Scoring Your Prediction on Unseen Data

Yuhao Chen Shen Zhang Renjie Song
MEGVII Technology
yhao.chen0617@gmail.com, zs_howl@163.com songrenjie@megvii.com

Abstract

The performance of deep neural networks can vary substantially when evaluated on datasets different from the training data. This presents a crucial challenge in evaluating models on unseen data without access to labels. Previous methods compute a single model-based indicator at the dataset level and use regression methods to predict performance. To evaluate the model more accurately, we propose a sample-level label-free model evaluation method for better prediction on unseen data, named Scoring Your Prediction (SYP). Specifically, SYP introduces low-level image-based features (e.g., blurriness) to model image quality that is important for classification. We complementarily combine model-based indicators and image-based indicators to enhance sample representation. Additionally, we predict the probability that each sample is correctly classified using a neural network named oracle model. Compared to other existing methods, the proposed method outperforms them on 40 unlabeled datasets transformed by CIFAR-10. Especially, SYP lowers RMSE by 1.83-3.97 for ResNet-56 evaluation and 2.32-9.74 for RepVGG-A0 evaluation compared with latest methods. ***Note that our scheme won the championship on the DataCV Challenge at CVPR 2023.*** *Source code is avaliabe at https://github.com/megvii-research/SYP.*

1. Introduction

The deployment of Deep Neural Networks(DNNs) in the real world faces the challenge of encountering unseen data. The conventional way to measure model performance is to calculate the evaluation metric based on the labeled test set. For example, Top-1 accuracy and Top-5 accuracy are two well-known metrics used in image classification, which evaluate whether the predicted class matches the ground truth. For object detection tasks, the mean average precision metric is used to measure the mean area under the precision-recall curve of each class.

However, these metrics are hard to compute in real-world settings due to the scarcity of labeled datasets. Acquiring these samples, even if successful, may introduce bias into the assessed performance due to their limited coverage of conditions. For instance, Annotating test data for image classification can be a costly endeavor. Even with labels available for every image, it may still be difficult to capture the diversity of real-world factors such as lighting conditions, shadows, and variations in viewpoints. Moreover, Real-world data usually follows a different distribution than the model's training data distribution, violating the IID assumption. This distribution shift is likely to lead to model performance degradation. These findings have raised a critical question regarding how to measure the generalization of models in real-world settings.

Several studies [2, 31, 37] have attempted to estimate the unforeseen performance shifts by proposing generalization bounds derived from network complexity analyses theoretically. However, these methods lack comprehensive empirical evaluation. Recent researches attempt to seek some dataset-level metrics based on the model's prediction. They reveal the effectiveness of distributional distances such as Fréchet distance [11, 47], Maximum Mean Discrepancy (MMD) [6], gap of average entropy [20] and discriminative discrepancy [4, 18]. They build a regression model based on these distances to predict accuracy on unseen distributions.

As mentioned above, current popular methods only focus on the model's output (e.g., prediction entropy) when evaluating the accuracy on unseen data, and failing to consider the impact of image quality. We emphasize here the fact that model always gives a higher prediction entropy when feeding with a heavily distorted input, although it may be correctly classified [8, 39]. Thus, the estimated accuracy will obviously cause drift if just considering the prediction entropy and ignoring the image quality. Moreover, these approaches neglect the specific sample characteristics, using a single indicator value to represent the entire dataset. As each sample may come from a different distribution, treating them as if they were the same distribution may lead to wrongly reflecting the distance between the known and unknown datasets.

In this work, we propose a sample-level label-free model

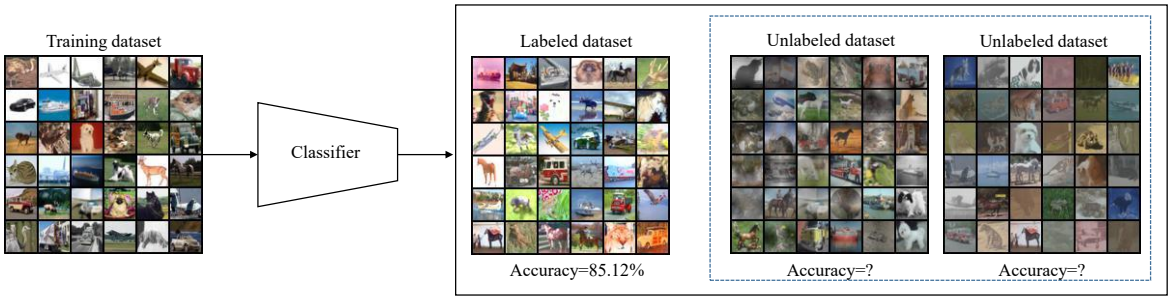


Figure 1. Problem definition. Evaluating model on a labeled test dataset is a commom way in academic research. However, there are scarce labeled test datasets in many real-world scenarios, leaving us unable to use standard evaluation methods. This motivates us to delve into the matter of model evaluation on unlabeled datasets.

evaluation method, named Scoring Your Prediction (SYP). Specifically, SYP introduces low-level image-based indicators including information richness and blurriness as image quality features. Then, SYP takes both model-based indicators and low-level image indicators into consideration to complementarily enhance sample representation. Moreover, we propose a neural network named oracle model to handle indicators and output the classification probability. We design two types of oracle models: one is a Multilayer Perceptron (MLP) network with pure fully connected layers, named ORA-A. We further combine ORA-A with multi-head self-attention [43] to propose another network, named ORA-B.

