Active Learning for Multiple Target Models

Sheng-Jun Huang, Yi Li and Ying-Peng Tang

Abstract—We present a novel setting of active learning (AL) where multiple target models are simultaneously learned. This setting arises in real-world applications where machine learning systems require training multiple models on the same labeled dataset to accommodate diverse devices with varying computational resources. However, traditional AL methods are often limited by their model dependence and non-transferability. In this paper, we address the question of whether an effective AL method can be designed for multiple target models. We analyze the query complexity of active and passive learning in this setting and demonstrate the potential for AL to achieve improved query complexity. Based on this insight, we further propose an agnostic AL sampling strategy which selects examples located in the joint disagreement regions of different target models. Experimental evaluations on classification and regression benchmarks validate the effectiveness of our approach over traditional AL methods.

Index Terms—Machine learning, active learning, query complexity.

1 Introduction

Data labeling is usually expensive due to the involvement of human annotators. Active learning (AL) is a main approach to reduce the labeling cost [54]. It assumes that different data have varying impacts on the model performance, and thus, efficient model training can be achieved by selectively labeling the informative examples. Active learning evaluates the utility of the unlabeled data based on the model to be learned, i.e., the target model, from various aspects and actively queries the ground-truth labels for the examples that would most benefit the model's performance improvement. Commonly used selection criteria include uncertainty [19], diversity [34], representativeness [33], among others. In the past decades, many works have validated the great potential of active learning in reducing training data while achieving the same performance across various tasks [31], [62], [69].

Existing active learning methods typically aim to fit a single specific target model with the fewest queries, such as SVM [31], hidden Markov model [52], neural networks with specific architectures [56]. However, in many real-world applications, machine learning systems are required to be deployed on multiple types of devices with different resource constraints [9]. For example, speech recognition software needs to support a wide range of machines, from high-performance workstations to mobile phones. Due to differences in computational resources, the applicable model architectures can vary considerably. A deep model which performs well on the cloud server may not be suitable for deployment on edge devices. As a result, it becomes necessary to train multiple models with different complexities on the same labeled dataset to accommodate these diverse devices.

Given multiple target models, it has become a practical and challenging problem to improve them effectively with the fewest labeled data. Conceivably, different models will have different preferences on the training data, which has been verified by many works showing that AL is usually model-dependent and nontransferable [46], [49], [70], i.e., the best query strategy for different target models can vary significantly [71]. In other words, the data queried by one model may be less effective when used to train another model [46]. These observations indicate that the existing active query strategies can hardly benefit all target models simultaneously, highlighting the necessity and challenge of designing AL algorithms for multi-model scenarios. This raises a natural question: "*Does there exist an active learning method which can query a set of labeled data in such a way that all the target models can be effectively trained using those data?*"

In this paper, we formally define the problem of active learning for multiple target models, where multiple heterogeneous models are learned on the same labeled dataset. Our goal is to actively query the informative unlabeled data that carry crucial information about the learning task in order to improve the performances of all target models simultaneously with the least possible queries. To verify the rationality and solvability of the problem, and demonstrate the potential improvement of AL under this novel setting. Based on this insight, we further propose an agnostic disagreement-based selection criterion for both classification and regression tasks. we first define and analyze the query complexity for both active and passive learning under the setting of multiple target models. This query complexity characterizes the number of labeled examples sufficient to train an ε-good classifier with probability at least $1-\delta$ for every target model. We establish that the query complexity of multiple models can be upper bounded by that of an appropriately designed single model in the realizable case (i.e., the target concept which generates the ground truth is contained in the hypothesis space), indicating the potential improvement of AL under this setting. To further explore

Sheng-Jun Huang and Ying-Peng Tang are with the MIIT Key Laboratory of Pattern Analysis and Machine Intelligence, Collaborative Innovation Center of Novel Software Technology and Industrialization, College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China (e-mail: huangsj@nuaa.edu.cn, tangyp@nuaa.edu.cn).

Yi Li is with the School of Physical and Mathematical Sciences and the College of Computing and Data Science, Nanyang Technological University, Singapore 637371 (e-mail: yili@ntu.edu.sg).

Ying-Peng Tang is the corresponding author.

This is an extended and revised version of a preliminary conference paper that was presented in NeurIPS 2022 [61].

the agnostic case, we propose an active selection method called DIAM (i.e., DIsagreement-based AL for Multi-models) to select the best examples beneficial to all target models. It prefers the data located in the joint disagreement regions of different models as they are expected to be more effective in reducing the soft version space (i.e., the set of hypotheses with lower errors). We provide a rigorous theoretical analysis of the DIAM method and propose efficient implementations for both deep classification and regression tasks. For classification tasks, our implementation exploits the models in the later training epochs to construct joint disagreement regions and further considers the diversity criterion in data querying to enable batch mode selection. For regression tasks, we consider two problem definitions and demonstrate that identifying the data in disagreement regions can be efficiently solved using linear algebraic techniques. To validate the importance of designing active query methods for multiple target models and to evaluate the effectiveness of our proposed approaches, we conduct experiments on the benchmarks representative of tasks which are typically required to support multiple types of devices. The first task is Optical Character Recognition (OCR) used for deep classification and the second is Facial Landmark Detection (FLD) used for regression. Our results show that the DIAM method significantly outperforms traditional active and passive learning methods for multiple models in terms of reducing the number of queries required while achieving higher mean accuracy.

We summarize the contributions of this work as follows.

1) We formally define the novel setting of active learning for multiple target models, which aims to reduce the labeling cost for the application scenarios that need to support a wide range of machines.
2) We establish that the query complexity of multiple models can be upper bounded by that of an appropriately designed single model under the realizable case, demonstrating the potential improvement of AL in this setting.
3) We propose an agnostic active learning algorithm for multiple target models, and provide theoretical analysis on its superiority in terms of the query complexity compared with the baseline methods.
4) We extend the proposed DIAM method for deep classification. Our implementation exploits the training process of neural networks to find disagreement regions and introduces the diversity criterion to enable batch mode selection.
5) We further extend the proposed DIAM method for deep regression. We demonstrate that identifying the data in disagreement regions in mean-squared loss can be efficiently solved using linear algebraic techniques.
6) Extensive experiments are conducted on the benchmarks of OCR and FLD tasks. The results show that the DIAM method can significantly outperform the other baseline methods in terms of reducing the number of queries required while achieving a higher mean accuracy.

Note that, a preliminary version of this work has been published in [61]. We summarize the updated contents as follows. i) We have improved the original DIAM query strategy by introducing a diversity criterion to boost its performance in batch mode querying. ii) We have extended the proposed DIAM method for deep regression tasks with efficient implementations using linear algebraic techniques. iii) We have employed more classification and regression benchmarks to validate the effectiveness of our method. iv) We significantly improved the presentation of the paper.

The remainder of the paper is organized as follows. Section 2 provides a review of related work. In Sec. 3, we formally define the AL for the multiple target model problem and present a general result which bridges the query complexity between single and multiple models. Section 4 explores the potential improvement of AL under this novel setting. Next, in Section 5, we propose and analyze an agnostic active selection criterion. Section 6 presents the empirical studies conducted to evaluate the proposed method. Finally, we conclude our work in Section 7.

2 Related Work

2.1 Learning Multiple Target Models

As the complexity of deep models increases, many machine learning systems need to learn light models to ensure user experience on diverse devices with varying computational constraints [47]. To achieve this goal, most studies aim at reducing the size of a big model with high accuracy, while minimizing the performance loss. This can usually be implemented by knowledge distillation (KD) [22] and model compression [12]. The former focuses on distilling knowledge from a larger teacher model to a smaller student model. Based on the types of knowledge being distilled, there are primarily three categories of approaches, Response-Based KD [4], [30], which regularizes the logits or soft predictions of the teacher and student models, Feature-Based KD [51], which transfers the intermediate representations, and Relation-Based KD [68], which mines the relations between different layers or data. The latter aims at pruning the less important nodes or units from the model [24], [25], [42], or quantizing the parameters and activations to low-precision data types [35], [38] to reduce the model size and accelerate the inference speed. The pruning-based methods calculate the saliency score of different parameters to identify the non-informative nodes, leading to less performance loss after eliminating them. Quantisation-based methods usually rescale, clamp, or transform the weights into fixed-point, rather than floating-point, numbers to reduce memory occupation and improve efficiency.

Recently, Neural Architecture Search (NAS) [18] is extended to search hardware-aware models [9], [10], [29] to support devices with different computational resources. For example, He *et al.* [29] incorporate model compression into the model search phase and employ reinforcement learning to optimize both compression policy and model architectures. Cai *et al.* [9] propose an efficient NAS method to search different architectures for various devices with only training the super-net once, so that small models can be efficiently evaluated by pruning the super-net with weight inheriting. All of the aforementioned methods address the challenge of supporting devices with limited computational

resources from the model perspective. In this work, we aim to tackle this challenge from a data perspective.

2.2 Active Learning

Active learning has been widely applied to address the increasing demand for labeled data in training deep models [50]. One of the tasks of AL is evaluating the potential contribution of each candidate query to the performance improvement of the target model. Most of the existing selection criteria can be categorized into informativeness and representativeness. The informativeness-based methods [21], [36], [66] select the data which is close to the decision boundary, while the representativeness-based methods [44], [53], [57] impose the constraints to regularize the queried data to be dissimilar with each other or conform to the latent data distribution. Many works also try to combine both criteria to achieve better performances [17], [59], [67]. Beyond these hand-crafted selection criteria, several meta-active-learning query strategies [37], [48], [64] are proposed to learn a generalizable query strategy across tasks. Most of the existing AL query strategies target on improving one specific target model.

From the theoretical view, one of the interested properties of an AL algorithm is the query complexity [26], [28], which characterizes the number of queries needed to obtain an ε-good classifier with probability at least $1-\delta$. To bound this value, disagreement coefficient [7], [8] and shattering [11], [27] are two commonly used techniques. While most works deal with the single model setting, Balcan *et al.* [6] study the query complexity of the hypothesis space and its subclasses, which sheds light on this work. However, they mainly focus on how to construct subclasses to achieve a certain query complexity, while we aim to find an effective AL algorithm on the given hypothesis spaces.

Recently, some AL methods have addressed the scenario where the target model has not been given before querying. Instead, only a candidate set of models is available. In this setting, these methods face the challenge of identifying the most effective model from the candidate set for the current task and fitting it with the fewest queries. To this end, ALMS [1] maintains two sets of data: an unbiased labeled set for evaluating candidate models and an informative dataset for effective model learning. At each iteration, the method computes a utility score to determine whether to query based on expected error reduction or query randomly. Active-iNAS [20], designed for the deep learning setting, employs NAS to search for an optimal model architecture iteratively while querying examples to enhance the performance of the currently identified best network architecture. Recently, Tang and Huang [60] propose a unified framework to incorporate model selection and active data querying. They employ truncated importance sampling to overcome the data bias in model evaluation and select data based on the inconsistency among the candidate models for querying. While all these methods focus on identifying the most effective model configurations from a set of target models, our work tries to improve all target models simultaneously.

3 Query Complexity of Single Model and Multiple Models

3.1 Notations and Definitions

Throughout the paper, $\mathcal{X}$ and $\mathcal{Y}$ denote the feature space and the label space, respectively. A hypothesis is a mapping from $\mathcal{X}$ to $\mathcal{Y}$ and there is an unknown target hypothesis h^*, which generates the ground-truth label $y \in \mathcal{Y}$ for each $\boldsymbol{x} \in \mathcal{X}$. We assume that there is an unknown distribution $\mathcal{D}_X$ over $\mathcal{X}$, from which the data are sampled. The generalization error of a hypothesis h is then defined as $\mathrm{err}(h) = \mathbb{P}_{\boldsymbol{x}\sim\mathcal{D}_X}(h(\boldsymbol{x}) \neq h^*(\boldsymbol{x}))$.

Consider a dataset of n data points which are sampled randomly and independently from $\mathcal{D}_X$. In active learning, the dataset usually consists of a small labeled set $\mathcal{L} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n_l}$ of size n_l, which is used for model initialization, and a large unlabeled set $\mathcal{U} = \{\boldsymbol{x}_i\}_{i=n_l+1}^{n_l+n_u}$ of size n_u, which is used for data querying. Here, $n_l \ll n_u$ and $n = n_l + n_u$. The goal of an active learning algorithm is to produce a hypothesis h of small generalization error by querying the labels of data points in the unlabeled set $\mathcal{U}$ as few times as possible.

More specifically, in the single model setting, we are given a target model, i.e., a hypothesis space (e.g., SVM, decision tree, multi-layer perceptron, etc.), which implicitly define a set of hypotheses, namely, a hypothesis space, $\mathcal{C}$ before querying. A learning algorithm seeks to output a hypothesis $h \in \mathcal{C}$ such that its generalization error $\mathrm{err}(h)$ is close to the minimum error $\nu = \inf_{h\in\mathcal{C}} \mathrm{err}(h)$. When $h^* \in \mathcal{C}$, we say the learning task is *realizable*, which indicates that $\nu = 0$. Otherwise, it is called *agnostic* learning.

Meanwhile, the active learning algorithm aims at minimizing the number of label queries for points in $\mathcal{U}$. This is characterized and assessed through the notion of query complexity [28]. Below is the definition of the query complexity for the single target model.

