Appendix

A. Details on the Theoretical Evidence

A.1. Proof of Theorem 1

We adopt the settings of [54], with a slight modification of the assumption that sample-level local elasticity affects the training dynamics instead of class-level local elasticity.

Consider the binary classification problem with two classes $k = 1, 2$ where class 1 consists of both certain (easy) and uncertain (hard) samples and class 2 only consists of samples with the 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 $J_m \in \mathcal{S}$ with class L_m is sampled uniformly from the whole training set $\mathcal{S}$ with replacement. Training using this sample J_m via SGD affects the training dynamics of other samples $s \in \mathcal{S}$ of class k as:

$$X_s^k(m) = X_s^k(m-1) + hE_{s,J_m} X_{J_m}^{L_m}(m-1) + \sqrt{h}\zeta_s^k(m-1), \tag{12}$$

where $X > 0$ is 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 [17] 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, we consider 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

$$\bar{X}^{1,e}(t) = \frac{1}{n_{1,e}} \sum_{s\in\mathcal{S}_{1,e}} X_s^1(t), \bar{X}^{1,h}(t) = \frac{1}{n_{1,h}} \sum_{s\in\mathcal{S}_{1,h}} X_s^1(t), \bar{X}^2(t) = \frac{1}{n_2} \sum_{s\in\mathcal{S}_2} X_s^2(t) \tag{13}$$

be the averaged logits for certain samples in class 1, uncertain samples in class 1, 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. We use the values $\alpha_e > \alpha_h > \beta > 0$ to define the easiness such 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.

- $E_{s,s'} = \alpha_e$ 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 $(s, s') \in (\mathcal{S}_{1,e} \times \mathcal{S}_{1,h}) \cup (\mathcal{S}_{1,h} \times \mathcal{S}_{1,e}) \cup (\mathcal{S}_{1,h} \times \mathcal{S}_{1,h})$ (either $s \in \mathcal{S}_{1,h}$ or $s' \in \mathcal{S}_{1,h}$),
- $E_{s,s'} = \beta$ otherwise.

Intuitively, one can interpret the above assumption as easy samples being clustered with each other [22, 36], 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 [53]. 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) On average, the convergence speed of logit is faster for easy samples than hard samples. Formally:*

$$\frac{d\bar{X}^{1,e}(t)}{dt} > \frac{d\bar{X}^{1,h}(t)}{dt}. \tag{14}$$

Proof. Fix a target sample $s \in \mathcal{S}$, and execute the dynamics (12) r times since step m. Accumulated change for feature X becomes

$$X_s^k(m+r) - X_s^k(m) = h\sum_{q=1}^{r} E_{k,L_{m+q}} X_{J_{m+q}}^{L_{m+q}}(m+q-1) + \epsilon_{s,k,r,h}, \tag{15}$$

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 a training candidate at iteration $\nu(= m+q)$:

$$E_{k,J_\nu} X_{J_\nu}^{L_\nu}(\nu-1) = \mathbf{1}_{J_\nu\in\mathcal{S}_{1,e}} E_{k,J_\nu} X_{J_\nu}^{1}(\nu-1) + \mathbf{1}_{J_\nu\in\mathcal{S}_{1,h}} E_{k,J_\nu} X_{J_\nu}^{1}(\nu-1) + \mathbf{1}_{J_\nu\in\mathcal{S}_{2}} E_{k,J_\nu} X_{J_\nu}^{2}(\nu-1), \tag{16}$$

hence the summand from (15) becomes (omitting time index for X for simplicity)