We conduct extensive experiments to show that the proposed method outperforms other existing methods. Following the DataCV Challenge settings, on 40 unlabeled datasets transformed by CIFAR-10, SYP lowers RMSE by 1.83-3.97 for ResNet-56 evaluation and 2.32-9.74 for RepVGG-A0 evaluation compared with other advanced methods. On 100 unknown test sets, SYP achieved 6.37 RMSE and **won first place in the 1st DataCV Challenge**.

To sum up, the contributions of this work are:

1) We introduce low-level image-based indicators including information richness and blurriness to extract the image quality representation.

2) We complementarily combine both model-based indicators and image-based indicators to enhance the image representation.

3) Two types of oracle models are proposed to predict model performance. The experiments on both validation and test sets demonstrate the effectiveness.

2. Related Work

Our goal is to evaluate the model's performance on unseen and label-free datasets. The topic involves multiple related research fields. This section reviews some research relevant to our method.

Model generalization prediction. Predicting generalization performance of models on in-distribution data using typical machine learning methods has been some excellent work, including [31–33]. Additionally, some work [16, 25] focuses on using unseen unlabeled data to predict generalization errors. [7] proposes using models ensemble are better than a single model when detecting errors and estimating accuracy. [31] points to several factors that can affect a model's generalization performance, including the model's architecture, network size, optimization methods and raining dataset. Furthermore, the prediction consistency of different augmented versions of the same input can also be used for generalization estimation [14, 30, 35, 36]. [3] introduces a novel measure named (effective) prediction depth to represent the prediction difficulty. Differently, we not only focus on a single metric but comprehensively consider the features of dataset, image and model's output.

Out-of-distribution (OoD) detection. OoD [12, 23, 27, 28, 44] is an important research area that aims to detect samples that are different from the training data distribution. Anomaly detection [1], open-set prediction [5] and rejection [9] are OoD research subfieilds. [45] firstly assigns pseudo-labels on the unlabeled test data, and then train a new model based on these pseudo-labels, OoD detection performance can be reflected by the parameter differences. [23] uses the softmax output of the final layer as a confidence score to detect misclassified and OoD examples. [17] believes that OoD sample always produces a high prediction entropy. Differently, our work takes into account multiple statistical characteristics of test datasets, which can effectively improve the accuracy prediction.

Domain adaptation. Domain adaptation [15] aims to address the matter of model deployment in target domains with different statistical properties than training domains. Recent works on domain adaptation have also explored unsupervised domain adaptation, where there is no labeled data available in the target domain [29, 42, 48]. Some schemes have been developed for this field [34, 41, 46].

(a) The backbone of other existing methods.

(b) The backbone of our proposed SYP.

Figure 2. Comparisons between other existing methods and our proposed methods. Other existing methods neglect the specific sample characteristics and image quality features, using a single model-based indicator value to represent the entire dataset, leading to sub-optimal prediction. Our method is sample-level, combining model-based indicators and image-based indicators to predict whether each sample is correctly classified, and then computing the accuracy on the entire dataset.

DAN [29] minimizes a joint objective function that incorporates both the classification loss and the maximum mean discrepancy (MMD) [19] distance, which measures the difference between the source and target domain distributions in the feature space. In this work, we not only consider the feature statistics at the dataset level but also introduce the sample-level features.

3. Method

We first define the problem, then propose a strong pipeline for label-free model evaluation and illustrate the basic procedure, including model-based indicators selection, image-based indicators selection, and oracle model architecture.

3.1. Definition

In this section, we describe the label-free model evaluation problem. As is shown in Fig. 1, consider a classification problem with k classes. We can train a classifier f given a labeled training dataset $L = \{\boldsymbol{x}_i, y_i\}_{i=1}^{n_l}$. The non-linear classifier f maps $\boldsymbol{x}_i$ to a predicted class $\hat{y}_i = f(\boldsymbol{x}_i)$. To evaluate the performance of f, one common way is to test it on another labeled dataset $V = \{\boldsymbol{x}_i, y_i\}_{i=1}^{n_v}$ and calculate the classification accuracy:

$$Acc = \frac{\Sigma_{i=1}^{n_v} \mathbf{1}\{f(\boldsymbol{x}_i) == y_i\}}{n_v}, \tag{1}$$

where $\mathbf{1}\{*\}$ is a characteristic function. However, the above evaluation approach is not feasible in real-world scenarios as the samples are often unlabeled. So we want to extend the model generalization evaluation: estimate the model performance on a new unlabeled dataset $U = \{\boldsymbol{x}_i\}_{i=1}^{n_u}$.

3.2. Model-based Indicators

Given a classifier f and a sample $\boldsymbol{x}_i \in U$, the softmax layer of the classifier outputs a softmax vector $\boldsymbol{s} = f(\boldsymbol{x}_i) \in \mathbb{R}^k$ to predict which class this sample belongs to. The predicted label is decided by the class with the maximal score, i.e. $i^* = \arg\max_i \boldsymbol{s}_i$. In order to judge whether f correctly classifies $\boldsymbol{x}_i$, it is intuitively necessary to make full use of s. Define an indicator function $I : \boldsymbol{s} \to \mathbb{R}$ that maps s to a scalar. We expect the scalar to be correlated to s so that we can determine whether f is correct or not based on the value of the scalar at $\boldsymbol{x}_i$. We introduce several indicator functions: entropy, averaged threshold confidence (ATC), and Fréchet distance (FD).

