Data Acquisition for Domain Adaptation of Closed-Box Models

Yiwei Liu
York University
Canada
e-well@outlook.com

Xiaohui Yu
York University
Canada
xhyu@yorku.ca

Nick Koudas
University of Toronto
Canada
koudas@cs.toronto.edu

***Abstract*—Machine learning (ML) marketplaces, pivotal for numerous industries, often offer models to customers as "closed boxes". These models, when deployed in new domains, might experience lower performance due to distributional shifts. Our paper proposes a framework designed to enhance closed-box classification models. This framework allows customers, upon detecting performance gaps on their validation datasets, to gather additional data for creating an auxiliary "padding" model. This model assists the original closed-box model in addressing classification weaknesses in the target domain. The framework includes a "weakness detector" that identifies areas where the model falls short and an Augmented Ensemble method that combines the original and padding models to improve accuracy and expand the diversity of the ML marketplace. Extensive experiments on several popular benchmark datasets confirm the superiority of our proposed framework over baseline approaches.**

I. Introduction

As more industries adopt machine learning (ML) techniques to facilitate their business, ML marketplaces are becoming a major resource for solutions [13]. Such marketplaces often take the form of data markets and/or model markets. Data markets such as Amazon Data Exchange [2], Databricks Marketplace [5] offer datasets for sale, while a model market can take a variety of forms, e.g., offering the use of ML models through APIs (e.g., Amazon [1] and Google Cloud Vision [6]), or providing downloadable versions of the models with different licensing terms (e.g., Open Model Zoo [14]).

However, a direct application of models from the market would lead to performance degradation due to distribution discrepancy. Figure 1 shows an example of this problem. The current closed-box classifier is to distinguish class *large mammals* and class *medium-sized mammals*. However, when the customer applies this classifier to the target data set containing *wolf* images for class *large mammals*, and *fox* images for *medium-sized mammals* (two categories of images that were not part of the training data for the model), the model will likely exhibit poor accuracy on such samples. It is therefore of critical importance to study how to improve the model and its accuracy in these new categories. Interestingly, if a customer can obtain an improved model, they can introduce this new model back to the model market for resale, further contributing to the diversity of the market.

A promising approach recently proposed to improve the accuracy of the ML model accuracy is to acquire suitable data from a data market to fine-tune the model [10]. However, it does not apply to settings where the models are closed boxes, as neither the model parameters nor their training data are accessible, making it impossible to fine-tune the model parameters based on the newly acquired data. Moreover, it is often not sensible cost-wise to train an entirely new model on the acquired data [12] [19], because models from the market are built from a large collection of data, but new categories described above account for only a small percentage of the target domain. Instead, it is more viable to repurpose the existing model and "augment" it externally to derive a model suitable for the target domain.

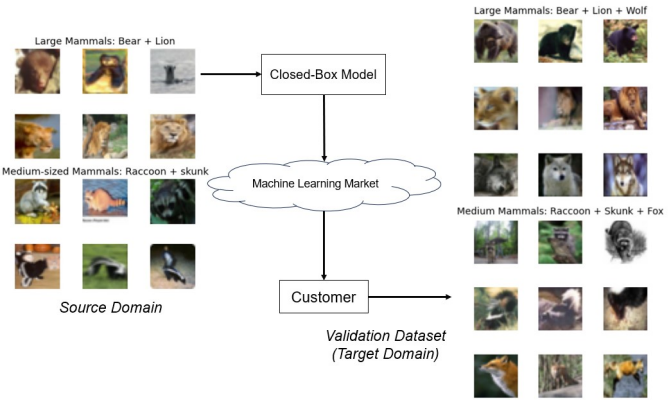

Fig. 1: Closed-Box Model with Shifted Target Domain

In this paper, we focus on classification models and propose a framework based on this idea of model augmentation for closed-box models. When a customer detects performance degradation (relative to the model's advertised accuracy) for a given model $\mathcal{M}_\mathbf{s}$ on their validation dataset drawn from the target domain, they can procure data from a data market of choice, subject to a stipulated budget, to build a new "padding" model $\mathcal{M}_\mathbf{p}$. This model aims to exhibit improved accuracy, particularly in data that the source model $\mathcal{M}_\mathbf{s}$ does not handle well.

Two challenges arise when establishing such a framework. The first challenge is to effectively identify the data samples to acquire from the data market that would make the best use of the available budget; the second challenge is how to combine $\mathcal{M}_\mathbf{p}$ and $\mathcal{M}_\mathbf{s}$ to obtain a model $\mathcal{M}_\mathbf{t}$ with improved accuracy in the target domain. For the first challenge, we build a "weakness

detector", WEDE responsible for detecting when $\mathcal{M}_\mathbf{s}$ does not perform well. WEDE is essentially a classifier based on the customer's validation data that are correctly or incorrectly classified by $\mathcal{M}_\mathbf{s}$. We use its output on any data instance from the data market as an estimate of its utility to the model accuracy improvement. Different methods of acquiring data are also discussed, which vary in the frequency of acquisition and classifier update. For the second challenge, we propose an approach named Augmented Ensemble (AE) that constructs $\mathcal{M}_\mathbf{t}$ as an ensemble of $\mathcal{M}_\mathbf{s}$ and $\mathcal{M}_\mathbf{p}$. Compared to using only $\mathcal{M}_\mathbf{s}$ or $\mathcal{M}_\mathbf{p}$ in $\mathbf{T}$, $\mathcal{M}_\mathbf{t}$ enhances the overall performance by leveraging the specialization of the two models in $\mathbf{T}$.

