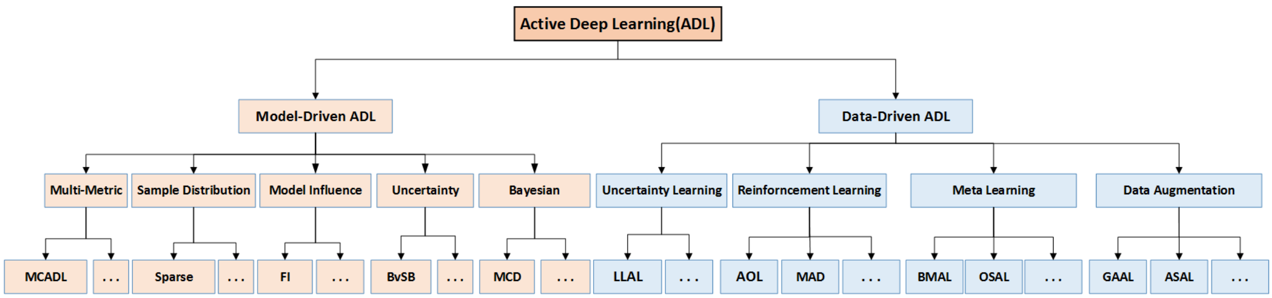
Graphical Abstract

From Model-driven to Data-driven: A Survey on Active Deep Learning

Peng Liu,Guojin He,Lei Zhao



Highlights

From Model-driven to Data-driven: A Survey on Active Deep Learning

Peng Liu,Guojin He,Lei Zhao

- We category ADL into two large categories by whether their selectors are model-driven or data-driven
- The advantages and disadvantages between data-driven ADL and model-driven ADL are thoroughly analyzed.
- We pointed out that, the selector in ADL also is experiencing the stage from model-driven to data-driven.

From Model-driven to Data-driven: A Survey on Active Deep Learning

Peng Liu[a,c,*], Guojin He[a,c,**] and Lei Zhao[b]

[a]Aerospace Information Research Institute, Chinese Academy of Sciences, Beijing 100094, China
[b]School of Information Science and Technology, Beijing Forestry University, Beijing 100083, China
[c]College of Resources and Environment, University of Chinese Academy of Sciences, Beijing 100049, China

ARTICLE INFO

Keywords:
Active Deep learning
Data-dirven
Model-dirven
Labelling Sample

ABSTRACT

Which samples should be labelled in a large data set is one of the most important problems for training of deep learning. So far, a variety of active sample selection strategies related to deep learning have been proposed in many literatures. We defined them as Active Deep Learning (ADL) only if their predictor is deep model, where the basic learner is called as predictor and the labeling schemes is called selector. In this survey, three fundamental factors in selector designation were summarized. We category ADL into model-driven ADL and data-driven ADL, by whether its selector is model-driven or data-driven. The different characteristics of the two major type of ADL were addressed in in detail respectively. Furthermore, different sub-classes of data-driven and model-driven ADL are also summarized and discussed emphatically. The advantages and disadvantages between data-driven ADL and model-driven ADL are thoroughly analyzed. We pointed out that, with the development of deep learning, the selector in ADL also is experiencing the stage from model-driven to data-driven. Finally, we make discussion on ADL about its uncertainty, explanatory, foundations of cognitive science etc. and survey on the trend of ADL from model-driven to data-driven.

1. Introduction

Recently, deep learning have made incredible progress in many fields. The successes of deep learning mainly attributed to three points: the advanced deep network architectures such as ResNet [1], powerful computing devices such as GPU, and large data set with labels such as ImageNet [2]. Among the three important points, deep network architectures and computing devices were developed fatly and smoothly, but it is still very hard to label large quantity data for training sets. High efficiency labels of large scale data set are especially important [3] for most of deep learning architectures. Therefore, how to label data is becoming one of the bottle neck problems for deep learning and it is attracting more and more attention of researchers in this community.

For many machine learning tasks such as image classification, object tracking, image segmentation and change detection etc., labeling samples means very heavy workloads and high costs. Specifically, in different fields such as remote sensing, medical images and natural language processing etc, we face different difficulties when trying to provide the labels for the training set. In remote sensing applications, there are often very few small targets in a very large context so that huge workloads have to be supported for finding them or labeling them [4]. For medical images, the too high requirements for medical knowledge make the labeling of medical images very hard to implement. For natural language processing, the most important difficulty in labelling is the language ambiguity since the language will show different meaning in different contexts. All of these difficulties make most of the labeling process inevitably resort to a manual operation or human-computer interaction. Generally speaking, the natural images from internet such as ImageNet [2] are relatively easy to label but their too large quantity is still a serous problem. Over all, the labeling process is not a trivial by different complex factors although the sample's labeling is the important start point to realize currently smart AI models

It is necessary to consider which samples should be labeled since labeling samples would be costly. For some applications such as image segmentation, manually assigning a label to each pixel of an image is so laborious that it almost can not be implemented in big data. Furthermore, the informative values of different labeled samples are often very different to the learners. In many cases, some unimportant labeled samples cannot provide enough information for the training of deep architectures, so that they make an algorithm fail if the learner is trained with a randomly selected data set. How to effectively train deep neural networks by selectively labelling few important samples is a fundamental problem for current research on AI algorithms. Advances in deep learning techniques such as Active Deep Learning (ADL) or Human-Machine Cooperation (HMC) have shown promising progress in overcoming the key challenges outlined above. Active learning aims at building efficient training sets by iteratively improving the model performance through sampling. In another aspect, active learning is a effective way to improve the generalization ability of the classifier [5].

In machine learning community, active learning is one of fundamental studies on labelling high quality samples. It is a learning algorithm can interactively query a user (or some other information source) to label new data points with the desired outputs[6]. In statistics literature [7] , it is sometimes also called Optimal Experimental Design (OED). The

[⋆] This work was supported by in part by the NSFC under Grant 61731022, and Grant 41971397.
[*]Corresponding author
[**]Principal corresponding author
liupeng202303@aircas.ac.cn (P. Liu); hegj@aircas.ac.cn (G. He)
https://scholar.google.com/citations?user=5wOjyo4AAAAJ&hl=en/ (P. Liu)
ORCID(s): 0000-0003-3292-8551 (P. Liu)

information source for labels also called teacher or oracle [8]. Now, with the development of deep learning, the community of machine learning is **experiencing the stage** from model-driven to data-driven. At the same time, active deep learning also is experiencing the similar process from model-driven to data-driven. There are already some other surveys on active learning such as a survey of active learning in collaborative filtering recommender systems [9], a survey of active learning algorithms for supervised remote sensing image classification[10], a survey of deep active Learning [11], a survey of active learning for text classification using deep neural networks[12], and a survey on active learning and human-in-the-loop deep learning for medical image analysis [13]. In this survey, we take a new insight into the development of ADL in recent years. Different type of ADLs are categorized into model-driven and data-driven and will be summarized comprehensively. The future data-driven trend of ADL will also be discussed in detail.

The survey is organized as follows: In Section 2, we take a overview of ADL as its notation, core issue, and taxonomy. In Section 3, model-driven ADL methods are comprehensively discussed which consisted in ADL with uncertainty, ADL with representativeness, ADL with diversity and etc. In Section 4, data-driven ADL methods are discussed in detail, which consistsed in ADL with meta learning, ADL with reinforcement learning, ADL with data augmentation, and ADL with uncertainty learning etc. In Section 5, some of other learning problem related to ADL are simply discussed. In Section 6, we discuss some of the important open problems in ADL such as explainable for data-driven ADL, reproducible for ADL, and limitation of ADL etc. Finally, we concluded the survey in Section 7.

2. Overview

2.1. Notations

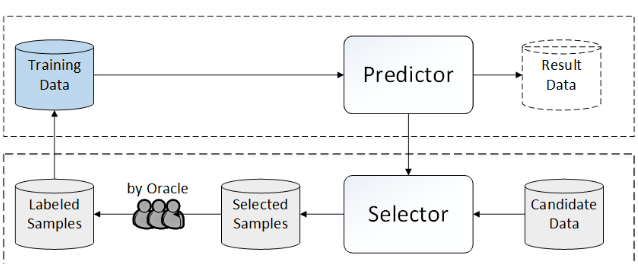


Figure 1: Active Learning.

We first define D^{tr} as training data set, where $D^{tr} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{I}$, such that $\mathbf{x}_i$ is the feature vector of the i-th sample and y_i is its label (i.e., class). The task of supervised learning is to search a mapping function $F_\theta : X \rightarrow Y$, where X is the input space, Y is the output space, and θ is the set of parameters that are trained by D^{tr}. The function F_θ is an element of the space of possible functions, usually called the hypothesis space. To validate the effectiveness of F_θ, we define the testing data set D^{ts} for F_θ, where $D^{ts} = \{(\mathbf{x}_j, y_j)\}_{j=1}^{J}$.

Table 1
Definition for Variables of Basic Notations

Variables	Definition
D^{tr}	training data set
D^{ts}	testing data set
D^{cdd}	candidate data set
$F_\theta(\cdot)$	predictor with parameter θ
$S_\psi(\cdot)$	selector with parameter ψ
B_n	selected data set in the n-the iteration
$(\mathbf{x}_i, y_i)$	the i-th sample $\mathbf{x}_i$ with label y_i

In ADL, except for initial training data D^{tr}, more training samples will be selected from candidate data set D^{cdd} and provided (after labeled oracle) to F_θ in the iteration of training, where $D^{cdd} = \{(\mathbf{x}_k, y_k)\}_{k=1}^{K}$ whose labels are inintially unknown.

In this survey, we define $S_\psi(\cdot)$ as selector that is the strategy or model to select samples from candidate data set D^{cdd}, where ψ is a set of parameters. For convenient, to distinguish from selector $S_\psi(\cdot)$, we call the basic learner $F_\theta(\cdot)$ as "predictor". As in Fig.1, in an arbitrary n-th iteration, AL uses its selector $S_\psi(\cdot)$ to select a few of unlabeled samples as set B_n from D^{cdd}, labels them, and then re-trains the predictor $F_\theta(\cdot)$ by the new D^{tr}_n, where $D^{tr}_n = D^{tr}_{n-1} \cup B_n$.

The relationship between predictor $F_\theta(\cdot)$ and selector $S_\psi(\cdot)$ should be worthy of our attention. Without selector $S_\psi(\cdot)$, the predictor $F_\theta(\cdot)$ is just a conventional classifier of supervised learning. Please refer to Fig. 1 as the process of active learning. If the selector $S_\psi(\cdot)$ is un-related to the predictor $F_\theta(\cdot)$, the AL problem will be more simple. However, in many cases, $S_\psi(\cdot)$ and $F_\theta(\cdot)$ are influenced by each other. Up to now, there is still not so much research on the relationship between $S_\psi(\cdot)$ and $F_\theta(\cdot)$.

If we only select one sample from D^{cdd} to label in each iteration of AL, the selector can be defined as

$$\hat{\mathbf{x}} = \arg \max_{\mathbf{x}_k \in D^{cdd}} S_\psi(\mathbf{x}_k, D^{cdd}, D^{tr}, \theta) \tag{1}$$

where selector $S_\psi(\mathbf{x}_k, D^{cdd}, D^{tr}, \theta)$ can be related to D^{cdd}, D^{tr}, θ and even other variations or datasets. Obviously, we can also select a few of samples $\{\hat{\mathbf{x}}_n\}_{n=1}^{M}$ to form the set B_n in each iteration by computing $S_\psi(\cdot)$ for D^{cdd} and ranking them.

Active Deep Learning (ADL): Most traditional AL selectors are designed to serve for a shallow model of $F_\theta(\cdot)$. However, for a deep predictor $F_\theta(\cdot)$, how to design a good selector is still a open problem. In this survey, we call it ADL only if the predictor $F_\theta(\cdot)$ is a deep model no matter whether $S_\psi(\cdot)$ is deep model or shallow model. Actually, there are four different combinations for deep or shallow selector or predictor, but we only focus on ADL. Similar to traditional AL, ADL involves in a iterative sampling scheme, where a deep model $F_\theta(\cdot)$ is trained regularly by feeding it with new labeled sample set as $D^{tr}_n = D^{tr}_{n-1} \cup B_n$.

Model-driven or Data-driven: In most of traditional ALs, the selectors $S_\psi(\cdot)$ are model-driven. It means that we need to design a concrete form $S_\psi(\cdot)$ by handcraft feature or metric. This can be analogous to the model-driven conception in common supervised deep learning. However, if the selectors $S_\psi(\cdot)$ is a deep architecture, it means that we do not need to design explicit form of selector by handcraft feature or metric. Similar to the data-driven conception in deep learning, we called it as data-driven selector. In recent years, the study of selector is experiencing the stage form model-driven to data-driven.

2.2. Core Issue

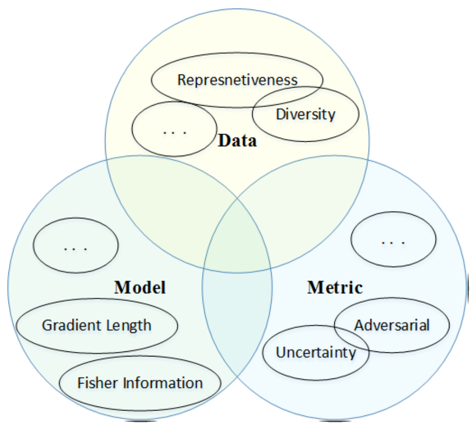


Figure 2: Three fundamental factors in designation of selector.

Usually, we hope that the new sample set B_n selected by $S_\psi(\cdot)$ should be most beneficial for the improvement of the performance of the deep predictor $F_\theta(\cdot)$. This leads to some core issues that will be addressed in the following.

Design a Selector: How to design a effective selector $S_\psi(\cdot)$ is the first core issue of ADL. There are at least three fundamental factors that determine rules of designation of selector $S_\psi(\cdot)$. As in Fig. 2, they are model, data, and metrics. **The model factor** refers to the form or structure of predictor $F_\theta(\cdot)$. For examples, if we design a selector for the predictor of Logistic Regression, it will be different from a selector for SVM in a large probability. The model factors are always existing but it plays different role in different active learning methods. For an example, there is a study such as [14] where make a benchmark and comparison of active learning for Logistic Regression. Furthermore, for ADL, its selector need to adapt to the deep predictor which differers from traditional AL. The characteristics of predictor have very deep influences on the designation of selector although not all type of selectors are based on the model influence of predictors. **The data factor** mainly refers to the intrinsic structure or statistical distribution of the labeled or un-labeled data. In most cases, the process of constructing predictors can be taken as a regression problem. The latent data distribution is the data priors for the regression problems. The relationship between one sample and its neighborhood is complex both in local or global. This often leads to variety of sample values in training set. Some selection strategies such as representativeness and diversity, are typical designation based on data distribution. **The metric factor** mainly refers to loss or status of the prediction results. The status and distribution of prediction errors will provide many references to the designation of selector. The status of prediction for un-labeled samples can show the confidence of predictor to the samples. The most common used methods such as uncertainty sampling, query by committee and error variation etc. are all belong to typical methods based on the metric factors.

For ADL, many of the traditional $S_\psi(\cdot)$ who are unrelated or weakly related to predictor $F_\theta(\cdot)$ can be directly transfer to deep network architecture such as CNN, DBN, LSTM or GANs. However, most of predictor-related selector $S_\psi(\cdot)$ can not be directly utilized for deep network architecture. For an example, the AL scheme of Fisher Information (FI) [15] is to query the instance that would impart the greatest change to the current model if we knew its label. Obviously, Fisher Information can not be directly transferred to ADL because for deep predictors how to compute Fisher Information will be thoroughly different from that for shallow predictor. For deep network architectures such as CNN, DBN, LSTM and their varieties, we have to design a new metric to evaluate the sample influences on the model.

