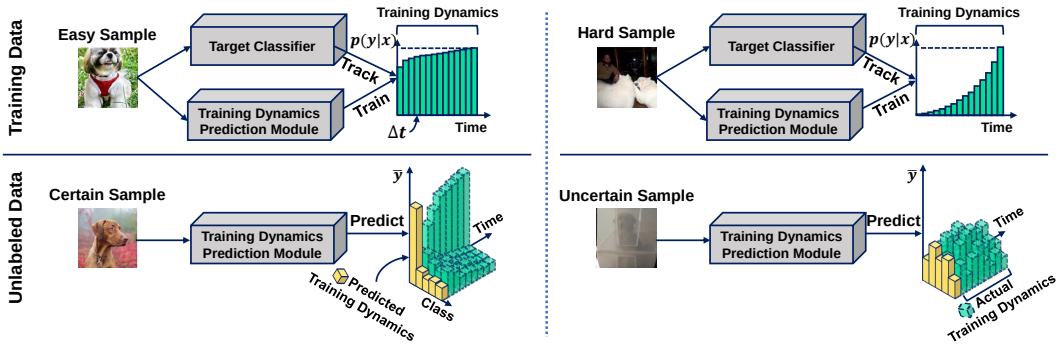


Figure 1: Our proposed TiDAL. TD of training samples x may differ even if they converge to the same final predicted probability $p(y|x)$ (Upper row). Hence, we are motivated to utilize the readily available rich information generated during training, i.e., leveraging TD. We estimate TD of large-scale unlabeled data using a prediction module instead of tracking the actual TD of all the unlabeled samples to avoid the computational overhead (Lower row).

where tracking their TD requires an impractical amount of computation (*e.g.*, inference all the unlabeled samples every training epoch). (2) Some of the existing TD methods (Pleiss et al., 2020; Swayamdipta et al., 2020; Park & Caragea, 2022) require true labels for measuring the contribution of each data sample (*e.g.*, easy-to-learn or hard-to-learn). However, AL methods need to infer the uncertainty of each sample without the true labels to select samples worth labeling.

Therefore, we propose TiDAL (***T**rai**D**ynamics for **A**ctive **L**earning*), a novel AL method that efficiently estimates the uncertainty of unlabeled data by leveraging their TD. We avoid tracking the TD of large-scale unlabeled data every epoch by predicting the TD with a TD prediction module. The module is trained to learn the TD of labeled data, which is readily available during model optimization. During the data selection phase, we predict the TD of unlabeled data with the trained module to quantify their uncertainties. The module efficiently obtains TD, which avoids inferring all the unlabeled samples every epoch. Furthermore, we quantify uncertainties of unlabeled samples by carefully incorporating TD into common uncertainty estimators, such as entropy and margin, without using the true labels. The design of TD prediction module is influenced by several previous methods that use additional modules to predict target model outputs and their statistics (*e.g.*, loss prediction (Yoo & Kweon, 2019) or confidence prediction (Corbière et al., 2019)). The major difference is that TiDAL leverages TD, whereas the others rely only on the model snapshot captured after the model is fully trained. Our motivation and proposed method are illustrated in Figure 1.

We further support the above method by providing theoretical and empirical evidence that TD is more effective in separating uncertain and certain data than static information from a model snapshot captured after fully-trained. Moreover, experimental results demonstrate that our TiDAL achieves better or comparable performance to existing AL methods on both balanced and imbalanced datasets. Additional analyses show that our prediction module successfully predicts TD, and the predicted TD is useful in estimating uncertainties of unlabeled data.

Contributions of our study: (1) We bridge the concept of training dynamics and active learning with the theoretical and experimental evidence that training dynamics is effective in estimating data uncertainty. (2) We propose a new method that estimates uncertainties of unlabeled data by leveraging their training dynamics which are efficiently predicted by the prediction module. (3) Our proposed method achieves better or comparable performance on both balanced and imbalanced benchmark datasets compared to existing active learning methods.

2 METHOD

In this section, we describe our novel AL method, TiDAL. After summarizing the preliminaries, we define TD with the corresponding TD-aware uncertainty estimators. Then, we provide a motivating observation and theoretical justification on using TD for AL. Finally, we describe the TD prediction module that efficiently produces the estimated TD of unlabeled data, which is jointly trained with the target classifier.

2.1 PRELIMINARIES

Uncertainty-based active learning. In this work, we focus on uncertainty-based AL for multi-class classification problems. Let $\boldsymbol{p} = [p(1|x),\ p(2|x),\ \cdots,\ p(C|x)]^T \in \mathbb{R}^C$ as the predicted probabilities of the given sample x for C classes by the classifier f, where we denote the true label of x as y. $\mathcal{D}$ and $\mathcal{D}_u$ denote a labeled dataset and an unlabeled data pool, respectively. The general cycle of uncertainty-based AL is in two steps: (1) train the target classifier f on the labeled dataset $\mathcal{D}$ and (2) select top-k uncertain data samples from the unlabeled data pool $\mathcal{D}_u$. Selected samples are then given to the human annotators to expand the labeled dataset $\mathcal{D}$, cycling back to the first step.

Data uncertainty. Even though there are several ways to quantify data uncertainty, we adopt the two most common and straightforward: *entropy* (Shannon, 1948) and *margin* (Roth & Small, 2006). We employ both entropy and margin, with and without taking TD into account, to demonstrate the effectiveness of utilizing TD information generated during training. We first define the TD-free estimators in this section, and introduce their TD-aware variants in §2.2.

TD-free entropy H is defined as follows:

$$H(\boldsymbol{p}) = -\sum\nolimits_{c=1}^{C} p(c|x) \log p(c|x), \tag{1}$$

where the sample x is from the unlabeled data pool $\mathcal{D}_u$. Entropy only concentrates on the level of the model's confidence on the given sample x and gets bigger when the prediction across the classes becomes uniform (*i.e.*, uncertain).

TD-free margin M measures the difference between the true and the maximum probability. However, $\hat{y}$ is used as a substitute for the true label y, which is not accessible for the unlabeled samples:

$$M(\boldsymbol{p}) = p(\hat{y}|x) - \max_{c \neq \hat{y}} p(c|x), \tag{2}$$

where $\hat{y}$ denote the predicted label by f, defined as $\hat{y} = \arg\max_c p(c|x)$. Both entropy and margin are computed with the predicted probabilities $\boldsymbol{p}$ of the fully trained classifier f, only taking the snapshot of f into account.

2.2 DEFINITION OF TRAINING DYNAMICS

Our TiDAL targets to leverage TD of unlabeled data to estimate their uncertainties. TD can be defined as any model behavior during optimization. For example, Pleiss et al. (2020) utilize the area under the margin between logit values of the target class and the other largest class, and Swayamdipta et al. (2020) utilize the variance of the predicted probabilities generated at each epoch. In this work, we define the TD $\bar{\boldsymbol{p}}^{(t)}$ as the area under the predicted probabilities of each data sample x obtained during the t time steps of optimizing the target classifier f as follows:

$$\bar{\boldsymbol{p}}^{(t)} = [\bar{p}^{(t)}(1|x),\ \bar{p}^{(t)}(2|x),\ \cdots,\ \bar{p}^{(t)}(C|x)]^T = \sum\nolimits_{\tau} \boldsymbol{p}^{(\tau)} \Delta\tau \simeq \sum\nolimits_{i=1}^{t} \boldsymbol{p}^{(i)}/t, \tag{3}$$

where $\boldsymbol{p}^{(i)} = [p^{(i)}(1|x),\ p^{(i)}(2|x),\ \cdots,\ p^{(i)}(C|x)]^T$ is the predicted probabilities of a target classifier f at the i-th time step. $\Delta\tau$ is the unit time step to normalize the predicted probabilities. For simplicity, we record $\boldsymbol{p}^{(i)}$ every epoch and choose $\Delta\tau = 1/t$, namely, averaging the predicted probabilities during t epochs (Swayamdipta et al., 2020; Song et al., 2019). The TD $\bar{\boldsymbol{p}}^{(t)}$ takes all the predicted probabilities during model optimization into account; hence it encapsulates the overall tendency of the model during t epochs of optimization, avoiding being biased towards a snapshot of $\boldsymbol{p}^{(t)}$ in the final epoch t.

Training Dynamics-Aware Uncertainty Estimation. We believe that data uncertainty could be captured from TD, and it is effective in distinguishing uncertain samples from certain samples. To this end, we introduce two uncertainty estimation strategies to quantify data uncertainty with TD that do not use the true labels. Our strategies are simple variants of entropy and margin (§2.1), replacing the predictions $\boldsymbol{p}$ of the trained target classifier f with TD $\bar{\boldsymbol{p}}$ of Equation 3.

TD-aware entropy $\bar{H}$ is defined by swapping $\boldsymbol{p}$ with $\bar{\boldsymbol{p}}$:

$$\bar{H}(\bar{\boldsymbol{p}}) = -\sum\nolimits_{c=1}^{C} \bar{p}(c|x) \log \bar{p}(c|x). \tag{4}$$

Entropy $\bar{H}$ is maximized when $\bar{\boldsymbol{p}}$ is uniform, *i.e.*, the sample is uncertain for the target classifier.

TD-aware margin $\bar{M}$ is also similarly defined:

$$\bar{M}(\bar{\boldsymbol{p}}) = \bar{p}(\hat{y}|x) - \max_{c \neq \hat{y}} \bar{p}(c|x). \tag{5}$$

The smaller the margin, the more uncertain the sample becomes. There are several possible variants of $\bar{M}$ depending on the definition of $\hat{y}$. We conduct experiments to compare $\bar{M}$ with its variants. The experimental details and results are provided in Appendix C.4.

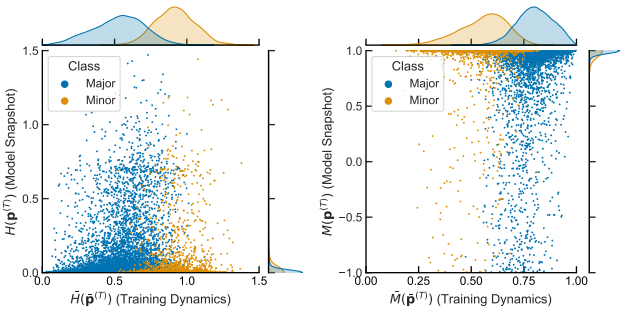
2.3 Is Training Dynamics Useful for Quantifying Uncertainty?