Definition 1 (Query complexity for single target model, [28])**.** *Suppose that $\varepsilon, \delta \in (0,1)$ and $\mathcal{A}$ is an active learning algorithm. We say that $\mathcal{A}$ achieves query complexity $\Lambda(\mathcal{A}; \varepsilon, \delta, \mathcal{D}_X)$ on the hypothesis space $\mathcal{C}$ and distribution $\mathcal{D}_X$ if, for every query budget $t \geq \Lambda(\mathcal{A}; \mathcal{D}_X)$ and for every target hypothesis h^*, the algorithm $\mathcal{A}$, using at most t queries, returns a hypothesis $h_{t,\delta}$ such that*

$$\mathbb{P}\left(\mathrm{err}(h_{t,\delta}) \leq \nu + \varepsilon\right) > 1 - \delta, \tag{1}$$

where the probability is over the random samples drawn from $\mathcal{D}_X$.

Moreover, we say that $\mathcal{A}$ achieves (distribution-independent) query complexity

$$\Lambda\left(\mathcal{A}; \varepsilon, \delta\right) = \sup_{\mathcal{D}_X} \Lambda(\mathcal{A}; \varepsilon, \delta, \mathcal{D}_X)$$

on the hypothesis space $\mathcal{C}$.

When ε and δ are clear from the context, we may omit them and simply write $\Lambda(\mathcal{A}; \mathcal{D}_X)$ and $\Lambda(\mathcal{A})$.

In the multiple target models setting, there are k hypothesis spaces $\mathcal{C}_1, \ldots, \mathcal{C}_k$ and a learning algorithm seeks to output k hypotheses $h_1, \ldots, h_k$ such that $h_i \in \mathcal{C}_i$ and $\mathrm{err}(h_i)$ is close to the minimum error $\nu_i = \inf_{h\in\mathcal{C}_i} \mathrm{err}(h)$ in the i-th hypothesis space $\mathcal{C}_i$. Next is the formal definition of query complexity of active learning for multiple target models.

Definition 2 (Query complexity of multiple target models)**.** *Suppose that $\varepsilon, \delta \in (0,1)$ and $\mathcal{A}$ is an active learning algorithm. We say that $\mathcal{A}$ achieves query complexity $\tilde{\Lambda}(\mathcal{A}, \varepsilon, \delta, \mathcal{D}_X)$ on the hypothesis spaces $\mathcal{C}_1, \ldots, \mathcal{C}_k$ and distribution $\mathcal{D}_X$ if, for every query budget $t \geq \tilde{\Lambda}(\mathcal{A}, \mathcal{D}_X)$ and for every target hypothesis h^*, the algorithm $\mathcal{A}$, using at most t queries, returns hypotheses $h^i_{t,\delta} \in \mathcal{C}_i$ $(i = 1, \ldots, k)$ such that*

$$\mathbb{P}\left(\mathrm{err}(h^i_{t,\delta}) \leq \nu_i + \varepsilon\right) > 1 - \delta, \quad \forall i = 1, \ldots, k, \tag{2}$$

where the probability is over the random samples drawn from $\mathcal{D}_X$.

Moreover, we say that $\mathcal{A}$ achieves (distribution-independent) query complexity for multiple target models

$$\tilde{\Lambda}(\mathcal{A}; \varepsilon, \delta) = \sup_{\mathcal{D}_X} \tilde{\Lambda}(\mathcal{A}; \varepsilon, \delta, \mathcal{D}_X)$$

on the hypothesis spaces $\mathcal{C}_1, \ldots, \mathcal{C}_k$.

When ε and δ are clear from the context, we may omit them and simply write $\tilde{\Lambda}(\mathcal{A}; \mathcal{D}_X)$ and $\tilde{\Lambda}(\mathcal{A})$.

Before presenting our main results, we introduce a function to evaluate the difference between hypotheses, which is a pseudometric of hypotheses. This function plays a crucial role in the proof of the theorem.

Definition 3. *Given $\mathcal{D}_X$, the probability of disagreement between two classifiers h_1 and h_2 is defined as $d(h_1, h_2) = \mathbb{P}_{\boldsymbol{x} \sim \mathcal{D}_X}(h_1(\boldsymbol{x}) \neq h_2(\boldsymbol{x}))$.*

It is not difficult to verify that $d(\cdot, \cdot)$ as defined above is indeed a pseudometric; see, e.g., [26]. In particular, this means that the triangle inequality holds.

Finally, given a labeled set $\mathcal{L}$, we define the empirical error of a hypothesis h on $\mathcal{L}$ to be $\text{e-err}_{\mathcal{L}}(h) = \frac{1}{|\mathcal{L}|} \sum_{(x,y) \in \mathcal{L}} \mathbb{I}[h(\boldsymbol{x}) \neq y]$, where $\mathbb{I}[\cdot]$ is the indicator function. We also define $\mathrm{Log}(a) = \max\{\ln(a), 1\}$ for all $a > 0$.

3.2 Translating the Query Complexity of Single Model to Multiple Models

Denote by $\Lambda_i(\mathcal{P})$ and $\Lambda_i(\mathcal{A})$ the query complexity on $\mathcal{C}_i$ achieved by passive learning $\mathcal{P}$ and a specific active learning algorithm $\mathcal{A}$, respectively. When applying the existing algorithms to multiple target models setting, passive learning has a trivial query complexity for multiple models, namely, $\tilde{\Lambda}(\mathcal{P}') \leq \max_i \Lambda_i(\mathcal{P})$, where $\mathcal{P}'$ queries data randomly and outputs the empirical minimizer of each target model. This bound follows from the definition of query complexity of passive learning and is clearly tight without extra assumptions. The query complexity of active learning for multiple models, however, is much less understood. The trivial upper bound, assumed to be obtained by algorithm $\tilde{\mathcal{A}}$ which applies the AL algorithm $\mathcal{A}$ to each target model individually, is much worse: $\tilde{\Lambda}(\tilde{\mathcal{A}}) \leq \sum_i \Lambda_i(\mathcal{A})$. Our theorem below provides a possible direction to improve the upper bound of query complexity for AL. It shows that finding an ε-good classifier from each hypothesis space $\mathcal{C}_1, \ldots, \mathcal{C}_k$ is equivalent to finding an $(\varepsilon/2)$-good classifier in the combined hypothesis space $\tilde{\mathcal{C}} = \mathcal{C}_1 \cup \cdots \cup \mathcal{C}_k$ in the realizable case.

Theorem 1. *Suppose that $\varepsilon, \delta \in (0,1)$, $\mathcal{C}_1, \ldots, \mathcal{C}_k$ are k hypothesis spaces and $h^* \in \tilde{\mathcal{C}} = \bigcup_{i=1}^k \mathcal{C}_i$. If there exists an active learning algorithm $\mathcal{A}$ which achieves query complexity $\Lambda(\mathcal{A}, \varepsilon, \delta)$ on $\tilde{\mathcal{C}}$, Then, there exists an active learning algorithm $\mathcal{A}'$ which achieves the query complexity $\tilde{\Lambda}(\mathcal{A}', \varepsilon, \delta) = \Lambda(\mathcal{A}, \varepsilon/2, \delta)$.*

Proof. Given $\mathcal{A}$, we define an algorithm $\mathcal{A}'$ as follows. First, $\mathcal{A}'$ runs the algorithm $\mathcal{A}$ on $\tilde{\mathcal{C}}$ to query $t \geq \Lambda(\mathcal{A}, \varepsilon/2, \delta)$ labels and outputs a classifier h_A. By definition, $d(h_A, h^*) \leq \varepsilon/2$ with probability at least $1 - \delta$. Next, for each $\mathcal{C}_i$, the algorithm $\mathcal{A}'$ outputs the classifier $\hat{h}_i \in \mathcal{C}_i$, which is given by $\hat{h}_i = \arg\min_{h_i \in \mathcal{C}_i} d(h_i, h_A)$.

We claim that $\mathcal{A}'$ outputs the desired classifier for each $\mathcal{C}_i$, that is, $\mathrm{err}(\hat{h}_i) - \nu_i \leq \varepsilon$ holds with probability at least $1 - \delta$.

By Definition 3, bounding $\mathrm{err}(\hat{h}_i)$ is equivalent to bounding $d(\hat{h}_i, h^*)$. Let $h^*_i = \arg\min_{h_i \in \mathcal{C}_i} \mathrm{err}(h_i)$ so $\nu_i = d(h^*_i, h^*)$. By the triangle inequality,

$$d(\hat{h}_i, h^*) \leq d(\hat{h}_i, h_A) + d(h_A, h^*). \tag{3}$$

The first term can be bounded as

$$d(\hat{h}_i, h_A) \leq d(h^*_i, h_A) \leq d(h^*_i, h^*) + d(h^*, h_A) \tag{4}$$

using the definition of $\hat{h}_i$ and the triangle inequality. Combining Eqs (3) and (4) yields that

$$d(\hat{h}_i, h^*) \leq d(h^*_i, h^*) + 2d(h_A, h^*) \leq \nu_i + 2 \cdot \frac{\varepsilon}{2} = \nu_i + \varepsilon.$$

The proof is now complete. □

Remark 1. Theorem 1 provides a general guarantee, namely, an algorithm $\mathcal{A}$ which achieves distribution-independent query complexity $\Lambda(\mathcal{A}, \varepsilon/2, \delta)$ on the combined hypothesis space $\tilde{\mathcal{C}}$ derives an algorithm $\mathcal{A}'$ to achieve query complexity $\tilde{\Lambda}(\mathcal{A}', \varepsilon, \delta)$ on $\mathcal{C}_1, \ldots, \mathcal{C}_k$. This result enables the application of traditional AL methods to solve the problem of AL for multiple target models. Note that Theorem 1 also works for multi-class classification, it is applicable for a wide range of existing active learning algorithms, such as [7], [27].

By applying Theorem 1 and query complexity result in finite VC dimension [6, Corollary 1], we can immediately get the following corollary for multiple models in binary classification.

Corollary 1. *Consider binary classification tasks. Given k hypothesis spaces $\mathcal{C}_1, \ldots, \mathcal{C}_k$. Suppose that $h^* \in \tilde{\mathcal{C}} = \bigcup_{i=1}^k \mathcal{C}_i$, and $\tilde{\mathcal{C}}$ has a finite VC dimension $d < \infty$. Then, for any $\varepsilon \in (0, 1/2)$, $\delta \in (0, 1/4)$, there exists an active learning algorithm $\bar{\mathcal{A}}$ which achieves the query complexity $\tilde{\Lambda}(\bar{\mathcal{A}}, \varepsilon, \delta, \mathcal{D}_X) = o(1/\varepsilon)$.*

Remark 2. Corollary 1 gives a general result for AL for multiple models in binary classification, suggesting a great potential. Concretely, it establishes that even when the true hypothesis may lie in any one of k different model classes, as long as their union has finite VC-dimension, active learning still enjoys a strictly sub-$(1/\varepsilon)$ distribution-dependent query complexity for multiple models. We note that, it is also easy to derive a distribution-independent query complexity for multiple target models using [28, Theorem 8.2] and Theorem 1 with filtering trivial distributions.

In Sections 4 and 5, we will show the potential of active learning in multiple models setting, and propose a more effective algorithm.

4 Potential Improvements of Active over Passive

In this section, we discuss the potential of AL under the multiple models setting. Our discussion will be focused on the realizable setting, leaving the agnostic setting for future work.

We first introduce the notion of disagreement coefficient, which roughly characterizes the behavior of the size of disagreement region $\mathrm{DIS}(\cdot)$ as a function of the hypotheses within a radius r around the classifier h. The formal definition is as follows.

Definition 4 (Disagreement region and coefficient)**.** *Suppose that $\mathcal{C}$ is a set of hypotheses. Given the data distribution $\mathcal{D}_X$, the disagreement region of $\mathcal{C}$ is defined as*

$$\mathrm{DIS}(\mathcal{C}) = \{\boldsymbol{x} \in \mathrm{supp}(\mathcal{D}_X) \mid \exists h, h' \in \mathcal{C} \text{ s.t. } h(\boldsymbol{x}) \neq h'(\boldsymbol{x})\},$$

where $\mathrm{supp}(\mathcal{D}_X)$ is the support of $\mathcal{D}_X$. Let $h \in \mathcal{C}$ be a classifier and $r_0 \geq 0$. The disagreement coefficient of h with respect to $\mathcal{C}$ on $\mathcal{D}_X$ is defined as

$$\theta_h^{\mathcal{C}}(r_0) = \sup_{r > r_0} \max\left\{\frac{\mathbb{P}_{\boldsymbol{x}\sim\mathcal{D}_X}\left[\boldsymbol{x} \in \mathrm{DIS}(\mathrm{B}_{\mathcal{C}}(h, r))\right]}{r}, 1\right\},$$

where $\mathrm{B}_{\mathcal{C}}(h, r) = \{g \in \mathcal{C} \mid d(h, g) \leq r\}$.

For empirical risk minimization of binary classification, upper bounds of the query complexity of passive learning algorithms are known for single hypothesis space [28].

Lemma 1 ([28])**.** *Consider the binary classification problem with a hypothesis space $\mathcal{C}$ of VC dimension d. The passive learning algorithm* ERM *achieves a query complexity $\Lambda(\mathsf{ERM})$ such that, for any $\mathcal{D}_X$ and any $\varepsilon, \delta \in (0, 1)$,*

$$\Lambda(\mathsf{ERM}, \varepsilon, \delta) \lesssim \frac{1}{\varepsilon}\left(d\,\mathrm{Log}(\theta_{h^*}^{\mathcal{C}}(\varepsilon)) + \mathrm{Log}\,\frac{1}{\delta}\right) \tag{5}$$

in the realizable case and

$$\Lambda(\mathsf{ERM}, \nu + \varepsilon, \delta) \lesssim \frac{\nu + \varepsilon}{\varepsilon^2}\left(d\,\mathrm{Log}(\theta_{h^*}^{\mathcal{C}}(\nu + \varepsilon)) + \mathrm{Log}\,\frac{1}{\delta}\right) \tag{6}$$

in the agnostic case, where $\theta_{h^*}^{\mathcal{C}}(\cdot)$ is the disagreement coefficient.