Optimization: For both data-driven and model-driven ADL, their selectors often need to be optimized in different ways. For some handcraft metrics such as entropy, it may not need to be optimized directly. However, for Fisher Information, Sparse Representation or Gaussian Mixed model etc., it usually need to compute the gradients in searching of the solution. For data-driven ADL, the optimization for selectors usually would be more important becuase the searching of important samples may be a regression process and the selector has no explicit definition. Especially, if the selector of ADL is based on the architecture of CNN, GAN, VAE or LSTM, they need to be optimized by the type of gradient decent such as back prorogation algorithm. In another aspect, the time consumption for training of the selector is a important measurement to evaluate its feasibility since ADL is iterative method and can be time costing.

Initial Training: The initial training samples will have seriously influence on the performance of ADL. There are two type of initial training sample set: The first type is consisted in random samples; The second is consisted in selected samples. The initial sample set will determine the initial parameters for the predictors. The initial samples set will impose very complex influence on the training process of active learning since the iteration of ADL can be viewed as sequential decision problem. For deep predictors such as CNN, different initial training set will lead to different status of its parameters so that its initial predict ability will show large uncertainty. If the initial training sample set is selected by any measurement, we also call it warm start or core set method [16], other wise it is a cool star method.

Query Types: There are three main query types: Stream-Based Query, Membership Query Synthesis and Pool-Based Sampling. The choice of query types mainly depends on what type of un-annotated data we have access to and what type of annotated data the classifier requires. **Stream-Based Query**, uses the selector $S_\psi(\cdot)$ to decide for each incoming data-point, whether or not need to be annotated. This query type is usually computationally expensive and only be suitable for some classifiers that can deal with few samples. **Membership Query Synthesis** generates one or many new data-point $\hat{\mathbf{x}}$ that need to be annotated. The generated samples are informative to current model, which are usually based on un-annotated data and current status of the model. For some real-world tasks, Membership Query Synthesis may generate strange instances that are unnatural or difficult for humans to interpret. However, recent advances of Generative Adversarial Networks (GANs) show potential for generating data-points since GANs can implicitly fit complex data distributions and generate high quality samples in batches. **Pool-Based Sampling** draws samples from large un-annotated real world dataset and select a batch of samples to be query. Pool-Based Sampling usually use one selector to measure all un-annotated samples and then it ranks all data-point in the un-annotated set, and finally selects the top N samples to be annotated. Stream-Based Query, Membership Query Synthesis and Pool-based Sampling are all computationally expensive in some extents because in every iteration they usually need to re-evaluate the informativeness for every data-point. Deep learning relies on large training data set and is inherently a model with batch-based training. In the study on ADL, Membership Query Synthesis and Pool-based Sampling methods have shown promising performances when they are directly combined with deep learning frameworks.

2.3. Taxonomy

For most of surveys on traditional ALs such as [9], [10], [13] and [6], they classify ALs based on type of selectors. For examples, the selector can be based on information theory, distribution of unlabeled samples or influences on predictors etc. Obviously, both of these traditional predictors and selectors are model-driven whose features are designed by handcraft. Today, with the development of deep learning, most of the predictors had already been data-driven. At the same time, the selectors in ADL also is becoming data-driven. In this survey, if the predictor is deep model we call it ADL no matter whether its selector is or not deep model. More important, for the taxonomy, as in Fig. 3, we first propose to category ADL into two large types which are "model-driven ADL" and "data-driven ADL" based on whether its selector is data-driven or model-driven.

For model-driven ADL, their selectors are similar to that of traditional ALs. There are at least three basic kinds of approaches in model-driven ADL. The first is based on the uncertainty of unlabeled samples, such as uncertain sampling, or query-by-committee. The second is based on the influence on the model by the unlabeled samples such as length of gradients, or Fisher Information. The third is based on the intrinsic distribution and structure of the unlabeled samples such as manifold learning, Kullback–Leibler (KL) divergence similarity , Gaussian similarity, and cluster. We list Bayesian method as one separate type but it is actually often related to uncertainty methods. There are also some mixed model-driven ADL methods that employ multi-criteria in selecting new training samples for active learning. For all model-driven ADL, their common characteristics are that their selectors are model-driven which need to be designed by handcraft. Most of the selectors in model-driven ADL are transferred from traditional AL methods after changed and improved

For data-driven ADL, their selectors adopted deep architecture whose features are automatically generated but not by handcraft. In analogy with automatic feature learning, data-driven selectors is becoming more and more important and popular in the regime of active deep learning. In this survey, based on how to realize data-driven, we category data-driven ADL into four types: ADL with Meta Learning, ADL with Reinforcement Learning, ADL with Uncertainty Learning and ADL with Data Augmentation. In data-driven ADL, we may not find any name of handcraft feature or metric such as Entropy, Kullback–Leibler (KL) divergence or Gaussian similarity etc. For data-driven ADL, the most important is how to design its data-driven mechanism and structures. The advanced ideas in Reinforcement Learning, Meta Learning and Generative Adversarial Network all can be utilized to establish a data-driven selector for ADL. Therefore, most of data-driven ADL were also based on the existed new progress in current machine learning studies.

3. Model-Driven Active Deep Learning

"Model-Driven" in ADL mainly refers to that the selector $S_\psi(\cdot)$ (acquisition function or selection schemes) is an explicit model which is designed by handcraft feature or handcraft metric. In the early studies, almost all of active learning methods are model-driven. For a long time, the conception of data-driven is not well developed at the same pace for active learning with deep learning, although deep learning with data-driven has already spilled over into every area of machine learning. There are some studies that transplant the model-driven selector to serve for the deep predictor. For most of model-driven ADL, their selector is an explicit model as in traditional active learning (such as entropy, fisher information, etc.), while its predictor is a deep architecture such as CNN, DBN or LSTM etc. It is worth noting that selectors of model-driven ADL are almost all un-supervised models.

When compared with data-driven ADL, model-driven ADL have both advantages and disadvantages. **The advantages**: 1. It has clear statistical or physical meaning; 2. Many traditional methods may be directly transferred to a deep learning predictor to form new ADL; 3. The construction of selector does not need too much samples; **The disadvantages**: 1. It seriously relies on human experiences and prior assumptions. 2. The model-driven selectors are shallow model, so that they may lose efficacy and show unsteady for some deep predictor. 3. Its efficiency could be related to the type of predictors, but it is not a trivial to utilize the parameter states of the predictor of deep model.

The most obvious characteristic of model-driven ADL is that the predictor is a deep model but the selector is shallow and based on handcraft feature or metric. Many handcraft acquisition schemes such as marginal sampling, sparse representation, clustering, Fisher Information, and KL divergence are all popular model-driven selector. In the early studies, many researches mainly focus on how to transfer the tra-

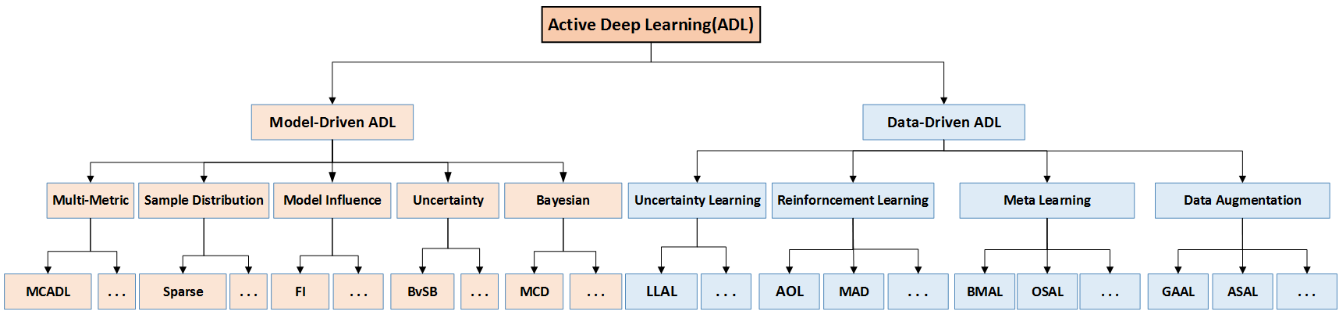


Figure 3: Taxonomy for ADL. MCADL[17], Sparse[18], FI[15], BvSB[19], MCD [20], LLAL[21] , AOL[22], MAD[23], BMAL[24], OSAL[25], GAAL [26], ASAL[27].

ditional handcraft metric into the advanced deep learning architectures. Many of these kind of transplant research are effective somehow. In the following, the different kind of model-driven ADL will be separately discussed in detail.

3.1. Model-Driven ADL with Uncertainty

Uncertainty may be the most often used metric by the selectors in active learning. This kind of methods assumes that the uncertainty samples provide more information for the training of predictor if they are labeled. Marginal Sampling [28], Best-versus-Second Best (BvSB) [29], MaxiMum Confidence Uncertainty (MCU)[30], Multiple Peak Entropy (MPE) [31] and Query by Committee (QBC) [32] methods are all belong to uncertainty methods. They are all explicit model, since they are model-driven. For ADL, these uncertainty models almost can be directly added onto deep learning predictors. To clearly address it, the output by predictor $F_\theta(\cdot)$ with input $\mathbf{x}_i$ is defined as

$$\{y_i^1, \cdots, y_i^c, \cdots, y_i^C\} = F_\theta(\mathbf{x}_i), \tag{2}$$

where y_i^c is the possibility of $\mathbf{x}_i$ belong to the c-th class, as well as

$$y_i^c \propto p(y_i = c|\mathbf{x}_i). \tag{3}$$

The Minimum Confidence Uncertainty (MCU) [30] is defined as

$$\mathbf{x} = \arg\max_{1\le i\le I}\{1 - \arg\max_{1\le c\le C} p(y_i = c|\mathbf{x}_i)\} \tag{4}$$

The Minimum Margin Uncertainty (MMU) [29] is defined as

$$\mathbf{x} = \arg\min_{1\le i\le I}\{\overset{1}{\max_{1\le c\le C}} p(y_i = c|\mathbf{x}_i) - \overset{2}{\max_{1\le c\le C}} p(y_i = c|\mathbf{x}_i)\} \tag{5}$$

The Entropy Uncertainty (EP) is defined as

$$\mathbf{x} = \arg\max_{1\le i\le I}\sum_{c=1}^{C} p(y_i = c|\mathbf{x}_i)\log p(y_i = c|\mathbf{x}_i) \tag{6}$$

We can observe that Eq. (4) (5) and (6) are mainly based on the prediction labels $\{y_i^1, \cdots, y_i^c, \cdots, y_i^C\}$ but not sample data distributions. These simple forms are very convenient when designing selectors for model-driven ADL.

For ADL, BvSB uncertainty [29] [19] may be one of most popular methods who are easily combined with different deep architecture. For example, in reference [33], BvSB uncertainty is integrated into CNN predictor for hyperspectral classification. BvSB measures the class probability difference between the most confused classes, i.e., the first and the second most probable classes. It can alleviate side influences of small class probabilities of unimportant classes. In [34], for the training data, the selection cost is taken into consideration in addition to the data labeling cost constraint. Entropy Uncertainty (EP) is also very widely used in model-driven ADL. In [35], Uncertainty query strategy was combined with deep neural networks to form a framework for detecting social bots. In reference [36], cross entropy was introduced into bidirectional gated recurrent neural networks to form a cost sensitive active learning using for imbalanced fault diagnosis. These cost constraints combined with BvSB, MCU and EP, all can be constructed as selector to form model-driven ADLs. The advantage of the algorithms in [34] is that they can be applied to any network model or data model of deep learning such as VGG16 or LSTM. In reference [37], the author proposed a active transfer learning network (ATLN) who is based on joint spectral-spatial feature learning model. ATLN is exploited to transfer the pre-trained SSAE network [38] from the source domain to the target domain, where the BvSB uncertainty is utilized to select few samples to fine-tune SSAE.

There are also some other uncertainties who are different from traditional ones. In [39], the authors uses the compatibility between spatial and temporal samples to measure the inference uncertainty and refine the multitask network In [40], Relative Support Function Difference was used as uncertainty in ADL for data stream classification. Related to uncertainty, a very interesting study is Adversarial Active Learning for Deep Networks (AALDN) [28]. It proposed a new active selection criterion based on the sensitiveness of unlabeled examples to adversarial attacks such as DeepFool [41]. It queries the unlabeled samples which are the closest to their adversarial attacks, and it labels not only the unlabeled sample but its adversarial counterparts. Both DeepFool and Marginal Sampling are explicit models, so AALDN is still a model-driven ADL. However, adversarial attacks also can be extended by a data-driven way to form a data-driven ADL.

Some studies [42] have reported that directly using uncertainty as selector may lead to worse or unsteady performances of ADL [43]. In [28], for example, it is also noticed that in the experiments uncertainty sampling may perform worse than passive random selection. In [44], there are also experiment results in which Query by Committee (QBC) and Marginal Sampling methods perform worse than passive random sampling. But there are still very less studies on why querying and annotating for the unlabeled samples with uncertainty may lead to worse predictions with lowest confidence. This survey believes that it is important to make clear the mechanism about how uncertainty metric works for a deep predictor.

3.2. Model-Driven ADL with Model Influences

In this kind of ADL, the samples are consider important and need to be annotated if they have a great impact on the model parameters of deep predictors. If the selector is a explicit model that measure the influence on the parameters of deep predictor, it is model-driven ADL with model influence. The most commonly used metric to measure the influence on the parameters is Fisher Information [45]. For ADL, it is not easy to make estimation on the significant large parameter space of the deep model such as CNN, DBN or LSTM. Designing a selector based on the influence on model by a sample has been shown to be theoretically beneficial for active learning in classical shallow models such as the Logistic Regression or Support Vector Machine. However, it still needs many improvements to extend them into ADL. To address the problem, Fisher Information is defined as

$$I(\theta) = \mathbf{E}_{\mathbf{x}}\{[\frac{\partial}{\partial\theta}F_\theta(\mathbf{x})]^2\} = \mathbf{E}_{\mathbf{x}}\{[\frac{\partial^2}{\partial\theta^2}F_\theta(\mathbf{x})]\} \tag{7}$$

where $F_\theta(\mathbf{x}_i) = \{y_i^1, \cdots, y_i^c, \cdots, y_i^C\}$, $\mathbf{E}_{\mathbf{x}}\{\cdot\}$ is the expectation. For all samples, there is

$$J(\theta|D^{tr}) = \sum_{\mathbf{x}_i \in D^{tr}} I(\mathbf{x}_i|\theta) \tag{8}$$

where θ is the parameter of deep architecture $F_\theta(\cdot)$. The number of parameter θ will be $L \times H$ if the layer number is L and the unit number is H. When we have $n-1$ training data in set D^{tr} and the corresponding predictor $F_\theta(\cdot)$ with parameter $\theta_{(n-1)}$, we select the next sample $\mathbf{x}^{(n)}$ by object function

$$\hat{\mathbf{x}}^{(n)} = \arg\min_{\mathbf{x}\in D^{cdd}} \mathrm{Tr}[I(\theta_{(n-1)})J^{-1}(\theta_{(n-1)}|D^{tr} \cup \mathbf{x})] \tag{9}$$

where $\mathrm{Tr}[\cdot]$ is the trace of a matrix. For deep architecture $F_\theta(\cdot)$, the dimension $L \times H$ of parameter θ is usually very huge. $I(\theta)$ will be a matrix with dimension $H \times L \times C$ (C is the class number) and it is very hard to compute inverse matrix $J(\theta)^{-1}$. More seriously, for ADL with Fisher Information, we may need to repeatedly compute large inverse matrix of $J(\theta)^{-1}$ for every selection in iteration.