In this section, we provide empirical and theoretical evidence to support our claim: TD is more effective in separating uncertain data from certain data than static information, where we define the latter as the model snapshot captured at the end of model training.

2.3.1 Motivating Observation

Settings. We emphasize that it is nontrivial to directly measure sample-wise difficulty, inhibiting the quantitative analysis of data uncertainty. To avoid this, we borrow the theoretical and empirical results of long-tailed visual recognition (Liu et al., 2019; Cao et al., 2019; Hong et al., 2021): it is hard for the deep neural network-based model to train with fewer samples. Hence, we regard major and minor class samples to contain many certain and uncertain samples for the model, respectively. We train the target classifier f on the long-tailed dataset during T epochs to observe the TD-free and TD-aware entropy (H, $\bar{H}$) and margin (M, $\bar{M}$) scores of the training data. More details and discussions are described in Appendix B.

Results. Figure 2 shows the distribution of TD-aware (*x-axis*) and TD-free (*y-axis*) scores. We can observe that TD-aware entropy and margin scores ($\bar{H}, \bar{M}$) successfully separate the major and the minor class samples, whereas TD-free scores (H, M) fail to do so. We conclude that compared to model snapshots, TD is more helpful in separating uncertain samples from certain samples.



(a) Entropy Distribution (b) Margin Distribution

Figure 2: Score distribution after long-tailed training.

2.3.2 Theoretical Evidence

Theorem 1. *(Informal) Under the LE-SDE framework (Zhang et al., 2021b), with the assumption of sample-level local elasticity (He & Su, 2019), certain samples and uncertain samples reveal different TD; especially, certain samples converge quickly than uncertain samples.*

The above theorem discusses different model behaviors depending on the easiness of the sample. We assume that, compared to the uncertain sample, the certain sample has the same class samples nearby, following the level set estimation (Jiang et al., 2018a) and nearest neighbor (Papernot & McDaniel, 2018) literature. We suspect that, due to the local elasticity of deep nets, samples close by have a bigger impact on the certain sample, hence changing its predicted probability more rapidly. As the certain sample is quicker to converge, its TD is larger than that of the hard sample.

Theorem 2. *(Informal) TD-aware estimators such as Entropy (Equation 4) and Margin (Equation 5) successfully capture the difference of TD between easy and hard samples even for the case where it cannot be distinguished via the predicted probabilities of the model snapshot.*

The above theorem discusses the validity of the TD-aware estimators on whether they can successfully differentiate between two samples of different TD but with the same final prediction. With Theorem 1, one can conclude that the TD-aware estimators are effective in capturing the sample uncertainty. Due to the space constraints, we provide the details of the above results in Appendix A.

2.4 TRAINING DYNAMICS PREDICTION MODULE

As described in §1, it is not computationally feasible to track TD for the large-scale unlabeled data as it requires model inference on all the unlabeled data every training epoch. Thus, we use the TD prediction module m to efficiently predict the TD of unlabeled data at the t-th epoch. The TD prediction module produces the C-dimensional predictions $\tilde{\boldsymbol{p}}_m^{(t)} = [\tilde{p}_m^{(t)}(1|x),\ \tilde{p}_m^{(t)}(2|x),\ \cdots,\ \tilde{p}_m^{(t)}(C|x)]^T \in [0,1]^C$ estimating the actual TD $\bar{\boldsymbol{p}}^{(t)}$ of the given sample x in Equation 3. At the data selection phase, we use the predicted TD $\tilde{\boldsymbol{p}}_m^{(T)}$ instead of the actual TD $\bar{\boldsymbol{p}}^{(T)}$ in Equation 4 & 5 to estimate the TD-aware uncertainty of the unlabeled sample x at the final epoch T.

One can offer several ways to design the module m, but we adopt the architecture of the loss prediction module (Yoo & Kweon, 2019) except for the last layer. We use fully-connected layer with softmax activation to output the TD predictions $\tilde{\boldsymbol{p}}_m^{(t)}$. Similar to the loss prediction module, we extract several hidden feature maps of the target classifier f to feed the TD prediction module m. Refer to Yoo & Kweon (2019) for architecture details.

2.5 TRAINING OBJECTIVES

To train the target classifier f at the t-th epoch, we use the cross-entropy loss function $\mathcal{L}_{\text{target}}$ on the predicted probability $\boldsymbol{p}^{(t)}$ and a one-hot encoded vector $\boldsymbol{y} \in \{0,1\}^C$ of the true label y:

$$\mathcal{L}_{\text{target}} = \mathcal{L}_{\text{CE}}(\boldsymbol{p}^{(t)}, \boldsymbol{y}) = -\log p^{(t)}(y|x). \tag{6}$$

Meanwhile, the TD prediction module m learns the TD of a given sample x by minimizing the Kullback–Leibler (KL) divergence between the predicted TD $\tilde{\boldsymbol{p}}_m^{(t)}$ and the actual TD $\bar{\boldsymbol{p}}^{(t)}$:

$$\mathcal{L}_{\text{module}} = \mathcal{L}_{\text{KL}}(\bar{\boldsymbol{p}}^{(t)} || \tilde{\boldsymbol{p}}_m^{(t)}) = \sum\nolimits_{c=1}^{C} \bar{p}^{(t)}(c|x) \log\left(\bar{p}^{(t)}(c|x) \Big/ \tilde{p}_m^{(t)}(c|x)\right). \tag{7}$$

The final objective function of our proposed method is defined as follows:

$$\mathcal{L} = \mathcal{L}_{\text{target}} + \lambda \mathcal{L}_{\text{module}} \tag{8}$$

where λ is a balancing factor to control the effect of $\mathcal{L}_{\text{module}}$ during model training.

3 EXPERIMENTS

3.1 EXPERIMENTAL SETUP

Datasets. To assess the performance of our proposed method and baseline methods, we conduct experiments on the following five datasets: CIFAR10/100 (Krizhevsky et al., 2009), FashionMNIST (Xiao et al., 2017), SVHN (Netzer et al., 2011), and iNaturalist2018 (Van Horn et al., 2018). Since CIFAR and FashionMNIST are both balanced, we further modify them to simulate the data imbalance in the real world, following the previous long-tail visual recognition studies (Cao et al., 2019; Liu et al., 2019; Zhou et al., 2020a; Hong et al., 2021). The imbalance ratio is defined as $N_{\text{max}}/N_{\text{min}}$ where N is the number of samples in each class. We make two variants with data imbalance ratios 10 and 100 for each dataset. Unlike the above, SVHN and iNaturalist18 are already imbalanced. Especially, iNaturalist2018 is commonly chosen to demonstrate how methods work in imbalanced real-world settings. The dataset statistics are summarized in Appendix C.

Baselines. For a fair comparison, we compare our TiDAL with the following baselines which train a target classifier with only labeled data. **Random sampling**: a simple baseline that randomly selects data samples from the unlabeled dataset. **Entropy sampling** (Shannon, 1948): an uncertainty-based method that selects data samples based on the maximum entropy. **BALD** (Gal et al., 2017): an uncertainty-based method that selects data samples based on the mutual information between the model prediction and the posterior. **CoreSet** (Sener & Savarese, 2018): a diversity-based method that selects representative data samples covering all data through a minimum radius. **LLoss** (Yoo & Kweon, 2019): an uncertainty-based method that learns to estimate the errors of the predictions (loss) made by the learner and select data samples based on the predicted loss. **CAL** (Zhang & Plank, 2021): recent work on using TD, gathering samplewise TD information on whether the classifier was consistently correct or not during training. CAL splits the samples into two classes by applying a

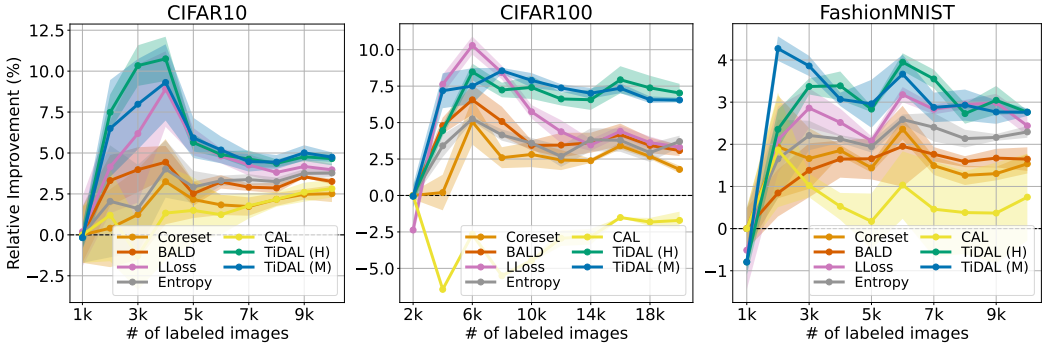


Figure 3: Averaged relative accuracy improvement curves and their 95% confidence interval (shaded) of AL methods over the number of labeled samples on balanced datasets. TiDAL ($\bar{H}$) and TiDAL ($\bar{M}$) denote the performance of TiDAL when with TD-aware entropy $\bar{H}$ and margin $\bar{M}$ as the data uncertainty estimation strategy, respectively.

heuristic threshold to the TD information to train a binary classifier that outputs uncertainty score. To verify the effectiveness of TiDAL, we further compare it with the two semi-supervised AL methods, **VAAL** (Sinha et al., 2019) and **TA-VAAL** (Kim et al., 2021) in §C.3. Note that these methods further utilize unlabeled data for training the selection module, thus it is a rather unfair comparison for our TiDAL.

Active learning setting. We follow the same setting from Beluch et al. (2018); Yoo & Kweon (2019) for the detailed AL settings. For the initial step, we randomly select initial samples to be annotated from the unlabeled dataset, where we use them to train the initial target classifier. Then, we obtain a random subset from the unlabeled data pool $\mathcal{D}_u$ to choose the top-k samples based on the criterion of each method, where those samples will be annotated. We repeat the above cycle, training a classifier from scratch from the continuously expanding labeled set.

