An Empirical Study on Data Distribution-Aware Test Selection for Deep Learning Enhancement

QIANG HU, YUEJUN GUO, and MAXIME CORDY, University of Luxembourg
XIAOFEI XIE, Singapore Management University
LEI MA, University of Alberta
MIKE PAPADAKIS and YVES LE TRAON, University of Luxembourg

Similar to traditional software that is constantly under evolution, deep neural networks need to evolve upon the rapid growth of test data for continuous enhancement (e.g., adapting to distribution shift in a new environment for deployment). However, it is labor intensive to manually label all of the collected test data. Test selection solves this problem by strategically choosing a small set to label. Via retraining with the selected set, deep neural networks will achieve competitive accuracy. Unfortunately, existing selection metrics involve three main limitations: (1) using different retraining processes, (2) ignoring data distribution shifts, and (3) being insufficiently evaluated. To fill this gap, we first conduct a systemically empirical study to reveal the impact of the retraining process and data distribution on model enhancement. Then based on our findings, we propose DAT, a novel distribution-aware test selection metric. Experimental results reveal that retraining using both the training and selected data outperforms using only the selected data. None of the selection metrics perform the best under various data distributions. By contrast, DAT effectively alleviates the impact of distribution shifts and outperforms the compared metrics by up to five times and 30.09% accuracy improvement for model enhancement on simulated and in-the-wild distribution shift scenarios, respectively.

CCS Concepts: • **Software and its engineering** → *Empirical software validation*; • **Computing methodologies** → **Artificial intelligence;**

Additional Key Words and Phrases: Deep learning testing, test selection, data distribution

ACM Reference format:
Qiang Hu, Yuejun Guo, Maxime Cordy, Xiaofei Xie, Lei Ma, Mike Papadakis, and Yves Le Traon. 2022. An Empirical Study on Data Distribution-Aware Test Selection for Deep Learning Enhancement. *ACM Trans. Softw. Eng. Methodol.* 31, 4, Article 78 (July 2022), 30 pages.
https://doi.org/10.1145/3511598

1 INTRODUCTION

Deep neural networks (DNNs) are increasingly integrated into large software systems in various applications, such as face recognition [39], autonomous vehicles [2], speech recognition [52], and video gaming [44]. Despite the impressive success and great potential of DNNs, there

This work was supported by the Luxembourg National Research Funds (FNR) through CORE project C18/IS/12669767/STELLAR/LeTraon.
Authors' addresses: Q. Hu, Y. Guo (corresponding author), M. Cordy, M. Papadakis, and Y. Le Traon, University of Luxembourg, Luxembourg; email: firstname.lastname@uni.lu; X. Xie, Singapore Management University, Singapore; email: xiaofei.xfxie@gmail.com; L. Ma, University of Alberta, Canada; email: ma.lei@acm.org.

1049-331X/2022/07-ART78
https://doi.org/10.1145/3511598

are crucial accidents caused by quality issues of **deep learning (DL)** systems, such as Tesla/Uber accidents [45]. Therefore, similar to traditional software products, DNNs are required to undertake careful testing to check whether they match the expected requirements for reliable deployment. In practice, DNNs are mostly tested on a set of examples—the *test set*—that is extracted from the same dataset as the training set. As a result, by default, the test set and training set follow the same data distribution.

However, in real-world applications, DL systems face an important hurdle: the effectiveness (e.g., prediction accuracy) of the embedded DNN declines over time due to changes in data distribution. These distribution shifts [49] originate from multiple causes, such as changes in user behavior, seasonal data patterns, and benign alterations in the inputs. In such cases, software engineers have no choice but to manually re-engineer the DNN (i.e., design the architecture, set the hyper-parameters, and train on the data anew). And these re-engineering activities require considerable human and computational effort that is akin to the original production of the model. Distribution shift, therefore, constitutes one of the most important obstacles to the widespread dissemination of DL.

Similar to the general problem of software maintenance in conventional software engineering, distribution shift concerns enhancing the capability of the **machine learning (ML)** model to deal with unseen inputs. With the rapid growth of data that could follow a different data distribution, DL models may exhibit a misleading sense of achieving high performance on the original test data while having unexpected performance on the new data. Therefore, DL systems—in particular, the DNNs that are the essential backbone of these systems—also need to be evolved upon having the massive amount of collected new test data for continuous enhancement.

Fortunately, DL systems do not need to be re-engineered each time a distribution shift occurs but can rather cope with such shifts through a development strategy that promotes incrementality. Common strategies to combat drifts include retraining the DL model—that is, updating the DNN weights through additional training epochs using the new data. The retraining process can be entirely automated and therefore can avoid the heavy human and computational overhead of complete re-engineering. (Re)training a DNN requires labels of the collected data to calculate the loss information and guide the tuning of the model weights. However, data labeling is another important practical overhead. The reason is that although collecting massive new data (usually raw and unlabeled) is cheap and easy, labeling all of them is often manual, expensive, and prohibitively time consuming. For example, labeling the first version of the ImageNet dataset took groups of people more than 3 years [5]. In particular, the manual task of labeling can be more challenging in specific applications, when domain-specific knowledge is required.

Test selection refers to the area of research concerned with selecting, from a large set of unlabelled data, those data that are more likely to reveal errors in a given DNN [27]. Research has recently developed selection metrics to address this problem [3, 8, 16, 24, 38] as well as reduce the labeling effort. Once fault-revealing data have been found and labeled, the same data can be used to retrain the model (removing the errors that these data represent) and thereby improve its generalization. Although these metrics have demonstrated their potential to test and improve DNNs, we observed fundamental and experimental gaps that we aim to address in this article:

(1) *Utilization of two different retraining processes*: The retraining process plays a key role in the model enhancement, which leads the model to learn new information while keeping the original knowledge. However, in existing studies, two different retraining processes are used for model enhancement, and the impact of each process is still unclear and not explored. Taking three state-of-the-art metrics as an example, the **multiple-boundary clustering and prioritization (MCP)** [38] and the surprise adequacy guided metric [16] retrain a DNN

using only this subset. On the contrary, DeepGini [8] uses both the original training data and this subset.

(2) *Unaware of data distribution shift*: The shift of data distribution refers to the phenomenon that the distributions of training and test data are different, such as the images taken under different brightness. Usually, the data following the same or a different distribution are regarded as **in-distribution (ID)** or **out-of-distribution (OOD)** data, respectively. The distribution shift can be divided into two types: (1) synthetic distribution shift that comes from the computer-generated perturbation and (2) natural distribution shift that comes from unseen and unperturbed data. Data distribution has been proved to be critical in DL testing, especially for practical deployment of DL models [1, 6]. However, this factor is not considered in existing test selection metrics.

(3) *Evaluated by narrow experimental setups*: We observe that the effectiveness of existing selection metrics for model retraining is insufficiently evaluated. For instance, MCP is only evaluated on a combination of original test data (80%) and new data (20%), whereas DeepGini only selects data from the new data (100%) and retrains the model accordingly. The other combinations of data are uncovered and should be considered in the evaluation.

To elaborate on and address these limitations, in this article we conduct an empirical study to evaluate existing selection metrics under various data distribution shifts and answer the following three research questions:

RQ1: Which retraining process achieves better model enhancement?
RQ2: How effective are different test selection metrics under different data distributions for model enhancement?
RQ3: Concerning data distribution and class bias, what are the characteristics of the data selected by different metrics?

Overall, our empirical study evaluates six selection metrics over five datasets (including three image datasets and two text datasets) and two DNN architectures for each dataset (including both **feed-forward neural networks (FNNs)** and **recurrent neural networks (RNNs)**). In total, we retrained 71,280 models. By investigating the preceding research questions, we found that retraining using both the original training data and selected data achieves better results for model enhancement. Moreover, we observed that using this retraining process, existing selection metrics perform differently under different data distributions. For example, when OOD data are more than 70% in the new set, Random selection performs surprisingly the best. In addition, we found that class bias is another potential characteristic in addition to data distribution for data selection. Based on these findings, we further propose DAT, a distribution-aware test selection metric to alleviate the impact of distribution shifts on model retraining. The key idea of DAT is to *select uncertain and representative data from the ID and OOD sets*, respectively. In detail, we first utilize an OOD detector to split the new data into the ID set and OOD set. Afterward, for the ID set, DAT selects the most uncertain data that follow the same distribution as the training data but are not well learned by the model. For the OOD set, DAT selects the most representative data, which means that the selected data can represent the whole set. To demonstrate the effectiveness of our metric, we conduct experiments to answer the next two research questions. The experimental results show that DAT achieves the best performance among all the existing metrics. The two research questions are as follows:

RQ4: Under synthetic distribution shifts, how effective is DAT for model enhancement?
RQ5: Under natural distribution shifts, how effective is DAT for model enhancement?

In summary, the main contributions of this article are the following:

- To the best of our knowledge, we are the first to conduct a systemically empirical study of investigating how the retraining process and data distribution impact the test selection for model enhancement.
- This is the first study that analyzes and explores the characteristics of data selected by different metrics in terms of both data distribution and class bias.
- We propose DAT, the first distribution-aware test selection metric, which can reduce the impact of data distribution on model enhancement. In addition, we release our implementation and datasets for future use and research.[1]

The rest of the article is organized as follows. Section 2 introduces some background knowledge of this work. Section 3 highlights the problem we target. Section 4 presents the design of our empirical study. Section 5 details the results of our empirical study. Section 6 introduces and evaluates our DAT metric. Section 7 discusses the main findings and limitations of this work. Section 8 presents the related works, and Section 9 concludes the article.

2 BACKGROUND

We briefly introduce the background related to this work, including DNN, DL testing, and OOD detection.

2.1 Deep Neural Networks

A DNN is a type of artificial neural network with one or multiple hidden layers between the input and output layers. "Deep" in the name refers to the layers of the network being multiple. Each layer includes a large number of neurons that forms the basis of a DNN. The neurons in successive layers are connected with different weights that are tuned during the training process by minimizing the error between the prediction and the ground truth among a certain number of epochs.

Generally, building a DNN model requires three sets of data: the training set, validation set, and testing set. The training set is used to feed the model and tune the parameters during the training process. The validation set contributes to the training process to estimate how well a model has been trained. In practice, it is used for avoiding overfitting or underfitting, determining a stopping point for a possible best performance, finding the "optimal" number of hidden layers, and so on. The test set represents the unseen data for the trained model, which is independent of the training and validation sets. This set reveals how the model would behave when being applied to real-world data.

2.2 DL Testing and Test Selection

Software testing tries to reveal bugs in the software systems [31]. Normally, conventional software systems are designed and built by human logic. Testers could follow such logic to decide the test oracle, choose the testing techniques, and write test cases to test the systems. However, since DNNs are driven by training data and training processes, the logic inside the DNN models is unclear (known as their black-box property). As a result, it is hard to define bugs and design testing strategies for the DNNs. Recent works have proposed multiple techniques for DL testing [3, 7, 8, 16, 24, 26, 32, 38, 40, 43] that target different properties (e.g., fairness, adversarial robustness, and correctness) of a DNN model. For a review of ML testing, we refer to a survey by Ren et al. [34].

[1] https://github.com/code4papers/DAT.

Among massive testing approaches, test selection in DL is a technique that aims at solving two common problems: (1) to select data that can be used to represent the whole set and estimate the performance of the model on this set and (2) to select the data that are more likely to be misclassified by the model and then retrain an accurate model using the selected data. In this work, we focus on the second problem—how to utilize test selection to enhance the pre-trained model? Given that the data, in reality, are more complex than the data (which are carefully selected and organized) used for training a model, a pre-trained model goes through a retraining process with new unseen data to adapt to a specific application. In this article, we focus on testing the accuracy of a DNN model against new unseen data. In other words, given a pre-trained model and a set of unseen data, we retrain it by using a small subset of the unseen data to ensure high accuracy on both the original test data and this unseen data. Due to the high labeling cost, only a small set of data is selected in practice. Note that in contrast to the existing work [38], when testing the retrained model, both the original test data and the new unseen data are considered.

A related topic of test selection is active learning [30] in the ML community in the sense of reducing labeling cost. In active learning, a DNN is obtained iteratively through multiple steps. In each step, a set of data is selected to label and to update a pre-trained DNN obtained by the previous step. However, active learning and test selection have the following differences. One difference is the initial state. Generally, active learning starts from an early-stage DNN, whereas test selection already has a well-trained DNN. A second difference is the procedure. Active learning attains a DNN by multiple steps, and each step goes through a full training process. In test selection, the pre-trained DNN is enhanced using the selected data by retraining within several epochs (usually fewer than a full training process, such as only 5 or 10 epochs). Namely, test selection only needs one step. A thir difference is the goal. In active learning, the goal is to select a small amount of data to train a DNN that achieves similar performance as that using the entire data, whereas in test selection, the goal is to enhance the performance of a pre-trained DNN by retraining with a small amount of data.

2.3 OOD Detection

Generally, although we introduce some bias (e.g., applying image transformations to the training data [41]) into the model during the training process, the DNN model can mainly correctly predict the data that follow the same distribution as the training data. When testing the accuracy of DNNs on new data, the prediction may be erroneous and unreliable since the data may follow a different distribution compared with the training data.

The OOD technique, also known as outlier detection and anomaly detection, aims at distinguishing data concerning the distribution. Existing OOD approaches [14, 22, 25, 33, 36] use different methodologies to predict an anomaly score for a test input as the likelihood of following a learned distribution by a DNN model. However, the main goal of these approaches is to detect the data that come from two different datasets (e.g., MNIST and Fashion MNIST). In this work, we try to use the OOD detection method to detect the data that are from the mutated version derived from a dataset. The mutated version is generated by image transformations or adversarial attacks.

3 OBJECTIVES AND PROBLEM FORMULATION

Let us consider an N-class classification task over data $X \subseteq \mathbb{R}^d$ and labels $Y \subseteq \mathbb{Z}$. Let $f : \mathbf{x} \rightarrow y$ refer to a DNN trained on $X^{in} \subset X$, with $\mathbf{x} \in \mathbf{X}$ and $y \in Y$. We denote the distribution of the data X^{in} as $\mathcal{D}_{in}$ and refer to these data as the *ID data*. Now let X^{out} be a set of data that follows a mixture of distributions $\mathcal{D}_{in}, \mathcal{D}_{out}$, where $\mathcal{D}_{out}$ is an arbitrarily complex (possibly a mixture) distribution that differs from $\mathcal{D}_{in}$. In other words, X^{out} is a data sample that results from a distribution shift

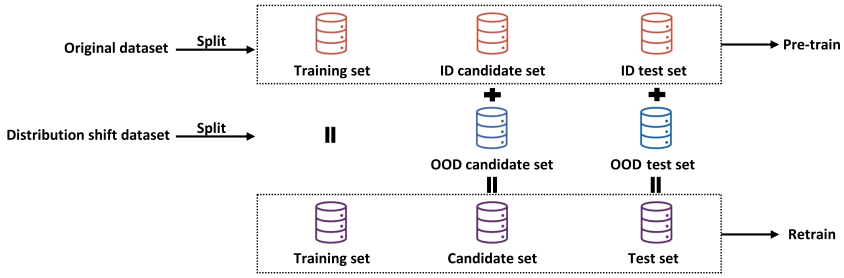


Fig. 1. Procedure of data preparation. All candidate sets are unlabeled, and the others are labeled.

from $\mathcal{D}_{in}$ to $\mathcal{D}_{in}, \mathcal{D}_{out}$. We furthermore assume that X^{out} is unlabeled and name these data the *OOD data*.

Our goal is to decrease the computational and human effort to improve models when distribution shift occurs. We aim to maximize the performance (e.g., accuracy) of the DNN on some $X^{out}_{test} \subset X^{out}$. We assume that we are allowed to change neither the architecture nor the hyperparameters of the DNN. Instead, we follow the straightforward and low computation cost method that consists of retraining the model for an additional number n of epochs with an independent sample $X^{out}_{train} \subset X^{out}$ that has no overlap with X^{out}_{test}. Given that we aim to minimize labeling cost, we also want $|X^{out}_{train}|$ to be under a pre-defined data budget b.

To address this challenge, we empirically investigate two key factors that may affect the effectiveness of retraining: the retraining process and the selection metric (i.e., the metric used to select X^{out}_{train} from X^{out}). Our analysis of the literature has revealed two types of retraining processes: retrain with X^{t}_{train} only or with a mixture of X^{out}_{train} and X_{in}. As for selecting X^{out}_{train}, we consider selection metrics that have been proposed in the literature and have also been used for retraining [3, 8, 16, 24, 38].

4 EMPIRICAL STUDY METHODOLOGY

First of all, to answer the first three research questions, we conduct a comprehensive empirical study to explore how retraining processes and data distribution affect the effectiveness of selection metrics for model retraining. This study provides the motivation for our proposed distribution-aware selection metric (Section 6).