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

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

$$\begin{aligned}
&h\sum_{q=1}^{r}\left(\mathbf{1}_{J_{m+q}\in\mathcal{S}_{1,e}}E_{k,J_{m+q}}X^1_{J_{m+q}}+\mathbf{1}_{J_{m+q}\in\mathcal{S}_{1,h}}E_{k,J_{m+q}}X^1_{J_{m+q}}+\mathbf{1}_{J_{m+q}\in\mathcal{S}_{2}}E_{k,J_{m+q}}X^2_{J_{m+q}}\right)\\
&\approx hr\left(\mathbb{P}\left(J\in\mathcal{S}_{1,e}\right)\frac{\sum_{s\in\mathcal{S}_{1,e}}E_{k,s}X^1_s}{n_{1,e}}+\mathbb{P}\left(J\in\mathcal{S}_{1,h}\right)\frac{\sum_{s\in\mathcal{S}_{1,h}}E_{k,s}X^1_s}{n_{1,h}}+\mathbb{P}\left(J\in\mathcal{S}_{2}\right)\frac{\sum_{s\in\mathcal{S}_{2}}E_{k,s}X^2_s}{n_{2}}\right)\\
&\approx hr\left(\frac{n_{1,e}}{n}\frac{\sum_{s\in\mathcal{S}_{1,e}}E_{k,s}X^1_s}{n_{1,e}}+\frac{n_{1,h}}{n}\frac{\sum_{s\in\mathcal{S}_{1,h}}E_{k,s}X^1_s}{n_{1,h}}+\frac{n_2}{n}\frac{\sum_{s\in\mathcal{S}_{2}}E_{k,s}X^2_s}{n_{2}}\right)
\end{aligned}\tag{17}$$

As the components of E only depend on the subset sample relies, we can rewrite accumulated dynamics of logits (15) for three cases separately, utilizing the notation of averaged logit (13):

$$\begin{aligned}
X^{1,e}_s(m+r)-X^{1,e}_s(m)&=hr\left(\frac{n_{1,e}}{n}\alpha_e\bar{X}^{1,e}(m)+\frac{n_{1,h}}{n}\alpha_h\bar{X}^{1,h}(m)+\frac{n_2}{n}\beta\bar{X}^2(m)\right)+\epsilon_{s,k,r,h}\\
X^{1,h}_s(m+r)-X^{1,h}_s(m)&=hr\left(\frac{n_{1,e}}{n}\alpha_h\bar{X}^{1,e}(m)+\frac{n_{1,h}}{n}\alpha_h\bar{X}^{1,h}(m)+\frac{n_2}{n}\beta\bar{X}^2(m)\right)+\epsilon_{s,k,r,h}\\
X^{2}_s(m+r)-X^{2}_s(m)&=hr\left(\frac{n_{1,e}}{n}\beta\bar{X}^{1,e}(m)+\frac{n_{1,h}}{n}\beta\bar{X}^{1,h}(m)+\frac{n_2}{n}\alpha_e\bar{X}^2(m)\right)+\epsilon_{s,k,r,h},
\end{aligned}\tag{18}$$

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

$$\begin{aligned}
dX^{1,e}_s(t)&=\left(\frac{n_{1,e}}{n}\alpha_e\bar{X}^{1,e}(t)+\frac{n_{1,h}}{n}\alpha_h\bar{X}^{1,h}(t)+\frac{n_2}{n}\beta\bar{X}^2(t)\right)dt+\sigma dW^s(t)\\
dX^{1,h}_s(t)&=\left(\frac{n_{1,e}}{n}\alpha_h\bar{X}^{1,e}(t)+\frac{n_{1,h}}{n}\alpha_h\bar{X}^{1,h}(t)+\frac{n_2}{n}\beta\bar{X}^2(t)\right)dt+\sigma dW^s(t)\\
dX^{2}_s(t)&=\left(\frac{n_{1,e}}{n}\beta\bar{X}^{1,e}(t)+\frac{n_{1,h}}{n}\beta\bar{X}^{1,h}(t)+\frac{n_2}{n}\alpha_e\bar{X}^2(t)\right)dt+\sigma dW^s(t),
\end{aligned}\tag{19}$$

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

$$\begin{aligned}
d\bar{X}^{1,e}(t)&=\left(\frac{n_{1,e}}{n}\alpha_e\bar{X}^{1,e}(t)+\frac{n_{1,h}}{n}\alpha_h\bar{X}^{1,h}(t)+\frac{n_2}{n}\beta\bar{X}^2(t)\right)dt\\
d\bar{X}^{1,h}(t)&=\left(\frac{n_{1,e}}{n}\alpha_h\bar{X}^{1,e}(t)+\frac{n_{1,h}}{n}\alpha_h\bar{X}^{1,h}(t)+\frac{n_2}{n}\beta\bar{X}^2(t)\right)dt\\
d\bar{X}^{2}(t)&=\left(\frac{n_{1,e}}{n}\beta\bar{X}^{1,e}(t)+\frac{n_{1,h}}{n}\beta\bar{X}^{1,h}(t)+\frac{n_2}{n}\alpha_e\bar{X}^2(t)\right)dt,
\end{aligned}\tag{20}$$