The rest of the paper is organized as follows. Section 2 formally defines the problem. Section 3 details our proposal, and Section 4 presents experimental results. Section 5 discusses related work, and Section 6 concludes the paper.

II. Preliminaries and Problem Definition

Source Domain and Target Domain. A customer would like to have a classification model $\mathcal{M}_\mathbf{t}$ for a *target domain* $\mathbf{T} = \{X, Y\}$, where X is the feature space and Y is the label space. The customer owns a small and labeled *validation set* $\mathcal{D}_V$ from the target domain, $\mathcal{D}_V \subset \mathbf{T}$, while labels of any other data from $\mathbf{T}$ are revealed with delays. To accomplish the classification task in $\mathbf{T}$, the customer selects a suitable closed-box model $\mathcal{M}_\mathbf{s}$ from the model market according to the advertised model performance. We call the data domain that $\mathcal{M}_\mathbf{s}$ is trained on the *source domain* $\mathbf{S}$.

Data Pool. The data pool, denoted as $\mathcal{D} \subset \mathbf{T}$, is a labeled dataset provided by a seller from the data market that allows the customer to purchase samples to enhance the model performance. Every purchase of samples from $\mathcal{D}$ comes at a price. We assume a scenario where the customer is restricted by budget B to acquire no more than B samples from the data pool. Data acquisition can be executed in several rounds. Each round involves the purchase of one or multiple samples from $\mathcal{D}$.

Interaction with the Data Market. A round of interaction between the customer and the seller in the data market consists of two steps: (1) In the i-th round, the customer sends to the seller a data valuation function $F_i(\mathbf{x})$ where $\mathbf{x}$ is an input data instance from $\mathcal{D}$, and a data acquisition budget for this round, b_i. Furthermore, the customer aims to preserve his model and target data privacy as much as possible. (2) The seller computes $F_i(\mathbf{x})$ on relevant data (i.e., data instances within the label space Y) and then returns to the customer those instances with the highest $F_i(\mathbf{x})$ values, $\mathcal{D}'$, such that $|\mathcal{D}'| = b_i$ and $\sum_i b_i \leq B$.

Definition 1. *Given a data acquisition budget B, a closed-box model $\mathcal{M}_\mathbf{s}$, a data pool $\mathcal{D} \subset \mathbf{T}$, where $\mathbf{T}$ is the target domain, and a validation set $\mathcal{D}_V \subset \mathbf{T}$, the objective of **data acquisition for domain adaptation** of $\mathcal{M}_\mathbf{s}$ is to identify the B samples to acquire from $\mathcal{D}$ (denoted by $\mathcal{D}_B$) to optimize the accuracy of a target model $\mathcal{M}_\mathbf{t}$ in $\mathbf{T}$.*

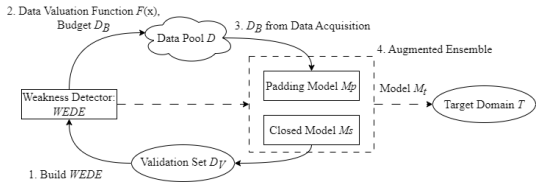

Fig. 2: Data Acquisition with Model Ensemble for Improving a Closed-Box Model in the Target Domain

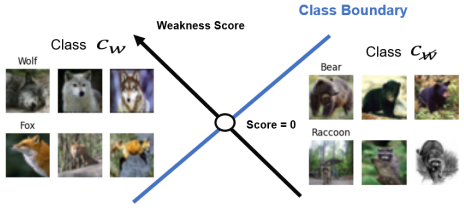

Fig. 3: Support Vector Machine as WEDE

III. Methodology

A. Overview

Our proposed solution to data acquisition for domain adaptation of closed-box models is illustrated in Figure 2. To construct a model $\mathcal{M}_\mathbf{t}$ for the target domain $\mathbf{T}$ with the source model $\mathcal{M}_\mathbf{s}$ and data acquisition budget B, the customer first trains a binary classifier WEDE by correct and incorrect predictions of $\mathcal{M}_\mathbf{s}$ in the validation set $\mathcal{D}_V$ from $\mathbf{T}$ to learn about the feature of $\mathcal{M}_\mathbf{s}$'s weaknesses. Then, the customer acquires a set of data instances $\mathcal{D}_B$ from a data pool $\mathcal{D}$ by providing the seller with budget B and a data valuation function $F(\mathbf{x})$ influenced by WEDE. The acquired data $\mathcal{D}_B$ is used to train a padding model $\mathcal{M}_\mathbf{p}$ to deal with weaknesses of $\mathcal{M}_\mathbf{s}$ in $\mathbf{T}$. $\mathcal{M}_\mathbf{p}$ can have any kind of classification architecture. Finally, the output of $\mathcal{M}_\mathbf{t}$ in $\mathbf{T}$ is constructed with Augmented Ensemble (AE) by combining predictions from $\mathcal{M}_\mathbf{s}$ and $\mathcal{M}_\mathbf{p}$ according to target data characteristics identified by WEDE.