4.1 Study Design

To conduct the empirical study, we first prepare the data as shown in Figure 1. Given a dataset, we randomly split it into three separate sets—the training set, ID candidate set, and ID test set—to build pre-trained DNNs. Afterward, we partition the distribution shift (OOD) dataset into the OOD candidate and test sets. Please refer to Section 4.4 for details on obtaining distribution shift datasets. Finally, we combine ID and OOD data with a certain ratio to simulate different distribution shifts. For instance, 10% ID + 90% OOD indicates that the new data has a dramatic shift where 90% data are unseen by pre-trained DNNs. In our study, we use 11 different combinations with the ratio ranging from 0% to 100% at a 10% interval. The candidate set represents new unlabeled data for selection and retraining, and the test set follows the same distribution as the candidate set for performance evaluation.

Figure 2 gives an overview of our empirical study. We first prepare pre-trained models for each dataset, then utilize different selection metrics to select and label data. Next, we use the selected data to retrain the pre-trained model with another few epochs. Finally, we test the retrained models on both the ID and new test sets.

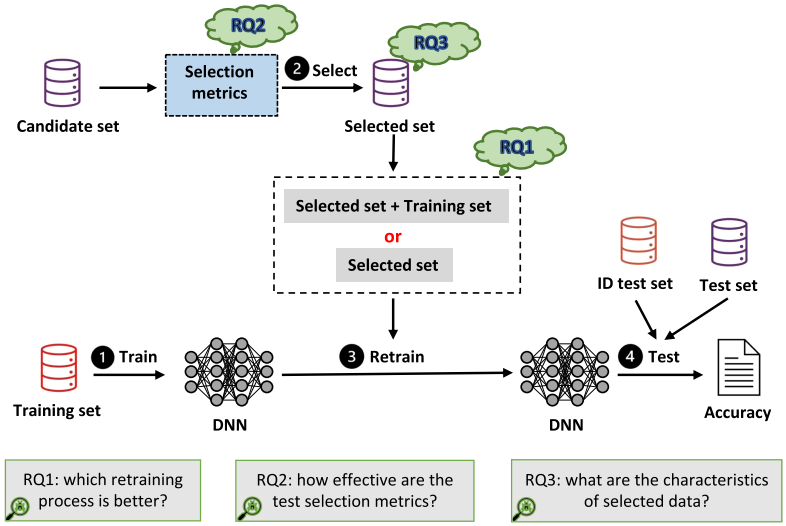


Fig. 2. Overview of our empirical study.

One factor that could highly affect the performance of the retrained model is the retraining process. In the literature, there are mainly two processes for model retraining. One is to retrain using only the selected data [16, 38]. The other is using both the training and selected data [8]. To answer RQ1, we apply both processes separately to produce two retrained DNNs. Next, we test the retrained models on test sets and compute their performance. Later, based on the findings of RQ1, we will apply the better retraining process to analyze how the data distribution would affect the effectiveness of each selection metric for model retraining and answer RQ2. In this phase, we only consider the test accuracy of retrained models. Furthermore, we investigate the properties of selected data by different selection metrics to answer RQ3.

4.2 Datasets and DNNs

In our empirical study, we consider five publicly available datasets: MNIST [21], Fashion-MNIST [51], CIFAR-10 [19], IMDb [28], and Newsgroups [20]. MNIST is a collection of grayscale images of handwritten digits (e.g., 1, 2). Fashion-MNIST includes grayscale images of fashion products (e.g., coat, shirt). CIFAR-10 contains color images (e.g., airplane, bird). IMDb is a dataset containing movie reviews that are widely used for sentiment analysis (i.e., positive or negative). Newsgroups is a text dataset that includes 20 different newsgroup subjects (e.g., space, baseball). For MNIST, Fashion-MNIST, and CIFAR-10, we randomly pick 10,000 data from the training set as the candidate set. For IMDb and Newsgroups, we randomly collect 5,000 and 4,000 data from the training set and the candidate set, respectively. For each dataset, we use two different well-known DNN models in previous research. For the image datasets, we consider the famous convolutional neural networks (e.g., LeNet and ResNet(. Since RNNs are good at handling sequential data, we utilize embedding layers to encode the text into vectors first, then we use RNNs to process the vectors and predict sentiment results. In addition, we follow Hendrycks and Gimpel [13] to build the fully connected neural network for the Newsgroups dataset. Hence, our study covers image and text data, as well as FNNs and RNNs. All of the detailed model architectures and hyperparameters are available on our project website.[1] Table 1 shows details of the datasets and DNNs. We measure model performance in terms of accuracy, as it is the metric originally used for the tasks and datasets that we study.

Table 1. Datasets and DNN Models

Dataset	Data Type	#Training	#Test	#Classes	DNN	#Layers	#Parameters	Test Accuracy (%)
MNIST	Image	60,000	10,000	10	LeNet-1	5	3,246	97.91
					LeNet-5	7	107,786	98.90
Fashion-MNIST	Image	60,000	10,000	10	LeNet-1	5	3,246	87.29
					LeNet-5	7	107,786	90.29
CIFAR-10	Image	50,000	10,000	10	ResNet-20	20	274,442	85.79
					NiN	23	972,658	87.16
IMDb	Text	25,000	5,000	2	LSTM	5	2,694,206	85.61
					GRU	5	2,661,694	86.46
Newsgroups	Text	4,000	1,000	20	NN1	2	450,650	86.70
					NN2	3	452,600	81.30

"Test accuracy" is the accuracy (%) of the ID test set (see Figure 1).

Note that since we do not use all of the training data to train the model, the test accuracy of each model may not achieve the state of the art.

4.3 Selection Metrics

Various selection metrics have been proposed and evaluated for data prioritization and data labeling effort reduction. In this study, we choose four metrics (MCP, DeepGini, CES, and DSA) proposed in the SE community. Note that MCP, CES, and DSA have been evaluated as the best metrics in a recent study [38] compared with the others, such as likelihood-based surprise adequacy (LSA) [16] and adaptive active learning (AAL) [23]. DeepGini is a newly proposed method for enhancing the performance of DNNs. In addition, we take the Random selection metric as the baseline. Given that the task of active learning within each stage is similar to test selection (please refer to Section 2.2 for more details), the most basic and popular metric, Entropy, [46] is also considered for comparison. We briefly introduce each metric as follows.

Throughout the article, we use $p_i(\mathbf{x})$, $0 \leq i \leq N$ to represent the predicted probability of $\mathbf{x}$ belonging to the ith class.

Random. Random selection is basic and the simplest selection method. It draws data directly from the given set regardless of the model's behavior. Each data is randomly selected—namely, it has the same probability of being chosen.

Multiple-boundary clustering and prioritization. MCP [38] selects test data limited in decision boundary areas. Concretely, it proceeds in three steps. First, the DNN model runs on each test sample to give a sequence of output probabilities. Second, MCP conducts a boundary area clustering to divide the data into different clusters. A cluster (the boundary area between two classes) is formed according to the top-2 classes of test data. In addition, for each test data, MCP computes its priority in its belonging cluster as the ratio of the probability of the first class to the probability of the second class. Finally, test data with high priorities are evenly selected from each non-empty cluster. The intuition behind MCP is that if the top-2 probabilities of a test sample are close, this sample is close to the decision boundary between the corresponding two classes.

Cross entropy based sampling. The main idea of **cross entropy based sampling (CES)** [24] is to select a subset of test data that can maximally represent the distribution of the entire test dataset via the cross entropy. More specifically, this subset should have the minimum cross entropy with the entire test dataset. To solve this optimization problem, CES utilizes a similar algorithm to the random walk [35]. It starts with a random subset T (smaller than the budget) with a few test data, then repeatedly enlarges T by merging another subset P that is randomly selected and has the minimum cross entropy with T.

Distance-based surprise adequacy. **Distance-based surprise adequacy (DSA)** [16] is an adequacy criterion that aims at measuring how surprising a test sample is to a DNN model concerning

the training data. It computes the surprise adequacy by the Euclidean distance between the model's behaviors represented by the activation traces of the test sample and the training set. Finally, the data with high adequacy are selected.

DeepGini. Similar to Entropy, DeepGini [8] also selects the most uncertain data using the output probabilities by

$$\arg\max_{\mathbf{x}\in X}\left(1-\sum_{i=1}^{N}\left(p_i\left(\mathbf{x}\right)\right)^2\right). \tag{1}$$

Entropy-based metric (Entropy). As a widely used information-theoretic metric, entropy, also known as Shannon entropy [37], measures the average level of information required to obtain a possible prediction. In other words, it calculates the uncertainty for a DNN model to output a prediction. Based on this concept, Entropy [46] selects the test data that have the maximum uncertainties, and its formal definition is

$$\arg\max_{\mathbf{x}\in X}\left(-\sum_{i=1}^{N}p_i\left(\mathbf{x}\right)\log p_i\left(\mathbf{x}\right)\right). \tag{2}$$

Most of the aforementioned metrics (MCP, CES, DeepGini, and Entropy) are only designed for classification tasks since they require the output probability of each class in their methodologies. The only exception is DSA, which also works for regression tasks. Our metric DAT is also designed for classification tasks—one objective of DAT is to collect data with balanced classes. Therefore, in our study, we only focus on the classification tasks. Nonetheless, to the best of our knowledge, our study is the largest one that considers both image and text classification tasks with both synthetic and natural distribution shifts.

4.4 OOD Data Preparation

In our study, we consider two types of distribution shift: synthetic and natural. Both are widely studied in recent works [13, 42].

4.4.1 OOD Data with Synthetic Distribution Shift. Synthetic distribution shift comes from the computer-generated perturbation. In the literature [1, 17, 38, 43], there are two types of image mutation methods to generate noise data: image transformation [41] and adversarial attack [34]. Table 2 describes the six image transformations and the two adversarial attacks used in our study. Image transformation applies basic geometric transformations to mimic different real-world conditions such as changing the contrast or brightness of images and rotating the camera. Here, we consider transformations that are common in the real world and whose relevance has been shown in previous studies [1, 38, 41]: rotation, shear, translation, scaling, brightness, and contrast. We follow Berend et al. [1] and Shen et al. [38] to set up the parameters of these transformations. For example, for MNIST-scale, we set the scale coefficient as 0.8. All parameters of image transformations can be found on our project site.[1] Adversarial attacks add an imperceptible perturbation into an image to mislead DNNs. These attacks have been associated with distribution shifts and can be useful to improve the generalization ability of ML models [10]. We use two of the most common attack algorithms: FGSM [10] and PGD [29]. We utilize the L_{inf} distance to calculate the perturbation with a commonly used [29, 50] maximum size of 0.3 (8/255) for MNIST and Fashion-MNIST (CIFAR-10).

To make sure that each mutation method (i.e., each image transformation and adversarial attack) introduces distribution shifts, we empirically show that there is a greater distribution difference (1) between the original training set and the original test set and (2) between the original training set and the mutated test set. If (2) is greater than (1), then it would mean that the mutations induce a

Table 2. Description of Mutation Operators

Type	Mutation Operator	Description
Transformation	Rotation	Rotate an image by a certain angle
	Shear	Shear an image horizontally
	Translation	Translate several pixels down right
	Scale	Change the size of an image
	Brightness	Adjust the brightness of an image
	Contrast	Adjust the contrast of an image
Attack	FGSM	Fast gradient sign method
	PGD	Project gradient descent

Table 3. JSD Between Training Set and Other Sets

	Test	Brightness	Contrast	Rotration	Scale	Shear	Translation	FGSM	PGD	OOD
MNIST	0.05	0.77	0.52	0.61	0.62	0.57	0.56	0.73	0.65	0.77
Fashion-MNIST	0.05	0.62	0.36	0.48	0.53	0.55	0.52	0.56	0.38	0.44
CIFAR-10	0.07	0.21	0.50	0.51	0.57	0.47	0.47	0.40	0.25	0.60

distribution shift compared to the natural difference that is due to data generalization. To measure such distribution differences, we combine a state-of-the-art **outlier exposure (OE)** detector [14] (more details in Section 6.1) and **Jensen-Shannon divergence (JSD)** score [9]. OE enables the identification of data that do not belong to a given distribution (in our case, the original training set determines the distribution). It assigns a score to each example, where a higher score means that the example is farther from the given distribution. To build the OOD detector, we need a baseline of OOD that are clearly not from the original distribution. In our case, to build the OOD detectors for MNIST, Fashion-MNIST, and CIFAR-10, we respectively use Fashion-MNIST, MNIST, and SVHN. The reason behind this choice [1] is that MNIST and Fashion-MNIST are black-and-white images, whereas CIFAR-10 and SVHN are colored. Once we have an OOD detector, we predict the score of the examples in the two test sets and build the corresponding two histograms. We calculate the JSD between the two histograms. JSD is an established metric for the dissimilarity between two probability distributions. A higher JSD indicates higher dissimilarity.

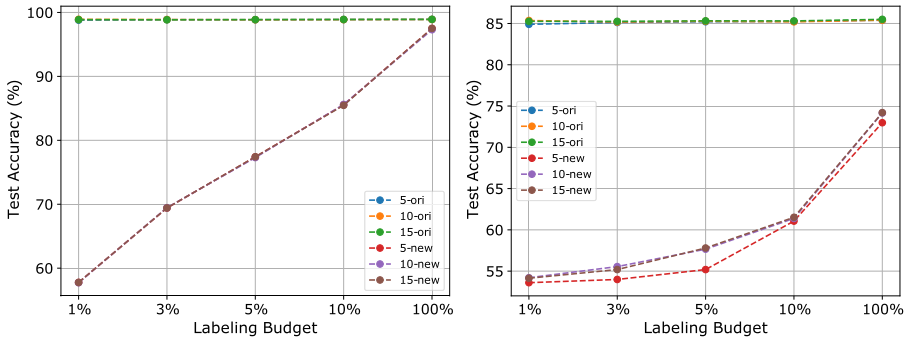
Table 3 lists the results. The JSD between the original training and test sets (at most 0.07) is much smaller than the JSD between the training and mutated sets (at least 0.21). For Fashion-MNIST, some mutated sets have an even greater JSD scores than the OOD sets, revealing that the shifts that mutations induce can be more significant than a shift to a completely different dataset. In conclusion, these results confirm that the used mutations are indeed appropriate to emulate distribution shifts.

OOD data with natural distribution shift. Natural distribution shift comes from unseen environments. For the text datasets, it is easy to collect this kind of OOD data that targets the same task as the ID data (e.g., we can collect the movie reviews from different websites and groups of people). We obtain such datasets (IMDb and Newsgroups) from the baseline work [13] directly. Following the same settings as Hendrycks and Gimpel [13], for the IMDb dataset, we use the combination of customer reviews and movie reviews as the OOD data. For Newsgroups, we randomly choose 10 groups as the ID data and 10 groups as the OOD data.

Table 4 lists the average accuracy of models on test sets under different distributions. We can see that the accuracy degrades gradually when the test set includes OOD data, which confirms that distribution shift indeed weakens the reliability of the pre-trained DNN and it is necessary to enhance this DNN.

Table 4. Average Test Accuracy (%) of the Test Set (see Figure 1)

Distribution	MNIST		Fashion-MNIST		CIFAR10		IMDb		Newsgroups		Average
ID + OOD	LeNet-1	LeNet-5	LeNet-1	LeNet-5	NiN	ResNet20	LSTM	GRU	NN	NN2	
0% + 100%	28.65	36.23	22.62	20.19	51.05	46.90	68.36	67.58	0.40	6.90	34.89
10% + 90%	35.59	42.44	29.11	27.37	54.68	50.85	70.04	69.44	9.00	14.00	40.25
20% + 80%	42.40	48.66	35.51	34.33	58.35	54.69	71.78	71.56	17.50	21.90	45.67
30% + 70%	49.28	54.82	41.86	41.30	61.95	58.50	73.64	73.94	26.20	29.60	51.11
40% + 60%	56.19	61.03	48.31	48.02	65.48	62.51	75.48	75.56	35.20	37.50	56.53
50% + 50%	63.14	67.46	54.87	54.95	68.99	66.29	77.34	77.60	44.40	45.00	62.00
60% + 40%	70.14	73.67	61.36	61.95	72.74	70.27	79.04	79.46	53.10	52.30	67.40
70% + 30%	77.04	79.74	67.75	68.97	76.34	74.28	81.26	81.52	61.70	60.00	72.86
80% + 20%	84.10	86.06	74.26	76.17	79.90	78.03	83.08	83.52	69.70	66.50	78.13
90% + 10%	90.88	92.24	80.79	83.18	83.51	81.95	84.46	85.20	78.20	73.70	83.41
100% + 0%	97.91	98.90	87.29	90.29	87.16	85.79	86.06	86.94	86.70	81.00	88.80
Average	63.21	67.39	54.88	55.16	69.10	66.37	77.32	77.48	43.83	44.40	61.91



(a) MNIST, LeNet-5 (b) CIFAR-10, ResNet20

Fig. 3. Test accuracy of original test data and new test data after using a Random selection metric to select different budgets of data and retrain the model. Note that 5-ori means the test accuracy on original test data after retraining the model with five epochs.