Implementation details. For a fair comparison, we use the same backbone network ResNet-18 (He et al., 2016) except for iNaturalist2018, where we use ResNet-50 (He et al., 2016) pretrained on ImageNet (Deng et al., 2009). All models are trained with SGD optimizer with momentum 0.9, weight decay $5 \cdot 10^{-4}$, and learning rate (LR) decay of 0.1. For CIFAR10/100 and SVHN, we train the model for 200 epochs with an initial LR of 0.1 and decay at epoch 160. For FashionMNIST, 100 epochs with an initial LR of 0.1 and decay at epoch 80. For iNaturalist2018, 50 epochs with an initial LR of 0.01 and decay at epoch 40. For CIFAR10/100, SVHN and FashionMNIST, we set the batch size and the unlabeled subset size to be 128 and 10^4, respectively. For iNaturalist2018, which is much larger than other datasets, we set the batch size and the unlabeled subset size to 256 and 10^6, respectively. To compare with other state-of-the-art baselines, we show the average accuracy and 95% confidence interval with three trials.

3.2 Results on Balanced Datasets

We first evaluate our TiDAL against the state-of-the-art methods on various balanced datasets: CIFAR10, CIFAR100, and FashionMNIST. Figure 3 and 10 shows the performance improvement relative to that of Random sampling, which is the most naive baseline. For CIFAR10, the two variants of TiDAL outperform all the baselines at all AL cycles. Similarly, TiDAL shows better performance than the baselines except for LLoss, which shows better improvement at only around 6k than TiDAL ($\bar{M}$) on CIFAR100. CAL that uses training dynamics underperforms other baselines. It seems that CAL is very sensitive for threshold according to dataset. Nonetheless, our TiDAL achieves better final performance than all the baselines, including LLoss and CAL. For FashionMNIST, TiDAL also shows a large performance gap compared to other baselines up to 7k labeled samples.

3.3 Results on Imbalanced Datasets

Synthetically imbalanced datasets. Similar to the above, Figure 4, 9, and 11 shows the performance improvements on the synthetically imbalanced datasets with the two imbalance ratios, 10

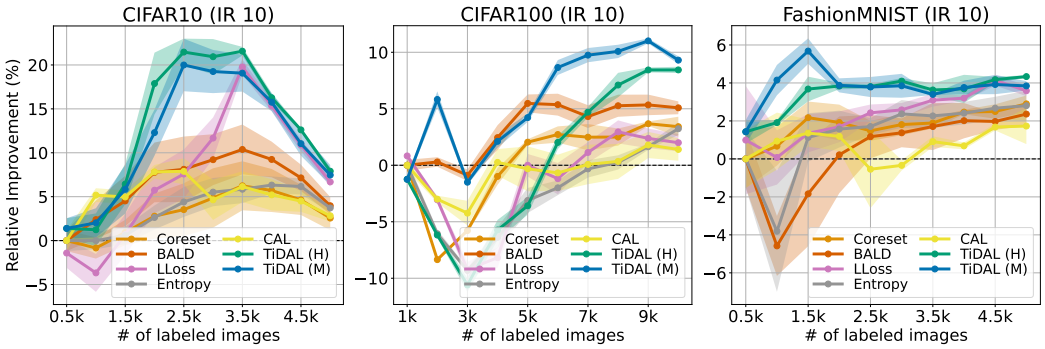


Figure 4: Averaged relative accuracy improvement curves and their 95% confidence interval (shaded) of AL methods over the number of labeled samples on synthetically imbalanced datasets. We use the imbalance ratio (IR) of 10 on CIFAR10, CIFAR100, and FashionMNIST. For the results on the IR of 100, consult the Appendix.

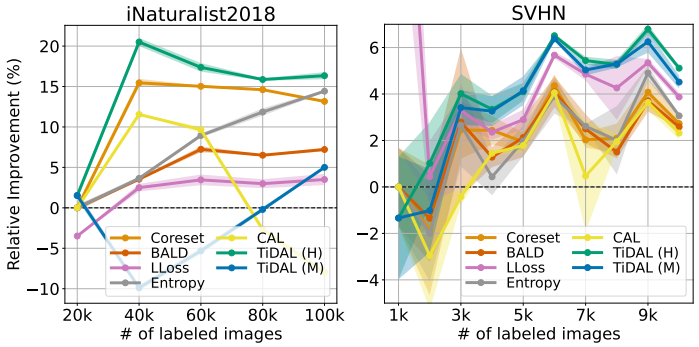


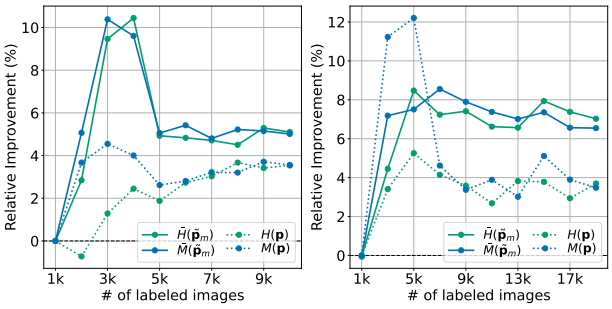
Figure 5: Averaged relative accuracy improvement curves and its 95% confidence interval (shaded) of AL methods over the number of labeled samples on real-world imbalanced datasets: iNaturalist2018 and SVHN. For SVHN, LLoss shows a substantial improvement of 20.02% $\pm$ 6.77% at the initial phase (1k), but we clip the plot to show the performance afterward more clearly.

and 100. Except for the CIFAR10 with the imbalance ratio of 100, our methods show superb performance across all the imbalanced settings. TiDAL performs especially well with a small variance in imbalanced CIFAR100, where the number of classes is the largest. In imbalanced FashionMNIST, the performance quickly rises till 2.5k labeled images and then saturates. This implies that FashionMNIST is easier than other datasets, and needs to focus more on the early steps of training to compare with other models. TiDAL also shows overall better performance on FashionMNIST, especially in early steps.

Real-world imbalanced datasets. Figure 5 and 10 shows evaluation results on real-world imbalanced datasets. For iNaturalist2018, which is the large-scale long-tailed classification dataset, TiDAL shows outstanding performance compared to other methods. For SVHN, TiDAL shows the best improvements with low variance as the number of labeled images increases except for the initial stage. LLoss shows outstanding performance only in the initial stage, where we presume that the loss prediction module of LLoss acts as a regularizer during model optimization.

3.4 Analysis on TD Prediction Module

Effectiveness of TD prediction module. In order to see the effectiveness of using the predicted TD $\tilde{\boldsymbol{p}}_m$, we conduct an ablation test that compares the performance between when using and not using the TD prediction module m. Figure 6 show the results on balanced CIFAR10 and CIFAR100. We observe that $\bar{H}(\tilde{\boldsymbol{p}}_m)$ and $\bar{M}(\tilde{\boldsymbol{p}}_m)$ using the predicted TD $\tilde{\boldsymbol{p}}_m$ to estimate the data uncertainty significantly outperform the methods $H(\boldsymbol{p})$ and $M(\boldsymbol{p})$ that use only the final predicted probabilities $\boldsymbol{p}$ of the target classifier f, showing better performance in the whole training cycle. Even $M(\boldsymbol{p})$



(a) CIFAR10 (b) CIFAR100

Figure 6: Ablation test results. $\bar{H}(\tilde{\boldsymbol{p}}_m)$ and $\bar{M}(\tilde{\boldsymbol{p}}_m)$ use the predicted TD $\tilde{\boldsymbol{p}}_m$ of the prediction module m. In contrast, $H(\boldsymbol{p})$ and $M(\boldsymbol{p})$ use the predicted probability of the model snapshot $\boldsymbol{p}$.

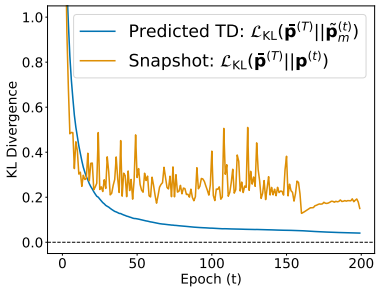


Figure 7: KL divergence scores of the actual TD $\bar{\boldsymbol{p}}^{(T)}$ with the predicted TD $\tilde{\boldsymbol{p}}_m^{(t)}$ and the predicted probability of the model snapshot $\boldsymbol{p}^{(t)}$, respectively, during model optimization.

shows temporary improvement in earlier steps on CIFAR100, $\bar{H}(\tilde{\boldsymbol{p}}_m)$ and $\bar{M}(\tilde{\boldsymbol{p}}_m)$ maintain stable improvement, eventually winning over $M(\boldsymbol{p})$. This indicates that the predicted TD $\tilde{\boldsymbol{p}}_m$ of the TD prediction module m produces better data uncertainty estimation than the predicted probability $\boldsymbol{p}$ of the target classifier f.

Predictive performance of the TD prediction module. We verify whether the TD prediction module m accurately predicts the actual TD $\bar{\boldsymbol{p}}$. Its prediction performance is crucial as we use the predicted TD $\tilde{\boldsymbol{p}}_m$ of the module m to quantify uncertainties of unlabeled data. Using the KL divergence $\mathcal{L}_{\mathrm{KL}}$, we analyze that the predicted TD $\tilde{\boldsymbol{p}}_m$ converges to the actual TD $\bar{\boldsymbol{p}}$ at the data selection phase. We calculate $\mathcal{L}_{\mathrm{KL}}(\bar{\boldsymbol{p}}^{(T)}||\tilde{\boldsymbol{p}}_m^{(t)})$ and compare it with $\mathcal{L}_{\mathrm{KL}}(\bar{\boldsymbol{p}}^{(T)}||\boldsymbol{p}^{(t)})$ which is set as a baseline computed with the actual TD $\bar{\boldsymbol{p}}$ and the predicted probabilities $\boldsymbol{p}$ (snapshot) of the target classifier f. In this analysis, we use the balanced CIFAR10 where the sample-wise averaged KL divergence scores are computed on the test set. Figure 7 shows that the final predicted TD successfully approximates the actual TD, while the predicted probability is highly different from the actual TD. We conclude that the TD prediction module m can produce the TD efficiently, leading to performance improvement, and the predicted TD acts as a better approximation of the actual TD than the predicted probability of a model snapshot captured at each epoch.

3.5 Limitation

We found two potential limitations of our TiDAL derived from the fact that it relies on the outputs of the target classifier to compute the TD. First, TiDAL is designed only for classification tasks, and thus it cannot be applied to AL targeting other tasks, such as regression (Cohn et al., 1994; Gong et al., 2022). Second, TiDAL is highly influenced by the performance of the target classifier, especially when the target classifier wrongly classifies the hard negative samples with a high confidence during model optimization. These samples can be treated as certain samples (i.e. will not be selected for annotation) because they have low estimated uncertainties from the predicted TD, even though the target classifier fails to predict the true label of the samples correctly. As a future work, we will study extending our TiDAL in the task-agnostic ways with a safeguard combating the wrongly classified samples.