B. Data Acquisition with WEDE

WEDE for Data Utility Estimation. To extract the features of $\mathcal{M}_\mathbf{s}$'s weakness in $\mathbf{T}$ and derive the associated *weakness score* $g(\mathbf{x})$ of an instance $\mathbf{x}$ from $\mathcal{D}$, we build a weakness detector WEDE to discriminate correct and incorrect predictions of $\mathcal{M}_\mathbf{s}$ in $\mathcal{D}_V$ as suggested by the research on model vulnerability interpretation [9]. We define incorrect predictions as class c_w while correct predictions as class $c_{\not w}$. For this binary classification task, we use Support Vector Machine (SVM) to implement WEDE; however, other classification models can also be used. SVM separates data of class c_w and class $c_{\not w}$ in a hyperspace to maximize the margin between data instances from different classes but closest to the class boundary. The distance of a data instance from the boundary to the side of class c_w is used as the weakness score. For example, in Figure 3, the solid arrow indicates the direction in which weakness scores increase. Data instances on the boundary have a score of 0.

Acquisition Strategy. The customer acquires data in a one-shot or a sequential manner. The *One-shot Strategy* returns $\mathcal{D}_B$ from $\mathcal{D}$ by using up budget B at once. On the contrary, the *Sequential Strategy* operates by multiple rounds. The i-th acquisition round consumes budget b_i and $\sum_i b_i \leq B$. After each round, WEDE is retrained on all the data instances that have been acquired so far, as well as the validation set $\mathcal{D}_V$. That is, the training data of WEDE after the i-th acquisition round is $\mathcal{D}_V \cup \{\mathcal{D}_{b_j}\}, j \in [0, i]$. Regardless of the acquisition strategy, after B is exhausted, all the acquired data instances $\mathcal{D}_B$ are used to train $\mathcal{M}_\mathbf{p}$.

C. Augmented Ensemble

We build a model ensemble $\mathcal{M}_\mathbf{t}$ with $\mathcal{M}_\mathbf{s}$ and $\mathcal{M}_\mathbf{p}$ for better accuracy in the target domain $\mathbf{T}$ where labels are released with delays. Compared to only applying $\mathcal{M}_\mathbf{s}$ to $\mathbf{T}$, $\mathcal{M}_\mathbf{t}$ achieves a better performance in $\mathbf{T}$ by including $\mathcal{M}_\mathbf{p}$ that is specialized in addressing the prediction weakness of $\mathcal{M}_\mathbf{s}$ in $\mathbf{T}$.

For this purpose, we propose a method called Augmented Ensemble (AE) to aggregate predictions from $\mathcal{M}_\mathbf{s}$ and $\mathcal{M}_\mathbf{p}$ and produce the output of $\mathcal{M}_\mathbf{t}$ under two ubiquitous application scenarios: (1) $\mathcal{M}_\mathbf{s}$ outputs only *hard labels*, which are predicted class labels for inputs, and (2) in addition to hard labels, $\mathcal{M}_\mathbf{s}$ also returns *soft labels*, which are prediction probabilities for every class. $\mathcal{M}_\mathbf{p}$ always gives customers both hard and soft labels, because it is built by customers rather than purchased as a closed box from the ML market.

When $\mathcal{M}_\mathbf{s}$ only returns hard labels, the model ensemble $\mathcal{M}_\mathbf{t}$ outputs a predicted class label $h_t(\mathbf{z})$ of a target data instance $\mathbf{z}$ solely based on the prediction result $h_p(\mathbf{z})$ or $h_s(\mathbf{z})$ from $\mathcal{M}_\mathbf{p}$ and $\mathcal{M}_\mathbf{s}$ respectively. Since $\mathcal{M}_\mathbf{p}$ is designed for the weakness of $\mathcal{M}_\mathbf{s}$ in the target domain, it is natural to select $h_p(\mathbf{z})$ as $h_t(\mathbf{z})$ if $\mathbf{z}$ is likely to be incorrectly predicted by $\mathcal{M}_\mathbf{s}$. To identify such target data for $\mathcal{M}_\mathbf{p}$, we design *AE-C* strategy that uses a criterion based on the weakness probability, which is calculated by Equation 1.

$$f(\mathbf{x}) = P(\theta_\mathbf{w}|g(\mathbf{x})) = \frac{P(g(\mathbf{x})|\theta_\mathbf{w})P(\theta_\mathbf{w})}{P(g(\mathbf{x})|\theta_\mathbf{w})P(\theta_\mathbf{w}) + P(g(\mathbf{x})|\theta_{\not\mathbf{w}})P(\theta_{\not\mathbf{w}})} \quad (1)$$

$f(\mathbf{x})$ estimates the probability that a data instance $\mathbf{x}$, represented by its weakness score $g(\mathbf{x})$, is drawn from the weakness distribution $\theta_\mathbf{w}$ of $\mathcal{M}_\mathbf{s}$. Figure 4 illustrate $\theta_\mathbf{w}$ and $\theta_{\not\mathbf{w}}$ (the non-weakness distribution) from SVM-based WEDE.