4.5 Retraining Settings

Like previous studies [8, 16, 38] and following our working assumptions, during the retraining process, the hyperparameters are set in the same way as the pre-trained DNN, such as the DNN architecture, momentum, batch size, activation function, dropout, optimization, learning rate, and loss function. In addition, for the number of epochs, we retrain LeNet-1 and LeNet-5 with an additional 5 epochs as in the work of Shen et al. [38], and 10 epochs for ResNet-20 and NiN as in the work of Zhang et al. [55]. We do not follow the same setting as Shen et al. [38] to use 5 epochs to retrain the CIFAR-10 based models. The reason is that in some cases we found that 5 epochs are not enough for the model weights to converge. As shown in Figure 3, for MNIST-based models, the test accuracy of original test data and new test data are almost the same after using 5, 10, and 15 epochs to retrain the models. However, for the CIFAR-10-based models, there are clear gaps of the test accuracy on the new data when using 5 epochs to retrain models compared with using 10 and 15 epochs to retrain, especially when the labeling budgets are 3% and 5%. Since this is the first work to evaluate the aforementioned selection metrics for model retraining on text datasets, we follow our practical experience to set 5 epochs to retrain IMDb- and Newsgroups-based models.

4.6 Repetitions and Infrastructure

Each experiment is repeated five times to reduce the randomness introduced in the training process. All experiments run on a high-performance computer cluster, and each cluster node runs a 2.6-GHz Intel Xeon Gold 6132 CPU with an NVIDIA Tesla V100 16-G SXM2 GPU.

5 EXPERIMENTAL RESULTS

We report the experimental results to answer each research question and summarize our findings. Remember that the combined candidate set without labels represents the new coming data where selected data for model retraining come from. The combined test set with labels is for testing the resulting accuracy of DNNs. We create these two sets in a way that they contain the same percentage of ID data and OOD data.

5.1 RQ1: Different Retraining Processes

Our goal is to analyze which retraining process can maintain high accuracy on original test data and meanwhile achieve high accuracy on new data. We denote by Type 1 the process that retrains the model with the new data only, and by Type 2 the process that retrains the model using a combination of new data and previous training data. To determine which retraining process is better, we compare the accuracy improvement of DNNs after retraining using each process. For each DNN, we create 11 sets of unlabeled data as well as 11 sets of test data following different data distributions by combining ID and OOD data. Next, each metric (of 6) selects a certain ratio (budget) of data from each unlabeled candidate set for labeling and model retraining. In our study, the ratio of selected data is set to 1%, 3%, 5%, and 10% as in the work of Shen et al. [38]. In addition, to exclude the effect of selection metrics on the retrained models, we also consider using all candidate data (i.e., with budget 100%) to retrain the DNN models by different retraining processes. Finally, we calculate the accuracy improvement of DNNs after retraining. In total, we have retrained 71,280 DNN models, 3 datasets $\times$ 2 models $\times$ (6 selection metrics $\times$ 4 budgets + 1 budget) $\times$ 11 distributions $\times$ 8 operators $\times$ 5 repetitions image-based models, and 2 datasets $\times$ 2 models $\times$ 6 selection metrics $\times$ 4 budgets $\times$ 11 distributions $\times$ 5 repetitions text-based models. Tables 5 and 6 show the statistical improvements of test accuracy over the 71,280 DNNs of the original and new test data, respectively. In each table, the first column represents the data distribution of the candidate set. For instance, 10% + 90% indicates that the candidate set consists of 10% ID data and 90% OOD data.

In the case of maintaining performance on original test set, as demonstrated by Table 5, in most cases (512 out of 550) over five datasets, the retraining process of Type 2 achieves better results than Type 1. And on average, in all cases, the retraining process of Type 2 achieves better (by up to 29.52%) results than Type 1. Namely, retraining using the combination of new selected data and training data is a better option than using only the new selected data for this objective. Now looking at Table 6, surprisingly, retraining with only the new data does not ensure higher accuracy on the new test data in most cases. In general, in only 153 cases (out of 550 cases), the retraining process of Type 1 achieves better accuracy than Type 2. On average, we can see that only when more (at least 80%) OOD data are included in the candidate set, the retraining process of Type 1 can achieve better results (by up to 4.28%) than Type 2. Note that, meanwhile, the accuracy of the original test data is greatly sacrificed. For instance, in the case of 100% OOD data and Budget 10%, Type 1 improves the accuracy on the new test set by 48.46%, but the accuracy on the original test set drops significantly by 29.84%. In addition, this outperformance on new test sets degrades when a smaller budget is available. Overall, on average, retraining using both the training data and the newly selected data better enhances the model without losing the high performance on the

Table 5. Average (over All Selection Metrics) Improvement of Test Accuracy (%) on Original Test Sets (ID Test Data) with Different Selection Budgets

Distribution ID + OOD	Budget 1%		Budget 3%		Budget 5%		Budget 10%		Budget 100%			Budget 1%		Budget 3%		Budget 5%		Budget 10%		Budget 100%	
	Type 1	Type 2	Type 1	Type 2	Type 1	Type 2	Type 1	Type 2	Type 1	Type 2		Type 1	Type 2	Type 1	Type 2	Type 1	Type 2	Type 1	Type 2	Type 1	Type 2
0% + 100%	−8.50	0.36	−18.87	0.32	−16.82	0.27	−22.28	0.25	−29.04	−0.06		−13.41	−0.03	−25.70	−0.03	−25.95	−0.04	−35.98	−0.01	−30.84	0.01
10% + 90%	−5.25	0.40	−15.35	0.35	−14.64	0.34	−21.11	0.27	−8.27	−0.02		−12.39	−0.02	−22.47	−0.04	−23.83	−0.03	−28.27	−0.02	−4.21	0.02
20% + 80%	−4.70	0.40	−11.79	0.34	−9.96	0.31	−21.46	0.27	−5.35	−0.01		−10.37	−0.03	−19.42	0.00	−18.54	0.01	−23.72	−0.02	−2.61	0.02
30% + 70%	−3.71	0.39	−9.30	0.35	−9.08	0.35	−17.07	0.28	−3.92	0.04		−8.14	−0.04	−17.83	−0.06	−15.01	−0.03	−18.59	−0.01	−1.78	0.05
40% + 60%	−2.39	0.39	−9.68	0.37	−9.08	0.37	−15.11	0.26	−2.91	0.08		−7.70	−0.04	−12.87	−0.04	−11.73	−0.04	−13.30	−0.01	−1.37	0.02
MNIST LeNet1	−1.07	0.41	−5.54	0.38	−6.99	0.38	−13.08	0.28	−2.27	0.15	MNIST LeNet5	−6.70	−0.03	−9.99	−0.01	−8.53	0.01	−9.84	0.01	−1.05	0.04
60% + 40%	−0.92	0.40	−4.97	0.38	−4.58	0.37	−7.82	0.31	−1.82	0.20		−4.11	−0.05	−6.40	−0.03	−5.53	−0.01	−9.51	−0.01	−0.73	0.05
70% + 30%	−0.65	0.39	−3.14	0.40	−3.91	0.37	−6.76	0.34	−1.41	0.20		−3.32	−0.03	−4.91	0.00	−4.72	0.02	−6.09	0.02	−0.53	0.05
80% + 20%	−0.89	0.42	−2.62	0.38	−2.42	0.40	−3.05	0.35	−1.07	0.25		−1.84	0.01	−3.48	−0.01	−3.22	0.01	−4.40	0.03	−0.38	0.04
90% + 10%	−0.36	0.42	−1.44	0.41	−1.22	0.38	−1.36	0.40	−0.66	0.30		−0.96	0.03	−2.62	0.04	−1.66	0.04	−2.23	0.06	−0.27	0.05
100% + 0%	0.07	0.42	0.07	0.44	0.14	0.47	0.19	0.47	0.16	0.47		−0.75	0.06	−0.69	0.08	−0.18	0.07	−0.20	0.07	−0.07	0.06
Average	−2.58	0.40	−7.51	0.37	−7.14	0.37	−11.72	0.32	−5.14	0.14		−6.34	−0.02	−11.49	−0.01	−10.81	0.00	−13.83	0.01	−3.99	0.04
0% + 100%	−20.65	0.07	−23.08	−0.02	−24.28	−0.35	−23.59	−0.15	−28.63	−1.05		−19.26	0.07	−22.65	0.00	−20.98	−0.22	−20.46	−0.07	−25.62	−0.24
10% + 90%	−13.23	0.19	−16.27	0.07	−17.83	−0.26	−17.57	−0.14	−9.24	−0.92		−13.63	0.00	−19.67	0.05	−18.48	−0.24	−18.42	−0.05	−5.90	−0.22
20% + 80%	−9.80	0.17	−13.47	0.01	−14.81	−0.24	−15.03	−0.07	−6.81	−0.77		−9.83	0.04	−16.87	0.05	−17.59	−0.06	−17.33	0.00	−4.35	−0.11
30% + 70%	−6.05	0.04	−11.65	0.05	−13.12	−0.07	−13.26	0.03	−5.45	−0.72		−8.23	−0.01	−14.81	0.05	−17.23	−0.09	−15.42	−0.05	−3.14	−0.04
40% + 60%	−3.60	0.08	−10.37	0.12	−10.72	0.01	−11.72	0.03	−4.45	−0.49		−6.97	0.04	−13.98	0.00	−15.21	−0.11	−14.14	−0.04	−2.36	0.07
F-MNIST LeNet1	−3.31	0.14	−9.43	0.12	−10.02	0.00	−10.25	0.04	−3.85	−0.47	F-MNIST LeNet5	−6.51	0.02	−11.44	0.02	−13.79	−0.03	−13.91	0.04	−1.89	0.08
60% + 40%	−2.51	0.11	−5.56	0.22	−8.26	0.01	−9.48	0.05	−3.18	−0.32		−4.03	0.09	−9.71	0.14	−12.37	−0.18	−11.77	0.10	−1.57	0.16
70% + 30%	−1.97	0.12	−3.99	0.17	−5.94	−0.04	−6.88	0.10	−2.48	−0.21		−2.70	0.10	−9.22	0.15	−10.84	0.05	−10.12	0.19	−1.28	0.21
80% + 20%	−1.84	0.03	−2.45	0.20	−4.01	0.01	−5.21	0.18	−1.89	−0.13		−1.91	0.23	−6.31	0.16	−8.84	0.08	−7.62	0.18	−0.93	0.26
90% + 10%	−1.07	0.04	−1.55	0.24	−2.38	0.16	−2.70	0.19	−1.19	0.04		−1.33	0.24	−4.44	0.17	−5.96	0.07	−4.37	0.24	−0.59	0.35
100% + 0%	−0.40	−0.03	−0.57	0.30	−0.52	0.10	−0.25	0.29	0.01	0.27		0.00	0.23	0.11	0.25	−0.02	0.15	−0.45	0.29	0.12	0.44
Average	−5.86	0.09	−8.94	0.14	−10.17	−0.06	−10.54	0.05	−6.10	−0.43		−6.76	0.10	−11.73	0.09	−12.85	−0.05	−12.18	0.08	−4.32	0.09
0% + 100%	−6.64	−0.43	−9.39	−0.61	−9.31	−0.52	−11.37	−0.59	−8.08	−0.39		−14.28	−0.58	−16.17	−0.70	−18.82	−0.57	−22.54	−0.63	−13.47	−0.45
10% + 90%	−3.74	−0.59	−7.97	−0.37	−6.84	−0.54	−9.94	−0.58	−3.70	−0.31		−12.13	−0.56	−17.90	−0.51	−15.82	−0.58	−18.16	−0.47	−6.87	−0.41
20% + 80%	−2.10	−0.52	−6.19	−0.38	−5.92	−0.39	−8.66	−0.44	−2.97	−0.34		−7.56	−0.51	−12.84	−0.54	−13.40	−0.45	−16.12	−0.43	−5.77	−0.29
30% + 70%	−2.13	−0.46	−6.44	−0.35	−4.50	−0.40	−7.38	−0.26	−2.31	0.00		−3.31	−0.48	−9.54	−0.36	−11.45	−0.35	−16.14	−0.28	−5.14	−0.24
40% + 60%	−1.33	−0.57	−3.54	−0.32	−2.98	−0.24	−5.92	−0.23	−1.82	−0.21		−2.34	−0.46	−3.92	−0.40	−8.29	−0.35	−14.24	−0.23	−4.75	−0.30
CIFAR-10 ReNet20	50% + 50%	−0.37	−0.38	0.11	−0.28	−2.67	−0.24	−6.14	−0.38	−1.58	−0.03	CIFAR-10 NiN	−1.81	−0.41	−3.05	−0.33	−4.72	−0.34	−12.74	−0.29	−4.28	−0.18
60% + 40%	0.88	−0.35	0.61	−0.30	−0.04	−0.20	−4.69	−0.18	−1.26	0.08		−1.78	−0.41	−2.82	−0.28	−3.64	−0.33	−10.09	−0.29	−3.83	−0.21
70% + 30%	1.06	−0.30	0.55	−0.25	−0.09	−0.21	−4.61	−0.16	−0.77	0.09		−1.66	−0.39	−2.95	−0.31	−3.78	−0.37	−9.59	−0.19	−3.19	0.00
80% + 20%	1.18	−0.24	0.51	−0.17	0.29	−0.19	−3.43	−0.14	−0.43	0.15		−1.63	−0.36	−2.98	−0.32	−3.51	−0.25	−8.20	−0.15	−2.67	−0.02
90% + 10%	1.18	−0.19	0.58	−0.19	0.26	−0.18	−2.97	−0.13	0.12	0.18		−1.61	−0.35	−2.72	−0.25	−3.30	−0.24	−7.59	−0.21	−2.29	0.13
100% + 0%	1.14	−0.24	0.57	−0.21	0.66	−0.13	−4.35	−0.14	0.60	0.27		−1.64	−0.30	−2.48	−0.36	−3.40	−0.23	−7.20	−0.18	−1.97	0.21
Average	−0.99	−0.39	−2.78	−0.31	−2.83	−0.29	−6.31	−0.29	−2.02	−0.05		−4.52	−0.44	−7.04	−0.40	−8.19	−0.37	−12.96	−0.30	−4.93	−0.16
0% + 100%	−0.44	0.68	−2.20	0.78	−3.76	0.75	−2.13	0.71	−6.07	0.78		−0.03	0.54	−1.78	0.42	−4.42	0.34	−4.36	0.46	−4.78	0.59
10% + 90%	−0.16	0.64	−1.27	0.70	−1.12	0.69	−0.68	0.71	−0.31	0.68		−0.12	0.47	−2.34	0.34	−1.30	0.48	−1.88	0.47	0.36	0.67
20% + 80%	−0.25	0.49	−1.54	0.82	−1.29	0.81	0.09	0.63	1.23	1.01		−0.06	0.51	−0.57	0.56	−1.68	0.37	−0.52	0.42	0.69	1.01
30% + 70%	−0.30	0.75	−1.32	0.78	−0.10	0.77	0.35	0.78	0.81	1.07		−0.18	0.45	−0.42	0.47	−0.52	0.56	−0.32	0.52	0.35	1.03
40% + 60%	−0.49	0.81	−0.03	0.86	0.38	0.85	0.33	0.87	1.57	1.16		−0.31	0.49	−0.18	0.54	−0.02	0.47	0.12	0.48	1.19	0.88
IMDb-LSTM	50% + 50%	0.27	0.90	−0.21	0.85	0.39	0.85	0.67	0.87	1.75	1.31	IMDb-GRU	0.16	0.41	−0.08	0.63	0.18	0.64	0.46	0.60	1.05	1.25
60% + 40%	0.36	0.93	−0.45	0.86	0.49	0.86	0.58	0.97	1.73	1.30		0.13	0.61	0.05	0.47	0.43	0.67	0.43	0.73	1.81	1.40
70% + 30%	0.27	0.87	−0.34	0.86	0.08	0.95	0.79	1.01	1.92	1.60		0.11	0.49	0.25	0.52	0.19	0.64	0.74	0.67	1.97	1.49
80% + 20%	0.18	0.85	0.48	0.90	0.68	0.90	0.96	0.95	1.77	1.47		0.09	0.64	0.18	0.51	0.24	0.67	0.64	0.76	1.92	1.52
90% + 10%	0.35	0.89	0.18	0.80	0.45	1.04	1.14	1.00	2.04	1.58		0.15	0.57	0.16	0.66	0.39	0.58	0.84	0.75	2.06	1.56
100% + 0%	0.17	0.85	0.63	0.98	0.74	0.94	0.89	1.03	2.01	1.65		0.16	0.52	0.10	0.65	0.63	0.67	0.85	0.78	1.95	1.71
Average	0.00	0.79	−0.55	0.83	−0.28	0.86	0.27	0.87	0.77	1.24		0.01	0.52	−0.42	0.52	−0.53	0.55	−0.27	0.60	0.78	1.19
0% + 100%	−27.82	0.33	−62.95	0.46	−66.43	0.39	−67.55	0.19	−79.23	−2.37		−73.60	0.75	−72.67	0.16	−67.37	0.36	−63.77	−0.31	−72.62	−0.03
10% + 90%	−18.86	0.43	−47.79	0.29	−45.39	0.40	−37.10	0.50	−29.82	−1.37		−48.54	1.27	−32.12	0.59	−29.14	0.44	−27.85	0.31	−16.06	1.83
20% + 80%	−14.97	0.30	−32.33	0.51	−31.81	0.40	−25.16	0.38	−15.26	−0.54		−38.16	1.18	−25.26	0.88	−18.47	0.27	−20.16	0.78	−5.90	2.40
30% + 70%	−12.22	0.31	−16.04	0.48	−20.89	0.45	−15.80	0.40	−7.11	−0.08		−25.13	1.11	−18.88	1.04	−15.75	1.16	−13.60	0.80	−2.32	3.10
40% + 60%	−9.20	0.54	−7.64	0.44	−15.92	0.52	−10.75	0.57	−4.53	0.11		−18.16	1.19	−16.44	0.85	−13.18	1.19	−9.46	1.52	−0.39	3.01
Newsgroups NN	50% + 50%	−2.75	0.57	−6.31	0.56	−11.86	0.58	−7.54	0.50	−2.20	0.75	Newsgroups NN2	−12.45	1.73	−10.30	1.62	−9.17	1.40	−8.39	1.94	1.83	4.14
60% + 40%	−1.72	0.42	−3.69	0.60	−6.30	0.60	−3.81	0.69	−0.31	1.37		−9.84	1.52	−8.13	1.65	−7.01	2.26	−6.43	2.04	3.18	4.77
70% + 30%	−0.16	0.66	−3.23	0.68	−2.68	0.81	−1.69	0.75	0.41	1.84		−8.34	1.34	−6.69	1.82	−4.93	2.05	−3.83	2.48	3.88	5.16
80% + 20%	0.14	0.79	−0.69	0.79	−0.53	0.72	−0.55	0.96	1.36	2.17		−7.94	1.56	−4.04	1.90	−3.24	2.02	−1.81	2.64	4.65	5.71
90% + 10%	0.21	0.60	0.01	0.65	−0.14	0.89	0.18	1.11	1.73	2.22		−2.83	1.76	−4.30	2.18	−1.46	2.58	0.78	2.96	5.19	6.14
100% + 0%	0.20	0.62	−0.01	0.68	0.57	0.84	0.66	1.14	2.22	2.59		−5.62	1.81	−1.45	2.17	−0.12	2.74	1.90	3.45	6.19	6.30
Average	−7.92	0.51	−16.42	0.56	−18.31	0.60	−15.37	0.65	−12.07	0.61		−22.78	1.38	−18.21	1.35	−15.44	1.50	−13.87	1.69	−6.58	3.86