4 Related Work

4.1 Active Learning

AL methods target to construct a dataset with the most useful samples based on the assumption that each sample has different importance in model training (Ren et al., 2021). Two mainstream AL approaches exist for efficiently querying the unlabeled data: pool-based methods (Lewis & Gale, 1994; Yoo & Kweon, 2019; Sinha et al., 2019) use various ways to extract samples from an unlabeled data pool effectively, and synthesis-based methods (Angluin, 1988; Zhu & Bento, 2017; Tran et al.,

2019) generate informative samples for the model. Pool-based methods can be roughly divided based on query strategies: uncertainty-based (Gal et al., 2017; Yoo & Kweon, 2019; Sinha et al., 2019; Huang et al., 2021) and diversity-based (Sener & Savarese, 2018; Gissin & Shalev-Shwartz, 2019; Parvaneh et al., 2022) methods, where some methods use the hybrid of both (Ash et al., 2019; Shui et al., 2020; Kim et al., 2021). Uncertainty-based methods focus on finding which samples would be the most uncertain for the model, whereas diversity-based methods aim to construct a subset of representative samples of the input distribution. Our proposed method, TiDAL, lies in uncertainty-based methods. The significant difference between TiDAL and previous uncertainty-based methods is that TiDAL estimates data uncertainty using TD that contains additional hints generated during model training. In contrast, the previous methods leverage only static information (e.g., loss (Yoo & Kweon, 2019; Huang et al., 2021) and predicted probabilities (Gal et al., 2017; Sinha et al., 2019; Kim et al., 2021)) obtained by a model snapshot at the data selection phase.

4.2 Training Dynamics

TD focuses on how deep neural networks are optimized under back-propagation-based stepwise weight updates. Many studies try to understand how the gradient descent method can effectively obtain the global minimum by analyzing the loss landscape of neural networks (Kawaguchi, 2016; Li et al., 2018) or its loss trajectory (Arora et al., 2018). Some also import alternative models that are more mathematically approachable to analyze, such as neural tangent kernels (Jacot et al., 2018), deep Gaussian processes (Lee et al., 2018), or stochastic differential equations (Zhang et al., 2021b). On the other hand, the phenomenological and practical viewpoint of TD also exists. Toneva et al. (2018) coin the term Forgetting Dynamics to assert that unforgettable samples are often less helpful, and Chang et al. (2017) show that the model could prefer samples that are often wrongly predicted throughout model training. TD is also commonly used in noisy label literature to find potential noisy labels as they tend to fit later on model training (Arazo et al., 2019; Pleiss et al., 2020) or locate samples that can be relabeled correctly (Song et al., 2019). Furthermore, Zhou et al. (2020b) calculate the Dynamic Instance Hardness score by monitoring losses of each sample or whether the prediction gets flipped so that higher scored samples can be prioritized for curriculum learning, and Jiang et al. (2018b) feed the loss history to the auxiliary neural network to mediate the curriculum for training. Samuli & Timo (2017) also introduce temporal ensembling for semi-supervised learning, where the model fits towards averaged probability outputs. Swayamdipta et al. (2020); Park & Caragea (2022) devise Data Maps to inspect datasets with two TD measures; confidence and its variability across epochs on the true class prediction. Zhang & Plank (2021) further extend the Data Maps for AL, whether the target classifier was consistently correct or not during training. The proposed method splits the labeled samples by applying a heuristic threshold on the level of consistency to train a binary classifier that is trained to discern uncertain samples. Even though the work, similar to ours, also utilizes TD, it mainly relies on empirical observations and heuristic choices to divide the certain and uncertain samples. In this study, we link the concept of TD to AL with both empirical and theoretical results to estimate the uncertainty of the unlabeled samples, which is often neglected in previous TD studies.

5 Conclusion

We propose a novel active learning method, Training Dynamics for Active Learning (TiDAL), by linking the concept of training dynamics to active learning. We provide motivating observations and theoretical evidence for using training dynamics to estimate the uncertainty of unlabeled data. Since tracking the training dynamics of large-scale unlabeled data is infeasible, TiDAL utilizes a training dynamics prediction module to efficiently predict the training dynamics of the unlabeled data. Furthermore, we provide two data uncertainty estimation strategies that quantify the data uncertainty by using the predicted training dynamics of the prediction module. Extensive experiments on multiple benchmark datasets demonstrate the effectiveness of our method, surpassing the existing state-of-the-art active learning methods. We further analyze that our training dynamics prediction module successfully predicts the TD of unlabeled data.

REPRODUCIBILITY STATEMENT

We release the source code `https://anonymous.4open.science/r/TiDAL-D1BE` for the main experiments, which all use public datasets that are widely available. We also describe further experimental details in the Appendix. Finally, while we mention informal descriptions of the theorems in the manuscript for ease of understanding, we provide thorough descriptions and proofs in the Appendix.

REFERENCES

Dana Angluin. Queries and concept learning. *Machine learning*, 2(4):319–342, 1988.

Eric Arazo, Diego Ortego, Paul Albert, Noel O'Connor, and Kevin McGuinness. Unsupervised label noise modeling and loss correction. In *International conference on machine learning*, pp. 312–321. PMLR, 2019.

Sanjeev Arora, Nadav Cohen, Noah Golowich, and Wei Hu. A convergence analysis of gradient descent for deep linear neural networks. In *International Conference on Learning Representations*, 2018.

Jordan T Ash, Chicheng Zhang, Akshay Krishnamurthy, John Langford, and Alekh Agarwal. Deep batch active learning by diverse, uncertain gradient lower bounds. *arXiv preprint arXiv:1906.03671*, 2019.

Les E Atlas, David A Cohn, and Richard E Ladner. Training connectionist networks with queries and selective sampling. In *Advances in neural information processing systems*, pp. 566–573. Citeseer, 1990.

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*, pp. 9368–9377, 2018.

Javad Zolfaghari Bengar, Joost van de Weijer, Laura Lopez Fuentes, and Bogdan Raducanu. Class-balanced active learning for image classification. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 1536–1545, 2022.

Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. *Advances in Neural Information Processing Systems*, 32:1567–1578, 2019.

Haw-Shiuan Chang, Erik Learned-Miller, and Andrew McCallum. Active bias: Training more accurate neural networks by emphasizing high variance samples. *Advances in Neural Information Processing Systems*, 30, 2017.

David Cohn, Les Atlas, and Richard Ladner. Improving generalization with active learning. *Machine learning*, 15(2):201–221, 1994.

Charles Corbière, Nicolas Thome, Avner Bar-Hen, Matthieu Cord, and Patrick Pérez. Addressing failure prediction by learning model confidence. *Advances in Neural Information Processing Systems*, 32, 2019.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Yarin Gal, Riashat Islam, and Zoubin Ghahramani. Deep bayesian active learning with image data. In *International Conference on Machine Learning*, pp. 1183–1192. PMLR, 2017.

Daniel Gissin and Shai Shalev-Shwartz. Discriminative active learning. *arXiv preprint arXiv:1907.06347*, 2019.

Jia Gong, Zhipeng Fan, Qiuhong Ke, Hossein Rahmani, and Jun Liu. Meta agent teaming active learning for pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11079–11089, 2022.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International conference on machine learning*, pp. 1321–1330. PMLR, 2017.

Hangfeng He and Weijie Su. The local elasticity of neural networks. In *International Conference on Learning Representations*, 2019.

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*, pp. 770–778, 2016.

Youngkyu Hong, Seungju Han, Kwanghee Choi, Seokjun Seo, Beomsu Kim, and Buru Chang. Disentangling label distribution for long-tailed visual recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6626–6636, 2021.

Siyu Huang, Tianyang Wang, Haoyi Xiong, Jun Huan, and Dejing Dou. Semi-supervised active learning with temporal output discrepancy. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3447–3456, 2021.

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in neural information processing systems*, 31, 2018.

Heinrich Jiang, Been Kim, Melody Guan, and Maya Gupta. To trust or not to trust a classifier. *Advances in neural information processing systems*, 31, 2018a.

Lu Jiang, Zhengyuan Zhou, Thomas Leung, Li-Jia Li, and Li Fei-Fei. Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In *International Conference on Machine Learning*, pp. 2304–2313. PMLR, 2018b.

Kenji Kawaguchi. Deep learning without poor local minima. *Advances in neural information processing systems*, 29, 2016.

Kwanyoung Kim, Dongwon Park, Kwang In Kim, and Se Young Chun. Task-aware variational adversarial active learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8166–8175, 2021.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. The cifar-10 dataset. *online: http://www.cs.toronto.edu/kriz/cifar.html*, 55(5), 2014.

Jaehoon Lee, Yasaman Bahri, Roman Novak, Samuel S Schoenholz, Jeffrey Pennington, and Jascha Sohl-Dickstein. Deep neural networks as gaussian processes. In *International Conference on Learning Representations*, 2018.

David D Lewis and William A Gale. A sequential algorithm for training text classifiers. In *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 3–12, 1994.

Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. Visualizing the loss landscape of neural nets. *Advances in neural information processing systems*, 31, 2018.

Ziwei Liu, Zhongqi Miao, Xiaohang Zhan, Jiayun Wang, Boqing Gong, and Stella X Yu. Large-scale long-tailed recognition in an open world. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2537–2546, 2019.

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 2011*, 2011. URL http://ufldl.stanford.edu/housenumbers/nips2011_housenumbers.pdf.

Curtis Northcutt, Lu Jiang, and Isaac Chuang. Confident learning: Estimating uncertainty in dataset labels. *Journal of Artificial Intelligence Research*, 70:1373–1411, 2021.

Nicolas Papernot and Patrick McDaniel. Deep k-nearest neighbors: Towards confident, interpretable and robust deep learning. *arXiv preprint arXiv:1803.04765*, 2018.

Seo Yeon Park and Cornelia Caragea. A data cartography based mixup for pre-trained language models. *arXiv preprint arXiv:2205.03403*, 2022.

Amin Parvaneh, Ehsan Abbasnejad, Damien Teney, Gholamreza Reza Haffari, Anton van den Hengel, and Javen Qinfeng Shi. Active learning by feature mixing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12237–12246, 2022.

Geoff Pleiss, Tianyi Zhang, Ethan Elenberg, and Kilian Q Weinberger. Identifying mislabeled data using the area under the margin ranking. *Advances in Neural Information Processing Systems*, 33:17044–17056, 2020.