There are already some research on how to compute Fisher Information in AL for shallow model such as [46], [47], [48], [49] and [50]. In reference [51], the authors attempt to bridge gap between the underlying theory and the motivation of its usage in practice, and then provide a rigorous framework for analyzing existing Fisher Information-based active learning methods. They pointed out that Fisher Information can be asymptotically viewed as an upper bound of the expected variance of the log-likelihood ratio.

For ADL, there are also some attempts to driven the active learning process by the model of Fisher Information (FI). In [15], Fisher Information was introduced into ADL for a CNN predictor to query samples for the first time. It developed a FI-based ADL for CNNs by an implicit re-parameter-ization of the Fisher Information model which makes it a tractable approximation. Explicit FI matrix estimation is avoided. The method makes no assumptions about the structure of the model parameters so that can be extended to different deep predictors. The FI-based ADL shows well performances while evaluated on fine-tuned a pre-trained model obtained from a source data set in [15].

Expected-Gradient-Length (EGL) is another very important metric to measure the model influences of candidate samples on the predictor of ADL. In [52], the first Expected-Gradient-Length strategy has been proposed to construct a selector for ADL. The EGL consists in selecting instances with a high magnitude gradient. The samples will have an impact on the current model parameter estimates, at the same time they will modify the shape of the decision boundaries. However, computing the true gradient for a given sample is intractable when its ground-truth label is unknown. In practice, they approximate the gradient with the expectation over the gradients conditioned on every possible class assignments. In [53], the author designed a new algorithm by Diverse Gradient Embeddings (BADGE), which samples groups of points that are disparate and high magnitude when represented in a hallucinated gradient space. It is a strategy to incorporate both predictive uncertainty and sample diversity based on EGLs. It proposed that the length of this hypothetical gradient vector captures the uncertainty of the model. For the difficulty that we need to know the label to compute the gradient, BADGE computed the gradient by assumption that current predictions on the example is the true label. However, the true label prediction hypothesis may lead to some inaccuracy problems in theories.

Overall, Fisher Information is relative rigorous in theory but very hard to compute in practice. The explicit model for EGL is relative easy to implemented. However, unlike Fisher Information, such simplified scheme often ignores the interaction between the query orders and sometimes results in poor performances in ADL.

3.3. Model-Driven ADL with Distribution of Samples

The distribution of samples is the intrinsic characteristic of data. It is usually agnostic to the structure of predictor and label consistent of prediction results. It means that some samples are important just because their position in geometric distribution and their relationship to their neighborhood. There are many methods that can help to estimate the distribution of samples. K-means, sparse representation [18] [54], PCA manifold learning [55] [56] [57] and Expectation-maximization [58] and so on are all explicit model for estimating distributions of data. Some of them, as way of model-driven, have been proved efficiency in traditional shallow predictor. However, for ADL, their effectiveness still need to be validated furhter. There are at least three type of method to utilize distribution of samples: representativeness, diversity and core-set. They connected the sample selection with the sample distribution from different point of views, since we still need a metric to make a choice even the sample distribution was well known.

The most often used method may be the representativeness of a sample. Since the data are always redundant, it is intuitively good to train deep predictor with more representative samples. In [18], sparse representation by dictionary learning is employed to search representative samples in ADL. The proposed algorithm selects training samples that maximize two selection criteria, namely representative and uncertainty. The proposed algorithm is applied for remote sensing images and shown efficient in classifying hyperspectral images. In [59], the authors proposed a Single shot active learning using pseudo annotators, where the pseudo annotators can be taken as a special way to find the most representative sample.

Core-set can also be view as a representativeness who is the representativeness of global training set but not local geometry characteristic of sub-set. In [43] and [42], it provided an upper bound for the loss function of the core-set selection problem and introduced K-Center-Greedy into ADL as a scheme to select the core-set for training. In this method, choosing center points means that the largest distance between a data repeat point and its nearest center is minimized. It is similar to cluster algorithm but not so sensitive to far and discrete sample point. The idea of core-set can also be related to Warm-Starting Neural Network Training [16]. Core-set method showed promising performances in some datasets.

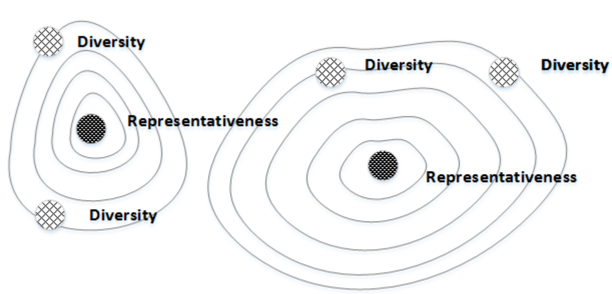


Figure 4: Diversity vs. Representativeness.

Diversity is another important data characteristic who is contrary to representativeness. Some research categories it into uncertainty method. In [60], it utilized uncertainty and similarity information provided by FCN [61] and formulate a generalized version of the maximum set cover problem to determine the most representative and uncertain areas for annotation. The method is applied to ultrasound image segmentation data set, and it is shown that state-of-the-art segmentation performance can be achieved by using only 50% of training data. In [62], the author introduced learning to diversify deep belief networks into hyperspectral image classification. In reference [63], the paper proposed a divide-and-conquer idea that directly transfers the AL sampling as the geometric sampling over the clusters. By dividing the points of the clusters into cluster boundary and core points, it theoretically discussed their margin distance and hypothesis relationship. This survey believe that diversity can also be learned by data-driven, although aforementioned ADLs with diversity are model driven.

The distribution of samples usually has nothing to do with the state of predictor. It is often integrated with other model-driven methods such as uncertainty, influence on model, Bayesian inference or data augmentation etc. The distribution of samples as a single metric is sometimes not enough to construct a good selector for ADL, because samples in some key intrinsic structure of data may not always keep being important when the state of predictor is changing.

3.4. Model-Driven ADL with Bayesian Learning

Bayesian inference is a important way to select important samples in ADL. Most of Bayesain methods make assumption on data distribution or parameter priors and they are explicit models, as well as model-driven. Some other methods are also based on Bayesian theory but without rigorous explicit model (such as VAE) so that we believe that it belongs to data-driven ADL. Some studies sometimes category Bayesian Learning into uncertainty methods, since Bayesian inference always related to the possibility and distribution. However, in this survey we believe that Bayesian Learning is not only about uncertainty. Other metrics such as representativeness, influence on model, sample distribution all can be inferred by Bayesian theory. Bayesian Learning can become an independent theory frame about selector in ADL.

In some early studies such as [64], the Bayesian inference is introduced into shallow model such as SVM. Parameters are estimated by using the evidence Bayesian approach, the kernel trick, and the marginal distribution of the observations instead of the posterior distribution of the adaptive parameters. This approach allows us to deal with infinite-dimensional feature spaces. In other research [65], the unlabeled samples whose predicted results vary before and after the MRF processing step is considered as uncertainty by Bayesian inference. However, for ADL, the predictor is a deep structure with millions of parameters so that it is untractable for directly Bayesian inference.

For conventional CNN architecture, there are no mechanisms to compute the distributions of the nodes in all hidden layer. Therefore, in some Bayesian methods, the features of Bayesian Neural Networks (BNN) are combine with the classical CNNs to form B-CNN. The most important, B-CNNs provide the basic for uncertainty model to apply Bayesain estimation for ADL. Specifically, Bayesian approach need to put some prior distribution over the space of functions by $p(F)$, so we can define a probability or likelihood on the output y given the input $\mathbf{x}$ and the predictor function F as in [66], as well as $p(y|\mathbf{x}, D^{tr})$.

$$p(y|\mathbf{x}, D^{tr}) = \int p(y|F)p(F|\mathbf{x}, \theta)p(\theta|D^{tr})dFd\theta \quad (10)$$

where θ is the parameters in hidden layer of deep networks in the predictor F. However, the probability distribution $p(\theta|D^{tr})$ is intractable for a BNN. To infer the model posterior in a simple way, variational inference [67] is often used. In this situation, the intractable probability distribution $p(\theta|D^{tr})$ is replaced by the approximate distribution $q(\theta)$ that belongs to a tractable family. And then $q(y|\mathbf{x}, D^{tr})$ is

$$q(y|\mathbf{x}, D^{tr}) = \int p(y|F)p(F|\mathbf{x}, \theta)q(\theta)dFd\theta \quad (11)$$

At the same time it needs to minimize the Kullback–Leibler divergence as

$$KL(q(\theta)||p(\theta|D^{tr})) = 0 \quad (12)$$

where $KL(\cdot)$ is a measurement of the similarity between the two distributions.

Actually, $KL(\cdot)$ is still not easy to compute because of the millions of parameters for a deep network. In [20] , the authors combine recent advances in Bayesian deep learning with the active learning framework in a practical way. It developed an active learning framework for high dimensional data by taking advantage of specialised models such as Bayesian convolutional neural networks. The uncertainty become easy to computed because the statistical model parameters of very nodes are provided by Bayesian CNN structure and can be updated with Monte Carlo Dropout (MCD) [20]. It is a very practical way for Bayesian ADL. However, not all deep model can give out the statistical model parameters of all nodes. This limited its application to a certain extent.

In [66], the Bayesian CNN (B-CNN) [20] is extended to spectral-spatial hyper-spectral image classification. B-CNN is able to offer uncertainty estimates and a probabilistic interpretation of DL models by inferring distributions over the model weights. Similar to [20], the Bernoulli approximation variational inference in BNNs was implemented by adding dropout layers after certain weight layers in a network [66]. The proposed approach improves the generalization capacity by extending the studies in [20]. It showed some robustness to over fitting on small labeled sets when applied to hyperspectral images classification. To deal with large-scale problems, in [68], the authors proposed a Bayesian batch active learning approach. The algorithm produces diverse batches that enable efficient active learning at scale. In [68], it derived interpretable closed-form solutions for the selector and was tested by several different large scale data sets.

In some studies, Bayesian method also can be combined with other deep model to form ADL. In [69], the author combined deep generative and discriminative models for Bayesian semi-supervised learning, so that it allows models to learn from labelled and unlabelled data, as well as naturally account for uncertainty in predictive distributions. In [70], the authors proposed to combine representativeness with uncertainty by Bayesian sampling, which in nature is an ADL for segmentation based on Bayesian sample queries. To be more organized, more hybrid methods will addressed in the next Section.

In summary, there are two core problem for Bayesian ADL: How to establish the statistical relationship between the parameters and the predictions; How to compute the maximum posterior probability of the parameter distribution. Most of recent studies on model-driven Bayesian ADL are based on the idea of Baysian CNN proposed by [20]. Its special architecture provide the probability of each parameter in the network, which makes it possible to perform Bayesian inference or variation inference. Actually, other type of deep model such as SAE, LSTM, GANs, DBN etc. all have their own statistical interpretations which can be leverage to inference the importance of unlabeled samples. Since the parameter space in deep predictor are usually large, how to simplify the computation is very important for the future studies on Bayesian ADL.

3.5. Hybrid Methods

In many cases, it is not easy to select the most reasonable and informative samples if the selector based on unique acquisition strategy. For an example, as in Fig. 5 the samples with most uncertainty are often clustered within a small feature space so that only few of uncertainty samples are effective to training. If we use uncertainty metric to select samples, it often lead to select samples repeatedly in a small neighborhood, which seriously decrease the effectiveness of ADL. In fact, only one sample can represent all the uncertainty samples in such small area. However, at the same time, single strategy of representativeness without uncertainty is also often not enough because some most representative samples will become unimportant with the change of parameters of predictor in training. Other method such FI, diversity, EGL and Bayesian are all face similar problem if we use one of them as unique acquisition strategy. Therefore, different selection metrics are often combined with uncertainty to construct a new selector for ADL but not singly used. Hybrid-criteria or Multi-criteria is a obvious trend in model-driven ADL, since the selection metrics in model-driven ADL are often with clear and unique physical meaning.

The combination of different strategies is not hard for most of explicit models since they have clear definition and formula. For examples: in [72], the authors propose a simple yet effective selection metric based on sample consistency which implicitly balances sample uncertainty and diversity during selection; in [73], representatives incremental learning was combined with BvSB and crowding distance; in [74], entropy and density were combined to select unlabeled instances. The combinations do not limit to traditional metric such entropy, representatives etc. but can be some new different metrics. In [75], the author proposed to combine two new type of criteria: pairwise , which depends on pairs of datapoints, and pointwise which only depends on each datapoint. They were applied to autonomous vehicle videos

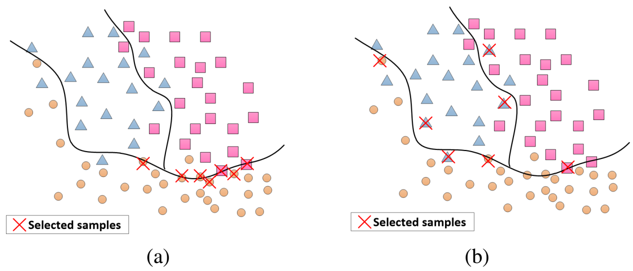


Figure 5: Hybrid Method [71]. (a) Using uncertainty as a single metric to select samples, many similar samples tend to cluster into a small area. (b) Combining uncertainty with representativeness, it can prevent the selected points from being too similar and lacking representativeness.

active deep learning. In [76], the authors proposed self-paced active learning for deep CNNs by combining clustering with effective loss function.

The Hybrid method can also be constructed by more than two type of metrics. In [17], the author devised a novel solution "multi-criteria active leep learning" (MCADL) to learn an active learning strategy for deep neural networks in image classification. The "multi-criteria" selected informative samples by simultaneously considering density, similarity, uncertainty, and label-based measure [17]. There are also similar studies such as [77] where the proposed query function considers three criteria: informativeness, diversity, and information density in CNN predictors. In another [78], the authors show that ensemble-based uncertainties consistently outperform other methods of signle uncertainty estimation (in particular Monte Carlo Dropout) and lead to state-of-the-art active learning performance on MNIST [79] and CIFAR-10 [80] dataset. It also validated the power of hybrid method for active learning in image classification.

In survey [11], it is pointed out that we can consider this hybrid query strategy in either implicit or explicit way. BADGE [53] considers the hybrid query scheme in an implicit way. Since both the prediction uncertainty of the model and the diversity of the samples in a batch are considered simultaneously, BADGE [53] can automatically balance the forecast uncertainty and sample diversity without manual hyperparameter adjustments. Wasserstein Adversarial Active Learning (WAAL) [81] proposes a hybrid query strategy that explicitly balances uncertainty and diversity [11]. In addition, WAAL [81] models the interactive procedure in AL as a distribution matching problem by using Wasserstein distance.

Multiple criteria ADL show promising performances on some popular image datasets, especially exhibits adequate performance when the problem of imbalance occurs among classes during the training process. However, there are also some limitations. Firstly, it increases the computation time because it needs to calculate the informativeness values from multiple criteria. Secondly, the weights between different strategies needs to be well balanced in the training process. If more than two strategies are simultaneously used, the coordination between them will be complex.

4. Data-Driven Active Deep Learning

"Data-Driven" ADL mainly refers that the selectors (acquisition function or selection schemes) in ADL are data-driven which are often with deep architecture, while the predictors in ADL are assumed in default to be deep architecture such as CNN, DBN, SAE or LSTM etc. The deep architectures are already very popular in many machine learning tasks such as classification, segmentation, object detection, or natural language processing etc. However, in ADL, the research on how to design a data-driven selector with deep architectures is still in its early stages. We mainly focus on the data-driven selectors, since this survey is about ADL. There are also very few of data-driven ADL methods who designed a deep selector for shallow predictor (such as SVM). We still categorize them into data-driven ADL.