Distribution ID + OOD	Budget 1%		Budget 3%		Budget 5%		Budget 10%		Budget 100%	
	Type 1	Type 2	Type 1	Type 2	Type 1	Type 2	Type 1	Type 2	Type 1	Type 2
0% + 100%	−18.46	0.18	−25.55	0.08	−25.81	0.04	−27.40	−0.01	−29.84	−0.32
10% + 90%	−12.80	0.22	−18.32	0.15	−17.44	0.07	−18.10	0.10	−8.40	−0.01
20% + 80%	−9.78	0.20	−14.03	0.22	−13.35	0.10	−14.81	0.15	−4.71	0.24
30% + 70%	−6.94	0.21	−10.62	0.24	−10.76	0.23	−11.72	0.22	−3.00	0.42
40% + 60%	−5.25	0.25	−7.86	0.24	−8.68	0.27	−9.42	0.32	−1.98	0.43
50% + 50%	−3.46	0.34	−5.62	0.36	−6.72	0.33	−8.07	0.36	−1.25	0.70
60% + 40%	−2.35	0.33	−4.11	0.37	−4.68	0.41	−6.26	0.44	−0.60	0.88
70% + 30%	−1.74	0.32	−3.37	0.40	−3.66	0.43	−4.80	0.52	−0.15	1.04
80% + 20%	−1.44	0.39	−2.14	0.43	−2.46	0.44	−3.27	0.58	0.24	1.14
90% + 10%	−0.63	0.40	−1.62	0.47	−1.50	0.53	−1.83	0.64	0.61	1.26
100% + 0%	−0.67	0.39	−0.37	0.50	−0.15	0.56	−0.80	0.73	1.12	1.40
Average	−5.77	0.29	−8.51	0.32	−8.66	0.31	−9.68	0.37	−4.36	0.65

Note: The better result between the two types of retraining processes is highlighted in gray. Type 1: Using only the new data; Type 2: using the combination of new selected data and training data. "Distribution" represents different distribution shifts by different percentages of ID and OOD data in the candidate set. Baseline: Please refer to Table 1 for the accuracy of pre-trained DNNs.

original test data. We can conclude that this retraining strategy achieves a good balance between the original and new test sets.

Answer to RQ1: Retraining DNNs with only the new selected data sacrifices accuracy on the original distribution for improvement on the new distribution. By contrast, mixing the training data with the new data can achieve high accuracy on both the original and the new distributions. More specifically, retraining on new data achieves higher test accuracy only when there are more than 80% OOD data in the candidate set. Overall, combining training data and selected data remains the best option.

5.2 RQ2: Effectiveness of Different Selection Metrics

Based on the answer of RQ1, we use the retraining process that combines the training and new data for our remaining studies. In addition, since the accuracy on the original test set is highly

Table 6. Average (over All Selection Metrics) Improvement of Test Accuracy (%) on New Test Sets with Different Selection Budgets

Distribution		Budget 1%		Budget 3%		Budget 5%		Budget 10%		Budget 100%			Budget 1%		Budget 3%		Budget 5%		Budget 10%		Budget 100%	
ID + OOD		Type 1	Type 2	Type 1	Type 2	Type 1	Type 2	Type 1	Type 2	Type 1	Type 2		Type 1	Type 2	Type 1	Type 2	Type 1	Type 2	Type 1	Type 2	Type 1	Type 2
	0% + 100%	6.71	9.86	15.22	16.73	19.23	21.43	35.63	30.88	61.87	63.82		23.99	21.55	32.05	33.18	37.09	41.18	43.69	49.33	60.56	61.27
	10% + 90%	5.79	9.20	12.49	14.84	15.84	18.57	29.51	27.38	54.96	56.97		19.12	19.55	27.68	30.81	30.78	36.63	35.81	44.37	54.19	55.13
	20% + 80%	4.99	8.20	9.41	13.02	13.24	16.50	24.31	24.04	48.24	50.11		15.57	16.92	22.03	27.33	25.75	32.92	29.77	39.61	47.97	48.84
	30% + 70%	4.41	7.47	8.20	11.57	11.15	14.48	18.63	20.24	41.50	43.33		12.62	15.54	16.96	24.34	20.89	28.31	23.88	34.48	41.89	42.70
	40% + 60%	3.79	6.70	5.97	9.95	8.58	12.64	14.06	17.64	34.92	36.56		10.26	13.74	13.15	20.99	16.82	24.82	18.51	29.87	35.88	36.62
MNIST	50% + 50%	2.87	5.61	3.89	8.42	6.29	11.00	9.93	14.92	28.46	29.89	MNIST	8.59	11.89	9.84	17.70	12.36	20.97	14.03	24.95	29.74	30.35
LeNet1	60% + 40%	2.44	4.62	3.23	7.00	4.28	9.03	6.65	12.22	22.01	23.24	LeNet5	6.66	9.69	6.99	14.29	8.51	16.56	9.71	19.90	23.74	24.24
	70% + 30%	1.36	3.40	1.82	5.46	2.53	6.95	3.85	9.30	15.82	16.58		4.40	7.06	3.79	10.69	5.25	12.43	7.53	14.96	17.83	18.17
	80% + 20%	0.70	2.28	0.53	3.85	1.30	4.63	2.69	6.20	9.54	10.09		2.50	4.62	1.84	7.20	2.87	8.07	4.27	9.62	11.61	11.82
	90% + 10%	0.32	1.32	0.14	2.01	0.66	2.42	1.68	3.09	3.87	4.20		1.06	2.53	−0.05	3.65	1.29	4.04	1.66	4.71	5.58	5.66
	100% + 0%	0.07	0.41	0.07	0.43	0.14	0.47	0.18	0.47	0.15	0.46		−0.76	0.09	−0.68	0.12	−0.16	0.11	−0.17	0.10	−0.04	0.10
	Average	3.04	5.37	5.54	8.48	7.57	10.74	13.38	15.13	29.21	30.48		9.46	11.20	12.15	17.30	14.68	20.55	17.15	24.72	29.91	30.44
	0% + 100%	10.15	4.59	19.07	8.02	26.35	12.59	32.24	20.28	54.49	51.27		14.66	12.28	28.15	19.80	36.37	26.75	45.44	37.14	59.74	61.99
	10% + 90%	4.11	4.28	13.73	6.79	21.94	9.67	28.71	16.44	47.60	45.19		10.44	11.05	24.13	17.14	31.04	23.07	39.44	32.33	53.26	55.27
	20% + 80%	3.21	3.81	8.59	5.96	15.37	8.56	22.37	13.21	41.54	39.82		8.39	9.99	17.91	15.31	23.70	19.84	32.33	28.22	46.84	48.83
	30% + 70%	2.68	3.78	6.45	5.46	10.70	7.52	15.81	11.83	35.43	34.12		6.11	8.69	12.13	13.29	18.85	17.63	26.16	24.74	40.81	42.43
	40% + 60%	1.51	3.12	4.82	4.66	8.39	6.76	12.07	10.09	29.68	28.58		4.11	7.45	9.03	11.60	13.67	15.45	20.12	21.23	34.76	36.08
F-MNIST	50% + 50%	0.57	2.60	2.73	4.11	5.05	5.74	8.28	8.53	23.54	22.90	F-MNIST	2.87	6.12	6.28	10.26	9.05	13.26	13.79	17.76	28.80	29.48
LeNet1	60% + 40%	−0.36	1.98	1.65	3.20	3.20	4.60	5.07	6.87	18.01	17.60	LeNet5	1.83	5.10	4.53	8.55	6.11	10.62	9.18	14.25	22.52	23.11
	70% + 30%	−0.31	1.52	1.50	2.64	1.48	3.39	2.86	5.09	12.92	12.53		0.80	3.95	1.57	6.77	2.54	8.18	5.12	10.73	16.46	17.08
	80% + 20%	−0.71	1.12	0.84	1.98	−0.20	2.32	0.57	3.49	7.55	7.41		0.29	2.67	−0.53	4.63	−0.46	5.39	1.62	6.89	10.30	10.94
	90% + 10%	−0.66	0.68	−0.32	1.14	−0.98	1.24	−0.63	1.71	2.83	2.92		−0.33	1.48	−2.10	2.29	−2.28	2.66	−0.42	3.24	4.55	5.05
	100% + 0%	−0.40	0.26	−0.56	0.31	−0.53	0.12	−0.26	0.30	0.02	0.27		−0.02	0.22	0.11	0.25	−0.02	0.14	−0.42	0.35	0.12	0.44
	Average	1.80	2.52	5.32	4.02	8.25	5.68	11.55	8.90	24.87	23.87		4.47	6.27	9.20	9.99	12.60	13.00	17.49	17.90	28.92	30.06
	0% + 100%	12.94	7.30	14.70	8.65	18.00	10.76	22.69	14.49	33.81	27.26		4.89	7.88	4.67	8.55	7.50	10.21	11.60	13.83	27.54	28.19
	10% + 90%	7.41	6.27	9.63	7.21	13.33	8.58	18.37	11.84	29.76	23.95		2.68	6.94	2.81	6.60	5.75	7.79	8.84	10.42	24.06	25.09
	20% + 80%	6.56	5.74	7.82	5.99	9.31	7.21	12.66	9.39	26.08	20.95		−0.22	6.23	−0.04	5.69	2.97	6.60	5.01	8.05	20.72	21.57
	30% + 70%	5.49	4.96	5.85	5.52	7.27	6.14	8.66	7.74	22.37	17.83		−2.87	5.44	−1.62	4.75	−0.26	5.51	0.97	6.75	17.24	18.42
	40% + 60%	4.53	4.13	4.55	4.20	5.74	4.84	6.28	5.59	18.62	14.55		−2.68	0.93	−2.41	4.00	−2.68	4.48	−1.16	5.49	13.83	15.17
CIFAR-10	50% + 50%	3.59	3.44	3.67	3.49	3.94	3.96	4.06	4.43	15.10	11.58	CIFAR-10	−2.26	−0.24	−2.67	3.11	−3.68	3.52	−3.12	4.45	10.70	11.95
ReNet20	60% + 40%	2.81	0.40	2.96	2.47	2.76	2.99	2.24	3.21	11.46	8.81	NiN	−2.85	−0.89	−2.74	2.33	−3.48	2.73	−4.64	3.33	7.10	9.02
	70% + 30%	2.33	−0.08	2.12	1.75	1.85	1.93	0.08	1.93	7.95	6.03		−2.45	−1.64	−3.19	1.59	−3.40	1.82	−6.40	2.33	4.32	6.13
	80% + 20%	1.85	−0.64	1.52	0.98	1.34	1.17	−1.31	1.09	4.91	3.68		−2.08	−2.32	−3.07	0.77	−3.42	0.98	−6.97	1.41	1.72	3.54
	90% + 10%	1.46	−1.11	1.01	0.20	0.67	0.34	−2.45	0.20	2.33	1.53		−2.07	−2.72	−2.62	0.25	−3.11	0.32	−7.13	0.53	−0.47	1.41
	100% + 0%	1.11	−1.46	0.55	−0.22	0.63	−0.13	−4.34	−0.48	0.60	0.27		−1.92	−3.15	−2.45	−0.34	−3.36	−0.23	−7.32	−0.74	−1.92	0.22
	Average	4.55	2.63	4.94	3.66	5.90	4.34	6.08	5.40	15.73	12.40		−1.08	1.50	−1.21	3.39	−0.65	3.98	−0.94	5.08	11.35	12.79
	0% + 100%	0.83	0.42	0.88	0.65	1.17	0.55	1.28	0.63	2.63	1.89		1.44	1.56	0.39	1.45	0.07	1.47	0.00	1.45	2.74	3.00
	10% + 90%	0.48	0.57	0.50	0.71	0.92	0.46	0.93	0.59	1.90	2.01		1.37	1.52	−0.30	1.56	0.54	1.42	−0.11	1.35	2.58	2.88
	20% + 80%	0.36	0.57	0.44	0.72	0.80	0.66	1.14	0.58	1.87	1.59		0.94	1.14	−0.12	1.11	−0.79	1.36	0.26	1.51	2.15	2.21
	30% + 70%	0.26	0.51	0.07	0.54	0.71	0.61	0.53	0.54	1.83	1.67		0.86	1.09	0.03	1.10	0.23	1.18	0.01	0.95	1.93	1.94
	40% + 60%	0.01	0.48	0.37	0.55	0.27	0.33	0.85	0.41	1.67	1.51		0.62	1.07	0.28	1.05	0.81	1.16	0.57	1.10	2.44	1.72
IMDb-	50% + 50%	0.33	0.60	0.17	0.51	0.64	0.53	0.94	0.63	2.03	1.14	IMDb-	0.41	1.02	0.16	1.09	0.57	1.13	0.83	0.99	2.10	1.98
LSTM	60% + 40%	0.34	0.55	−0.30	0.74	0.21	0.59	0.94	0.74	1.70	1.45	GRU	0.33	0.87	0.32	0.89	0.42	0.90	1.01	0.94	2.11	1.64
	70% + 30%	0.17	0.33	−0.39	0.66	0.11	0.53	1.06	0.65	2.14	1.21		0.31	0.60	0.44	0.69	0.41	0.67	1.13	0.72	2.39	1.51
	80% + 20%	0.11	0.59	0.31	0.67	0.58	0.65	1.03	0.61	1.81	1.18		0.25	0.63	0.20	0.57	0.40	0.70	0.90	0.71	2.11	1.66
	90% + 10%	0.32	0.80	0.17	0.75	0.48	0.95	1.15	0.96	2.02	1.43		0.21	0.53	0.16	0.63	0.32	0.63	0.93	0.78	2.12	1.46
	100% + 0%	0.17	0.85	0.63	0.98	0.74	0.94	0.89	1.03	2.01	1.65		0.16	0.52	0.10	0.65	0.63	0.67	0.85	0.78	1.95	1.71
	Average	0.31	0.57	0.26	0.68	0.60	0.62	0.98	0.67	1.97	1.52		0.63	0.96	0.15	0.98	0.33	1.03	0.58	1.03	2.24	1.97
	0% + 100%	18.78	7.65	30.21	17.30	37.25	26.71	48.27	44.86	94.37	93.90		24.99	10.82	34.88	23.09	45.11	32.40	55.98	46.03	86.88	85.08
	10% + 90%	14.13	6.20	21.24	14.69	26.63	22.30	38.04	38.34	81.79	83.70		15.89	8.42	26.31	20.39	33.01	26.86	44.90	39.01	75.94	76.00
	20% + 80%	12.52	5.24	14.22	12.40	21.84	19.64	31.35	32.65	72.48	73.93		9.18	6.60	18.75	16.92	25.30	23.10	36.39	33.74	66.53	67.30
	30% + 70%	9.28	4.97	9.91	9.33	16.12	15.57	25.12	26.33	63.41	64.06		4.85	5.47	12.70	13.49	19.80	18.53	29.51	28.41	57.79	58.76
	40% + 60%	6.41	3.93	7.62	7.79	9.67	11.87	20.53	21.43	52.36	53.43		3.47	3.99	6.67	10.48	12.93	14.74	21.26	21.65	48.04	48.83
Newsgroups	50% + 50%	3.13	2.92	4.26	5.42	5.83	8.48	14.03	15.84	41.83	42.38	Newsgroups	−2.20	3.15	3.29	7.72	7.32	11.24	15.80	16.87	39.35	39.73
NN	60% + 40%	2.56	2.25	2.98	4.25	5.09	5.82	10.33	11.10	31.67	31.86	NN2	−1.73	2.60	1.36	5.89	4.63	8.03	10.56	12.43	31.13	31.06
	70% + 30%	1.27	1.64	1.55	3.15	3.69	4.38	6.53	7.05	21.60	22.35		−3.11	2.00	0.40	4.64	2.55	6.71	7.16	9.28	23.29	23.71
	80% + 20%	0.95	1.60	0.95	1.94	2.15	2.42	2.95	3.76	13.16	13.07		−4.58	1.84	−1.20	3.17	0.52	4.36	3.92	6.09	15.85	16.00
	90% + 10%	0.53	0.91	0.68	1.15	1.06	1.79	1.17	2.01	5.24	5.67		−2.13	1.89	−3.23	2.67	−0.25	3.20	2.49	4.20	9.23	9.89
	100% + 0%	0.20	0.62	−0.01	0.68	0.57	0.84	0.66	1.14	2.22	2.59		−5.62	1.81	−1.45	2.17	−0.12	2.74	1.90	3.45	6.19	6.30
	Average	6.34	3.45	8.51	7.10	11.81	10.89	18.09	18.59	43.65	44.27		3.27	4.42	8.95	10.06	13.71	13.81	20.90	20.11	41.84	42.06

