Divide-and-Aggregate Learning for Evaluating Performance on Unlabeled Data

Shuyu Miao*, Jian Liu, Lin Zheng, Hong Jin

Ant Group, China
{miaoshuyu.msy, rex.lj, zhenglin.zhenglin, jinhong.jh}@antgroup.com

Abstract

Artificial Intelligence (AI) models have become an integral part of modern society, significantly improving human lives. However, ensuring the reliability and safety of these models is of paramount importance. One critical aspect is the continuous monitoring and verification of model performance to prevent any potential risks. Real-time online evaluation of AI models is necessary to maintain their effectiveness and mitigate any harm caused by performance degradation. The traditional approach to model evaluation involves supervised methods that rely on manual labeling to compare results with model predictions. Unfortunately, this method is not suitable for online model monitoring due to its inherent lag and high cost. While there have been attempts to explore free-label model evaluation, these approaches often consider only the global features of the entire dataset. Additionally, they can only perform model evaluation based on a single dimension of model confidence or features. In this paper, we propose a novel approach called *Divide-and-Aggregate Learning* (DAL) for unsupervised model evaluation. Our method addresses the limitations of previous approaches by dividing the output of the model into buckets, capturing local information of the distribution. We then aggregate this local information to obtain global information and further represent the relationship between the distribution and model performance. Importantly, our method can simultaneously handle the confidence distribution and feature distribution of the model output. Extensive experiments have been conducted to demonstrate the effectiveness of our DAL model. The results show that our approach outperforms previous methods on four widely used datasets. We will make our source code publicly available.

Introduction

Artificial intelligence (AI) models have become integral to modern life, offering unparalleled convenience in numerous applications. However, given the profound significance of these models, any issues or challenges that emerge can have far-reaching consequences. For instance, if a model's performance significantly deteriorates, it can drastically diminish its value and lead to serious economic losses. As such, effective online model evaluation has emerged as a critical topic in the applications of AI models.

*The corresponding author.

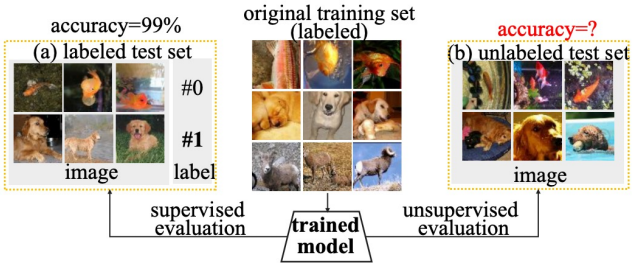

Figure 1: The illustration of label-free model evaluation (AutoEval). The left is the traditional supervised model evaluation with human labels, and the right is the unsupervised model evaluation.

Model evaluation has traditionally relied on a test dataset consisting of test samples and human-labeled ground truth labels. However, this approach can be problematic in real-world settings due to its complexity, expense, and lag. To circumvent these issues, label-free model evaluation, or Automatic model Evaluation (AutoEval) (Deng and Zheng 2021), has emerged as a promising alternative. This method involves asking the model to autonomously evaluate its performance on a dataset without relying on manual labels. Figure 1 visually demonstrates the contrast between supervised model evaluation (He et al. 2016; Deng et al. 2009; Miao et al. 2022; Lin et al. 2014; Everingham et al. 2006), which uses human labels for evaluation, and AutoEval, which does not require human intervention. The lack of explicit labels presents the most significant challenge in AutoEval, as the models must identify intrinsic patterns and structures in the test data without relying on ground truth labels. *Please note that AutoEval is committed to predicting the performance of trained models on unlabeled test data, rather than improving the original performance of trained models.*

Recent studies have demonstrated the promising performance of label-free model evaluation (Garg et al. 2022; Guillory et al. 2021; Deng and Zheng 2021; Deng, Gould, and Zheng 2021; Jiang et al. 2021; Chen et al. 2021; Miao et al. 2023). Some of these studies leverage feature calibration of distribution shifts on the entire unlabeled test data to provide consistent estimates and predict the model's performance (Deng and Zheng 2021; Miao et al. 2023;

Deng, Gould, and Zheng 2021). However, these approaches consider the global features of the model feature distribution on test data and disregard the individual heterogeneity of the sample. Others models (Garg et al. 2022; Guillory et al. 2021; Hendrycks and Gimpel 2017) use confidence calibration of the models to reduce the impact of unexpected changes in performance on unlabeled test data. However, confidence calibration in unlabeled data may result in poor estimates because an individual threshold on the entire dataset, whether fixed, learned, or averaged, cannot represent a relatively complete scoring distribution.

Therefore, the two main challenges in this work are: (i) devising a method that can model the effectiveness of the model based on feature output and confidence output, which are common outputs of actual deployment models; and (ii) designing a modeling approach that incorporates more local information, given that samples exhibit specificity and not just global dataset information. To address the aforementioned challenges, we propose a novel approach called *Divide-and-Aggregate Learning* (DAL) for achieving unsupervised model evaluation. DAL consists of two stages: *Divide Learning* and *Aggregate Learning*. In the *Divide Learning* stage, we divide the model's output distribution into buckets and calculate the distribution shifts to capture local information. In the subsequent *Aggregate Learning* stage, we aggregate the local distribution shifts to obtain global shifts and further establish their relationship with accuracy. Furthermore, DAL demonstrates competence in evaluating both the feature output and confidence output of the model. We have conducted extensive experiments to validate the efficacy of our proposed approach. Our model outperforms multiple trained models, showcases differentiation in distribution across unlabeled test data, and exhibits superior performance across diverse datasets. The experimental results also indicate that unsupervised model evaluation methods align closely with the outcomes of supervised model evaluation, with minimal deviations. This suggests the potential for truly automated online model evaluation.