We can usually take the designation of selector in ADL as a process of constructing feature or metric. Data-driven means the auto-feature learning or auto-metric learning. In most of model-driven ADLs, their selectors are un-supervised handcraft metrics or features. Different from model-driven ADL, the selectors in data-driven ADL can be either supervised or un-supervised auto-feature learning. But it is not easy to design a mechanism of supervision to realize the end to end samples acquisition, because the selector of ADL mainly faces to the unlabelled samples.

To realize data-driven, the selector can be established by either supervised or un-supervised way, or both. For the type of un-supervised data-driving, to avoid designing handcraft features, the selectors are usually based on auto-encoder (such as VAE) who can provide the representativeness of the samples, or GANs who can enhance the diversity of samples. For the type of supervised data-driven, on one hand, it is necessary to provide supervising information for training selector by setting up new task pattern with new data structure (introducing the loss, the parameter, or the states of the predictor); on the other hand, some new architectures of machine learning (meta learning, deep reinforcement learning and etc.) can be explored to model the connection between selector and predictor. The architecture of data-driven selector is not independently developed but often based on the data structure of the task and the parameter status of predictor. In many situations, we hope that the selector should be Predictor-Agnostic (PA). However, it is hard to be real Predictor-Agnostic.

When compared with model-driven ADL, data-driven ADL shows both advantages and disadvantages. **The advantages**: 1. It does not seriously relies on the human experiences and prior assumptions. 2. Its training may be slow but its prediction is usually very fast. 3. Its ability to select informative samples will be more powerful than model-driven methods. **The disadvantages**: 1. It may need more samples in training. 2. It is not a trivial to design a good network architecture for data-driving. 3. Its physical and statistical meaning are not very clear.

For most of data-driven selectors, designing the architecture and training scheme are the two fundamental problems. There are already some different studies on how to design a selector for data-driven ADL. If we take the data-driven selector as problem of few-shot learning, and then it leads to a meta learning selector. If we take the data-driven selector as an agent to act to environment (predictor), and then it leads to a reinforcement learning selector. We can also design a selector based on data augmentation with generative and adversarial networks. If we want to construct a selector by learning uncertainty, it can also be learned by deep architectures. In the following, some major types of data-driven ADL will be separately discussed in detail.

4.1. Data-Driven ADL with Meta Learning

To some extent, the goal of ADL is closely related to few-shot learning, so meta learning is a good way to realize data-driven selector in ADL. Meta-learning, also known as "learning to learn", intends to design models that can learn new skills or adapt to new environments rapidly with a few training examples. More detailed survey on meta learning can be referred to [82]. Three characteristics in meta learning are consistent with the goals of active learning. 1. New tasks: Every time when active learning makes a decision on un-labeled samples, it is similar to facing a new task; 2. Learn new skills: The selector serving for a predictor needs to acquire the new learning skills since the predictor keep being enhanced by training with newly added sample set; 3. Few training examples: The goal of active learning and meta learning both aim to obtain higher training performances by less training samples. Therefore, data-driven ADL with meta learning is close to the conception of "learning to active learn" in some extent. It is a very advanced idea to introduce meta learning into ADL. The survey believes that meta learning will play more and more important role in active learning community.

One of important features for all of ADLs with meta learning is that they all perform task repartitioning for the training set. To address it in detail, we need to first review the task structures of a meta learning. For general meta learning, the element in its training (meta) set and testing set is task which is consisted in support set (for traing) and query set (for testing). The training (meta) set is defined as

$$D^{tr} = \{\langle D_k^{tr-spt}, D_k^{tr-qry}\rangle\}_{k=1}^{K} \tag{13}$$

where

$$D_k^{tr-spt} = \{(\mathbf{x}_k^1, y_k^1), \cdots, (\mathbf{x}_k^p, y_k^p)\} \tag{14}$$

$$D_k^{tr-qry} = \{(\mathbf{x}_k^1, y_k^1), \cdots, (\mathbf{x}_k^q, y_k^q)\} \tag{15}$$

Similarly, the testing set is

$$D^{ts} = \{\langle D_n^{ts-spt}, D_n^{ts-qry}\rangle\}_{n=1}^{N} \tag{16}$$

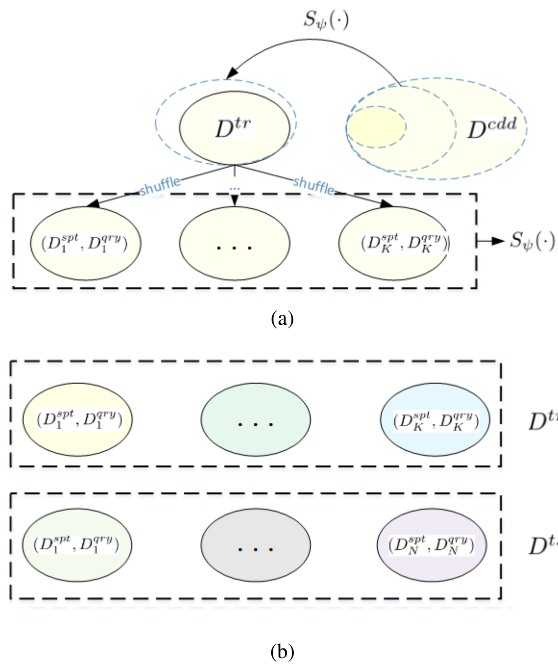


Figure 6: Task structure of Active Learning and Meta Learning. (a) is the task structure for ADL with meta learning. (b) is the task structure for conventional meta learning. The different tasks of conventional meta learning are symbolised by different colors.

where the definition of $\langle D_n^{ts-spt}, D_n^{ts-qry}\rangle$ is similar to Eq.(14) or (15). This is different from a common supervised learning (or active learning) whose element in the training set or testing set is a pair of labeled sample $(\mathbf{x}_i, y_i)$. The fundamental element for training set and testing set of meta learning is a task. For each task, $\langle D_k^{tr-spt}, D_k^{tr-qry}\rangle$ or $\langle D_n^{ts-spt}, D_n^{ts-qry}\rangle$, they usually only have very few samples. The problem of meta learning is denoted as : The meta parameter ψ should be trained in advance by different tasks in set D^{tr} as in Eq. (17), and then the parameter ϕ will be searched based on both new task in D^{ts} and the meta parameter ψ as in Eq. (18).

$$\begin{aligned}\hat{\psi} &= \arg\max_{\psi} \log p(\psi | D^{tr}) \\ &= \arg\min_{\psi} \sum_{k=1}^{K} F_{\psi}(D_k^{tr-qry} | D_k^{tr-spt})\end{aligned} \tag{17}$$

and then

$$\phi = \arg\max_{\phi} \log p(\phi | D^{ts}, \hat{\psi}) \tag{18}$$

For meta learning, the most important is to find a good meta parameter ψ so that one can train a new parameter ϕ based on ψ and $\langle D_n^{ts-spt}, D_n^{ts-qry}\rangle$, where there are only very few labeled samples for D_n^{ts-spt}. However, for ADL problem, both ψ and ϕ become the parameters of selector $S_\psi(\cdot)$ or $S_\phi(\cdot)$, and $S_\psi(\cdot)$ is the meta selector. We hope that we can train a meta selector $S_\psi(\cdot)$ based on D^{tr} and then it will select out new samples from D^{cdd} and is applicable to D^{ts} at the same time.

If constructing a selector with meta learning, the first problem is how to define D^{tr} and D^{ts} for ADL. It is necessary to reform the structure of training and testing set to make them suitable to train meta selector. As in Fig. 6 (a), **for training set** $D^{tr} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{I}$, based on idea of shuffle [22] and re-dividing schemes, one can reform the general training set into structure of the task set as Eq. (13) by K shuffle. **For testing set** D^{ts}, in ADL problem, there is not necessary testing task structure as $\langle D_n^{ts-spt}, D_n^{ts-qry}\rangle$ in D^{ts}. However, after selecting and labeling new samples from D^{cdd}, the increased samples will be merged into D^{tr} as in Fig. 6. The new D^{tr} also needs to be shuffled [22] again for training new selector before the next selection.

Although we can reform the structure of the data set for meta selector, there are still some intrinsical differences between ADL and meta learning. For ADL, as illustrated in Fig. 6, tasks are constructed by re-dividing the training set which are similar tasks but not necessary different tasks. However, for conventional meta learning, the tasks are usually different kind of tasks. In ADL, the meta learner becomes a meta selector. There are also studies such as [22] in which the meta learning is directly transformed into active meta learning because they serve for the scenarios of multi-tasks learning.

Then another important problem is risen: what kind of model is the basic predictor and do we need to establish a single separate selector who is independent to the predictor? There are some studies in which the meta learner can achieve both prediction and selection at the same time. To accomplish this no independent selection, a token bit is added into the output vector of meta learner such as in [22] [83]. The token decides whether the prediction of the sample can be trusted or the sample need to be labeled by oracle. In another way, if the basic predictor is not meta learner but a common classifier, we can construct a independent meta selector. The outputs of the meta selector are the importance of the unlabeled samples but not the prediction error. After sorting the importance of unlabeled samples, usually the top n samples will be labeled. The meta selector usually are not designed by handcraft but automatic feature learning and we do not need to explicitly defined what is importance to the training. Therefore, ADL with meta learning is typical data-driven method.

If we want to establish a separate selector who is independent to the predictor in ADL with meta learning, after predictor F_θ (conventional classier) is trained in initial, the meta selector S_ψ is trained by shuffling the training set D^{tr} for K times. It is

$$\hat{\psi} = \arg\min_{\psi} \sum_{k=1}^{K} \mathbf{E}_\pi[F_\theta(D_k^{tr-qry} | S_\psi(D_k^{tr-spt}))] \tag{19}$$

where π is an episode by shuffling D^{tr} and dividing it into $\langle D_k^{tr-spt}, D_k^{tr-qry}\rangle$. Function $\mathbf{E}_\pi[\cdot]$ is expectation of selection results for all possible episode π. After initial ψ obtained, in each iteration the new samples are selected by S_ψ from D^{cdd}, the training data set should be updated as $D_i^{tr} = D_{i-1}^{tr} \cup S_\psi(D^{cdd})$, and then both predictor F_θ and meta selector S_ψ are updated, and ADL enters the next cycle.

If the task in D^{tr} and D^{ts} are same type, a effective ϕ is initial value of meta parameter ψ. For ADL, the selector S_ψ and predictor F_θ will be alternatively updated by Eq. (19) and Eq. (20).

$$\hat{\theta} = \arg\min_{\theta} F_\theta(D^{tr}) \tag{20}$$

Furthermore, for ADL with meta learning, the task in D^{ts} is not necessary different task from D^{tr} and D^{cdd}. Although there are parameters ψ, ϕ and θ, for ADL, we mainly focus on selector parameter ψ. Meta selector $S_\psi(\cdot)$ will return a decision on D^{cdd} about who should be annotated by oracle. About ψ and ϕ, they can be similar, since for common supervised learning (not meta learning) the meta task and testing task often are not different type of tasks.

Algorithm 1 Framework of ADL with meta learning.

Input: Training data set D^{tr}; Candidate data set D^{cdd}; Testing data set D^{ts};
Output: Predictor parameter θ ; Selector parameter ϕ ; Selector parameter ψ

```
Train F_θ(·) by D^tr
repeat
    i = i+1
    repeat
        Shuffle D^tr and divide it into ⟨D_k^{tr-spt}, D_k^{tr-qry}⟩
    until k=K
    ψ̂ = Σ_{k=1}^K E_π[F_θ(D_k^{tr-qry} | S_ψ(D_k^{tr-spt}))]
    B = S_ψ(D_cdd) and label B
    D^tr = D^tr ∪ B, and D^cdd = D^cdd \ B
    Train F_θ(·) by new D^tr
until i=N
Return: S_ψ;
```

There are at least three type of common meta learning approaches: 1) Learn an efficient distance metric (metric-based); 2) Use (recurrent) network with external or internal memory (model-based); 3) Optimize the model parameters explicitly for fast learning (optimization-based). Similarly, ADL with meta learning can also be implemented in these three types, which mainly refers to designing selectors based on data-driven.

ADL with Metric-Based Meta Learning: The core idea in metric-based meta learning is similar to nearest neighbors algorithms and kernel density estimation. Each prototype is the mean vector of the embedded support points belonging to its class. In reference [24], Meta-Learning for Batch Mode Active Learning (ML-BMAL) introduced prototypical networks into samples selector of ADL. It calculates a set of statistics of each unlabeled $\mathbf{x}_i$ to the set of prototypes. And they use these statistics as input to compute two distributions quality and diversity, which represent two metrics for unlabeled item as to make decision whether to send to oracle for annotations. Learning Algorithm for Active Learning (LAAL) [84] attempts to actively learn on tasks sampled from a distribution over tasks using supervised feedback to improve its expected performance on new tasks. In LAAL, active learning takes place in the part of context-sensitive embeddings. The model states, unlabeled samples and labeled samples are all taken as inputs of the reward function, which is optimized by policy gradients of reinforcement learning. Its selector is a memory augmentation network, but its predictor is a Matching Network [85]. Therefore, it is not a pure metric based method. Acturally, LAAL is a data-driven ADL for meta learning but not only meta learning. Other metric-based method such as Convolutional Siamese Neural Network [86] [87], Relation Network [88], Matching Network [85] and etc. also can be reformed and applied to ADL in the future.

ADL with Optimization-Based Meta Learning: The goal of this type methods is to efficiently update the learner S_ψ parameters ψ using a small support set so that the learner can adapt to the new task quickly. The optimization algorithm can be explicitly modeled named as “meta-learner” [89]. For ADL problem, we not only need to derive the gradient of ϕ but also need to give out the gradient of ψ. The selector parameter ψ is updated associating with ϕ in each iteration.

The loss for meta learning parameter ψ is defined as

$$L(\psi) = \sum_{k=1}^{K} \mathbf{E}_\pi [F_\theta(D_k^{tr-qry} | S_\psi(D_k^{tr-spt})] \tag{21}$$

The parameter of meta selector is updated by gradient decent.

$$\psi \leftarrow \psi + \eta \nabla_\psi L(\psi) \tag{22}$$

where η is a learning rate parameter.

In [25] ML-OSAL is a typical ADL with meta learning of optimization-based method. There are two items in the object function $L(\psi)$ of ML-OSAL: one is for predicting error condition on the selection of unlabeled data by S_ψ; the other is the cost of labelling such as the size of the labeled set. The gradient with respect to the parameter ψ were computed using inspired policy-gradient method (likelihood-ratio trick). It also used the Monte-Carlo sampling to evaluate the explicit form of the gradient.

The optimization-based method shows clear psychical meanings and the gradient updating can be very fast. However, it could be hard to compute the gradient if the object function $L(\psi)$ for meta parameter is complex or undifferentiated. Monte-Carlo sampling will decrease its speed.

ADL with Model-Based Meta Learning: Similar to conventional meta learner, the meta selector depends on a model designed specifically for fast learning selection, as well as a model that updates its parameters rapidly with a few training steps. This rapid parameter updating can be achieved by its internal architecture or controlled by another meta-learner model. Many of them are related to RNN or LSTM. LALD [90] is a typical ADL method with meta selector of model-based method. It uses properties of classifiers (predictors) and data to predict the potential error reduction. The authors expect LALD to learn a model that automatically adapts its selection to the relative prevalence of the two classes without having to explicitly state such a rule. However the influence on classifier by a sample is related to the consequential order of that sample entering into the training data set. LAAL [84] can be also taken as model-based method of ADL with meta learning. LAAL’s meta selector is similar to a MANN [91], which consists of controller and reader. It also re-divide the training set and testing set to satisfy the requirement for training a meta selector. Another method (AMN-AOL) [92] improved an active one-shot learning method equipped with memory augmented architectures, in which it introduced the Class Margin Sampling (CMS) to improve sampling efficiency for AL. To some extent, we can take some schemes of Reinforcement Learning (RL) as a kind of model-based meta learning method, and there are many intersection between Meta Learning and Reinforcement Learning. In this survey, ADL methods with RL are listed as a independent category and discussed in the next subsection.