Distribution		Budget 1%		Budget 3%		Budget 5%		Budget 10%		Budget 100%	
ID + OOD		Type 1	Type 2	Type 1	Type 2	Type 1	Type 2	Type 1	Type 2	Type 1	Type 2
	0% + 100%	11.94	8.39	18.02	13.74	22.81	18.41	29.68	25.89	48.46	47.77
	10% + 90%	8.14	7.40	13.82	12.07	17.98	15.54	24.44	22.21	42.61	42.62
	20% + 80%	6.15	6.44	9.90	10.45	13.75	13.64	19.56	19.10	37.44	37.51
	30% + 70%	4.37	5.79	7.07	8.94	10.55	11.55	14.93	16.20	32.42	32.53
	40% + 60%	2.90	4.55	5.00	7.53	7.42	9.71	11.31	13.45	27.22	27.30
	50% + 50%	1.79	3.71	3.16	6.18	4.74	7.98	7.86	10.94	22.17	22.14
Average	60% + 40%	1.20	2.72	2.10	4.96	3.17	6.19	5.11	8.50	17.15	17.20
	70% + 30%	0.48	1.88	0.96	3.80	1.70	4.70	2.89	6.20	12.47	12.53
	80% + 20%	−0.07	1.24	0.14	2.57	0.51	3.07	0.97	3.99	7.86	7.94
	90% + 10%	−0.13	0.63	−0.62	1.48	−0.21	1.76	−0.16	2.14	3.73	3.92
	100% + 0%	−0.70	0.02	−0.37	0.50	−0.15	0.57	−0.80	0.64	1.13	1.40
	Average	3.28	3.89	5.38	6.57	7.48	8.46	10.53	11.75	22.97	22.99

Note: The better result between the two types of retraining processes is highlighted in gray. Type 1: Using only the new data; Type 2: using the combination of new selected data and training data. "Distribution" represents different distribution shifts of the candidate set. Baseline: Please refer to Table 4 for the accuracy of pre-trained DNNs.

maintained, we only consider the performance on the new test set in the following research questions.

We observe that the evaluation of existing selection metrics for model retraining lacks insights regarding the amplitude of the distribution shift. For example, MCP is evaluated by only using one data combination (80% original test data + 20% mutated data), and DeepGini is evaluated by only (100%) mutated data. Thus, the actual effectiveness of these metrics when facing different data distributions is ambiguous. In this research question, we explore how different distributions of candidate data affect the effectiveness of each metric for model enhancement. To achieve this, we still follow the same experimental setting as our first study. In total, for each image dataset, each test combination has 64 (2 models × 8 operators × 4 budgets) retraining performances averaging on five repetitions. In this section, we only report the results of image datasets because we use natural OOD data for text datasets, whereas we can experimentally control the OOD data produced for images. Therefore, there are only a few combinations (8) for text data, and the statistical results

are insufficient to give conclusions. We report the results of text datasets in Section 6.4 (where we consider real-world distribution shift) by using the test accuracy improvement after retraining as the measurement metric.

Table 7 lists the frequency of each selection metric achieving the top-1 and top-3 best test accuracy over the 64 cases in each test combination. Note that we also report top-3 results since if the metric achieves top-3 best performance, it outperforms half of the metrics. Interestingly, when the new set contains more than 70% OOD data, Random selection defeats the other five carefully designed metrics in most cases (20 out of 24). Moreover, in total, the frequency of Random selection being the best is almost twice as the second-best metric. For example, the Random selection obtains 90 times the top-1 best performance in the 100% OOD test set, whereas the second-best, CES, only reaches 47 times. In addition, when the included OOD data are more than 70%, the two metrics CES and DSA outperform the uncertain-based metrics Entropy, DeepGini, and MCP. The reason is that when the new data consists of too much OOD data, a massive amount of information related to the new distribution has not been learned by the pre-trained model. In this case, the model needs to learn more from a representative sample of the new data rather than from the most uncertain data. Among all studied metrics, Random selection is the most effective because it does not bias the selection toward specific data and therefore achieves better representativeness.

With the increase of ID data in the candidate set and test set, the uncertain-based metrics (Entropy, DeepGini, and MCP) achieve better results than the other three metrics (CES, DSA, and Random selection). More specifically, in total, when the proportion of OOD data is between 40% and 70%, MCP performs consistently better than all of the others. However, when the test set contains more (≥80%) ID data, Entropy and DeepGini achieve the best results. This is because as a higher ratio of ID data is part of the new distribution, the pre-trained model has already learned more from this new distribution. The OOD data, in this case, can be seen as outliers that generate uncertainty in the model and are therefore naturally selected by the uncertainty-based metrics. Hence, retraining on these data fills the gap in model learning and achieves better performance.

Answer to RQ2: None of the selection metrics outperforms the others across all ranges of distribution shifts. When the new set contains much more (≥70%) OOD than ID data, the simple but effective Random selection defeats the others. On the contrary, when the new set contains more ID data, the uncertain-based metrics are more effective.

5.3 RQ3: Distribution and Bias of Selected Data

Following our findings presented previously, in this research question we further explore another property that may impact the effectiveness of the selection metrics: class bias of the data selected by each metric. In other words, we check if the selected data are evenly chosen from different classes, which is done by calculating the variance of labels of selected data. For example, given a three classes task, we select 100 data. If the numbers of selected data for each class is 30, 30, 40, the variance is $Variance(30, 30, 40) = 22.22$, whereas if the label numbers are 90, 5, 5, the variance should be $Variance(90, 5, 5) = 1,605.55$. A small variance indicates a slight bias in data.

First, Figure 4 illustrates the data distribution of selected data by different metrics. Compared with the data distribution in the candidate set (black dashed line), three metrics, CES, DSA, and random, select ID and OOD data following almost the same distribution. On the contrary, the uncertain-based metrics, Entropy, DeepGini, and MCP, tend to pick more OOD than ID data. The reason is that the uncertain-based metrics always choose the most informative data, and the OOD data have likely not been learned by the pre-trained DNNs. Thus, there is more chance for OOD data to be selected by these uncertain-based metrics.

Table 7. Frequency of Being the Top-1 and Top-3 Best of the Six Selection Metrics Under Different Data Distributions

	Distribution ID + OOD	Entropy	DeepGini	MCP	CES	DSA	Random	Entropy	DeepGini	MCP	CES	DSA	Random
		Top-1						Top-3					
MNIST	0% + 100%	1	0	1	19	4	39	3	4	36	60	26	63
	10% + 90%	1	2	6	14	9	32	2	8	32	60	30	60
	20% + 80%	0	3	7	20	8	26	5	13	36	51	30	57
	30% + 70%	4	5	17	16	1	21	12	19	44	48	21	48
	40% + 60%	5	9	21	11	5	13	20	32	47	35	19	39
	50% + 50%	3	12	28	9	2	10	29	40	52	29	13	29
	60% + 40%	8	16	23	3	6	8	36	50	54	19	10	23
	70% + 30%	7	25	23	3	4	2	54	58	54	10	9	7
	80% + 20%	20	26	14	0	1	3	59	59	57	3	5	9
	90% + 10%	29	26	6	0	2	1	60	59	59	0	7	7
	100% + 0%	15	17	13	9	3	7	40	34	33	30	27	28
	Average	8.45	12.82	14.45	9.45	4.09	14.73	29.09	34.18	45.82	31.36	17.91	33.64
Fashion-MNIST	0% + 100%	0	2	3	13	11	35	3	3	26	60	40	60
	10% + 90%	1	0	6	14	8	35	2	3	30	60	42	55
	20% + 80%	0	0	4	12	15	33	1	5	32	58	40	56
	30% + 70%	1	2	8	12	16	25	6	4	41	53	35	53
	40% + 60%	0	0	14	14	19	17	6	9	38	49	42	48
	50% + 50%	1	4	16	9	20	14	14	17	39	36	36	50
	60% + 40%	6	9	23	3	16	7	21	31	43	25	40	32
	70% + 30%	6	14	30	2	9	3	31	40	52	20	27	22
	80% + 20%	15	16	22	3	5	3	44	42	56	13	21	16
	90% + 10%	22	16	18	2	4	2	50	53	58	9	14	8
	100% + 0%	16	15	9	10	9	5	35	36	37	32	26	26
	Average	6.18	7.09	13.91	8.55	12.00	16.27	19.36	22.09	41.09	37.73	33.00	38.73
CIFAR-10	0% + 100%	8	14	5	15	6	16	28	36	25	33	31	39
	10% + 90%	7	15	3	10	10	19	29	39	14	37	30	43
	20% + 80%	17	14	0	11	6	16	35	41	11	39	24	42
	30% + 70%	15	18	1	9	4	18	38	47	17	30	20	40
	40% + 60%	14	22	4	7	8	9	41	49	18	32	21	31
	50% + 50%	19	24	1	4	11	5	43	50	23	21	20	35
	60% + 40%	18	25	4	5	8	4	47	48	24	23	22	28
	70% + 30%	24	19	7	5	5	4	48	51	34	21	16	22
	80% + 20%	25	24	6	2	2	5	51	49	46	14	14	18
	90% + 10%	20	27	6	4	3	4	49	54	46	16	12	15
	100% + 0%	10	15	20	6	7	6	37	36	48	31	22	18
	Average	16.09	19.73	5.18	7.09	6.36	9.64	40.55	45.45	28.00	27.18	21.09	29.73
Total	0% + 100%	9	16	9	47	21	90	34	43	87	153	97	162
	10% + 90%	9	17	15	38	27	86	33	50	76	157	102	158
	20% + 80%	17	17	11	43	29	75	41	59	79	148	94	155
	30% + 70%	20	25	26	37	21	64	56	70	102	131	76	141
	40% + 60%	19	31	39	32	32	39	67	90	103	118	82	116
	50% + 50%	23	40	45	22	33	29	86	107	116	86	69	112
	60% + 40%	32	50	50	11	30	19	104	129	121	67	72	83
	70% + 30%	37	58	60	10	18	9	133	149	140	51	52	51
	80% + 20%	60	66	42	5	8	11	154	150	159	30	40	43
	90% + 10%	71	69	30	6	9	7	159	166	163	25	33	30
	100% + 0%	41	47	42	25	19	18	112	106	118	93	75	72
	Average	30.73	39.64	33.55	25.09	22.45	40.64	94.5	107.6	117.7	90.6	69.5	96.1

Note: The best result is highlighted in gray. "Distribution" represents different distribution shifts of the candidate set.

Second, Table 8 shows the class bias presented by the variance of labels of selected data. Compared with the other selection metrics, random always selects data evenly from different classes. However, the variances of two uncertainty-based metrics (Entropy and DeepGini) are more than twice the others. Although MCP is designed to select data evenly from different boundary areas, this metric has a higher bias than CES and Random selection. The reason for Entropy, DeepGini, and MCP selecting bias classes is that they all use the predicted probability to measure the uncertainty, which is highly affected by the accuracy of the pre-trained model on the new data. When there are more OOD data, the prediction is more unreliable. For instance, MCP tends to decrease the bias in data when the proportion of ID data is above 50%.

Considering the results of RQ2, we conjecture that when there are more OOD (e.g., ≥70%) data in the candidate set, it is better to select data with a better class balance to retrain the model. As an illustration of this hypothesis, Random selection and CES achieve both higher accuracy and class

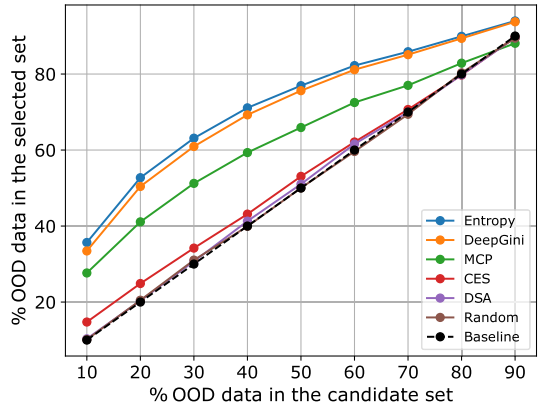


Fig. 4. Comparison of data distributions of the selected set by different metrics. Baseline: The selected set has the same data distribution (same percentage of OOD and ID data) as the candidate set.

Table 8. Class Bias (Label Variance) of Selected Data by Different Metrics

Distribution ID+OOD	Entropy	DeepGini	MCP	CES	DSA	Random
0% + 100%	479.42	429.86	361.13	213.13	279.09	188.75
10% + 90%	444.59	395.74	280.83	211.27	267.65	184.97
20% + 80%	423.30	377.19	270.06	210.26	263.01	182.18
30% + 70%	408.98	365.81	262.70	208.08	258.92	186.39
40% + 60%	387.65	350.30	250.64	207.57	256.99	181.09
50% + 50%	370.65	338.58	240.68	208.34	255.24	178.84
60% + 40%	350.82	324.47	233.23	209.83	255.07	177.36
70% + 30%	326.56	309.37	224.24	208.23	257.25	177.05
80% + 20%	307.41	299.57	218.07	207.98	261.64	178.10
90% + 10%	299.48	300.55	211.54	209.48	265.95	177.30
100% + 0%	392.50	385.06	216.62	213.89	268.97	177.93
Average	381.03	352.41	251.80	209.83	262.71	180.91

Note: The best result is highlighted in gray. "Distribution" represents different distribution shifts of the candidate set. The number means the average (over all selection metrics) variance in the number of examples that the metric selects for each class.

balance. Since in the candidate set most of the data have not been learned by the model, a better class balance can help represent a more diverse distribution and, in turn, lead the model to learn more diverse information.

Answer to RQ3: Uncertain-based selection metrics (Entropy, DeepGini, and MCP) tend to select more OOD data, and do so in a way that creates class imbalance in the set of retraining data. On the contrary, CES and random select data with more balanced classes and better representativeness of the new distribution. These two factors contribute to the difference in the effectiveness of the selection metrics, depending on how much ID data are still part of the new distribution.

6 DISTRIBUTION-AWARE TEST SELECTION

According to the findings of our empirical study, when the new data contain more (≥60%) ID than OOD data, the uncertain-based metrics outperform others in enhancing the performance of DNNs.

However, when there are more (≥70%) OOD data, none of the existing metrics (Entropy, DeepGini, MCP, CES, and DSA) defeats the Random selection. Therefore, there is room for proposing a new metric to deal with the second case in a better way than random. *Intuition*: since different selection metrics behave differently on different data distributions, we should consider different selection strategies for different distributions of data. From the ID data, we need to select the uncertain ones, whereas for the OOD data, we should consider the data representativity. Given our previous findings, the guiding principles of our new metric are twofold: (1) it must consider how much the data distribution has changed (by using OOD detector), and (2) it should preserve the balance between classes (by comparing the label balance between the selected data and the whole data). Based on these two principles, we propose a distribution-aware test selection metric named *DAT*.