In summary, our proposed approach offers several contributions, including:

- Introducing a novel approach called *Divide-and-Aggregate Learning* (DAL) for unsupervised model evaluation that aligns closely with supervised model evaluation outcomes, with minimal deviations.
- Developing a method that can model both the features and confidence output of the model simultaneously, and incorporates local sample information in addition to global dataset information.
- Conducting extensive experiments that demonstrate the superior performance of our approach over multiple trained models and widely used datasets, showcasing state-of-the-art results.

Related Work

Our work covers several related areas of research that have undergone significant progress in recent years.

Label-free Model Evaluation

Label-free model evaluation is a process that aims to predict the accuracy of an unseen test set when the ground truth is not accessible (Guillory et al. 2021; Deng, Gould, and Zheng 2021; Deng and Zheng 2021; Chen et al. 2021; Jiang et al. 2021; Hendrycks and Gimpel 2017; Garg et al. 2022; Yu et al. 2022; Miao et al. 2023). Deng et al. (Deng and Zheng 2021) constructed a meta-dataset comprised of datasets generated from the original images, used the Fréchet distance (Dowson and Landau 1982) to measure the domain gap of various data, and trained a regression model to predict model performance. Miao et al. (Miao et al. 2023) introduced a k-means clustering-based feature consistency alignment (KCFCA) to handle distribution shifts of various datasets. Confidence-based calibration (Guillory et al. 2021; Garg et al. 2022) was also explored to successfully estimate the trained model across several model architectures and types of distribution shifts. The models (Jiang et al. 2021; Yu et al. 2022) assessed model performance through calibration of the retraining model. However, these methods only consider global distributions from features or confidence as a whole for corrective prediction, ignoring the variability of local distributions. In contrast, our approach explores a model evaluation method that unifies feature-based and confidence-based autoeval approaches considering with local sample information.

Out-Of-Distribution Detection

The goal of Out-Of-Distribution (OOD) detection is to identify samples that do not conform to the distribution of the training data. Hendrycks et al. (Hendrycks, Mazeika, and Dietterich 2019) presented Outlier Exposure (OE), which improved deep anomaly detection by training anomaly detectors against an auxiliary dataset of outliers. A likelihood ratio method (Ren et al. 2019) was proposed for deep generative models, which effectively corrected for population-level background statistics. MOS (Huang and Li 2021) decomposed the large semantic space into smaller groups with similar concepts, allowing for simplification of the decision boundaries between in- vs. out-of-distribution data for effective OOD detection. Liang et al. (Liang, Li, and Srikant 2017) introduced ODIN to detect out-of-distribution images in neural networks based on the observation that using temperature scaling and adding small perturbations to the input could separate softmax score distributions between in- and out-of-distribution images. Unlike OOD detection task, the autoeval task requires further prediction of the accuracy of the unlabeled dataset instead of detecting unseen examples.

Our Method

In the realm of practical application, artificial intelligence (AI) models undergo optimization and training on raw data. This process results in a trained model, while the raw data is referred to as the training set and validation set. Following deployment, it is essential to evaluate the trained model's performance online with current data, which is known as the test set. The objective of our academic study is to propose a methodology for evaluating the trained model performance

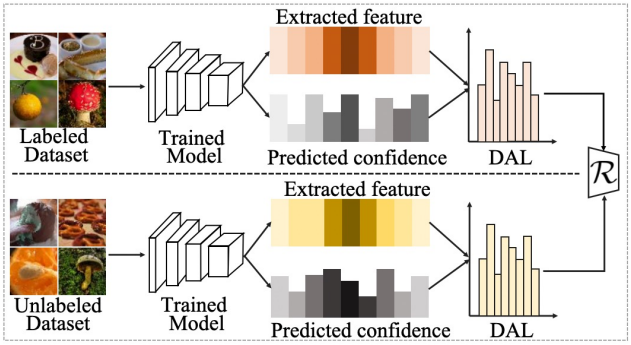

Figure 2: The illustration of Divide-and-Aggregate Learning (DAL). Initially, DAL trains a regression model $\mathcal{R}$ using training data to capture the relationship between distribution shift and model accuracy. Subsequently, the accuracy of the model is determined by feeding unlabeled test sets as inputs into the regression model $\mathcal{R}$. This process enables the estimation of model accuracy without the need for labeled data.

on unlabeled test sets. In the following sections, we will provide a comprehensive description of our proposed approach.