If the probability $f(\mathbf{z})$ of $\mathbf{z}$ is not less than a given criterion c, the prediction of $\mathcal{M}_\mathbf{p}$ on $\mathbf{z}$, $h_p(\mathbf{z})$, is considered as the final decision $h_t(\mathbf{z})$, i.e.,

$$h_t(\mathbf{z}) = \begin{cases} h_p(\mathbf{z}) & \text{if } f(\mathbf{z}) \geq c \\ h_s(\mathbf{z}) & \text{if } f(\mathbf{z}) \text{ ¡ } c \end{cases} \quad (2)$$

Criterion c is set manually. For example, customers can perform a grid search of c in a validation set to select c with the best result for AE-C.

When $\mathcal{M}_\mathbf{s}$ also provides prediction probabilities of every class as soft labels for $\mathbf{z}$, we propose *AE-W* method to aggregate soft labels from both $\mathcal{M}_\mathbf{s}$ and $\mathcal{M}_\mathbf{p}$ to produce $h_t(\mathbf{z})$ as the predicted label from the ensemble $\mathcal{M}_\mathbf{t}$. We denote the soft label of class k from $\mathcal{M}_\mathbf{s}$ and $\mathcal{M}_\mathbf{p}$ on a given $\mathbf{z}$ as $p_{\mathrm{s},k}(\mathbf{z})$ and $p_{\mathrm{p},k}(\mathbf{z})$, respectively. In the aggregation of $p_{\mathrm{s},k}(\mathbf{z})$ and $p_{\mathrm{p},k}(\mathbf{z})$, $p_{\mathrm{p},k}(\mathbf{z})$ should gain more importance than $p_{\mathrm{s},k}(\mathbf{z})$ when $\mathcal{M}_\mathbf{s}$ is not expected to predict $\mathbf{z}$ correctly. As such, we consider the weight of $p_{\mathrm{p},k}(\mathbf{z})$ in the soft-label aggregation as the probability that $\mathbf{z}$ is from the distribution of $\mathcal{M}_\mathbf{s}$'s weaknesses. Then, we use the weakness probability estimation $f(\mathbf{z})$ from Equation 1 to approximate the weight of $p_{\mathrm{p},k}(\mathbf{z})$ and $1 - f(\mathbf{z})$ for the weight of $p_{\mathrm{s},k}(\mathbf{z})$ to obtain the prediction probability of $\mathbf{z}$ belonging to class j generated by $\mathcal{M}_\mathbf{t}$, i.e.,

$$p_{\mathrm{t},k}(\mathbf{z}) = (1 - f(\mathbf{z}))p_{\mathrm{s},k}(\mathbf{z}) + f(\mathbf{z})p_{\mathrm{p},k}(\mathbf{z}). \quad (3)$$

The final classification decision from $\mathcal{M}_\mathbf{t}$, $h_t(\mathbf{z})$, is the class k with the highest probability generated by Equation 3, i.e., $h_t(\mathbf{z}) = \mathrm{argmax}_k p_{\mathrm{t},k}(\mathbf{z})$.

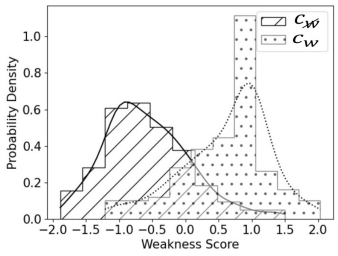

Fig. 4: Weakness Score Distribution

IV. EXPERIMENTS

A. Experiment Setup

Datasets. Our experiments utilize the *Core-50* and *Cifar-100* datasets. *Core-50* consists of images in ten categories, each divided into five object classes and 11 session classes. *Cifar-100* includes images from 20 superclasses, each containing five subclasses. We configured multi-class classification tasks: *Core-Object*, *Core-Session*, *Cifar-20-Class*, and *Cifar-4-Class*.

For each task, we divide the data set into training, data acquisition, validation, and test sets. To simulate data distribution shifts for the target domain, we selectively removed data from the training set while keeping the other sets unchanged. Specifically, for *Core-Object*, one object class from each category was removed; for *Core-Session*, one session class per category was omitted; for *Cifar-4-Class*, two subclasses per superclass were excluded; and for *Cifar-20-Class*, one subclass per superclass was removed. This creates a broader distribution in the target domain compared to the source domain.