Consider the setting of $h^* \in \tilde{\mathcal{C}}$ but $h^* \notin \mathcal{C}_1 \cap \cdots \cap \mathcal{C}_k$. We believe this scenario is more common in real-world applications, as multiple models tend to exhibit diversity. In this way, $\max_i \Lambda_i(\mathsf{ERM})$ has the form of Eq (6).

To show the potential of AL under this setting, we take the CAL method [15] as an example, which is a representative and well-analyzed approach in the active learning literature [28]. CAL queries the examples from the disagreement region of a set of consistent hypotheses, i.e., $\mathrm{DIS}(V)$, where $V = \{h \in \mathcal{C} \mid h(\boldsymbol{x}) = y, \forall(\boldsymbol{x}, y) \in \mathcal{L}\}$. It achieves the query complexity of $\Lambda(\mathrm{CAL}, \varepsilon, \delta) \lesssim \mathcal{O}(\theta_{h^*}^{\mathcal{C}}(\varepsilon)\log(1/\varepsilon)\log(\theta_{h^*}^{\mathcal{C}}(\varepsilon)\log(1/\varepsilon)))$ in the realizable case and binary classification task. Applying Theorem 1 immediately yields the algorithm CAL′ which achieves the following query complexity for the multiple target models.

Corollary 2. *Given target models $\mathcal{C}_1, \ldots, \mathcal{C}_k$ with $h^* \in \tilde{\mathcal{C}} = \bigcup_{i=1}^k \mathcal{C}_i$. Suppose $\tilde{\mathcal{C}}$ has VC dimension $d < \infty$ and $\varepsilon, \delta \in (0, 1)$. Let* CAL′ *be the algorithm that applies* CAL *on $\tilde{\mathcal{C}}$ to obtain $\varepsilon/2$-good classifier, then outputs $\hat{h}_i = \arg\min_{h_i \in \mathcal{C}_i} d(h_i, h_A)$ for $i = 1, \ldots, k$. It holds in the binary classification that*

$$\tilde{\Lambda}(\mathsf{CAL}', \varepsilon, \delta) \lesssim \theta_{h^*}^{\tilde{\mathcal{C}}}(\varepsilon/2)\,\mathrm{Log}(2/\varepsilon) \cdot \left(d\,\mathrm{Log}(\theta_{h^*}^{\tilde{\mathcal{C}}}(\varepsilon/2)) + \mathrm{Log}\left(\frac{\mathrm{Log}(2/\varepsilon)}{\delta}\right)\right). \tag{7}$$

To illustrate the potential improvement, consider the query complexity of passive learning, which is heavily influenced by the worst hypothesis space, quantified by $\max_i \min_{h \in \mathcal{C}_i} \mathrm{err}(h)$. Assuming that $\max_i \min_{h \in \mathcal{C}_i} \mathrm{err}(h) > \varepsilon$, Lemma 1 implies for passive learning a query complexity upper bound, i.e., $\Omega(1/\epsilon)$, for multiple target models. On the other hand, CAL′ has an upper bound of query complexity $\Omega(\mathrm{Log}(1/\varepsilon))$ by Corollary 2, leaving a huge room for improvement for active learning in this setting, which is an intriguing area for future investigation. Now we proceed to examine the agnostic case (i.e., $h^* \notin \tilde{\mathcal{C}}$).

5 An Agnostic Disagreement-based AL method for Multiple Models

Given a labeled set $\mathcal{L}$ and multiple hypothesis spaces $\mathcal{C}_1, \ldots, \mathcal{C}_k$ with $h^* \notin \tilde{\mathcal{C}} = \bigcup_{i=1}^k \mathcal{C}_i$. Inspired by the RobustCAL method [5], which is a disagreement-based AL algorithm for the agnostic setting, we propose DIAM (i.e., DIsagreement-based AL for Multi-models) query strategy for the multiple target models problem. Specifically, for each $\mathcal{C}_i$, we define soft version space $\hat{V}_i = \{h \in \mathcal{C}_i \mid \text{e-err}_{\mathcal{L}}(h) - \inf_{g \in \mathcal{C}_i} \text{e-err}_{\mathcal{L}}(g) \leq \sigma_i\}$, where σ_i is a constant. The soft version space $\hat{V}_i$ is analogous to the version space V_i; it is a set of classifiers that are largely consistent with the labeled dataset, i.e., those with lower empirical errors. We propose to query the examples located in the joint disagreement regions, i.e., $\mathrm{DIS}(\hat{V}_1) \cap \mathrm{DIS}(\hat{V}_2) \cap \cdots \cap \mathrm{DIS}(\hat{V}_k)$, located in as many disagreement regions of different target models as possible, i.e., the examples of large values of $\sum_{i=1}^k \mathbb{I}[\boldsymbol{x} \in \mathrm{DIS}(\hat{V}_i)]$, and dynamically update the soft version spaces by removing the hypotheses that have greater errors. Finally, the algorithm outputs an arbitrary $h_i \in \hat{V}_i$.

The motivation behind our method is that the data located in the most possible disagreement regions of target models have a greater potential to reduce the soft version spaces $\hat{V}_1, \ldots, \hat{V}_k$, ultimately leading to fewer queries. In Sec. 5.3, we will further demonstrate that selecting data within these disagreement regions is akin to identifying the data with the highest leverage score. This offers a unique perspective in elucidating the effectiveness of our method.

Next, we propose a stream-based version of DIAM and present the theoretical results in Sec. 5.1. The algorithm is summarized in Algorithm 1. The hyperparameter q in the algorithm controls its level of conservativeness. A larger value of q leads to more rejections of less-informative unlabeled data in the online setting. After that, we further propose efficient implementations of DIAM for pool-based active deep classification and regression settings in Sec. 5.2 and 5.3, respectively, coupled with extensive empirical validation in Sec. 6. Note that, in pool-based AL setting, we can directly query the data by $\arg\max_{\boldsymbol{x} \in \mathcal{U}} \sum_{i=1}^k \mathbb{I}[\boldsymbol{x} \in \mathrm{DIS}(\hat{V}_i)]$, rather than tuning the hyperparameter q. In the following, we

theoretically and empirically demonstrate that such query strategy has a greater potential to reduce the soft version spaces $\hat{V}_1, \ldots, \hat{V}_k$, resulting in fewer queries.

To simplify the theoretical analysis, we first propose an online version of DIAM. It is summarized in Algorithm 1. The hyperparameter q in the algorithm controls its level of conservativeness. A larger value of q leads to more rejections of less-informative unlabeled data in the online setting.

5.1 Theoretical Analysis

This section provides theoretical analysis of Algorithm 1. Since we are considering the agnostic setting, it is necessary to model the noise. Here we employ the commonly used Tsybakov noise condition [63].

Condition 1 (Tsybakov noise, [63]). *Let $a \in [1, \infty)$ and $\alpha \in [0, 1]$ be parameters. The Tsybakov noise condition refers to that*

$$\mathbb{P}\left(\boldsymbol{x} : h(\boldsymbol{x}) \neq f^\star(\boldsymbol{x})\right) \leq a\left(\operatorname{err}(h) - \operatorname{err}\left(f^\star\right)\right)^\alpha$$

for all $h \in \mathcal{C}$, where $f^\star$ attains the infimum $\inf_{h\in\mathcal{C}} \operatorname{err}(h)$.

We assume that Condition 1 is satisfied for each target model $\mathcal{C}_i$. Consider a conservative situation where the hyperparameter $q = 1$, and choosing the constants σ_i in the DIAM-online algorithm to be in the same form as in the RobustCAL method [5], which takes into account the properties of the noise, hypothesis space, and disagreement coefficient. Note that, the confidence arguments σ_i vary with m. Here, we follow the updating scheme in [28, Sec. 5.2] to update σ_i. We refer the readers to the reference for the updating details. We establish the following result for DIAM-online. The proof is deferred to the appendix.

Theorem 2. *Consider binary classification tasks. Given target models $\mathcal{C}_1, \ldots, \mathcal{C}_k$, in which $h^* \notin \tilde{\mathcal{C}}$ and each $\mathcal{C}_i$ has VC dimensions $d_i < \infty$ and satisfies Condition 1 with parameters a_i and α_i. Let $h_i^* = \arg\min_{h_i \in \mathcal{C}_i} \operatorname{err}(h_i)$ and $\varepsilon, \delta \in (0, 1)$.*

Given data distribution $\mathcal{D}_X$, the algorithm DIAM-online *outputs the desired classifier $h_i \in \mathcal{C}_i$ with $\operatorname{err}(h_i) \leq \operatorname{err}(h_i^*) + \varepsilon$ for each $\mathcal{C}_i$ with probability at least $1 - \delta$ with $t \geq \min\{\tilde{\Lambda}_1, \tilde{\Lambda}_2\}$ queries, where*

$$\tilde{\Lambda}_1 \lesssim \sum_{i=1}^{k} a_i^2 \theta_{h_i^*}^{\mathcal{C}_i}\left(a_i \varepsilon^{\alpha_i}\right) \varepsilon^{2\alpha_i - 2} \cdot \left(d_i \operatorname{Log} \theta_{h_i^*}^{\mathcal{C}_i}(a_i \varepsilon^{\alpha_i}) + \operatorname{Log}\left(\frac{\operatorname{Log}(a_i/\varepsilon)}{\delta}\right)\right) \operatorname{Log} \frac{1}{\varepsilon}$$

and

$$\tilde{\Lambda}_2 \lesssim \sum_{i=1}^{k} \theta_{h_i^*}^{\mathcal{C}_i}(\nu_i + \varepsilon)\left(\frac{\nu_i^2}{\varepsilon^2} + \operatorname{Log}\frac{1}{\varepsilon}\right) \cdot \left(d_i \operatorname{Log} \theta_{h_i^*}^{\mathcal{C}_i}(\nu_i + \varepsilon) + \operatorname{Log}\left(\frac{\operatorname{Log}(1/\varepsilon)}{\delta}\right)\right).$$

Theorem 2 considers a general situation with arbitrary target models and data distributions, even the unlabeled data will never fall into the joint disagreement regions. However, one may be more interested in the situation that if we can always query the $\boldsymbol{x}$ such that if $\boldsymbol{x}$ falls in every $\mathrm{DIS}(\hat{V}_i)$. Next, we prove that in such an ideal situation, DIAM-online will achieve a better query complexity than applying Theorem 1 to CAL even under the setting of $h^* \in \tilde{\mathcal{C}}$.

Theorem 3. *Considering binary classification tasks. Given target models $\mathcal{C}_1, \ldots, \mathcal{C}_k$. Assume $\tilde{\mathcal{C}}$ has VC dimension $d < \infty$ and $h^* \in \tilde{\mathcal{C}}$, each $\mathcal{C}_i$ has VC dimensions $d_i < \infty$ and satisfies Condition 1. Suppose that $\mathcal{X}$ satisfies that $\mathrm{DIS}(\hat{V}_1) = \cdots = \mathrm{DIS}(\hat{V}_k)$. Suppose that $\delta \in (0, 1)$, $\varepsilon \in (0, 1/e)$ and $\max_i \min_{h_i \in \mathcal{C}_i} \operatorname{err}(h_i) \leq \frac{\ln 2}{2}\varepsilon$. It then holds that*

$$\tilde{\Lambda}(\text{DIAM-online}, \varepsilon, \delta) \leq \tilde{\Lambda}(\text{CAL}', \varepsilon, \delta)\,. \tag{8}$$

The key in the proof is comparing the disagreement coefficients defined on different functions and hypothesis spaces, i.e., $\theta_{h_m^*}^{\mathcal{C}_m}$ and $\theta_{h^*}^{\tilde{\mathcal{C}}}$. We defer the proof to the appendix.

5.2 Efficient DIAM Implementation for Deep Classification

It is generally considered a non-trivial task to find disagreed pairs of classifiers from a set of hypotheses for a given $\boldsymbol{x}$. Commonly used methods include random sampling functions from the hypothesis space for validation and selecting the data close to the decision boundary. However, they can be expensive or inaccurate, especially in the deep learning setting.

To estimate efficiently the disagreement regions for neural networks, we propose to exploit the predictions of unlabeled data during later epochs in the training phase, typically after the network converges. Recall the definition of disagreement region $\mathrm{DIS}(\hat{V}_i)$, we should first identify the hypotheses which are largely consistent with the labeled data, and then determine whether there exists a pair of hypotheses which disagree on the given unlabeled data. To this end, we utilize the hypotheses obtained from the later training epochs of the network, as they are more likely to have converged and give consistent predictions. For the first goal, we construct the set of well-performed hypotheses by taking the hypotheses from the intermediate training epochs whose training errors are relatively small. For the second goal, we verify whether the example $\boldsymbol{x}$ falls into $\mathrm{DIS}(\hat{V}_i)$ by examining whether some well-performed hypotheses have inconsistent predictions on $\boldsymbol{x}$.