Problem Definition

We commence by defining the raw data $\mathcal{D}_r = (\mathcal{D}_t, \mathcal{D}_v)$, where $\mathcal{D}_t$ represents the labeled training data and $\mathcal{D}_v$ denotes the labeled validation data. Specifically, $\mathcal{D}_t = (\{x_i, y_i\})_{i=1}^{N_t}$, where x_i represents a training sample, y_i denotes its corresponding label, and N_t designates the total number of training samples. Similarly, $\mathcal{D}_v = \{x_i, y_i\})_{i=1}^{N_v}$, where N_v denotes the total number of validation samples. Consequently, we obtain a highly trained model $\mathcal{M}$ based on raw data. The ultimate objective is to evaluate the accuracy of the trained model on an unlabeled test set $\mathcal{D}_u = (\{x_i\})_{i=1}^{N_u}$ comprising N_u test samples. In the supervised model evaluation setting, if $\mathcal{D}_u$ is labeled with y_i, the accuracy can be easily calculated using the following equation:

$$acc = \frac{\sum_{i=1}^{N_u} \mathbb{I}[\mathcal{M}(x_i) == y_i]}{N_u}, \tag{1}$$

where $\mathbb{I}(x)$ represents the indicator function:

$$\mathbb{I}(x) = \begin{cases} 1, & if\ x\ is\ True \\ 0, & if\ x\ is\ False. \end{cases} \tag{2}$$

However, this method fails when the labels are unavailable. Hence, given the raw data $\mathcal{D}_r$ and the trained model $\mathcal{M}$, our aim is to design a regression model $\mathcal{R}$ for label-free model evaluation, which outputs an overall accuracy. The process can be expressed as

$$acc = \mathcal{R}(\mathcal{D}_u | (\mathcal{D}_r, \mathcal{M})). \tag{3}$$

From Eq. (3), it is clear that the accuracy of the unlabeled test set can be directly predicted by learning a model $\mathcal{R}$. However, learning $\mathcal{R}$ through unsupervised learning paradigms is not straightforward. Therefore, we adopt the approach proposed in (Deng and Zheng 2021) to convert the unsupervised problem into a supervised learning problem. In particular, we perform transformations based on the raw data $\mathcal{D}_r$ to obtain meta data $\mathcal{D}_m = \{D_i = (S_{xi}, S_{yi})\}_{i=1}^{T}$ following (Deng and Zheng 2021; Miao et al. 2023). Here, D_i represents the i-th training sample dataset, S_{xi} denotes the corresponding training datasets, S_{yi} represents the corresponding labels, and T denotes the total number of sample datasets. This means that the accuracy of the model on these sample datasets is known. Subsequently, we can learn a regression model $\mathcal{R}$ based on the T pairs of training data and corresponding labels. Thus, the completely unsupervised task is essentially converted into a supervised task. The primary challenge is to construct effective input features $\mathcal{F}$ for the regression model using these datasets. Furthermore, Eq. (3) can be simplified, as shown in Eq. (4):

$$acc = \mathcal{R}(\mathcal{F}). \tag{4}$$

Next, we will introduce how to construct the feature $\mathcal{F}$.

Divide-and-Aggregate Learning

We assume that the output distribution of the trained model for the raw data set and sample data set are $\mathcal{U}_r = \{u_r^1, u_r^2, \cdots, u_r^n\}$ and $\mathcal{U}_s = \{u_s^1, u_s^2, \cdots, u_s^m\}$ with n and m samples, respectively. Our goal is to calculate the distribution shift between two output distributions, further obtain T pair distribution shifts and the corresponding model accuracy, then construct the regression model between the distribution shift and the accuracy of the trained model. That is, the distribution shift is used as the feature, and the known accuracy is used as the ground truth to train the regression model in Eq. (4). Inspired by information theory (Shannon 2001; Kullback and Leibler 1951; Goodfellow, Bengio, and Courville 2016; Yurdakul 2018) we propose **Divide-and-Aggregate Learning (DAL)** to measure the distribution shift.

Intuitively, the Kullback-Leibler (KL) divergence is a commonly used method to measure the difference between two distributions, denoted as $p(x)$ and $q(x)$ (Shannon 2001; Kullback and Leibler 1951). The definition of KL divergence is as follows:

$$\begin{aligned} KL(p||q) &= -\sum_x p(x) log \frac{1}{p(x)} + \sum_x p(x) log \frac{1}{q(x)} \\ &= \sum_x p(x) log \frac{p(x)}{q(x)}, \end{aligned} \tag{5}$$

$$\begin{aligned} KL(q||p) &= -\sum_x q(x) log \frac{1}{q(x)} + \sum_x q(x) log \frac{1}{p(x)} \\ &= \sum_x q(x) log \frac{q(x)}{p(x)}. \end{aligned} \tag{6}$$

Observing Eqs. (5) and (6), two essential characteristics of KL divergence become apparent. Firstly, KL divergence requires the same number of samples for both distributions, i.e. $NUM(p(x)) = NUM(q(x))$. Secondly, KL divergence is asymmetric, i.e. $KL(p||q) \neq KL(q||p)$. These characteristics limit its application in our auto-evaluation method.