Model and WEDE Implementation. For both $\mathcal{M}_\mathbf{s}$ and $\mathcal{M}_\mathbf{p}$, we adopt Squeezenet [8] for *Core-Session* and *Core-Object*. To reduce model training time in *Cifar-4-Class* and *Cifar-20-Class*, we add a linear classification layer after feature extraction by a pretrained Resnet20 [4]. We train all models with 20 epochs and return the training checkpoint with the smallest loss in the validation set $\mathcal{D}_V$. SGD with 0.9 momentum and 5 x 10^{-4} weight decay is used to optimize the model parameters. The batch size of $\mathcal{M}_\mathbf{p}$'s training data is 256 for *Cifar-20-Class* and 64 for other tasks. We adjust the learning rate for model

parameter optimization batch-by-batch. It increases linearly from 0 to 0.1 by the end of the 10th epoch and then decreases linearly to 0 at the end of the training.

For WEDE, we utilize the SVM-based approach with the *SVC* optimization package for *Core-Session*, *Core-Object*, and *Cifar-4-Class*, and *LinearSVC* for the larger *Cifar-20-Class*. A grid search optimizes WEDE's hyperparameters.

Evaluation Metrics. We use ***Accuracy Improvement*** to measure acquisition performance. It is defined as the classification accuracy of $\mathcal{M}_\mathbf{t}$ on the test set minus the accuracy of $\mathcal{M}_\mathbf{s}$ on the same dataset, where accuracy refers to the number of correct predictions divided by the total number of predictions.

Environment. Our experiment uses an NVIDIA RTX A5000 GPU and Pytorch 1.12.1 for model training and testing, and scikit-learn 1.3.0 for building SVM-based WEDE. The vision transformer *ViT-B/32* from OpenAI [17] encodes images into vectors as inputs for WEDE. We adopt Scipy 1.10.1 for kernel density estimation in weakness probability function $f(\mathbf{x})$. All experiments are repeated 10 times, and the average results are reported.

B. Baselines

We implement four acquisition methods as baselines.

1. Random Acquisition. We randomly sample instances from the data pool $\mathcal{D}$ within budget B and use these samples to build $\mathcal{M}_\mathbf{p}$. This is equivalent to setting $F(\mathbf{x})$ as a random number generator in data acquisition. The training data of $\mathcal{M}_\mathbf{p}$ is expected to be from the same distribution as the test set, and thus we only apply $\mathcal{M}_\mathbf{p}$ to the test set.

2. Probability-at-Ground-Truth (PGT) Acquisition. We select labeled instances from $\mathcal{D}$ with the lowest probabilities on their ground-truth labels from $\mathcal{M}_\mathbf{s}$. $F(\mathbf{x})$ for a labeled instance $\mathbf{x}$ is set as the negative value of $\mathcal{M}_\mathbf{s}$'s prediction probability on the true label of $\mathbf{x}$. This baseline returns the most extreme weaknesses of $\mathcal{M}_\mathbf{s}$.

3. Random Weakness Acquisition. Instead of sampling from the entire data pool, this baseline randomly draws $\mathcal{D}_B$ instances from incorrect predictions of $\mathcal{M}_\mathbf{s}$ in $\mathcal{D}$. $F(\mathbf{x})$ in this baseline has two components: $F_1(\mathbf{x})$ to collect incorrect predictions from $\mathcal{M}_\mathbf{s}$ in $\mathcal{D}$ and a random number generator $F_2(\mathbf{x})$ to sample the results from $F_1(\mathbf{x})$.

4. Entropy-based Acquisition. We consider the high prediction entropy of $\mathcal{M}_\mathbf{s}$ as an alternative indication of $\mathcal{M}_\mathbf{s}$'s weakness. $F(\mathbf{x})$ equals the prediction entropy computed by $-\sum_k p_{s,k}(\mathbf{x})\log_2(p_{s,k}(\mathbf{x}))$, where $p_{s,k}(\mathbf{x})$ is $\mathcal{M}_\mathbf{s}$'s prediction probability of class k given an input $\mathbf{x}$. $\mathcal{M}_\mathbf{p}$ from this baseline is tested with AE.

After conducting *PGT*, *Random Weakness*, and *Entropy-based* data acquisition, we also apply AE to use $\mathcal{M}_\mathbf{p}$ in the target domain.

C. Results

We compare WEDE-based acquisition and baselines in building an accurate $\mathcal{M}_\mathbf{t}$ for the test set. Note that PGT, random weakness, and entropy-based baselines are all hypothetical, as customers are required to provide data sellers with their source model $\mathcal{M}_\mathbf{s}$ for data valuation by these baselines, which most prefer to keep $\mathcal{M}_\mathbf{s}$ private.

As shown in Figure 5 and Figure 6, our proposed acquisition with WEDE and AE achieves the highest accuracy gains.

Random acquisition returns $\mathcal{D}_B$ exactly from the target domain $\mathbf{T}$, but does not apply any of $\mathcal{M}_\mathbf{s}$ to the target domain, which means that $\mathcal{M}_\mathbf{p}$ has to prepare for $\mathbf{T}$ from the beginning by learning about $\mathcal{D}_B$. $\mathcal{D}_B$ acquired by a small budget B even fails to train a $\mathcal{M}_\mathbf{p}$ with competitive performance as $\mathcal{M}_\mathbf{s}$ in $\mathbf{T}$.