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

$$\frac{d\bar{X}^{1,e}(t)}{dt}-\frac{d\bar{X}^{1,h}(t)}{dt}=\frac{n_{1,e}}{n}(\alpha_e-\alpha_h)\bar{X}^{1,e}(t)>0.\tag{21}$$

□

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 2) and margin (Equation 3), 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 5), 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. By construction, 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 class c except the true class y, the area under the prediction curve is given by

$$s_c(x) = \frac{1 - s_y(x)}{C - 1}, \tag{22}$$

and the corresponding entropy can be calculated as

$$\begin{aligned} H(\boldsymbol{s}(x)) &= \sum_{c=1}^{C} (-s_c(x)\text{log}(s_c(x))) \\ &= -s_y(x)\text{log}s_y(x) - (C-1) \cdot \left(\frac{1 - s_y(x)}{C - 1}\right) \log \left(\frac{1 - s_y(x)}{C - 1}\right) \\ &= -\left\{s_y(x)\text{log}s_y(x) + (1 - s_y(x))\text{log}(1 - s_y(x))\right\} + (1 - s_y(x))\text{log}(C-1) \\ &= H_2(s_y(x)) + (1 - s_y(x))\text{log}(C-1). \end{aligned} \tag{23}$$

where $H_2(p) = -p\text{log}p - (1-p)\text{log}(1-p)$ stands for the binary entropy function. Since $H_2(p)$ is a decreasing function for $p > \frac{1}{2}$,

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

which proves the first inequality stated.

The first assumption also gives the simplified formulation for the margin

$$M(\boldsymbol{s}(x)) = s_y(x) - \frac{1 - s_y(x)}{C - 1} = \frac{C}{C-1} s_y(x) - \frac{1}{C-1}, \tag{24}$$

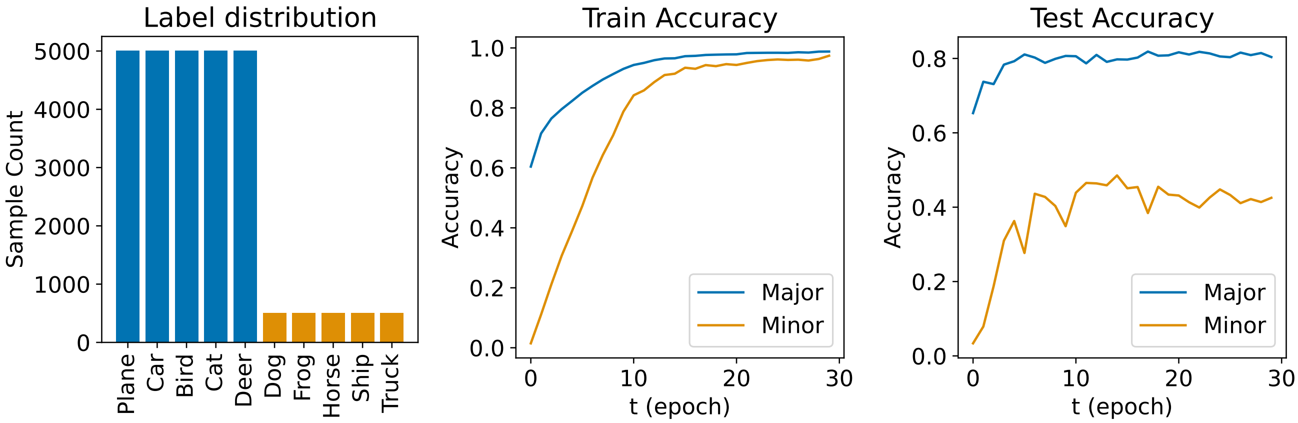
in whicn the second inequality directly follows:

$$M(\boldsymbol{s}(x_1)) - M(\boldsymbol{s}(x_2)) = \frac{C}{C-1} (s_{y_1}(x_1) - s_{y_2}(x_2)) < 0. \tag{25}$$

□

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 §3.1.

§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 [39, 35, 48]. 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 [33, 7]. [8] show that generalization error is bounded by the inverse square root of the dataset size. Further, many long-tail literature [33, 56, 19] 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 [16].