Entropy. Entropy [20] can be used as an uncertainty measurement of classification correctness. It indicates the confidence of the classifier in its own prediction. If the entropy of $\boldsymbol{s}$ is higher, then the distribution of elements of $\boldsymbol{s}$ is more uniform, indicating the classifier f is more likely to be incorrect. Therefore, we introduce the entropy of $\boldsymbol{s}$ as the first indicator function:

$$I_E(s) = \sum_i s_i log(s_i). \tag{2}$$

Averaged Threshold Confidence (ATC). Given a softmax output $\boldsymbol{s}$, we obtain the predicted label i^* such that $i^* = \arg\max_i \boldsymbol{s}_i$. ATC [17] predicts whether an image is classified correctly given a threshold t:

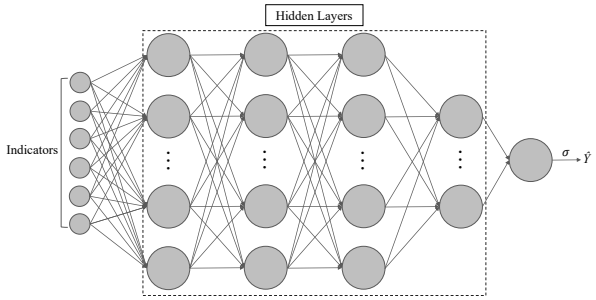
$$I_{ATC}(s) = \mathbf{1}\{s_{i^*} > t\}, \tag{3}$$

note that we can also replace maximum confidence with negative entropy:

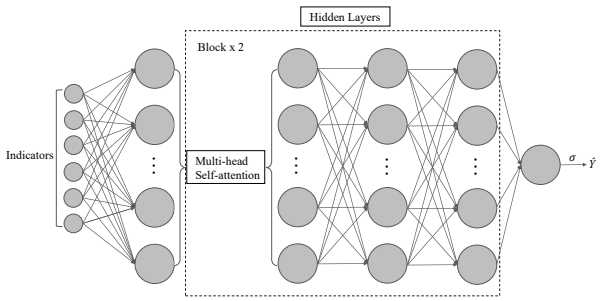
$$I'_{ATC}(s) = \mathbf{1}\{I_{NE}(s) > t\}. \tag{4}$$

Fréchet Distance (FD). Fréchet Distance is used to measure the similarity between curves. Besides that, Fréchet distance can also be used to measure the difference between probability distributions. Suppose two distributions $d_1 \sim \mathcal{D}_1(\mu_1, \Sigma_1)$, $d_2 \sim \mathcal{D}_2(\mu_2, \Sigma_2)$, The Fréchet distance between $\mathcal{D}_1$ and $\mathcal{D}_1$ is defined as:

$$d_F(\mathcal{D}_1, \mathcal{D}_2) = (\inf_{\gamma \sim \Gamma(\mathcal{D}_1, \mathcal{D}_2)} \int \|x - y\|^2 d\gamma(x, y))^{\frac{1}{2}}, \tag{5}$$



(a) The variant of our oracle model: ORA-A. It contains pure fully connected layers.



(b) The variant of our oracle model: ORA-B. It combines two global operation layers: the fully connected layer and the self-attention layer.

Figure 3. Two variants of our oracle model: ORA-A, ORA-B. The oracle model we proposed can effectively use the indicators to estimate the accuracy of the classifier.

where $\Gamma(\mathcal{D}_1, \mathcal{D}_2)$ is the joint probability whose marginals are $\mathcal{D}_1, \mathcal{D}_2$ respectively. It is also known as 2-Wasserstein distance. If $\mathcal{D}_1, \mathcal{D}_2$ are multidimensional Gaussian distributions, FD has a closed formulation:

$$d_F(\mathcal{D}_1, \mathcal{D}_2) = \|\mu_1 - \mu_2\| + Tr(\Sigma_1 + \Sigma_2 - 2(\Sigma_1\Sigma_2)^{\frac{1}{2}}). \quad (6)$$

In our task, we can compute the mean vectors of labeled training dataset L and target unlabeled dataset U, named μ_1', μ_2' respectively, and compute the covariance matrices of labeled training dataset L and target unlabeled dataset U, named Σ_1', Σ_2' respectively. We can estimate the distance between L and U:

$$I_{FD}(L, U) = \|\mu_1' - \mu_2'\| + Tr(\Sigma_1' + \Sigma_2' - 2(\Sigma_1'\Sigma_2')^{\frac{1}{2}}). \quad (7)$$

Note that $I_{FD}(L, U)$ is a dataset-level indicator. For each $\boldsymbol{x}_i \in U$, we append $I_{FD}(L, U)$ to its indicator list.

3.3. Image-based Indicators

Besides model-based indicators, it is crucial to take the low-level image features into consideration. We mainly consider two low-level features: information richness and blurriness. Specifically, when the image consists of more foreground and less background (i.e., high information richness), it will benefit the model classification. On the other hand, when feeding a clear image (i.e., low blurriness), the model must give an output with high confidence to be a correct prediction. For an image $\boldsymbol{x}_i \in U$, we utilize the variance of $\boldsymbol{x}_i$ and the entropy of $\boldsymbol{x}_i$ to measure the information richness and apply Laplacian operator on $\boldsymbol{x}_i$ to evaluate the blurriness.