Instead, biased acquisition methods associated with the model ensemble approach AE exploit the capability of $\mathcal{M}_\mathbf{s}$ in $\mathbf{T}$ in addition to $\mathcal{M}_\mathbf{p}$. As such, they achieve better results than random acquisition under small budgets. However, they also exhaust $\mathcal{M}_\mathbf{s}$'s weaknesses in $\mathcal{D}$ faster than random acquisition because they have acquisition bias in $\mathcal{M}_\mathbf{s}$'s weaknesses. As a result, we observe that as B increases, the accuracy improvement from biased strategies slows down while the accuracy improvement from random acquisition increases steadily.

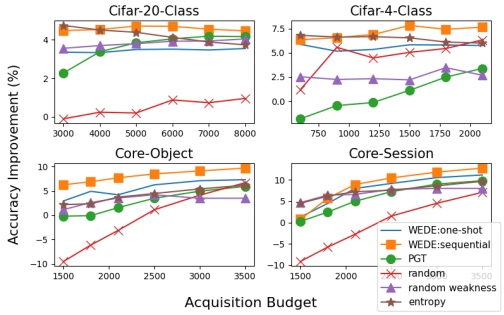

Fig. 5: WEDE-based Acquisition v.s. Baselines in Soft-Label-Available Scenario

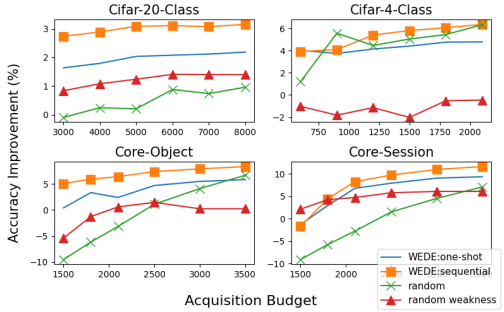

Fig. 6: WEDE-based Acquisition vs. Baselines in Hard-Label-Only Scenario

We then compare AE-W and AE-C in combining $\mathcal{M}_\mathbf{s}$ and $\mathcal{M}_\mathbf{p}$. First, we investigate the distribution of weakness probability $f(\mathbf{x})$ as the weight of $\mathcal{M}_\mathbf{p}$'s predictions in AE-W and a tool for test data assignment in AE-C. Figure 7 exhibits an overlap of the $f(\mathbf{x})$ distribution between $\mathcal{M}_\mathbf{s}$'s weaknesses c_w and non-weaknesses $c_{\not w}$. Despite this overlap, AE-W allows $\mathcal{M}_\mathbf{s}$ and $\mathcal{M}_\mathbf{p}$ to work together on every data instance. Even if $f(\mathbf{x})$ is more than 0.5 in the prediction aggregation for $\mathbf{x}$ of $c_{\not w}$

and $\mathcal{M}_\mathbf{p}$ produces incorrect predictions, $\mathcal{M}_\mathbf{s}$ can still reverse the aggregation result by its specialization in $c_{\not w}$. In contrast, AE-C adopts only one model's prediction as the ensemble output $h_t(\mathbf{x})$. When AE-C gives $\mathbf{x}$ of $c_{\not w}$ to $\mathcal{M}_\mathbf{p}$, $h_t(\mathbf{x})$ is often incorrect. As a result, we observe in Figure 5 and Figure 6 that AE-W ensembles are superior to AE-C ensembles, whichever acquisition strategy is adopted.

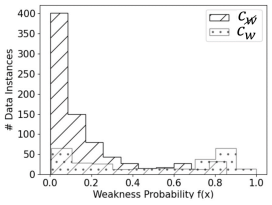

Fig. 7: Distribution of Weakness Probability

We also analyze the difference between one-shot and sequential acquisition. We first examine each round of the sequential acquisition. Each acquisition round in the sequential strategy is set to have the same budget. As the acquisition goes round by round, more instances of class c_w ($\mathcal{M}_\mathbf{s}$'s weaknesses in the target domain) are added to the validation set $\mathcal{D}_V$ to retrain WEDE as shown in Figure 8. Since WEDE has more data on c_w to learn, we see an improvement in the accuracy of weakness detection as shown in Figure 9.