We train ResNet-18 [18] on the CIFAR10 dataset [28, 8] with an imbalance ratio of 10 for 30 epochs using the Adam optimizer [26]. Figure 8a shows the label distribution of the training dataset. Similar to [7], we choose classes $0, 1, 2, 3$ and 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).

C. Details on the TD Prediction Module

One can offer numerous alternatives on the design of the TD prediction module m, but we adopt the architecture of the loss prediction module [52] except for the last layer. By adopting the architecture used in the previous study, it is intended to show that the performance improvement of TiDAL does not come from adopting an advanced prediction module architecture, but from using TD. The TD prediction module takes several hidden feature maps extracted between the mid-level blocks of the target classifier f as inputs. Through a global average pooling layer and a fully-connected layer, each feature map is reduced to a fixed dimensional feature vector. All the reduced feature vectors are concatenated to take multi-level knowledge

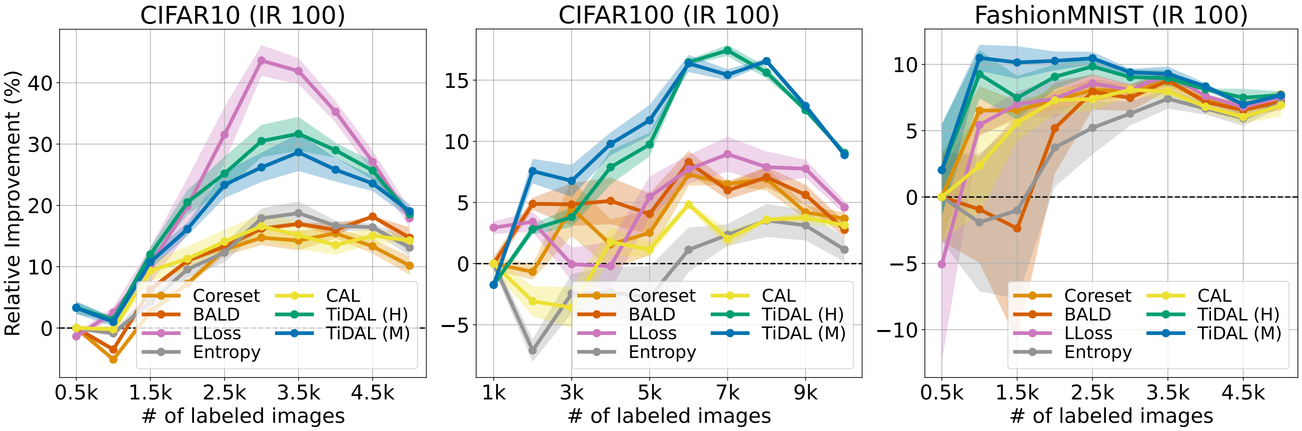


Figure 9: Averaged relative accuracy improvement curves and their 95% confidence interval (shaded) of AL methods over the number of labeled samples on synthetically imbalanced datasets. We use the imbalance ratio (IR) of 100 on CIFAR10, CIFAR100, and FashionMNIST.

of the target classifier into consideration for TD prediction. Using a single Softmax layer, the TD prediction module outputs a C-dimensional prediction $\tilde{y}^{(t)} \in [0, 1]^C$, which are used as the predicted TD.

For a better understanding of the architecture of our TD prediction module m, please refer to [52].

D. Additional Experiments

We conduct additional experiments to further demonstrate the effectiveness of our method, TiDAL. We provide the dataset statistics in Table 1.

Table 1: Statistics of the dataset used for experiments.

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

D.1. Additional Results on Imbalanced Datasets

Figure 9 shows the experimental results on the imbalance ratio 100. Except for CIFAR10, our methods show superiority over other state-of-the-art methods.

D.2. Additional Results on Absolute Accuracy

Figure 10 and 11 provides the absolute accuracy plots for the completeness of the evaluation for real and synthetic data, respectively. We can observe the superiority of our method further on many of the settings.

D.3. Additional Baselines

Figure 12 compares our TiDAL with VAAL [45] and TA-VAAL [25]. Except for the case of CIFAR10 with the imbalance ratio of 100, both TiDAL strategies excel in 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.