Divide Learning. Due to the varying number of elements in the distribution, direct calculation of the distribution distance is challenging. The proposed **'Divide Learning'** method outlines a process for transforming two distributions, $\mathcal{U}_r$ and $\mathcal{U}_s$, into fixed information buckets to enable distribution shift calculation. To achieve this, the distributions are divided into a fixed number of k buckets, denoted as $\mathcal{B}_1, \mathcal{B}_2, \cdots, \mathcal{B}_k$. Specifically, the distributions are sorted to generate new distributions $\mathcal{U}'_r$ and $\mathcal{U}'_s$. Equal-frequency or equal-width binning is then performed on these distributions, resulting in k buckets with bin quantiles of $\{[b_0, b_1], [b_1, b_2], \cdots [b_{k-1}, b_k]\}$ based on the sorted $\mathcal{U}'_r$ distributions. For equal-frequency binning, the quantity of each bucket is the same, i.e., $NUM(\mathcal{B}_1) = NUM(\mathcal{B}_2) = NUM(\cdots) = NUM(\mathcal{B}_k)$. For equal-width binning, the length of the numerical interval of each bucket is the same, i.e., $LEN(\mathcal{B}_1) = LEN(\mathcal{B}_2) = LEN(\cdots) = LEN(\mathcal{B}_k)$. The $\mathcal{U}_r$ and $\mathcal{U}_s$ distributions are then adjusted to k buckets based on the bin quantiles, yielding $\mathcal{U}'_r = \{\mathcal{B}^r_1, \mathcal{B}^r_2, \cdots, \mathcal{B}^r_k\}$ and $\mathcal{U}'_s = \{\mathcal{B}^s_1, \mathcal{B}^s_2, \cdots, \mathcal{B}^s_k\}$. Each bucket is then converted into a new distribution element by calculating the proportion of the bucket data to the total data, expressed as $u'^k_r = \frac{NUM(\mathcal{B}^r_k)}{NUM(\mathcal{U}_r)}$ and $u'^k_s = \frac{NUM(\mathcal{B}^s_k)}{NUM(\mathcal{U}'_s)}$. The new $\mathcal{U}_r$ and $\mathcal{U}_s$ distributions are then processed as $\mathcal{U}'_r = \{u'^1_r, u'^2_r, \cdots, u'^k_r\}$ and $\mathcal{U}'_s = \{u'^1_s, u'^2_s, \cdots, u'^k_s\}$ with the same sample numbers. Overall, our *Divide Learning* method overcomes the issue arising from different numbers of elements in the distributions and provides a method for comparing distribution shifts while considering local sample information.

Aggregate Learning. It is crucial to acknowledge that the distributions $\mathcal{U}_r$ and $\mathcal{U}_s$ do not have an explicit relative relationship. As a result, the distribution shift between them is symmetrical, which contradicts the asymmetry of KL divergence. Hence, directly calculating the KL divergence of the two distributions in our approach is not reasonable. To achieve distance symmetry, our **'Aggregate Learning'** method leverages the property that $KL(p||q) + KL(q||p) = KL(q||p) + KL(p||q)$ to represent the distribution shifts. Consequently, we define the shifts between the two distributions $\mathcal{U}_r$ and $\mathcal{U}_s$ as follows:

$$\begin{aligned} Dis(\mathcal{U}_r, \mathcal{U}_s) &= KL(\mathcal{U}_r||\mathcal{U}_s) + KL(\mathcal{U}_s||\mathcal{U}_r) \\ &= \sum \mathcal{U}_r log\frac{\mathcal{U}_r}{\mathcal{U}_s} + \sum \mathcal{U}_s log\frac{\mathcal{U}_s}{\mathcal{U}_r} \\ &\Rightarrow \sum \mathcal{U}'_r log\frac{\mathcal{U}'_r}{\mathcal{U}'_s} + \sum \mathcal{U}'_s log\frac{\mathcal{U}'_s}{\mathcal{U}'_r} \\ &= \sum (\mathcal{U}'_r - \mathcal{U}'_s) log\frac{\mathcal{U}'_r}{\mathcal{U}'_s}. \end{aligned} \tag{7}$$

With the establishment of distribution shifts as features, we proceed to learn the regression model using Eq. (4). This enables us to predict the accuracy of an unlabeled test set.

DAL-based Autoeval Models

In the training phase of our study, we propose the construction of a regression model that takes into account both the distribution shift and accuracy of the T-pair trained model output on both the sample dataset and raw dataset. This approach ensures greater precision and reliability in our model building process. During the testing phase, we calculate the distribution shift of the model's output on the unlabeled test set and raw data set using the same method as in the regression model input. By doing so, we can directly predict the accuracy of the trained model on the unlabeled test set. This process provides us with a robust methodology for evaluating and validating our model's performance in real-world scenarios. Additionally, we propose the use of DAL-based autoeval models which can be constructed with ease using the feature and confidence output of trained models.

Feature-based DAL Models. Most existing feature-based autoeval methods in the literature focus solely on global dataset information. In contrast, our approach incorporates DAL to manipulate features and incorporate local sample information. Specifically, we define the sample features extracted by the trained model $\mathcal{M}$ for the training sample dataset S_{xi} of the meta data set as $\mathcal{F}_i \in R^{N_v \times N}$. We first normalize the features at the first dimension and then perform feature dimensionality reduction through homogenization at the second dimension using Eqs. (8) and (9).

$$\mathcal{F}_i = \frac{\mathcal{F}_i - MEAN([\mathcal{F}_i]^0)}{STD([\mathcal{F}_i]^0)}, \tag{8}$$

$$\mathcal{F}_i = MEAN([\mathcal{F}_i]^1) + MAX([\mathcal{F}_i]^1) + MIN([\mathcal{F}_i]^1), \tag{9}$$

Here, $MEAN$, STD, MAX, and MIN represent the mean value, standard deviation, maximum value, and minimum value, respectively. Additionally, $[\cdot]^0$ or $[\cdot]^1$ refers to the first or second dimension. Consequently, we obtain $\mathcal{F}_i \in R^{N_v}$. Similarly, we define the source features extracted by the trained model $\mathcal{M}$ on the labeled training data $\mathcal{D}_t$ as $\mathcal{F}_t \in R^{N_t}$ using Eqs. (10) and (11).

$$\mathcal{F}_t = \frac{\mathcal{F}_t - MEAN([\mathcal{F}_t]^0)}{STD([\mathcal{F}_t]^0)}, \tag{10}$$

$$\mathcal{F}_t = MEAN([\mathcal{F}_t]^1) + MAX([\mathcal{F}_t]^1) + MIN([\mathcal{F}_t]^1) \tag{11}$$

Drawing from Eq. (7), we consider the distance $Dis(\mathcal{F}_i, \mathcal{F}_t)$ between the sample features and source features as an input feature, denoted as $\mathcal{F}$, for training the autoeval model in Eq. (4).