More concretely, we assume that the minimum empirical error $\inf_{h\in\mathcal{C}_i} \text{e-err}_{\mathcal{L}}(h)$ is attained by the hypothesis trained in the last epoch of the training process. Therefore, we heuristically select the hypotheses from the latter half of the training epochs to form the well-performing hypothesis set. This heuristic approach is based on the observation that hypotheses in the latter training epochs, according to the training loss curve, usually have smaller empirical errors. For each i, let $\hat{h}_i^j \in \mathcal{C}_i$ be the hypothesis obtained at training epoch j, then $\hat{V}_i$ can be defined as $\{\hat{h}_i^j \mid j = \lfloor \frac{T}{2} \rfloor, \lfloor \frac{T}{2} \rfloor + 1, \ldots, T\}$, where T denotes the total number of training epochs. To verify whether $\boldsymbol{x} \in \mathrm{DIS}(\hat{V}_i)$, we can compare the predictions of the hypotheses in $\hat{V}_i$ on the unlabeled data. If an instance $\boldsymbol{x}$ receives different predicted labels from the hypotheses in $\hat{V}_i$, it indicates that $\boldsymbol{x} \in \mathrm{DIS}(\hat{V}_i)$.

We also note that training deep models is much more expensive, thus the query batch size for deep models is usually large. To avoid overmuch information redundancy,

Algorithm 1 The DIAM-online Algorithm

Input: hypothesis spaces $\mathcal{C}_1, \ldots, \mathcal{C}_k$, labeled set $\mathcal{L}$, hyperparameters $q; \sigma_i, i = 1, \ldots, k$; query budget B.
Output: $h_1, \ldots, h_k$, where $h_i \in \hat{V}_i$ for $i = 1, \ldots, k$.

$m \leftarrow 0$; $n_q \leftarrow B$
$\hat{V}_i \leftarrow \mathcal{C}_i, \forall i = 1, \ldots, k$
while $n_q > 0$ and $m < 2^B$ **do**
  $m \leftarrow m + 1$
  Request an unlabeled data $\boldsymbol{x}_m$
  if $\sum_i \mathbb{I}[\boldsymbol{x}_m \in \mathrm{DIS}(\hat{V}_i)] \geq q$ **then**
    Query $h^*(\boldsymbol{x}_m)$
    $\mathcal{L} \leftarrow \mathcal{L} \cup \{(\boldsymbol{x}_m, h^*(\boldsymbol{x}_m))\}$
    $n_q \leftarrow n_q - 1$
  end if
  if m is a power of 2 **then**
    $\hat{V}_i \leftarrow \{h \in \hat{V}_i | \text{e-err}_{\mathcal{L}}(h) - \inf_{g \in \hat{V}_i} \text{e-err}_{\mathcal{L}}(g) \leq \sigma_i\}, \forall i = 1, \ldots, k$.
    Update σ_i following the procedures described in RobustCAL
  end if
end while
return arbitrary $h_i \in \hat{V}_i$ for each $i = 1, \ldots, k$

Algorithm 2 The DIAM Algorithm for Deep Classification.

Input: hypothesis spaces $\mathcal{C}_1, \ldots, \mathcal{C}_k$, labeled set $\mathcal{L}$, unlabeled set $\mathcal{U}$, training epochs T, query batch size τ.
Output: $h_1, \ldots, h_k$, where $h_i \in \hat{V}_i$ for $i = 1, \ldots, k$.

while labeling budget is not exhausted **do**
  for $i = 1, \ldots, k$ **do**
    Train model i for T epochs on $\mathcal{L}$ and obtain $\hat{h}_i^t$ from epoch t for $t = 1, \ldots, T$
    $\hat{V}_i \leftarrow \{\hat{h}_i^j \mid j = \lfloor \frac{T}{2} \rfloor, \lfloor \frac{T}{2} \rfloor + 1, \ldots, T\}$
    Calculate S^i by Eq. (10)
    $S^i \leftarrow \frac{1}{2}((S^i)^\top + S^i)$
  end for
  $\boldsymbol{v}_m \leftarrow \sum_i \mathbb{I}[\boldsymbol{x}_m \in \mathrm{DIS}(\hat{V}_i)]$ for $m = 1, \ldots, n_u$
  Solve minimization problem (9) to obtain $\boldsymbol{b}$
  $J \leftarrow$ indices of the largest τ coordinates in $\boldsymbol{b}$
  Query $h^*(\boldsymbol{x}_j)$ for all $j \in J$
  $\mathcal{L} \leftarrow \mathcal{L} \cup \{(\boldsymbol{x}_j, h^*(\boldsymbol{x}_j)) \mid j \in J\}$
  $\mathcal{U} \leftarrow \mathcal{U} \setminus \{\boldsymbol{x}_j \mid j \in J\}$
end while
$h_i \leftarrow \arg\min_{h \in \mathcal{C}_i}(\text{e-err}_{\mathcal{L}}(h))$ for each $i = 1, \ldots, k$
return h_i for each $i = 1, \ldots, k$

we further consider the selection diversity in the batch querying setting. Specifically, we minimize the similarities of the selected data by adding a diversity term to the selection criterion along with the informativeness measurement. However, one challenge here is that the example with high diversity score for one model may not hold the score with another model. Therefore, the traditional diversity measurements can hardly be directly applied.

To tackle this problem, we propose to exploit the powerful representation learning ability of deep models. Specifically, note that the deep models will implicitly learn the representation during the training process, and different feature representations for the same data will be extracted by the multiple networks, i.e., the output of the penultimate layer. In this case, we can try to identify those examples with high diversity scores for most of the models. To achieve this goal, we minimize the information redundancy under each target model, which can be formulated as

$$\begin{aligned} \min_{\boldsymbol{b}} \ & \sum_i \boldsymbol{b}^\top S^i \boldsymbol{b} + \beta \boldsymbol{b}^\top \boldsymbol{v} \\ \text{s.t.} \ & \boldsymbol{b} \in \{0, 1\}^{n_u}. \end{aligned} \tag{9}$$

Here, $\beta < 0$ is the trade-off parameter, $\boldsymbol{v} = [v_1, v_2, \ldots, v_{n_u}]^\top$ is the vector of informativeness scores in which $v_j = \sum_i \mathbb{I}[\boldsymbol{x}_j \in \mathrm{DIS}(\hat{V}_i)]$ for all $j = 1, \ldots, n_u$, and S^i is the similarity matrix of unlabeled data under the representation of target model i. The first term in Eq. (9) accounts for estimating the diversity of unlabeled data. Note that S is a similarity matrix. During the optimization of $\boldsymbol{b}$, the rows that are less similar to the others will be assigned a higher value of $\boldsymbol{b}$. This indicates that these instances have a higher degree of uniqueness within the dataset. We simply implement S^i using the linear kernel, i.e., taking the inner product of features as their similarity value. One challenge is that it can be problematic to use the same distance metric across the representations of multiple models because of the different ranges of feature values. Therefore, we take the nearest neighbors as follows,

$$S_{uv}^i = \begin{cases} 1 & \text{if } u \text{ is } v\text{'s neighbor} \\ 0 & \text{otherwise} \end{cases}. \tag{10}$$

The neighbor is calculated by the Euclidean distance between the data representations. Specifically, we consider a data point as a neighbor of a specific instance if its Euclidean distance to that instance falls within the smallest 1% of distances across all unlabeled data. To facilitate the computation, we symmetrize each S^i by replacing it with $\frac{1}{2}((S^i)^\top + S^i)$ and relax $\boldsymbol{b}$ to $[0, 1]^{n_u}$. With these modifications, we can solve the objective Eq. (9) efficiently using existing quadratic programming toolboxes. Subsequently, we query the unlabeled data with higher values of $\boldsymbol{b}$. The implementation is summarized in Algorithm 2.

The described implementation of DIAM for deep models is efficient. It evaluates the unlabeled data using hypotheses obtained in later training epochs and runs in time proportional to the product of the size of the well-performing hypotheses set and the computational cost of informativeness is comparable to that of the entropy method. Then, it solves a quadratic program to make the data selection.

5.3 Efficient DIAM Implementation for Deep Regression

In this section, we implement the DIAM method for the deep regression tasks. In our DIAM algorithm, we need first to find a set of hypotheses having good performances on the labeled dataset, i.e., $\hat{V}_i = \{h \in \mathcal{C}_i | \text{e-err}_{\mathcal{L}}(h) - \inf_{g \in \mathcal{C}_i} \text{e-err}_{\mathcal{L}}(g) \leq \sigma_i\}$. Then, given an unlabeled data point, we estimate whether there exists a pair of hypotheses in $\hat{V}_i$ that exhibit highly inconsistent predictions. Our implementation is based on the fact that a neural network is

Algorithm 3 The DIAM-svd Algorithm for Regression.

Input: feature matrices of the data under different models $L^i, U^i, i = 1, ..., k$; query batch size τ.
Output: the set of indices of the selected unlabeled data.

$\boldsymbol{s}^1, \ldots, \boldsymbol{s}^k, \boldsymbol{v} \leftarrow$ zero vector of length n_u
for $i = 1, ..., k$ **do**
  $A^i, \Sigma^i, (B^i)^\top \leftarrow \text{SVD}(L^i)$ ▷ Perform SVD on L^i.
  for each $j = 1, \ldots, n_u$ **do**
    $\boldsymbol{x}_j^\top \leftarrow j$-th row of U^i
    $\boldsymbol{s}_j^i \leftarrow \|(\Sigma^i)^{-1}(B^i)^\top \boldsymbol{x}_j\|_2$
  end for
end for
for $j = 1, \ldots, n_u$ **do**
  $\boldsymbol{v}_j \leftarrow \sum_{i=1}^k \mathbb{I}[\boldsymbol{s}_j^i$ is among the largest τ coordinates of $\boldsymbol{s}^i]$
end for
return the indices of the largest τ coordinates in $\boldsymbol{v}$

Algorithm 4 The DIAM-leverage Algorithm for Regression.

Input: feature matrices of the data under different models $L^i, U^i, i = 1, ..., k$; query batch size τ.
Output: the set of indexes of the selected unlabeled data.

$\boldsymbol{s}^1, \ldots, \boldsymbol{s}^k, \boldsymbol{v} \leftarrow$ zero vector of length n_u
for $i = 1, ..., k$ **do**
  $X^i \leftarrow \left[\begin{smallmatrix} U^i \\ L^i \end{smallmatrix}\right]$
  for $j = 1, ..., n_u$ **do**
    $\boldsymbol{s}_j^i \leftarrow$ the j-th leverage score of X^i
  end for
end for
for $j = 1, \ldots, n_u$ **do**
  $\boldsymbol{v}_j \leftarrow \sum_{i=1}^k \mathbb{I}[\boldsymbol{s}_j^i$ is among the largest τ coordinates of $\boldsymbol{s}^i]$
end for
return the indices of the largest τ coordinates in $\boldsymbol{v}$

composed by a feature extractor (i.e., the backbone) and a fully connected layer (i.e., the linear prediction layer). Note that the last layer of deep models is usually a fully connected layer, whose parameters are linear in the output of the previous layer. Therefore, we can approximate the active deep regression problem as the following: given the features of the data (e.g., the output of the network backbone), we need to learn a linear regression model with the least number of queries. In this way, multiple target models may lead to different feature representations, but the hypothesis spaces remain the same.

Next, we introduce our implementations of DIAM for regression tasks. Since our implementations identify the disagreement region of each model individually, we omit the superscript of the index of the target model in the remainder of this section.

5.3.1 An SVD-based Implementation

Recall that we need to maintain a soft version space and verify whether there exists a pair of hypotheses in the soft version space that have highly inconsistent predictions on the unlabeled data. This can be formulated as follows (using mean-square loss).

Problem 1. *Let $\boldsymbol{w} \in \mathbb{R}^c$ be the linear model parameters, $L \in \mathbb{R}^{n_l \times c}$ and $U \in \mathbb{R}^{n_u \times c}$ be the feature matrices of labeled and unlabeled data, respectively. Here $n_l > c$ and L is assumed to have full column rank. Define $\bar{\boldsymbol{w}} = \arg\min_{\boldsymbol{w}} \text{e-err}_{\mathcal{L}}(\boldsymbol{w})$. Given a set of hypotheses(i.e., a soft version space) such that $\tilde{V} = \{\boldsymbol{w} \mid \|L\boldsymbol{w} - L\bar{\boldsymbol{w}}\|_2^2 \leq \sigma\}$ and an unlabeled data $\boldsymbol{x}$, the problem asks to verify whether there exist $\boldsymbol{w}_1, \boldsymbol{w}_2 \in \tilde{V}$ such that $|\boldsymbol{w}_1^\top \boldsymbol{x} - \boldsymbol{w}_2^\top \boldsymbol{x}| \geq \hat{\sigma}$.*

In the definition above, σ and $\hat{\sigma}$ are hyperparameters, where σ controls the empirical errors of the hypotheses in $\tilde{V}$ and $\hat{\sigma}$ is the threshold of prediction inconsistency used to identify informative unlabeled data. Next, we explain how to solve the problem. Since L has full column rank, $\tilde{V}$ is an ellipsoid in $\mathbb{R}^c$ centered at $\bar{\boldsymbol{w}}$. To see this, we first provide the definition of ellipsoid as follows

Definition 5 ([23]). *If M is a real, symmetric, c-by-c positive-definite matrix, and $\bar{\boldsymbol{w}}$ is a vector in $\mathbb{R}^c$, then the set points $\boldsymbol{w}$ that satisfy the equation*

$$(\boldsymbol{w} - \bar{\boldsymbol{w}})^\top M(\boldsymbol{w} - \bar{\boldsymbol{w}}) = 1 \tag{11}$$

is an c-dimensional ellipsoid centered at $\bar{\boldsymbol{w}}$.