Pengzhen Ren, Yun Xiao, Xiaojun Chang, Po-Yao Huang, Zhihui Li, Brij B Gupta, Xiaojiang Chen, and Xin Wang. A survey of deep active learning. *ACM Computing Surveys (CSUR)*, 54(9):1–40, 2021.

Dan Roth and Kevin Small. Margin-based active learning for structured output spaces. In *European Conference on Machine Learning*, pp. 413–424. Springer, 2006.

Laine Samuli and Aila Timo. Temporal ensembling for semi-supervised learning. In *International Conference on Learning Representations (ICLR)*, volume 4, pp. 6, 2017.

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

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

Changjian Shui, Fan Zhou, Christian Gagné, and Boyu Wang. Deep active learning: Unified and principled method for query and training. In *International Conference on Artificial Intelligence and Statistics*, pp. 1308–1318. PMLR, 2020.

Samarth Sinha, Sayna Ebrahimi, and Trevor Darrell. Variational adversarial active learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5972–5981, 2019.

Hwanjun Song, Minseok Kim, and Jae-Gil Lee. Selfie: Refurbishing unclean samples for robust deep learning. In *International Conference on Machine Learning*, pp. 5907–5915. PMLR, 2019.

Swabha Swayamdipta, Roy Schwartz, Nicholas Lourie, Yizhong Wang, Hannaneh Hajishirzi, Noah A Smith, and Yejin Choi. Dataset cartography: Mapping and diagnosing datasets with training dynamics. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 9275–9293, 2020.

Mariya Toneva, Alessandro Sordoni, Remi Tachet des Combes, Adam Trischler, Yoshua Bengio, and Geoffrey J Gordon. An empirical study of example forgetting during deep neural network learning. In *International Conference on Learning Representations*, 2018.

Toan Tran, Thanh-Toan Do, Ian Reid, and Gustavo Carneiro. Bayesian generative active deep learning. In *International Conference on Machine Learning*, pp. 6295–6304. PMLR, 2019.

Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alex Shepard, Hartwig Adam, Pietro Perona, and Serge Belongie. The inaturalist species classification and detection dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 8769–8778, 2018.

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

Donggeun Yoo and In So Kweon. Learning loss for active learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 93–102, 2019.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115, 2021a.

Jiayao Zhang, Hua Wang, and Weijie Su. Imitating deep learning dynamics via locally elastic stochastic differential equations. *Advances in Neural Information Processing Systems*, 34, 2021b.

Mike Zhang and Barbara Plank. Cartography active learning. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pp. 395–406, 2021.

Boyan Zhou, Quan Cui, Xiu-Shen Wei, and Zhao-Min Chen. Bbn: Bilateral-branch network with cumulative learning for long-tailed visual recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9719–9728, 2020a.

Tianyi Zhou, Shengjie Wang, and Jeffrey Bilmes. Curriculum learning by dynamic instance hardness. *Advances in Neural Information Processing Systems*, 33:8602–8613, 2020b.

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

A Details on the Theoretical Evidence

A.1 Proof of Theorem 1

We adopt the setup and assumptions from (Zhang et al., 2021b), with a slight modification of the assumption that sample-level local elasticity is assumed to affect the training dynamics instead of class-level local elasticity.

Assume the binary classification problem with two classes $k = 1, 2$ and class 1 consists of both certain (easy) and uncertain (hard) samples where class 2 only consists of samples with same certainty (easiness). Let $\mathcal{S}_{1,e}$, $\mathcal{S}_{1,h}$ and $\mathcal{S}_2$ denote the easy samples from class 1, hard samples from class 1, and samples from class 2 respectively, which constitutes the partition of the whole set of training samples $\mathcal{S}$: $\mathcal{S} = \mathcal{S}_{1,e} \cup \mathcal{S}_{1,h} \cup \mathcal{S}_2$. Let the corresponding sample sizes be $n_{1,e} = |\mathcal{S}_{1,e}|$, $n_{1,h} = |\mathcal{S}_{1,h}|$, $n_2 = |\mathcal{S}_2|$ and $n = |\mathcal{S}| = n_{1,e} + n_{1,h} + n_2$, respectively.

At each iteration m, a training candidate sample $J_m \in \mathcal{S}$ is sampled uniformly from the whole training set $\mathcal{S}$ with replacement, having class L_m. Training using this sample J_m via SGD affects the training dynamics of other samples $s \in \mathcal{S}$ of class k as:

$$X_s^k(m) = X_s^k(m-1) + hE_{s,J_m} X_{J_m}^{L_m}(m-1) + \sqrt{h}\zeta_s^k(m-1), \tag{9}$$

where $X > 0$ is the logit of the true label, $h > 0$ is the step size, $\zeta \sim \mathcal{N}(0, \sigma^2)$ denotes the noise term arises during training, and $E \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{S}|}$ refers to the sample-level local elasticity (He & Su, 2019) where each entry $E_{s,s'}$ measures the strength of the local elasticity of s' by s. For simplicity, we assume this local elasticity does not depend on the time step m. Furthermore, assume that that the sample-level local elasticity only depends on the set $\mathcal{S}_{1,e}$, $\mathcal{S}_{1,h}$ and $\mathcal{S}_2$ in which each samples are in.

Let

$$\bar{X}^{1,e}(t) = \frac{\sum_{s\in\mathcal{S}_{1,e}} X_s^1(t)}{n_{1,e}}, \bar{X}^{1,h}(t) = \frac{\sum_{s\in\mathcal{S}_{1,h}} X_s^1(t)}{n_{1,h}}, \bar{X}^2(t) = \frac{\sum_{s\in\mathcal{S}_2} X_s^2(t)}{n_2} \tag{10}$$

be the averaged logits for certain samples in class 1, uncertain samples in class 2, and class 2 respectively.

Regarding the strength of local elasticity between "class" of samples, for some constants α_e, α_h and β, we set the value of $E_{s,s'}$ to model sample-level local elasticity for (1) between easy and hard samples in the class 1 and (2) between classes 1 and 2:

- $E_{s,s'} = \alpha_e$ if $(s, s') \in (\mathcal{S}_{1,e} \times \mathcal{S}_{1,e}) \cup (\mathcal{S}_2 \times \mathcal{S}_2)$,
- $E_{s,s'} = \alpha_h$ if $(s, s') \in (\mathcal{S}_{1,e} \times \mathcal{S}_{1,h}) \cup (\mathcal{S}_{1,h} \times \mathcal{S}_{1,e}) \cup (\mathcal{S}_{1,h} \times \mathcal{S}_{1,h})$,
- $E_{s,s'} = \beta$ otherwise (either $s \in \mathcal{S}_2$ or $s' \in \mathcal{S}_2$ but not both).

We assume $\alpha_e > \alpha_h > \beta > 0$, meaning that the power exerted by sample-level local elasticity between easy samples are stronger for the pair of easy samples than for the pair consists of one or more hard sample. Intuitively, one can interpret the above assumption as easy samples being clustered with each other (Jiang et al., 2018a; Papernot & McDaniel, 2018), hence having a stronger influence on each other due to the local elasticity. On the contrary, hard samples are often distant from other same-class samples. Their influence is often limited, as memorizing is easy for the neural nets due to their large capacity (Zhang et al., 2021a). Finally, we ignore the influence of other class samples in this proof for simplicity, as we are only considering the logits of the true label.

Theorem 1. *(Formal) Under the settings and notations stated in previous paragraphs, convergence speed of logit is faster for easy samples than hard samples on average:*

$$\frac{d\bar{X}^{1,e}(t)}{dt} > \frac{d\bar{X}^{1,h}(t)}{dt}. \tag{11}$$

Proof. Fix a target sample $s \in \mathcal{S}$, and execute the dynamics (9) r times since step m. Accumulated change for feature X becomes

$$X_s^k(m+r) - X_s^k(m) = h\sum_{q=1}^{r} E_{k,L_{m+q}} X_{J_{m+q}}^{L_{m+q}}(m+q-1) + \epsilon_{s,k,r,h}, \tag{12}$$

where $\epsilon = \sqrt{h}\sum_{q=1}^{r}\zeta_s^k(m+q-1) \sim \mathcal{N}(0,\sigma^2 rh)$ is the accumulated noise terms during r updates. Regarding terms inside the summation, we can divide cases based on which sample J_r (with corresponding class L_r) is actually selected as training candidate at iteration $\nu(=m+q)$:

$$\begin{aligned}&E_{k,J_\nu}X_{J_\nu}^{L_\nu}(\nu-1)\\&=\mathbf{1}_{J_\nu\in\mathcal{S}_{1,e}}E_{k,J_\nu}X_{J_\nu}^{1}(\nu-1)+\mathbf{1}_{J_\nu\in\mathcal{S}_{1,h}}E_{k,J_\nu}X_{J_\nu}^{1}(\nu-1)+\mathbf{1}_{J_\nu\in\mathcal{S}_{2}}E_{k,J_\nu}X_{J_\nu}^{2}(\nu-1),\end{aligned}\tag{13}$$

hence the summand from (12) becomes (omitting time index for X for simplicity)

$$h\sum_{q=1}^{r}\left(\mathbf{1}_{J_{m+q}\in\mathcal{S}_{1,e}}E_{k,J_{m+q}}X_{J_{m+q}}^{1}+\mathbf{1}_{J_{m+q}\in\mathcal{S}_{1,h}}E_{k,J_{m+q}}X_{J_{m+q}}^{1}+\mathbf{1}_{J_{m+q}\in\mathcal{S}_{2}}E_{k,J_{m+q}}X_{J_{m+q}}^{2}\right),$$

and for sufficiently large r we can approximate the summations as the sample-average dynamics:

$$\begin{aligned}&h\sum_{q=1}^{r}\left(\mathbf{1}_{J_{m+q}\in\mathcal{S}_{1,e}}E_{k,J_{m+q}}X_{J_{m+q}}^{1}+\mathbf{1}_{J_{m+q}\in\mathcal{S}_{1,h}}E_{k,J_{m+q}}X_{J_{m+q}}^{1}+\mathbf{1}_{J_{m+q}\in\mathcal{S}_{2}}E_{k,J_{m+q}}X_{J_{m+q}}^{2}\right)\\&\approx hr\left(\mathbb{P}\left(J\in\mathcal{S}_{1,e}\right)\frac{\sum_{s\in\mathcal{S}_{1,e}}E_{k,s}X_s^1}{n_{1,e}}+\mathbb{P}\left(J\in\mathcal{S}_{1,h}\right)\frac{\sum_{s\in\mathcal{S}_{1,h}}E_{k,s}X_s^1}{n_{1,h}}+\mathbb{P}\left(J\in\mathcal{S}_{2}\right)\frac{\sum_{s\in\mathcal{S}_{2}}E_{k,s}X_s^2}{n_2}\right)\\&\approx hr\left(\frac{n_{1,e}}{n}\frac{\sum_{s\in\mathcal{S}_{1,e}}E_{k,s}X_s^1}{n_{1,e}}+\frac{n_{1,h}}{n}\frac{\sum_{s\in\mathcal{S}_{1,h}}E_{k,s}X_s^1}{n_{1,h}}+\frac{n_2}{n}\frac{\sum_{s\in\mathcal{S}_{2}}E_{k,s}X_s^2}{n_2}\right)\end{aligned}\tag{14}$$