Confidence-based DAL Models. Most of the confidence-based calibration methods focus on the overall distribution of predicted confidences at a global dataset level. In contrast, we employ DAL to build a confidence-based DAL model that captures the unique characteristics of local sample information. We define the sample confidences, predicted by the trained model $\mathcal{M}$ for the training sample dataset S_{xi} of the meta data set, as $C_i \in R^{N_v}$ and the source confidences, predicted by the trained model $\mathcal{M}$ for the source validation data D_v, as $C_v \in R^{N_v}$. Using DAL and Eq. (4), we take the distribution shift between the sample confidence and source confidence as an input feature $\mathcal{F}$ by Eq. (7) for $\mathcal{F} = Dis(C_i, C_t)$ training the autoeval model.

In summary, we unify the feature-based and confidence-based autoeval algorithms under the DAL framework, enabling the capture of local sample information.

Pretrained	Model	RMSE ↓			
		CIFAR-10.1	CIFAR-10.1-C	CIFAR-10-F	**Overall**
ResNet-56	ConfScore (Hendrycks and Gimpel 2017)	2.190	9.743	2.676	6.985
	Entropy (Guillory et al. 2021)	2.424	10.300	2.913	7.402
	Rotation (Deng, Gould, and Zheng 2021)	7.285	6.386	7.763	7.129
	FID (Deng and Zheng 2021)	7.517	5.145	4.662	4.985
	ATC (Garg et al. 2022)	11.428	5.964	8.960	7.766
	KCFCA (Miao et al. 2023)	9.979	8.828	3.905	6.766
	DAL-c(ours)	3.868	9.951	**2.391**	5.625
	DAL-f (ours)	**1.948**	**2.854**	4.191	**3.570**
RepVGG-A0	ConfScore (Hendrycks and Gimpel 2017)	5.470	12.004	3.709	8.722
	Entropy (Guillory et al. 2021)	5.997	12.645	**3.419**	9.093
	Rotation (Deng, Gould, and Zheng 2021)	16.726	17.137	8.105	13.391
	FID (Deng and Zheng 2021)	10.718	6.318	5.245	5.966
	ATC (Garg et al. 2022)	15.168	8.050	7.694	8.132
	KCFCA (Miao et al. 2023)	11.876	7.087	4.797	6.236
	DAL-c (ours)	10.676	9.449	6.198	8.029
	DAL-f (ours)	**0.015**	**4.395**	4.839	**4.570**

Table 1: Comparison with state-of-the-art autoeval models based on the same trained model on the CIFIR-10 dataset with RMSE. "DAL-c" is our confidence-based DAL model, and "DAL-f" is our feature-based DAL model. "↓" means the smaller the numerical value, the better the performance. The bold number is optimal performance.

Experiments

Datasets

To verify the performance of the model, we use the known datasets to conduct experiments following the same dataset transformations as previous works (Deng and Zheng 2021; Deng, Gould, and Zheng 2021; Miao et al. 2023).

CIFAR-10. The original source dataset, CIFAR-10, comprises 60,000 color images across 10 classes, with each class containing 6,000 samples. Following the same transformation strategy proposed by (Deng and Zheng 2021), the training meta-dataset consists of 1,000 transformed sample samples from the original CIFAR-10 val set. The test set is composed of CIFAR-10.1 (Recht et al. 2018; Torralba, Fergus, and Freeman 2008), CIFAR-10.1-C (Hendrycks and Dietterich 2019) (add corruptions to CIFAR-10.1 dataset), and CIFAR-10-F (real-world images collected from Flicker.)

MNIST. The original source dataset for handwritten digits, MNIST, consists of a training set of 60,000 examples and a validation set of 10,000 examples across 10 classes. Following (Deng and Zheng 2021), the training meta-dataset and the test set consist of 1,000 and 200 transformed sample datasets from the original MNIST val set with different transformation styles.

TinyImageNet. The original source dataset, TinyImageNet, is a subset of the ImageNet (Russakovsky et al. 2015) dataset and comprises 200 classes. Following (Deng and Zheng 2021), the training meta-dataset and the test set consist of 1,000 and 200 transformed sample datasets from the original TinyImageNet val set with various transformation styles.

ImageNet. The original source dataset ImageNet (Russakovsky et al. 2015) is a large and widely used dataset with 1000 classes. Following (Deng and Zheng 2021), the training meta-dataset and the test set consist of 500 and 100 transformed sample datasets from the original ImageNet val set with various transformation styles.