Next, we perform singular value decomposition (SVD) on L to obtain $L = A\Sigma B^\top$, where A and B are unitary matrices. Then $\|L(\boldsymbol{w} - \bar{\boldsymbol{w}})\|_2^2 = \|A\Sigma B^\top(\boldsymbol{w} - \bar{\boldsymbol{w}})\|_2^2 = \|\Sigma B^\top(\boldsymbol{w} - \bar{\boldsymbol{w}})\|_2^2$. The last equation uses the property of unitary matrix that it is an isometry with respect to ℓ_2-norm. Without loss of generality, we may assume that $\bar{\boldsymbol{w}} = 0$; otherwise we can let $\tilde{\boldsymbol{w}} = \boldsymbol{w} - \bar{\boldsymbol{w}}$ and work with $\tilde{\boldsymbol{w}}$ instead, as such translation does not change the answer to our problem.

Now, $\tilde{V}$ is a centered ellipsoid given by $\{\boldsymbol{w} | \|\Sigma B^\top \boldsymbol{w}\|_2^2 \leq \sigma\}$. It is easy to see that the set formed by all $\boldsymbol{w}_1 - \boldsymbol{w}_2$ with $\boldsymbol{w}_1, \boldsymbol{w}_2 \in \tilde{V}$ is an ellipsoid twice as large as $\tilde{V}$, i.e.

$$\{\boldsymbol{w}_1 - \boldsymbol{w}_2 \mid \boldsymbol{w}_1, \boldsymbol{w}_2 \in \tilde{V}\} = \{\boldsymbol{w} \mid \|\Sigma B^\top \boldsymbol{w}\|_2^2 \leq 4\sigma\}.$$

After rescaling $\boldsymbol{w}$, our problem can be rephrased as determining whether there exists $\boldsymbol{w} \in \tilde{V}$ such that $\boldsymbol{w}^\top \boldsymbol{x} \geq \hat{\sigma}/2$. Here, we remove the absolute value due to the symmetry of the ellipsoid.

We further transform the ellipsoid $\tilde{V}$ into a ball by letting $\boldsymbol{\pi} = \Sigma B^\top \boldsymbol{w}$, so $\tilde{V}$ becomes $\{\boldsymbol{\pi} | \|\boldsymbol{\pi}\|_2^2 \leq \sigma\}$. Then $\boldsymbol{w}^\top \boldsymbol{x} = \boldsymbol{\pi}^\top \cdot \Sigma^{-1} B^\top \boldsymbol{x}$ and the problem is now determining whether there exists $\boldsymbol{\pi}$ with $\|\boldsymbol{\pi}\|_2^2 \leq \sigma$ satisfies $\boldsymbol{\pi}^\top \cdot \Sigma^{-1} B^\top \boldsymbol{x} = \hat{\sigma}/2$, which is equivalent to determine whether the hyperplane $\boldsymbol{\pi}^\top \cdot \Sigma^{-1} B^\top \boldsymbol{x} = \hat{\sigma}/2$ intersects the ball $\|\boldsymbol{\pi}\|_2^2 = \sigma$. This can be solved by calculating the distance from the origin to the hyperplane, which is given by

$$\hat{\sigma}/(2\|\Sigma^{-1} B^\top \boldsymbol{x}\|_2). \tag{12}$$

Our problem has a positive answer if and only if Eq. (12) is no larger than σ. This implies that if the data $\boldsymbol{x}$ falls into the disagreement region, it will have a larger value of $\|\Sigma^{-1} B^\top \boldsymbol{x}\|_2$. Therefore, in our implementation, we can rank the unlabeled data with this value and query the top-rated ones without deciding the values of σ and $\hat{\sigma}$. We refer to this method as DIAM-svd.

5.3.2 A Leverage-score-based Implementation

Here, we modify the definition of $\tilde{V}$ to include the requirement that the candidate models have similar predictions to the ERM hypothesis not only on the labeled data but also on the unlabeled data. Let $X = [\begin{smallmatrix} U \\ L \end{smallmatrix}] \in \mathbb{R}^{(n_u+n_l)\times c}$ and redefine $\tilde{V} = \{\boldsymbol{w} \,|\, \|X\boldsymbol{w} - X\bar{\boldsymbol{w}}\|_2^2 \leq \sigma\}$. This leads to the following problem.

Problem 2. *Let $\boldsymbol{w} \in \mathbb{R}^c$ be the linear model parameters, $L \in \mathbb{R}^{n_l \times c}$ and $U \in \mathbb{R}^{n_u \times c}$ be the feature matrices of labeled and unlabeled data of model i, respectively. Here $X = [\begin{smallmatrix} U \\ L \end{smallmatrix}]$ is assumed to have full column rank. Define $\bar{\boldsymbol{w}} = \arg\min_{\boldsymbol{w}} \text{e-err}_{\mathcal{L}}(\boldsymbol{w})$. Given a set of hypotheses such that $\tilde{V} = \{\boldsymbol{w} \,|\, \|X\boldsymbol{w} - X\bar{\boldsymbol{w}}\|_2^2 \leq \sigma\}$ and an unlabeled data $\boldsymbol{x}_j^\top$, which is the j-th row of X, the problem asks to verify that whether there exist $\boldsymbol{w}_1, \boldsymbol{w}_2 \in \tilde{V}$ such that $|\boldsymbol{w}_1^\top \boldsymbol{x}_j - \boldsymbol{w}_2^\top \boldsymbol{x}_j| \geq \hat{\sigma}$.*

Similarly, we can apply SVD to obtain $X = \bar{A}\bar{\Sigma}\bar{B}^\top$. Note that X contains all the unlabeled data, therefore, we can write the unlabeled data $\boldsymbol{x}_j$ as $\boldsymbol{x}_j^\top = \boldsymbol{e}_j^\top \bar{A}\bar{\Sigma}\bar{B}^\top$, where $\boldsymbol{e}_j$ is the j-th standard basis vector. By a similar argument to the previous section, we can transform to problem to deciding whether

$$\hat{\sigma}/(2\|\bar{\Sigma}^{-1}\bar{B}^\top \boldsymbol{x}_j\|_2) \leq \sigma,$$

or, equivalently,

$$\|\bar{\Sigma}^{-1}\bar{B}^\top \cdot \bar{B}\bar{\Sigma}\bar{A}^\top \boldsymbol{e}_j\|_2 \geq \hat{\sigma}/(2\sigma),$$

which is exactly

$$\|\bar{A}^\top \boldsymbol{e}_j\|_2 \geq \hat{\sigma}/(2\sigma).$$

Note that $\|\boldsymbol{e}_j^\top \bar{A}\|_2^2$ is exactly the leverage score [16] of the row $\boldsymbol{x}_j^\top$ in X. Therefore, in our implementation, we can rank the unlabeled data based on their leverage scores and query the top-ranked ones, rather than tuning the values of σ and $\hat{\sigma}$. We refer to this method as DIAM-leverage.

Note that our analysis indicates that selecting data within the disagreement region is equivalent to identifying data with the highest leverage score. This finding offers a unique perspective on understanding the effectiveness of these data. The leverage score can be interpreted as a measure of difficulty in representing a given instance as a linear representation of the remaining data [16]. An instance with a higher leverage score suggests a more difficult representation using other instances, suggesting that it possesses unique information and is therefore informative.

We summarize the above two implementations of DIAM for deep regression tasks in Algorithms 3 and 4.

6 Experiment

6.1 Empirical Settings

We validate our method first on classification tasks, using a scenario involving multiple target models. To create this scenario, we utilize the results of a recent neural architecture search (NAS) method called OFA [9], which is designed to efficiently search for model architectures that meet the hardware constraints of different devices by training a single supernet. They have published the effective architectures, from which we use the architectures optimized for the Samsung mobile phones, which include Samsung S7 Edge, Samsung Note8 and Samsung Note10, as our target models. Each of the phone model has 4 architectures, resulting in a total of 12 architectures. These architectures are pruned from a MobileNetV3 (which is the super-net), but have significant differences in terms of prediction time and accuracy. Their Multiply-Accumulate Operations (MACs) range from 66M to 237M, illustrating their diversity. Specifically, we take the following 12 target models, the details of which can be found at https://github.com/mit-han-lab/once-for-all:

- s7edge_lat@88ms_top1@76.3_finetune@25
- s7edge_lat@58ms_top1@74.7_finetune@25
- s7edge_lat@41ms_top1@73.1_finetune@25
- s7edge_lat@29ms_top1@70.5_finetune@25
- note8_lat@65ms_top1@76.1_finetune@25
- note8_lat@49ms_top1@74.9_finetune@25
- note8_lat@31ms_top1@72.8_finetune@25
- note8_lat@22ms_top1@70.4_finetune@25
- note10_lat@22ms_top1@76.6_finetune@25
- note10_lat@16ms_top1@75.5_finetune@25
- note10_lat@11ms_top1@73.6_finetune@25
- note10_lat@8ms_top1@71.4_finetune@25

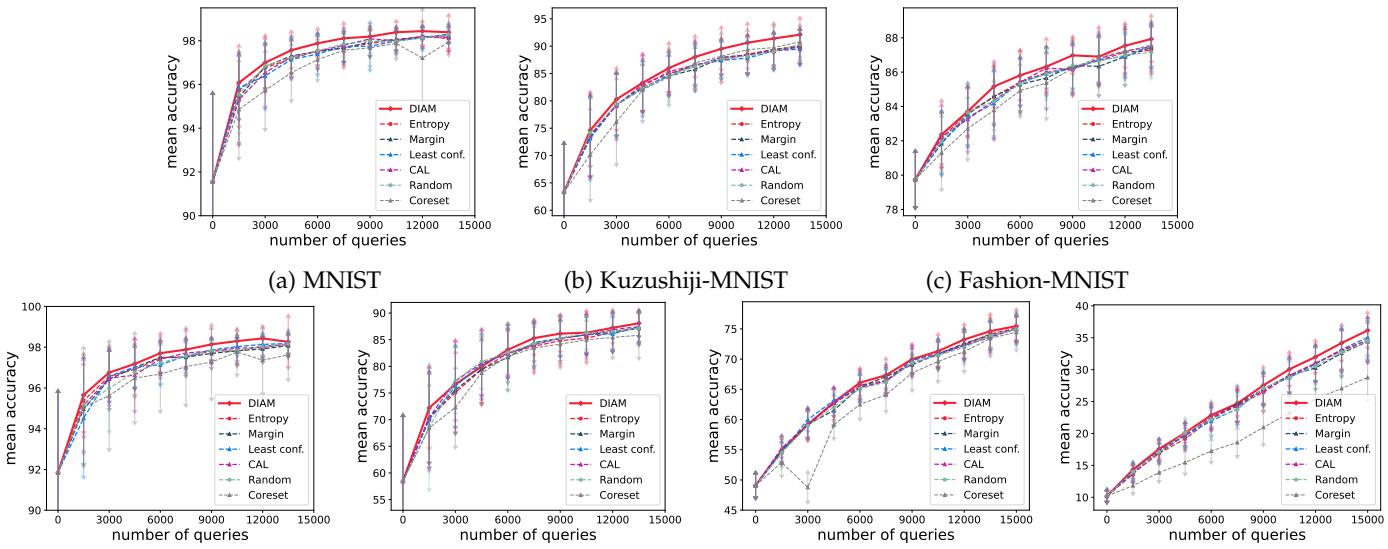
Table 1: The specifications of the datasets in the experiments.

Dataset	#Training	#Testing	#Label	License
MNIST	60,000	10,000	10	CC BY-SA 3.0
F.MNIST	60,000	10,000	10	MIT
K.MNIST	60,000	10,000	10	CC BY-SA 4.0
EMNIST let.	88,800	14,800	26	CC0 1.0
EMNIST dig.	240,000	40,000	10	CC0 1.0
CIFAR-10	50,000	10,000	10	Apache License 2.0
CIFAR-100	50,000	10,000	100	Apache License 2.0

We compare the following query strategies in our experiments.

- DIAM: Our proposed method in this paper, which selects data located in the joint disagreement regions of multiple target models, i.e., Algorithm 2.
- CAL [15]: This strategy queries data that falls into the disagreement region of any of the target models. It has a bounded query complexity for the multiple target models setting according to Theorem 1.
- Entropy [43]: This strategy selects data with the highest prediction entropy, based on the mean entropy across all target models, to accommodate the novel problem setting.
- Least Confidence [55]: This strategy queries data with the least prediction confidence, based on the mean confidence values across all target models.
- Margin [52]: This strategy selects data with the minimum prediction margin, based on the mean margin values across all target models.
- Coreset [53]: This strategy queries the most representative data. The distance is calculated using the features extracted by the supernet in OFA [9].
- Random: This strategy queries data randomly and is exactly the passive learning method. Note that the trivial upper bound $\tilde{\Lambda}(\mathsf{ERM}') \leq \max_i \Lambda_i(\mathsf{ERM})$ is tight without extra assumptions.