Variance. The variance of $\boldsymbol{x}_i$ is the variance of all pixels in the entire image:

$$I_{VAR}(\boldsymbol{x}_i) = Var(\boldsymbol{x}_i). \quad (8)$$

Image Entropy. Similar to image variance, The entropy of $\boldsymbol{x}_i$ is the entropy of all pixels in the entire image:

$$I_{IE}(s) = I_E(\boldsymbol{x}_i). \quad (9)$$

Laplacian Operator. The blurry image doesn't have well-defined edges so we can use an edge detection algorithm to compute the blurriness. In this paper, we use Laplacian operator for edge detection. Laplacian operator is a second derivative function designed to measure changes in intensity without being overly sensitive to noise. The output is an image that responds higher in the edge position. In other words, the variance of the Laplacian blurry image will be less as compared to that of the sharp image. The Laplacian operator takes the derivative in both x-axis and y-axis:

$$Laplace(\boldsymbol{x}_i) = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}, \quad (10)$$

then we compute the variance of the Laplacian image as the blurriness indicator:

$$I_B = Var(Laplace(\boldsymbol{x}_i)). \quad (11)$$

In conclusion, we have introduced six indicators given an image $\boldsymbol{x}_i$ and a classifier f. We concatenate them and regard it as the image representation:

$$I = [I_E; I_{ATC}; I_{FD}; I_{VAR}; I_{IE}; I_B]. \quad (12)$$

Then we use I to determine whether $\boldsymbol{x}_i$ can be correctly classified by f.

3.4. Oracle Model

Now our task is finding a mapping function that takes in I and outputs a scalar that represents the correct classification probability $O : I \rightarrow \mathbb{R}$. This is a typical binary classification problem. We use a neural network to solve this problem. When training, we adopt indicator I of a train set image as input and its top-1 prediction result (True or False) as label. As the dimension of indicator I is relatively low, using a large-scale neural network will cause overfitting, we turn to building a tiny neural network. We call it

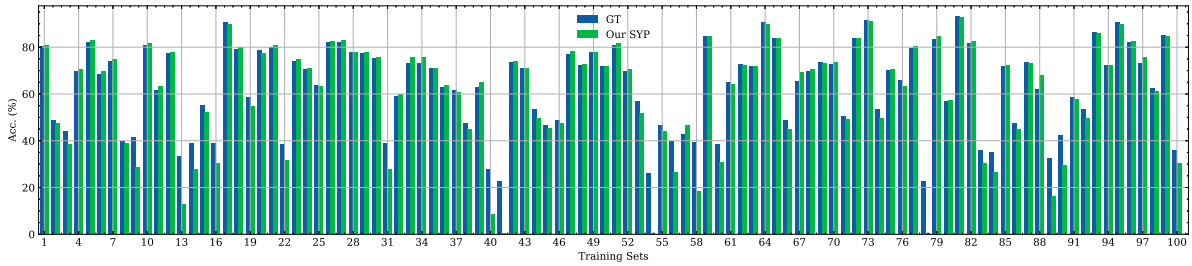


Figure 4. The predicted accuracy of ResNet-56 on the first 100 training datasets made by our ORA-A.

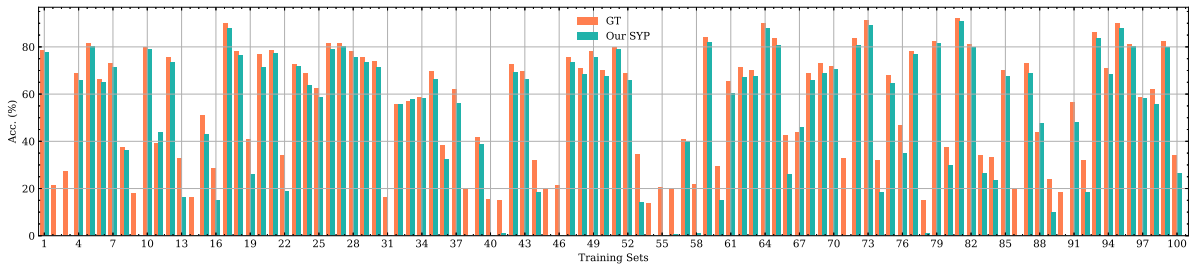


Figure 5. The predicted accuracy of RepVGG-A0 on the first 100 training datasets made by our ORA-A.

oracle model(ORA) as it can tell whether f classifies correctly without labels. We propose two variants of the oracle model. The first one is a simple Multilayer Perceptron (MLP) network with pure fully connected layers, shown in Fig. 3a. We call it ORA-A. It first upsamples I to increase the feature dimension, then gradually reduces the dimension, and finally outputs $p_i \in [0, 1]$. We show ORA-A can significantly improve model evaluation performance in the experiment section.

As is shown in Fig. 3b, we further combine ORA-A with self-attention [43] and propose the second oracle model, named ORA-B. We propose a block that contains a multi-head self-attention layer and two fully-connected layers. For the hidden feature a, we projects it to Q, K, V :

$$Q = fW^Q, K = fW^K, V = fW^V, \tag{13}$$

and compute the single-head self-attention function:

$$Head_i(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V. \tag{14}$$

We concatenate all $head_i$ and project them as the output:

$$O(Q, K, V) = Concat(head_1, ..., head_h)W^O. \tag{15}$$

After that, we feed the output into the fully-connected layers. We stack two blocks and ORA-B outputs the final predicted probability $p_i \in [0, 1]$. As seen below, our experiments verified that ORA-B outperforms other well-known approaches in label-free model evaluation. We also ensemble ORA-A and ORA-B to achieve better performance.