Implementation Details

Training Details. In all our experiments, we begin by training the original model, denoted as $\mathcal{M}$, on the original source dataset. This initial training step serves only to simulate the performance of the trained model, and our primary objective is to evaluate its performance on the unlabeled test dataset, rather than optimizing the performance of $\mathcal{M}$. To ensure a fair comparison, we employ the same trained models across all benchmarks, including ResNet-56 (He et al. 2016), RepVGG-A0 (Ding et al. 2021), ResNet-18 (He et al. 2016), AlexNet (Krizhevsky, Sutskever, and Hinton 2017). Additionally, we utilize the same LinearRegression model as the regression model $\mathcal{R}$ following (Deng and Zheng 2021; Miao et al. 2023). We will make the source code public[1].

Metrics. Consistent with previous studies (Deng and Zheng 2021; Miao et al. 2023), we employ the widely used root-mean-square error (RMSE) to measure the predicted accuracy and ground truth accuracy, $RMSE = \sqrt{\frac{\sum_{t=1}^{n}(\widehat{y}_t - y_t)^2}{n}}$, as the evaluation metric in all of our experiments.

Overall Performance

Results on CIFIR-10. To assess the performance of our autoeval models on CIFAR-10, we utilize ResNet-56 (He et al. 2016) and RepVGG-A0 (Ding et al. 2021) as our base trained models. The pre-trained weights of these models are

[1]https://github.com/miaoshuyu/dal-pytorch

Model	Source	MNIST		TinyImageNet		ImageNet	
		Trained	RMSE ↓	Trained	RMSE ↓	Trained	RMSE ↓
ConfScore (Hendrycks and Gimpel 2017)	ICLR'17	AlexNet	8.178	ResNet-18	3.603	ResNet-18	2.687
Entropy (Guillory et al. 2021)	ICCV'21	AlexNet	8.604	ResNet-18	3.572	ResNet-18	3.035
FID (Deng and Zheng 2021)	CVPR'21	AlexNet	10.588	ResNet-18	1.458	ResNet-18	2.984
ATC (Garg et al. 2022)	ICLR'22	AlexNet	9.869	ResNet-18	3.453	ResNet-18	3.110
KCFCA (Miao et al. 2023)	CVPRw'23	AlexNet	8.851	ResNet-18	1.352	ResNet-18	2.663
DAL-c (ours)	-	AlexNet	**7.060**	ResNet-18	3.271	ResNet-18	2.990
DAL-f (ours)	-	AlexNet	9.499	ResNet-18	**1.344**	ResNet-18	**2.022**

Table 2: Comparison with state-of-the-art autoeval models based on the same trained model on the MNIST, TinyImageNet, and ImageNet datasets.

obtained from the public repository available at the website[2].

As shown in Table 1, we provide independent results on three test datasets, namely, CIFAR-10.1, CIFAR-10.1-C, and CIFAR-10-F, as well as Overall results. The results presented in Table 1 lead us to the following conclusions: 1) Our methods exhibit the best performance. It is evident that our DAL-based autoeval models outperform all other methods in terms of performance and robustness. Specifically, for the ResNet-56 trained model, our feature-based DAL model achieves the best performance on separate CIFAR-10.1, CIFAR-10.1-C, and Overall datasets. Moreover, our confidence-based DAL model delivers the best performance on the CIFAR-10-F datasets. For the RepVGG-A0 trained model, our DAL-based models yield the best performance on separate CIFAR-10.1, CIFAR-10.1-C, and Overall datasets. 2) Our methods are effective for different trained models. We achieved optimal results for both ResNet-56 and RepVGG-A0, demonstrating the effectiveness of our method for different trained models. 3) Our methods exhibit the greatest robustness. Overall, our experimental results demonstrate that our methods maintain excellent performance for different trained models and metadata sets, thus validating the robustness of our approach.

Results on MNIST. To assess the efficacy of autoeval models on MNIST, we utilize AlexNet as our foundation for training. Initially, we train AlexNet with a specific structure comprising Conv2d layer, Relu layer, Conv2d layer, Relu layer, Max Pooling, Dropout, FC layer, Relu layer, Dropout, and FC layer, on the original source dataset-MNIST. This process involves a learning rate of 1.0 and 100 epochs. The results of the experiment conducted on the unlabeled test set for the autoeval task are tabulated in Table 2.

Upon comparison of the experimental results presented in the table, our DAL-based autoeval model stands out as the leading method. Notably, our approach surpasses the RMSE benchmark by 1.118 in comparison to ConfScore (Hendrycks and Gimpel 2017), 1.544 over Entropy (Guillory et al. 2021), 3.528 over FID (Deng and Zheng 2021), 2.809 over ATC (Garg et al. 2022), and 1.791 over KCFCA (Miao et al. 2023). Our method significantly improves the upper limit of the autoeval task on the MNIST dataset.

[2]https://github.com/chenyaofo/pytorch-cifar-models

Dataset	Method	Model	RMSE ↓
CIFIR-10	equal-frequency	DAL-c	5.306
	equal-width	DAL-c	6.629
	equal-frequency	DAL-f	4.157
	equal-width	DAL-f	5.138
MNIST	equal-frequency	DAL-c	7.060
	equal-width	DAL-c	8.386
	equal-frequency	DAL-f	9.499
	equal-width	DAL-f	9.093
TinyImageNet	equal-frequency	DAL-c	3.271
	equal-width	DAL-c	3.533
	equal-frequency	DAL-f	1.344
	equal-width	DAL-f	1.391

Table 3: Experimental results on equal-frequency or equal-width binning.

Results on TinyImageNet. To validate the effectiveness of autoeval models on TinyImageNet, we utilize ResNet-18 as our base-trained model. Our first step involves training ResNet-18 on the original source dataset-TinyImageNet, using SGD optimizer. This process uses a learning rate of 0.1 and 100 epochs. The results of experiments on the unlabeled test set for the autoeval task are tabulated in Table 2.