For the specifications of the classification datasets, we on the one hand consider the Optical Character Recogni-



(a) MNIST (b) Kuzushiji-MNIST (c) Fashion-MNIST

(d) EMNIST-digits (e) EMNIST-letters (f) CIFAR-10 (g) CIFAR-100

Figure 1: The learning curves with the mean accuracy of the target models of the compared methods. The error bars indicate the standard deviation of the performances of target models.

tion (OCR) application, which is a representative machine learning system that needs to be deployed on diverse devices. We employ five commonly used hand-writing characters classification benchmarks in our experiments, i.e., the MNIST [41], Fashion-MNIST [65], Kuzushiji-MNIST [13], EMNISTdigits and EMNISTletters [14] datasets. On the other hand, we employ two commonly used image classification datasets, CIFAR-10 and CIFAR-100 [39]. The dataset specifications are summarized in Table 1. We adopt the pool-based active learning setting, where we initially label 3000 randomly chosen data points for training and used the remaining data as the unlabeled pool. In each iteration, the compared sampling methods select $\tau = 1500$ unlabeled examples for querying and then retrain the models. The mean and standard deviation of the accuracy of multiple target models are reported.

Regarding the setting of model training, we mainly follow the training configurations of OFA. The hyperparameters are set to their default values in the project. For example, the learning rate is set to 7.5×10^{-3}, the batch size is set to 128, and the SGD optimizer is employed with a momentum of 0.9. Since the set of initially labeled data is small, a limited number of training epochs is adopted to mitigate the risk of over-fitting. Specifically, we initialize the models with the pre-trained weights on the image-net dataset and then fine-tune them for 20 epochs using labeled data. The code is publicly available at https://github.com/tangypnuaa/DIAM.

We run our experiments on three cloud servers, each equipped with 128GB of memory, four RTX 2080 graphic cards and an Intel Xeon Silver 4110 @ 2.10GHz CPU with eight cores. Each of the compared methods is run on a separate graphics card and the resource occupation of each individual process is reported. The minimum memory requirements for training and validating the models are 10GB main memory and 11GB CUDA memory, respectively. When running the Coreset and DIAM methods, an additional 10GB of main memory is needed to store the distance matrix.

6.2 Results

We report in Fig. 1 the trend of mean accuracy of multiple target models as the number of queries increases. The error bars represent the standard deviation of the performances of the multiple target models. It can be observed from the figure that the standard deviation is large for a small number of queries, indicating that the target models have highly inconsistent predictions on the data. This may imply the diverse preferences among different models of data selection. Under this scenario, our method DIAM outperforms the traditional active and passive learning methods. This result demonstrates the effectiveness of DIAM and the importance of designing an active query method in this practical setting. The uncertainty-based methods, i.e., Entropy, Least confidence and Margin, achieve comparable performances with random sampling. These results align with our expectation, as traditional AL methods are usually model-dependent and the data queried by one model may not be useful for training other models. The Coreset method is less stable than Random. A possible reason is that Coreset selects data based on the extracted features of deep models, which will be optimized along with the training procedures. Therefore, it may also suffer from the model dependence problem.

The statistical significance of the performance comparisons between our DIAM method and the other compared methods is demonstrated in Table 2. Specifically, we conduct paired t-tests at a 0.05 significance level of the performances of multiple target models after each iteration and report the Win/Tie/Loss results of the tests. Here, 'Win' (resp. 'Loss') means that the mean accuracy of our method is statistically significantly better (resp. worse) than that of the rival with statistical significance and 'Tie' means that no method is statistically significantly better. The results in the table show that our DIAM method consistently outperforms the other

Table 2: Win/Tie/Loss (W./T./L.) results of DIAM versus the other methods with varied numbers of queried batch based on paired t-tests at 0.05 significance level. The comparisons are based on the performances of 12 target models after each iteration.

Algorithms	Number of queried batch ($1,500$ examples per batch)									W./T./L.
	1	2	3	4	5	6	7	8	9	
MNIST										
Entropy	Tie	Tie	Win	Win	Win	Win	Win	Win	Win	7/2/0
Margin	Tie	Tie	Win	Win	Win	Tie	Win	Win	Tie	5/4/0
Least conf.	Tie	Win	Win	Win	Win	Tie	Win	Win	Tie	6/3/0
CAL	Win	Win	Win	Win	Win	Tie	Win	Win	Tie	7/2/0
Random	Win	Tie	Win	Win	Win	Win	Win	Win	Tie	7/2/0
Coreset	Win	Win	Win	Win	Win	Win	Win	Tie	Win	8/1/0
W./T./L.	3/3/0	3/3/0	6/0/0	6/0/0	6/0/0	3/3/0	6/0/0	5/1/0	2/4/0	40/14/0
Fashion-MNIST										
Entropy	Tie	Tie	Win	Tie	Tie	Win	Tie	Win	Win	4/5/0
Margin	Tie	Tie	Win	Win	Win	Win	Win	Win	Tie	6/3/0
Least conf.	Tie	Tie	Win	Win	Tie	Win	Tie	Win	Win	5/4/0
CAL	Tie	Tie	Tie	Tie	Tie	Win	Tie	Win	Win	3/6/0
Random	Tie	Tie	Win	Win	Tie	Win	Tie	Win	Win	5/4/0
Coreset	Win	Win	Win	Win	Win	Win	Tie	Tie	Tie	6/3/0
W./T./L.	1/5/0	1/5/0	5/1/0	4/2/0	2/4/0	6/0/0	1/5/0	5/1/0	4/2/0	29/25/0
Kuzushiji-MNIST										
Entropy	Win	Tie	Tie	Tie	Win	Win	Win	Win	Win	6/3/0
Margin	Tie	Win	Tie	Win	Win	Win	Win	Win	Win	7/2/0
Least conf.	Win	Tie	Win	Win	Win	Win	Win	Win	Win	8/1/0
CAL	Tie	Win	Tie	Win	Win	Win	Win	Win	Win	7/2/0
Random	Tie	Win	Win	Win	Win	Win	Win	Win	Win	8/1/0
Coreset	Win	Win	Win	Win	Win	Win	Win	Win	Win	9/0/0
W./T./L.	3/3/0	4/2/0	3/3/0	5/1/0	6/0/0	6/0/0	6/0/0	6/0/0	6/0/0	45/9/0
EMNIST-digits										
Entropy	Tie	Tie	Tie	Tie	Win	Win	Win	Win	Tie	4/5/0
Margin	Tie	Tie	Tie	Tie	Win	Win	Win	Win	Tie	4/5/0
Least conf.	Win	Tie	Tie	Win	Win	Win	Win	Win	Tie	6/3/0
CAL	Tie	Tie	Win	Tie	Tie	Win	Win	Win	Tie	4/5/0
Random	Tie	Tie	Win	Win	Win	Win	Win	Win	Tie	6/3/0
Coreset	Tie	Tie	Tie	Win	Tie	Win	Win	Win	Tie	4/5/0
W./T./L.	1/5/0	0/6/0	2/4/0	3/3/0	4/2/0	6/0/0	6/0/0	6/0/0	0/6/0	28/26/0
EMNIST-letters										
Entropy	Win	Tie	Tie	Win	Win	Win	Win	Win	Win	7/2/0
Margin	Tie	Tie	Tie	Tie	Win	Tie	Tie	Tie	Win	2/7/0
Least conf.	Win	Tie	Tie	Win	Win	Win	Tie	Win	Win	6/3/0
CAL	Tie	Tie	Tie	Tie	Win	Tie	Tie	Tie	Win	2/7/0
Random	Win	Tie	Tie	Tie	Win	Tie	Tie	Tie	Win	3/6/0
Coreset	Win	Win	Tie	Win	Win	Win	Win	Win	Win	8/1/0
W./T./L.	4/2/0	1/5/0	0/6/0	3/3/0	6/0/0	3/3/0	2/4/0	3/3/0	6/0/0	28/26/0
CIFAR-10										
Entropy	Tie	Tie	Tie	Tie	Tie	Tie	Tie	Win	Tie	1/8/0
Margin	Tie	Tie	Win	Win	Tie	Win	Tie	Win	Tie	4/5/0
Least conf.	Tie	Tie	Tie	Win	Win	Tie	Tie	Win	Win	4/5/0
CAL	Tie	Tie	Tie	Win	Win	Tie	Win	Win	Win	5/4/0
Random	Tie	Tie	Tie	Win	Win	Tie	Win	Win	Tie	4/5/0
Coreset	Win	Win	Win	Win	Win	Win	Win	Win	Win	9/0/0
W./T./L.	1/5/0	1/5/0	2/4/0	5/1/0	4/2/0	2/4/0	3/3/0	6/0/0	3/3/0	27/27/0
CIFAR-100										
Entropy	Tie	Win	Win	Win	Tie	Win	Win	Win	Win	7/2/0
Margin	Tie	Win	Win	Tie	Tie	Tie	Win	Win	Win	5/4/0
Least conf.	Tie	Tie	Tie	Win	Win	Win	Win	Win	Win	6/3/0
CAL	Win	Tie	Tie	Tie	Tie	Win	Win	Win	Win	5/4/0
Random	Tie	Tie	Tie	Tie	Win	Tie	Win	Win	Win	4/5/0
Coreset	Win	Win	Win	Win	Win	Win	Win	Win	Win	9/0/0
W./T./L.	2/4/0	3/3/0	3/3/0	3/3/0	3/3/0	4/2/0	6/0/0	6/0/0	6/0/0	36/18/0

methods significantly. There are no cases in which it is significantly worse than any baseline, and it achieves the best result on all benchmarks. Taken together, these findings indicate that DIAM selects informative query points that benefit all target models, leading to higher mean accuracy and more uniform gains.

6.3 The Best and Worst Performances of Multiple Models

To examine whether the compared methods can improve all target models evenly, which are usually equally important in real-world applications. We report the best and worst performances of multiple target models in Table 3a. Here we report the mean and standard deviation values of the

Table 3: The mean and standard deviation values of the learning curves of the best and worst performances of multiple target models, and the performances of our DIAM method with different values of the trade-off parameter β (mean accuracy $\pm$ mean standard deviation). The best performance is highlighted in boldface.

Methods	Datasets						
	MNIST	Fashion-MNIST	Kuzushiji-MNIST	EMNIST-digital	EMNIST-letter	CIFAR-10	CIFAR-100
Best							
DIAM	$\mathbf{98.44 \pm 0.69}$	$\mathbf{87.61 \pm 2.14}$	$\mathbf{90.49 \pm 5.92}$	$\mathbf{98.44 \pm 0.71}$	$\mathbf{87.01 \pm 5.07}$	$\mathbf{69.59 \pm 8.53}$	$\mathbf{27.73 \pm 9.24}$
Entropy	98.13 ± 0.66	86.97 ± 1.99	89.17 ± 5.26	98.10 ± 0.62	86.60 ± 5.06	69.00 ± 8.24	26.91 ± 8.75
Margin	98.20 ± 0.62	87.00 ± 2.08	89.09 ± 5.22	98.13 ± 0.61	86.42 ± 4.96	68.66 ± 8.06	26.88 ± 8.84
Least conf.	98.20 ± 0.64	87.09 ± 2.06	88.86 ± 5.16	98.14 ± 0.66	86.55 ± 4.96	68.88 ± 8.20	27.32 ± 8.89
CAL	98.22 ± 0.65	87.16 ± 2.12	89.17 ± 5.30	98.17 ± 0.65	86.66 ± 4.97	68.91 ± 8.27	27.51 ± 9.04
Random	98.22 ± 0.67	87.09 ± 1.99	89.18 ± 5.35	98.13 ± 0.65	86.84 ± 4.91	68.52 ± 8.18	27.36 ± 8.72
Coreset	98.07 ± 0.64	86.86 ± 2.20	89.58 ± 5.58	98.26 ± 0.67	85.66 ± 5.34	66.81 ± 9.59	22.60 ± 7.15
Worst							
DIAM	95.17 ± 3.74	$\mathbf{82.56 \pm 2.99}$	$\mathbf{76.96 \pm 11.73}$	94.66 ± 3.25	64.08 ± 23.61	61.92 ± 7.86	$\mathbf{21.37 \pm 6.76}$
Entropy	94.97 ± 3.54	82.01 ± 2.39	75.36 ± 11.41	$\mathbf{94.87 \pm 3.30}$	70.12 ± 13.42	61.84 ± 7.95	19.88 ± 6.40
Margin	95.02 ± 3.72	81.54 ± 2.57	75.79 ± 10.99	94.22 ± 3.67	$\mathbf{70.69 \pm 13.85}$	61.74 ± 8.10	19.77 ± 5.77
Least conf.	94.69 ± 3.58	82.18 ± 2.81	75.05 ± 11.37	94.78 ± 3.57	69.91 ± 14.51	$\mathbf{62.24 \pm 7.74}$	19.92 ± 6.49
CAL	94.83 ± 3.64	82.16 ± 3.08	74.94 ± 11.09	94.36 ± 3.51	69.86 ± 14.09	62.10 ± 7.66	20.04 ± 5.91
Random	$\mathbf{95.18 \pm 3.65}$	81.91 ± 2.59	75.60 ± 10.75	94.08 ± 3.67	70.63 ± 14.61	61.40 ± 8.15	19.88 ± 6.14
Coreset	93.53 ± 3.57	81.43 ± 3.05	73.94 ± 11.83	92.02 ± 2.96	68.89 ± 13.03	58.62 ± 8.61	15.80 ± 4.46

(a) The best and worst performances of multiple target models.