For a meta selector based on mode-based method, the advantages are that more priors can be introduced into model and more information can be utilized by the meta selector; the disadvantages are their architecture may be complex and their optimization is not a trivial.

For data-driven ADL with meta learning, we make some summarization: 1. In order to use meta learning to learn a meta selector, the structure of training set need to be reformed, the basic conception in meta learning such as task, support set, episode and shuffle are introduced into the training of selector, and samples in meta task are used in incremental way. 2. Many architectures and algorithms (metric-based, model-based, optimization-based, etc.) in meta learning can be used to construct the meta selector after slightly changed. The meta selector is to decide who will be labeled but not which class the samples belong to. In meta selector, there are usually no handcraft features to measure the diversity, uncertainty, and representativeness, etc. Meta selector for ADL is typical data-driven method. 3. For current existed meta selector, the value of initial labeled samples are exhaustedly exploited, but the value of unlabeled data is often not sufficiently exploited. In current studies, there are almost no mechanism to leverage the unlabeled samples in most of meta selectors. If the unlabeled samples are large but training data set is small, the distribution of the unlabeled data will carry so much information that they seriously affect the sample selection. Therefore, utilizing unlabeled samples is a problem that is worth noting in ADL with meta learning.

4.2. Data-Driven ADL with Reinforcement Learning

The data-driven selector for ADL can be realized by Deep Reinforcement Learning (DRL). DRL combines artificial neural networks with a reinforcement learning architecture so that it enables software-defined agents to learn the best actions possible in virtual environment to attain its goals. That is, it unites function approximation and target optimization, mapping state-action pairs to expected rewards.

The key factors in reinforcement learning are state s, action a, reward r, policy π and environment (for ADL environment can be $F_\theta(\cdot)$). In general, the policy π is a map-

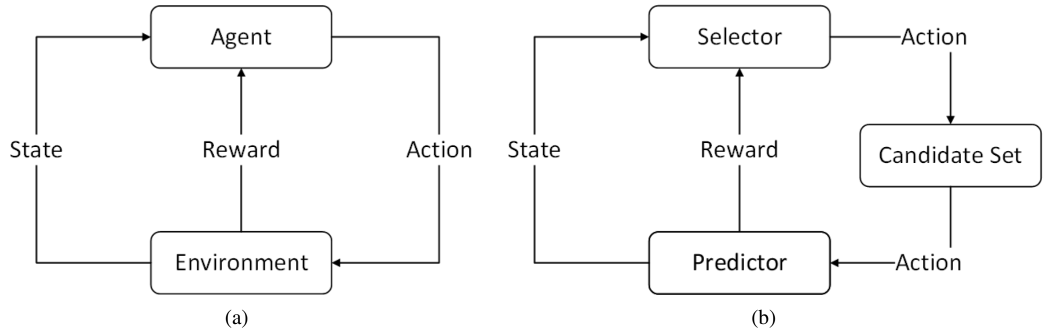


Figure 7: (a) Reinforcement Learning. (b) ADL with Reinforcement Learning.

Table 2
Definition for Variables of Reinforcement Learning

Variables	Definition
a_t	action at the t step
r_t	reward at the t step
π	policy
s_t	state at the t step
Q	predictor with parameter θ
$V^{\pi}(s)$	state value function
$Q^{\pi}(s,a)$	state-action value function with s and a
$Q_{t+1}(s_t,a_t)$	the accumulated reward at the $t+1$ step
τ	trajectory, where $\tau = \{s_1, a_1, s_2, a_2, \cdots, s_T, a_T\}$.
$p_{\psi}(\tau)$	the possibility for τ with selector parameter ψ
$J(\psi)$	the reward for possibility for all trajectory τ

ping from states to a probability distribution over actions: $\pi : \mathcal{S} \times \mathcal{A} \rightarrow p(a|s)$, where $\mathcal{S} = \{s_t\}_{t=1}^{T}$ and $\mathcal{A} = \{a_t\}_{t=1}^{T}$.

As in Fig.7, for ADL, the agent in reinforcement learning is the selector as well as policy π, the environment $F_{\theta}(\cdot)$ is the predictor (classifier). The agent (selector) observed the state s from environment $F_{\theta}(\cdot)$ and take a action a to max the reward r (prediction accuracy). The state s is the state of the predictor which can be the parameters θ in $F_{\theta}(\cdot)$. The action a is a decision on whether the unlabeled samples need to be send to oracle. The reward r is the accuracy of the predictor (classifier).

Similar to meta learning, in most cases we need to reform the structure of training data set when solving ADL problem by Reinforcement Learning (RL). The training for selector occurs within the support set created by shuffling and re-dividing the training set. At the same time, ADL with reinforcement learning can also be implemented in different ways such as Value-Based RL, Policy-Based RL and Model-Based RL. Obviously, all of them are designing selectors based on data-driven.

ADL with Value-Based RL: For a ADL, the selector $S_{\psi}(\cdot)$ mainly make decision on unlabeled samples by the states of predictor, so it is similar to the value-based RL. Referring to the idea of a value-based Reinforcement Learning method, we try to maximize a State Value Function $V^{\pi}(s)$ as in Eq.(23).

$$V^{\pi}(s) = \mathbf{E}_{\pi}[\sum_{t=0}^{T-1} \gamma^t r_{t+1} | s_t = s] \tag{23}$$

where $\mathbf{E}_{\pi}[\cdot]$ denotes the expected reward value given that the agent (selector) follows policy π, where r_{t+1} is the reward at the $(t+1)$-th step and γ^t is the discount factor. We call the Eq.(23) the State-Value Function for policy π. In this method, the agent is expecting a long-term accumulated reward expect to obtained after observing the current states s under policy π. For ADL, to construct a selector, only judging the state of predictor may be not enough. The action of labeling a also need to be considered. A common way is to define State-Action Value Function $Q(s,a)$ who is the accumulated reward expects to be obtained after seeing state s and then taking the action a under the policy π.

$$Q^{\pi}(s,a) = \mathbf{E}_{\pi}[\sum_{t=0}^{T-1} \gamma^t r_{t+1} | s_t = s, a_t = a] \tag{24}$$

With the Bellman Equation [93], Q-function $Q_{t+1}(s_t, a_t)$ is denoted as

$$\begin{aligned} &Q_{t+1}(s_t, a_t) \\ &= Q_t(s_t, a_t) + \alpha[r_t + \gamma \max_a Q_t(s_{t+1}, a) - Q_t(s_t, a_t)] \end{aligned} \tag{25}$$

This is also called Q-learning. Q-Learning is most often used in ADL with Value-based RL. The definition of state and Q-function are the two key problems when adopting RL in ADL. For different method such as classification task, state s_t can be related to the parameter of predictor, un-labeled data, labeled data, output of the predictor and etc. The reward r_t in Q-function $Q(s_t, a_t)$ is usually related to the accuracy of the classifier (predictor). When Q-function $Q(s_t, a_t)$ is taken as the selector in ADL, it need to be learned from many episodes. LHAL-DRL [94], Active One-shot Learning (AOL)[22] AOL-DQN [83] all are ADL with Value-Based RL and related to Q-Learning.

In reference [94], the author proposed LHAL-DRL based on Q-function. Let $\mathbf{x}$ be a unlabeled sample, θ be the parameter of predictor, and $A(\theta_i)$ be the prediction accuracy, then the selector of LHAL-DRL is a Q-function as

$$a_i = \arg\max_a Q^{\pi}(s_i, a) \tag{26}$$

where the state $s_t = \{\mathbf{x}, \theta_{i-1}\}$, the reward $R(s_{t-1}, a) = A(\theta_i) - A(\theta_{i-1})$, and the action $a = \{0, 1\}$. Following Deep Q-

Learning [95], LHAL-DRL makes use of a deep neural network to compute the expected Q-value. It takes the state representation as input , and two scalar values $a \in \{0, 1\}$ as output. This network uses a rectified linear unit (ReLU) activation function in its hidden layer. The parameters in the DQN are learned using stochastic gradient descent, based on a regression objective to matching the Q-values predicted by the DQN and the expected Q-values from the Bellman equation. The author showed [94] how these learned active learning policies can be transferred between languages, which empirically makes some improvements over baseline methods,

Active One-shot Learning (AOL)[22] combines reinforcement learning with one-shot learning. It introduced a classification task (predictor) in which a stream of images are presented. On each time step, a decision must be made to either predict a label or pay to receive the correct label. in AOL, the state is defined as $s_t = \{g_t, \mathbf{x}_t\}$, where $g_t = (l_t, d_t)$. The predictor output (l_t, d_t) is a one-hot vector of length $n+1$, where index t is time respectively. If the agent makes the decision to request the label, l_t will be zero-vector 0 and $d_t = 1$; otherwise, $d_t = 0$ and l_t is the predict result y_t. The token variable d_t connected the predictor with the selector network, as well as $Q(s_t, a_t)$. A recurrent neural network based Action-Value Function $Q(s_t, a_t)$ is constructed as selector to help predictor to make decision on samples. The reinforcement learning method used in AOL was quite simplistic. It can be extended by more powerful RL strategies such as better exploration [18], separate target network [13], or decomposing the action-value function [19].

In AOL-DQN [83], its task setting up is similar to AOL. With the setting up task, AOL-DQN introduced a deep Q-network strategy into one-shot learning (OL-DQN) to design a more intelligent learner to infer whether to label a sample automatically or request the true label for the active-learning set-up. The improvement is that AOL-DON used a separate target network from evaluation target network. AOL-DQN achieve a trade-off between prediction accuracy and the need of label requests. The similar enhanced study also extended to the application of Aircraft Type Recognition [96].

ADL with Policy-Based RL: The ADL can also be implemented by policy-based RL. In a policy-based RL method, it tries to set up such a policy that the action performed in every state helps to gain maximum reward in the future. Two types of policy-based methods are: Deterministic (For any state, the same action is produced by the policy π), and Stochastic (Every action has a certain probability, which is determined by the following equation). Let τ be trajectory, where $\tau = \{s_1, a_1, s_2, a_2, \cdots, s_T, a_T\}$. The possibility for τ is

$$p_\psi(\tau) = p(s_1) \prod_{t=1}^{T} p_\psi(a_t|s_t) p_\psi(s_{t+1}|s_t, a_t) \tag{27}$$

The total reward for policy τ is defined as $R(\tau) = \sum_{t=1}^{T} r_t$. The reward r_t is the accuracy of the predictor on testing data set D^{ts}. Considering the possibility for all τ, the reward is

$$J(\psi) = \sum_{\tau} R(\tau) p_\psi(\tau) \tag{28}$$

The gradient for $J(\psi)$ is

$$\nabla J(\psi) = \sum_{n=1}^{N} \sum_{t=1}^{T_n} R(\tau^n) \nabla log p_\psi(a_t^n|s_t^n) \tag{29}$$

The parameter ψ for $p_\psi(\tau)$ can be updated by

$$\psi = \psi + \eta \nabla J(\psi) \tag{30}$$

For policy-based method, we need to define the policy function $p_\psi(a_t^n|s_t^n)$, and then update its parameter ψ. If we used the gradient $\eta\nabla J(\psi)$ to update ψ, it is gradient policy method. For ADL, the predictor (learner) is the environment, and its state s_t can be defined as $s_t = \{D_t^{tr}, D_t^{cdd}, \theta_t\}$, where θ_t is the parameter for predictor F_θ in the step t. The reward r_t is the accuracy of the predictor on testing data set D_t^{ts}. The policy function is just about the parameter ψ of the selector $S_\psi(\cdot)$ in ADL.

The MLT-ALP-DRL method in [23] is a typical ADL with policy-based RL, who treating the underlying learner as part of the environment. MLT-ALP-DRL defined a DNN query criterion that is parameterised by a dataset embedding. It adapted the recently proposed auxiliary network idea in [97] to define a meta-network that provides unsupervised domain adaptation. The meta network generates a dataset embedding and produces the weight matrices that parameterise the main policy $p_\psi(a_t^n|s_t^n)$. This enables an end-to-end query policy. Furthermore, MLT-ALP-DRL framework is agnostic to the base classifier. A limitation thus far is that it only focused on a binary base classifier. In [98], the authors proposed a new active learning algorithm that selectively queries the expert, based on both a prediction of agent error and a proxy for agent risk. It is also a policy method based on Gradient loss [98].

ADL with Model-Based RL: Model-free approaches require access to an impractically large number of training interactions for most real-world problems [99]. In contrast, model-based methods reinforcement learing can quickly arrive at near-optimal control with learned models under fairly restricted dynamics classes [100]. Similarly, in ADL with model-based Reinforcement Learning method, you need to create a virtual model for current environment. The agent, as well as the selector, learns to perform in that specific environment. The model actually is a estimated status transferring matrix. If we want to learn the status transferring model or matrix, it is that we hope know s_{t+1} when we have obtained (s_t, a_t). For ADL, the action a_{t+1} is to select out the new informative samples. As mentioned in latest sub-section, the status can be defined as $s = \{D^{tr}, D^{cdd}, \theta\}$, which may be consisted in very high dimension data or parameters. Therefore, it is not a trivial to make estimation on such large volume status transferring matrix.

Up to now, we did not find any research on ADL with model-based Reinforcement Learning. In ADL, complex dynamics of training and selecting demand high-capacity models. The virtual environment model is difficult to predict. However, many improvements on model-based Reinforcement Learning such as [101] [102] [103] [104] [105] may be reformed to setup the selector in ADL. We still believe that it is a potential way that uses the knowledge of the probabilistic environment as a guide to plan the best sample selecting actions.

4.3. Data-Driven ADL with Data Augmentation

Instead of searching the informative samples within the un-labeled data, Membership Query Synthesis (MQS) method directly generates the informative samples that need to be annotated by oracle. Data augmentation is important Membership Query Synthesis way to realize data-driven selector for ADL. For deep learning, generating batch synthesis samples is also one of good way to enhance training since the most of architecture of deep learning require large scale training data. There are usually three parts in this type of ADL: predictor, selector and augmentation. Based on the definition of ADL, their predictors should be deep model but their selectors are not necessary deep model. "Data-driven" for this type of method often mainly refers to the part of data augmentation.

There are two core issues for MQS methods: How to generate samples and how to decide who are informative samples. Both explicit and implicit model can competent in generating samples. Before the rise of deep learning, there were already some studies on MQS. But most of them are explicit generative models. The explicit generative models are handcraft models analogous to handcraft features. Current ADLs with MQS method are mostly based on implicit generative models who are basically deep neural networks. GAN and VAE are mostly used generative models for current MQS methods. About how to decide informative samples, most current ADL with MQS sitll employed traditional metrics such as un-certatinty, diversity, representative etc. For MQS-ADL, it is possible to firstly generate enough samples and then select the most informative samples within them, or firstly select some informative samples and then generate more informative samples. These two type of methods are illustrated in Fig. 8. It is worth noting that MQS data-driven ADL often take traditional model-driven acquisition function as their sub-steps. Especially for data-driven ADL with GANs, many of them [26] [106] [107] leverage uncertainty sampling before or after generating new samples.