As the components of E only depends on the subset sample relies, we can rewrite accumulated dynamics of logits (12) for three cases separately, utilizing the notation of averaged logit (10):

$$\begin{aligned}X_s^{1,e}(m+r)-X_s^{1,e}(m)&=hr\left(\frac{n_{1,e}}{n}\alpha_e\bar{X}^{1,e}(m)+\frac{n_{1,h}}{n}\alpha_h\bar{X}^{1,h}(m)+\frac{n_2}{n}\beta\bar{X}^{2}(m)\right)+\epsilon_{s,k,r,h}\\X_s^{1,h}(m+r)-X_s^{1,h}(m)&=hr\left(\frac{n_{1,e}}{n}\alpha_h\bar{X}^{1,e}(m)+\frac{n_{1,h}}{n}\alpha_h\bar{X}^{1,h}(m)+\frac{n_2}{n}\beta\bar{X}^{2}(m)\right)+\epsilon_{s,k,r,h}\\X_s^{2}(m+r)-X_s^{2}(m)&=hr\left(\frac{n_{1,e}}{n}\beta\bar{X}^{1,e}(m)+\frac{n_{1,h}}{n}\beta\bar{X}^{1,h}(m)+\frac{n_2}{n}\alpha_e\bar{X}^{2}(m)\right)+\epsilon_{s,k,r,h},\end{aligned}\tag{15}$$

with a little bit of abbreviated notation for class 1: $X_s^{1,e}=X_s^1$ for easy sample s, and similarly for hard samples. The differential counterpart of above difference equation is

$$\begin{aligned}dX_s^{1,e}(t)&=\left(\frac{n_{1,e}}{n}\alpha_e\bar{X}^{1,e}(t)+\frac{n_{1,h}}{n}\alpha_h\bar{X}^{1,h}(t)+\frac{n_2}{n}\beta\bar{X}^{2}(t)\right)dt+\sigma dW^s(t)\\dX_s^{1,h}(t)&=\left(\frac{n_{1,e}}{n}\alpha_h\bar{X}^{1,e}(t)+\frac{n_{1,h}}{n}\alpha_h\bar{X}^{1,h}(t)+\frac{n_2}{n}\beta\bar{X}^{2}(t)\right)dt+\sigma dW^s(t)\\dX_s^{2}(t)&=\left(\frac{n_{1,e}}{n}\beta\bar{X}^{1,e}(t)+\frac{n_{1,h}}{n}\beta\bar{X}^{1,h}(t)+\frac{n_2}{n}\alpha_e\bar{X}^{2}(t)\right)dt+\sigma dW^s(t),\end{aligned}\tag{16}$$

where $W^s(t)$ is standard Wiener process per sample. Averaging each differential equations with respect to each set of samples and ignoring error terms yield a set of simultaneous deterministic differential equations for averaged logits:

$$\begin{aligned}d\bar{X}^{1,e}(t)&=\left(\frac{n_{1,e}}{n}\alpha_e\bar{X}^{1,e}(t)+\frac{n_{1,h}}{n}\alpha_h\bar{X}^{1,h}(t)+\frac{n_2}{n}\beta\bar{X}^{2}(t)\right)dt\\d\bar{X}^{1,h}(t)&=\left(\frac{n_{1,e}}{n}\alpha_h\bar{X}^{1,e}(t)+\frac{n_{1,h}}{n}\alpha_h\bar{X}^{1,h}(t)+\frac{n_2}{n}\beta\bar{X}^{2}(t)\right)dt\\d\bar{X}^{2}(t)&=\left(\frac{n_{1,e}}{n}\beta\bar{X}^{1,e}(t)+\frac{n_{1,h}}{n}\beta\bar{X}^{1,h}(t)+\frac{n_2}{n}\alpha_e\bar{X}^{2}(t)\right)dt,\end{aligned}\tag{17}$$

To compare the convergence speed of average logit between certain and uncertain samples in the same class 1, observe that

$$\frac{d\bar{X}^{1,e}(t)}{dt}-\frac{d\bar{X}^{1,h}(t)}{dt}=\frac{n_{1,e}}{n}(\alpha_e-\alpha_h)\bar{X}^{1,e}(t)>0.\tag{18}$$

□

With additional assumptions on the other class logits being the same, one can also conclude that the estimated probability of the true label will increase steeply during training for the easy samples. After increasing to some extent, the probability will saturate to one; hence the snapshot model predictions will contain less useful information than monitoring its training dynamics. However, future work on extending the above theorem is needed. Starting from the basic idea above that sample proximity and its amount influence the training dynamics, one can further relax the above assumptions, such as concentrating on the individuality of each sample or considering the changing elasticities during training. We hope our work ignites the theoretical research on uncertainty from the viewpoint of training dynamics.

A.2 Proof of Theorem 2

We aim to show the effectiveness of the proposed estimators, entropy (Equation 1) and margin (Equation 2), especially in the case where the probabilities converge. After training, it is commonly observed that the probabilities of the true label of all the samples tend to converge to one, whereas the speed of the convergence differs (Theorem 1). Hence, we show that the estimators can effectively discern the differences during training.

For each time step t during training, we have a sequence of predicted probabilities $p^{(t)}(y = c|x)$ corresponds to t, for each target class $c = 1, 2, \cdots, C$. In our paper, we regard the area under the predicted probability $\bar{p}^{(T)}(y = c|x)$ of the sample x as the training dynamics (Equation 3), which is indeed a well-known metric of area under the curve, except that it is normalized properly to have value between 0 and 1. For convenience, let

$$\boldsymbol{s}(x) = \begin{bmatrix} s_1(x) \\ s_2(x) \\ \vdots \\ s_C(x) \end{bmatrix} = \begin{bmatrix} \bar{p}^{(T)}(y = 1|x) \\ \bar{p}^{(T)}(y = 2|x) \\ \vdots \\ \bar{p}^{(T)}(y = C|x) \end{bmatrix}$$

be the vector consisting the area under the prediction curve for each class up to final epoch T. Observe that, by definition, the components in $\boldsymbol{s}(x)$ are nonnegative and sum to 1.

Theorem 2. *(Formal) Assume that all target classes have the same area under the prediction curve except for the true class y. Suppose two training samples $(x_1, y_1), (x_2, y_2) \in \mathcal{D}$ satisfies*

a. $p^{(T)}(y_1|x_1)$=$p^{(T)}(y_2|x_2)$ *(same predicted probability at the end of training)*

b. $\frac{1}{2} < s_{y_1}(x_1) < s_{y_2}(x_2)$ *(but different TD, in terms of the area under the curve)*

Then, the following inequalities hold:

1. $H(\boldsymbol{s}(x_1)) > H(\boldsymbol{s}(x_2))$;
2. $M(\boldsymbol{s}(x_1)) < M(\boldsymbol{s}(x_2))$.

Proof. By the assumption, for all target classes except the true class y, the area under the prediction curve is given by

$$s_c(x) = \frac{1 - s_y(x)}{C - 1}, \tag{19}$$

and the corresponding entropy can be calculated as

$$\begin{aligned} H(\boldsymbol{s}(x)) &= \sum_{c=1}^{C} (-s_c(x)\log(s_c(x))) \\ &= -\left\{ s_y(x)\log(s_y(x)) + (C-1) \cdot \frac{1 - s_y(x)}{C-1} \log\left(\frac{1 - s_y(x)}{C-1}\right) \right\}, \\ &= -\{s_y(x)\log(s_y(x)) + (1 - s_y(x))\log(1 - s_y(x))\} + (1 - s_y(x))\log(C-1), \\ &= H_2(s_y(x)) + (1 - s_y(x))\log(C-1). \end{aligned} \tag{20}$$

where we used the notation of binary entropy function $H_2(p) = -p\log(p) - (1-p)\log(1-p)$, which is a decreasing function of p for $p > \frac{1}{2}$. Therefore

$$H(\boldsymbol{s}(x_1)) - H(\boldsymbol{s}(x_2)) = \big(H_2(s_{y_1}(x_1)) - H_2(s_{y_2}(x_2))\big) + \big(s_{y_2}(x_2) - s_{y_1}(x_1)\big)\log(C-1) > 0.$$

The first assumption also gives the simplified formulation for the margin:

$$M(\boldsymbol{s}(x)) = s_y(x) - \frac{1 - s_y(x)}{C-1} = \frac{C}{C-1}s_y(x) - \frac{1}{C-1}, \tag{21}$$

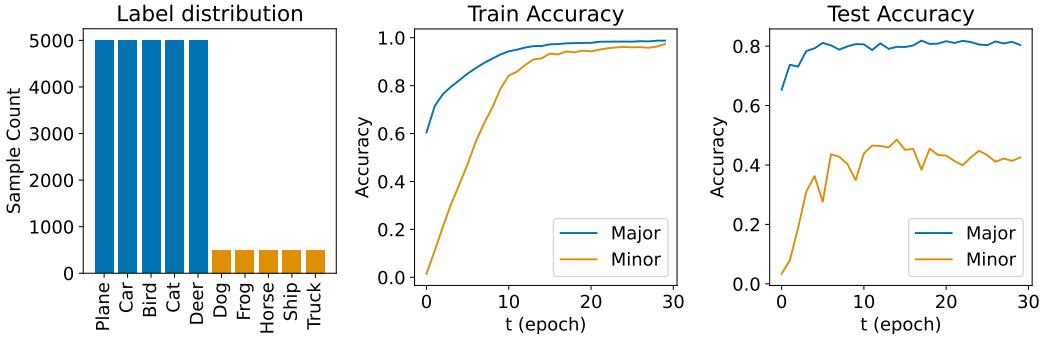
which results in the second inequality

$$M(\boldsymbol{s}(x_1)) - M(\boldsymbol{s}(x_2)) = \frac{C}{C-1}\left(s_{y_1}(x_1) - s_{y_2}(x_2)\right) < 0 \tag{22}$$

□

While the final predicted probabilities $p^{(T)}(y|x)$ of the training samples tend to converge to 1 for the true class y, otherwise 0, their TD (in this case $\boldsymbol{s}(x) = \bar{\boldsymbol{p}}^{(T)}$) may be different depending on the easiness of the samples. Thus, the degree of the easiness of the samples (i.e. uncertainty) could be captured from TD $\bar{\boldsymbol{p}}$, whereas the predictions $\boldsymbol{p}$ from a model snapshot cannot.