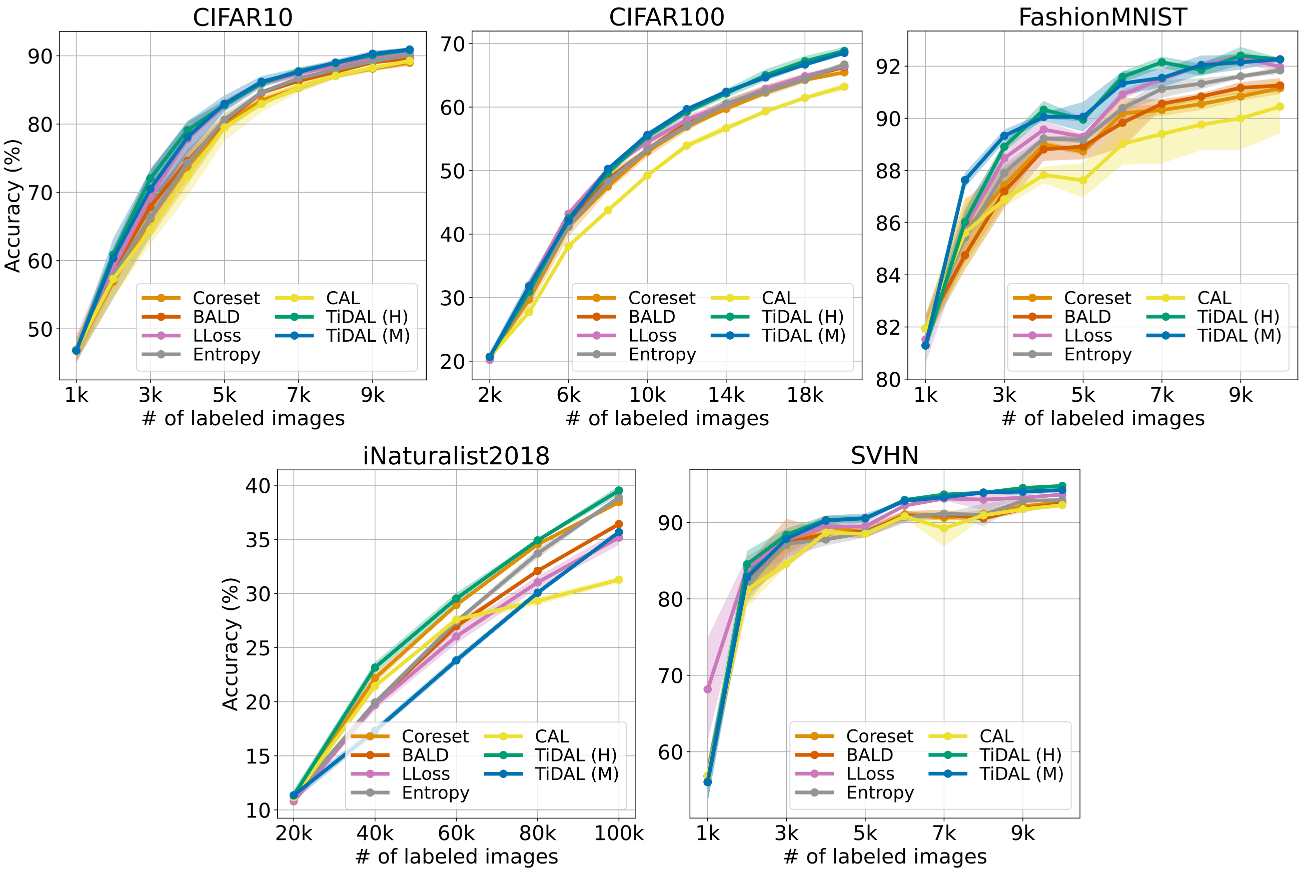


Figure 10: Averaged absolute accuracy improvement curves and its 95% confidence interval (shaded) of AL methods over the number of labeled samples on balanced and imbalanced datasets.

D.4. Variants of Training Dynamics-Aware Margin

We introduced two TD-aware strategies: entropy $\bar{H}$ and margin $\bar{M}$, in §2. We further demonstrate various uncertainty estimation strategies as follows:

$$\bar{M}_0(\tilde{\boldsymbol{p}}_m) = \tilde{p}_m(\tilde{y}|x) - \max_{c\neq\tilde{y}} \tilde{p}_m(c|x), \tag{26}$$

$$\bar{P}(\tilde{\boldsymbol{p}}_m) = \tilde{p}_m(\hat{y}|x), \tag{27}$$

$$\bar{P}_0(\tilde{\boldsymbol{p}}_m) = \tilde{p}_m(\tilde{y}|x), \tag{28}$$

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.