Parameter	Datasets						
	MNIST	Fashion-MNIST	Kuzushiji-MNIST	EMNIST-digits	EMNIST-letters	CIFAR-10	CIFAR-100
$\beta = 0.1$	$\mathbf{97.29 \pm 1.95}$	85.47 ± 2.52	$\mathbf{84.74 \pm 8.62}$	$\mathbf{97.16 \pm 1.87}$	80.16 ± 8.86	65.51 ± 8.00	$\mathbf{24.51 \pm 8.00}$
$\beta = 1.0$	97.26 ± 1.97	85.38 ± 2.54	84.69 ± 8.77	97.10 ± 1.93	80.47 ± 8.36	$\mathbf{65.68 \pm 8.02}$	24.14 ± 7.71
$\beta = 10.0$	97.24 ± 1.97	$\mathbf{85.52 \pm 2.51}$	84.47 ± 8.75	97.07 ± 1.93	$\mathbf{80.56 \pm 8.73}$	65.52 ± 7.96	24.17 ± 7.68

(b) Parameter sensitivity of DIAM method.

learning curves, rather than plotting them. The best performance of each case is highlighted in boldface.

It can be observed that DIAM usually achieves the best results in both cases. Even when it is not in the first place, its performance is the second best and closely competitive with the best-performing method. These results demonstrate our method's ability to impartially improve the target models with different architectures, which we believe is essential for applications involving multiple target models.

6.4 Study on the Parameter Sensitivity of DIAM Method

To study the sensitivity of the trade-off parameter β to our method, we evaluate the performances of DIAM with different values of β. Specifically, we set β to each of $\{-0.1, -1.0, -10.0\}$ and report the mean and standard deviation of the learning curves in Table 3b. For clarity of presentation, we omit the negative sign in the table; however, it should be noted that β is always assigned a negative value. The best performance of each case is highlighted in boldface.

The results show that our method is less sensitive to this parameter. For datasets with a larger number of classes, e.g., EMNIST-letters, a bigger value of β is preferable. Otherwise, using $\beta = 0.1$ yields good performance. A possible explanation for these phenomena is the degree of class imbalance caused by active querying. When $\beta = 0.1$, DIAM places more emphasis on informativeness in data selection, which may lead to class imbalance, especially in tasks with larger label spaces. Consequently, it is crucial to promote diversity in data selection in such situations.

6.5 Study on Different Numbers of Target Models

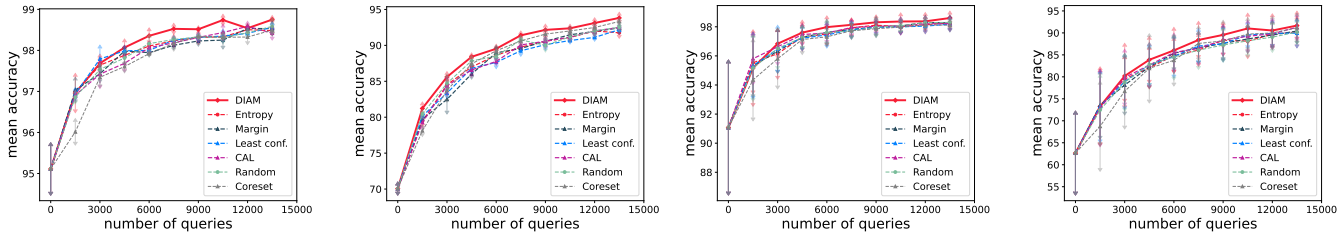
We further examine the performances of compared methods with different numbers of target models. We empirically take the first $2, 4, 6, 8$ specifications from the model configuration list in Sec. 6.1 as the target models set. We conduct this experiment on MNIST and Kuzushiji-MNIST datasets.

We report in Fig. 2 the trend of mean accuracy of multiple target models as the number of queries increases. The error bars represent the standard deviation of the performances of multiple target models. The results show that our method consistently outperforms the other compared methods under the settings of different numbers of target models, which demonstrates its robustness to the number of models.

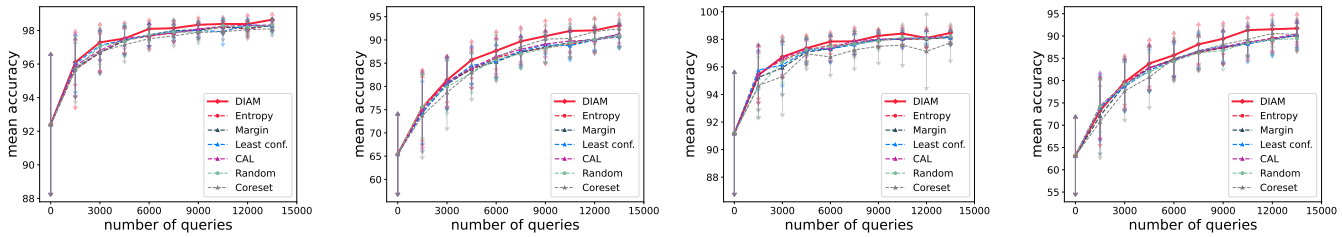
6.6 Study on Deep Regression Task

In this section, we validate the effectiveness of the proposed DIAM implementations for regression tasks. We continue to employ the architectures introduced in Sec. 6.1 as our target models since the convolutional neural network is commonly used in deep regression [40], [58]. For the empirical settings, we replace the loss function with the Mean Square Error (MSE) for regression. The data batch size in model training is reduced to 64. The following metrics are employed to evaluate the test performances: Mean Absolute Error (MAE) and MSE. We compare the following query strategies in our experiments.

- DIAM-svd: Our proposed method in this paper, which uses SVD to identify the data located in the joint disagreement regions, i.e., Algorithm 3.
- DIAM-leverage: Our proposed method in this paper, which uses leverage score to identify the data located in the joint disagreement regions, i.e., Algorithm 4.
- Coreset [53]: This strategy queries the most representative data.



(a) MNIST (2 models) (b) Kuzushiji-MNIST (2 models) (c) MNIST (4 models) (d) Kuzushiji-MNIST (4 models)



(e) MNIST (6 models) (f) Kuzushiji-MNIST (6 models) (g) MNIST (8 models) (h) Kuzushiji-MNIST (8 models)

Figure 2: Learning curves of the compared methods with different numbers of target models (2, 4, 6, 8 models). The error bars indicate the standard deviation of the performances of target models.

- BAIT [2]: This strategy queries batches of samples by optimizing a bound on the maximum likelihood estimators error in terms of the Fisher information.
- BADGE [3]: This strategy queries data points by considering both predictive uncertainty and sample diversity.
- Random: This strategy queries data uniformly from the unlabeled set.

We implement BADGE and BAIT algorithms using the BMDAL toolbox [32]. Note that, BADGE, BAIT and Coreset methods use the features extracted by the supernet in OFA to evaluate and select data points.

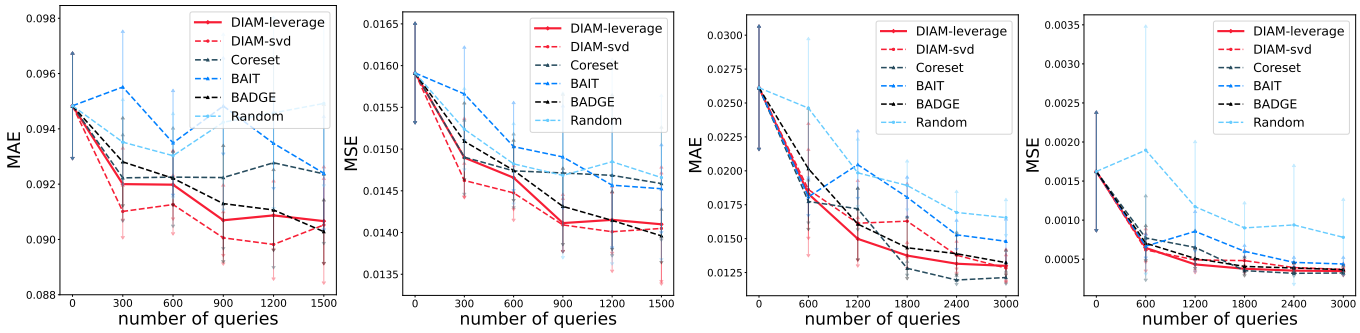
Facial landmark detection is another representative machine learning system that needs to be deployed on diverse devices. We employ CelebA [45], and LFW and NET facial landmark detection datasets [58] to validate the effectiveness of the proposed method, which are commonly used deep regression datasets [40]. The LFW and NET facial landmark detection datasets contain 13466 training instances, and 1521 test images. We follow the data partition provided in [58] to ensure fair comparisons. The initially labeled set contains 500 instances randomly sampled from the training set, and 300 images will be queried from the remaining training data at each iteration. The CelebA dataset comprises 162770 training instances and 19962 test instances. Given the relatively large size of this dataset, we have increased the query batch size to 600. All other settings remain the same.

The learning curves of the compared methods with 12 target models are presented in Fig. 3. We can observe that the proposed two implementations of DIAM significantly outperform the other compared methods. These results demonstrate the effectiveness of the proposed selection criterion. DIAM-svd and DIAM-leverage achieve comparable performances, which aligns closely with our expectation, since the problems they address share similar definitions. Coreset is better than Random, it also has a smaller performance variance. This result accords with the objective of Coreset, as it aims to cover the data distribution with the least data. BAIT demonstrates relatively lower effectiveness, suggesting that further refinement and redesign are necessary to better adapt it to the multi-model setting. In contrast, BADGE generally performs well, likely because it explicitly considers for both informativeness and diversity. This finding highlights the importance of incorporating both criteria in active learning for multiple models, as evidenced by the strong performance of DIAM in classification tasks.

We further examine the performances of the compared methods with different numbers of target models on the CelebA dataset. Specifically, we follow the strategy in Sec 6.5 to take the first 8 and 4 models defined in Sec. 6.1 as target models. we plot the learning curves of the compared methods in Fig. 4. The observations from the learning curve comparisons are in line with the results obtained from the 12 model settings, which indicates that our method is robust to the number of models in the regression task.

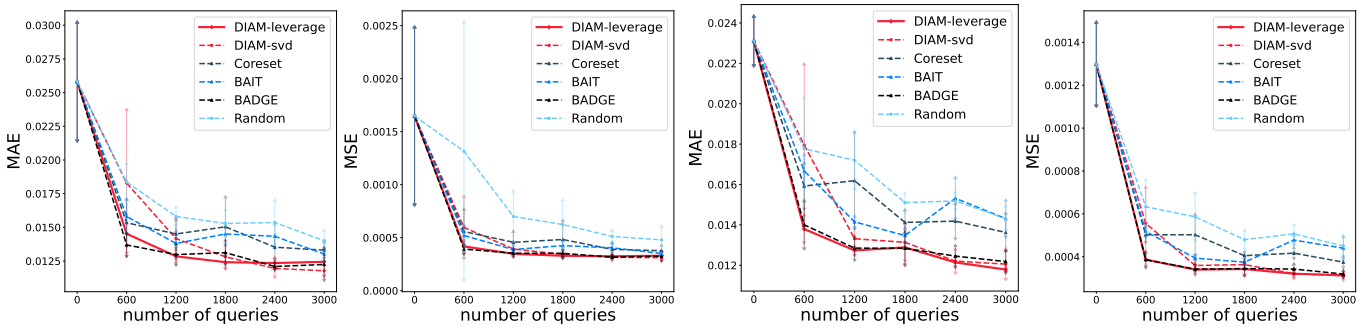
7 Conclusion

In this paper, we propose to study active learning in a novel setting, where the task is to select and label the most useful examples that are beneficial to the performances of multiple target models. We analyze the query complexity of both active and passive learning, demonstrating the potential of AL to achieve better query complexity than random sampling. Based on this insight, we further propose an active selection criterion DIAM to identify and select the data located in the joint disagreement regions of different target models. We provide efficient implementations of DIAM to extend its applications to deep classification and regression problems. Empirical experiments conducted on two representative tasks, OCR and FLD, which are often required to support diverse devices, show the effectiveness of our proposed



(a) The LFW and NET facial landmark detection datasets (b) CelebA dataset

Figure 3: The learning curves of the compared methods with the mean performances of 12 target models for the regression task. The metrics include MSE, MAE. The error bars indicate the standard deviation of the performances of target models.



(a) 8 target models (b) 4 target models

Figure 4: The learning curves of the compared methods with the mean performances of different numbers of target models for the regression task on the CelebA dataset. The metrics include MSE, MAE. The error bars indicate the standard deviation of the performances of target models.

method. As a future direction, we will explore more intricate learning tasks, such as object detection and semantic segmentation, and develop effective query strategies for multiple target models.

Acknowledgment

S.-J. Huang and Y.-P. Tang were supported in part by the National Natural Science Foundation of China grants U2441285 and 62222605. Y. Li was supported in part by the Singapore Ministry of Education AcRF Tier 2 grant MOE-T2EP20122-0001.

References

[1] Alnur Ali, Rich Caruana, and Ashish Kapoor. Active learning with model selection. In *AAAI Conference on Artificial Intelligence*, pages 1673–1679, 2014.

[2] Jordan Ash, Surbhi Goel, Akshay Krishnamurthy, and Sham Kakade. Gone fishing: Neural active learning with fisher embeddings. *Advances in Neural Information Processing Systems*, 34:8927–8939, 2021.

[3] Jordan T Ash, Chicheng Zhang, Akshay Krishnamurthy, John Langford, and Alekh Agarwal. Deep batch active learning by diverse, uncertain gradient lower bounds. In *International Conference on Learning Representations*, pages 1–13, 2020.

[4] Jimmy Ba and Rich Caruana. Do deep nets really need to be deep? In Zoubin Ghahramani, Max Welling, Corinna Cortes, Neil D. Lawrence, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, pages 2654–2662, 2014.