B Details on the Motivating Observation



(a) Training set label distribution. (b) Training accuracy per epoch. (c) Test accuracy per epoch.

Figure 8: Training label distribution and accuracy curves for the motivating experiment in §2.3.1.

§ 2.3.1 empirically show that using TD is effective in separating uncertain samples from certain samples. Before diving into the experimental details, we want to emphasize that it is difficult to control the level of data difficulty (or uncertainty). First and foremost, human perception of data difficulty will be highly subjective and potentially different from its model counterpart. This limitation hinders the quantitative analysis, and thus some previous works had to rely on qualitative substitutes or analyze mislabeled samples which are impossible to control its difficulty (Pleiss et al., 2020; Northcutt et al., 2021; Toneva et al., 2018). Also, even if we could obtain sample-wise difficulty, it is often nontrivial to analyze the overall trend during training due to sheer data size.

To avoid the two challenges above, we borrow the settings from studies on long-tail visual recognition (Liu et al., 2019; Bengar et al., 2022). Cao et al. (2019) show that generalization error is bounded by the inverse square root of the dataset size. Further, many long-tail literature (Liu et al., 2019; Zhou et al., 2020a; Hong et al., 2021) have also empirically shown that it is hard for the deep neural network-based model to train with fewer samples, showing lower accuracy. Hence, we consider the major and minor classes as certain and uncertain classes, as the binned classification error is often used as the definition of confidence (Guo et al., 2017).

We train ResNet-18 (He et al., 2016) on the CIFAR10 dataset (Krizhevsky et al., 2014; Cao et al., 2019) with an imbalance ratio of 10 for 30 epochs using the Adam optimizer (Kingma & Ba, 2015).

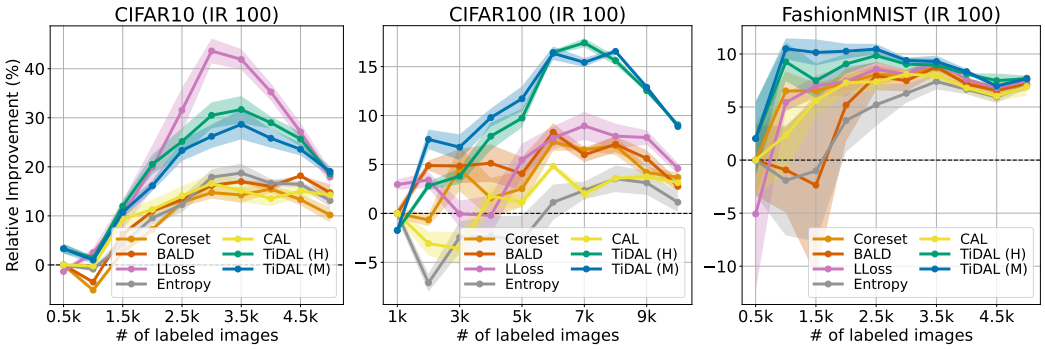


Figure 9: Averaged relative accuracy improvement curves and their 95% confidence interval (shaded) of AL methods over the number of labeled samples on synthetically imbalanced datasets. We use the imbalance ratio (IR) of 100 on CIFAR10, CIFAR100, and FashionMNIST.

Figure 8a shows the label distribution of the training dataset. Similar to Bengar et al. (2022), we choose classes 0 4 as the major class and the rest as the minor class, randomly removing 90% of the training samples for the minor class. We reduce the inter-class differences of CIFAR10 by merging five classes into one, and demonstrate both the overall distribution and samplewise scores in Figure 2. We conclude that TD successfully captures data uncertainties, where its characteristics are more helpful in separating uncertain samples from certain samples than the information obtained from a model snapshot. Also, we empirically reaffirm that the major classes being more advantageous than minor classes in terms of accuracy during model training (Figure 8b, 8c).

Table 1: The details of the training set of datasets.

Dataset	# of classes	# of samples	Imbalance ratio
CIFAR10	10	50k	{1, 10, 100}
CIFAR100	100	50k	{1, 10, 100}
FashionMNIST	10	60k	{1, 10, 100}
SVHN	10	73k	2.98
iNaturalist2018	8k	437k	500

C Additional Experiments

We conduct additional experiments to further demonstrate the effectiveness of our method, TiDAL. We provide the detailed implementations in `https://anonymous.4open.science/r/TiDAL-D1BE` and the dataset statistics in Table 1. We also supply dataset statistics in Table 1.

C.1 Additional Results on Imbalanced Datasets

As previously mentioned in the manuscript, we also supply the experimental results on the imbalance ratio 100. Except for CIFAR10, our methods show superiority over other state-of-the-art methods.

C.2 Additional Results on Absolute Accuracy

We also provide the absolute accuracy plots for the completeness of the evaluation. We can observe the superiority of our method further on many of the settings.

C.3 Additional Baselines

Figure 12 compares our TiDAL with VAAL Sinha et al. (2019) and TA-VAAL Kim et al. (2021). Except for the case of CIFAR10 with the imbalance ratio of 100, both TiDAL strategies excel in

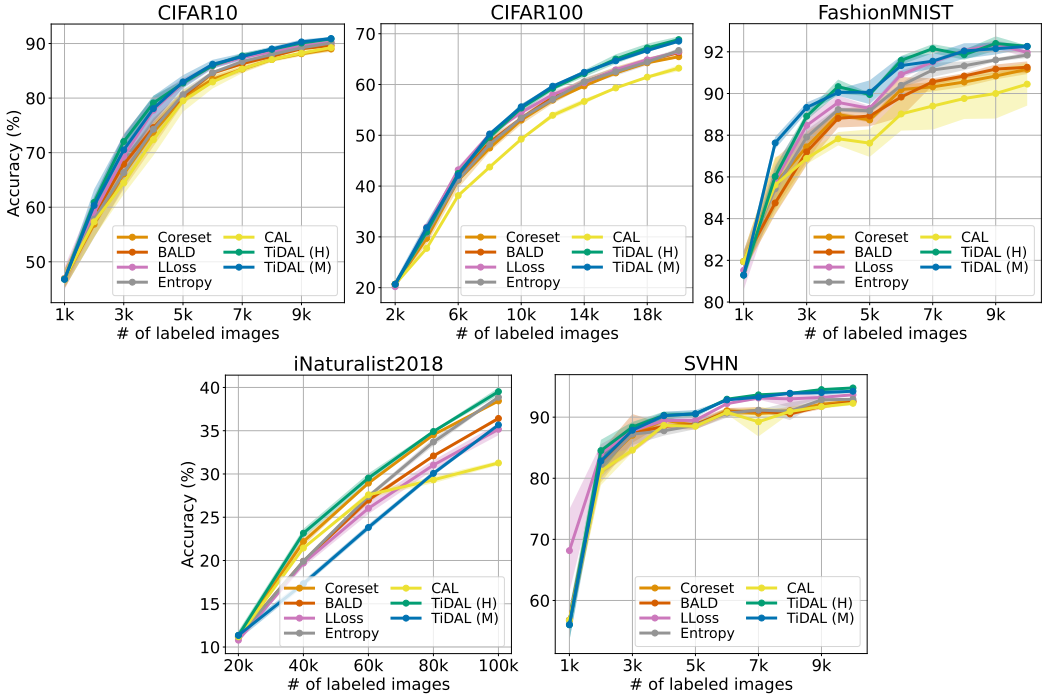


Figure 10: Averaged absolute accuracy improvement curves and its 95% confidence interval (shaded) of AL methods over the number of labeled samples on balanced and imbalanced datasets.

performance. Note that both VAAL and TA-VAAL use a semi-supervised approach to train the selection module and further leverage the unlabeled data for training.

C.4 Variants of Training Dynamics-Aware Margin

We introduced two TD-aware strategies: entropy $\bar{H}$ and margin $\bar{M}$, in §2.2. We further demonstrate various uncertainty estimation strategies as follows:

$$\bar{M}_0(\tilde{\boldsymbol{p}}_m) = \tilde{p}_m(\tilde{y}|x) - \max_{c \neq \tilde{y}} \tilde{p}_m(c|x), \tag{23}$$

$$\bar{P}(\tilde{\boldsymbol{p}}_m) = \tilde{p}_m(\hat{y}|x), \tag{24}$$

$$\bar{P}_0(\tilde{\boldsymbol{p}}_m) = \tilde{p}_m(\tilde{y}|x), \tag{25}$$

where $\tilde{y} = \text{argmax}_c\, \tilde{\boldsymbol{p}}_m(c|x)$ is the class of the maximum module output.

$\hat{M}_0$ is the naive variant of the margin $\hat{M}$ where it does not utilize the predicted label $\hat{y}$ of the target classifier f. It calculates the margin between the biggest and the second biggest outputs of the module m. $\bar{P}$ uses the module output on the predicted label $\hat{y}$ from the target classifier f and $\bar{P}_0$ is the naive variant of $\bar{P}$ that uses the maximum output of the module m.

Figure 13 shows the average accuracy of three runs for the entropy $\bar{H}$ and margin $\bar{M}$, where we show the accuracy of a single run for other strategies. We can observe that the naive variant of the margin $\bar{M}_0$ generally underperforms compared to the margin $\bar{M}$ except CIFAR100 with the imbalance ratio of 100. There seems to be no clear dominance between $\bar{P}$ and its naive variant $\bar{P}_0$. However, both $\bar{P}$ and $\bar{P}_0$ perform moderately well on both CIFAR100 and FashionMNIST despite its simplicity. Future studies may concentrate on broader query strategies based on various training dynamics and its module predictions.