Many researchers made improvements for MQS-ADLs from different aspects such as decision metrics, generators, discriminators, predictors, selectors, architectures and etc. Generative Adversarial Active Learning (GAAL) [26] may be the first research who introduced the GANs into active learning. GAAL is more like a method of data augmentation combined with uncertain decision. Actually, GAAL is not rigorous ADL since its classifier is SVM. It only use a deep learning (GANs) to help generate informative samples for shallow classifier model SVM. GAAL can not be directly extended to deep learning classifier such DBN or CNN because it uses the distance to the decision boundary as uncertainty metric which may not suitable for many types of neural networks. ADBA [108] is another active learning method that is mainly tested by shallow classifier. However, ADBA ask for annotations of the decision boundary but not samples. It achieves this using a deep generative model to create novel instances along a 1d line. For ADBA, it is hard for a human oracle to detect a class change for non-visual data and precisely annotating the decision boundary is often not trivial for the generated samples with a large margin. Furthermore, Both GAAL and ADBA are tested on small and very simple datasets and cover only binary classification tasks with shallow model.

There are already more studies that introduced GANs into predictors (classifiers) of deep learning such as VGG16, ResNet and etc. since deep models are current main stream of classifiers. For examples, ASAL[27], BGAL [109], CGANAL [106] and ASL-GAN [107], they are methods with deep predictors and designed for large data set.

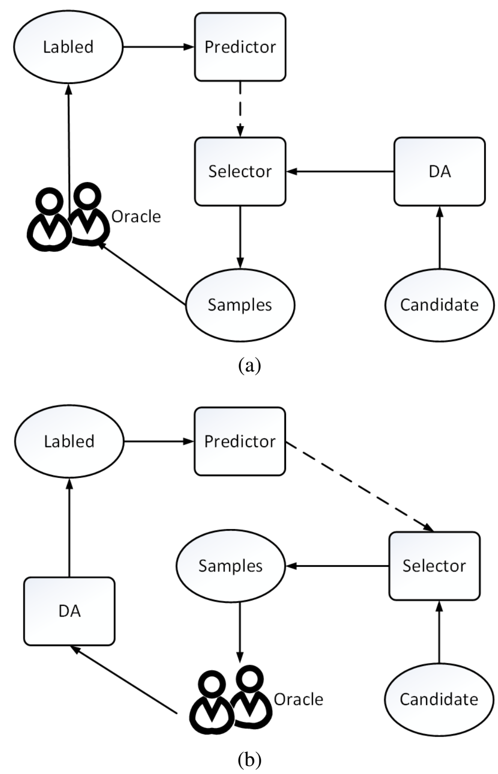


Figure 8: (a)Data Augmentation (DA) before oracle labelling. (b) Data Augmentation (DA) after oracle labelling.

ASAL method reuses the sample generation idea of GAAL [26] but takes information entropy as uncertainty score so that it can be directly extend it to multiple class problem with deep classifier. Another important difference is that ASAL compares real pool samples to generated synthetic samples in a feature space and retrieves the closest matches to be annotated, as well as Samples Matching.

Different from GAAL, ADBA and ASAL, the method of BGAL [109] generates the informative candidate samples after they are labeled by oracle. Therefore, the steps are: selecting uncertainty samples, annotating samples by oracle, and then generating more informative samples. Furthermore, BGAL trains the generative and classification models jointly. At the same time, "co-evolve" of generator and learner during training may lead to larger computational burden. It is shown that, in BGAL [109] a synthetic sample generated from the most informative sample belongs to its sufficiently small neighborhood. BGAL does not need to match samples from the set of un-labeled samples as ASAL.

ASL-GAN [107], similar to ASAL and BGAL, it simultaneously maintains three deep component modules, i.e., a generator, a discriminator, and a predictor. However, its metric for sample selection is a self-expressive correlation estimation method to reveal the underlying inter-instance correlation. The whole architecture is trained by jointly optimizing loss functions w.r.t. the corresponding component networks in an alternating update fashion. The architecture of ASL-GAN is similar to ASAL but it metric is representatives of samples. It may lead to computation burden, although its self-expressive correlation is constrained by sparseness in regularization.

In ALCGAN [106], the sample generator takes a test image and a manually segmented mask (and its variations) as input and generates realistic looking images, where the mask is the condition for GANs. The predictor adopted a Bayesian Neural Network makes a prediction on the generated samples and it can directly provide information for uncertainty estimation. ALCGAN generate images with mask, which are taken as half labelled but not unlabelled samples. It is different from ASAL and BGAL, although they all generate samples before oracle labelling.

Not all methods generate augmented data with GANs. In AL-RAC [110], without using GANs, it propose a active learning method for skin lesion analysis, which is belong to the type of augmentation after labelling. The framework consists of sample selection and sample aggregation. Specifically, it designs dual-criteria to select informative samples. Furthermore, for effective utilization of the selected samples, it also design an aggregation strategy by augmenting intra-class images in pixel space, so that it captures richer and more distinguishable features from these valuable yet ambiguous selected samples. The approach are validated on ISIC 2017 Skin Lesion Classification Challenge [111] and achieve well performances.

In data-driven ADL with data augmentation, most of studies put the emphasis on generating new informative samples, so that many of them adopt traditional selector such as entropy or marginal sampling etc. It means that the data-driven part of active learning is generator but not selector. However, technically, both generator and selector can be data-driven. Many studies combine model-driven selector with data augmentation in ADL just because it is simple and easy to implement. This survey believe that combination of deep selector with data augmentation will be more promising in ADL. There is a obvious advantage for data-driven ADL with data augmentation: It can conveniently provide more labelled samples to train a deep architecture. This is important because deep learning does need large quantity of samples. The side effect is: If we first generate unlabeled samples, for some real-world tasks, MQS data augmentation may generate strange instances that are unnatural or difficult for humans to interpret; if we generate the selected and labelled samples, it is hard to ensure that the new samples can always be accurately related to its expected label. Overall, data augmentation as recent popular research is showing its value and opening a new way for data-driven ADL.

4.4. Data-Driven ADL with Uncertainty Learning

Uncertainty is one of the most popular method in model-driven ADL, for examples entropy or marginal sampling. At the same time, uncertainty also can be estimated by automatic feature learning, as well as data-driven uncertainty learning. In data-driven uncertainty, there are no handcraft features but only automatic features such as hidden layers in CNN. Furthermore, the definition of uncertainty may be more feasible if it is data-driven. There are at least three type of data-driven uncertainties in ADL: Loss Learning [21], Discriminate Learning [113] and Adversarial Learning [114]. Different type of data-driven uncertainty can also be combined to form hybrid methods, such as [115] and [116].

If we take the sample selection problem in ADL as binary classification problem, it will lead to discriminate learning for data-driven ADL. In this case, the selector makes a discrimination on labeled or un-labeled samples. As in Discriminative Active Learning [113] (DAL), assuming the unlabeled pool is large enough to represent the true distribution, we can instead ask for every example how certain we are that it came from the unlabeled pool as opposed to our current labeled set. If the examples from the labeled set are indistinguishable from the unlabeled pool, then it means that the selectors have successfully represented the distribution with the labeled set. DAL [113] believe that if we can determine with high probability that an unlabeled example came from the unlabeled set, then the unlabeled sample is different from the current labeled data set, and so it should be informative. This proposition allows us to pose the active learning problem as a binary classification problem between the unlabeled D^{cdd} and the labeled D^{tr}.

Usually, we think the samples are with uncertainty if they are close to the decision boundary. These samples can also be taken as adversarial samples to deep predictors. Therefore, the samples who are selected by an adversarial model such as DeepFool [117] may be well probably informative samples and need to be labeled. In reference [114], it focuses on examples lying close to the decision boundary. While measuring the exact distance to the decision boundaries is intractable, the authors propose to rely on adversarial examples. The authors do not consider anymore them as a threat instead but exploit the information they provide on the distribution of the input space. It is empirically shown that adversarial active queries yield faster convergence of CNNs trained on MNIST, Shoe-Bag and Quick-Draw datasets.

The uncertainty of representation loss can be also com-

Table 3
Data-driven ADL with Data Augmentation

Method	Selector	Generator	Discriminator	Predictor	Data Augmentation	Match	Architecture
GAAL [26]	Uncertainty	CNN	CNN	SVM	Before Oracle	×	DCGAN
ADBA [108]	Uncertainty	MLP	MLP	SVM	After Oracle	×	GAN
ASAL [27]	Entropy	CNN	CNN	CNN	Before Oracle	√	WGAN
BGAL [109]	MC-DropOut	VAE	BNN	ResNet18	After Oracle	×	VAE-ACGAN
CGANAL [106]	MC-DropOut	ResNet	CNN	VGG16	Before Oracle	×	C-GAN
ASCGAN [107]	self-expressive	DeConv	CNN	CNN	Before Oracle	×	W-GAN
AL-RAC [110]	Loss+Similar	t-SNE	-	ResNet-101	After Oracle	×	-
ARAL [112]	Reconstruction Loss	VAE	MLP	DRN+VGG16	Before Oracle	×	VAE+GAN -

bined with adversarial learning. VAAL[115] proposed an active learning algorithm that implicitly learns this sampling mechanism in an adversarial manner. This method combined Variational Auto-Encoder (VAE) with an adversarial network and then trained to discriminate between unlabeled and labeled data. The mini-max game between the VAE and the adversarial network is played. While the VAE tries to trick the adversarial network into predicting that all data points are from the labeled pool, the adversarial network learns how to discriminate between dissimilarities in the latent space. VAAL was evaluate on various image classification and semantic segmentation benchmark datasets such as CIFAR10/100, Caltech-256, ImageNet, Cityscapes, and BDD100K. The method demonstrates promising results in large-scale settings and provides for a computationally efficient sampling method.

TAVAAL [116] is another hybrid uncertainty learning method. It simultaneously takes advantage of data distribution and model uncertainty approaches. Similar to VAAL, it also considered data distribution of both labeled and unlabeled pools. TAVAAL incorporated learning loss prediction module and RankCGAN [118] concept into VAAL by modeling loss prediction as a ranker. The experiments for TAVAAL demonstrated some advantages on various balanced and imbalanced benchmark datasets.

Loss in the predictor can also be taken as kind of uncertainty. In Loss Learning for Active Learning (LLAL) [21] the authors propose a novel active learning method that is simple but task-agnostic for deep predictors. It attached a small parametric module, named "loss prediction module," to a selector, and learn it to predict prediction losses of unlabeled inputs. Then, this selector can suggest data that the predictor model is likely to produce a wrong prediction. The method were validated on image classification, object detection, and human pose estimation, with the recent network architectures. And it also was promoted and applied to hyperspectral image classification [119]

For data-driven ADL with uncertainty learning, the most advantages is that the conception of uncertainty is becoming wide but not limited to entropy , marginal sampling and etc. Conventional model-driven uncertainties are usually based on unsupervised model who are easy to implemented. But the data-driven uncertainty sometime may need more complex metric than that of model-driven uncertainty if we want to design a supervised scheme to implement uncertainty learning. Overall, we still believe data-driven uncertainty is a very promising direction for ADL.

5. Relevant Learning Problems

Transfer Learning: Transfer Learning [121] [122] is another interesting paradigm to tackle the problem of insufficient samples. It focuses on reusing the gained knowledge, as well as applying it to a different but related problem. For example, knowledge gained to recognize horses could apply when trying to recognize deers. Many deep networks have good transfer-ability. It is a popular approach in deep learning that pre-trained models are used as the starting point on some related computer vision and natural language processing tasks. A initial network on a big dataset such as ImageNet [2] is often as the initial weights in a new classification task. Understanding the relationship between transferred domains is an one of most important research tasks. Different from ADL, transfer learning puts its emphasis on the transferring (instance-based transfer learning, feature-representation transfer learning, parameter-transfer learning, relational-knowledge transfer learning) but not labeling new samples, although both of them can be leveraged to tackle the problem of insufficient samples.

Few-shot Learning: Few-shot represent another paradigm for building models with extremely limited data. Few-Shot Learning can rapidly generalize from new tasks of limited supervised experience using prior knowledge. Many related research such as Meta Learning [123], Neural Architecture Search [124], Metric Learning [125], all can be taken as few-shot learning. The unreliable empirical risk minimizer is the core issue of FSL [126]. Based on how prior knowledge is used to deal with the core issue, in reference [126], the authors categorize different few-show learning into three perspectives: data uses the prior knowledge to augment the supervised experience, model constrains the hypothesis space by prior knowledge, and algorithm uses prior knowledge to alter the search for the parameter of the best hypothesis in the hypothesis space. Comparing with ADL, few-show learning takes more emphasis on the generalization ability of model but not the selection ability for samples. We believe that few-show learning is bridging this gap between AI and human-like learning.

Table 4
Active Learning with Uncertainty Learning. Matching Networks (MN), Random Forest (RF), Multi-Layered Perceptrond(MLP), Variational Auto-Encoder (VAE), Single Shot multibox Detector (SSD), Fully Connected layer (FC). θ is the paramter of selector $F_\theta(\cdot)$

Method	Selector	Predictor	Selector of Input	Loss	TestingData
LAL [90]	RF	RF	θ and D^{cdd}	L_2	BRATS [120] etc.
LLAL [21]	FC	SSD	θ and D^{cdd}	Ranking Loss	PASCAL VOC 2007
DL [113]	MLP	VGG16, LeNet	$D^{tr} and D^{cdd}$	Log Loss	MINIST, CIFAR-10, etc.
AAL [114]	Deep-Fool	VGG8, LeNet5	D^{cdd}	Perturbation	MINIST, Shoe-Bag , Quick-Draw
TAVAAL [116]	VAE +MLP	ResNet18	$D^{tr} and D^{cdd}$	Ranking Loss	MINIST, CIFAR-10
VAAL[115]	VAE +MLP	VGG16	$D^{tr} and D^{cdd}$	Log Loss	BDD100K, CIFAR-10, Caltech-256

Weak-supervised Learning: Weak-supervised learning has very strong relationship with ADL. Except from the difficulty of labelling large quantity samples, in many tasks it is also difficult to get high quality samples with fully ground-truth labels (strong supervision) due to the high cost of data labeling process. Weak-supervised learning is a machine learning techniques who are able to work with weak supervision. Typically, there are three types of weak supervision [127]: incomplete supervision, inexact supervision, inaccurate supervision. In some studies, ADL is categorized into incomplete supervision learning of weak-supervision problem. In this case, ADL is a subbranch of weak-supervised learning. ADL is mainly about how to select and label more high informative samples but it does not care about the quality of the label. Weak-supervised learning is not only about the quantity of samples but also the quality of the samples or their labels. Weakly supervised learning is an umbrella covering a variety of studies which attempt to construct predictive models by learning with weak supervision [127].

6. Discussion

Uncertainty for ADL. ADL can use uncertainty to select samples, but it does not mean it can totally avoid uncertainty in training process. For most of shallow models such as SVM, Logistic Regression etc., their uncertainty is not so obvious in training. However, the uncertainty of ADL is very obvious as it is based on deep model. Training a predictor based on neural network is always affected by stochastic factors such as data augmentation, mini-batch selection, normalization, initial labeled data, size of the selected sample set and etc. When both predictor and selector are deep model by data-driving, the uncertainty will be more difficult to eliminate. Some studies [42] [43] [28] [28] [44] have reported that the newly designed selector may lead to worse or unsteady performances of ADL, or even perform worse than passive random selection. The performance comparisons in many studies are not controlled by same baseline such as parameters, initial samples, optimization and etc. In reference [44], it detailed addressed the problem of towards robust and reproducible ADL. This survey believe the steady and reproducible problem will be more and more important in the regime of ADL research.