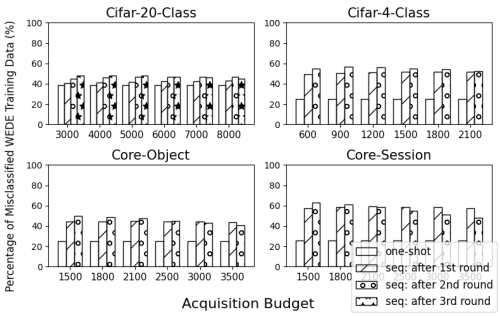

Fig. 8: Mis-classification by $\mathcal{M}_\mathbf{s}$ in WEDE's Training Data

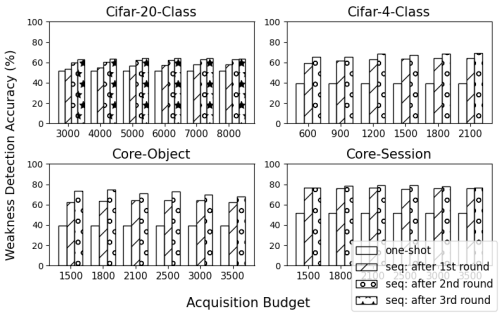

Fig. 9: Weakness Detection Accuracy in WEDE-Based Acquisition

After applying different WEDE to test data, we observe from Figure 10 and Figure 11 that $f(\mathbf{x})$ with WEDE updated by sequential acquisition estimates more test data misclassified by $\mathcal{M}_\mathbf{s}$ with a weakness probability over 0.5, while it gives fewer test data correctly classified by $\mathcal{M}_\mathbf{s}$ a weakness probability estimation larger than 0.5. Consequently, Figure 6 and Figure 5 show that sequential acquisition performs better than one-shot in improving the accuracy of $\mathcal{M}_\mathbf{t}$ in all classification tasks.

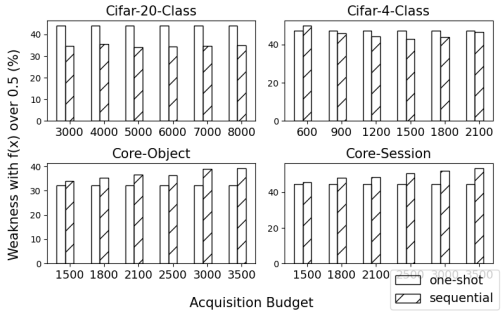

Fig. 10: $\mathcal{M}_\mathbf{s}$'s Incorrect Predictions with Weakness Probability over 0.5

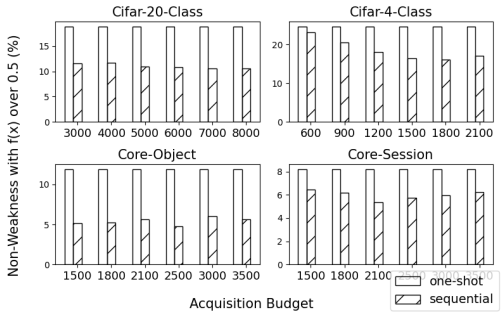

Fig. 11: $\mathcal{M}_\mathbf{s}$'s Correct Predictions with Weakness Probability over 0.5

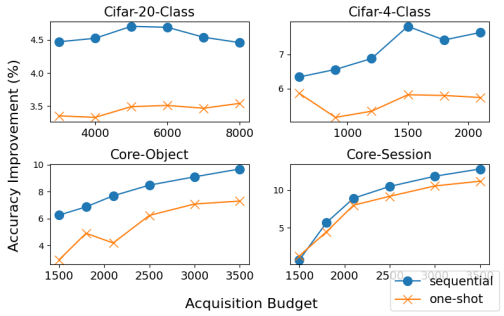

Fig. 12: One-Shot Acquisition vs. Sequential Acquisition in Soft-Label Scenario

Finally, we investigate how distribution shift degree affects the effectiveness of our proposed method. We generate various degrees of distribution shift by removing 1, 2, or 3 subclasses from each superclass in *Cifar-2-class*, and 2, 3, or 5 sessions from each category in *Core-50*. Datasets before and after the data removal are considered to be drawn from source and target domains. The more subclasses/sessions removed, the higher the discrepancy between the source and target domains.

Figure 13 shows the effect of degree of shifts in the distribution, where each curve represents the accuracy improvement obtained by WEDE-based acquisition over random acquisition with a different number of sessions/subclasses removed. As the degree of distribution shift becomes greater, e.g. from 2 sessions to 5 sessions removed from each category (with each category originally containing 11 sessions) or from 1 subclass to 3 subclasses removed from each superclass (with each superclass originally containing 5 subclasses), WEDE-based acquisition presents less advantage over random acquisition. This result agrees with the intuition that model $\mathcal{M}_\mathbf{s}$ is of little use to the prediction ensemble if the target domain is significantly different from the source domain.

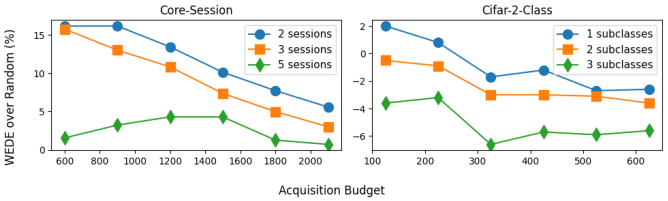

Fig. 13: Acquisition under Different Distribution Shifts

V. Related Work

The framework of the ML market, especially data acquisition, is detailed in existing literature [3], [13]. Techniques for selecting data instances vary: Some aim to acquire data based on novelty [10] or anticipated improvement to model confidence [11], while others cluster the data pool and then proportionally select from each cluster to avoid data acquisition bias [20]. Our approach differs by assessing an instance's utility based on its likelihood of misclassification by $\mathcal{M}_\mathbf{s}$, aiding in the enhancement of the padding model $\mathcal{M}_\mathbf{p}$.