Measurements	Resnet-56	RepVGG-A0
Rotation [10]	7.13	13.39
ConfScore [23]	6.99	8.72
Entropy [20]	7.40	9.09
ATC [17]	7.77	8.13
FD [11]	4.99	5.97
SYP (ORA-A)	**3.16**	**3.65**
SYP (ORA-B)	<u>3.93</u>	<u>4.34</u>

Table 1. Results comparison among different methods. Bold indicates the best result and underline indicates the second-best result. We can see that our SYP achieves lower RMSE than other methods regardless of oracle model structure

4. Experiments

Datasets and evaluation metrics. We conduct contrast experiments on two typical baselines: ResNet-56 [21] and RepVGG-A0 [13] on CIFAR-10 [26] and compare our method with other existing methods. ResNet-56 [21] and RepVGG-A0 [13] are trained on CIFAR-10 [26] training dataset and we load the models' weight to evaluate them. The training dataset of the oracle models consists of 1,000 datasets transformed from the original CIFAR-10 test set, using the transformation strategy proposed by [11]. The validation dataset includes 40 datasets, composed of CIFAR-10.1 [38], CIFAR-10.1-C [22], and CIFAR-10-F. The test set comprises 100 unknown datasets. The image size is 32×32 while the input size is set to 6 for training the oracle models. For quantitative comparisons, we report root mean squared error (RMSE) between ground truth accuracy

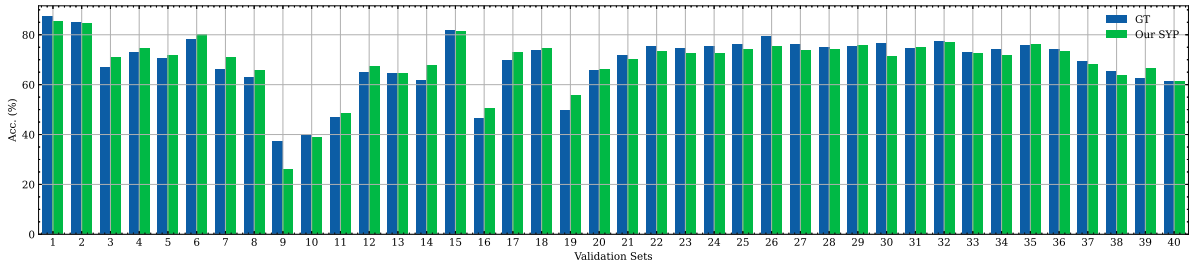


Figure 6. The predicted accuracy of ResNet-56 on each validation dataset made by our ORA-A.

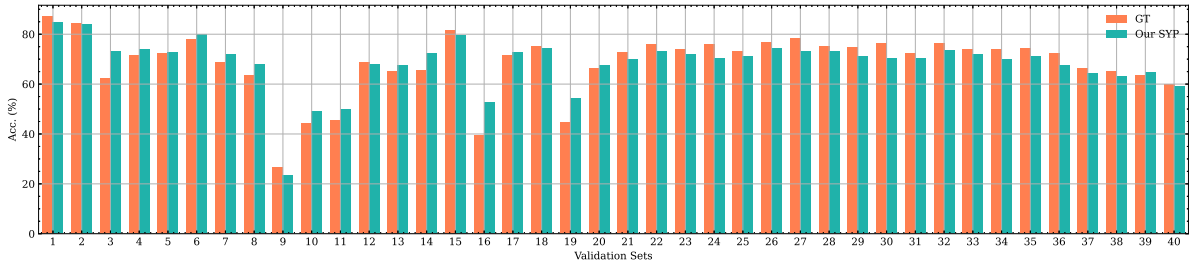


Figure 7. The predicted accuracy of RepVGG-A0 on each validation dataset made by our ORA-A.

Acc and predicted accuracy $\hat{Acc}$ on unlabeled datasets:

$$RMSE = \sqrt{\Sigma_{i=1}^{m}(Acc_i - \hat{Acc}_i)^2}. \tag{16}$$

Experiments Details. We train two types of oracle models introduced before. We adopt the SGD as the optimizer and a linear decay learning rate scheduler. We train the oracle models with 30,000 iterations and the batch size is set to 10,000. The basic learning rate is set to 0.1 for ResNet-56 evaluation and 0.05 for RepVGG-A0 evaluation. Dropout [40] is used to avoid overfitting. The training loss is mean squared error (MSE). We adjust the loss to avoid the class imbalance problem. Specifically, if the accuracy of the predefined model is less than 30% in a dataset, we triple the loss of positive samples. We apply normalization to the indicators. For each indicator I_* in I except I_{FD}, we compute the maximum value $max(I_*)$ and minimum value $min(I_*)$ in U and perform min-max normalization to scale data in the range $[0, 1]$:

$$I'_* = \frac{I_* - min(I_*)}{max(I_*) - min(I_*)}. \tag{17}$$

For I_{FD}, as all samples in U share the same value, performing min-max normalization like other indicators will scale I_{FD} to 1, which makes no sense. Inspired by batch normalization [24], We can compute the maximum value $max(I_{FD})$ and the minimum value $min(I_{FD})$ of all 1,000 training sets and use them on unlabeled dataset:

$$I'_{FD} = \frac{I_{FD} - min(I_{FD})}{max(I_{FD}) - min(I_{FD})}. \tag{18}$$

Methods	Results
Six indicators + ORA-A	6.69
+ Loss adjustment	6.40
+ ORA-B ensemble	6.37

Table 2. Results on test sets.

Indicators	Resnet-56	RepVGG-A0
Entropy	6.11	7.82
+ ATC	6.11	7.83
+ FD	2.92	5.78
+ Image indicators	3.16	3.65

Table 3. The RMSE of adding different indicators. The indicators are sequentially added to the model training.

Note that we train the oracle model on a single NVIDIA RTX 2080Ti.