[5] Maria-Florina Balcan, Alina Beygelzimer, and John Langford. Agnostic active learning. *Journal of Computer and System Sciences*, 75(1):78–89, 2009.

[6] Maria-Florina Balcan, Steve Hanneke, and Jennifer Wortman Vaughan. The true sample complexity of active learning. *Machine Learning*, 80(2):111–139, 2010.

[7] Alina Beygelzimer, Daniel J. Hsu, John Langford, and Chicheng Zhang. Search improves label for active learning. In *Advances in Neural Information Processing Systems*, pages 3342–3350, 2016.

[8] Alina Beygelzimer, Daniel J. Hsu, John Langford, and Tong Zhang. Agnostic active learning without constraints. In *Advances in Neural Information Processing Systems*, pages 199–207, 2010.

[9] Han Cai, Chuang Gan, Tianzhe Wang, Zhekai Zhang, and Song Han. Once-for-all: Train one network and specialize it for efficient deployment. In *International Conference on Learning Representations*, 2020.

[10] Han Cai, Ligeng Zhu, and Song Han. Proxylessnas: Direct neural architecture search on target task and hardware. In *International Conference on Learning Representations*. OpenReview.net, 2019.

[11] Xiaofeng Cao and Ivor W Tsang. Shattering distribution for active learning. *IEEE Transactions on Neural Networks and Learning Systems*, 33(1):215–228, 2022.

[12] Yu Cheng, Duo Wang, Pan Zhou, and Tao Zhang. A survey of model compression and acceleration for deep neural networks. *CoRR*, abs/1710.09282, 2017.

[13] Tarin Clanuwat, Mikel Bober-Irizar, Asanobu Kitamoto, Alex Lamb, Kazuaki Yamamoto, and David Ha. Deep learning for classical japanese literature. arXiv cs.CV/1812.01718, 2018.

[14] Gregory Cohen, Saeed Afshar, Jonathan Tapson, and André van Schaik. EMNIST: an extension of MNIST to handwritten letters. arXiv cs.CV//1702.05373, 2017.

[15] David Cohn, Les Atlas, and Richard Ladner. Improving generalization with active learning. *Machine Learning*, 15(2):201–221, 1994.

[16] Petros Drineas, Malik Magdon-Ismail, Michael W Mahoney, and David P Woodruff. Fast approximation of matrix coherence

and statistical leverage. *The Journal of Machine Learning Research*, 13(1):3475–3506, 2012.

[17] Bo Du, Zengmao Wang, Lefei Zhang, Liangpei Zhang, Wei Liu, Jialie Shen, and Dacheng Tao. Exploring representativeness and informativeness for active learning. *IEEE Transactions on Cybernetics*, 47(1):14–26, 2015.

[18] Thomas Elsken, Jan Hendrik Metzen, and Frank Hutter. Neural architecture search: A survey. *The Journal of Machine Learning Research*, 20(1):1997–2017, 2019.

[19] Yarin Gal, Riashat Islam, and Zoubin Ghahramani. Deep bayesian active learning with image data. In *International Conference on Machine Learning*, pages 1183–1192. PMLR, 2017.

[20] Yonatan Geifman and Ran El-Yaniv. Deep active learning with a neural architecture search. In *Advances in Neural Information Processing Systems*, pages 5976–5986, 2019.

[21] Ran Gilad-Bachrach, Amir Navot, and Naftali Tishby. Query by committee made real. In *Advances in Neural Information Processing Systems*, pages 443–450, 2005.

[22] Jianping Gou, Baosheng Yu, Stephen J Maybank, and Dacheng Tao. Knowledge distillation: A survey. *International Journal of Computer Vision*, 129:1789–1819, 2021.

[23] Martin Grötschel, László Lovász, and Alexander Schrijver. *Geometric algorithms and combinatorial optimization*, volume 2. Springer Science & Business Media, 2012.

[24] Song Han, Huizi Mao, and William J Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. In *International Conference on Learning Representations*, 2016.

[25] Song Han, Jeff Pool, John Tran, and William Dally. Learning both weights and connections for efficient neural network. *Advances in neural information processing systems*, 28, 2015.

[26] Steve Hanneke. A bound on the label complexity of agnostic active learning. In *International Conference on Machine Learning*, pages 353–360, 2007.

[27] Steve Hanneke. Activized learning: Transforming passive to active with improved label complexity. *The Journal of Machine Learning Research*, 13(1):1469–1587, 2012.

[28] Steve Hanneke. Theory of active learning. *Foundations and Trends in Machine Learning*, 7(2-3), 2014.

[29] Yihui He, Ji Lin, Zhijian Liu, Hanrui Wang, Li-Jia Li, and Song Han. AMC: automl for model compression and acceleration on mobile devices. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *European Conference on Computer Vision*, volume 11211 of *Lecture Notes in Computer Science*, pages 815–832. Springer, 2018.

[30] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[31] Steven C. H. Hoi, Rong Jin, Jianke Zhu, and Michael R. Lyu. Semi-supervised SVM batch mode active learning for image retrieval. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[32] David Holzmüller, Viktor Zaverkin, Johannes Kästner, and Ingo Steinwart. A framework and benchmark for deep batch active learning for regression. *Journal of Machine Learning Research*, 24(164):1–81, 2023.

[33] Sheng-Jun Huang, Rong Jin, and Zhi-Hua Zhou. Active learning by querying informative and representative examples. *IEEE Transactions on pattern analysis and machine intelligence*, 36(10):1936–1949, 2014.

[34] Sheng-Jun Huang and Zhi-Hua Zhou. Active query driven by uncertainty and diversity for incremental multi-label learning. In *International Conference on Data Mining*, pages 1079–1084, 2013.

[35] Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew Howard, Hartwig Adam, and Dmitry Kalenichenko. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In *IEEE conference on computer vision and pattern recognition*, pages 2704–2713, 2018.

[36] Andreas Kirsch, Joost van Amersfoort, and Yarin Gal. Batchbald: Efficient and diverse batch acquisition for deep bayesian active learning. In *Advances in Neural Information Processing Systems*, pages 7026–7037, 2019.

[37] Ksenia Konyushkova, Raphael Sznitman, and Pascal Fua. Learning active learning from data. In *Advances in Neural Information Processing Systems*, pages 4225–4235, 2017.

[38] Raghuraman Krishnamoorthi. Quantizing deep convolutional networks for efficient inference: A whitepaper. *arXiv preprint arXiv:1806.08342*, 2018.

[39] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.

[40] Stéphane Lathuilière, Pablo Mesejo, Xavier Alameda-Pineda, and Radu Horaud. A comprehensive analysis of deep regression. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(9):2065–2081, 2019.

[41] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[42] Yann LeCun, John Denker, and Sara Solla. Optimal brain damage. *Advances in neural information processing systems*, 2, 1989.

[43] David D Lew is and Jason Catlett. Heterogeneous uncertainty sampling for supervised learning. In *Machine Learning: Proceedings of the 11th International Conference*, pages 148–156. Elsevier, 1994.

[44] Changsheng Li, Handong Ma, Zhao Kang, Ye Yuan, Xiao-Yu Zhang, and Guoren Wang. On deep unsupervised active learning. In *International Joint Conferences on Artificial Intelligence*, pages 2626–2632, 2020.

[45] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *2015 IEEE International Conference on Computer Vision*, 2015.

[46] David Lowell, Zachary C. Lipton, and Byron C. Wallace. Practical obstacles to deploying active learning. In *EMNLP-IJCNLP*, pages 21–30, 2019.

[47] Gaurav Menghani. Efficient deep learning: A survey on making deep learning models smaller, faster, and better. *ACM Computing Surveys*, 2021.

[48] Kunkun Pang, Mingzhi Dong, Yang Wu, and Timothy Hospedales. Meta-learning transferable active learning policies by deep reinforcement learning. *arXiv preprint arXiv:1806.04798*, 2018.

[49] Davi Pereira-Santos, Ricardo Bastos Cavalcante Prudêncio, and André CPLF de Carvalho. Empirical investigation of active learning strategies. *Neurocomputing*, 326:15–27, 2019.

[50] Pengzhen Ren, Yun Xiao, Xiaojun Chang, Po-Yao Huang, Zhihui Li, Xiaojiang Chen, and Xin Wang. A survey of deep active learning. *arXiv preprint arXiv:2009.00236*, 2020.

[51] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. In Yoshua Bengio and Yann LeCun, editors, *International Conference on Learning Representations*, 2015.

[52] Tobias Scheffer, Christian Decomain, and Stefan Wrobel. Active hidden markov models for information extraction. In *International Symposium on Intelligent Data Analysis*, pages 309–318. Springer, 2001.

[53] Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. In *International Conference on Learning Representations*, 2018.

[54] Burr Settles. Active learning literature survey. Technical report, University of Wisconsin-Madison, 2009.

[55] Burr Settles and Mark Craven. An analysis of active learning strategies for sequence labeling tasks. In *Conference on Empirical Methods in Natural Language Processing*, pages 1070–1079, 2008.

[56] Changjian Shui, Fan Zhou, Christian Gagné, and Boyu Wang. Deep active learning: Unified and principled method for query and training. In *International Conference on Artificial Intelligence and Statistics*, pages 1308–1318, 2020.

[57] Samarth Sinha, Sayna Ebrahimi, and Trevor Darrell. Variational adversarial active learning. In *International Conference on Computer Vision*, pages 5972–5981, 2019.

[58] Yi Sun, Xiaogang Wang, and Xiaoou Tang. Deep convolutional network cascade for facial point detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3476–3483, 2013.

[59] Ying-Peng Tang and Sheng-Jun Huang. Self-paced active learning: Query the right thing at the right time. In *AAAI Conference on Artificial Intelligence*, volume 33, pages 5117–5124, 2019.

[60] Ying-Peng Tang and Sheng-Jun Huang. Dual active learning for both model and data selection. In *International Joint Conference on Artificial Intelligence*, pages 3052–3058, 2021.

[61] Ying-Peng Tang and Sheng-Jun Huang. Active learning for multiple target models. In *Advances in Neural Information Processing Systems*, volume 35, pages 38424–38435, 2022.

[62] Ying-Peng Tang, Xiu-Shen Wei, Borui Zhao, and Sheng-Jun Huang. Qbox: Partial transfer learning with active querying for object detection. *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

[63] Alexander B Tsybakov. Optimal aggregation of classifiers in statistical learning. *The Annals of Statistics*, 32(1):135–166, 2004.

[64] Thuy-Trang Vu, Ming Liu, Dinh Phung, and Gholamreza Haffari. Learning how to active learn by dreaming. In *Annual Meeting of the Association for Computational Linguistics*, pages 4091–4101, 2019.
[65] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. arXiv cs.LG/1708.07747, 2017.
[66] Yifan Yan and Sheng-Jun Huang. Cost-effective active learning for hierarchical multi-label classification. In *International Joint Conferences on Artificial Intelligence*, pages 2962–2968, 2018.
[67] Yi Yang, Zhigang Ma, Feiping Nie, Xiaojun Chang, and Alexander G Hauptmann. Multi-class active learning by uncertainty sampling with diversity maximization. *International Journal of Computer Vision*, 113(2):113–127, 2015.
[68] Junho Yim, Donggyu Joo, Ji-Hoon Bae, and Junmo Kim. A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 7130–7138. IEEE Computer Society, 2017.
[69] Bayable Teshome Zegeye and Begüm Demir. A novel active learning technique for multi-label remote sensing image scene classification. In *Image and Signal Processing for Remote Sensing*, volume 10789, page 107890B. International Society for Optics and Photonics, 2018.
[70] Xueying Zhan, Huan Liu, Qing Li, and Antoni B. Chan. A comparative survey: Benchmarking for pool-based active learning. In *International Joint Conferences on Artificial Intelligence*, pages 4679–4686, 2021.
[71] Yilun Zhou, Adithya Renduchintala, Xian Li, Sida Wang, Yashar Mehdad, and Asish Ghoshal. Towards understanding the behaviors of optimal deep active learning algorithms. In *International Conference on Artificial Intelligence and Statistics*, pages 1486–1494. PMLR, 2021.

Sheng-Jun Huang received the BSc and PhD degrees in computer science from Nanjing University, China, in 2008 and 2014, respectively. He is now a professor with the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics. His main research interests include machine learning and data mining. He has been selected to the Young Elite Scientists Sponsorship Program by CAST in 2016, and won the China Computer Federation Outstanding Doctoral Dissertation Award in 2015, the KDD Best Poster Award at the in 2012, and the Microsoft Fellowship Award in 2011. He is a junior associate editor of the Frontiers of Computer Science.

Yi Li received the B.Eng. degree in computer science and engineering from Shanghai Jiaotong University in 2008 and the Ph.D. degree in computer science and engineering from the University of Michigan—Ann Arbor in 2013. He is currently an Associate Professor with the School of Physical and Mathematical Sciences and the College of Computing and Data Science, Nanyang Technological University. His research interests lie in theoretical computer science, mostly in algorithms for massive data, including compressive sensing, data stream algorithms, randomized numerical linear algebra, and dimensionality reduction.

Ying-Peng Tang received the BSc, Master and Ph.D. degrees from the Nanjing University of Aeronautics and Astronautics, China, in 2017, 2020, 2024, respectively. He is currently a research fellow with the College of Computing and Data Science, Nanyang Technological University, Singapore. His current research interests include active learning and machine learning. He has been awarded for China National Scholarship during both Master and Ph.D. phases in 2019 and 2022, respectively, and the Excellent Master thesis in Jiangsu Province in 2021.