Upon review of the table, it is evident that our method outperforms all previous autoeval methods, achieving new state-of-the-art performance. Specifically, our approach surpasses the RMSE benchmark by 2.259 in comparison to ConfScore (Hendrycks and Gimpel 2017), 2.228 over Entropy (Guillory et al. 2021), 0.114 over FID (Deng and Zheng 2021), 2.109 over ATC (Garg et al. 2022), and 0.008 over KCFCA (Miao et al. 2023). This demonstrates that DAL-based autoeval methods hold a competitive advantage.

Results on ImageNet. To validate the effectiveness of autoeval models on ImageNet, we utilize the trained ResNet-18 pretrained by Pytorch[3] as our base-trained model. The results of experiments on the unlabeled test set for the autoeval task are tabulated in Table 2.

Based on the table, it shows that our method achieves the best performance. Specifically, our approach surpasses

[3]https://github.com/pytorch/vision/tree/main/torchvision

No.	Buckets	DAL-c		DAL-f	
		equal-f	equal-w	equal-f	equal-w
1	5	5.258	6.902	4.328	5.069
2	10	5.306	6.629	4.157	5.138
3	20	5.328	6.479	4.146	5.092
4	30	5.365	6.331	4.234	5.068
5	40	5.375	6.253	4.324	5.077
6	50	5.409	6.184	4.371	5.072

Table 4: Experimental results on different quantities of buckets with ResNet-56 trained model based on the CIFIR-10 dataset. "equal-f" means the equal-frequency binning, and "equal-w" means the equal-width binning.

RMSE by 0.665 in comparison to ConfScore (Hendrycks and Gimpel 2017), 0.962 over FID (Deng and Zheng 2021), 1.088 over ATC (Garg et al. 2022), 0.641 and over KCFCA (Miao et al. 2023).

Ablation Study

Experiment on equal-frequency or equal-width binning. Either equal-frequency binning or equal-width binning can be utilized for the bucket binning of unit information. Comprehensive comparative experiments are conducted to verify the effects of these different binning methods. We conduct ablation studies on ResNet-56 trained model for CIFIR-10, AlexNet trained model for MNIST, and ResNet-18 trained model for TinyImageNet in Table 3.

It can be observed from the table that equal-frequency binning outperforms equal-width binning with the settings of DAL-c and DAL-f on CIFIR-10, DAL-c on MNIST, and DAL-c and DAL-f on TinyImageNet. Nonetheless, with the settings of DAL-f on MNIST, equal-width binning performs better than equal-frequency binning. The results of the correlational analysis presented in Table 3 indicate that the DAL model based on equal-frequency binning performs almost consistently better than the DAL model based on equal-width binning. Hence, in our DAL-based autoeval methods, we prioritize utilizing the equal-frequency binning-based DAL model. Moreover, all experiment results in Tables 1 and 2 are obtained using equal-frequency binning.

Experiment on different quantities of buckets. Our DAL involves setting the number of buckets. The number of buckets reflects the local information range of the information distribution. We perform detailed experiments with ResNet-56 trained model based on the CIFIR-10 dataset, and the results are presented in Table 4.

A more in-depth examination of the table indicates that the performance of the autoeval model is not always directly proportional to the number of buckets. For confidence-based DAL models with equal-frequency binning, the higher the number of buckets, the better the performance. However, for confidence-based DAL models with equal-width binning, the results are the opposite. For feature-based DAL models, there is no clear correlation between the number of buckets and the performance, regardless of whether equal-frequency binning or equal-width binning is used. Overall, in our DAL-based autoeval methods, we prioritize adopting 10 buckets. Moreover, all experimental results in Tables 1 and 2 are obtained using 10 buckets.

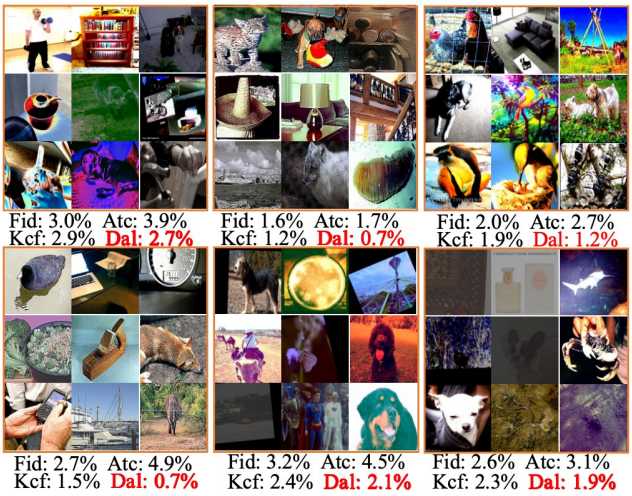

Figure 3: The illustration of the accuracy of trained model on sample datasets. The numbers mean the RMSE bewteen the predicted accuracy and ground truth accuracy on various sample datasets. Our DAL achieves the best performance.

Overall Analysis

In conclusion, our proposed methodology has demonstrated state-of-the-art performance on all four datasets, as evidenced by the rigorous experimentation presented in Tables 1 and 2. Moreover, our approach exhibits excellent robustness across different trained models, further highlighting its effectiveness (Tables 1 and 2). Of particular note, the data presented in the tables and Figure 3 suggests that unsupervised model evaluation can produce results that are comparable to those obtained using supervised model evaluation. This finding is both surprising and significant, as it suggests the possibility of achieving reliable model validation without the need for extensive labeled data sets. We believe that our work can provide valuable insights and ideas for the online validation of AI models.