4.1. Main Results

Comparison to other existing methods. As the Table 1 shows, we compare SYP on 40 validation datasets with other existing methods: rotation prediction (rotation) [10], averaged confidence (ConfScore) [23], entropy [20], averaged threshold confidence (ATC) [17], Fréchet distance (FD) [11]. All other methods are dataset-level and we compute the indicators' values for each dataset and then use linear regression to directly predict the accuracy of the classifier on validation sets. While SYP is sample-level, predicting whether each sample is correctly classified by the oracle model, and then calculating the accuracy of the entire

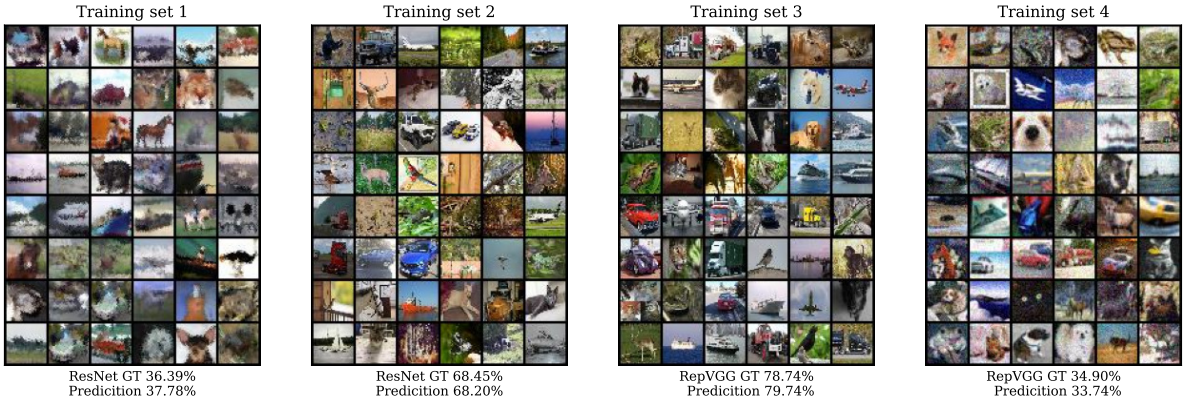


Figure 8. Visualization of different training datasets and the predicted accuracy of our SYP.

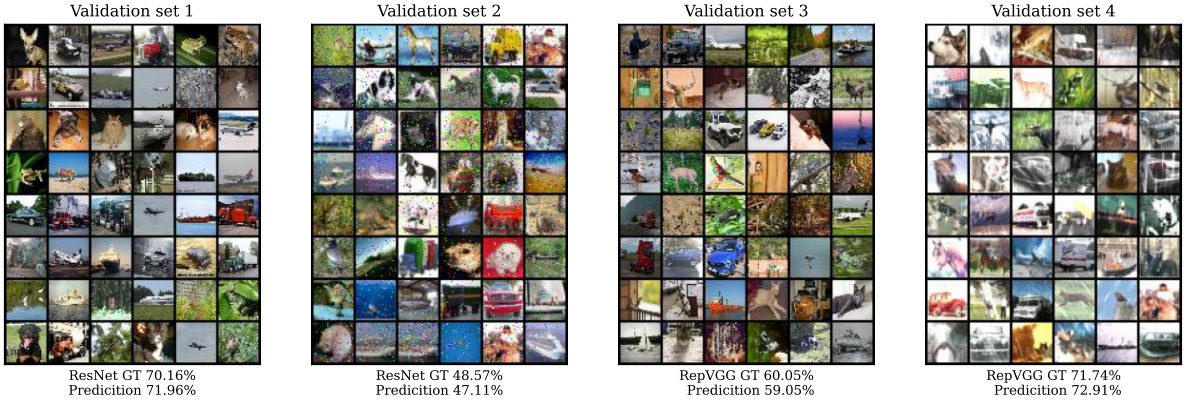


Figure 9. Visualization of different validation datasets and the predicted accuracy of our SYP. The validation datasets are applied to various transformations, such as Gaussian blur, and zoom blur. They are different from the training dataset of the classifier. As can be seen, under different transformations, our method accurately predicts the accuracy of both ResNet-56 and RepVGG-A0.

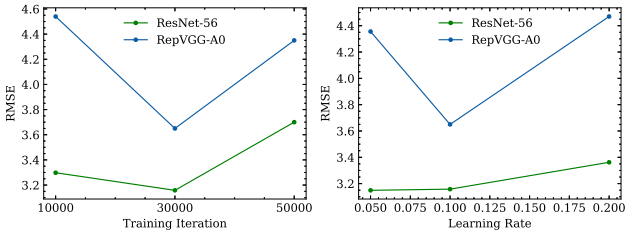


Figure 10. The sensitivity of the oracle model to the training parameters. The left is the influence of training iterations, and the right is the influence of learning rate. Compared with ResNet-56, the oracle model is more sensitive to parameters on RepVGG-A0. But both perform best at 30000 training iterations and lr=0.1.

data set. The empirical results show RMSE of SYP is much lower, outperforming all other existing methods in both ResNet-56 evaluation and RepVGG-A0 evaluation. For example, On ResNet-56, the best RMSE of other methods is 4.99, and SYP can significantly improve it to 3.16. On

Expansion ratio	Resnet-56	RepVGG-A0
r=2	3.22	4.39
r=4	3.21	4.37
r=6	3.58	4.38
r=8	3.16	3.65
r=10	3.17	4.39

Table 4. Comparison among the oracle model with different model capacity. There is a weak correlation between model capability and model performance and our experiment shows the prediction is the best when expansion ratio=8.

	Resnet-56	RepVGG-A0
With normalization	3.16	3.65
Without normalization	32.99	33.96

Table 5. The influence of normalization. It can be clearly seen that the oracle model benefits a lot from normalization. This shows the importance of pre-processing in our pipeline.

RepVGG-A0, SYP can improve RMSE from 5.97 to 3.65. For ResNet-56 and RepVGG-A0, we present the predicted accuracy of our method on the first 100 training datasets in Fig. 4 and Fig. 5 respectively. Note that our model trains well on ResNet data, but struggles with RepVGG data due to the presence of long-tailed low-accuracy datasets. we show the predicted accuracy of our method on each validation dataset in Fig. 6 and Fig. 9 respectively. It can be seen that the predicted accuracy is very close to the ground truth accuracy.