6.1 OOD Detector

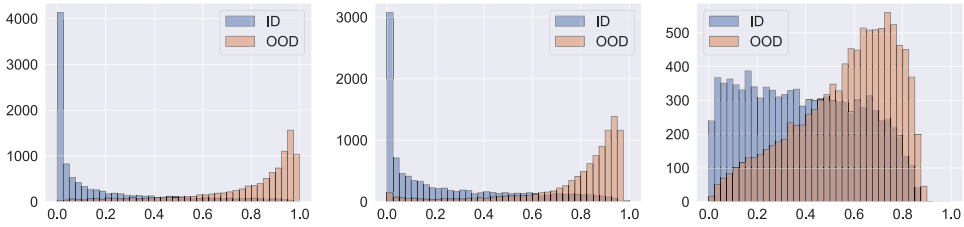
Before looking into DAT, we introduce an OOD detection approach employed in our metric. The OE detector [14] is currently the best OOD detection method as assessed in a recent empirical study [1]. Given a distribution $\mathcal{D}_{in}$, the detector aims at identifying if a sample is derived from $\mathcal{D}_{in}$ or not. The main idea is to separately train a DNN that additionally optimizes the loss on OOD data. In real applications, the distribution $\mathcal{D}_{out}$ is unknown and difficult to be inferred precisely. Therefore, in practice, the OOD data (OE dataset, following $\mathcal{D}_{out}^{OE}$) fed into the detector can be the same as or disjoint from the test OOD data. Given a DNN f that learned the distribution $\mathcal{D}_{in}$ and an OE dataset, the objective of the OOD detector is to minimize:

$$\mathbb{E}_{(x,y)\sim\mathcal{D}_{in}}\left[\mathcal{L}\left(f\left(x\right),y\right)+0.5*\mathbb{E}_{x'\sim\mathcal{D}_{out}^{OE}}\left[\mathcal{L}_{OE}\left(f\left(x'\right)\right)\right]\right], \tag{3}$$

where $\mathcal{L}$ is the loss function of f. The OE loss function $\mathcal{L}_{OE}$ is set as the cross entropy from $f(x')$ to the uniform distribution. In particular, although learning from $\mathcal{D}_{out}^{OE}$, the OOD detector has been proved [1, 14] to generalize well to $\mathcal{D}_{out}$. Our experimental results in Table 3 (Section 4.4) also confirm this conclusion.

Concretely, given a pre-trained model f and its training set $X^{in}\sim\mathcal{D}_{in}$, first, we prepare the ID data and OOD data to train the OOD detector. For image datasets, we use all of the eight considered image mutation operators to mutate the training set and generate eight mutated sets. Then, we evenly select $\frac{|X^{in}|}{8}$ data from each mutated set and combine them as the OOD training set $X^{out}\sim\mathcal{D}_{out}^{OE}$. For the text datasets, we split the data from the OOD set as X^{out} directly. Note that the OOD data we select for training the OOD detector are not from the candidate set and test data. Next, an OE model is trained using both X^{in} and X^{out} according to Equation (3). This model predicts an OE score (probability) of a test being OOD. Finally, we train a regression classifier based on the OE scores of data in X^{in} and X^{out} predicted by the OE model. All OOD detectors in this article are available on our project site.[1]

Figure 5 shows the distribution of OE scores of three image candidate sets: MNIST, Fashion-MNIST, and CIFAR-10. For the description of candidate sets, please refer to Section 4. The blue and orange histograms represent the distributions of the ID and OOD data, respectively. For MNIST and Fashion-MNIST, the OOD detector can recognize and separate the ID and OOD data clearly, and the performance on CIFAR-10 is also acceptable. In addition, we calculate the **area under the curve of receiver characteristic operator (AUC-ROC)** score of our OOD detectors. The AUC-ROC provides an overall evaluation of the ability of the OOD detector to distinguish between OOD and ID data. A high AUC-ROC indicates good performance. For MNIST, the AUC-ROC scores are 87.74% and 92.69% of LeNet-1 and LeNet-5, respectively. For Fashion-MNIST, the scores are 88.62% and 90.81% of LeNet-1 and LeNet-5, respectively. For CIFAR-10, the scores are 74.52% and 74.36% of ResNet-20 and NiN, respectively.



(a) MNIST, LeNet-5 (b) Fashion-MNIST, LeNet-5 (c) CIFAR-10, ResNet-20

Fig. 5. Histograms of OE scores of image candidate sets.

6.2 DAT Algorithm

In Algorithm 1, we present our proposed DAT selection metric.

Basically, DAT includes five steps to select data:

(1) Given a candidate set X_c, we first utilize the OOD detector, $OOD_{Detector}$, to divide this dataset into ID and OOD sets, X_c^{id} and X_c^{out} (line 1). If the OOD score given by the $OOD_{Detector}$ is greater than (less than or equal to) δ, we say the input sample is OOD (ID) data. By default, we set δ to 0.5; however, the appropriate value depends on the used detector (we discuss this in more details later).

(2) From the results of CES, DSA, and Random selection in Table 7 and Figure 4, we know that the selected set, X_s, from X_c should follow a similar data distribution. Thus, we determine the labeling budgets, n^{id} and n^{out}, for the ID and OOD data in X_s by the proportion of ID data in X_c (lines 2 through 7). Note that in practice, we select slightly more OOD data like all of the selection metrics do because OOD data are more informative for a pre-trained DNN. Here, we use a pre-defined threshold δ to limit the amount of ID data used for retraining.

(3) We first select the ID data. According to our study, we try to select more uncertain data from ID set. As the result shown in Table 7 suggest, DeepGini is appropriate for this because it achieves the highest average top-1 performance among the uncertain-based metrics when the OOD data are below 70%. Thus, we apply DeepGini to select the most uncertain data, X^{id}, from X_c^{id} (line 8).

(4) To select the OOD data, we consider the class bias as suggested by RQ3. Using the test data, X_t, as a reference, we select OOD data within several iterations. In detail, first, we create the histogram, LD_t, of the predicted labels, Y_t, of X_t by the pre-trained DNN model f (lines 9 and 10). In each iteration, a set of OOD data, X_*^{out}, are randomly selected from X_c^{out} (line 13). Based on the distance of histograms between the selected and test sets, X^{out} is updated to be more balanced (lines 11 through 20).

(5) Output the combination of the selected ID and OOD data (lines 21 and 22).

6.3 RQ4: Effectiveness of DAT on Synthetic Distribution Shift

To evaluate the effectiveness of DAT, we conduct a similar comparison as RQ2. First, we consider the synthetic distribution shift. An important component in DAT is the OOD detector that determines the threshold δ to control the size of selected ID and OOD data. Generally, δ is set to 0.5, which means the data sample is OOD (ID) if the detector score of this sample is greater (smaller) than 0.5. However, we experimentally found out that it might be better to set a smaller δ to reduce the number of selected ID data. This is because the OOD detector may not be able to perfectly separate ID data from OOD data. In our experiments, we set δ as 0.01, 0.1, and 0.3 for MNIST, Fashion-MNIST, and CIFAR-10, respectively. In addition, we set the iteration number as 1,000 for

ALGORITHM 1: DAT: Distribution-Aware Test Selection

Input : $OOD_{Detector}$: out-of-distribution detector
f, X_t, X_c : DNN, test set, candidate set
$uncertainSelect$: uncertainty-based selection metric
δ: threshold to limit the size of selected ID data
n: size of labeling budget
ite: number of iterations

Output : X_s: selected data

/* Step1: Check data distribution of X_c */

$X_c^{in}, X_c^{out} = OE\_Detector(X_c, \delta)$

/* Step2: Determine data distribution of X_s */

if $\frac{|X_c^{in}|}{|X_c|} > \delta$ **then**
  $n^{in} = \delta \times n$
else
  $n^{in} = \delta \times \frac{|X_c^{in}|}{|X_c|}$
end
$n^{out} = n - n^{in}$

/* Step3: Select ID data */

$X^{in} = uncertainSelect\left(X_c^{in}, n^{in}\right)$

/* Step4: Select OOD data */

$Y_t = f\left(X_t\right)$
$LD_t = histogram\left(Y_t\right)$; // Histogram of labels
$d_{min} = \infty$
for $i = 0 \rightarrow ite$ **do**
  $X_*^{out} = randomSelect\left(X_c^{out}, n^{out}\right)$
  $LD_r = histogram\left(Y_r\right)$
  if $|LD_t - LD_r| < d_{min}$ **then**
    $X^{out} = X_*^{out}$
    $d_{min} = |LD_t - LD_r|$
  end
end

/* Step5: Output selected data */

$X_s = X^{id} \cup X^{out}$
return X^s

all datasets. For the backbone uncertainty metric that DAT uses to select ID data, we choose DeepGini as discussed before.

Table 9 lists the frequency of each selection metric achieving the top-1 and top-3 accuracy improvement over 64 and 192 cases, respectively. On average, DAT is the best metric regardless of the distribution shift and dataset. For example, in the case of "Top-1," DAT is five and two times better than the worst (Random) and the second-best (MCP), respectively. In the case of "Top-3," although the gap between metrics becomes smaller, DAT still achieves nearly 24% better than the second-best (MCP). Particularly, DAT always outperforms the others when there are more than 70% OOD data. In the other distribution ratios, there is no unique winner, but DAT remains generally competitive.

In addition, to check whether it is important and useful to consider the data distribution in DAT, we conduct an ablation study. Elaborately, we remove the distribution detection steps

Table 9. Effectiveness of DAT: Comparison of Frequency of Being the Top-1 and Top-3 Best of Seven Selection Metrics

	Distribution	Entropy	DeepGini	MCP	CES	DSA	Random	DAT	Entropy	DeepGini	MCP	CES	DSA	Random	DAT
	ID + OOD	Top-1							Top-3						
MNIST	0% + 100%	1	0	0	15	3	22	23	2	1	15	48	11	55	60
	10% + 90%	0	0	2	6	1	20	35	2	5	10	46	15	56	58
	20% + 80%	0	2	6	10	3	9	34	2	9	17	39	16	46	63
	30% + 70%	3	4	10	5	0	10	32	9	12	30	39	11	36	55
	40% + 60%	3	4	18	10	5	12	12	16	28	33	33	17	34	31
	50% + 50%	3	6	26	7	2	8	12	20	31	44	25	11	26	35
	60% + 40%	6	8	20	3	5	7	15	28	40	43	18	10	20	33
	70% + 30%	6	12	20	3	4	2	17	50	50	46	7	8	6	25
	80% + 20%	18	12	11	0	1	3	19	55	51	50	3	4	8	21
	90% + 10%	25	16	5	0	2	1	15	56	49	50	0	7	7	23
	100% + 0%	13	7	10	7	3	5	19	30	28	29	26	26	25	28
	Average	7.09	6.45	11.64	6.00	2.64	9.00	21.18	24.55	27.64	33.36	25.82	12.36	29.00	39.27
Fashion-MNIST	0% + 100%	0	1	3	10	8	16	26	1	3	7	45	30	52	54
	10% + 90%	1	0	4	8	8	15	28	1	0	11	40	31	50	59
	20% + 80%	0	0	3	6	8	15	32	1	2	11	35	33	52	58
	30% + 70%	0	2	4	7	8	12	31	4	2	20	38	28	46	54
	40% + 60%	0	0	10	11	6	14	23	3	5	29	40	33	31	51
	50% + 50%	0	2	10	4	5	10	33	5	9	31	40	25	31	51
	60% + 40%	5	7	18	2	1	7	24	14	25	38	20	18	31	46
	70% + 30%	6	12	20	1	1	5	19	26	34	47	17	11	21	36
	80% + 20%	13	15	16	2	0	1	17	40	37	45	12	11	19	28
	90% + 10%	21	14	16	1	1	2	9	49	53	52	6	7	12	13
	100% + 0%	4	7	4	3	2	5	39	31	32	29	18	23	13	46
	Average	4.55	5.45	9.82	7.45	4.27	6.91	25.55	15.91	18.36	29.09	32.09	26.00	25.45	45.09
CIFAR-10	0% + 100%	5	11	4	9	5	13	17	26	29	21	30	19	29	38
	10% + 90%	6	15	3	8	7	6	19	22	34	12	35	19	30	40
	20% + 80%	15	14	0	7	4	4	20	33	35	10	30	18	26	40
	30% + 70%	15	15	1	3	4	10	16	35	37	10	22	15	32	41
	40% + 60%	11	13	4	7	6	7	16	36	40	15	32	20	21	28
	50% + 50%	16	16	1	4	10	4	13	37	44	21	19	19	26	26
	60% + 40%	15	17	4	5	7	4	12	40	38	22	23	21	22	26
	70% + 30%	22	9	6	5	4	4	14	38	45	30	17	15	20	27
	80% + 20%	24	16	5	2	2	5	10	44	47	40	12	12	16	21
	90% + 10%	17	18	5	3	3	4	14	45	48	40	14	10	10	25
	100% + 0%	9	9	18	6	7	4	11	32	32	45	26	17	15	25
	Average	14.09	13.91	4.64	5.36	5.36	5.91	14.73	35.27	39.00	24.18	23.64	16.82	22.45	30.64
Total	0% + 100%	6	12	7	51	34	16	66	29	33	43	136	123	60	152
	10% + 90%	7	15	9	41	22	16	82	25	39	33	136	121	65	157
	20% + 80%	15	16	9	28	23	15	86	36	46	38	124	104	67	161
	30% + 70%	18	21	15	32	15	12	79	48	51	60	114	99	54	150
	40% + 60%	14	17	32	30	23	25	51	55	73	77	95	98	68	110
	50% + 50%	19	24	37	16	16	22	58	62	84	96	92	69	61	112
	60% + 40%	26	32	42	13	9	19	51	82	103	103	62	59	62	105
	70% + 30%	34	33	46	7	9	13	50	114	129	123	43	35	44	88
	80% + 20%	55	43	32	10	2	4	46	139	135	135	36	26	35	70
	90% + 10%	63	48	26	6	4	7	38	150	150	142	23	21	29	61
	100% + 0%	26	23	32	12	15	15	69	93	92	103	58	75	56	99
	Average	25.73	25.82	26.09	22.36	15.64	14.91	61.45	75.73	85.00	86.64	83.55	75.45	54.64	115.00

Note: The best result is highlighted in gray. "Distribution" represents different distribution shifts of the candidate set.

(steps 1 through 3) in Algorithm 1 and only use the fourth step to select all candidate data. In this way, DAT ignores the data distribution. Table 10 provides the results of our ablation study. Compared with taking into consideration the data distribution (Table 9), the performance drops a lot (presented by the numbers in brackets). On average, the frequencies of being the best top-1 and top-3 have reduced by 12.09 and 11, respectively. This ablation study demonstrates that considering data distribution is critical for DAT.

Answer to RQ4: On the synthetic distribution shift, when there are more OOD data in the new coming set (OOD data $\geq$ 70%), DAT outperforms other compared metrics in all of our considered datasets. In lower ratios of OOD, DAT is not always the best metric, but it remains competitive overall. In addition, our ablation study demonstrates the importance of taking into account the data distribution when selecting data.

Table 10. Ablation Study of DAT That Shows the Importance of Considering the Data Distribution

	Distribution	DAT with the OOD Detector		DAT without the OOD Detector		Improvement Drop	
	ID + OOD	Top-1	Top-3	Top-1	Top-3	Top-1	Top-3
MNIST	0% + 100%	23	60	19	57	4	3
	10% + 90%	35	58	24	56	11	2
	20% + 80%	34	63	25	57	9	6
	30% + 70%	32	55	25	50	7	5
	40% + 60%	12	31	8	28	4	3
	50% + 50%	12	35	10	33	2	2
	60% + 40%	15	33	14	30	1	3
	70% + 30%	17	25	14	25	3	0
	80% + 20%	19	21	17	20	2	1
	90% + 10%	15	23	14	22	1	1
	100% + 0%	19	28	16	26	3	2
	Average	21.18	39.27	16.91	36.73	4.27	2.55
Fashion-MNIST	0% + 100%	26	54	20	51	6	3
	10% + 90%	28	59	22	55	6	4
	20% + 80%	32	58	30	53	2	5
	30% + 70%	31	54	31	54	0	0
	40% + 60%	23	51	17	49	6	2
	50% + 50%	33	51	25	47	8	4
	60% + 40%	24	46	20	45	4	1
	70% + 30%	19	36	18	30	1	6
	80% + 20%	17	28	13	25	4	3
	90% + 10%	9	13	7	10	2	3
	100% + 0%	39	46	26	38	13	8
	Average	25.55	45.09	20.82	41.55	4.73	3.55
CIFAR-10	0% + 100%	17	38	16	30	1	8
	10% + 90%	19	40	12	33	7	7
	20% + 80%	20	40	16	33	4	7
	30% + 70%	16	41	11	31	5	10
	40% + 60%	16	28	11	24	5	4
	50% + 50%	13	26	9	21	4	5
	60% + 40%	12	26	10	22	2	4
	70% + 30%	14	27	12	24	2	3
	80% + 20%	10	21	10	20	0	1
	90% + 10%	14	25	12	23	2	2
	100% + 0%	11	25	9	22	2	3
	Average	14.73	30.64	11.64	25.73	3.09	4.91
Total	0% + 100%	66	152	55	138	11	14
	10% + 90%	82	157	58	144	24	13
	20% + 80%	86	161	71	143	15	18
	30% + 70%	79	150	67	135	12	15
	40% + 60%	51	110	36	101	15	9
	50% + 50%	58	112	44	101	14	11
	60% + 40%	51	105	44	97	7	8
	70% + 30%	50	88	44	79	6	9
	80% + 20%	46	70	40	65	6	5
	90% + 10%	38	61	33	55	5	6
	100% + 0%	69	99	51	86	18	13
	Average	61.45	115.00	49.36	104.00	12.09	11.00

"Distribution" represents different distribution shifts of the candidate set. "Improvement drop" presents the drop of accuracy (%) improvement of DAT without the OOD detector compared with using the OOD detector when selecting data. Baseline: Please refer to Table 4 for the accuracy of pre-trained DNNs.