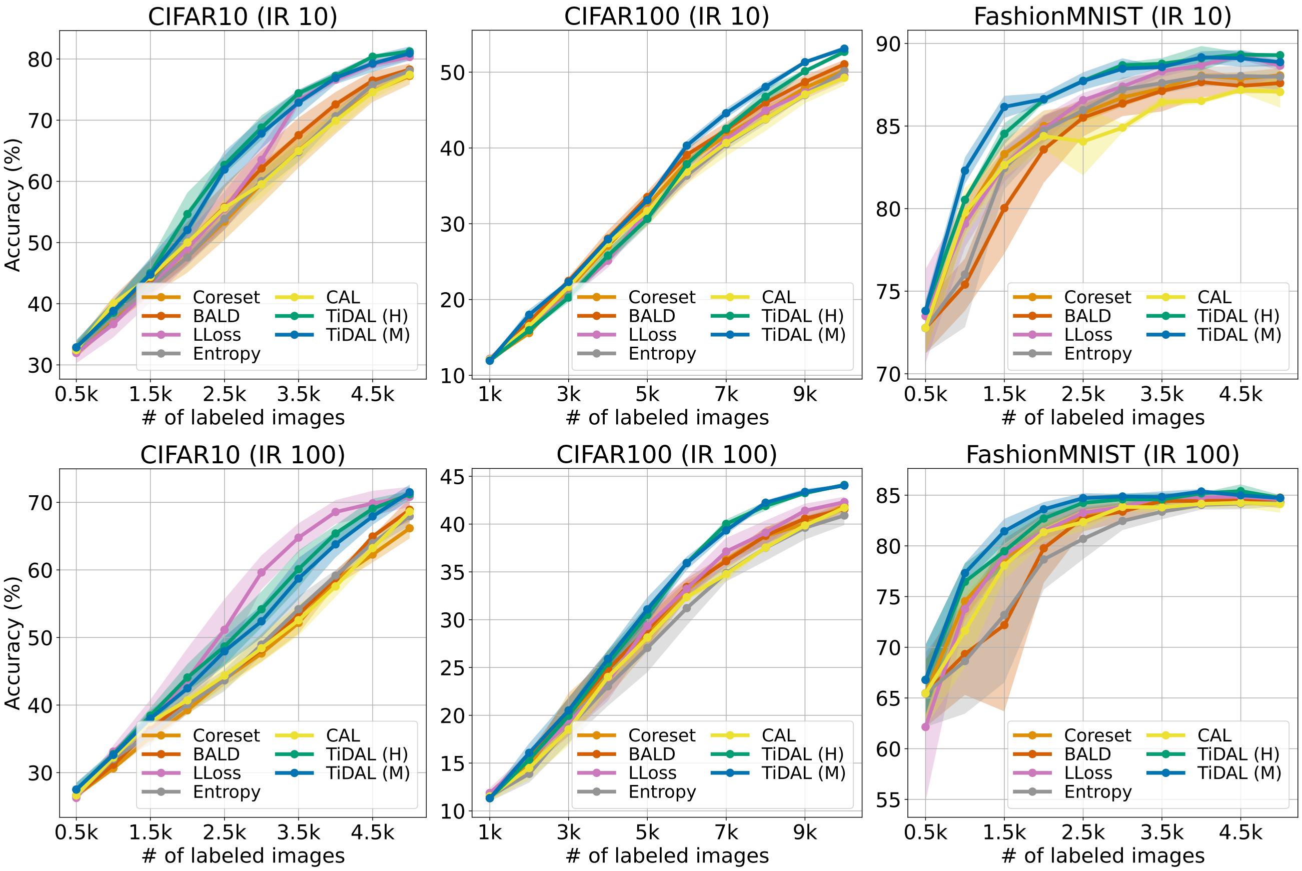


Figure 11: Averaged absolute accuracy improvement curves and their 95% confidence interval (shaded) of AL methods over the number of labeled samples on synthetically imbalanced datasets. We use the imbalance ratio (IR) of 10 and 100 on CIFAR10, CIFAR100, and FashionMNIST.