Results on test set. Table 2 shows our performance on the test set, it also demonstrates the effectiveness of our method.

4.2. Ablation Study

We perform some contrast experiments to demonstrate the effectiveness of SYP deeply. Note that we choose ORA-A as the oracle model in this part.

Indicator design We explore the impact of indicators on the performance of the oracle model. We add entropy, ATC, FD, and image-based indicators (image entropy, variance, blurriness) to the input of the oracle model one by one and compute RMSE. Table 3 shows the results. When entropy and ATC are added, the performance of the oracle model is unsatisfactory. After adding FD, the oracle model is greatly improved. Furthermore, by introducing our proposed image-based indicator, the RMSE on RepVGG-A0 decreases from 5.78 to 3.65, indicating that the oracle model is further refined. These results serve as evidence for the efficacy of our image-based indicator.

Sensitivity to training parameters. We explore the impact of two parameters on oracle model training: iteration numbers and learning rate. Fig. 10 shows the results. For ResNet-56 evaluation, the training of the oracle model is relatively not sensitive to parameters. For RepVGG-A0, the selection of different parameters has a greater impact on the oracle model's performance. It is worth noting that the oracle model does not perform better as the number of training iterations increases. Therefore, when facing different tasks, the parameters need to be adjusted appropriately.

Oracle model capacity. We explore the impact of different capacities of the oracle model on performance. We can control the model capacity by adjusting the expansion ratio r. As Table 4 shows, there is no significant difference in accuracy predicted by the oracle model of different capacities. It may be due to our training dataset (the input is only 6-dimensional) being relatively simple. The oracle model works best on the validation set with expansion ratio $r = 8$, which is also the model we submitted on the test set.

Influence of normalization. Table 5 shows that indicator normalization plays a key role in our oracle model training. When there is no normalization, the oracle model cannot converge, and the prediction accuracy is all 1 on all validation datasets. By employing normalization, we can make precise predictions about the classifier's performance on different datasets. Consequently, it is imperative to preprocess the data prior to model training.

5. Conclusion

In this paper, we propose a sample-level label-free model evaluation approach, named SYP. Different with the previous methods, SYP takes extra low-level image-based indicators into account, which can benefit the estimated accuracy on unseen data. In addition, we use oracle models to predict the probability of each sample being classified correctly. We proposed two variants of the oracle model and verified their effectiveness. Extensive experiment results strongly confirm the effectiveness of our method. We believe this work will provide new insights to explore label-free model evaluation.

References

[1] Jerone Andrews, Thomas Tanay, Edward J Morton, and Lewis D Griffin. Transfer representation-learning for anomaly detection. JMLR, 2016. 2

[2] Sanjeev Arora, Rong Ge, Behnam Neyshabur, and Yi Zhang. Stronger generalization bounds for deep nets via a compression approach. In *International Conference on Machine Learning*, pages 254–263. PMLR, 2018. 1

[3] Robert Baldock, Hartmut Maennel, and Behnam Neyshabur. Deep learning through the lens of example difficulty. *Advances in Neural Information Processing Systems*, 34:10876–10889, 2021. 2

[4] Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira. Analysis of representations for domain adaptation. *Advances in neural information processing systems*, 19, 2006. 1

[5] Abhijit Bendale and Terrance Boult. Towards open world recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1893–1902, 2015. 2

[6] Karsten M Borgwardt, Arthur Gretton, Malte J Rasch, Hans-Peter Kriegel, Bernhard Schölkopf, and Alex J Smola. Integrating structured biological data by kernel maximum mean discrepancy. *Bioinformatics*, 22(14):e49–e57, 2006. 1

[7] Jiefeng Chen, Frederick Liu, Besim Avci, Xi Wu, Yingyu Liang, and Somesh Jha. Detecting errors and estimating accuracy on unlabeled data with self-training ensembles. *Advances in Neural Information Processing Systems*, 34:14980–14992, 2021. 2

[8] Yuhao Chen, Xin Tan, Borui Zhao, Zhaowei Chen, Renjie Song, Jiajun Liang, and Xuequan Lu. Boosting semi-supervised learning by exploiting all unlabeled data. *arXiv preprint arXiv:2303.11066*, 2023. 1

[9] Corinna Cortes, Giulia DeSalvo, and Mehryar Mohri. Learning with rejection. In *Algorithmic Learning Theory: 27th International Conference, ALT 2016, Bari, Italy, October 19-21, 2016, Proceedings 27*, pages 67–82. Springer, 2016. 2

[10] Weijian Deng, Stephen Gould, and Liang Zheng. What does rotation prediction tell us about classifier accuracy under varying testing environments? In *International Conference on Machine Learning*, pages 2579–2589. PMLR, 2021. 5, 6

[11] Weijian Deng and Liang Zheng. Are labels always necessary for classifier accuracy evaluation? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15069–15078, 2021. 1, 5, 6

[12] Terrance DeVries and Graham W Taylor. Learning confidence for out-of-distribution detection in neural networks. *arXiv preprint arXiv:1802.04865*, 2018. 2

[13] Xiaohan Ding, Xiangyu Zhang, Ningning Ma, Jungong Han, Guiguang Ding, and Jian Sun. Repvgg: Making vgg-style convnets great again. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13733–13742, 2021. 5

[14] Pinar Donmez, Guy Lebanon, and Krishnakumar Balasubramanian. Unsupervised supervised learning i: Estimating classification and regression errors without labels. *Journal of Machine Learning Research*, 11(4), 2010. 2