6.4 RQ5: Effectiveness on Natural Distribution Shifts

In addition to testing on synthetic distribution shifts, we further evaluate DAT on natural distribution shifts. In our study, we consider three datasets: iWildCam, IMDb, and Newsgroups.

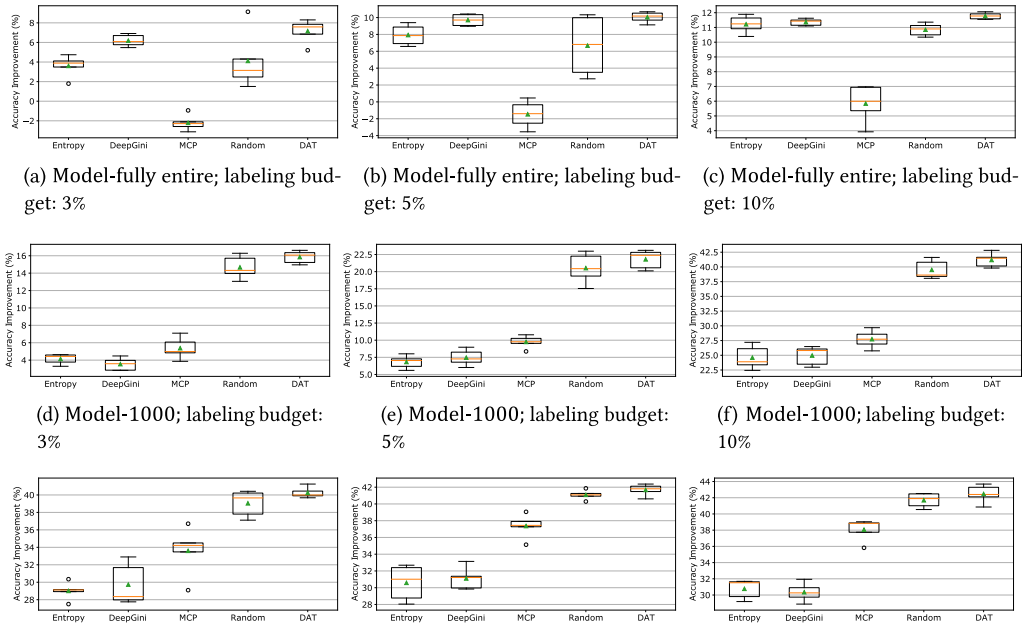
Datasets. iWildCam is from a recently released benchmark with real-world distribution shifts—WILDS [17]. The shift of iWildCam comes from camera traps. Concretely, researchers collect data

using specific camera traps, then use these data to train an ML model for animal recognition. However, when users deploy this model in the wild, the change of camera traps may cause distribution shifts and harm the performance of the model. In total, iWildCam contains 129,809 training data (ID), 14,961 OOD validation data, 7,314 ID validation data, and 42,791 OOD test data. The data are divided into 182 different categories. Please refer to Section 4.2 for details on IMDb and Newsgroups.

Setup. For iWildCam, we use all training data to train a ResNet-50 model as our pre-trained model. We chose ResNet-50 because it is the recommended model architecture by the WILDS benchmark. Then, we randomly split the test data (all of which are OOD) into three parts: one (20,000 data) for training the OOD detector with the ID training data, one (10,000 data) as the candidate set for selection, and the rest (12,791 data) as the test data. In addition, we follow setup simular to that in the work of Kossen et al. [18], which reduces the number of training data to check the performance of each metric on the model that has bad performance, to train models with a small number of training data. In this way, we can check the effectiveness of each metric on both the well-trained model and the model trained by limited labeled data. Thus, we train the other two models using randomly selected 1,000 and 2,000 training data for our evaluation. For IMDb and Newsgroups, we follow the same procedure as iWildCam to split the OOD data into the training data for the OOD detector, the candidate set for selecting, and the test set for evaluation, respectively. After the preparation, we employ different selection metrics to select the candidate data and retrain the pre-trained models. Finally, we record the test accuracy improvement on the test data before and after retraining. This setting is the same as the (0% + 100%) distribution combination in the previous research questions. The AUC-ROC scores of the OOD detectors we trained for this RQ are 79.77%, 68.87%, 70.44%, 82.09%, and 77.37% for iWildCam-ResNet50, IMDb-LSTM, IMDb-GRU, Newsgroups-NN, and Newsgroups-NN2, respectively. We set the δ in Algorithm 1 for iWildCam, IMDb, and Newsgroups as 0.5, 0.5, and 0.1, respectively.

Results. Figure 6 depicts the accuracy improvement on the test data by using each metric to select (3%, 5%, and 10%) candidate data for model retraining. It is worth noting that since both DSA and CES cause out-of-memory problems, we cannot run these two metrics on the iWildCam dataset. In the figure, Model-fully, Model-1000, and Model-2000 represent the model that is pre-trained by all training data, 1,000 training data, and 2,000 training data, respectively. The test accuracy of each model on the test data before retraining is 70.85%, 32.94%, and 35.53%, respectively. From the results, we can see that DAT outperforms the other metrics in all cases. Specifically, on average, DAT can improve test accuracy by 9.25%, 8.60%, 8.65%, and 1.61% more than Entropy, DeepGini, MCP, and Random. When the model is well trained (using all training data), in addition to DAT, DeepGini is a promising metric. However, when the model is trained by limited training data, DeepGini, Entropy, and MCP perform much worse than Random selection and DAT. Compared with the Random selection, in addition to the higher test accuracy improvement, DAT is more stable, and the standard deviation of DAT (0.83) is 47% lower than the Random selection (1.56).

Figure 7 shows the results of IMDb and Newsgroups. In most cases (10 out of 12), DAT outperforms the other metrics. On average, for IMDb, DAT can improve the test accuracy by 0.42%, 0.55%, 0.51%, 1.07%, 2.66%, and 0.44% more than Entropy, DeepGini, MCP, Random, CES, and DSA, respectively. The test accuracy improvement seems to be insignificant. We checked the models retrained using all of the new data, and the test accuracy improvement is less than 3%. One possible reason is that the natural OOD data for IMDb is similar to the ID data. For Newsgroups, DAT improves the test accuracy by 9.85%, 11.12%, 9.46%, 4.75%, 30.10%, and 8.77% more than Entropy, DeepGini, MCP, Random, CES, and DSA, respectively. Additionally, in this dataset, the Random selection defeats the other metrics except for DAT, which is consistent with our conclusion in RQ2.



(a) Model-fully entire; labeling budget: 3%
(b) Model-fully entire; labeling budget: 5%
(c) Model-fully entire; labeling budget: 10%
(d) Model-1000; labeling budget: 3%
(e) Model-1000; labeling budget: 5%
(f) Model-1000; labeling budget: 10%
(g) Model-2000; labeling budget: 3%
(h) Model-2000; labeling budget: 5%
(i) Model-2000; labeling budget: 10%

Fig. 6. Box plot of the accuracy improvement of different selection metrics on the dataset iWildCam, DNN ResNet-50. The pre-trained models are learned by using the entire training set (first row), 1,000 data (second row), and 2,000 data (third row), respectively. The budgets for retraining are 3% (first column), 5% (second column), and 10% (third column), respectively.

Answer to RQ5: In the three datasets with real-world distribution shifts, DAT outperforms existing selection metrics by up to 30.10% test accuracy improvement after retraining.

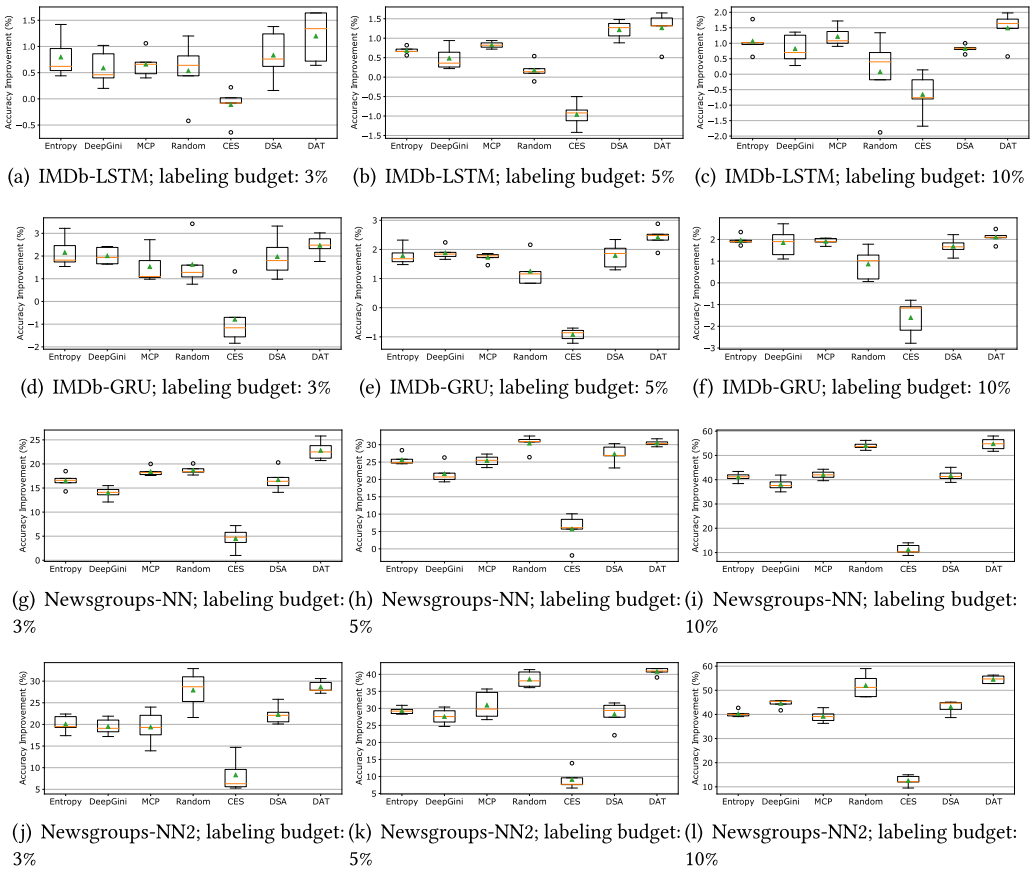
7 DISCUSSION

Based on our study, we first highlight our novel findings and research guidance, then discuss the threats to the validity of our work.

7.1 Novel Findings and Research Guidance

(1) *Retraining process*: Both retraining strategies for model enhancement (only using the selected new data and merging the new data with training data to process retraining) are commonly used in the literature. According to our comprehensive comparison (RQ1), the second process works better. Indeed, only using the new data can improve the accuracy on the new test data; however, the accuracy on the original test set is greatly sacrificed, especially when the new data include more OOD than ID data. By contrast, combining the original training data and new data to retrain a DNN can enhance the performance on the new data, meanwhile maintaining the high accuracy on the original test set.

Research guidance. Based on our experimental results, retraining DNNs by adding new data to the original training set is a better option. There is still room for improving this process. For example, how much original training data is really necessary? Can we reduce the size of the original data to achieve better efficiency? Instead of only selecting new data,



(a) IMDb-LSTM; labeling budget: 3% (b) IMDb-LSTM; labeling budget: 5% (c) IMDb-LSTM; labeling budget: 10%

(d) IMDb-GRU; labeling budget: 3% (e) IMDb-GRU; labeling budget: 5% (f) IMDb-GRU; labeling budget: 10%

(g) Newsgroups-NN; labeling budget: 3% (h) Newsgroups-NN; labeling budget: 5% (i) Newsgroups-NN; labeling budget: 10%

(j) Newsgroups-NN2; labeling budget: 3% (k) Newsgroups-NN2; labeling budget: 5% (l) Newsgroups-NN2; labeling budget: 10%

Fig. 7. Box plot of the accuracy improvement of different selection metrics on the dataset IMDb and Newsgroups.

proposing a metric to carefully select both the original training data and new data for retraining might be a promising research direction.

(2) *Test selection under different data distributions*: Our experiments have demonstrated that none of the existing selection metrics (Entropy, DeepGini, MCP, CES, DSA, and Random) can always outperform others under different data distributions. Most of them (Entropy, DeepGini, and MCP) can select useful data for model retraining when the new data mostly contains the ID data. However, for the contrary case where OOD data occupy more in the new data, they fail to win against the Random selection. To deal with this specific case, we propose the distribution-aware metric, DAT, and it has been proved to be effective.

Research guidance. For model retraining, in the case of more ID data existing in the new data, uncertain-based metrics are better options, whereas when OOD data are more than ID, our metric DAT can alleviate the influence of distribution shifts and outperform other metrics. Before choosing a metric, the distribution of ID and OOD data should first be checked by some methods (e.g., OOD detector). However, it is still challenging to develop an almighty metric that can deal with all data distributions. A promising solution could be considering multiple existing metrics strategically in the retraining process based on the distribution of new data.

(3) *Data distribution and bias of selected data*: In terms of data distribution, since OOD data are more likely than ID data to be unlearned by pre-trained DNNs, the uncertain-based selection metrics (Entropy, DeepGini, and MCP) choose more OOD data for retraining under all of the different distributions, whereas CES, DSA, and Random selection can follow almost the same data distribution of the candidate data to pick data. However, concerning the class bias of the selected data, CES and Random selection seem to make a good balance among different classes. Yet there is no clear clue to show that using more OOD or balanced data is more helpful in model enhancement.

Research guidance. Concerning the importance of data distribution and class bias, it could be promising to further improve the effectiveness of uncertainty-based metrics (Entropy, DeepGini, and MCP) by considering these two factors. In addition, we observe that most selected data by the uncertainty-based metrics are misclassified by pre-trained DNNs. Thus, the prediction accuracy of the selected data might be another factor that impacts the retraining performance.

7.2 Threats to Validity

The external threat to validity mainly comes from the DNN models and datasets used in our study. Regarding the datasets, we consider six commonly used and public datasets in the literature. To reduce the threat from the DNN models, we employ two well-known architectures for each dataset (except iWildCam, in which we use the state-of-the-art model recommended by the WILDS benchmark) to limit the impact of model dependency to some extent. Our datasets include both image and text data, and our models cover both FNN and RNN.

The internal threat could be caused by the implementation of DAT and the selection metrics in comparison. To counter this issue, we borrow the available implementations of the compared methods from released codes by their authors and carefully check our code.

The construct threat lies in the OOD detector in DAT, the hyperparameter setting, and the randomness in the training process. Following the guidance of a comprehensive empirical study [1], we incorporate the current-best OOD detector into our new metric. In addition, we utilize their implementation and recommend settings to train our OOD detectors. We believe that with a better OOD detector, our method will achieve better results as well. For the hyperparameter, δ is important since it determines how to consider data as ID or OOD. It also limits the ratio of ID data selected for retraining. By default, we set δ to 0.5—that is, up to 50% of the selected data can come from the original distribution. This default ratio worked well for real-world datasets (WILDS and IMDb). For the other datasets, we experimentally found out that setting a lower δ increases the effectiveness of retraining with DAT. Ultimately, at the cost of additional labeling, we can improve the effectiveness of DAT through the setting of a better δ value. In practical applications, this opens the perspective to determine better δ values from past distribution shifts. Regarding the randomness, we repeat each experiment five times and retrained more than 71,280 models.

8 RELATED WORK

We review related works in three aspects: test selection in DL systems, distribution-aware DL testing, and empirical study for DL systems.

Test selection in DL systems. In the literature, many selection metrics have been proposed to reduce the labeling effort. Based on the similarity between the training set and test data, Kim et al. [16] proposed the surprise-guided testing metrics for model retraining. DeepGini [8] was proposed to prioritize the test data and select the most informative data that are more likely to be misclassified by the model. Its authors have also demonstrated that DeepGini is useful to guide the model retraining. MCP [38] is another uncertainty-based selection metric. It selects data close to

the decision boundaries by the top-2 predicted probabilities. Wang et al. [47] proposed robustness-oriented testing metrics as well as selection metrics. However, their objective is the adversarial robustness of DNNs, which is different from our study. Thus, those metrics are not considered in our article. Wang et al. [48] proposed a new selection metric that uses image mutation and DNN model mutation to select data that are likely to be misclassified by the model for revealing DNN bugs. Recently, Guo et al. [12] proposed a novel active learning approach that can train a more robust model. Meanwhile, they demonstrated that this approach can be used for test selection based model enhancement.