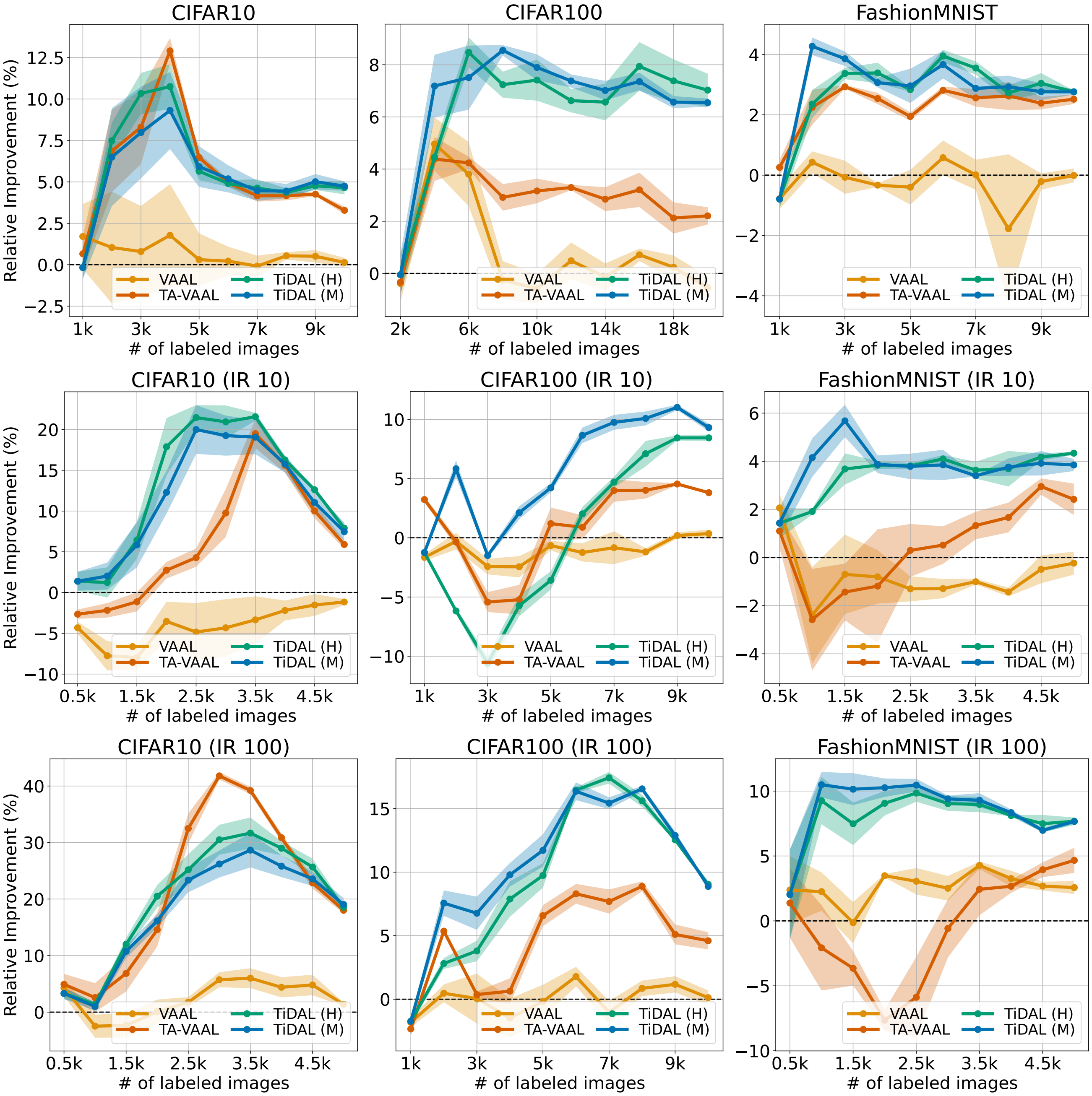


Figure 12: Averaged relative accuracy improvement curves and their 95% confidence interval (shaded) of AL methods over the number of labeled samples on balanced and synthetically imbalanced datasets. We use the imbalance ratio (IR) of 10 and 100 on CIFAR10, CIFAR100, and FashionMNIST to synthetically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