Explainability for ADL. Explainable for ADL is another very important problem that is worthy to notice. For most of model-driven Al or ADL, there are not explainable problems since their models are with clear physical meaning or statistical meaning. For data-driven ADL, both of the predictor and the selector may be deep model with automatic feature learning. In many cases, we can not simply interpret the them by uncertainty, diversity, preventiveness or others. There are already many studies about explainability of common deep learning, however, the studies on the explainability of model-driven ADL are very few especially for the selector in data-driven ADL. Their explainability will be obviously different from common research of explainability on deep architecture by at leas two reasons: The selector in data-driven ADL is responsible for determining the informativeness of the samples but not their classes; The selector is interacted with predictor in training and selecting process. The research on explainability of data-driven ADL will give us a new insight into the mechanisms about how ADL works.

Connection with Cognition Science. We can understand ADL from the point of view of cognition science such as a curiosity learning. Maybe,this is the first time to interpret ADL with curiosity conception. When we were young, our consciousness or mental is not very maturity. At this stage, young kids always have more curiosity on this world and will like to engage in risky activities. This kind of experiments can improve the ability of cognition faster. The grouping up of a person is similar to the training process of a classifier or predictor and the curiosity is similar to the active learning. With the person obtaining more and more experiences, its curiosity usually become weak just as the function of active learning will be vanished when there are enough labelled samples for the classifier. Similarly, the curiosity of a kid is usually strong than that of a old man.

Active Learning VS. Passive Learning. There are many other studies on how to decrease training samples but increase efficiency of training, such as transfer learning, meta learning, etc. But most of them a passive learning, oracle labelling is not involved into these passive learning methods. When comparing to passive learning, there are both advantages and disadvantages for active learning. **Advantages**: 1. The information provided by oracle labelling is accurate in most cases, so that active learning is a direct and practice method but not only in theory. **Disadvantages**: 1. It is not totally automatic method, so that the human interaction sometimes may increase the inconvenient and limit the scenario of its application. This survey believe that how to labelling samples is becoming important because most current AI algorithms rely on large quantity labelled samples.

Model-driven VS. Data-driven. In this survey, for the first time, ADL methods are categorised into model-driven type and data-driven type. In some aforementioned sections, we already summarized the advantages and disadvantages of the two type of ADL. Model-driven ADLs usually can be understood well, since most of them are inherited from traditional active shallow learning. Data-driven ADL is still not fully developed and still in its early stages. Or even for some cases, we can not prove that data-driven ADL is obviously better than model-driven one. But data-driven has began to show its power in constructing a selector for ADL. We believe in the future the researcher will find better way to design data-driven ADL and fully play to its strengths. Furthermore, except for data-driven, we believe that Knowledge-Driven ADL and Curiosity-Driven ADL are not very far from us in future in the community of active learning.

7. Conclusion

In this article, we surveyed the state-of-the-art of active deep learning. We have performed a comprehensive analysis and classified a wide range of ADL strategies, along the discriminative criterion: whether their selectors are data-driven or data-driven, and how to design a data-driven selector with deep architectures. In fact, ADL strategies do have very different features and developed very fast under the stimulus of study of deep learning. More important, we also pointed out that the selector in ADL is experiencing the stage from model-driven to data-driven.

We believe that the classification and systematization of these ADL studies can help reader to better understand and use them.

8. Acknowledge

This work was supported by the National Natural Science Foundation of China (Nos. 61731022, 41971397)

References

[1] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778. IEEE Computer Society, 2016.

[2] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA*, pages 248–255. IEEE Computer Society, 2009.

[3] Peng Liu, Kim-Kwang Raymond Choo, Lizhe Wang, and Fang Huang. Svm or deep learning? a comparative study on remote sensing image classification. *Soft Computing*, 21(23):7053–7065, 2017.

[4] Peng Liu, Liping Di, Qian Du, and Lizhe Wang. Remote sensing big data: theory, methods and applications, 2018.

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

[6] Burr Settles. Active learning literature survey. Technical report, University of Wisconsin-Madison Department of Computer Sciences, 2009.

[7] Valerii Fedorov. Optimal experimental design. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2(5):581–589, 2010.

[8] Fredrik Olsson. A literature survey of active machine learning in the context of natural language processing. 2009.

[9] Mehdi Elahi, Francesco Ricci, and Neil Rubens. A survey of active learning in collaborative filtering recommender systems. *Comput. Sci. Rev.*, 20:29–50, 2016.

[10] D. Tuia, M. Volpi, L. Copa, M. Kanevski, and J. Munoz-Mari. A survey of active learning algorithms for supervised remote sensing image classification. *IEEE Journal of Selected Topics in Signal Processing*, 5(3):606–617, 2011.

[11] Pengzhen Ren, Yun Xiao, Xiaojun Chang, Po-Yao Huang, Zhihui Li, Xiaojiang Chen, and Xin Wang. A survey of deep active learning. *CoRR*, abs/2009.00236, 2020.

[12] Christopher Schröder and Andreas Niekler. A survey of active learning for text classification using deep neural networks. *CoRR*, abs/2008.07267, 2020.

[13] Samuel Budd, Emma C. Robinson, and Bernhard Kainz. A survey on active learning and human-in-the-loop deep learning for medical image analysis. *CoRR*, abs/1910.02923, 2019.

[14] Yazhou Yang and Marco Loog. A benchmark and comparison of active learning for logistic regression. *Pattern Recognition*, 83:401–415, 2018.

[15] Jamshid Sourati, Ali Gholipour, Jennifer G. Dy, Xavier Tomas-Fernandez, Sila Kurugol, and Simon K. Warfield. Intelligent labeling based on fisher information for medical image segmentation using deep learning. *IEEE Trans. Med. Imaging*, 38(11):2642–2653, 2019.

[16] Jordan T. Ash and Ryan P. Adams. On the difficulty of warm-starting neural network training. *CoRR*, abs/1910.08475, 2019.

[17] Jin Yuan, Xingxing Hou, Yaoqiang Xiao, Da Cao, Weili Guan, and Liqiang Nie. Multi-criteria active deep learning for image classification. *Knowl. Based Syst.*, 172:86–94, 2019.

[18] P. Liu, H. Zhang, and K. B. Eom. Active deep learning for classification of hyperspectral images. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 10(2):712–724, Feb 2017.

[19] Xiaoming Lv, Fajie Duan, Jia-Jia Jiang, Xiao Fu, and Lin Gan. Deep active learning for surface defect detection. *Sensors*, 20(6), 2020.

[20] Yarin Gal, Riashat Islam, and Zoubin Ghahramani. Deep Bayesian Active Learning with Image Data. 2017.

[21] Donggeun Yoo and In So Kweon. Learning loss for active learning. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 93–102. Computer Vision Foundation / IEEE, 2019.

[22] Mark Woodward and Chelsea Finn. Active one-shot learning. *CoRR*, abs/1702.06559, 2017.

[23] Kunkun Pang, Mingzhi Dong, Yang Wu, and Timothy M. Hospedales. Meta-learning transferable active learning policies by deep reinforcement learning. *CoRR*, abs/1806.04798, 2018.

[24] Sachin Ravi and Hugo Larochelle. Meta-learning for batch mode active learning. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Workshop Track Proceedings*. OpenReview.net, 2018.

[25] Gabriella Contardo, Ludovic Denoyer, and Thierry Artières. A meta-learning approach to one-step active learning. *CoRR*, abs/1706.08334, 2017.

[26] Jia-Jie Zhu and José Bento. Generative Adversarial Active Learning. pages 1–11, 2017.

[27] Christoph Mayer and Radu Timofte. Adversarial sampling for active learning. In *The IEEE Winter Conference on Applications of Computer Vision*, pages 3071–3079, 2020.

[28] Melanie Ducoffe and Frederic Precioso. Adversarial active learning for deep networks: a margin based approach. *arXiv preprint arXiv:1802.09841*, 2018.

[29] Ajay J. Joshi, Fatih Porikli, and Nikolaos Papanikolopoulos. Multi-

class active learning for image classification. In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA*, pages 2372–2379. IEEE Computer Society, 2009.

[30] Vít Ruzicka, Stefano D'Aronco, Jan Dirk Wegner, and Konrad Schindler. Deep active learning in remote sensing for data efficient change detection. *CoRR*, abs/2008.11201, 2020.

[31] Buyu Liu and Vittorio Ferrari. Active learning for human pose estimation. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 4373–4382. IEEE Computer Society, 2017.

[32] Seho Kee, Enrique del Castillo, and George Runger. Query-by-committee improvement with diversity and density in batch active learning. *Information Sciences*, 454-455:401 – 418, 2018.

[33] Xiangyong Cao, Jing Yao, Zongben Xu, and Deyu Meng. Hyperspectral Image Classification With Convolutional Neural Network and Active Learning. *IEEE Transactions on Geoscience and Remote Sensing*, pages 1–13, 2020.

[34] G. Joo and C. Kim. Midas: Model-independent training data selection under cost constraints. *IEEE Access*, 6:74462–74474, 2018.

[35] Yuhao Wu, Yuzhou Fang, Shuaikang Shang, Jing Jin, Lai Wei, and Haizhou Wang. A novel framework for detecting social bots with deep neural networks and active learning. *Knowledge-Based Systems*, 211:106525, 2021.

[36] Peng Peng, Wenjia Zhang, Yi Zhang, Yanyan Xu, Hongwei Wang, and Heming Zhang. Cost sensitive active learning using bidirectional gated recurrent neural networks for imbalanced fault diagnosis. *Neurocomputing*, 407:232 – 245, 2020.

[37] Cheng Deng, Yumeng Xue, Xianglong Liu, Chao Li, and Dacheng Tao. Active transfer learning network: A unified deep joint spectral-spatial feature learning model for hyperspectral image classification. *IEEE Trans. Geosci. Remote. Sens.*, 57(3):1741–1754, 2019.

[38] J. Xu, L. Xiang, Q. Liu, H. Gilmore, J. Wu, J. Tang, and A. Madabhushi. Stacked sparse autoencoder (ssae) for nuclei detection on breast cancer histopathology images. *IEEE Transactions on Medical Imaging*, 35(1):119–130, 2016.

[39] Yan Tian, Guohua Cheng, Judith Gelernter, Shihao Yu, Chao Song, and Bailin Yang. Joint temporal context exploitation and active learning for video segmentation. *Pattern Recognition*, 100:107158, 2020.

[40] Paweł Ksieniewicz, Michał Woźniak, Bogusław Cyganek, Andrzej Kasprzak, and Krzysztof Walkowiak. Data stream classification using active learned neural networks. *Neurocomputing*, 353:74 – 82, 2019. Recent Advancements in Hybrid Artificial Intelligence Systems.

[41] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: A simple and accurate method to fool deep neural networks. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 2574–2582. IEEE Computer Society, 2016.

[42] Ozan Sener and Silvio Savarese. A geometric approach to active learning for convolutional neural networks. *arXiv*, abs/1708.00489, 2017.

[43] Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. *arXiv preprint arXiv:1708.00489*, 2017.

[44] Prateek Munjal, Nasir Hayat, Munawar Hayat, Jamshid Sourati, and Shadab Khan. Towards robust and reproducible active learning using neural networks. *arXiv*, pages arXiv–2002, 2020.

[45] R. A. Fisher. *On the Mathematical Foundations of Theoretical Statistics*, pages 11–44. Springer New York, New York, NY, 1992.

[46] Kenji Fukumizu. Statistical active learning in multilayer perceptrons. *IEEE Trans. Neural Networks Learn. Syst.*, 11(1):17–26, 2000.

[47] Tong Zhang. The value of unlabeled data for classification problems. In *Proceedings of the Seventeenth International Conference on Machine Learning*, pages 1191–1198. Morgan Kaufmann, 2000.

[48] Burr Settles, Mark Craven, and Soumya Ray. Multiple-instance active learning. In *In Advances in Neural Information Processing Systems (NIPS*, pages 1289–1296. MIT Press, 2008.

[49] Steven C. H. Hoi, Rong Jin, and Michael R. Lyu. Batch mode active learning with applications to text categorization and image retrieval. *IEEE Trans. Knowl. Data Eng.*, 21(9):1233–1248, 2009.

[50] Kamalika Chaudhuri, Sham M. Kakade, Praneeth Netrapalli, and Sujay Sanghavi. Convergence rates of active learning for maximum likelihood estimation. In Corinna Cortes, Neil D. Lawrence, Daniel D. Lee, Masashi Sugiyama, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 1090–1098, 2015.

[51] Jamshid Sourati, Murat Akçakaya, Todd K. Leen, Deniz Erdogmus, and Jennifer G. Dy. Asymptotic analysis of objectives based on fisher information in active learning. *J. Mach. Learn. Res.*, 18:34:1–34:41, 2017.

[52] Ye Zhang, Matthew Lease, and Byron C. Wallace. Active discriminative text representation learning. In Satinder P. Singh and Shaul Markovitch, editors, *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, pages 3386–3392. AAAI Press, 2017.

[53] Jordan T. Ash, Chicheng Zhang, Akshay Krishnamurthy, John Langford, and Alekh Agarwal. Deep batch active learning by diverse, uncertain gradient lower bounds. *CoRR*, abs/1906.03671, 2019.

[54] Peng. Liu, Meng. Wang, Lizhe Wang, and Wei Han. Remote-sensing image denoising with multi-sourced information. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(2):660–674, 2019.

[55] Chin-Chun Chang and Po-Yi Lin. Active learning for semi-supervised clustering based on locally linear propagation reconstruction. *Neural Networks*, 63:170–184, 2015.

[56] Lijun Zhang, Chun Chen, Jiajun Bu, Deng Cai, Xiaofei He, and Thomas S. Huang. Active learning based on locally linear reconstruction. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(10):2026–2038, 2011.

[57] Oriane Siméoni, Mateusz Budnik, Yannis Avrithis, and Guillaume Gravier. Rethinking deep active learning: Using unlabeled data at model training, 2019.

[58] Andrew McCallum and Kamal Nigam. Employing em and pool-based active learning for text classification. In *Proceedings of the Fifteenth International Conference on Machine Learning*, ICML '98, page 350–358, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc.

[59] Yazhou Yang and Marco Loog. Single shot active learning using pseudo annotators. *Pattern Recognition*, 89:22 – 31, 2019.

[60] Lin Yang, Yizhe Zhang, Jianxu Chen, Siyuan Zhang, and Danny Z Chen. Suggestive annotation: A deep active learning framework for biomedical image segmentation. In *International conference on medical image computing and computer-assisted intervention*, pages 399–407. Springer, 2017.

[61] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 3431–3440. IEEE Computer Society, 2015.

[62] Ping Zhong, Zhiqiang Gong, Shutao Li, and Carola-Bibiane Schönlieb. Learning to diversify deep belief networks for hyperspectral image classification. *IEEE Trans. Geosci. Remote. Sens.*, 55(6):3516–3530, 2017.

[63] Xiaofeng Cao. A divide-and-conquer approach to geometric sampling for active learning. *Expert Syst. Appl.*, 140, 2020.

[64] P. Ruiz, J. Mateos, G. Camps-Valls, R. Molina, and A. K. Katsaggelos. Bayesian active remote sensing image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 52(4):2186–2196, 2014.

[65] S. Sun, P. Zhong, H. Xiao, and R. Wang. An mrf model-based active learning framework for the spectral-spatial classification of hyperspectral imagery. *IEEE Journal of Selected Topics in Signal Processing*, 9(6):1074–1088, 2015.

[66] J. M. Haut, M. E. Paoletti, J. Plaza, J. Li, and A. Plaza. Active

learning with convolutional neural networks for hyperspectral image classification using a new bayesian approach. *IEEE Transactions on Geoscience and Remote Sensing*, 56(11):6440–6461, Nov 2018.