Distribution shift, a well-studied topic [16], includes methods like reconstruction errors, domain classification confidence, and density estimation using out-of-distribution detection [15], [18]. However, these methods generally assume access to source data, which is not available in closed-box models, limiting their applicability.

Existing research on domain adaptation for closed-box models often involves building a new model via knowledge transfer. Liang et al. [12] use knowledge distillation, filtering pseudo-labels from a "teacher" model to train a new model for the target domain. Xia et al. [21] employ model predictions to identify the source and the target domain's overlap and differences, and then capture their correlation by contrastive learning. However, these strategies typically require extensive data to learn to transfer knowledge from the closed-box model to a new model for the target domain. With a focus on data acquisition for domain adaption, He et al. [7] collect data from predefined domains via active learning to mimic the target domain, but this approach involves retraining the original model in an iterative fashion, which violates the nature of closed-box model.

VI. Conclusions

In this paper, we present a secure domain adaptation framework for closed-box models that uses additional data purchased from the machine-learning market to address performance drops due to distribution shifts, while ensuring customer privacy. Our approach begins with a classifier, WEDE, which identifies prediction errors in the target domain. This information allows us to efficiently select and acquire new data to train a supplemental model, improving the closed-box model's accuracy. We introduce an Augmented Ensemble (AE) technique that merges the new and original models for enhanced performance. Our results demonstrate superior effectiveness over baselines, especially under tight budget constraints.

References

[1] Amazon. Amazon rekognition, 2023.
[2] Amazon. Aws data exchange, 2023.
[3] A. Asudeh and F. Nargesian. Towards distribution-aware query answering in data markets. *PVLDB*, 15(11):3137–3144, July 2022.
[4] chenyaofo. Pytorch cifar models. https://github.com/chenyaofo/pytorch-cifar-models/, 2023.
[5] Databricks. Databricks marketplace, 2023.
[6] Google. Google cloud vision api, 2023.
[7] Y. He, D. Li, P. Tian, H. Yu, J. Liu, H. Zou, and P. Cui. Domain-wise data acquisition to improve performance under distribution shift. In *ICML*, 2024.
[8] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size, Nov. 2016. arXiv:1602.07360 [cs].
[9] S. Jain, H. Lawrence, A. Moitra, and A. Madry. Distilling model failures as directions in latent space. In *ICLR*, 2023.
[10] Y. Li, X. Yu, and N. Koudas. Data Acquisition for Improving Machine Learning Models. *PVLDB*, 14(10):1832–1844, June 2021.
[11] Y. Li, X. Yu, and N. Koudas. Data acquisition for improving model confidence. *Proc. ACM Manag. Data*, 2(3):131, 2024.
[12] J. Liang, D. Hu, J. Feng, and R. He. DINE: Domain Adaptation from Single and Multiple Black-box Predictors. In *CVPR*, pages 7993–8003. IEEE, June 2022.
[13] J. Liu, J. Lou, J. Liu, L. Xiong, J. Pei, and J. Sun. Dealer: an end-to-end model marketplace with differential privacy. *PVLDB*, 14(6):957–969, 2021.
[14] OpenVINO. Open model zoo, 2023.
[15] S. Pidhorskyi, R. Almohsen, D. A. Adjeroh, and G. Doretto. Generative probabilistic novelty detection with adversarial autoencoders. In *NeurIPS*, page 6823–6834, 2018.
[16] S. Rabanser, S. Günnemann, and Z. C. Lipton. Failing loudly: An empirical study of methods for detecting dataset shift. In *NeurIPS*, 2019.
[17] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever. Learning Transferable Visual Models From Natural Language Supervision, Feb. 2021. arXiv:2103.00020 [cs].
[18] L. Ruff, R. Vandermeulen, N. Goernitz, L. Deecke, S. A. Siddiqui, A. Binder, E. Müller, and M. Kloft. Deep one-class classification. In *ICML*, pages 4393–4402, 2018.
[19] Y. Shi, K. Wu, Y. Han, Y. Shao, B. Li, and F. Wu. Source-free and black-box domain adaptation via distributionally adversarial training. *Pattern Recognition*, 143:109750, Nov. 2023.
[20] T. Wang, S. Huang, Z. Bao, J. S. Culpepper, V. Dedeoglu, and R. Arablouei. Optimizing data acquisition to enhance machine learning performance. *PVLDB*, 17(6):1310–1323, May 2024.
[21] M. Xia, J. Zhao, G. Lyu, Z. Huang, T. Hu, G. Chen, and H. Wang. A separation and alignment framework for black-box domain adaptation. *AAAI*, 38(14):16005–16013, Mar. 2024.