Conclusion

To enhance the stability and reliability of online AI models, we propose a novel approach called Divide-and-Aggregate Learning (DAL) for unsupervised model evaluation. Unlike existing methods, DAL eliminates the need for manual labeling, enabling real-time online evaluation of model performance. By incorporating local sample information, DAL goes beyond considering only the global dataset information, thus ensuring a more comprehensive assessment. One notable advantage of our approach is its ability to simultaneously handle both the confidence distribution and feature distribution of the model output. We hope that our work contributes to the advancement of safe and effective AI model applications in our daily lives.

References

Chen, M.; Goel, K.; Sohoni, N. S.; Poms, F.; Fatahalian, K.; and Ré, C. 2021. Mandoline: Model evaluation under distribution shift. In *International Conference on Machine Learning*, 1617–1629. PMLR.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, 248–255. Ieee.

Deng, W.; Gould, S.; and Zheng, L. 2021. What does rotation prediction tell us about classifier accuracy under varying testing environments? In *International Conference on Machine Learning*, 2579–2589. PMLR.

Deng, W.; and Zheng, L. 2021. Are labels always necessary for classifier accuracy evaluation? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 15069–15078.

Ding, X.; Zhang, X.; Ma, N.; Han, J.; Ding, G.; and Sun, J. 2021. Repvgg: Making vgg-style convnets great again. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 13733–13742.

Dowson, D.; and Landau, B. 1982. The Fréchet distance between multivariate normal distributions. *Journal of multivariate analysis*, 12(3): 450–455.

Everingham, M.; Zisserman, A.; Williams, C.; and Gool, L. V. 2006. The 2005 PASCAL Visual Object Classes Challenge. *lecture notes in computer science*.

Garg, S.; Balakrishnan, S.; Lipton, Z. C.; Neyshabur, B.; and Sedghi, H. 2022. Leveraging unlabeled data to predict out-of-distribution performance. In *International Conference on Learning Representations*.

Goodfellow, I.; Bengio, Y.; and Courville, A. 2016. *Deep learning*. MIT press.

Guillory, D.; Shankar, V.; Ebrahimi, S.; Darrell, T.; and Schmidt, L. 2021. Predicting with confidence on unseen distributions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 1134–1144.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Hendrycks, D.; and Dietterich, T. 2019. Benchmarking Neural Network Robustness to Common Corruptions and Perturbations. *Proceedings of the International Conference on Learning Representations*.

Hendrycks, D.; and Gimpel, K. 2017. A Baseline for Detecting Misclassified and Out-of-Distribution Examples in Neural Networks. In *International Conference on Learning Representations*.

Hendrycks, D.; Mazeika, M.; and Dietterich, T. 2019. Deep Anomaly Detection with Outlier Exposure. In *International Conference on Learning Representations*.

Huang, R.; and Li, Y. 2021. MOS: Towards Scaling Out-of-Distribution Detection for Large Semantic Space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 8710–8719.

Jiang, Y.; Nagarajan, V.; Baek, C.; and Kolter, J. Z. 2021. Assessing generalization of sgd via disagreement. *arXiv preprint arXiv:2106.13799*.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2017. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6): 84–90.

Kullback, S.; and Leibler, R. A. 1951. On information and sufficiency. *The annals of mathematical statistics*, 22(1): 79–86.

Liang, S.; Li, Y.; and Srikant, R. 2017. Enhancing the reliability of out-of-distribution image detection in neural networks. *arXiv preprint arXiv:1706.02690*.

Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, 740–755. Springer.

Miao, S.; Du, S.; Feng, R.; Zhang, Y.; Li, H.; Liu, T.; Zheng, L.; and Fan, W. 2022. Balanced single-shot object detection using cross-context attention-guided network. *Pattern Recognition*, 122: 108258.

Miao, S.; Zheng, L.; Liu, J.; and Jin, H. 2023. K-Means Clustering Based Feature Consistency Alignment for Label-Free Model Evaluation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 3299–3307.

Recht, B.; Roelofs, R.; Schmidt, L.; and Shankar, V. 2018. Do CIFAR-10 Classifiers Generalize to CIFAR-10? https://arxiv.org/abs/1806.00451.

Ren, J.; Liu, P. J.; Fertig, E.; Snoek, J.; Poplin, R.; Depristo, M.; Dillon, J.; and Lakshminarayanan, B. 2019. Likelihood ratios for out-of-distribution detection. *Advances in neural information processing systems*, 32.

Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. 2015. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115: 211–252.

Shannon, C. E. 2001. A mathematical theory of communication. *ACM SIGMOBILE mobile computing and communications review*, 5(1): 3–55.

Torralba, A.; Fergus, R.; and Freeman, W. T. 2008. 80 Million Tiny Images: A Large Data Set for Nonparametric Object and Scene Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(11): 1958–1970.

Yu, Y.; Yang, Z.; Wei, A.; Ma, Y.; and Steinhardt, J. 2022. Predicting out-of-distribution error with the projection norm. In *International Conference on Machine Learning*, 25721–25746. PMLR.

Yurdakul, B. 2018. *Statistical properties of population stability index*. Western Michigan University.