In our work, we study all selection metrics that are proposed for selecting data and enhancing the model. We also propose a novel selection metric (DAT). Different from existing metrics, DAT is the first one to consider the data distribution in test selection.

Distribution-aware DL testing. Recently, researchers have revealed that data distribution might impact DL testing, especially in the scenario of test generation. Berend et al. [1] conducted a comprehensive empirical study to explore the relationship between DL testing criteria and data distribution. In addition, they provided some research guidance—for example, DL testing tools should be aware of distribution. Different from their study, our work mainly focuses on how distribution affects test selection.

Dola et al. [6] proposed a distribution-aware test generation method that is based on variational auto-encoder (VAE). They first studied the validity of the data generated by existing test generation methods (e.g., DeepXplore), then proposed the test generation method to check if the generated data are valid or not at the generation time. Different from their work, we focus on how to select data with the distribution information rather than generating test data.

Empirical study for DL systems. DL systems are continuously adopted in many SE applications (e.g., DL for code function prediction). SE researchers pay more attention to study DL systems, and a few empirical studies have been conducted. Ma et al. [27] performed a comparison study on different selection metrics for testing DL systems. They revealed that neuron coverage based selection metrics cannot achieve competitive results, and more efficient metrics are on demand. Guo et al. [11] studied the performance difference between different DL frameworks as well as the model changes after model migration. Zhang et al. [54] conducted a comparative study to explore how different uncertainty metrics distinguish adversarial examples from benign examples. Hu et al. [15] empirically explored the limitations of active learning, which is a commonly used training process for both SE and ML tasks. In addition, a series of works [4, 53] have been performed to study the challenges in deploying DL systems.

Compared with the existing empirical study works, our study investigates the potential problems in test selection for model enhancement that is missing in the literature.

9 CONCLUSION

In this article, we first conducted a systemically empirical study to explore how different retraining processes and data distributions impact the test selection for model enhancement. In total, based on six selection metrics in comparison, we retrained 71,280 models over five popular datasets and 10 DNN models for our empirical study. In terms of enhancing the performance on new data under various distributions and meanwhile maintaining the high accuracy on the original test set, our experimental results reveal that using the combination of training and selected data is better than only using the selected data. In addition, none of the existing selection metrics can always outperform the others across all data distributions. Interestingly, when the new set contains more ($\geq$70%) OOD data, the simple but effective random manner defeats the others, which gives us an insight that this special case has not been uncovered in existing metrics. Thus, based on the findings, we propose DAT, a novel and effective distribution-aware metric, to deal with this case. In

the experiments, we compared DAT with our studied test selection metrics, the results demonstrate that DAT outperforms other metrics by up to five times better for model enhancement when deals the synthetic distribution shift. Besides, the results on the datasets with natural distribution shifts also prove that DAT can achieve better model enhancement than the other metrics when facing the in-the-wild scenario. Moreover, based on our findings from the five research questions, we open research directions for further improving the performance of existing metrics as well as proposing new selection metrics.

REFERENCES

[1] David Berend, Xiaofei Xie, Lei Ma, Lingjun Zhou, Yang Liu, Chi Xu, and Jianjun Zhao. 2020. Cats are not fish: Deep learning testing calls for out-of-distribution awareness. In *Proceedings of the 35th IEEE/ACM International Conference on Automated Software Engineering (ASE'20)*. ACM, New York, NY, 1041–1052. https://doi.org/10.1145/3324884.3416609

[2] Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D. Jackel, et al. 2016. End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316* (2016).

[3] Junjie Chen, Zhuo Wu, Zan Wang, Hanmo You, Lingming Zhang, and Ming Yan. 2020. Practical accuracy estimation for efficient deep neural network testing. *ACM Transactions on Software Engineering and Methodology* 29, 4 (October 2020), Article 30, 35 pages. https://doi.org/10.1145/3394112

[4] Zhenpeng Chen, Yanbin Cao, Yuanqiang Liu, Haoyu Wang, Tao Xie, and Xuanzhe Liu. 2020. A comprehensive study on challenges in deploying deep learning based software. In *Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*. 750–762.

[5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. ImageNet: A large-scale hierarchical image database. In *Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, Los Alamitos, CA, 248–255.

[6] Swaroopa Dola, Matthew B. Dwyer, and Mary Lou Soffa. 2021. Distribution-aware testing of neural networks using generative models. In *Proceedings of the IEEE/ACM 43nd International Conference on Software Engineering (ICSE'21)*. IEEE, Los Alamitos, CA.

[7] Xiaoning Du, Xiaofei Xie, Yi Li, Lei Ma, Yang Liu, and Jianjun Zhao. 2019. DeepStellar: Model-based quantitative analysis of stateful deep learning systems. In *Proceedings of the 2019 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*. 477–487.

[8] Yang Feng, Qingkai Shi, Xinyu Gao, Jun Wan, Chunrong Fang, and Zhenyu Chen. 2020. DeepGini: Prioritizing massive tests to enhance the robustness of deep neural betworks. In *Proceedings of the 29th ACM SIGSOFT International Symposium on Software Testing and Analysis (ISSTA'20)*. ACM, New York, NY, 177–188. https://doi.org/10.1145/3395363.3397357

[9] Bent Fuglede and Flemming Topsoe. 2004. Jensen-Shannon divergence and Hilbert space embedding. In *Proceedings of the International Symposium on Information Theory (ISIT'04)*. IEEE, Los Alamitos, CA, 31.

[10] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and harnessing adversarial examples. In *Proceedings of the International Conference on Learning Representations*. http://arxiv.org/abs/1412.6572.

[11] Qianyu Guo, Sen Chen, Xiaofei Xie, Lei Ma, Qiang Hu, Hongtao Liu, Yang Liu, Jianjun Zhao, and Xiaohong Li. 2019. An empirical study towards characterizing deep learning development and deployment across different frameworks and platforms. In *Proceedings of the 2019 34th IEEE/ACM International Conference on Automated Software Engineering (ASE'19)*. IEEE, Los Alamitos, CA, 810–822.

[12] Yuejun Guo, Qiang Hu, Maxime Cordy, Mike Papadakis, and Yves Le Traon. 2021. Robust active learning: Sample-efficient training of robust deep learning models. *arXiv preprint arXiv:2112.02542* (2021).

[13] Dan Hendrycks and Kevin Gimpel. 2016. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *arXiv preprint arXiv:1610.02136* (2016).

[14] Dan Hendrycks, Mantas Mazeika, and Thomas Dietterich. 2019. Deep anomaly detection with outlier exposure. In *Proceedings of the International Conference on Learning Representations*. https://openreview.net/forum?id=HyxCxhRcY7.

[15] Qiang Hu, Yuejun Guo, Maxime Cordy, Xie Xiaofei, Wei Ma, Mike Papadakis, and Yves Le Traon. 2021. Towards exploring the limitations of active learning: An empirical study. In *Proceedings of the 36th IEEE/ACM International Conference on Automated Software Engineering*.

[16] Jinhan Kim, Robert Feldt, and Shin Yoo. 2019. Guiding deep learning system testing using surprise adequacy. In *Proceedings of the 41st International Conference on Software Engineering (ICSE'19)*. IEEE, Los Alamitos, CA, 1039–1049. https://doi.org/10.1109/ICSE.2019.00108

[17] Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, et al. 2020. Wilds: A benchmark of in-the-wild distribution shifts. *arXiv preprint arXiv:2012.07421* (2020).
[18] Jannik Kossen, Sebastian Farquhar, Yarin Gal, and Tom Rainforth. 2021. Active testing: Sample-efficient model evaluation. *arXiv preprint arXiv:2103.05331* (2021).
[19] Alex Krizhevsky. 2009. *Learning Multiple Layers of Features from Tiny Images*. Technical Report. University of Toronto.
[20] Ken Lang. 1995. NewsWeeder: Learning to filter netnews. In *Machine Learning Proceedings 1995*. Elsevier, 331–339.
[21] Yann Lecun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* 86, 11 (Nov. 1998), 2278–2324.
[22] Kimin Lee, Honglak Lee, Kibok Lee, and Jinwoo Shin. 2018. Training confidence-calibrated classifiers for detecting out-of-distribution samples. In *Proceedings of the International Conference on Learning Representations*. https://openreview.net/forum?id=ryiAv2xAZ.
[23] Xin Li and Yuhong Guo. 2013. Adaptive active learning for image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 859–866. https://doi.org/10.1109/CVPR.2013.116
[24] Zenan Li, Xiaoxing Ma, Chang Xu, Chun Cao, Jingwei Xu, and Jian Lü. 2019. Boosting operational DNN testing efficiency through conditioning. In *Proceedings of the 2019 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE'19)*. ACM, New York, NY, 499–509. https://doi.org/10.1145/3338906.3338930
[25] Shiyu Liang, Yixuan Li, and R. Srikant. 2018. Enhancing the reliability of out-of-distribution image detection in neural networks. In *Proceedings of the International Conference on Learning Representations*.
[26] Lei Ma, Felix Juefei-Xu, Fuyuan Zhang, Jiyuan Sun, Minhui Xue, Bo Li, Chunyang Chen, et al. 2018. DeepGauge: Multi-granularity testing criteria for deep learning systems. In *Proceedings of the 33rd ACM/IEEE International Conference on Automated Software Engineering (ASE'18)*. ACM, New York, NY, 120–131. https://doi.org/10.1145/3238147.3238202
[27] Wei Ma, Mike Papadakis, Anestis Tsakmalis, Maxime Cordy, and Yves Le Traon. 2021. Test selection for deep learning systems. *ACM Transactions on Software Engineering and Methodology* 30, 2 (2021), 1–22.
[28] Andrew Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. 142–150.
[29] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2018. Towards deep learning models resistant to adversarial attacks. In *Proceedings of the International Conference on Learning Representations*.
[30] Kayo Matsushita, Kayo Matsushita, and Hasebe. 2018. *Deep Active Learning*. Springer.
[31] Glenford J. Myers, Tom Badgett, Todd M. Thomas, and Corey Sandler. 2004. *The Art of Software Testing*. Vol. 2. Wiley Online Library.
[32] Kexin Pei, Yinzhi Cao, Junfeng Yang, and Suman Jana. 2017. DeepXplore: Automated whitebox testing of deep learning systems. In *Proceedings of the 26th Symposium on Operating Systems Principles (SOSP'17)*. ACM, New York, NY, 1–18. https://doi.org/10.1145/3132747.3132785
[33] Jie Ren, Peter J. Liu, Emily Fertig, Jasper Snoek, Ryan Poplin, Mark A. DePristo, Joshua V. Dillon, and Balaji Lakshminarayanan. 2019. Likelihood ratios for out-of-distribution detection. *arXiv preprint arXiv:1906.02845* (2019).
[34] Kui Ren, Tianhang Zheng, Zhan Qin, and Xue Liu. 2020. Adversarial attacks and defenses in deep learning. *Engineering* 6, 3 (2020), 346–360. https://doi.org/10.1016/j.eng.2019.12.012
[35] Pál Révész. 2005. *Random Walk in Random and Non-Random Environments* (2nd ed.). World Scientific. https://doi.org/10.1142/5847
[36] Joan Serrà, David Álvarez, Vicenç Gómez, Olga Slizovskaia, José F. Núñez, and Jordi Luque. 2020. Input complexity and out-of-distribution detection with likelihood-based generative models. In *Proceedings of the International Conference on Learning Representations*.
[37] C. E. Shannon. 1948. A mathematical theory of communication. *Bell System Technical Journal* 27, 3 (1948), 379–423. https://doi.org/10.1002/j.1538-7305.1948.tb01338.x
[38] Weijun Shen, Yanhui Li, Lin Chen, Yuanlei Han, Yuming Zhou, and Baowen Xu. 2020. Multiple-boundary clustering and prioritization to promote neural network retraining. In *Proceedings of the IEEE/ACM International Conference on Automated Software Engineering (ASE'20)*. IEEE, Los Alamitos, CA, 410–422.
[39] Yi Sun, Ding Liang, Xiaogang Wang, and Xiaoou Tang. 2015. DeepID3: Face recognition with very deep neural networks. *CoRR* abs/1502.00873 (2015). http://dblp.uni-trier.de/db/journals/corr/corr1502.html#SunLWT15.
[40] Youcheng Sun, Min Wu, Wenjie Ruan, Xiaowei Huang, Marta Kwiatkowska, and Daniel Kroening. 2018. Concolic testing for deep neural networks. In *Proceedings of the 33rd ACM/IEEE International Conference on Automated Software Engineering*. 109–119.
[41] Richard Szeliski. 2010. *Computer Vision: Algorithms and Applications*. Springer-Verlag, Berlin, Germany.

[42] Rohan Taori, Achal Dave, Vaishaal Shankar, Nicholas Carlini, Benjamin Recht, and Ludwig Schmidt. 2020. Measuring robustness to natural distribution shifts in image classification. *arXiv preprint arXiv:2007.00644* (2020).

[43] Yuchi Tian, Kexin Pei, Suman Jana, and Baishakhi Ray. 2018. DeepTest: Automated testing of deep-neural-network-driven autonomous cars. In *Proceedings of the 40th International Conference on Software Engineering (ICSE'18)*. ACM, New York, NY, 303–314. https://doi.org/10.1145/3180155.3180220

[44] Oriol Vinyals, Igor Babuschkin, Wojciech M. Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H. Choi, et al. 2019. Grandmaster level in Starcraft II using multi-agent reinforcement learning. *Nature* 575, 7782 (2019), 350–354.

[45] Daisuke Wakabayashi. 2018. Self-driving Uber car kills pedestrian in Arizona, where robots roam. New York Times. Retrieved April 25, 2022 from https://www.nytimes.com/2018/03/19/technology/uber-driverless-fatality.html.

[46] Dan Wang and Yi Shang. 2014. A new active labeling method for deep learning. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN'14)*. IEEE, Los Alamitos, CA, 112–119.

[47] Jingyi Wang, Jialuo Chen, Youcheng Sun, Xingjun Ma, Dongxia Wang, Jun Sun, and Peng Cheng. 2021. RobOT: Robustness-oriented testing for deep learning systems. In *Proceedings of the 43rd International Conference on Software Engineering (ICSE'21)*. IEEE, Los Alamitos, CA.

[48] Zan Wang, Hanmo You, Junjie Chen, Yingyi Zhang, Xuyuan Dong, and Wenbin Zhang. 2021. Prioritizing test inputs for deep neural networks via mutation analysis. In *Proceedings of the IEEE/ACM 43nd International Conference on Software Engineering (ICSE'21)*.

[49] Geoffrey I. Webb, Loong Kuan Lee, François Petitjean, and Bart Goethals. 2017. Understanding concept drift. *CoRR* abs/1704.00362 (2017). http://arxiv.org/abs/1704.00362.

[50] Eric Wong, Leslie Rice, and J. Zico Kolter. 2019. Fast is better than free: Revisiting adversarial training. In *Proceedings of the International Conference on Learning Representations*.

[51] Han Xiao, Kashif Rasul, and Roland Vollgraf. 2017. Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms. *arXiv:cs.LG/1708.07747* [cs.LG] (2017).

[52] Wayne Xiong, Jasha Droppo, Xuedong Huang, Frank Seide, Mike Seltzer, Andreas Stolcke, Dong Yu, and Geoffrey Zweig. 2016. Achieving human parity in conversational speech recognition. *arXiv preprint arXiv:1610.05256* (2016).

[53] Tianyi Zhang, Cuiyun Gao, Lei Ma, Michael Lyu, and Miryung Kim. 2019. An empirical study of common challenges in developing deep learning applications. In *Proceedings of the International Symposium on Software Reliability Engineering (ISSRE'19)*. IEEE, Los Alamitos, CA, 104–115. https://doi.org/10.1109/ISSRE.2019.00020

[54] Xiyue Zhang, Xiaofei Xie, Lei Ma, Xiaoning Du, Qiang Hu, Yang Liu, Jianjun Zhao, and Meng Sun. 2020. Towards characterizing adversarial defects of deep learning software from the lens of uncertainty. In *Proceedings of the IEEE/ACM 42nd International Conference on Software Engineering (ICSE'20)*. IEEE, Los Alamitos, CA, 739–751.

[55] Zixun Zhang, Zhen Li, Lin Lin, Na Lei, Guanbin Li, and Shuguang Cui. 2020. MetaSelection: Metaheuristic substructure selection for neural network pruning using evolutionary algorithm. *Frontiers in Artificial Intelligence and Applications* 325 (2020), 2808–2815.

Received June 2021; revised January 2022; accepted January 2022