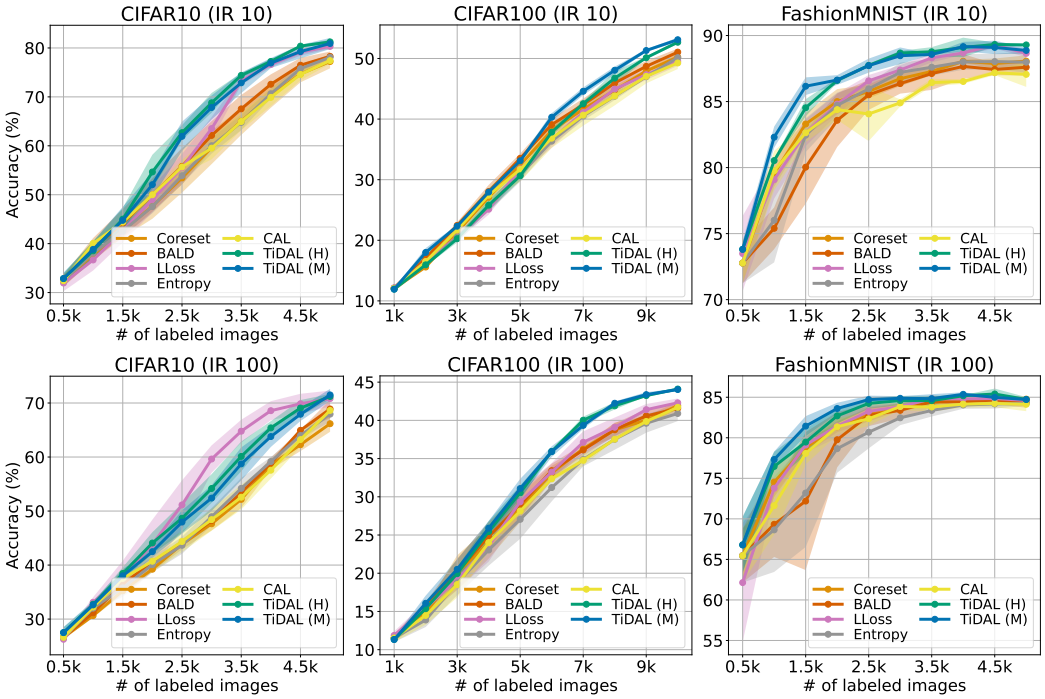


Figure 11: Averaged absolute accuracy improvement curves and their 95% confidence interval (shaded) of AL methods over the number of labeled samples on synthetically imbalanced datasets. We use the imbalance ratio (IR) of 10 and 100 on CIFAR10, CIFAR100, and FashionMNIST.

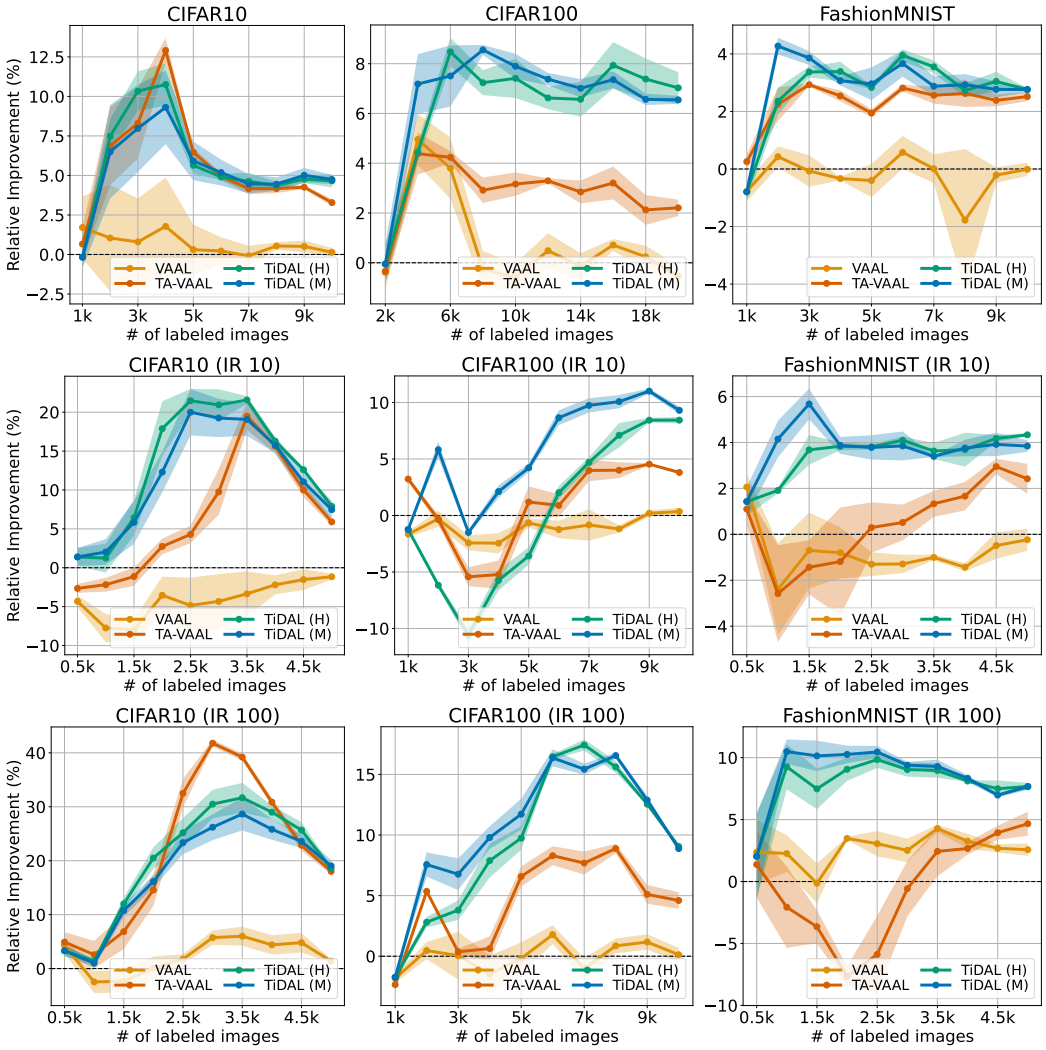


Figure 12: Averaged relative accuracy improvement curves and their 95% confidence interval (shaded) of AL methods over the number of labeled samples on balanced and synthetically imbalanced datasets. We use the imbalance ratio (IR) of 10 and 100 on CIFAR10, CIFAR100, and FashionMNIST to synthetically imbalance the dataset.

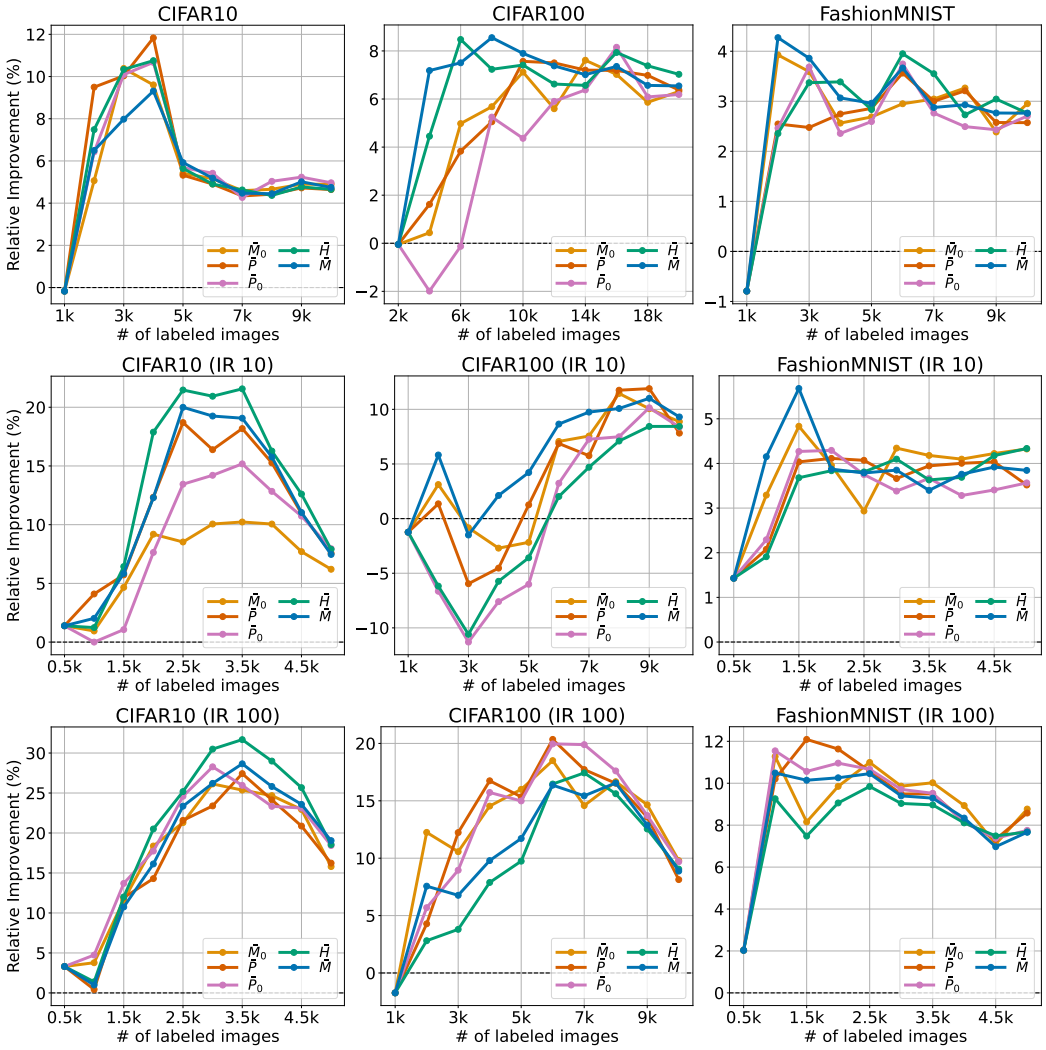


Figure 13: Averaged relative accuracy improvement curves of different uncertainty estimation strategies over the number of labeled samples on balanced and synthetically imbalanced datasets. We use the imbalance ratio (IR) of 10 and 100 on CIFAR10, CIFAR100, and FashionMNIST to synthetically imbalance the dataset.