[67] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural networks. *arXiv preprint arXiv:1505.05424*, 2015.

[68] Robert Pinsler, Jonathan Gordon, Eric T. Nalisnick, and José Miguel Hernández-Lobato. Bayesian batch active learning as sparse subset approximation. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 6356–6367, 2019.

[69] Jonathan Gordon and José Miguel Hernández-Lobato. Combining deep generative and discriminative models for bayesian semi-supervised learning. *Pattern Recognition*, 100:107156, 2020.

[70] Firat Ozdemir, Zixuan Peng, Philipp Fuernstahl, Christine Tanner, and Orcun Goksel. Active learning for segmentation based on bayesian sample queries. *Knowledge-Based Systems*, page 106531, 2020.

[71] Wen Shu, Peng Liu, Guojin He, and Guizhou Wang. Hyperspectral image classification using spectral-spatial features with informative samples. *IEEE Access*, 7:20869–20878, 2019.

[72] Mingfei Gao, Zizhao Zhang, Guo Yu, Sercan O. Arik, Larry S. Davis, and Tomas Pfister. Consistency-based semi-supervised active learning: Towards minimizing labeling cost, 2020.

[73] David Muñoz, Camilo Narváez, Carlos Cobos, Martha Mendoza, and Francisco Herrera. Incremental learning model inspired in rehearsal for deep convolutional networks. *Knowledge-Based Systems*, 208:106460, 2020.

[74] Luiz F.S. Coletta, Moacir Ponti, Eduardo R. Hruschka, Ayan Acharya, and Joydeep Ghosh. Combining clustering and active learning for the detection and learning of new image classes. *Neurocomputing*, 358:150 – 165, 2019.

[75] Soumi Das, Sayan Mandal, Ashwin Bhoyar, Madhumita Bharde, Niloy Ganguly, Suparna Bhattacharya, and Sourangshu Bhattacharya. Multi-criteria online frame-subset selection for autonomous vehicle videos. *Pattern Recognition Letters*, 133:349 – 355, 2020.

[76] Tianxiang Yin, Ningzhong Liu, and Han Sun. Self-paced active learning for deep cnns via effective loss function. *Neurocomputing*, 2020.

[77] Sergio Matiz and Kenneth E. Barner. Inductive conformal predictor for convolutional neural networks: Applications to active learning for image classification. *Pattern Recognition*, 90:172 – 182, 2019.

[78] William H Beluch, Tim Genewein, Andreas Nürnberger, and Jan M Köhler. The power of ensembles for active learning in image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9368–9377, 2018.

[79] Yann LeCun, Corinna Cortes, and Christopher J.C. Burges. The mnist database of handwritten digits. Technical report, 2017.

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

[81] Changjian Shui, Fan Zhou, Christian Gagné, and Boyu Wang. Deep active learning: Unified and principled method for query and training. In Silvia Chiappa and Roberto Calandra, editors, *The 23rd International Conference on Artificial Intelligence and Statistics, AISTATS 2020, 26-28 August 2020, Online [Palermo, Sicily, Italy]*, volume 108 of *Proceedings of Machine Learning Research*, pages 1308–1318. PMLR, 2020.

[82] Timothy M. Hospedales, Antreas Antoniou, Paul Micaelli, and Amos J. Storkey. Meta-learning in neural networks: A survey. *CoRR*, abs/2004.05439, 2020.

[83] Li Chen, Honglan Huang, Yanghe Feng, Guangquan Cheng, Jincai Huang, and Zhong Liu. Active one-shot learning by a deep q-network strategy. *Neurocomputing*, 383:324–335, 2020.

[84] Philip Bachman, Alessandro Sordoni, and Adam Trischler. Learning algorithms for active learning. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 301–310. PMLR, 2017.

[85] Oriol Vinyals, Charles Blundell, Tim Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. In Daniel D. Lee, Masashi Sugiyama, Ulrike von Luxburg, Isabelle Guyon, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 3630–3638, 2016.

[86] S. Zagoruyko and N. Komodakis. Learning to compare image patches via convolutional neural networks. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4353–4361, 2015.

[87] Gregory R. Koch. Siamese neural networks for one-shot image recognition. 2015.

[88] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. S. Torr, and T. M. Hospedales. Learning to compare: Relation network for few-shot learning. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1199–1208, 2018.

[89] Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

[90] Ksenia Konyushkova, Raphael Sznitman, and Pascal Fua. Learning active learning from data. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 4225–4235, 2017.

[91] Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy P. Lillicrap. Meta-learning with memory-augmented neural networks. In Maria-Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 1842–1850, 2016.

[92] Andreas Kvistad, Massimiliano Ruocco, Eliezer de Souza da Silva, and Erlend Aune. Augmented memory networks for streaming-based active one-shot learning. *CoRR*, abs/1909.01757, 2019.

[93] Richard Stuart Sutton. Temporal credit assignment in reinforcement learning. 1984.

[94] Meng Fang, Yuan Li, and Trevor Cohn. Learning how to active learn: A deep reinforcement learning approach. In Martha Palmer, Rebecca Hwa, and Sebastian Riedel, editors, *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 595–605. Association for Computational Linguistics, 2017.

[95] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin A. Riedmiller, Andreas Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nat.*, 518(7540):529–533, 2015.

[96] Honglan Huang, Yanghe Feng, Jincai Huang, Jiarui Zhang, and Li Chen. A reinforcement one-shot active learning approach for aircraft type recognition. *IEEE Access*, 7:147204–147214, 2019.

[97] Adriana Romero, Pierre Luc Carrier, Akram Erraqabi, Tristan Sylvain, Alex Auvolat, Etienne Dejoie, Marc-André Legault, Marie-Pierre Dubé, Julie G. Hussin, and Yoshua Bengio. Diet networks: Thin parameters for fat genomics. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

[98] Paul Budnarain, Renato Ferreira Pinto Junior, and Ilan Kogan. Radgrad: Active learning with loss gradients. *CoRR*, abs/1906.07838,

2019.

[99] Meng Fang, Yuan Li, and Trevor Cohn. Learning how to active learn: A deep reinforcement learning approach. In Martha Palmer, Rebecca Hwa, and Sebastian Riedel, editors, *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 595–605. Association for Computational Linguistics, 2017.

[100] Sarah Dean, Horia Mania, Nikolai Matni, Benjamin Recht, and Stephen Tu. On the sample complexity of the linear quadratic regulator. *Found. Comput. Math.*, 20(4):633–679, 2020.

[101] Sébastien Racanière, Theophane Weber, David P. Reichert, Lars Buesing, Arthur Guez, Danilo Jimenez Rezende, Adrià Puigdomènech Badia, Oriol Vinyals, Nicolas Heess, Yujia Li, Razvan Pascanu, Peter W. Battaglia, Demis Hassabis, David Silver, and Daan Wierstra. Imagination-augmented agents for deep reinforcement learning. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 5690–5701, 2017.

[102] Jacob Buckman, Danijar Hafner, George Tucker, Eugene Brevdo, and Honglak Lee. Sample-efficient reinforcement learning with stochastic ensemble value expansion. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada*, pages 8234–8244, 2018.

[103] Vladimir Feinberg, Alvin Wan, Ion Stoica, Michael I. Jordan, Joseph E. Gonzalez, and Sergey Levine. Model-based value estimation for efficient model-free reinforcement learning. *CoRR*, abs/1803.00101, 2018.

[104] Shixiang Gu, Timothy P. Lillicrap, Ilya Sutskever, and Sergey Levine. Continuous deep q-learning with model-based acceleration. In Maria-Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 2829–2838. JMLR.org, 2016.

[105] Thanard Kurutach, Ignasi Clavera, Yan Duan, Aviv Tamar, and Pieter Abbeel. Model-ensemble trust-region policy optimization. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018.

[106] Dwarikanath Mahapatra, Behzad Bozorgtabar, Jean-Philippe Thiran, and Mauricio Reyes. Efficient active learning for image classification and segmentation using a sample selection and conditional generative adversarial network. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 580–588. Springer, 2018.

[107] Xiao-Yu Zhang, Haichao Shi, Xiaobin Zhu, and Peng Li. Active semi-supervised learning based on self-expressive correlation with generative adversarial networks. *Neurocomputing*, 345:103–113, 2019.

[108] Miriam W. Huijser and Jan C. van Gemert. Active decision boundary annotation with deep generative models. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 5296–5305. IEEE Computer Society, 2017.

[109] Toan Tran, Thanh-Toan Do, Ian D. Reid, and Gustavo Carneiro. Bayesian generative active deep learning. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 6295–6304. PMLR, 2019.

[110] Xueying Shi, Qi Dou, Cheng Xue, Jing Qin, Hao Chen, and Pheng-Ann Heng. An active learning approach for reducing annotation cost in skin lesion analysis. In Heung-Il Suk, Mingxia Liu, Pingkun Yan, and Chunfeng Lian, editors, *Machine Learning in Medical Imaging - 10th International Workshop, MLMI 2019, Held in Conjunction with MICCAI 2019, Shenzhen, China, October 13, 2019, Proceedings*, volume 11861 of *Lecture Notes in Computer Science*, pages 628–636. Springer, 2019.

[111] Noel C. F. Codella, Veronica Rotemberg, Philipp Tschandl, M. Emre Celebi, Stephen W. Dusza, David Gutman, Brian Helba, Aadi Kalloo, Konstantinos Liopyris, Michael A. Marchetti, Harald Kittler, and Allan Halpern. Skin lesion analysis toward melanoma detection 2018: A challenge hosted by the international skin imaging collaboration (ISIC). *CoRR*, abs/1902.03368, 2019.

[112] Ali Mottaghi and Serena Yeung. Adversarial representation active learning, 2019.

[113] Daniel Gissin and Shai Shalev-Shwartz. Discriminative active learning. *arXiv preprint arXiv:1907.06347*, 2019.

[114] Melanie Ducoffe and Frederic Precioso. Adversarial active learning for deep networks: a margin based approach. *CoRR*, abs/1802.09841, 2018.

[115] Samarth Sinha, Sayna Ebrahimi, and Trevor Darrell. Variational adversarial active learning. *CoRR*, abs/1904.00370, 2019.

[116] Kwan-Young Kim, Dongwon Park, Kwang In Kim, and Se Young Chun. Task-aware variational adversarial active learning. *CoRR*, abs/2002.04709, 2020.

[117] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: A simple and accurate method to fool deep neural networks. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 2574–2582. IEEE Computer Society, 2016.

[118] Yassir Saquil, Kwang In Kim, and Peter M. Hall. Ranking cgans: Subjective control over semantic image attributes. In *British Machine Vision Conference 2018, BMVC 2018, Newcastle, UK, September 3-6, 2018*, page 131. BMVA Press, 2018.

[119] Peng Liu Lei Zhao, Yi Zeng and Xiaohui Su. Active deep learning for hyperspectral image classification with uncertainty learning. *IEEE Geoscience and Remote Sensing Letters*, xx:xx, 2021.

[120] B. H. Menze, A. Jakab, S. Bauer, J. Kalpathy-Cramer, K. Farahani, J. Kirby, Y. Burren, N. Porz, J. Slotboom, R. Wiest, L. Lanczi, E. Gerstner, M. Weber, T. Arbel, B. B. Avants, N. Ayache, P. Buendia, D. L. Collins, N. Cordier, J. J. Corso, A. Criminisi, T. Das, H. Delingette, Ç. Demiralp, C. R. Durst, M. Dojat, S. Doyle, J. Festa, F. Forbes, E. Geremia, B. Glocker, P. Golland, X. Guo, A. Hamamci, K. M. Iftekharuddin, R. Jena, N. M. John, E. Konukoglu, D. Lashkari, J. A. Mariz, R. Meier, S. Pereira, D. Precup, S. J. Price, T. R. Raviv, S. M. S. Reza, M. Ryan, D. Sarikaya, L. Schwartz, H. Shin, J. Shotton, C. A. Silva, N. Sousa, N. K. Subbanna, G. Szekely, T. J. Taylor, O. M. Thomas, N. J. Tustison, G. Unal, F. Vasseur, M. Wintermark, D. H. Ye, L. Zhao, B. Zhao, D. Zikic, M. Prastawa, M. Reyes, and K. Van Leemput. The multimodal brain tumor image segmentation benchmark (brats). *IEEE Transactions on Medical Imaging*, 34(10):1993–2024, 2015.

[121] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Trans. Knowl. Data Eng.*, 22(10):1345–1359, 2010.

[122] Karl R. Weiss, Taghi M. Khoshgoftaar, and Dingding Wang. A survey of transfer learning. *J. Big Data*, 3:9, 2016.

[123] Timothy M. Hospedales, Antreas Antoniou, Paul Micaelli, and Amos J. Storkey. Meta-learning in neural networks: A survey. *CoRR*, abs/2004.05439, 2020.

[124] Pengzhen Ren, Yun Xiao, Xiaojun Chang, Po-Yao Huang, Zhihui Li, Xiaojiang Chen, and Xin Wang. A comprehensive survey of neural architecture search: Challenges and solutions. *CoRR*, abs/2006.02903, 2020.

[125] Mahmut Kaya and Hasan Sakir Bilge. Deep metric learning: A survey. *Symmetry*, 11(9):1066, 2019.

[126] Yaqing Wang, Quanming Yao, James T. Kwok, and Lionel M. Ni. Generalizing from a few examples: A survey on few-shot learning. *ACM Comput. Surv.*, 53(3):63:1–63:34, 2020.

[127] Zhi-Hua Zhou. A brief introduction to weakly supervised learning. *National Science Review*, 5(1):44–53, 2018.

PENG LIU currently is an associate professor at the Institute of Remote Sensing and Digital Earth, Chinese Academy of Sciences. He received the M.S. degree in 2004 and the Ph.D. degree in 2009, both in signal processing, from Chinese Academic of Science. From May 2012 to May 2013, he was with Department of Electrical and Computer Engineering, George Washington University as a Visiting Scholar. He has published 30+ scientific, peer-reviewed papers. He is currently an associate editor of Frontiers in Environmental Science and IEEE Access. He is also the reviewer for Applied Remote Sensing, IEEE JSTAR, Neurocomputing, Signal Processing, etc. His research is focused on big data, sparse represen- tation, compressed sensing, deep learning and their applications to remote sensing data processing.

GUOJIN HE was born in Fujian, China, in 1968. He received the B.Sc. degree in geology from Fuzhou University, Fuzhou, China, in 1989, the M.Sc. degree in remote sensing of geology from China University of Geosciences, Wuhan, China, in 1992, and the Ph.D. degree in geology from the Institute of Geology, Chinese Academy of Sci- ences (CAS), Beijing, China, in 1998. From 1992 to 2007, he was with the Information Process- ing Department, China Remote Sensing Satellite Ground Station (RSGS), CAS. In 2001, he became the Deputy Director of the Information Processing Department, RSGS, CAS. Since 2004, he has been a Professor and the Director of the Information Processing Depart- ment, RSGS, and also the head of the research group of Remote Sensing Information Mining and Intelligent Processing. From 2008 to 2012, he was a Professor and the Director of the Value-added Product Department and the Deputy Director of the Spatial Data Center,Center for Earth Observation and Digital Earth, CAS. Since 2013, he has been a Professor and the Director of the Satellite Data Based Value-added Product Department and the Deputy Director of RSGS, Institute of Remote Sensing and Digital Earth, CAS. A large part of his earlier research dealt with information processing and applications of satellite remote sensing data. His current research interests are focused on optical high-resolution remote sensing image understanding as well as using information retrieved from satellite remote sensing images in combination with other sources of data to support better understanding of the Earth.