[15] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *International conference on machine learning*, pages 1180–1189. PMLR, 2015. 2

[16] Saurabh Garg, Sivaraman Balakrishnan, Zico Kolter, and Zachary Lipton. Ratt: Leveraging unlabeled data to guarantee generalization. In *International Conference on Machine Learning*, pages 3598–3609. PMLR, 2021. 2

[17] Saurabh Garg, Sivaraman Balakrishnan, Zachary C Lipton, Behnam Neyshabur, and Hanie Sedghi. Leveraging unlabeled data to predict out-of-distribution performance. *arXiv preprint arXiv:2201.04234*, 2022. 2, 3, 5, 6

[18] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 513–520, 2011. 1

[19] Arthur Gretton, Karsten Borgwardt, Malte Rasch, Bernhard Schölkopf, and Alex Smola. A kernel method for the two-sample-problem. *Advances in neural information processing systems*, 19, 2006. 3

[20] Devin Guillory, Vaishaal Shankar, Sayna Ebrahimi, Trevor Darrell, and Ludwig Schmidt. Predicting with confidence on unseen distributions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1134–1144, 2021. 1, 3, 5, 6

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 5

[22] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*, 2019. 5

[23] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *arXiv preprint arXiv:1610.02136*, 2016. 2, 5, 6

[24] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. pages 448–456, 2015. 6

[25] Yiding Jiang, Vaishnavh Nagarajan, Christina Baek, and J Zico Kolter. Assessing generalization of sgd via disagreement. *arXiv preprint arXiv:2106.13799*, 2021. 2

[26] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 5

[27] Kimin Lee, Honglak Lee, Kibok Lee, and Jinwoo Shin. Training confidence-calibrated classifiers for detecting out-of-distribution samples. *arXiv preprint arXiv:1711.09325*, 2017. 2

[28] Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. *Advances in neural information processing systems*, 31, 2018. 2

[29] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. Learning transferable features with deep adaptation networks. In *International conference on machine learning*, pages 97–105. PMLR, 2015. 2, 3

[30] Omid Madani, David Pennock, and Gary Flake. Co-validation: Using model disagreement on unlabeled data to validate classification algorithms. *Advances in neural information processing systems*, 17, 2004. 2

[31] Behnam Neyshabur, Srinadh Bhojanapalli, David McAllester, and Nati Srebro. Exploring generalization in deep learning. *Advances in neural information processing systems*, 30, 2017. 1, 2

[32] Behnam Neyshabur, Zhiyuan Li, Srinadh Bhojanapalli, Yann LeCun, and Nathan Srebro. Towards understanding the role of over-parametrization in generalization of neural networks. *arXiv preprint arXiv:1805.12076*, 2018. 2

[33] Behnam Neyshabur, Ryota Tomioka, and Nathan Srebro. Norm-based capacity control in neural networks. In *Conference on learning theory*, pages 1376–1401. PMLR, 2015. 2

[34] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1406–1415, 2019. 2

[35] Emmanouil Platanios, Hoifung Poon, Tom M Mitchell, and Eric J Horvitz. Estimating accuracy from unlabeled data: A probabilistic logic approach. *Advances in neural information processing systems*, 30, 2017. 2

[36] Emmanouil Antonios Platanios, Avinava Dubey, and Tom Mitchell. Estimating accuracy from unlabeled data: A bayesian approach. In *International Conference on Machine Learning*, pages 1416–1425. PMLR, 2016. 2

[37] Tomaso Poggio, Kenji Kawaguchi, Qianli Liao, Brando Miranda, Lorenzo Rosasco, Xavier Boix, Jack Hidary, and Hrushikesh Mhaskar. Theory of deep learning iii: explaining the non-overfitting puzzle. *arXiv preprint arXiv:1801.00173*, 2017. 1

[38] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do cifar-10 classifiers generalize to cifar-10? *arXiv preprint arXiv:1806.00451*, 2018. 5

[39] Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. Fixmatch: Simplifying

semi-supervised learning with consistency and confidence. *Advances in neural information processing systems*, 33:596–608, 2020. 1

[40] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014. 6

[41] Baochen Sun, Jiashi Feng, and Kate Saenko. Return of frustratingly easy domain adaptation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 30, 2016. 2

[42] Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. Deep domain confusion: Maximizing for domain invariance. *arXiv preprint arXiv:1412.3474*, 2014. 2

[43] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 2, 5

[44] Apoorv Vyas, Nataraj Jammalamadaka, Xia Zhu, Dipankar Das, Bharat Kaul, and Theodore L Willke. Out-of-distribution detection using an ensemble of self supervised leave-out classifiers. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 550–564, 2018. 2

[45] Yaodong Yu, Zitong Yang, Alexander Wei, Yi Ma, and Jacob Steinhardt. Predicting out-of-distribution error with the projection norm. In *International Conference on Machine Learning*, pages 25721–25746. PMLR, 2022. 2

[46] Zhen Zhang, Mianzhi Wang, Yan Huang, and Arye Nehorai. Aligning infinite-dimensional covariance matrices in reproducing kernel hilbert spaces for domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3437–3445, 2018. 2

[47] Julian Zilly, Hannes Zilly, Oliver Richter, Roger Wattenhofer, Andrea Censi, and Emilio Frazzoli. The frechet distance of training and test distribution predicts the generalization gap. 2019. 1

[48] Yiming Zuo, Weichao Qiu, Lingxi Xie, Fangwei Zhong, Yizhou Wang, and Alan L Yuille. Craves: Controlling robotic arm with a vision-based economic system. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4214–4223, 2019. 2