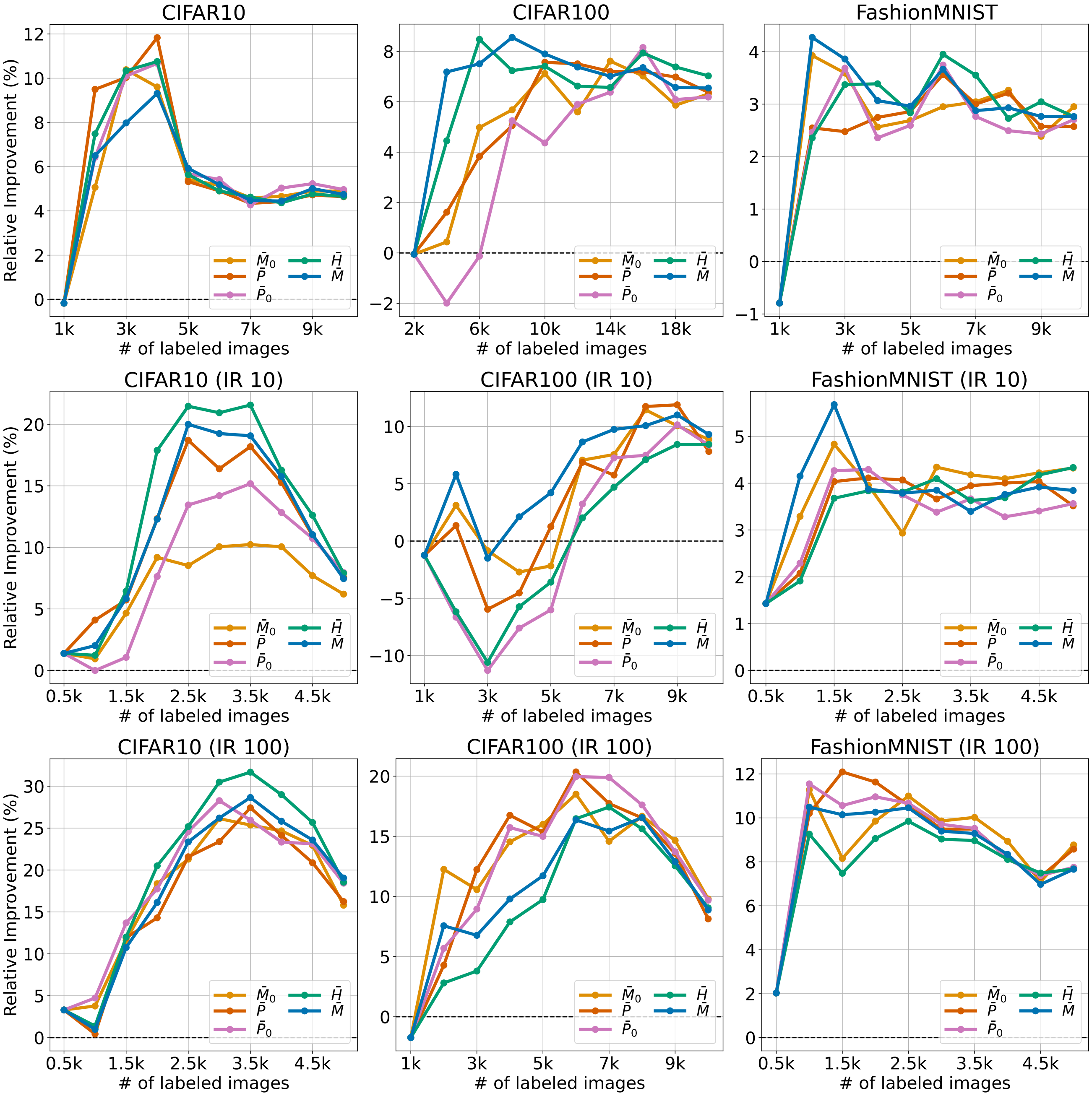


Figure 13: Averaged relative accuracy improvement curves of different uncertainty estimation strategies over the number of labeled samples on balanced and synthetically imbalanced datasets. We use the imbalance ratio (IR) of 10 and 100 on CIFAR10, CIFAR100, and FashionMNIST to synthetically imbalance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